A FEASIBILITY STUDY OF AN IN SITU RETORTING PROCESS FOR OIL SHALE

By

ALLEN LAWRENCE BARNES

Bachelor of Petroleum Engineering
Ohio State University
Columbus, Ohio
1955

Master of Science
Pennsylvania State University
University Park, Pennsylvania
1960

Submitted to the faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
DOCTOR OF PHILOSOPHY
May, 1968

A FEASIBILITY STUDY OF AN IN SITU

RETORTING PROCESS FOR OIL SHALE

Thesis Approved:

Allen M. Rowe
Thesis Adviser

J. A. Wickett

J. A. Homer

D. D. Stevenson

N. Durham
Dean of the Graduate College

ACKNOWLEDGMENTS

The author acknowledges advice from committee members: Dr. A. M. Rowe, Dr. J. A. Wiebelt, Professor A. G. Comer and Dr. D. D. Grosvenor.

For technical assistance, the author expresses thanks to Mr. C. Martin.

For manuscript preparation, the author expresses gratitude to Mr. J. F. Ostrander and Mrs. T. I. Asher.

Lastly, the author acknowledges the support of Sinclair Oil & Gas Company through the use of their facilities in conducting this investigation.

TABLE OF CONTENTS

Chapter		Page
I.	INTRODUCTION	1
II.	REVIEW OF LITERATURE	3
III.	THEORETICAL DEVELOPMENT	15
	General Description of Problem	15
	Geometric Considerations	15
	Process Considerations	18
	Flow Path	18
	Investigation Limitations	19
	Important Controllable Parameters	21
	Value of Heat Transfer Study	21
	Plan of Attack	21
	Models I and II	21
	Model III	26
	Model IV	28
	Variable Coefficients	31
	Thermal Capacity and Endothermic Heat Loss	31
	Thermal Conductivity	34
	Description of Process Variables	37
	Temperature	37
	Injection Rate	38
	Well Spacing	38
	Time	38
	Heat Soaking	39
	Air-Oil Ratio	39
IV.	APPROACH TO NUMERICAL SOLUTIONS	41
	Numerical Procedure	41
	Description of Numerical Models	42
	Model I	44
	Model II	49
	Model III	51
	Model IV	53
	Determination of Iteration Parameters	55
	Grid Spacing	55
	Time Incrementation	55
	Tolerance Incrementation	57

Chapter Page

Solution Validity 59
Check of Lauwerier Solution 59
Solutions Using Limiting Coefficients 60
Heat Balance 63
Effect of Assumptions 63
Convection Difference Equation 63
Velocity 65
Infinite Heat Transfer Coefficient 66

V. RESULTS AND DISCUSSION 69

Results from Model I and Model II 69
General Discussion 69
Effect of Temperature 96
Effect of Injection Rate 100
Effect of Soaking 106
Effect of Bounded Systems 109
Air-Oil Ratios 115
Heat Utilization 121
Results from Model IV 122
Results from Model III 127
Significance of Results 133
General Remarks 133
Feasible Air-Oil Ratios 133
Project Design 136
Experimental Pilot 136

VI. SUMMARY AND CONCLUSIONS 138

A SELECTED BIBLIOGRAPHY 141

DEFINITION OF TERMINOLOGY 144

APPENDICES

A. DIFFERENTIAL EQUATIONS 146

Derivation of Convection Equation 146
Derivation of Conduction Equation 147
Derivation of Combined Conduction-Convection Equation 148

B. DIFFERENCE EQUATIONS 151

Variable Spacing Scheme 151
Convection Difference Equations for THERMAB3 and THERMAB4 152
Conduction Difference Equations for THERMAB3 and THERMAB4 155
Conduction Difference Equations for THERMAB1 . . . 160

Chapter Page

Combined Conduction-Convection Difference Equations for THERMAB2 161
Conduction Difference Equations for THERMAB5 . . . 166

C. MATRIX INVERSION TECHNIQUE 171

D. VARIABLE GRID SPACING AND TIME STEPS 173

Variable Grid Spacing 173
Time Step Incrementation 176

E. EXAMPLE OUTPUT DATA 179

Example Output Data from THERMAB3 180
Example Output Data from THERMAB4 182

F. FLOW SHEET AND PROGRAM LISTINGS 183

THERMAB1 184
THERMAB2 189
THERMAB3 199
THERMAB4 215
THERMAB5 232

LIST OF FIGURES

Figure		Page
1.	Schematic Diagram of In Situ Oil Shale Retorting Arrangement	16
2.	Pattern Arrangement of Wells	17
3.	Injected Gas and Hydrocarbon Flow Through Retorted Low Permeability Zone	20
4.	Model I -- Convection Through Constant Width Fracture	22
5.	Model II -- Countercurrent Flow Through Constant Width Fractures	24
6.	Model III -- Convection Through Growing Retorted Limited Permeability Zone	27
7.	Isometric Schematic of Symmetric Element of Conical Ellipsoidal Retorted Region	30
8.	Total Heat Requirement for Retorting Oil Shale	33
9.	Thermal Conductivity Versus Temperature	36
10.	Arrangement of Grid Network for Models I and III	45
11.	Arrangement of Grid Network for Model II	50
12.	Arrangement of Grid Network for Model IV	54
13.	Time Increment and Tolerance Sequence	58
14.	Comparison of Numerical and Analytical Solutions -- Fracture Temperatures	61
15.	Comparison of Numerical and Analytical Solutions -- Positions of 600°F Isotherms	62
16.	Comparison of 700°F Isotherm Positions using Constant and Variable Values of Nonlinear Coefficients	64
17.	Effect of Infinite Heat Transfer Coefficient Assumption	68

Figure		Page
18.	Position of 700°F Isotherm Versus Time -- Run 1	71
19.	Temperature Distribution After 86,100 Hours -- Run 1 . .	72
20.	Position of 700°F Isotherm Versus Time -- Run 2	73
21.	Temperature Distribution After 86,100 Hours -- Run 2 . .	74
22.	Position of 700°F Isotherm Versus Time -- Run 3	75
23.	Temperature Distribution After 86,100 Hours -- Run 3 . .	76
24.	Position of 700°F Isotherm Versus Time -- Run 4	77
25.	Temperature Distribution After 89,100 Hours -- Run 4 . .	78
26.	Position of 700°F Isotherm Versus Time -- Run 5	79
27.	Temperature Distribution After 86,100 Hours -- Run 5 . .	80
28.	Position of 700°F Isotherm Versus Time -- Run 6	81
29.	Temperature Distribution After 86,100 Hours -- Run 6 . .	82
30.	Position of 700°F Isotherm Versus Time -- Run 7	83
31.	Temperature Distribution After 88,600 Hours -- Run 7 . .	84
32.	Position of 700°F Isotherm Versus Time -- Run 8	85
33.	Temperature Distribution After 86,100 Hours -- Run 8 . .	86
34.	Position of 700°F Isotherm Versus Time -- Run 9	87
35.	Temperature Distribution After 83,000 Hours -- Run 9 . .	88
36.	Positions of 700°F Isotherms Versus Time -- Run 10 . . .	89
37.	Temperature Distribution After 86,100 Hours -- Run 10 . .	90
38.	Positions of 700°F Isotherms Versus Time -- Run 11 . . .	91
39.	Temperature Distribution After 86,100 Hours -- Run 11 . .	92
40.	Positions of 700°F Isotherms Versus Time -- Run 12 . . .	93
41.	Temperature Distribution After 84,800 Hours -- Run 12 . .	94
42.	Effect of Injection Temperature Upon Retorting Rate with a Heat Injection Rate of 5000 BTU/hr-ft	97

Figure Page

43. Effect of Injection Temperature Upon Retorting Rate with a Heat Injection Rate of 10,000 BTU/hr-ft 98

44. Effect of Injection Temperature Upon Retorting Rate with a Heat Injection Rate of 20,000 BTU/hr-ft 99

45. Effect of Injection Temperature Upon Retorting Rate with an Air Injection Rate of 278 SCF/hr-ft 101

46. Effect of Injection Temperature Upon Retorting Rate with an Air Injection Rate of 556 SCF/hr-ft 102

47. Effect of Injection Rate Upon Retorting Rate with an Injection Temperature of 1000°F 103

48. Effect of Injection Rate Upon Retorting Rate with an Injection Temperature of 1500°F 104

49. Effect of Injection Rate Upon Retorting Rate with an Injection Temperature of 2000°F 105

50. Effectiveness of Soaking with an Injection Temperature of 1500°F 107

51. Effectiveness of Soaking with an Injection Temperature of 2000°F 108

52. A Comparison of Retorting Rates with Bounded and Unbounded Systems 110

53. A Comparison of Retorting Rates with Bounded and Unbounded Systems 111

54. A Comparison of Retorting Rates with Bounded and Unbounded Systems 112

55. A Comparison of Retorting Rates of Runs Using Bounded Systems 114

56. Air-Oil Ratio for Injection Temperature of 1000°F 116

57. Air-Oil Ratios at Injection Temperature of 1500°F 117

58. Air-Oil Ratios at Injection Temperature of 2000°F 118

59. Air-Oil Ratio for Injection Temperature of 3000°F 119

60. Vertical Profile of 700°F Isotherm at 0 Feet Down Fracture After 89,100 hours 123

61. Vertical Profile of 700°F Isotherm at 100 Feet Down Fracture After 89,100 Hours 124

Figure Page

62. Vertical Profile of 700°F Isotherm at 200 Feet Down Fracture After 89,100 Hours 125

63. Vertical Profile of 700°F Isotherm at 400 Feet Down Fracture After 89,100 Hours 126

64. Position of 700°F Isotherm Versus Time -- Run 14 129

65. Temperature Distribution After 86,100 Hours -- Run 14 . . 130

66. Positions of 700°F Isotherms After 86,100 Hours -- Runs 13 and 14 131

67. Air-Oil Ratios at Injection Temperature of 2000°F -- Runs 13 and 14 132

CHAPTER I

INTRODUCTION

Oil reservoirs are becoming more expensive to discover. The consumption rate of domestic crude reserves has in recent times surpassed the discovery rate of new domestic reserves, and the trend is likely to continue since consumption continues to increase approximately 3 per cent per year. Because of this situation, oil companies are actively trying to develop substitute sources of liquid hydrocarbons.

It has been recognized for some time that oil shale deposits contain a tremendous hydrocarbon reserve. Over half of the known oil shale reserves are located in the United States, most of which lie in the Piceance Creek Basin of Western Colorado. The Colorado oil shale outcrops on the edges of the Piceance Creek Basin. At the outcrops the shale beds are relatively thin, from 25 to 50 feet thick. In the center of the basin the oil shale is as great as 2000 feet thick covered with 1000 feet of overburden. It has been estimated that there are over 1000 billion barrels of oil in shales having an oil content over 15 gallons per ton in this basin. This is some 30 times the present proven crude oil reserves in the United States.

Oil shale does not contain free oil but an organic matter called kerogen. Kerogen is only slightly soluble, if at all, when extracted with ordinary solvents for petroleum at room temperature. Kerogen yields petroleum hydrocarbons by destructive distillation. It must be

heated to approximately 700°F where it decomposes into shale oil, gases and coke.

The United States Bureau of Mines and, more recently, oil companies have conducted considerable research on methods to economically recover oil from this shale. These methods involve the retorting of crushed oil shale at the surface, which is either strip or room and pillar mined. To date with prevailing oil prices this approach has not proved economically attractive.

Another approach to exploit the oil shale deposits, in particular the thick portion that is overlain by 1000 feet of overburden, is to retort the oil shale in place and produce the liquid and gaseous hydrocarbons through wells drilled into the shale. Little research has been done on this approach. There are several variations to the "in situ" retorting approach. These variations can be considered to belong to one of two groups based on the geometry of the system. The first group is in situ retorting conducted in a highly fractured or broken up matrix. The second is in situ retorting conducted through single fractures between producing and injection wells. This dissertation is concerned with the latter group.

The objective of this work was to make a preliminary feasibility study of in situ retorting oil shale by hot gas injection through wells interconnected by single vertical fractures. A knowledge of the temperature distribution under various assumed conditions will indicate whether experimental pilot testing of this approach to in situ retorting is justified. This dissertation is specifically concerned with theoretically predicting temperature distributions and total heat contents in oil shale under various assumed conditions.

CHAPTER II

REVIEW OF LITERATURE

Although there was little literature related to this specific problem, a number of papers on various aspects of oil shale have helped in developing the problem and in formulating a plan of attack.

Jaffe (1,2) gave an extensive description of worldwide oil shale deposits as to their compositions, sizes, locations and the extent to which they have been exploited. A more detailed description of the oil shale in the Green River Formations of Western Colorado was given in Reference 2. Williamson (3,4) described the nature of kerogen and its products of decomposition when retorted. In another paper, Williamson (5) described the various oil shale retorts that have been used in research efforts in this country as well as those used commercially in other countries. Scores of other articles concerning oil shale were referenced in these five papers.

The First Symposium on Oil Shale (6) held in 1964 resulted from the renewed interest in oil shale research. Papers in this symposium were essentially a review of oil shale technology up to that time. In one paper V. Dean Allred described characteristic properties of oil shale that might influence in situ processing. This paper was concerned with the process in a highly fractured matrix where it was visualized to be similar to in situ combustion in oil reservoirs.

The Second Symposium on Oil Shale (7) held in 1965 covered current

research in shale oil recovery. Lekas and Carpenter discussed the possibilities of fracturing oil shale with nuclear explosives for in situ retorting. They stated that information from surface retorts was partially applicable to in situ retorting the resulting fractured systems.

There were numerous U. S. Bureau of Mines reports available on oil shale assays, distillation properties, other chemical properties, mining methods, retorting and physical properties. Hubbard and Robinson (8) made a thermal decomposition study of the kerogen in three Colorado oil shales. They found the rate of decomposition increased with temperature and varied little between shale samples. At temperatures over 750°F essentially all hydrocarbons were driven off.

Sohns et al. (9) found for an experimental entrained-solids, oil-shale retort that as the retorting temperature was raised, the portion of the organic matter converted to gas increased, and that converted to oil decreased. Liquid and gas products became more dehydrogenated. Oils from Colorado oil shale gave good yields of stable gasolines having low sulfur contents. High-temperature retorting resulted in more naphtha production than can be obtained by cracking a conventionally retorted shale oil.

Carbonate decomposition varied from 4 per cent at 1000°F to 89 per cent at 1650°F. Carbonate decomposition was 40 per cent complete at 1400°F and 30 per cent complete at 1300°F. They reported that the heat of retorting to 1200°F for 28 gallons-per-ton (GPT) shale with zero carbonate decomposition was approximately 425 BTU per pound. Carbonate decomposition required additional heat of 238 BTU per pound of shale.

Sohns et al. (10) made a study to determine the total heat necessary to retort Colorado oil shale to produce shale oil and other by-products under conditions that would exist in commercial aboveground retorting methods. This total heat consisted of: 1) the heat content of the mineral and other nonvolatile portions of the shale at the final retort temperature; 2) the heat of reaction resulting from conversion of the organic matter in the shale to gas, oil and coke; 3) the heat of decomposition of that portion of mineral carbonates that decompose under the experimental conditions and other heats of reaction due to changes in the mineral content of the shale; 4) the heat of vaporization of the oil and water; and 5) the heat content of the gas and oil vapors at their exit temperature. The first three items comprised the predominate portion of the heat requirement.

These investigators published plots of heat requirements versus temperature up to 1100°F for 28 GPT shale, for 57 GPT shale and for spent shale. They found for 57 GPT shale that the heat of retorting between 750 to 1100°F ranged from 261 to 378 BTU per pound. Heat of retorting ranged from 104 to 356 BTU per pound for 28 GPT shale between 450 to 1100°F. Heat content above 77°F for spent shale was determined at a temperature interval of 500 to 1100°F and ranged from 94 to 272 BTU per pound. Their data showed that the endothermic reactions occurring during the distillation of kerogen utilize 15 to 30 per cent of the heat supplied.

Gavin and Sharp (11) measured physical properties of oil shale. They found that average specific heats between 20 and 90°C ranged from 0.223 to 0.265 calorie per gm-°C for the samples studied. Thermal conductivities ranged from 0.00382 to 0.00518 calorie per sec-cm-°C.

Somerton and Boozer (12) reported thermal diffusivities and conductivities of some typical sedimentary rocks at temperatures from 200 to 1800°F. They used an unsteady state technique that became less accurate at high temperatures. Over 1500°F the estimated error using this technique could be as large as 20 per cent. Thermal conductivities of all sedimentary rocks followed a similar and significant decreasing trend with temperature. Over 1200°F it was found that thermal conductivities of rock become constant.

Recent work by Somerton (13) showed that the thermal conductivity at room temperature of a typical oil shale averages 0.6 BTU/hr-ft°F parallel to bedding planes and 0.4 BTU/hr-ft°F perpendicular to the bedding planes.

Shaw (14) reported average specific heats for five samples of Colorado oil shale measured at temperatures ranging from 50 to 450°F. Oil shale richnesses ranged from 1 to 89.2 gallons per ton. An empirical equation as a function of oil shale richness and temperature was derived from these measurements.

Somerton (15) presented results of experimental measurements of heat capacities of some sandstones, shales and siltstones. He found measured heat capacities agree closely with values calculated from known chemical compositions of the rocks. He found that heat capacities of these rocks increased from 0.19 to 0.21 BTU/lb-°F at 100°F to 0.28 to 0.30 BTU/lb-°F at 1000°F. At 1000°F the change in thermal capacities of these rocks with temperature had become negligible.

Jukkola et al. (16) measured the thermal decomposition rates of carbonates in oil shale. They found that dolomite in oil shale began to decompose somewhat below 1050°F, while the calcite began to

dissociate in the range of 1150-1200°F. Both dissociation temperatures were nearly 400°F lower than that of the corresponding carbonate in a relatively pure state. These investigators showed that at 1550°F near complete carbonate decomposition occurred in less than one hour in a nitrogen atmosphere at 760 mm pressure. At lower temperatures thermal decomposition rate was considerably slower. The reaction rate was increased by decreasing partial pressures of carbon dioxide.

Feldkirchner and Linden (17) in their study of hydrogasification of oil shale reported carbonate decomposition rates in excess of those reported by Jukkola (15). It was hypothesized that a lower concentration of carbon dioxide partially explained this. Also, ratios of calcite to dolomite were larger in these studies and therefore over 1200°F, decomposition rates would be expected to increase rapidly.

Dannenberg and Matzick (18) included a discussion of carbonate decomposition. They reported that Green River oil shale contains approximately 17 per cent mineral carbonates and that during surface retorting approximately 30 per cent of these carbonates are decomposed. They presented a detailed mineral composition of a typical Green River shale which shows that mineral carbonates are composed of 62 per cent dolomite and 38 per cent calcite. They estimated from this composition that 57,000 BTU per pound mol of carbon dioxide are liberated.

Matzick et al. (19) presented results from research and development carried out by the Bureau of Mines on the gas-combustion oil shale retorting process. This Bulletin covered work from 1944 through 1955. It included much of the material published in other Bureau publications. A 28-GPT shale was reported to be approximately 23 per cent dolomite and 16 per cent calcite. Decomposition of dolomite required 500 BTU per

pound and calcite required 700 BTU. From this it was calculated that a pound of oil shale would require 227 BTU for complete carbonate decomposition.

The reaction time for kerogen decomposition was studied and results showed that 90 per cent decomposition occurred in 10 minutes at 850°F; it occurred in 1000 minutes at 700°F; and it occurred in 5000 minutes at 650°F. Essentially 100 per cent decomposition occurred at 750°F given sufficient time.

Thomas (20) presented results from retorting tests on oil shale conducted under simulated overburden pressures. The range of the investigation was from 25 to 2500 psi at temperatures up to 1000°F. No visible fracturing or exfoliation was found to occur in oil shales retorted under confining stresses of 100 to 2500 psia. Pore structure and permeability were created by removal of oil and water, decomposition of carbonates and the creation of microscopic expansion cracks. Thermal conductivities measured on raw and spent shale varied no more than 13 per cent under stress conditions in excess of 1000 psi. Induced permeabilities averaged about 10 md for overburden pressures in excess of 1000 psi. This compared with cores of spent shale obtained from a field test which showed an 11.3 md average. Qualitative results based on X-ray diffraction patterns indicated carbonate decomposition of dolomite under overburden pressure began at about 650°F compared to the usual 1000°F. This was attributed to the LeChatelier-Braun effect which requires that a stressed system react in a direction that leads to an equilibrium condition.

Several investigators have solved heat transfer problems applicable to rock matrices. Thomas (21) presented a simplified

mathematical model of underground conduction heating. He assumed that a horizontal fracture existed between wells and that the radial temperature distribution could be approximated by a step function along the fracture surface. Heat transfer away from the fracture was assumed to be by one-dimensional vertical conduction and all convection effects were neglected. An analytic solution for this model was obtained. He showed that for a wide range of air injection rates the heat transfer coefficient ranged from 0.5 to 4.0, but it could be considered infinite and very little error results.

Lesser et al. (22) developed a mathematical model that represented the conduction heating of oil shale. They assumed heat would be introduced by the injection of hot condensing gas into horizontal fractures through the formation. The fractures were assumed to be equally spaced; the flow of fluid was linear. The thermal conductivity and thermal capacity of the rock was considered constant and endothermic heat loss due to kerogen distillation was disregarded. Maximum temperature was taken to be 1000°F and thus carbonate decomposition was assumed to be negligible. They presented limited results for a 500-ft system, for 10 and 20-ft fracture spacing, for 1000 lbs/ft^{2}hr injection rate at 1000°F and for two diffusivity values. In all cases several years were required to heat all of the formation to 700°F. Less than 32 per cent of the injected heat was utilized for heating the shale.

Lauwerier (23) solved a heat transfer problem analytically where convection heat transfer occurred in one dimension and heat was given up by conduction into the bounding media. Heat transfer in the bounding media was assumed to be by conduction only in a direction perpendicular to the direction of flow. The downstream boundary was infinite

in the direction of flow and coefficients were constant. He used the Laplace transformation to obtain the solution. This analytical solution has been useful in checking the accuracy of numerical solutions.

In order to conduct an adequate theoretical heat transfer study of the proposed problem it was necessary to solve nonlinear partial differential equations that described conduction and convection heat transfer. These equations had to be solved numerically. A review of numerical techniques was made to determine which technique to use.

Birkhoff (24) summarized the current status of partial difference methods for computing approximate numerical solutions of boundary value problems. He pointed out that the practical solution of most boundary value problems by difference methods was still more of an art than a science and that theoretical error bounds usually are hopelessly pessimistic, except for difference methods having a low order of accuracy. He stated that, although the order of magnitude of truncation errors could be computed theoretically, the actual bound depended on the size of the derivatives usually not known beforehand. For mesh lengths used in practice, cumulative round-off errors were usually much smaller than truncation errors. If round-off errors accumulated significantly, it was usually because they were amplified by "divergent" computing methods. He suggested that the usefulness of partial difference methods owed far less to the science of numerical analysis than it owed to improvements in computing machine technology.

Birkhoff pointed out that the Crank-Nicolson method was not well adapted to the solution of parabolic differential equations involving two-space variables because the method was implicit and therefore it converged very slowly for fine meshes. He suggested that the ADIP

(alternating-direction implicit procedure) was an excellent practical solution to circumvent this difficulty.

Peaceman and Rachford (25) described the alternating-direction implicit procedure as it was related to the conduction heat equation with constant coefficients. It involved difference equations similar to those for the implicit method; however, only one of the second derivatives, say $\partial^2 T/\partial x^2$, was replaced by a second difference evaluated in terms of the unknown temperature while the other derivative, $\partial^2 T/\partial y^2$, was replaced by a second difference evaluated in terms of known values of the temperature. Sets of simultaneous equations were formed that could be solved easily by matrix inversion. These equations were said to be implicit in the x direction. If the procedure was repeated for a second time step of equal size, with the difference equations implicit in the y direction, the overall procedure for the two time steps was stable for any given time step. Their analysis of this procedure showed that it required fewer operations than either the implicit or explicit approach. Example solutions of the two-dimensional steady and unsteady state heat equations were given for rectangular boundaries.

Douglas and Peaceman (26) extended the applicability of the ADIP by presenting solutions of more practical problems. The first example was that of flow around a corner which showed that a more complicated region than a rectangle can be treated. A problem involving a radiation boundary condition, which was nonlinear, was given; and therefore the method was extended to more general equations. The third example involved point heat sources and sinks in an elliptical region and so this extended the method to treat curved boundaries and singular points.

ADIP required less calculating time for these solutions than did other methods.

Douglas (27) discussed the problem of stability and error growth for linear parabolic and hyperbolic equations. He showed for wide classes of difference analogies of linear parabolic and hperbolic differential equations with twice-dependent coefficients that stability implies convergence in the mean.

Brian (28) presented a finite difference method for solving three-dimensional transient heat conduction problems. The method was a modification of ADIP preserving its advantages of unconditional stability and simplicity while achieving the higher-order accuracy of Crank-Nicholson. The first three steps of the method were equivalent to ADIP for half of a time step. In the fourth step the temperatures at the advanced time level N+1 were computed explicitly with the distance differences formulated in terms of the $T_x^{n+\frac{1}{2}}$, $T_y^{n+\frac{1}{2}}$ and $T_z^{n+\frac{1}{2}}$ values found in the first three steps. The form of the fourth step was similar to the Crank-Nicholson form. The method had not been proven by example when the article was written.

Bruce et al. (29) developed a numerical procedure to solve a second order nonlinear partial differential equation which described unsteady state gas flow. The Crank-Nicholson method was used. A difference equation was written for each point in the domain and thus a matrix of equations was solved. The method of solution consisted of factoring the nonlinear terms into a product of assumed values of P and unknown values of P. The resulting linear set of simultaneous equations was solved by a Gaussian type elimination procedure for the unknown P's. These values were then used as assumed values of P for

the next iteration. The iterations were continued until the unknown values were equal to the assumed values. The Gaussian type elimination procedure was first suggested by L. H. Thomas of the Watson Scientific Computing Laboratory. The procedure avoided error growth associated with the back solution of the plain Gaussian elimination and also minimized the storage problems in machine computation. This elimination procedure (Thomas' algorithm) was applicable for a tridiagonal matrix. Proof of this algorithm was given in this paper.

Blair and Peaceman (30) presented experimental verification of numerical solutions of nonlinear equations which described a gas drive in an oil reservoir. The numerical solutions presented were obtained using the ADIP. A scale fluid sand model was used in the experimental tests. Excellent agreement was obtained for all tests presented. A convenient method to handle variable grid spacing in the numerical equations was also presented.

Stone and Brian (31) gave a method of analyzing the accuracy of finite-difference analogies of the differential systems which characterized some convective transport problems. The method could be used to treat equations containing both first and second order terms, but was limited to consideration of linear differential systems.

Larkin (32) showed that the alternating-direction explicit procedure (ADEP) suggested by Saul'ev could be extended to two or three dimensions. The technique for approximating the diffusion equation was explicit, stable for time steps of any size, and he concluded it was competitive with other methods of approximating the diffusion equation. Because it was an explicit method, it held a speed advantage over implicit computations over a single time level.

Quon et al. (33) used the ADEP to solve a nonlinear partial differential equation describing two-dimensional gas reservoir behavior. They found the method to be straightforward and they encountered no calculational problems. They concluded the method offered advantage over the forward difference method from a stability standpoint, thus permitting larger time steps. They said there was an advantage over the ADIP from a computational point of view since only one nonlinear algebraic equation had to be solved at a time, rather than a set of equations, representing a row or a column of grid points.

Coats and Terhune (34) compared the ADIP and ADEP for two-dimensional flow calculations. They performed analyses and example applications to compare accuracy and computing speed. They showed that the truncation error for the ADEP was an order of magnitude larger than that for the ADIP. The ADEP was found to be nonconservative in that it failed to preserve no-flow conditions at exterior boundaries. This caused errors in potential and in material balance which could become extremely severe if wells were near the insulated boundaries. Comparison of two example solutions showed ADIP accuracy superior to ADEP; however, the ADIP required about 60 per cent more computing time than the ADEP.

CHAPTER III

THEORETICAL DEVELOPMENT

General Description of Problem

Geometric Considerations

In situ retorting of oil shale requires that communication be established between wells drilled into the shale matrix because oil shale has no natural matrix permeability. This research is concerned with a process using a single fracture communication between wells. These fractures are induced hydraulically at pressures of approximately 2200 psia, and they have been observed to leave the well bore in a vertical orientation. The fracture height can be controlled within a range of a few to 100 feet.

Figure 1 illustrates the geometry of this problem. Gas is injected down the injection well through the fracture in both directions to two producers. A combustible gas mixture could be burned at the surface and the hot products of combustion injected or the combustible mixture could be injected and then ignited downhole near the fracture. The hot gas gives up heat to the shale matrix as it travels through the fracture and it is recovered at the producing wells.

Figure 2 is an areal view of the pattern arrangement of a system of wells. It is expected parallel lines of wells would be drilled, all reaching the same horizon in the shale. The wells would be

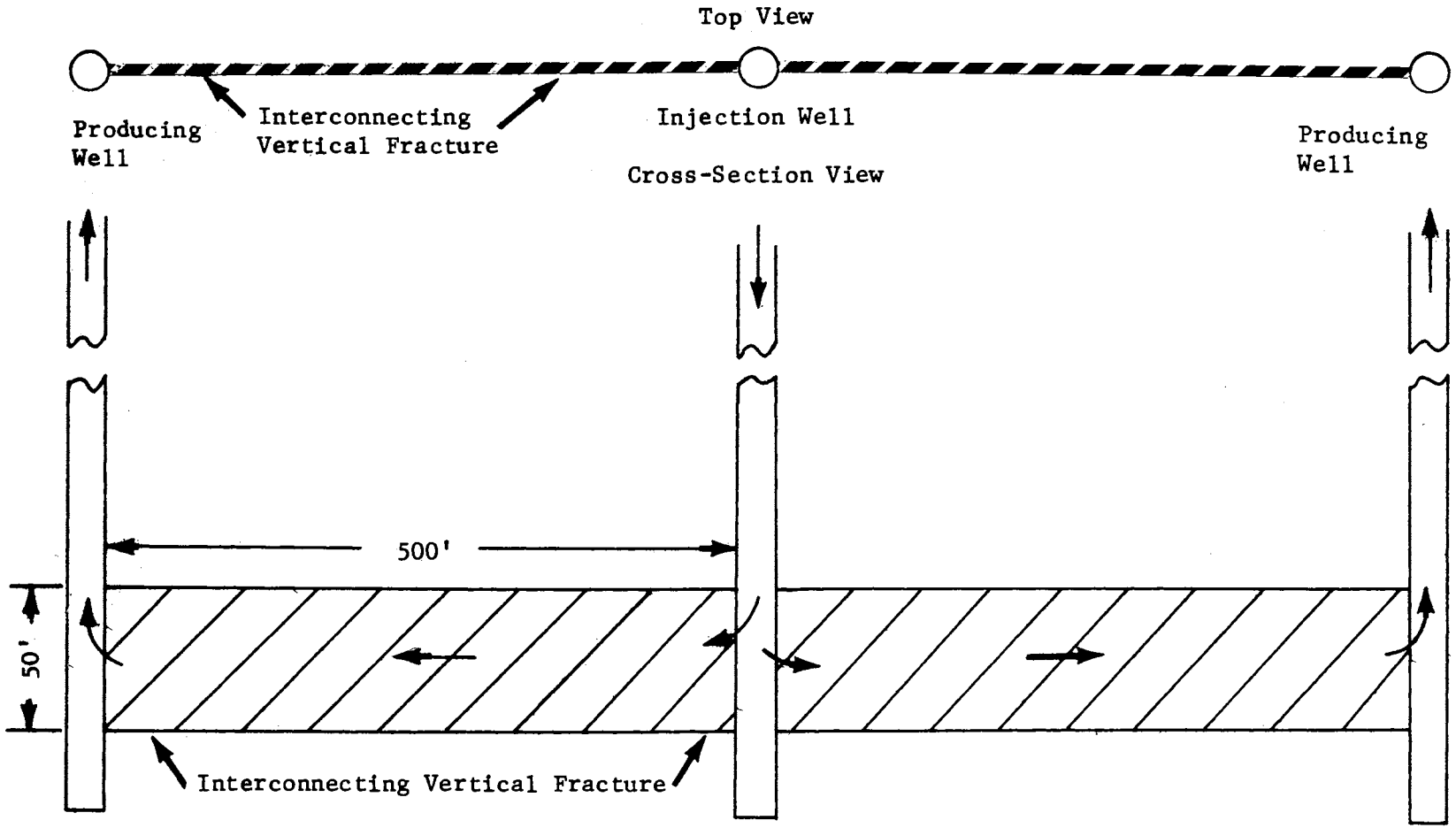


Figure 1. Schematic Diagram of In Situ Oil Shale Retorting Arrangement

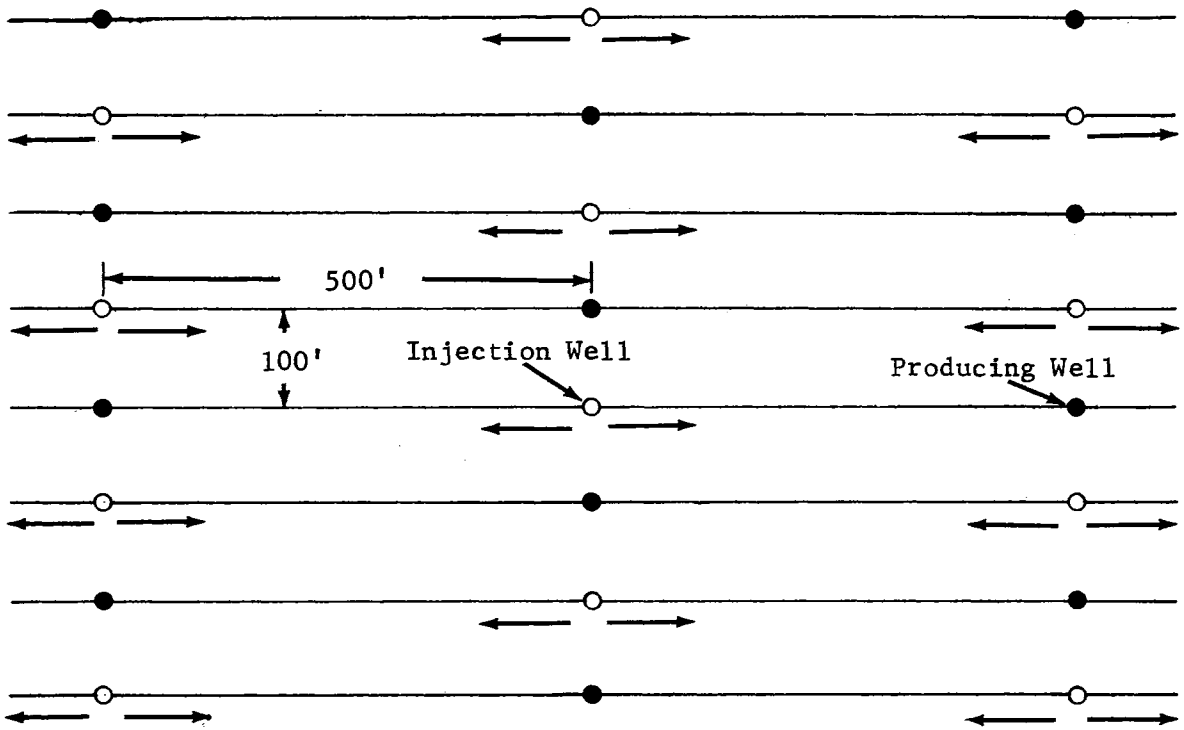


Figure 2. Pattern Arrangement of Wells

alternately producing and injection wells. The spacing between wells and the lines would be determined ultimately by the optimum economic life of the process. A staggered injection-producing well sequence from one line to the next, thus resulting in a countercurrent flow arrangement, would be the most efficient pattern based on heat transfer considerations.

Process Considerations

Essentially all liquid and gaseous hydrocarbons are released from oil shale at or above 700°F; therefore the rate of movement of the 700°F isotherm in the matrix is an important consideration. The rate of movement of this isotherm is not simply a function of the source temperature and rate of heat injection. A complicated set of endothermic heat reactions takes place as oil shale is heated. Endothermic heat loss is experienced from 600 to 800°F when the kerogen decomposes into oil, gas and coke. Also, endothermic heat loss occurs as the mineral carbonates of the shale decompose. Under surface retorting conditions most of this carbonate decomposition takes place over 1000°F and it is essentially complete above 1600°F. Thermal conductivity of sedimentary rocks is known to decrease significantly with temperature, and the thermal capacity of shale has been shown (10,14) to increase somewhat with temperature. These variable properties of oil shale complicate the problem as to the optimum fluid injection temperature.

Flow Path

As the hot injected gases retort oil shale next to the fracture, this shale is expected to develop a limited matrix permeability on the

order of 10-15 md (20). Figure 3 is a schematic illustrating this. Although limited injected gas flow may occur through this zone, most of the gas flow is expected to remain in the high capacity fracture until the retorted zone reaches the producing well. The fracture must remain open ahead of the retorted zone to maintain a flow path to the producer. Because of this requirement, injection pressures would remain high, in order to maintain a fracture along the entire flow path, through both retorted and nonretorted portions. Permeability in the retorted zone should be sufficient so that gases and vapors from the interior shale undergoing retorting can escape to the fracture and be carried to the producing well.

Investigation Limitations

The process of in place oil shale retorting is extremely complex. Therefore, this investigation was limited to heat transfer. In particular, the limitations of this investigation were:

1. Pressure was not considered. It was assumed injection rates used in this study were possible. Any effect of pressure on carbonate decomposition and retorting temperature was not taken into account.
2. Retorting efficiency and ultimate recovery efficiency were not studied in any aspect, although outflow temperatures were limited to 900°F. This was done because it was recognized sustained high temperatures would crack heavy hydrocarbons to gases or even to carbon.
3. Oil richness was considered only with regard to heat retorting requirements. A 28-GPT sample was used for this.

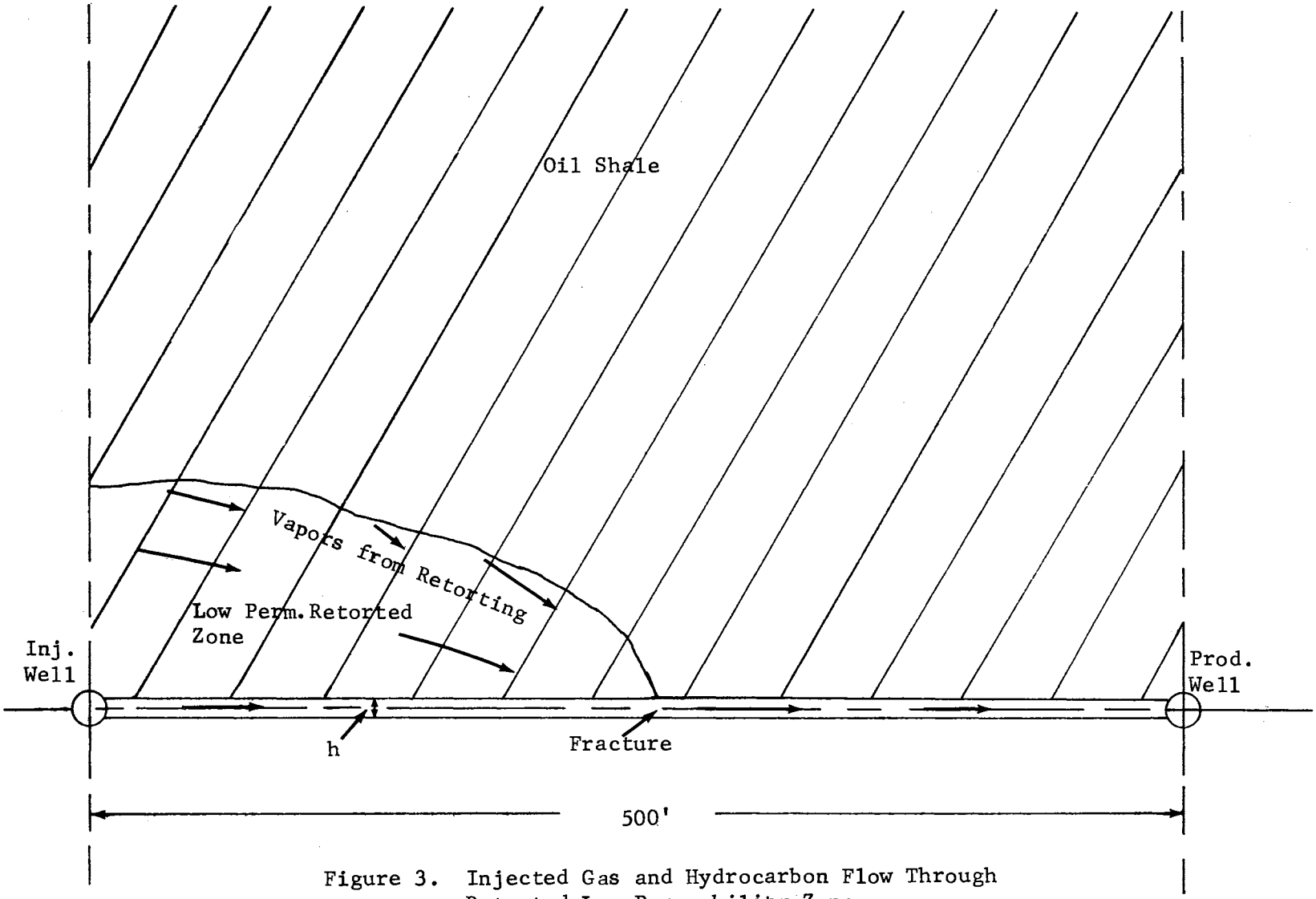


Figure 3. Injected Gas and Hydrocarbon Flow Through Retorted Low Permeability Zone

4. The effect of vaporization and condensation on heat transfer was not considered explicitly. Mass transfer within or out of the shale was not taken into account in the heat transfer equations written for this problem.

Important Controllable Variables

Controllable variables that affect heat transfer in this process are injection temperature, injection rate, well spacing and time. Carbonate decomposition has an effect on the process but its effect will be implicit with the temperature effect.

Value of Heat Transfer Study

A knowledge of the temperature distribution under various conditions will determine the feasibility of this retorting process from a heat transfer standpoint. The most optimistic air-oil ratio can be predicted from this knowledge. Therefore, heat transfer results can be used to determine if large expenditures for an experimental pilot test are warranted. The pilot test operation is necessary to determine optimum pressure and recovery operating conditions and to evaluate the overall economics of a commercial project.

Plan of Attack

Four models were used in this heat transfer study.

Models I and II

Figure 4 is a schematic of Model I. This was the basic model used to investigate the feasibility of oil shale in situ retorting by use

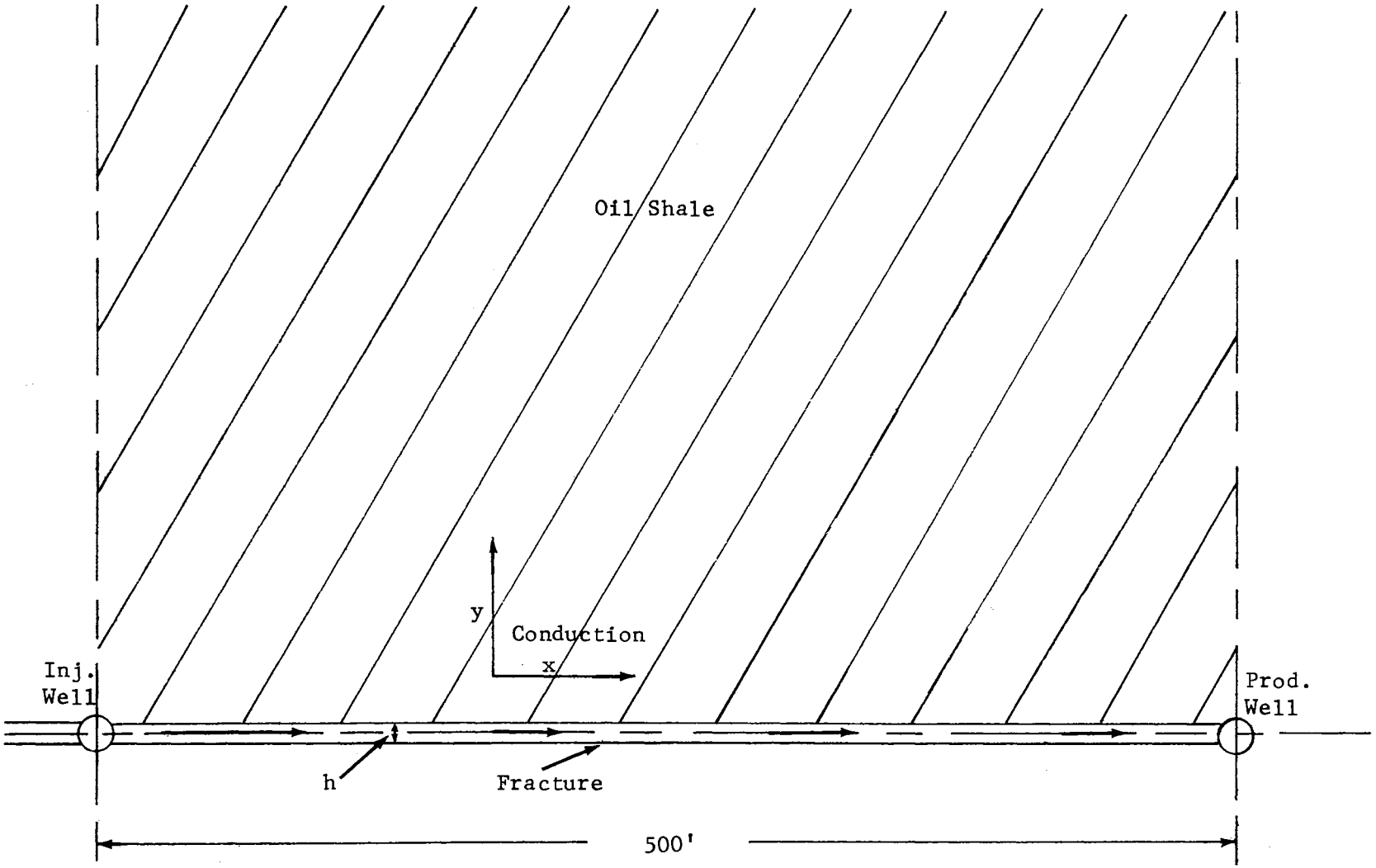


Figure 4. Model I -- Convection through Constant Width Fracture

of single fractures between wells. Heat is convected through the fracture from the injection well to the producing well. Heat is transferred to the shale by conduction. Conduction heat transfer is considered only in the horizontal direction, both parallel and perpendicular to the fracture. Vertical heat conduction is not considered in this model because of computer size and time limitations. The section cross-hatched in Figure 4 represents a symmetric element in a linear array of such a well system. Therefore it is necessary to consider only one such element because adiabatic boundary conditions exist on three sides and an infinite boundary exists in the y direction perpendicular to the fracture.

Model II shown in Figure 5 is a representative element in a symmetric areal array of wells. Figure 5 is similar to Figure 4 except two sides are bounded by fractures which carry hot gas flowing in opposite directions. In this model only the cross-hatched element between fractures is considered, as the no-heat flow condition exists on all sides. In both of these models the flow rate of only one half the fracture is considered as demanded by the adiabatic boundary condition of the symmetric elements.

The following two nonlinear partial differential equations describe heat transfer in these two models.

$$\frac{h}{2}\,\rho_g c_g v_g \frac{\partial T_g}{\partial x} + \frac{h}{2}\,\rho_g c_g \frac{\partial T_g}{\partial t} = K(T) \left.\frac{\partial T_s}{\partial y}\right|_{y=\frac{h}{2}} . \qquad \text{(A-1)}$$

$$\frac{\partial}{\partial x}\left[K_x(T)\,\frac{\partial T_s}{\partial x}\right] + \frac{\partial}{\partial y}\left[K_y(T)\,\frac{\partial T_s}{\partial y}\right] - q(T) = \rho_s \frac{\partial}{\partial t}\left[c_s(T)\,T_s\right] . \qquad \text{(A-2)}$$

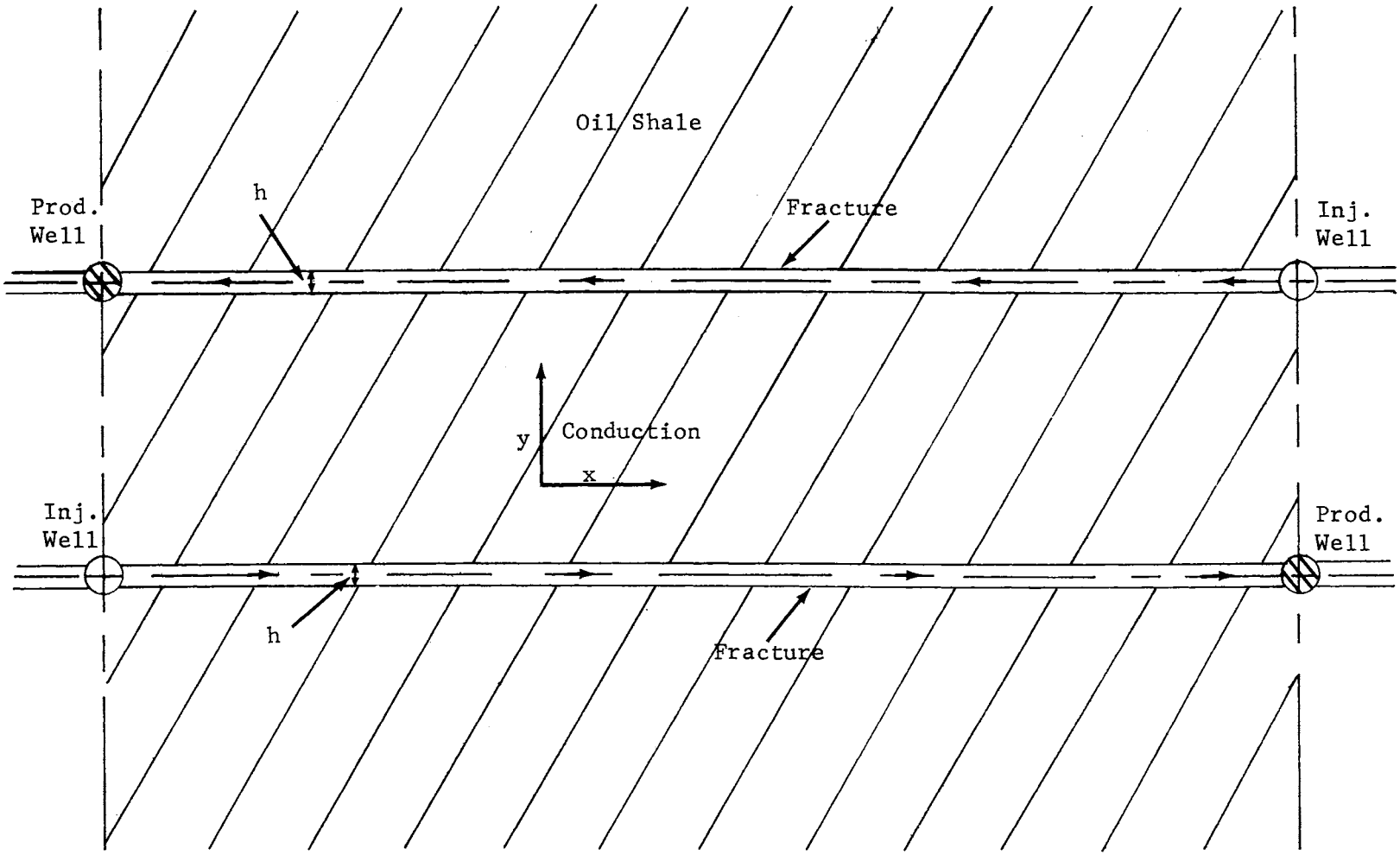


Figure 5. Model II -- Countercurrent Flow through Constant Width Fractures

Derivations of these equations are shown in Appendix A. Equation (A-1) describes convection heat transfer in the fracture with heat loss into the shale. The product $\rho_g v_g$ represents the mass velocity which is considered constant. This is based on the assumption that near steady state conditions occur in the fracture. This condition is approached more nearly with time as the fracture is heated. The thermal capacity, c_g, of the gas is considered to be constant. With these assumptions the rate of heat transfer from an injection well can be taken as constant. The Model I boundary condition for the fracture is:

$$\text{At } x = 0,\ T = T_{injection} .$$

For Model II we have in addition:

$$\text{At } x = x_I,\ T = T_{injection} .$$

Also, the heat transfer coefficient between the gas stream and fracture surface was considered infinite. This is equivalent to saying the rock surface and the gas stream temperatures are identical at any point along the fracture. Thomas (21) has shown this assumption leads to negligible error.

Equation (A-2) describes conduction heat transfer in the shale. Thermal conductivity and thermal capacity of the shale are considered a function of temperature. Endothermic heat losses are represented by q, which is also a function of temperature. Boundary and initial conditions in the shale for Model I are:

1. At $t = 0,\ T = T_0 \quad 0 \le x \le x_I,\ 0 \le y \le \infty$.
2. At $x = 0,\ \frac{\partial T}{\partial x} = 0 \quad \frac{h}{2} \le y \le \infty$.

3. At $x = x_I$, $\frac{\partial T}{\partial x} = 0$ $\quad \frac{h}{2} \leq y \leq \infty$.

4. At $y = \frac{h}{2}$, T = fracture temperature $\quad 0 \leq x \leq x_I$.

5. At $y = \infty$, $\frac{\partial T}{\partial y} = 0$ $\quad 0 \leq x \leq x_I$.

For Model II the boundary conditions are:

1. At $t = 0$, $T = T_0$ $\quad 0 \leq x \leq x_I$, $0 \leq y \leq y_J$.

2. At $x = 0$, $\frac{\partial T}{\partial x} = 0$ $\quad \frac{h}{2} \leq y \leq y_{J-\frac{h}{2}}$.

3. At $x = x_I$, $\frac{\partial T}{\partial x} = 0$ $\quad \frac{h}{2} \leq y \leq y_{J-\frac{h}{2}}$.

4. At $y = \frac{h}{2}$, T = fracture temperature $\quad 0 \leq x \leq x_I$.

5. At $y = y_{J-\frac{h}{2}}$, T = fracture temperature $\quad 0 \leq x \leq x_I$.

Model III

This model was designed to investigate the change in heat transfer that might result from limited convective flow through the retorted zone. Figure 6 is a schematic of Model III. Heat transfer by conduction is considered in the horizontal direction both parallel and perpendicular to the fracture. Vertical heat conduction is assumed zero. Convection heat transfer occurs not only through the fracture but to a predetermined limited extent through the retorted shale adjacent to the fracture. As more shale is heated over 700°F and the retorted area grows, convection heat transfer is assumed to occur in the growing permeable zone. The proportional amounts of gas flowing in the fracture and permeable zone would change with the varying pressure conditions, and therefore these quantities must be arbitrarily set since pressure is not considered in this study. The

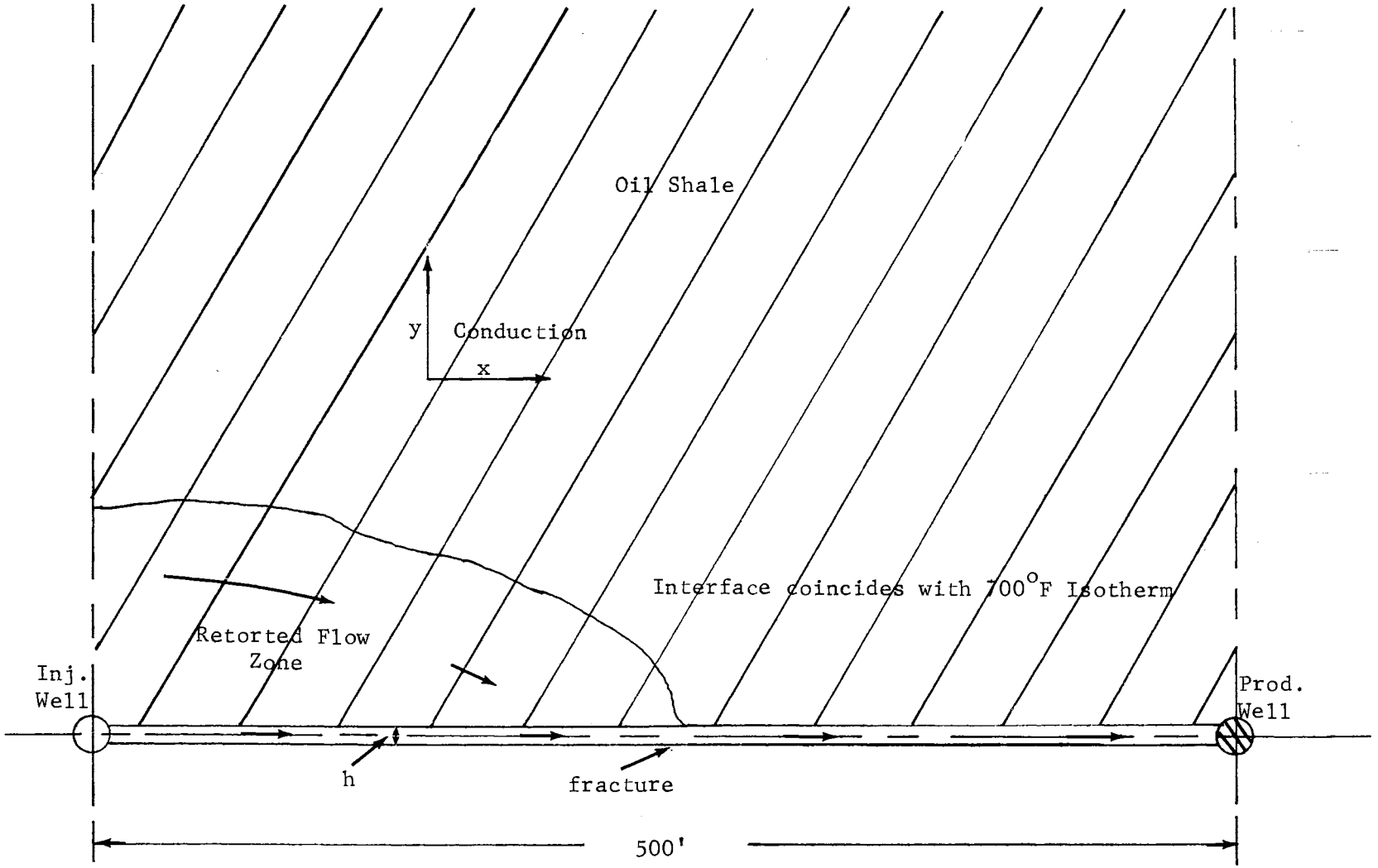


Figure 6. Model III -- Convection through Growing
Retorted Limited Permeability Zone

cross-hatched section in Figure 6 represents a symmetric section in a linear array of wells. Adiabatic boundary conditions exist on three sides and an infinite boundary exists on the fourth side.

The following equation describes heat transfer in this model and is derived in Appendix A.

$$\frac{\partial}{\partial x}\left[K_x(T)\ \frac{\partial T_s}{\partial x}\right] + \frac{\partial}{\partial y}\left[K_y(T)\ \frac{\partial T_s}{\partial y}\right] - q(T) - \alpha\ \frac{\partial T_s}{\partial x} = \rho_s\ \frac{\partial}{\partial t}\left[c_s(T)\ T_s\right] . \tag{A-3}$$

This equation describes one-dimensional convection heat transfer and two-dimensional conduction heat transfer. In the portion of the matrix where convection is not present, the value of α is zero. The assumptions made for Models I and II regarding the convective parameters were also made for this model. Thermal conductivity, thermal capacity and endothermic heat loss were treated as functions of temperature. Boundary and initial conditions for this model are:

1. At $t = 0$, $T = T_0$ $\qquad 0 \le x \le_I,\ 0 \le y \le \infty$.
2. At $x = 0$, $\frac{\partial T}{\partial x} = 0$ $\qquad \frac{h}{2} \le y \le \infty$.
3. At $x = 0$, $T = T_{constant}$ $\qquad 0 \le y \le \frac{h}{2}$.
4. At $x = x_I$, $\frac{\partial T}{\partial x} = 0$ $\qquad 0 \le y \le \infty$.
5. At $y = 0$, $\frac{\partial T}{\partial y} = 0$ $\qquad 0 \le x \le x_I$.
6. At $y = \infty$, $\frac{\partial T}{\partial y} = 0$ $\qquad 0 \le x \le x_I$.

Model IV

Model IV was designed to evaluate the assumption of no vertical conduction heat transfer in the first three models. In the real physical situation heat conduction will occur in three directions and

therefore the configuration of the retorted zone (which coincides with the 700°F isotherm) will be a conical ellipsoid whose cross section diminishes in the downstream fracture direction. A symmetric element of the ellipsoid is illustrated in Figure 7. If it is assumed that there is no heat flow in the vertical direction then the 700°F isotherm would be a straight line in the z direction. But if heat flow occurs in the z direction, the resulting isotherm would be as illustrated. The objective of Model IV is to compare the relative magnitudes of cross-hatched A and B and A' and B', etc. With this comparison the effect of the assumption that no heat flows vertically can be qualitatively evaluated.

In order to determine the profiles in the y-z direction, a conduction model was considered which used a heat source based on the temperature history at selected positions along the fracture of Model I. The use of this temperature history involves the assumption that the fracture temperature histories of a two- and three-dimensional problem would be the same. This assumption is probably good after the temperature envelope has progressed some distance from the fracture. In any event results from Model IV are considered only qualitatively to get an estimate of the error involved in the two-dimensional conduction heat flow assumption.

With this background Model IV involved the solution of

$$\frac{\partial}{\partial z}\left[K_z(T)\ \frac{\partial T_s}{\partial z}\right] + \frac{\partial}{\partial y}\left[K_y(T)\ \frac{\partial T_s}{\partial y}\right] - q(T) = \rho_s \frac{\partial}{\partial t}\left[c_s(T)\ T_s\right],$$

with boundary conditions of:

1. At $t = 0$, $T = T_0$ $\qquad 0 \le z \le \infty,\ 0 \le y \le \infty$.
2. At $z = 0$, $\frac{\partial T}{\partial z} = 0$ $\qquad \frac{h}{2} \le y \le \infty$.

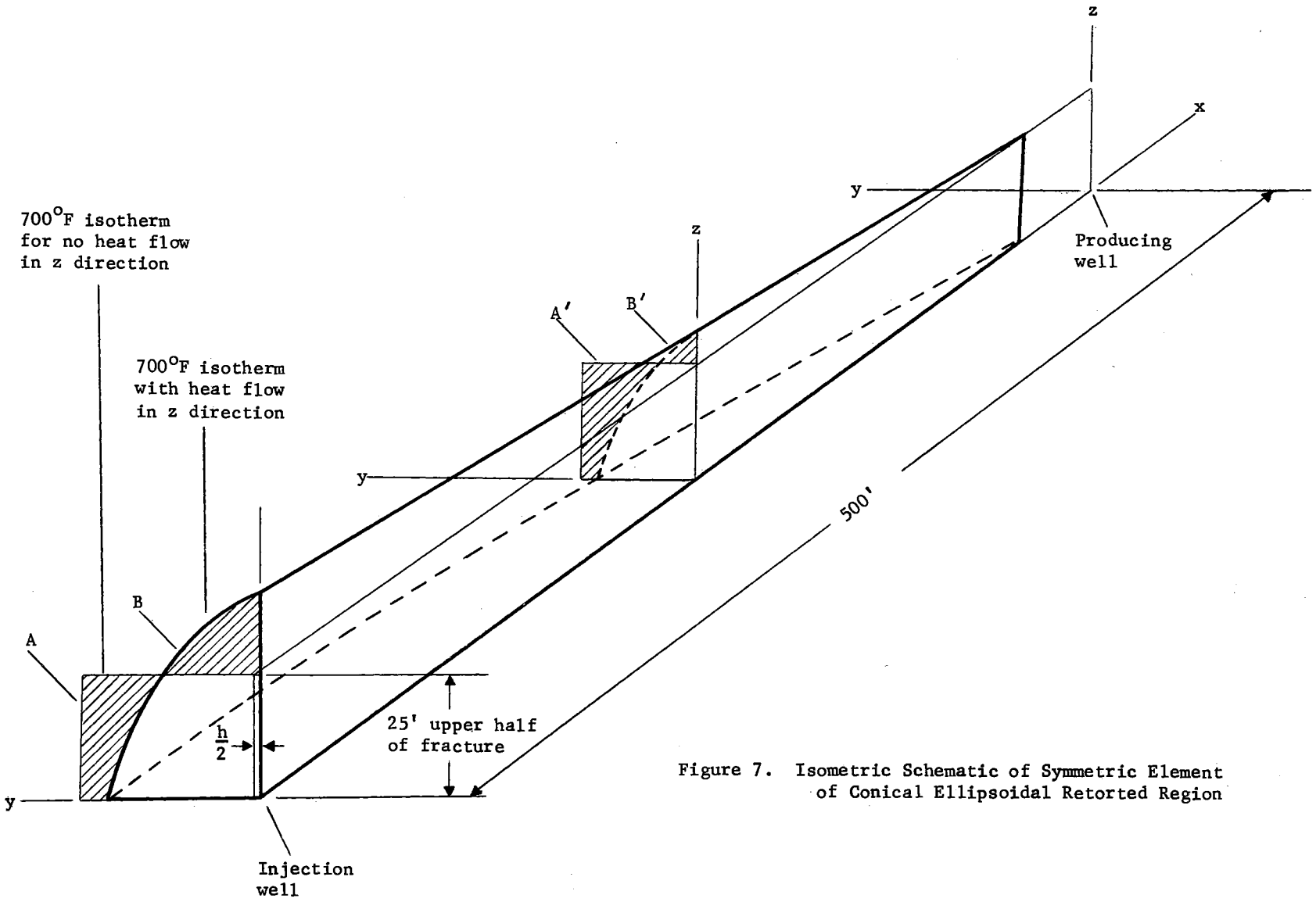


Figure 7. Isometric Schematic of Symmetric Element of Conical Ellipsoidal Retorted Region

3. At $z = \infty$, $\frac{\partial T}{\partial z} = 0$ $\frac{h}{2} \leq y \leq \infty$.

4. At $y = \frac{h}{2}$ T = Fracture temperature from selected points in Model I. $0 \leq z \leq 25$.

5. At $y = \frac{h}{2}$, $\frac{\partial T}{\partial y} = 0$ $25 \leq z \leq \infty$.

6. At $y = \infty$, $\frac{\partial T}{\partial y} = 0$ $0 \leq z \leq \infty$.

Variable Coefficients

Thermal Capacity and Endothermic Heat Loss

The function of oil shale thermal capacity, $c_s(T)$, versus temperature is difficult to describe over a wide temperature range. The thermal capacity is easily measured up to 450°F. Beyond this temperature the character of oil shale begins to change as liquids are driven off and mineral composition is altered. In the 600 to 800°F range kerogen is distilled off and therefore there is a considerable endothermic heat requirement. Carbonate decomposition begins over 900°F and continues to at least 1600°F at atmospheric pressure. These endothermic heat requirements are comparable to the total heat content of the rock and therefore must be considered in this heat transfer problem.

Instead of obtaining a relationship of the heat capacity of oil shale versus temperature, it is more convenient to measure total heat requirement to raise a unit mass of oil shale to a specific temperature. This total heat requirement represents the heat content of the rock, endothermic heat loss, as well as heats of vaporization and the heat content of gas and vapors driven off. These latter two components of the total heat requirement are relatively minor; however their inclusion helps to compensate for the errors in the assumption that no mass

flow occurs out of the matrix. This assumption was made in the derivation of the conduction heat transfer equation in Appendix A. These quantities can be treated together as long as the temperature at any point increases monotonically for the duration of a calculation. If temperature decreases then only rock heat content can be considered. Since most of the runs in this work involved increasing temperatures only, the heat content of the rock and the endothermic heat loss were handled as a single quantity. For those cases where "heat soaking" was studied, a rock heat content curve was used. For these cases, temperatures were declining. Curves of total heat requirement and oil shale heat content are shown in Figure 8. These curves were used in all heat transfer calculations in this work.

The curves were constructed on information from the literature. The heat content of the shale up to 450°F was obtained from Shaw (14). From 450 to 1100°F the total heat requirement curve was based on data from Sohns et al. (10). In this work oil shale heat requirements for spent shale, 28 GPT shale and 57 GPT shale at atmospheric pressures were presented. The 28 GPT data were used here. The Shaw (14) data were for 30 GPT shale. Calculations by Sohns et al. (10) indicated that at 1100°F total heat requirement of the 28 GPT shale above 77°F was 358 BTU/lb while it was 272 BTU/lb for spent shale. The difference of 86 BTU/lb represented heat loss up to 1100°F. Dannenberg and Matzick (18) presented material balance calculations, indicating that heat losses other than by carbonate decomposition would be 58 BTU/lb, thus indicating that at 1100°F, 28 BTU/lb was due to carbonate decomposition. Dannenberg and Matzick (18) and Matzick et al. (19) presented

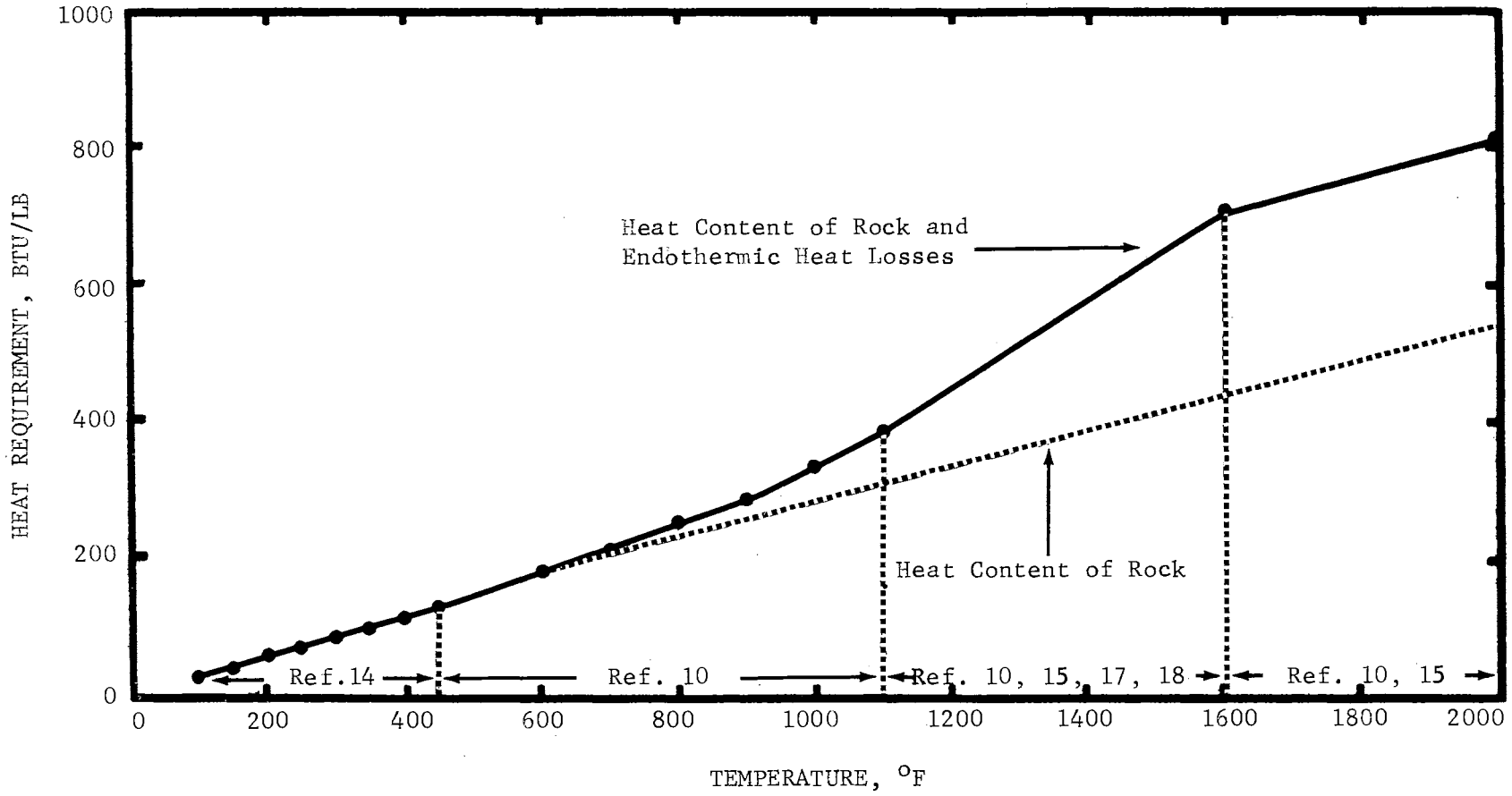


Figure 8. Total Heat Requirement for Retorting Oil Shale

data indicating that the total carbonate decomposition endothermic heat requirement was 220 BTU/lb for a shale containing 17 per cent by weight mineral carbon dioxide.

The total heat requirement curve from 1100 to 1600°F was approximated by assuming that the remaining carbonate decomposition would take place in this range and that the endothermic heat requirement for this decomposition would occur linearly. In addition to this heat requirement, it was assumed that the thermal capacity of spent shale at 1100°F would remain constant at 0.268 BTU/lb-°F. This assumption is supported by work of Somerton (15) on sedimentary rocks. Combining the endothermic heat and spent shale heat content, the curve shown in Figure 8 was obtained. Over 1600°F the increase in heat requirement is due solely to the 0.268 BTU/lb-°F thermal capacity.

The heat content curve was assumed to be equivalent to the total heat curve up to 600°F. At 1100°F spent shale has a heat content above 0°F of 295 BTU/lb. From 600 to 1100°F, it was assumed the increase in heat capacity was linear. This assumption is supported by Sohns et al. (10) data. Above 1100°F a thermal capacity of 0.268 BTU/lb-°F was used.

Thermal Conductivity

A relationship of the thermal conductivity of oil shale as a function of temperature is not available. Such a relationship would be extremely difficult to measure because the oil shale composition changes markedly with temperature. The thermal conductivity of oil shale has been measured at low temperatures by several investigators (11), (13). Gavin and Sharp (11) report that conductivities measured

on 42 GPT shale at temperatures between 77 and 167°F ranged from 0.76 to 1.25 BTU/ft-hr°F. They reported data on a number of substances including limestone whose conductivities at a temperature of 200°F ranged from 0.94 to 1.18 BTU/ft-hr°F. Recently, Somerton (13) reported the thermal conductivity of 30 GPT oil shale parallel to the bedding planes to be 0.63 BTU/ft-hr°F at room temperature. He measured the thermal conductivity of the same samples perpendicular to the bedding planes to be 0.45 BTU/ft-hr°F.

Somerton and Boozer (12) reported thermal conductivities of sedimentary rocks as functions of temperature from 700 to 1800°F. They found that the thermal conducitivities of all rocks showed a similar dependency on temperature and most of them at 200°F ranged from 0.8 to 1.09 BTU/ft-hr°F. Their values dropped to approximately 0.36 at 1200°F. Over 1200°F conductivities were constant.

It was decided to use the thermal conductivity of limestone for this work because information on oil shale conductivity as a function of temperature was not available. The limestone values were used because of the similarity of the low temperature values of both limestone and oil shale as reported by Gavin and Sharp (11). Also, the mineral content of limestone and retorted oil shale are more alike than any of the other sedimentary rocks. The mineral content of oil shale is approximately 50 per cent dolomite and calcite. The thermal conductivity relationship of limestone as a function of temperature used for this research work is shown in Figure 9. Values range from 0.93 BTU/ft-hr°F at 200°F to 0.36 BTU/ft-hr°F at 1200°F. At temperatures less than 200°F a constant value of 0.93 BTU/ft-hr°F was used. For temperatures in excess of 1200°F it was assumed that thermal

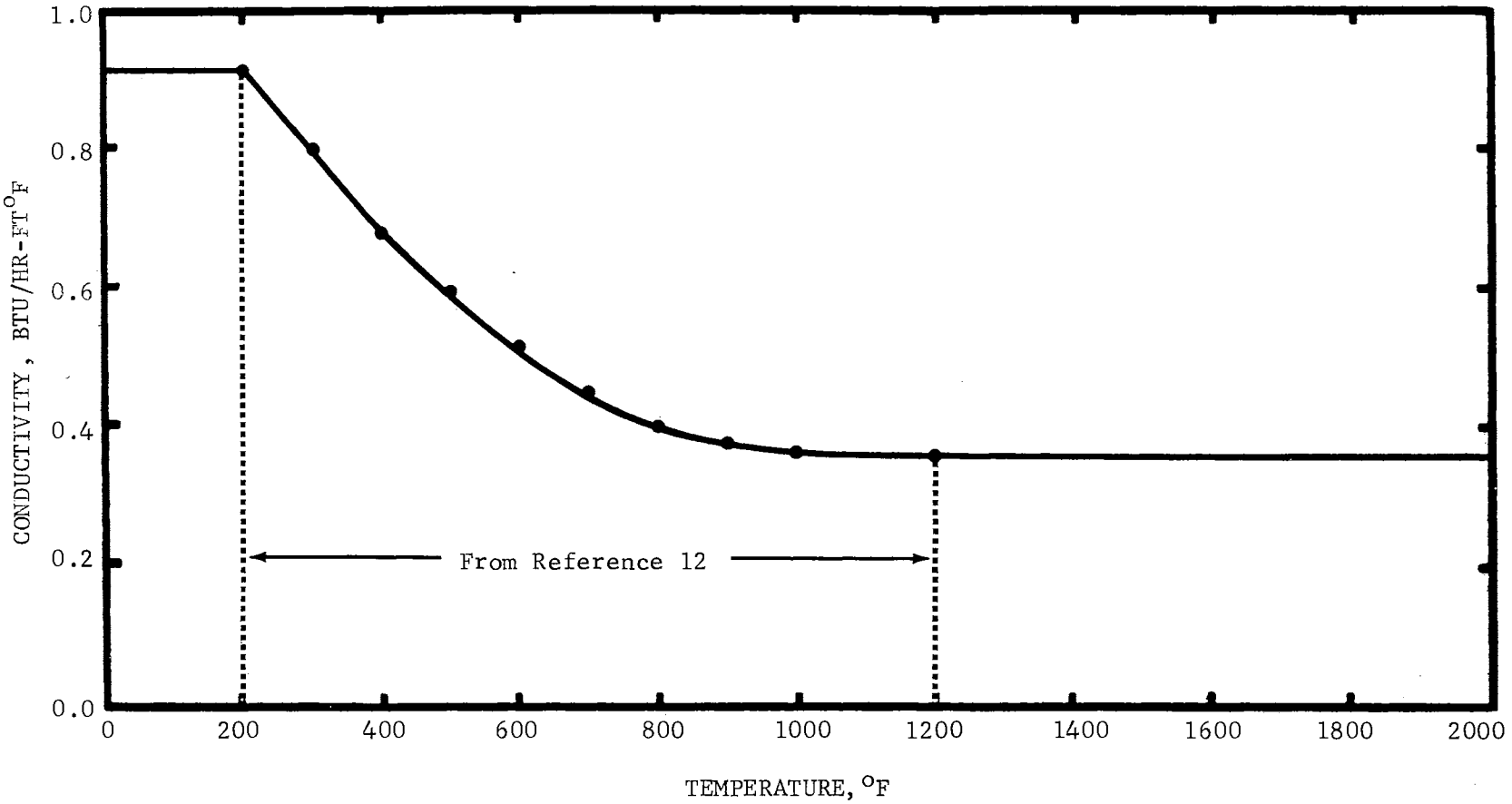


Figure 9. Thermal Conductivity Versus Temperature

conductivity was constant at 0.36 BTU/ft-hr^{o}F. The 0.93 BTU/ft-hr^{o}F lies in the range of measured shale values in the literature; however it is 0.18 BTU/ft-hr^{o}F higher than Somerton (13) recently measured.

Description of Process Variables

Temperature

As mentioned in the description of the problem, injection temperature is an important process variable that can be controlled. The temperature must be greater than 700^{o}F to achieve substantially complete distillation of the kerogen. It would seem to be desirable from the standpoint of heating rate and thus retorting rate that the source temperature be as high as possible. Also, for the same injected heat, injection rates can be reduced as injection temperatures are increased. Reduced injection rates improve the economics of the process. At temperatures of 1800 to 2000^{o}F silicates will begin to fuse. Therefore, temperatures in excess of these might glaze the injection bore hole and the entrance portion of the fracture and could possibly cause pressure and flow problems. Because of this, 2000^{o}F is considered to be the most likely temperature to be used.

Endothermic heat losses become increasingly large over 1000^{o}F. It is not intuitively evident whether the adverse effects of these heat losses could be overcome by either reduced injection rates or greater rates of heating possible with high injection temperatures. The differences in the rates of retorting rock versus time for all injection temperatures investigated when compared at equal heating rates are an important consideration in this study. Should these differences be insignificant, then the highest injection temperature would be preferred

because of reduced gas injection rates.

Injection Rate

A larger injection rate at the same injection temperature means an increase in heat injection per unit time. This serves to raise fracture temperatures and thus the rate of retorting rock will be increased. However, after hot gas has reached the producing well, a larger quantity of heat will be exhausted at the higher injection rate. Therefore, per cent heat utilization drops with increased injection rates after heat "breakthrough." This problem may be partially alleviated if provisions for utilization of exhaust heat have been made. The effect of injection rate on the process was investigated.

Well Spacing

The effect of spacing on retorting efficiency may not be pronounced; however, larger distances between injection and producing wells may improve utilization of heat. Spacing between lines of wells will be determined by the rate of heat penetration. The larger spacing that is possible without affecting retorting efficiency, the more economic the process will become because fewer wells would be required.

Time

The optimum operating life of a project would be a function of equipment life and investment return, as well as process parameters. When the rate of retorting drops below a certain value, oil production will not pay for operating expenses. This is obviously the termination point of the project. This time can be controlled by the rate

of heat injection, injection temperature and spacing. Results from this study should provide information to predict optimum project length.

Heat Soaking

In those cases where the producing well temperature becomes far in excess of the retorting temperature, continued operation would be impractical. This is so from a heat utilization standpoint as well as from the standpoint of cracking all liquid petroleum products into gas or, even worse, into carbon. It is anticipated that at this time the producing and injecting wells would be shut in and the shale would be allowed to "soak." When temperatures dropped sufficiently, injection would be resumed. The effect of this "soaking" on retorting efficiency and heat utilization was investigated.

Air-Oil Ratio

The air-oil ratio is defined here as the volume of injected air plus fuel required to produce one barrel of oil. From this ratio the compression cost to produce a barrel of oil can be calculated. This cost is the most important one when considering process economics. The amount of injected air required to produce a barrel of oil is dependent on all process variables; however, the injection temperature will probably be the one which influences it most. Two-thousand-degree gas contains approximately twice the heat of 1000°F gas. Therefore, if retorted shale volumes were the same for equivalent heat injection rates, then the total volume of air injection at 2000°F would only be one half of that at 1000°F. Thus producing air-oil ratios would be

only one half as much, also.

In this study a theoretical optimum air-oil ratio versus time was determined for all cases studied. Oil production was considered a direct function of the volume of oil shale retorted while air injection was constant for all runs.

CHAPTER IV

APPROACH TO NUMERICAL SOLUTIONS

Numerical Procedure

The alternating direction implicit procedure (ADIP) was used to solve the difference equations describing heat transfer in the oil shale. The literature indicated that for heat models the alternating direction implicit procedure (ADIP) possessed advantages of speed over the explicit and implicit techniques. Coats (34) showed the method for two-phase fluid problems to be some 60 per cent slower than alternating direction explicit procedure (ADEP) but it was more accurate. The ADIP was used here because the literature contained much more supporting evidence of its versatility to heat flow problems. For steady state conduction heat transfer problems the ADIP has been proven absolutely convergent for any time step. However, the method is not absolutely convergent for any time step for nonlinear problems or problems containing convective terms.

The ADIP for two dimensions involves the use of two sets of difference equations. One set has the second derivative $\frac{\partial^2 T}{\partial x^2}$ replaced by a second difference written in terms of unknown temperatures while $\frac{\partial^2 T}{\partial y^2}$ is substituted by a second difference written in terms of known temperatures of the previous calculation. The second set of difference equations has $\frac{\partial^2 T}{\partial y^2}$ described by unknown temperatures and $\frac{\partial^2 T}{\partial x^2}$

described by known temperatures. The first set of difference equations is said to be implicit in the x direction and explicit in the y direction. The second set is implicit in y and explicit in x. There are as many equations in each set as there are mesh points in the grid representing the problem. The method of solution involves starting with either type equation written for each point in the grid. This set of equations is then solved by matrix inversion. The matrix inversion technique used in this work is described in Appendix C. For the next time step or iteration (depending on the problem), equations written implicit in the opposite direction are solved by matrix inversion. This alternating procedure is then repeated over and over again throughout the duration of the problem.

Description of Numerical Models

The differential equations representing each model, derived in Appendix A, are represented in terms of difference equations in Appendix B. The difference equation analogous to the differential equation that describes convection heat transfer with heat loss into shale, as described for Models I and II, is derived. It uses a two-point backward difference term to represent change in fracture temperature and a three-point forward difference term to describe heat transfer into the shale. The solution is explicit as only one boundary condition is required.

Since ADIP was used, difference equations that describe heat flow in the shale are derived both implicitly and explicitly in each direction that is considered for each of the models. Central differences are used to represent the second order differential conduction terms.

Central differences are also used for the first order convection term in Model III. These equations are all rearranged into a similar form as follows:

$$A_i T_{i-1} + B_i T_i + C_i T_{i+1} = D_i$$

where the coefficients A_i, B_i, C_i and D_i are comprised of constants, functions of known temperatures, and known temperatures. The subscripts refer to the mesh point the equation represents.

Appendix B also includes a description of the variable spacing scheme used in addition to the derivations of difference equations for the five computer programs that were developed for this research work. The five computer programs were named THERMAB1, THERMAB2, THERMAB3, THERMAB4 and THERMAB5. THERMAB1 and THERMAB3 were developed to solve Model I. THERMAB2 was written to solve Model III. THERMAB4 was used for the solution of Model II and THERMAB5 was the program used to solve Model IV. Boundary conditions for each program are listed in Appendix B.

It is to be noted that the numerical derivations in Appendix B were derived using specific endothermic heat requirement and specific thermal capacity. Since only total heat requirement and content data were readily available, these data were used in all programs. To use it, all that was necessary was to replace the composite specific total heat requirement (identified as C in the appendix) by the total heat requirement divided by the temperature.

The curves describing thermal conductivity, total heat requirement and total heat content shown in Figures 8 and 9 were described by equations so that they could be used in these computer programs. The

thermal conductivity curve from 200 to 1200°F was represented by a fourth order polynomial obtained by a least squares fit. Above and below these temperatures conductivity values were constant. The curves for the total heat requirement and total heat content were each described by a series of linear equations.

Model I

Figure 10 shows the grid network representation of Models I and III. Grid points in the y direction are spaced closely near the fracture and farther apart away from the fracture. This permits fewer grid points to be used and still maintain uniform accuracy throughout the grid system. A method to determine the y grid spacing is given in Appendix D. The distance in the y direction is sufficient to represent an infinite boundary. The x spacing for these models is constant. The center line in the fracture coincides with the first row of mesh points.

A computer program, THERMAB1, was first written to numerically solve Model I assuming that thermal conductivity and capacity were constant and there was no endothermic heat loss. Difference equations for this program are the same as those used in THERMAB3 and THERMAB4, but with the above assumptions the resulting coefficients of the difference equations are simpler than those for THERMAB3 and THERMAB4. The difference coefficients for THERMAB1 are listed in Appendix B along with the appropriate boundary conditions for this model. A flow chart and program listing of THERMAB1 are shown in Appendix F.

The program for the general numerical solution of Model I is called THERMAB3. The detailed derivation of the numerical equations

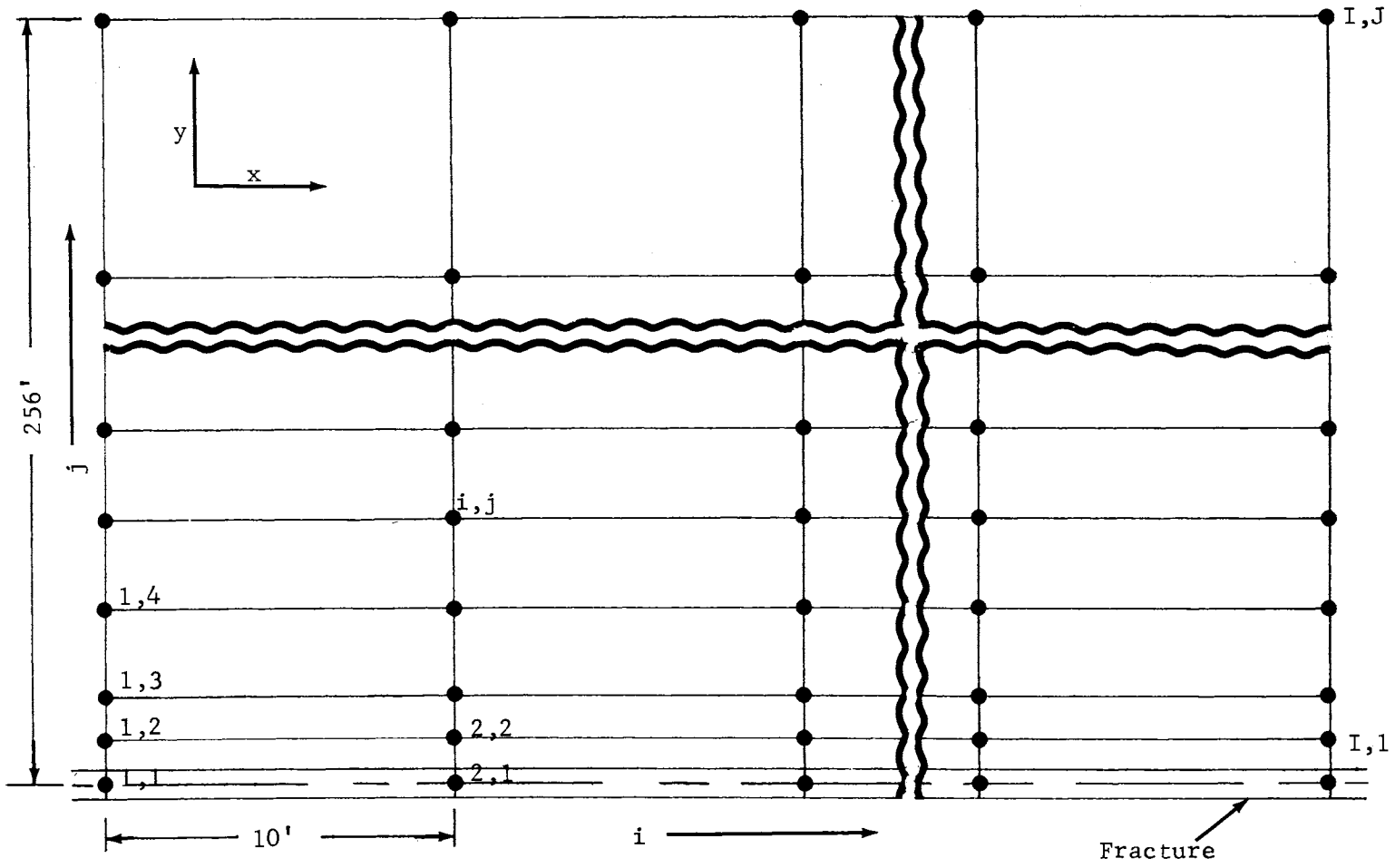


Figure 10. Arrangement of Grid Network for Models I and III

and resulting coefficients used in this program are presented in Appendix B. In these equations it is assumed there is directional variable thermal conductivity, variable thermal capacity and endothermic heat loss, all functions of temperature. The boundary conditions for Model I are also listed. A simplified flow sheet and program listing of THERMAB3 are given in Appendix F.

The basic method of solution of Model I with reference to the THERMAB3 flow chart is summarized as follows:

1. Read in all input data and initialize all parameters.
2. Compute values of velocity as functions of x distance.
3. Calculate the variable spacing values as described in Appendix B (CALCY subroutine).
4. Calculate initial values of thermal conductivity, total heat requirement, total heat content and store all in temporary arrays. The latter two sets of values as well as temperature are also stored in permanent arrays.
5. Solve the convection equation explicitly (CONV subroutine).
6. Calculate new values for thermal conductivity and total heat requirement in the j=1 row (COND5 and CAPAC1 subroutines).
7. Solve the conduction equation using ADIP first sweeping in the x direction (COND1 subroutine).
8. Calculate new values of thermal conductivity and total heat requirement (COND4 and CAPAC subroutines).
9. Check to see whether all newly calculated temperatures have changed from previously calculated temperatures by a prescribed tolerance.

10. If the changes for all temperatures are less than tolerance, store temperatures and latest values of total heat requirement in permanent arrays and go to next time step.
11. If tolerance is not satisfied for all temperatures, solve convection equation again (CONV subroutine).
12. Calculate new values for thermal conductivity and total heat requirement in the j=1 row (COND5 and CAPAC1).
13. Solve conduction equation sweeping in the y direction (COND2 subroutine).
14. Calculate new values of thermal conductivity and total heat requirement.
15. Check to see if new temperatures are within tolerance.
16. If so, repeat step 10.
17. If not, solve convection equation again, etc.

This is basically the procedure for THERMAB3 and THERMAB1; however, the thermal conductivity and total heat requirement steps are not included in THERMAB1. In short, the convection equation is solved along the j=1 row of the matrix. Using this as a boundary condition, the heat distribution in the shale is determined by first solving the conduction equation which is implicit in x and explicit in y. A check is made to see if temperatures have changed less than a prescribed tolerance. If they have, time is incremented; if not, the convection equation is resolved. Then the conduction equation written implicit in y and explicit in x is solved. A check is made to see if the new temperatures have changed less than the tolerance. If tolerance is satisfied, time is incremented; if not, the procedure is repeated

until the iteration procedure satisfies the tolerance. Convergence is obtained rapidly with this procedure.

The program has a built-in procedure to lengthen the time increment and to reduce the tolerance of iteration as time increases. Details of this are given later in this chapter and in Appendix D. The variable spacing sequence, convection routine, both conduction equation routines (COND1 and COND2), the thermal conductivity, and total heat requirement determination were written in subroutine form to facilitate programming.

A heat balance is included in this and all other programs used in this work. A double integration of the total heat requirement of the shale is made to obtain total heat utilized. The procedure used for integration is a two-dimensional Simpson rule with variable increments. The heat which flows in at the injection well and that which flows out of the producer is easily calculated since these end point temperatures in row j=1 are known at the end of each time step. The balance of heat in, with the heat out plus heat utilized, is a main controlling factor as to the validity and accuracy of the solution.

To obtain information needed to evaluate the assumption that no heat flows vertically, THERMAB3 is written so that temperatures and heat utilizations can be printed out every time step for prescribed positions along the fracture. These temperatures serve as input data for THERMAB5.

A contour subroutine is used to record the y distances from the fracture of several desired temperatures at all positions in the x direction. Linear interpolation is used between grid points.

If the fracture temperature at the producing well becomes a

specified amount greater than the retorting temperature (taken to be 900°F here), the program is written so that the use of the convection routine is stopped, this information is printed out, and the shale matrix begins to "soak." To do this, boundary conditions of the two conduction subroutines (COND1 and COND2) must be changed to be adiabatic on all sides. Instead of changing these boundary conditions, two new subroutines (COND6 and COND7) which are analogous to COND1 and COND2 except for the changed boundary conditions are used. When the temperature of the producing well falls below 800°F, the program is written to print this out and revert to the normal calculating routine.

As long as temperatures at a given matrix point are increasing, the total heat requirement values are used. They represent both heat content of the rock and endothermic heat requirements which are combined in the difference equations. During soaking if temperatures decrease only the heat content of the rock is available to be given up. Therefore a decision on which "heat capacity" value to use is made on the basis of whether the temperature of the point in question has been increasing or decreasing. A separate subroutine (CAPSK) to calculate rock heat content is used for the "soak" alternative.

Model II

Figure 11 represents the grid network of Model II. This model is bounded on two sides by fractures. The spacing of grid points in the y direction was determined in the same manner as it was for Model I, but the presence of the two fractures causes the y spacing to be greatest at the center of the model. The x spacing for this model is

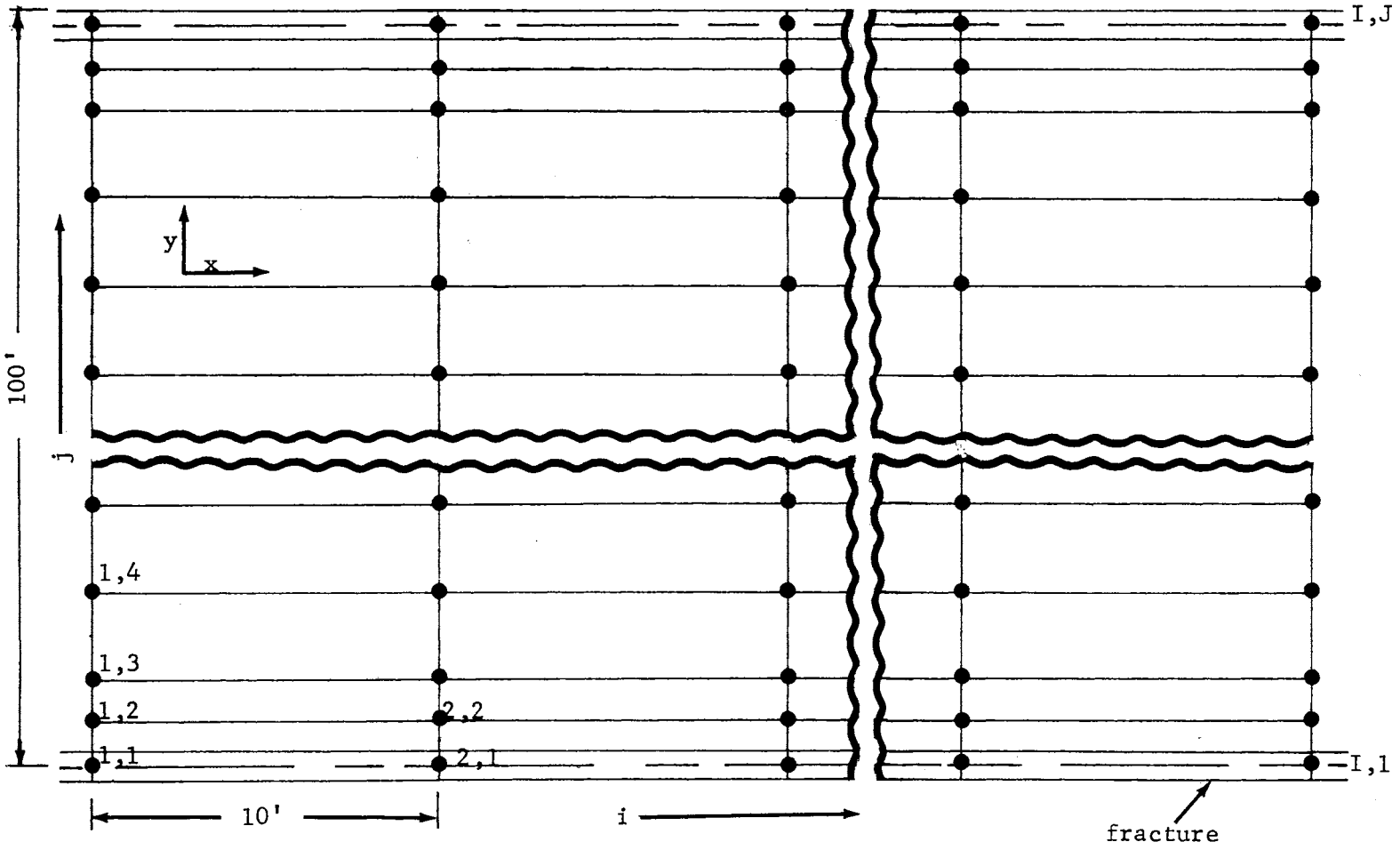


Figure 11. Arrangement of Grid Network for Model II

constant. The center lines of the two fractures coincides with the first and last row of mesh points. The distance between fractures is 100 feet and the distance from injection to producing well along a fracture is 500 feet.

The program for the numerical solution of Model II is called THERMAB4. The numerical equations for convection heat transfer are similar to the one for THERMAB3 with the appropriate point to point nomenclature changes made for the convection heat transfer equation along the j=J row. The numerical equations for conduction heat transfer in this program are the same as the ones for THERMAB3. The boundary conditions appropriate for THERMAB4 are listed in Appendix B. A flow chart and program listing are given in Appendix F.

Procedures in THERMAB4 are similar to THERMAB3. The basic difference is that in place of solving one convection equation, two convection equations are solved (CONV and CONV2 subroutines). Then new values of thermal conductivity and total heat requirement in both the j=1 and j=J rows are calculated (COND5, COND8, CAPAC1 and CAPAC2 subroutines). The procedures to lengthen the time increment and to reduce the tolerance of iteration are the same as those in THERMAB3. A heat balance, a contour subroutine and the "soak" feature are also included in this program.

Model III

Figure 10 as described for Model I also represents the grid network for Model III. THERMAB2 is the program for the numerical solution of this model. The difference equations both implicit in x and explicit in y, and implicit in y and explicit in x are derived in

Appendix B. The appropriate boundary conditions are also listed. A flow sheet and program listing are listed in Appendix F.

This model, used to evaluate the influence of convection heat transfer in the shale matrix, does not contain a separate convection equation whose solution is a boundary condition for the solution of the conduction equations. Instead, the convection term is combined with the conduction equations and the boundary conditions are all adiabatic, except at j=1, i=1, where a constant injection temperature is the boundary condition. Initially, convection heat transfer occurs only in the j=1 row (the fracture); therefore a switch (called HH in THERMAB2) to set the convective term equal to zero elsewhere is used. When the matrix temperature at any point exceeds 700°F, it is assumed limited permeability is developed and therefore the convective term is included by setting HH=1. The relative proportion of the convection present above the fracture must be set arbitrarily as it is a function of pressure which is not considered in this study.

Basically, this program is similar to THERMAB3. Although iteration is not necessary between a convection and conduction solution, it is necessary because of the nonlinear temperature-dependent coefficients. Subroutines for this program include COND1 and COND2 which are the basic routines for solving (by matrix inversion) the implicit in x and implicit in y forms of the difference equation that describes this model. Within each of these subroutines there is a built-in procedure which is written to determine the thickness of the retorted zone in the y direction. In other words, the procedure is designed to determine how many grid points are above 700°F. After this is determined, the convection heat transfer can be distributed as desired

throughout this zone. In the run made with this model it was assumed 90 per cent of the gas flow occured in the fracture and 10 per cent was distributed evenly along a vertical profile of the retorted zone.

Other subroutines used in this program are the same as those described for THERMAB3. The "soak" routine is not included.

Model IV

Figure 12 represents the grid network used to solve Model IV. In the y direction, distances between points are variable and are the same as the grid spacing for Model I. In the z direction, the first six grid points (representing 24.5 feet) are evenly spaced. From 24.5 feet to 281 feet the grid spacing is analogous to that in the y direction. The first grid points (m=1 through 6) at j=1 are held at the same temperature during a time step, but this temperature changes for each new time step. This temperature history is obtained from the resulting temperature history of a selected point along the fracture in Model I. For example, the temperature history of $T_{10,1}$ (x = 100 feet down the fracture) in Model I could serve as the source temperature for m=1 through 6, at j=1 for Model IV. All other boundary conditions for this model are the adiabatic condition. A flow chart and program listing are given in Appendix F.

The program for the numerical solution of Model IV is called THERMAB5. The numerical solution for this model consists of an iteration procedure between the two basic subroutines, COND1 and COND2. These subroutines represent the alternate conduction equations derived for the ADIP. Iteration is necessary, as before, because of the nonlinear temperature-dependent coefficients. The same auxiliary

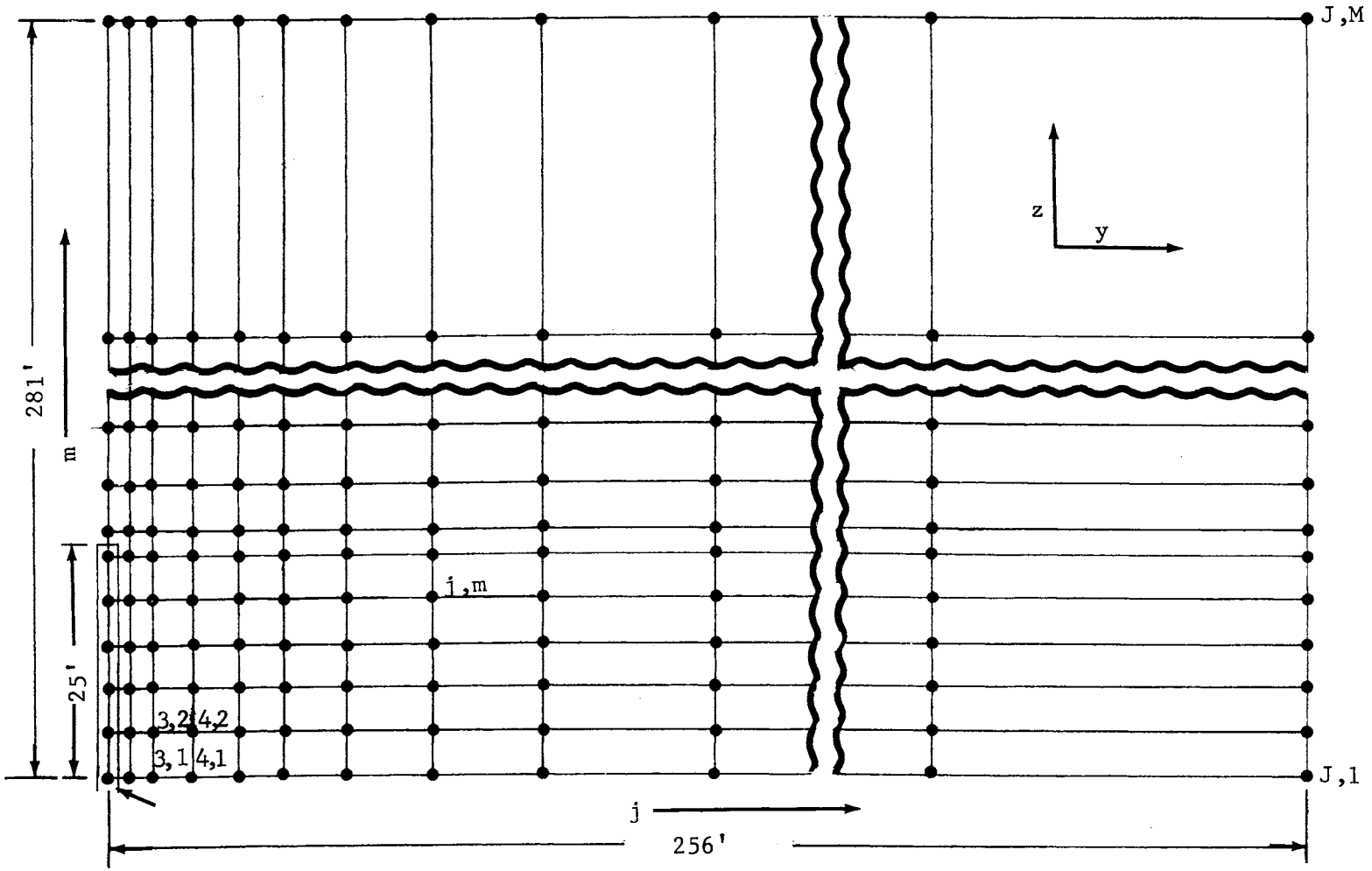


Figure 12. Arrangement of Grid Network for Model IV

subroutines, mentioned before, are utilized in this program. In addition, a subroutine, CALCX, is necessary for this program because of the variable spacing in the z direction. This program is written to calculate variable spacing values needed for the difference terms written for the z direction.

Determination of Iteration Parameters

Grid Spacing

The first three grid points in the y direction for all models were chosen one foot apart; the third through sixth point were two feet apart. Beyond this, grid spacings in the y direction were determined basically by the technique developed in Appendix D. This spacing technique allows the use of a minimum number of grid points, while requiring the temperature difference between any two adjacent points be less than a predescribed $(\Delta T)_{max}$.

Grid spacing in the x direction was constant. Using THERMAB3, it was determined that a Δx spacing of 10 feet was most satisfactory under the circumstances involved. A heat balance discrepancy of approximately 18 per cent after 30,000 hours of calculated history was obtained using a Δx of 20 feet. This compared to a heat balance discrepancy of 6 per cent using a Δx of 10 feet, other parameters being equal. Delta x spacings of less than 10 feet were impractical because of the number of grid points involved for the size of system being studied.

Time Incrementation

The size of the time increments used in the initial work with THERMAB1, which included constant coefficients, had an insignificant effect on the accuracy and no effect on numerical stability. This was at least true with Δt's of 1 to 100 hours. After 5000 hours of computed history the error in the heat balance was 2.5 per cent and after 10,000 hours, it was 1.0 per cent using a constant Δt of 50 hours. Systematically increasing Δt values from 1 to 50 hours resulted in a heat balance error of only 0.6 per cent after 5000 hours. It was possible to start calculations with Δt = 100 hours with no apparent instability.

With the use of the temperature-dependent coefficients of thermal conductivity, thermal capacity and endothermic heat loss, the time increment size and time of utilization were very critical. The development of a time increment sequence that was suitable from stability and speed of solution standpoints was developed to some extent from theoretical considerations, but largely by trial and error. Initially using THERMAB3, it was discovered that the numerical procedure became unstable after only 40 time steps (200 hours of computed history) when using a Δt of 5 hours and a Δx of 10 feet. Because of this, the technique described in Appendix D was used to increase Δt systematically. Using this approach after five initial Δt's of one hour, it was possible to increase the size of Δt to 38 hours before instability developed. Additional work showed, however, that this procedure had to be discontinued after a Δt of 15 hours for absolute stability. A Δt of 15 hours was used up to a cumulative time of 1000 hours of computed history; it was then increased to 30 hours and held constant up to 3000 hours. At 5000 hours' history, Δt was increased

to 50 hours and at 10,000 hours of history it was increased to 100 hours. This sequence was developed by trial and error using a Δt_0 of 2000°F, a Δx of 10 feet, an α of 10 BTU/hr-ft°F and a tolerance of iteration of 0.05. Actually with this tolerance it was possible to go to a Δt of 100 hours earlier but the tolerance sequence described below required that Δt not be incremented to 100 until 10,000 hours of history. The size of the time step was less critical when a Δx spacing of 20 feet was used. Computations were stable with a Δt of 50 hours as early as 1000 hours of history, with a Δt of 100 hours as early as 3000 hours and with a Δt of 200 hours as early as 10,000 hours of history. Also, the systematic incrementing procedure could be used up to 38 hours and remain there without stability trouble. But, because of heat balance error, the smaller Δx spacing of 10 feet was adopted.

Tolerance Incrementation

As temperatures increased, a tolerance of 0.05 represented a sizable temperature spread and therefore for accuracy it had to be decreased. By trial and error with Δt values of 15, 30, 50 and 100 at the times stated above, it was determined that the tolerance could be reduced to 0.04 at 3000 hours, to 0.03 at 4000 hours, to 0.02 at 5000 hours and to 0.01 at 20,000 hours. The time increment and tolerance sequences used for these programs are illustrated in Figure 13. With small time steps, a smaller tolerance could have been used earlier; however, due to the computer time involved with these runs, it was necessary to sacrifice some accuracy for speed.

A change in injection temperature and injection rate from the

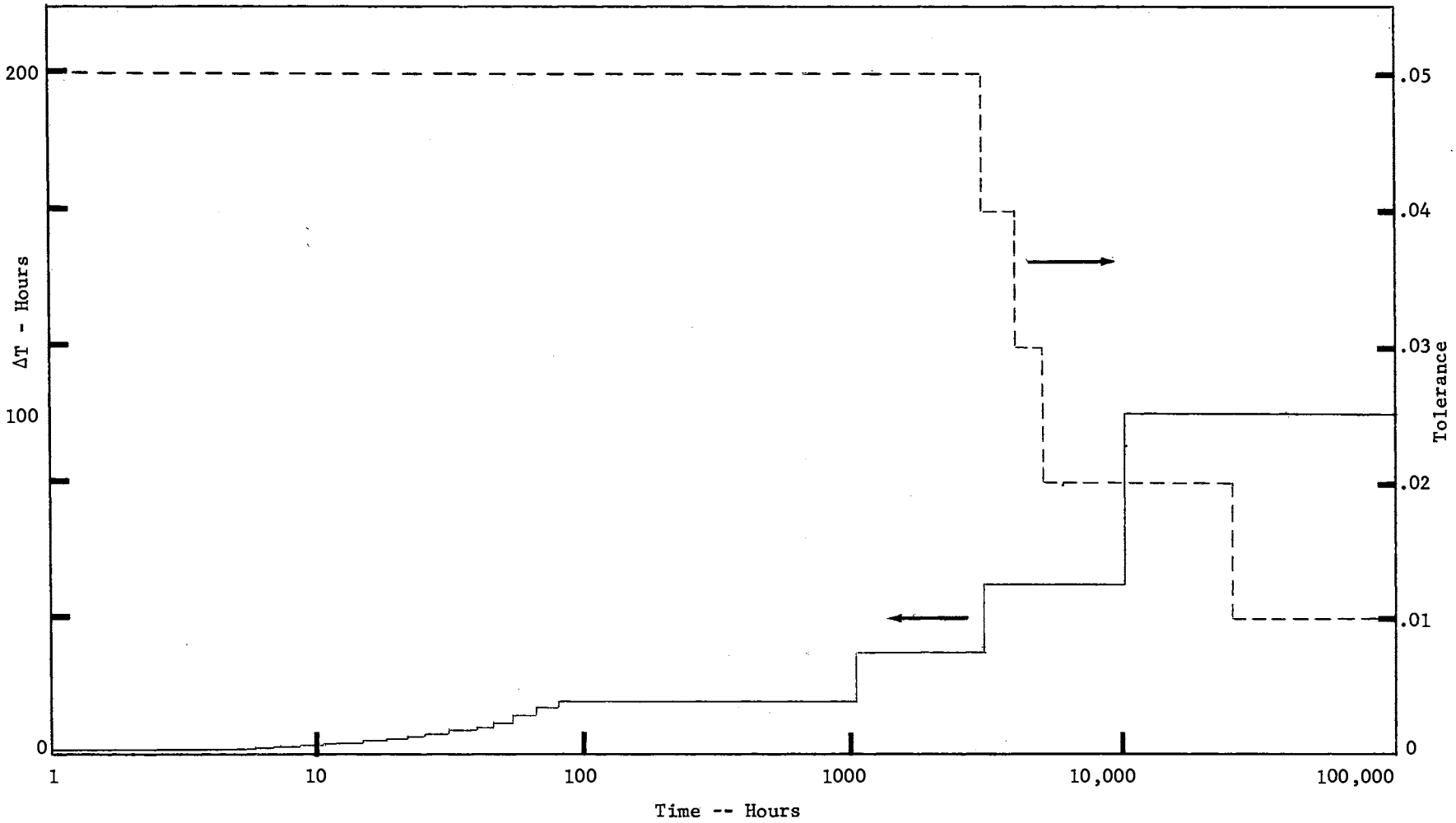


Figure 13. Time Increment and Tolerance Sequence

T_0 = 2000°F and α = 10 BTU/hr-ft°F, for which these Δt and tolerance sequences were developed, can cause instability when Δt is large and tolerance small. To avoid this, all programs were written as follows:

1. At 20,000 hours, tolerance was set at 0.01.
2. If instability resulted, tolerance was reset at 0.02, etc.
3. At 25,000 hours, a tolerance of 0.01 was tried again, etc.

In all cases a tolerance of 0.01 was used after 25,000 hours. Instability was determined by installing a counter on the number of iterations per time step. After ten iterations the programs were set to alter the tolerance as described. After twenty iterations in the same time step at any stage throughout a run, all programs were written to stop calculations. This safeguard prevented excessive computer usage.

Solution Validity

Check of Lauwerier Solution

An analytical solution of a simplified version of the heat transfer problem solved here served as a check as to the validity of the numerical procedures used in this work. Lauwerier (23) solved a one-dimensional convection heat transfer problem which loses heat by conduction perpendicular to the direction of flow. Conduction heat flow in the bounding media was one dimensional perpendicular to the direction of fluid flow. It was assumed that flow path was infinitely long and coefficients were constant. The THERMAB1 solution (constant coefficients) should be very close to the Lauwerier solution up until the time of heat breakthrough at the producing well since the heat

conductance in the x direction has a relatively small effect compared to heat conductance in the y direction.

Figure 14 is a comparison of the fracture temperature determined with the Lauwerier solution to that determined with THERMAB1 after 5000 hours of history. Little difference can be noted between the solutions. Figure 15 is a comparison of the position of the 600°F isotherms after 5000 hours as determined by the two methods. Again little difference is noted. This comparison is the more significant because the position of the 600°F isotherm represents the degree of oil shale retorting as determined both numerically and analytically. The 600°F isotherm was considered because the ambient temperature was set at zero for these solutions but was considered to be 100°F in the physical problem.

Solutions Using Limiting Coefficients

Since nonlinear temperature-dependent coefficients were involved in this research problem, the above comparison does not represent complete verification of the numerical procedures. Another test that was used to support the validity of these numerical procedures (THERMAB3 and THERMAB4) was to calculate constant coefficient solutions using values of the coefficients determined at the minimum temperature and maximum temperature encountered in this study. If these solutions encompass a solution with variable coefficients then the results were qualitatively in the proper direction. Two tests of 5426 hours each were run. The first one used a thermal conductivity of 0.93 BTU/ft-hr°F and a thermal capacity of 31.5 BTU/ft^3-°F. The second one used a thermal conductivity of 0.36 BTU/ft-hr°F and a

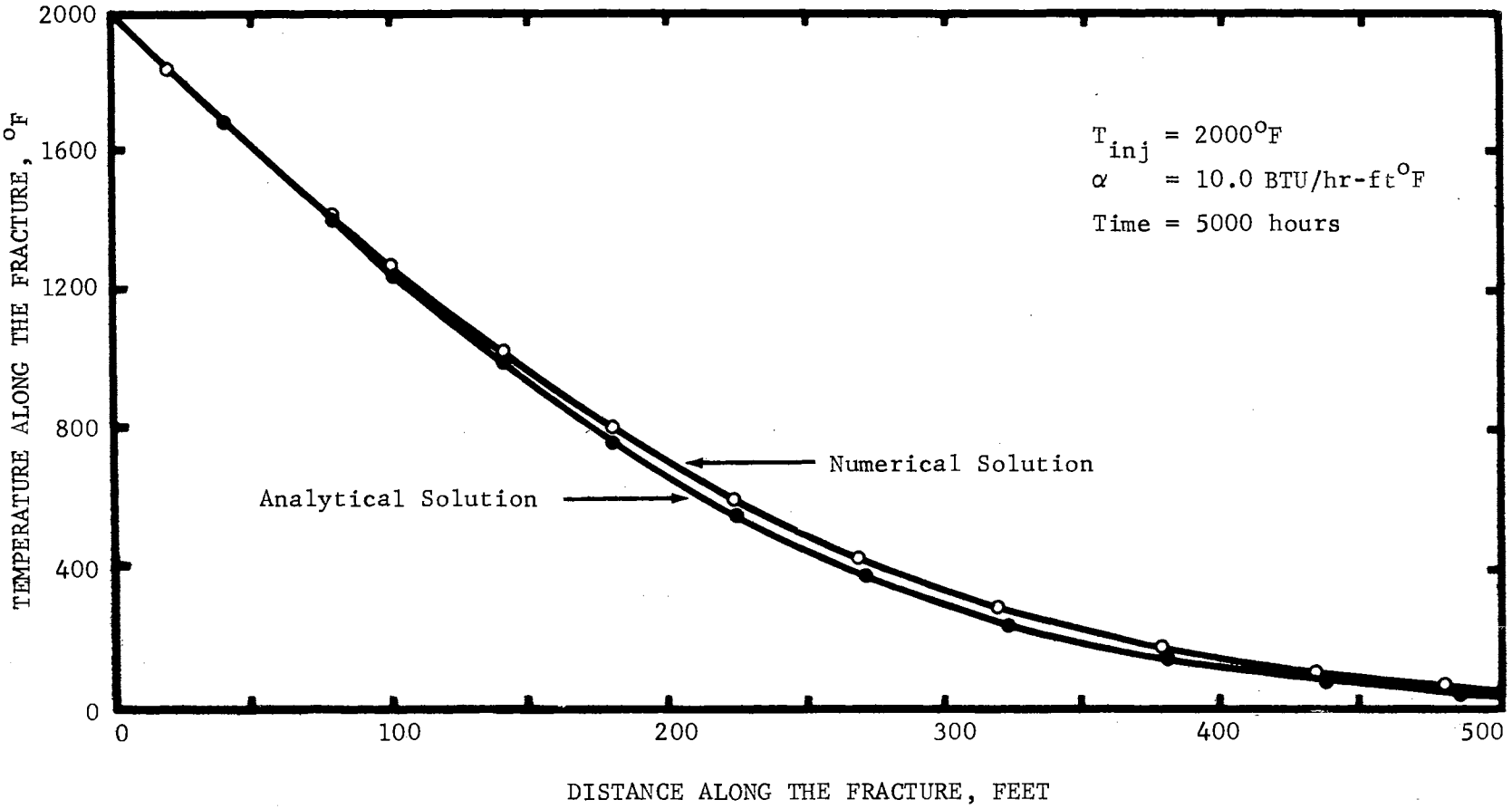


Figure 14. Comparison of Numerical and Analytical Solutions -- Fracture Temperatures.

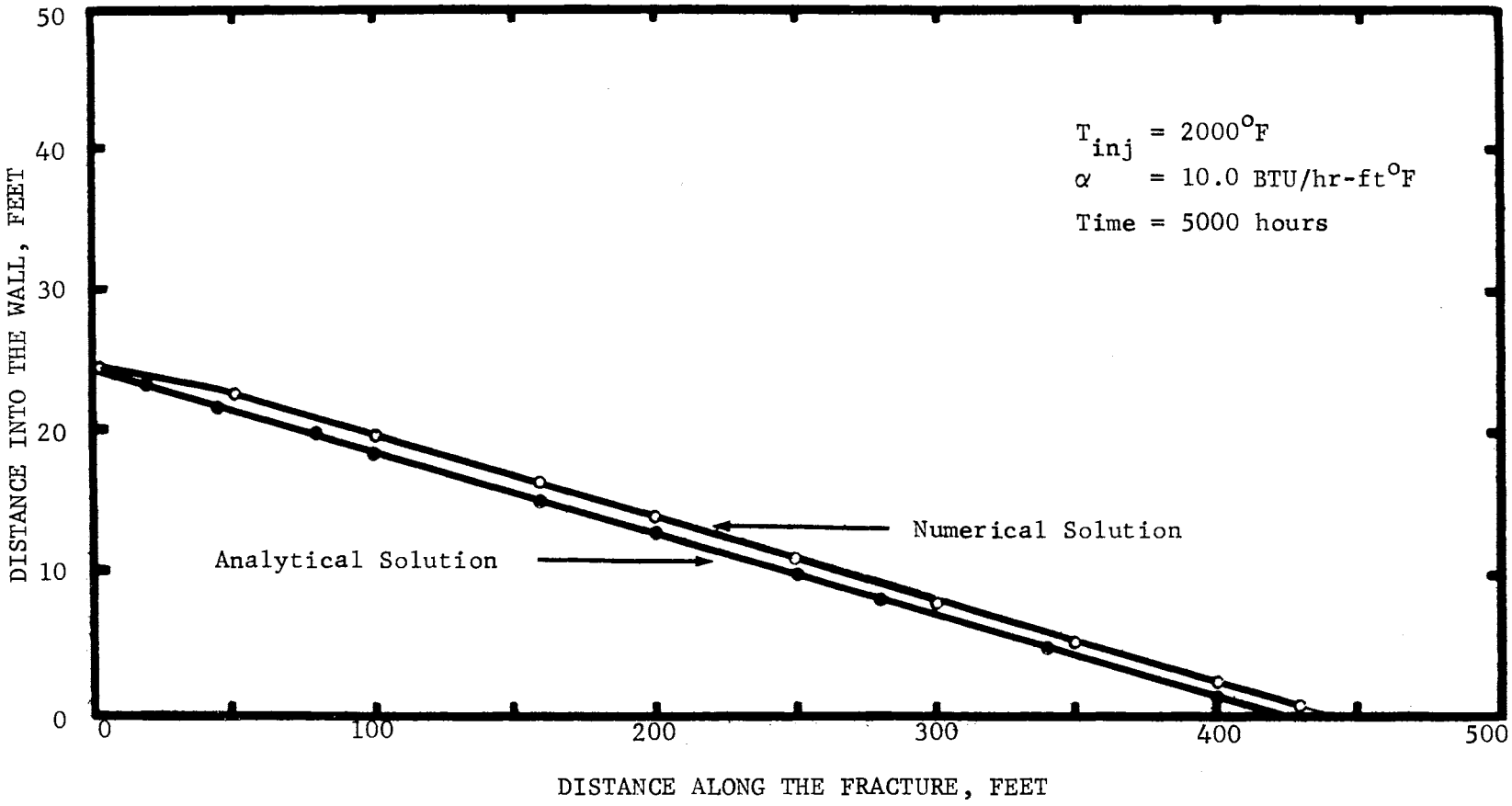


Figure 15. Comparison of Numerical and Analytical Solutions -- Positions of 600°F Isotherms

thermal capacity of 54.0 BTU/ft^3-^{o}F. These cases represent values of these two coefficients at 100 and 1600^{o}F, respectively. Figure 16 shows results of these two tests and also the result of a test using variable coefficients. Since the results from the latter test fall between the results from the two bounding tests, it was concluded the numerical procedures gave results that are qualitatively reasonable.

Heat Balance

As mentioned, heat balances using THERMAB1 (constant coefficients) were excellent very early into a run. By 2500 hours the heat balance error was less than 1 per cent. Errors in the heat balance for the variable coefficient solution were somewhat larger, especially at early times. One of the main reasons for this was because the iterative tolerance was started at 0.05 compared to 0.01 for the constant coefficient solutions.

The fractional error in the heat balance during any one time step monotonically decreased throughout the life of a run. This error was denoted in the program listings of THERMAB3 and THERMAB4 as ER2 (see Appendix E) and had values as high as 0.20 during the first few hundred hours of a run but decreased to values of 0.02 to 0.13, depending on the parameters of the problem. Cumulative heat balance discrepancies for the various runs ranged from 0.02 to 0.14 after 80,000 hours. In all cases discrepancies were positive, i.e., stored heat plus produced heat was greater than injected heat.

Effect of Assumptions

Convection Difference Equation

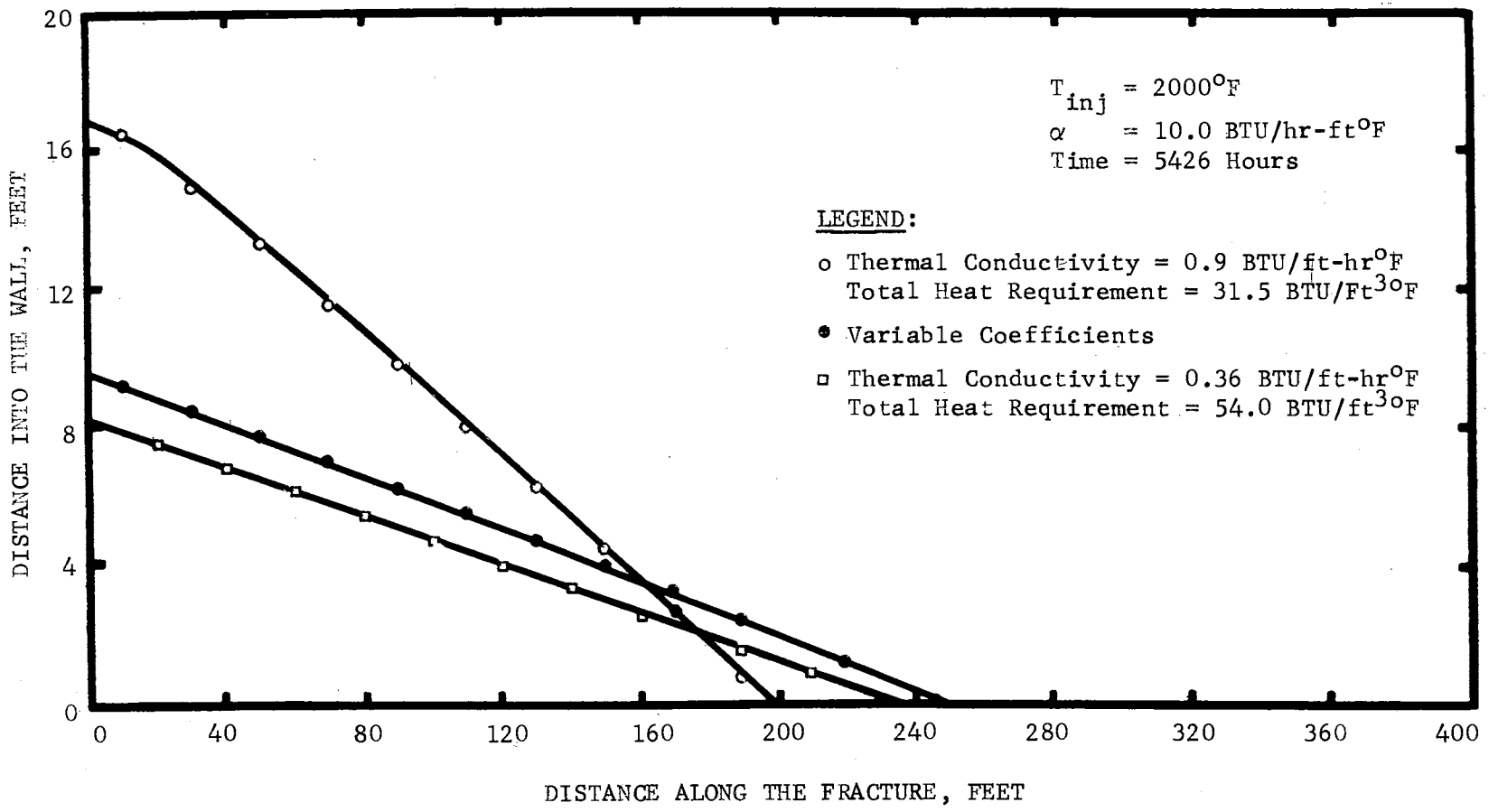


Figure 16. Comparison of 700°F Isotherm Positions using Constant and Variable Values of Nonlinear Coefficients

Comparison runs were made using both two-point and three-point backward difference representations to approximate the differential term describing convection heat transfer. The three-point approximation would be expected to give greater accuracy where temperature gradients were not linear. Results from these comparison runs showed that after a few hundred hours there was no perceptible difference in accuracy and therefore the two-point difference representation was used.

Velocity

Referring to Equations (B-13), (B-14), (B-15) and (B-16), it can be seen that there are two terms containing v, velocity, in their denominators. Assuming a fracture width of one-eighth inch, it was calculated that fracture entrance velocity would be approximately 600 ft/hr at an injection pressure of 3000 psi and a heat flow rate (α) of 10 BTU/hr-ft^{o}F. The velocity at the fracture outflow end would be on the order of 20,000 ft/hr if outflow pressures were 100 psi. Therefore, the value of v would be large any place within the fracture. The terms containing v are inversely proportional to v and thus they would be extremely small and probably insignificant.

The actual values of v used in these numerical solutions were based on the assumption that pressure drop through the fracture would be linear and the injected gas would behave as a perfect gas. Therefore the velocity was an inverse function of the pressure and range from 600 and 20,000 ft/hr for an alpha of 10 BTU/hr-ft^{o}F. An exponential equation with the x distance as the variable was used to

to calculate velocity at any x position. The equation was

$v = 600 + 0.8\ x + e^{.0182x}$.

A test was made to determine whether changes in the values of these velocities would affect the temperature solution. One run was made using velocities determined by doubling the values of all constants in the above equation. Fracture temperatures between the two runs varied less than two degrees at any point after 2500 hours. This showed that the effect of velocity on temperature distribution was insignificant providing the velocities were large. Because of this, the method of determining these velocities had little effect on the solution.

Infinite Heat Transfer Coefficient

In these numerical solutions it was assumed that the temperature of the shale at the fracture surface was the same as the gas temperature. This is the same as saying the heat transfer coefficient between gas and shale is infinite. Thomas (21) showed that for heat transfer coefficients of 5 to ∞ BTU/ft^2-hr$^\circ$F, the heat penetration rates into a substance of low thermal conductivity were essentially the same. He also showed for coefficients as low as 0.05 BTU/ft^2-hr$^\circ$F the average heat penetration was reduced only about 10 per cent. Eckert (36) indicates that heat transfer coefficients of air range from 2 to 50 BTU/ft^2-hr$^\circ$F over various flat surfaces. Assuming that the heat transfer coefficient of a fracture surface in oil shale fell into this category, we would expect that an infinite heat transfer coefficient assumption for this work would be valid. A check of this assumption was made by making a short run with the quantity K/L between the j=1

and j=2 rows set equal to a boundary layer heat transfer coefficient of 5 BTU/ft^2-hr$^\circ$F. The j=2 row was then considered to be the shale surface with the j=1 row being the boundary layer surface.

Figure 17 shows a comparison of the results of this run with a run assuming an infinite heat transfer coefficient. As can be seen the positions of the 700°F isotherms for the two cases are very close, thus supporting the infinite heat transfer assumption. Longer runs would show even a less percentage difference in the positions of the isotherms.

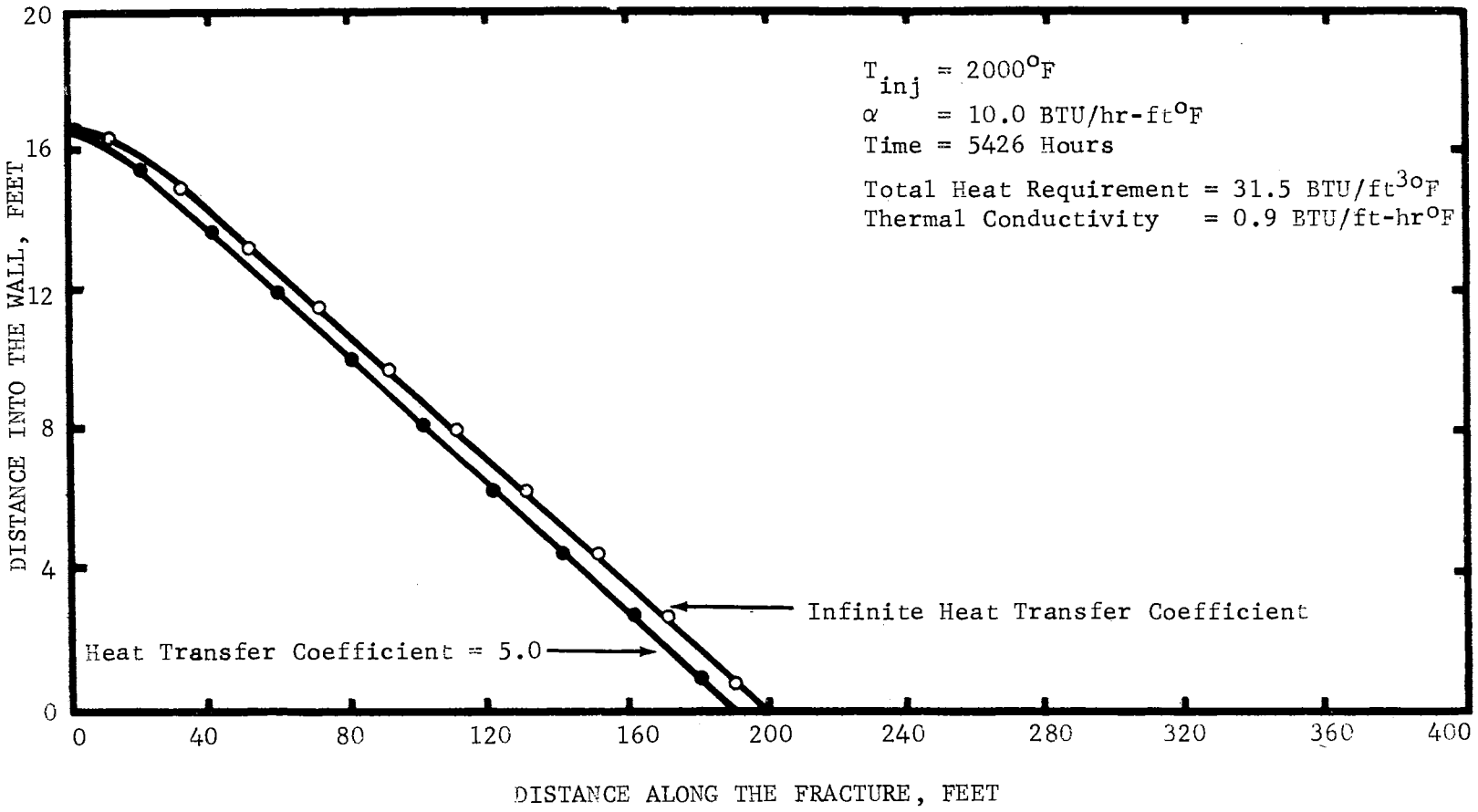


Figure 17. Effect of Infinite Heat Transfer Coefficient Assumption

CHAPTER V

RESULTS AND DISCUSSION

A total of 18 runs were made for this study. Table I summarizes these runs along with their pertinent parameters. All runs were conducted for 86,000 to 90,000 hours (approximately 10 years) real time, excepting for Run 11, which was 110,000 hours. Nine runs were made for Model I (THERMAB3), three runs for Model II (THERMAB4), two runs for Model III (THERMAB2) and four runs for Model IV (THERMAB5). Example output sheets for THERMAB3 and THERMAB4 are shown in Appendix E.

Computer time for these runs varied from 35 minutes to three hours in length. Model IV was solved in 35 minutes using THERMAB5. Conventional 500-foot runs using THERMAB3 required about 1 hour 20 minutes of computer time, Model III required 1 hour 30 minutes to solve using THERMAB2 and Model II required 1 hour 40 minutes with THERMAB4. Run 9, 1000-foot system, required approximately three hours to solve using THERMAB3. This run used 2121 grid points.

Results from Model I and Model II

General Discussion

Results from the nine runs made for Model I are presented in Figures 18 through 35, and results from the three runs made for Model II are presented in Figures 36 through 41. Two figures are presented for each run. The first is a plot of the position of the 700°F

TABLE I

SUMMARY OF COMPUTER RUNS

Run Number	Model	Fracture Length, ft	Injection Temperature, °F	Injection Rate α^*	Btu/hr-ft	SCF/hr-ft	MMSCF/D - Well
1	I	500	1000	5.00	5,000	278	1.33
2	I	500	1000	10.00	10,000	556	2.67
3	I	500	1500	6.67	10,000	370	1.78
4	I	500	1500	13.33	20,000	740	3.55
5	I	500	2000	2.50	5,000	139	0.67
6	I	500	2000	5.00	10,000	278	1.33
7	I	500	2000	10.00	20,000	556	2.67
8	I	500	3000	3.33	10,000	156	0.75
9	I	1000	2000	10.00	20,000	556	2.67
10	II	500	1500	6.67	10,000	370	1.78
11	II	500	2000	5.00	10,000	278	1.33
12	II	500	2000	10.00	20,000	556	2.67
13**	III	500	2000	2.50	5,000	139	0.67
14	III	500	2000	2.50	5,000	139	0.67
15	IV	Cross-section Profile at 0 feet along fracture of Run 6.					
16	IV	Cross-section Profile at 100 feet along fracture of Run 6.					
17	IV	Cross-section Profile at 200 feet along fracture of Run 6.					
18	IV	Cross-section Profile at 400 feet along fracture of Run 6.					

*BTU/hr-ft°F

**100 per cent of gas flow in fracture.

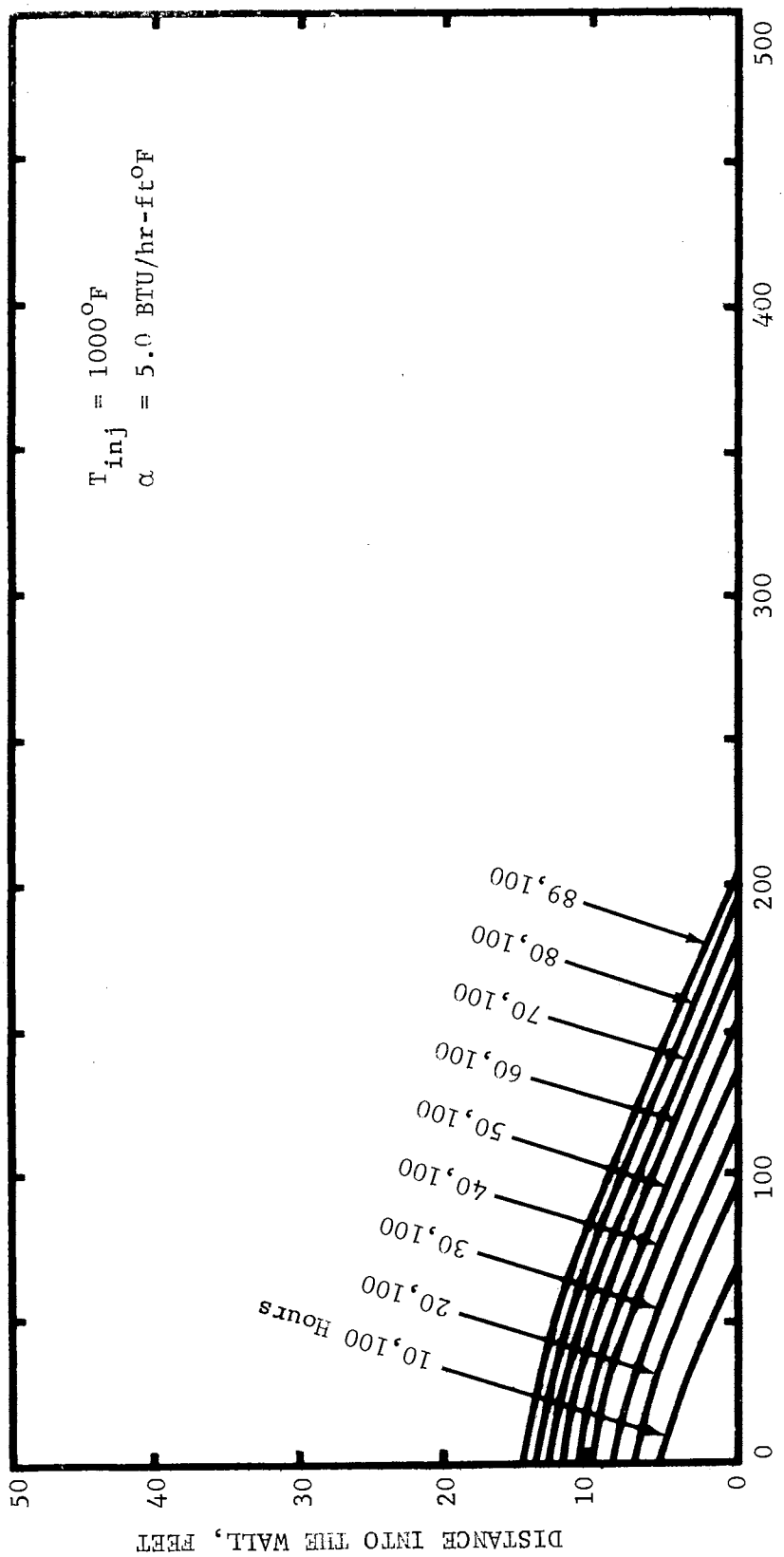


Figure 18. Position of 700°F Isotherm Versus Time -- Run 1

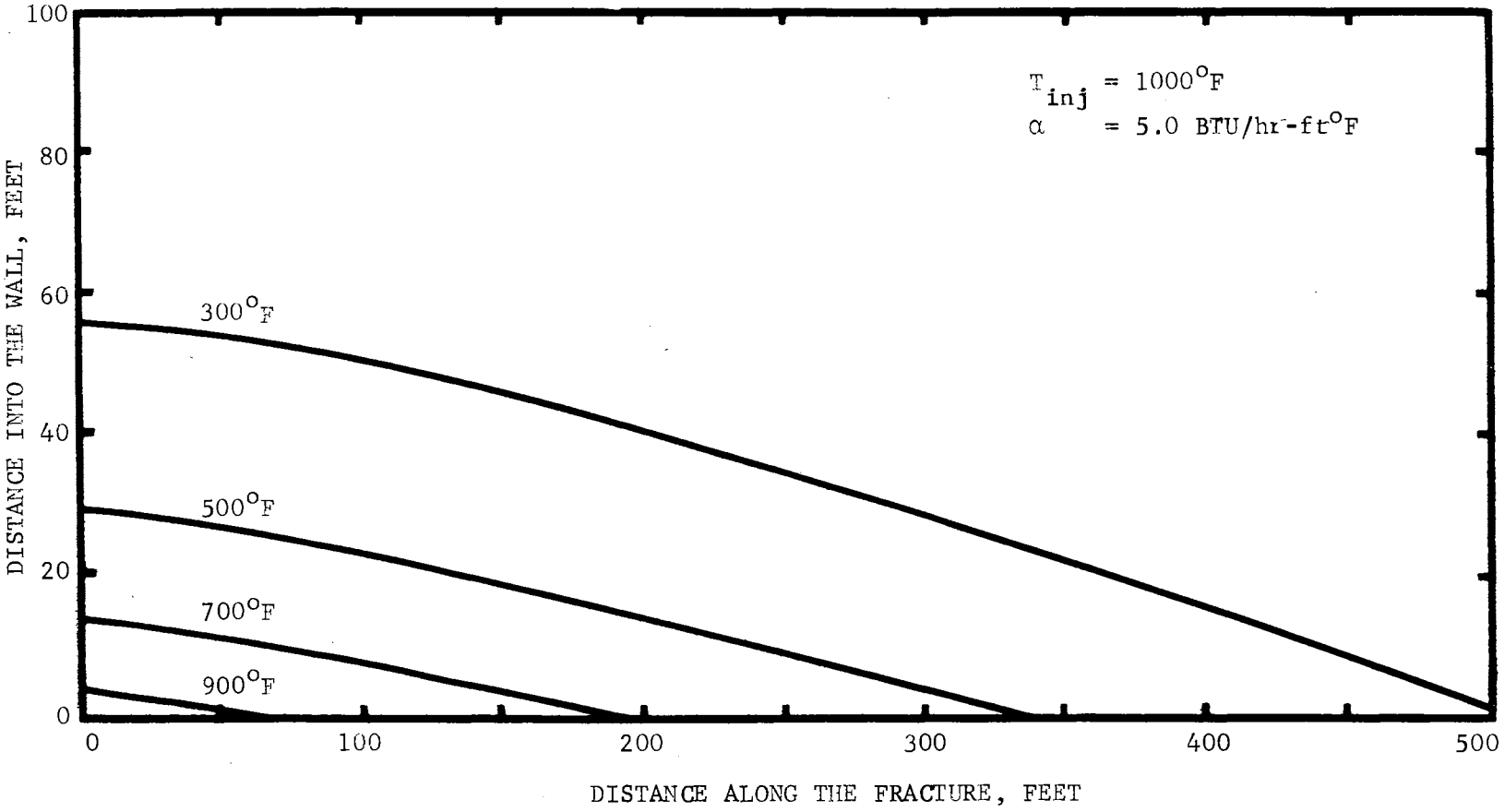


Figure 19. Temperature Distribution After 86,100 Hours -- Run 1

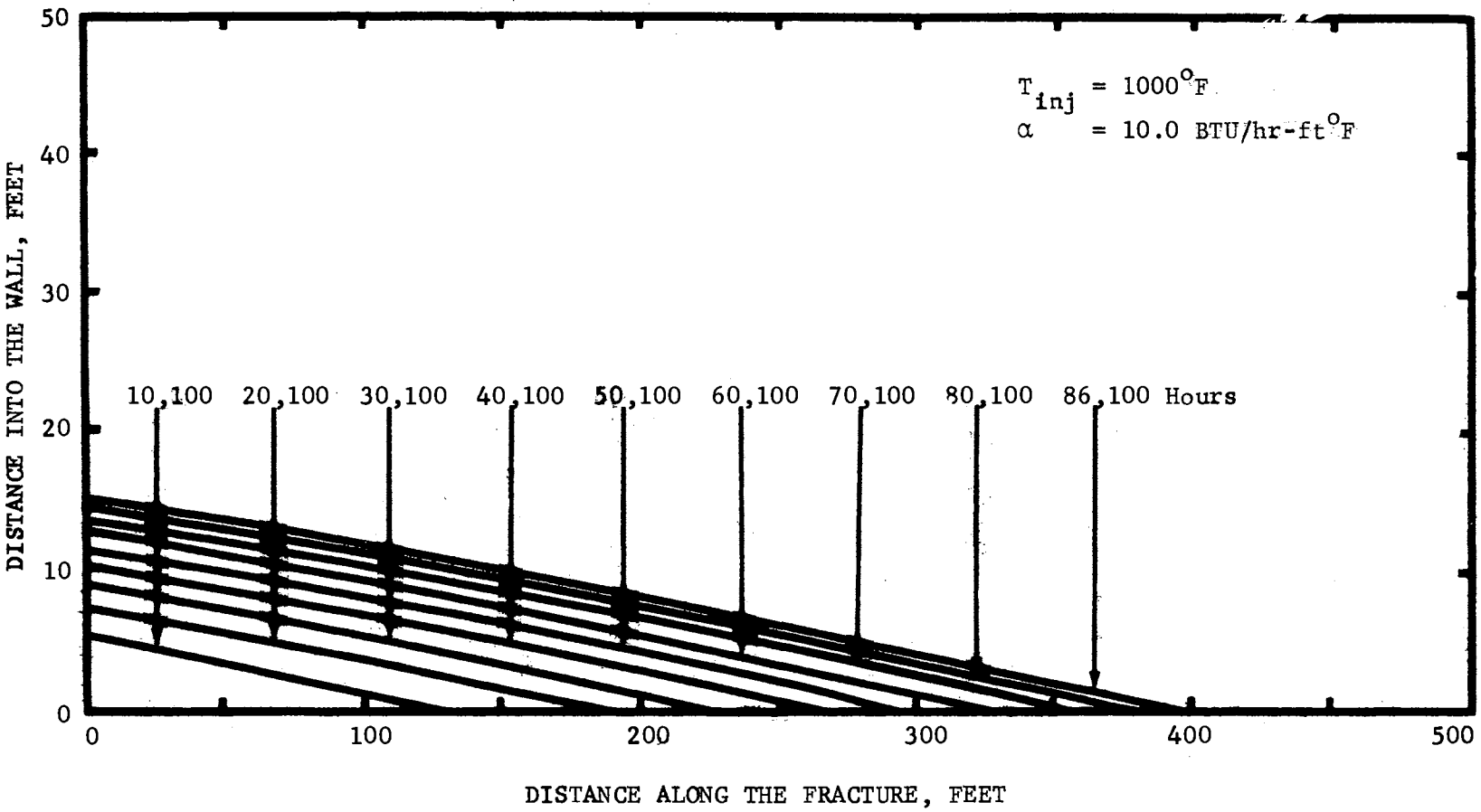


Figure 20. Position of 700°F Isotherm Versus Time -- Run 2

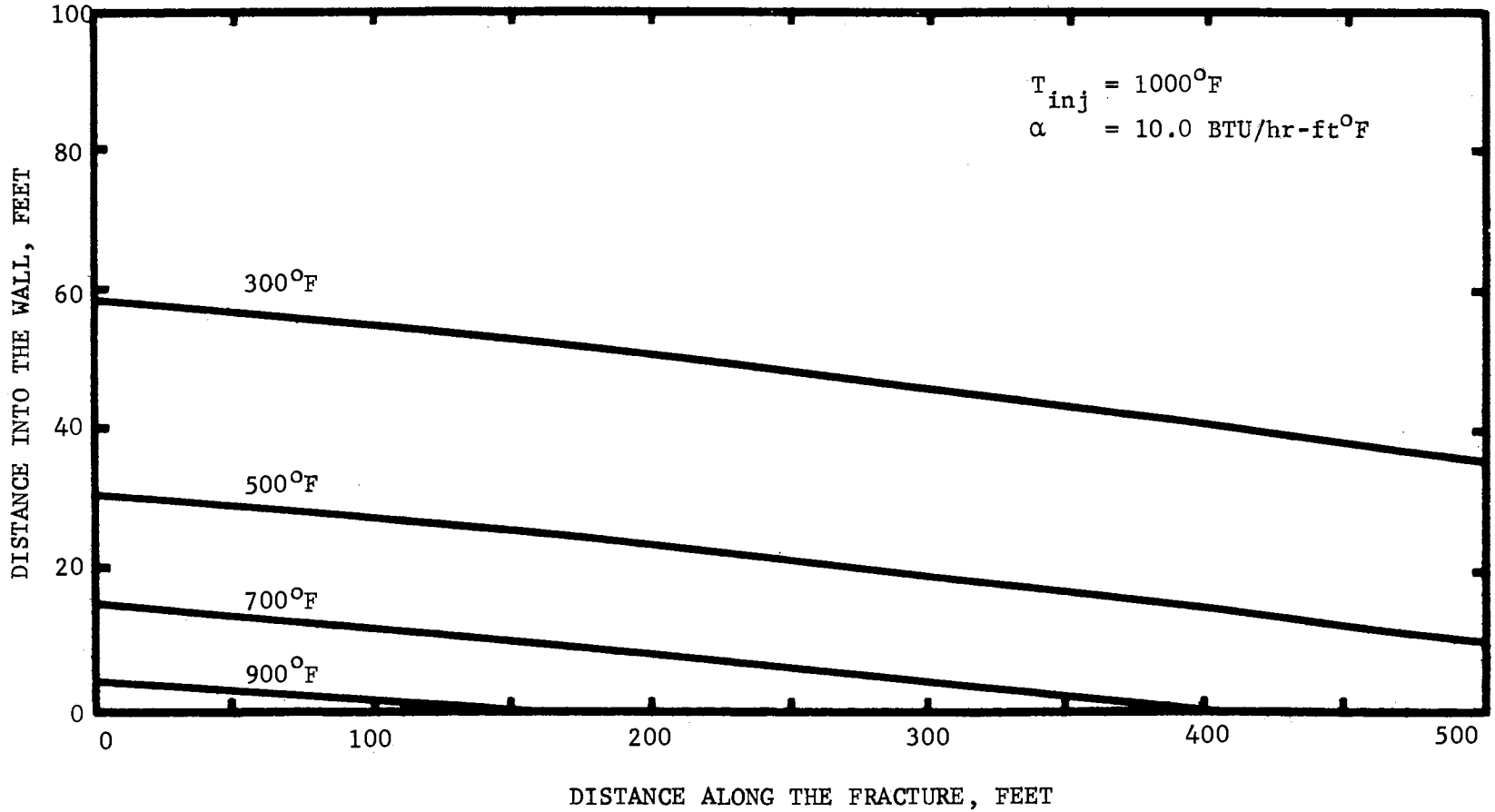


Figure 21. Temperature Distribution After 86,100 Hours -- Run 2

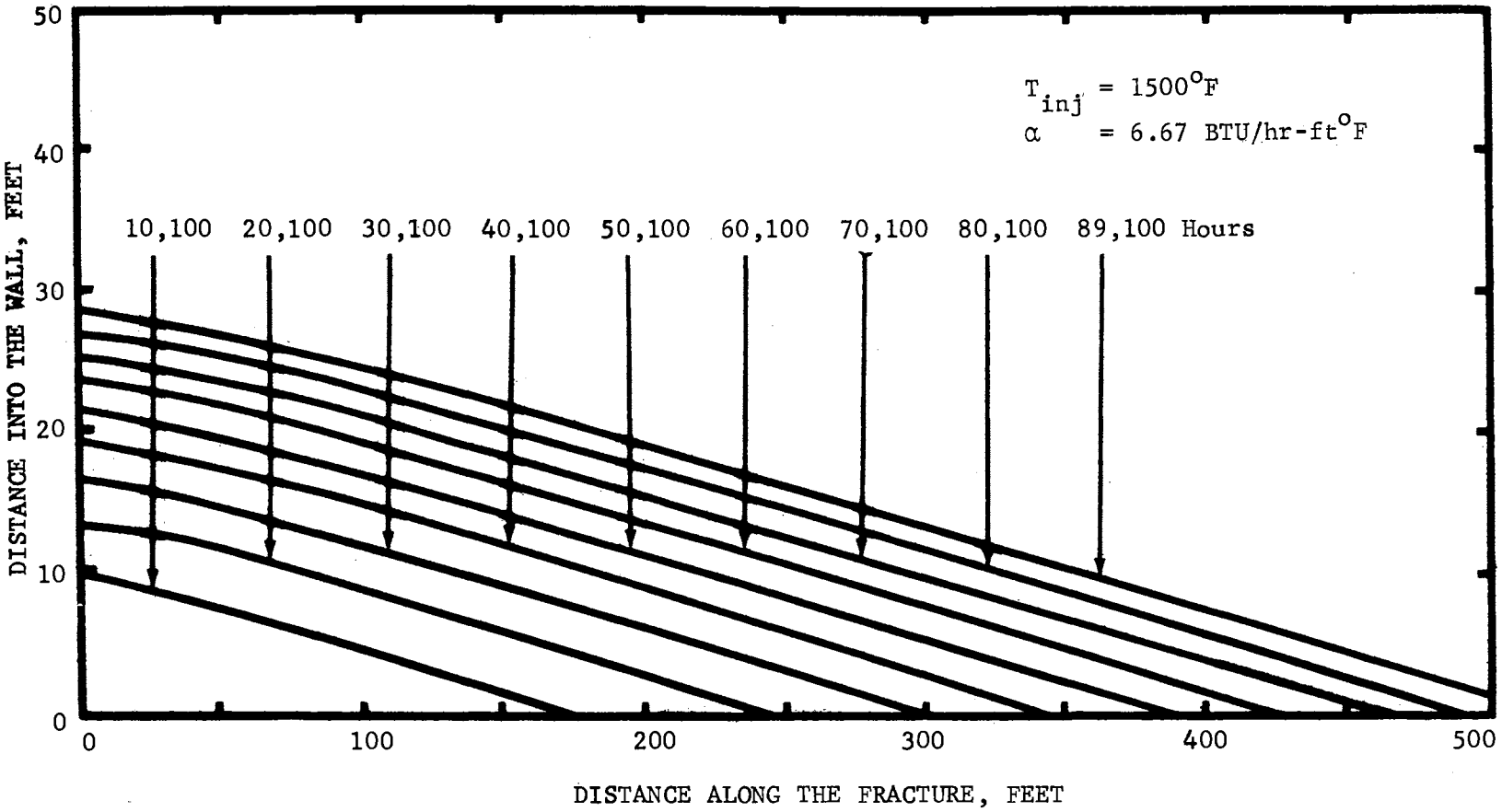


Figure 22. Position of 700°F Isotherm Versus Time -- Run 3

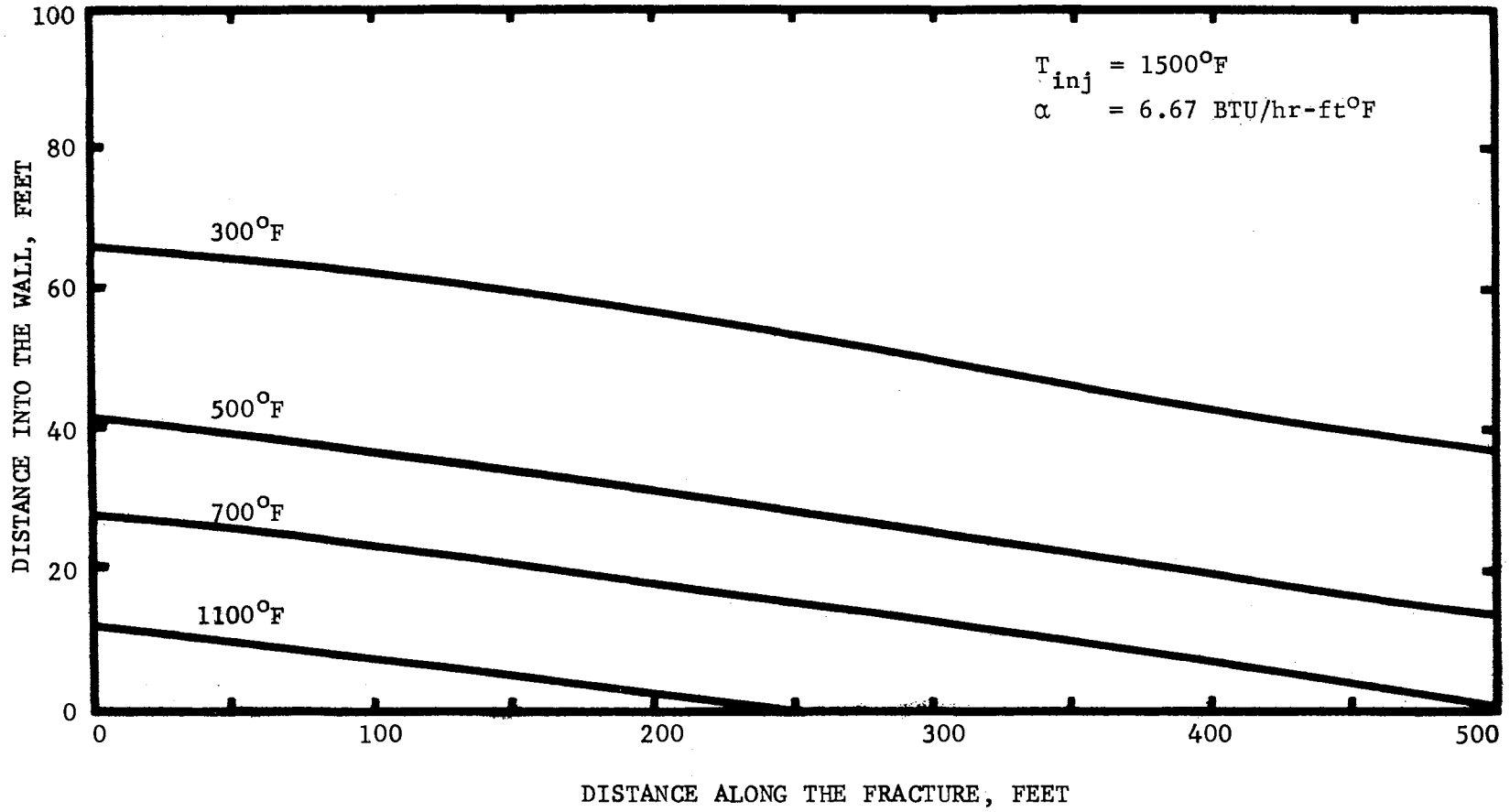


Figure 23. Temperature Distribution After 86,100 Hours -- Run 3

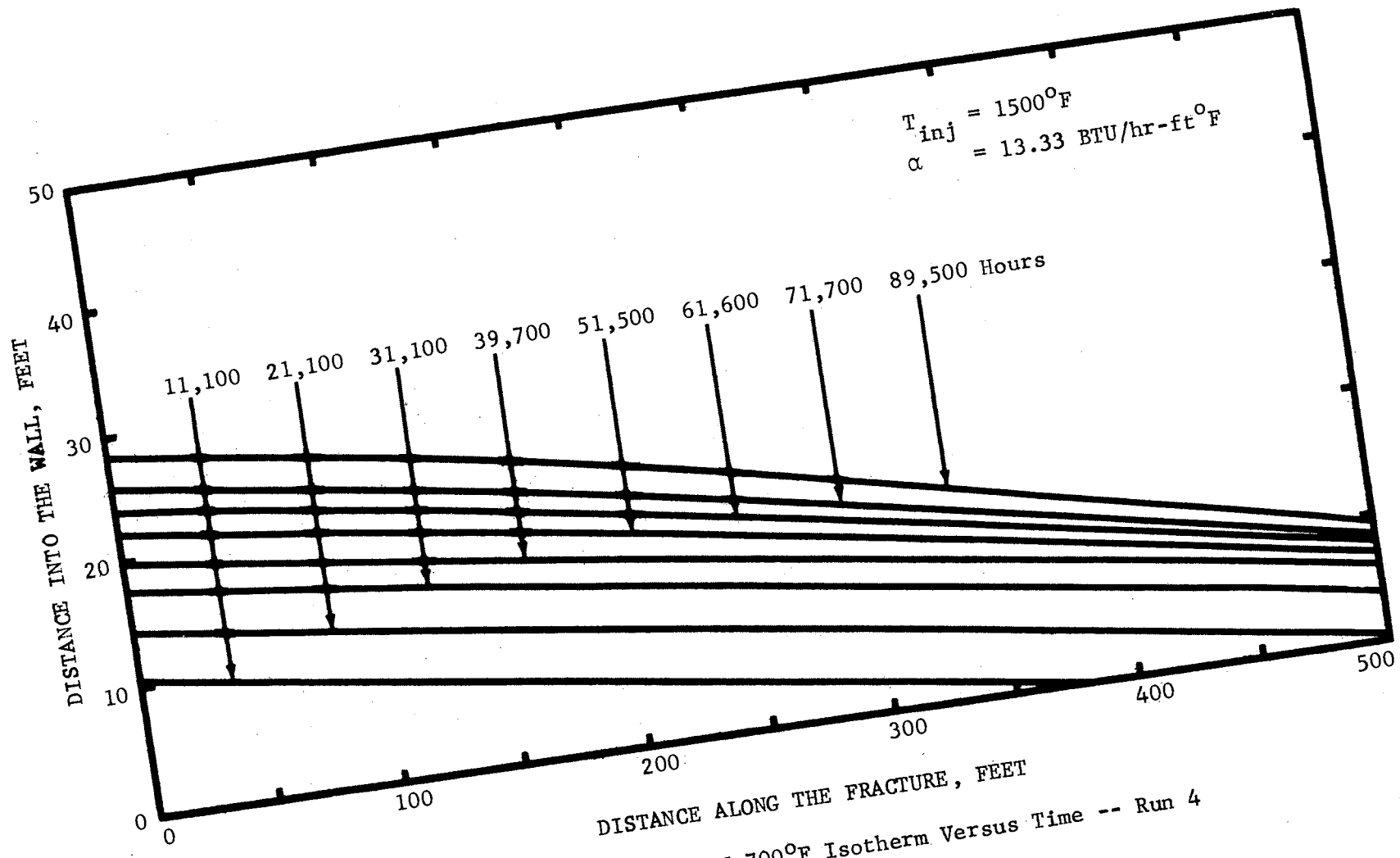


Figure 24. Position of 700°F Isotherm Versus Time -- Run 4

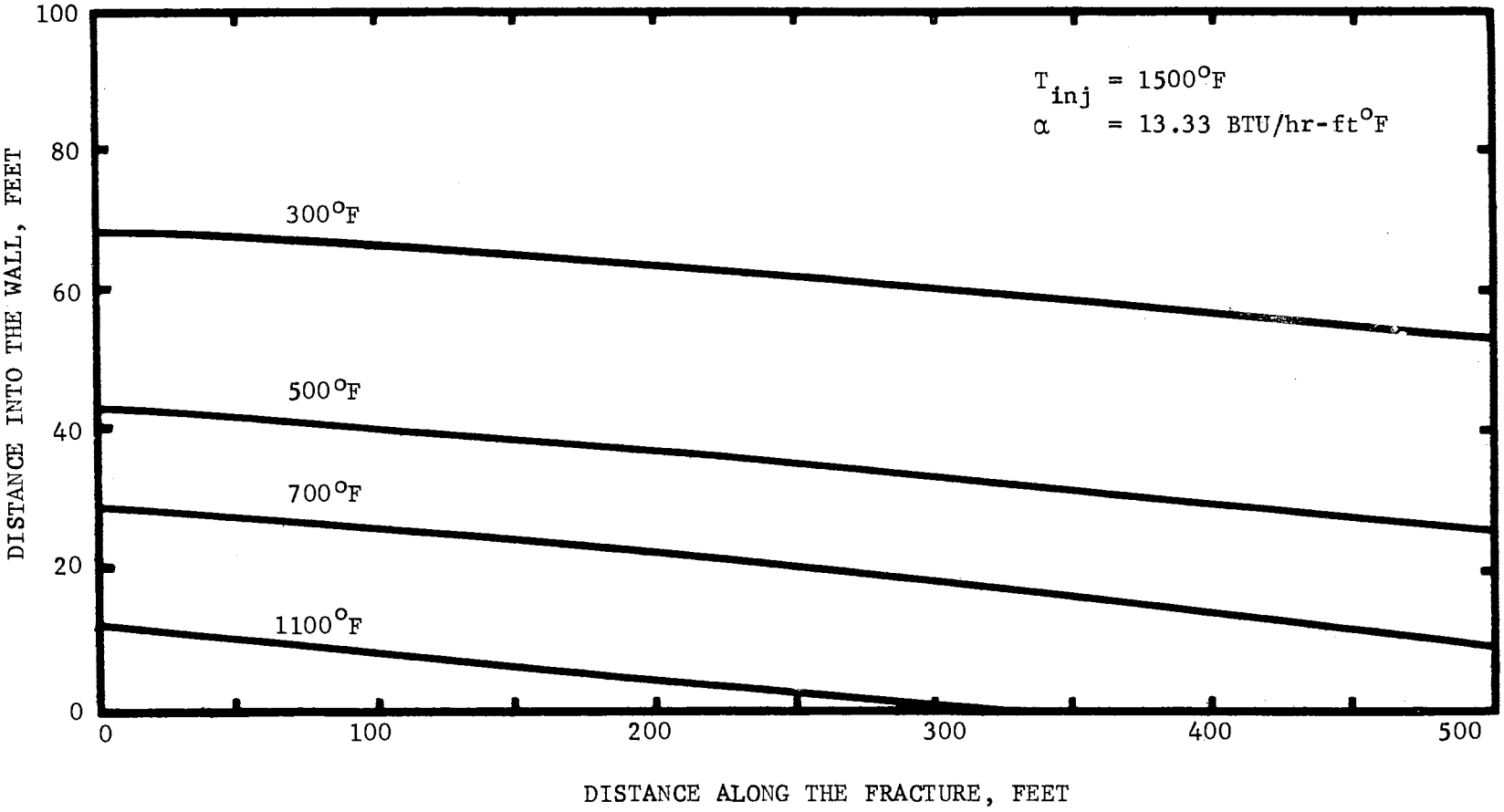


Figure 25. Temperature Distribution After 89,100 Hours -- Run 4

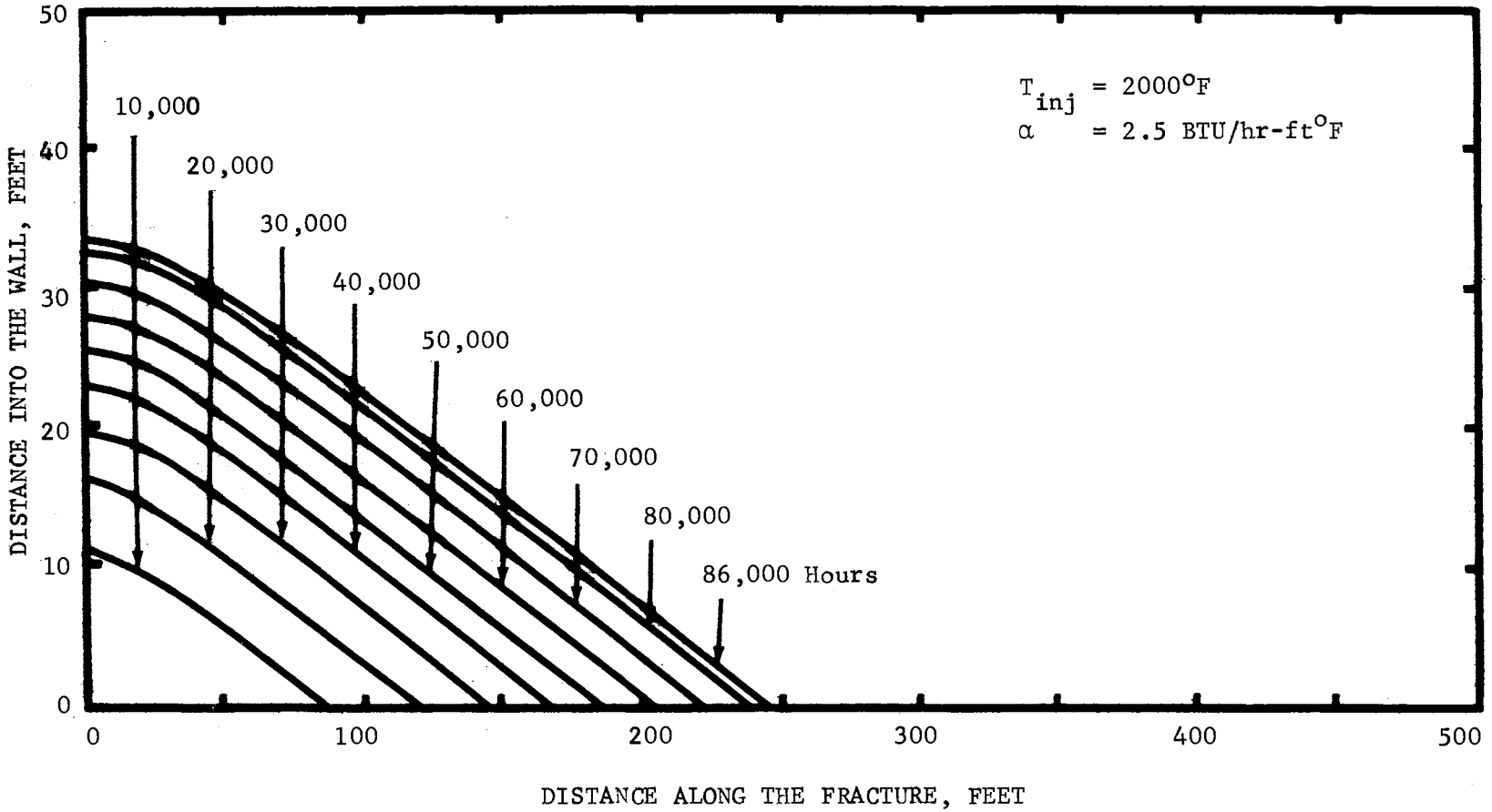


Figure 26. Position of 700°F Isotherm Versus Time -- Run 5

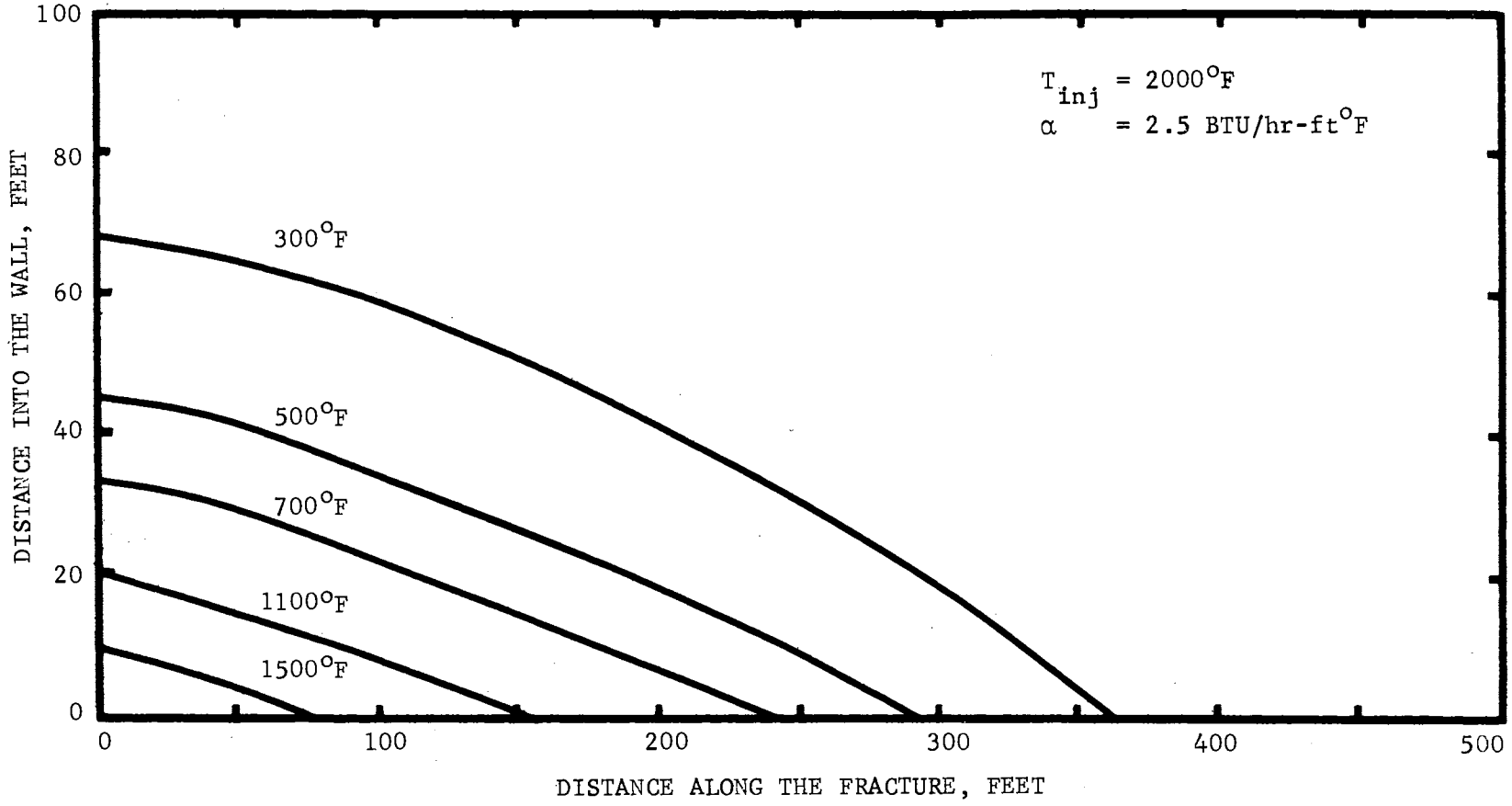


Figure 27. Temperature Distribution After 86,100 Hours -- Run 5

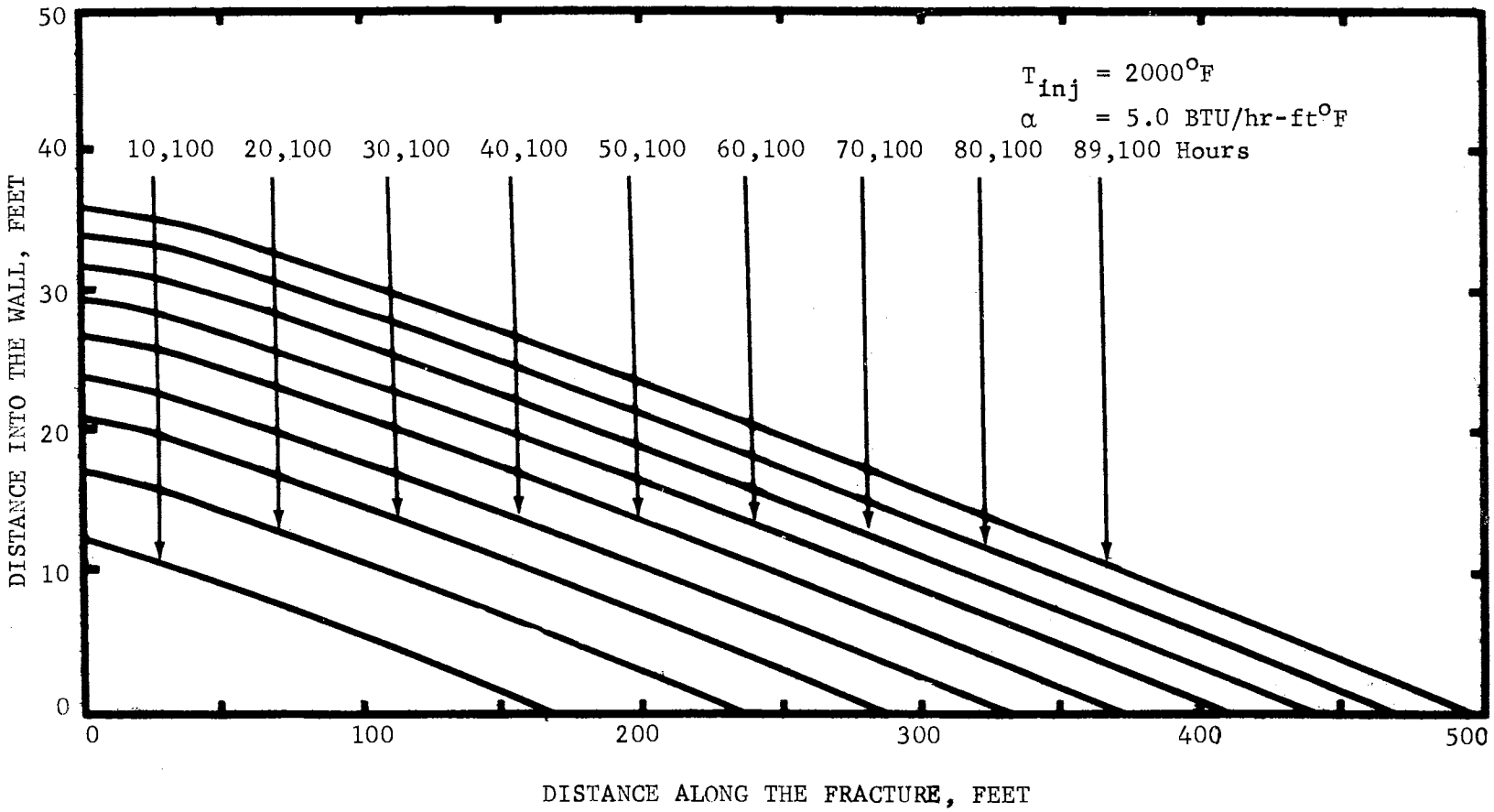


Figure 28. Position of 700°F Isotherm Versus Time -- Run 6

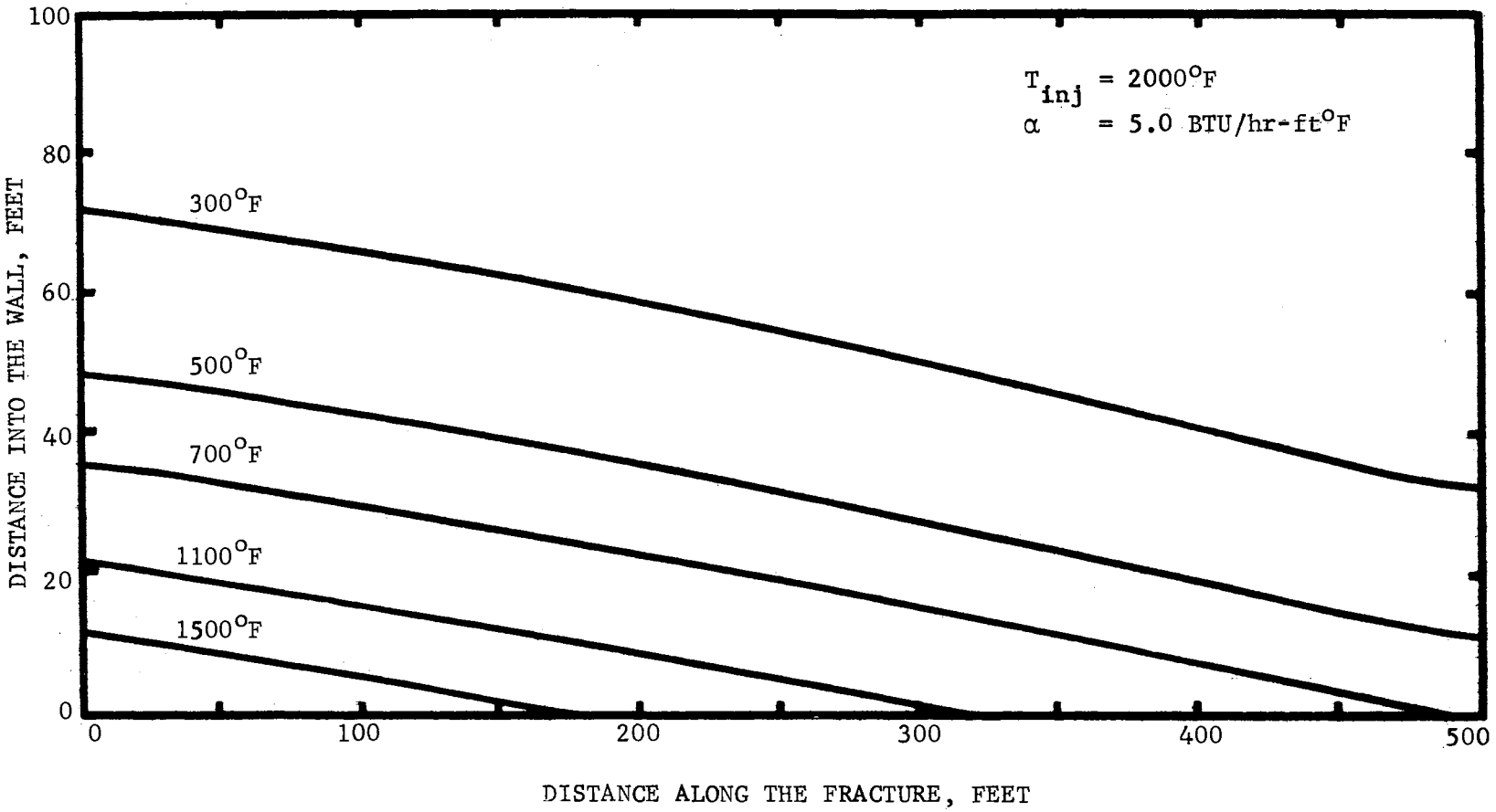


Figure 29. Temperature Distribution After 86,100 Hours -- Run 6

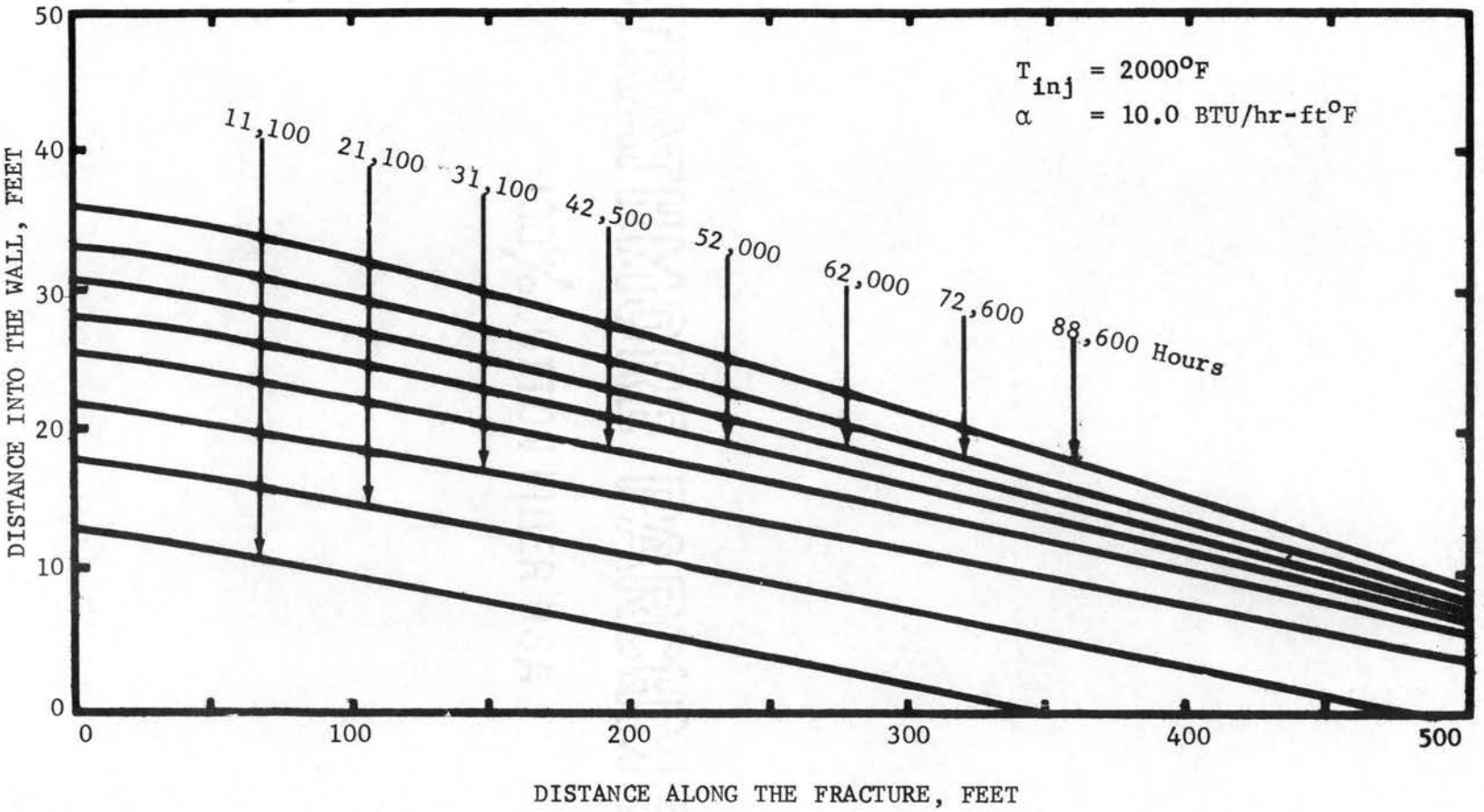


Figure 30. Position of 700°F Isotherm Versus Time -- Run 7

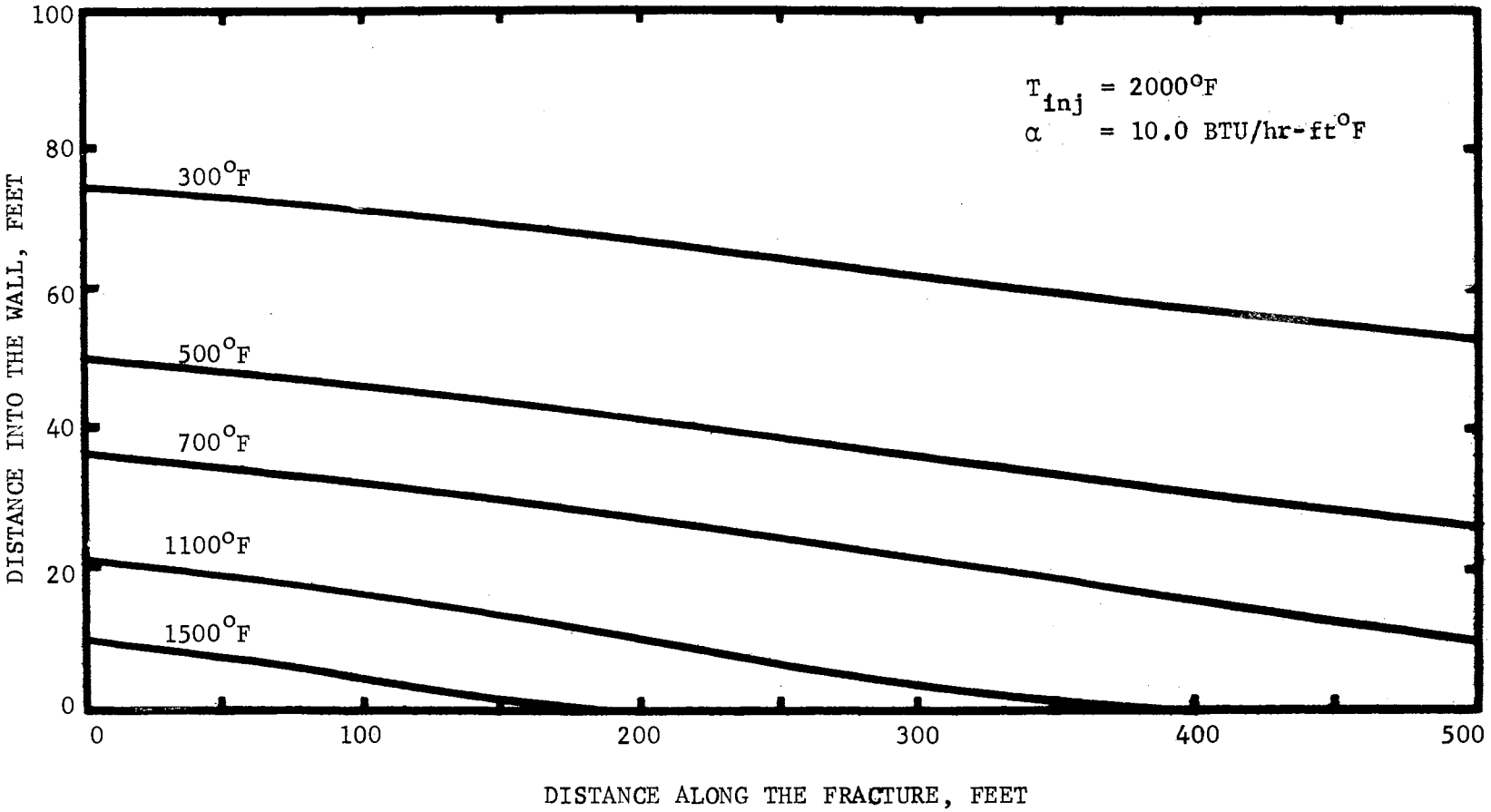


Figure 31. Temperature Distribution After 88,600 Hours -- Run 7

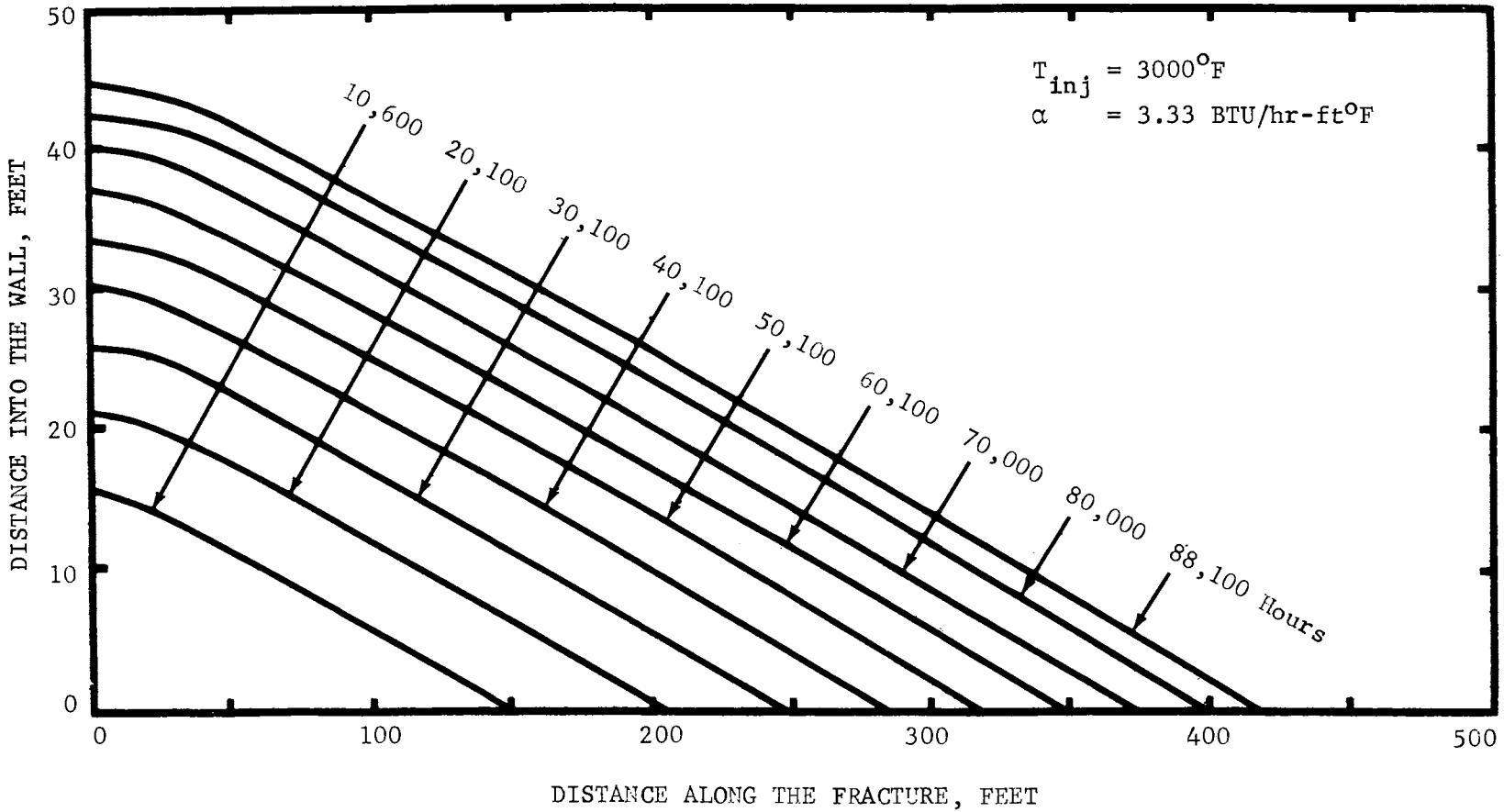


Figure 32. Position of 700°F Isotherm Versus Time -- Run 8

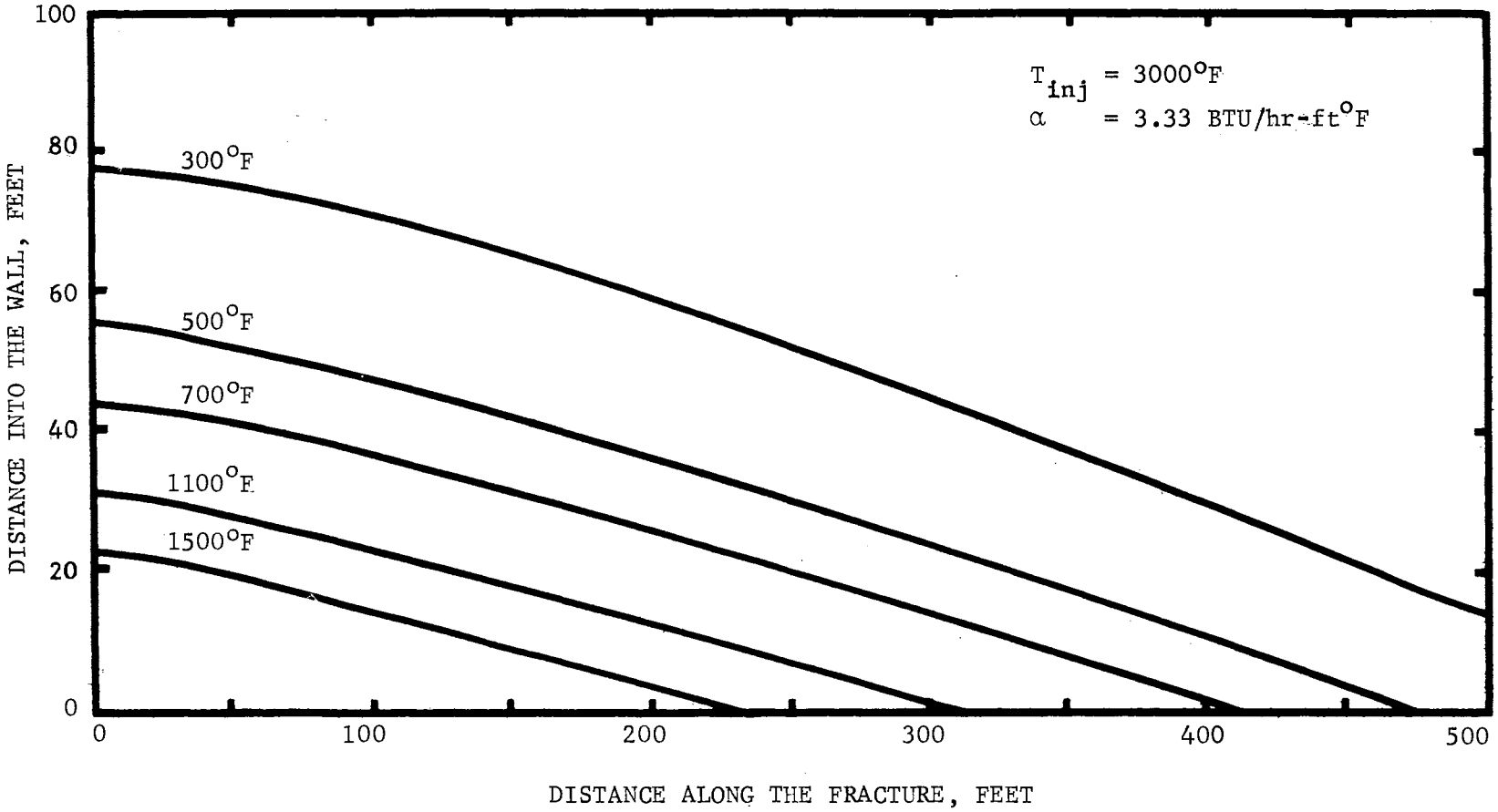


Figure 33. Temperature Distribution After 88,100 Hours -- Run 8

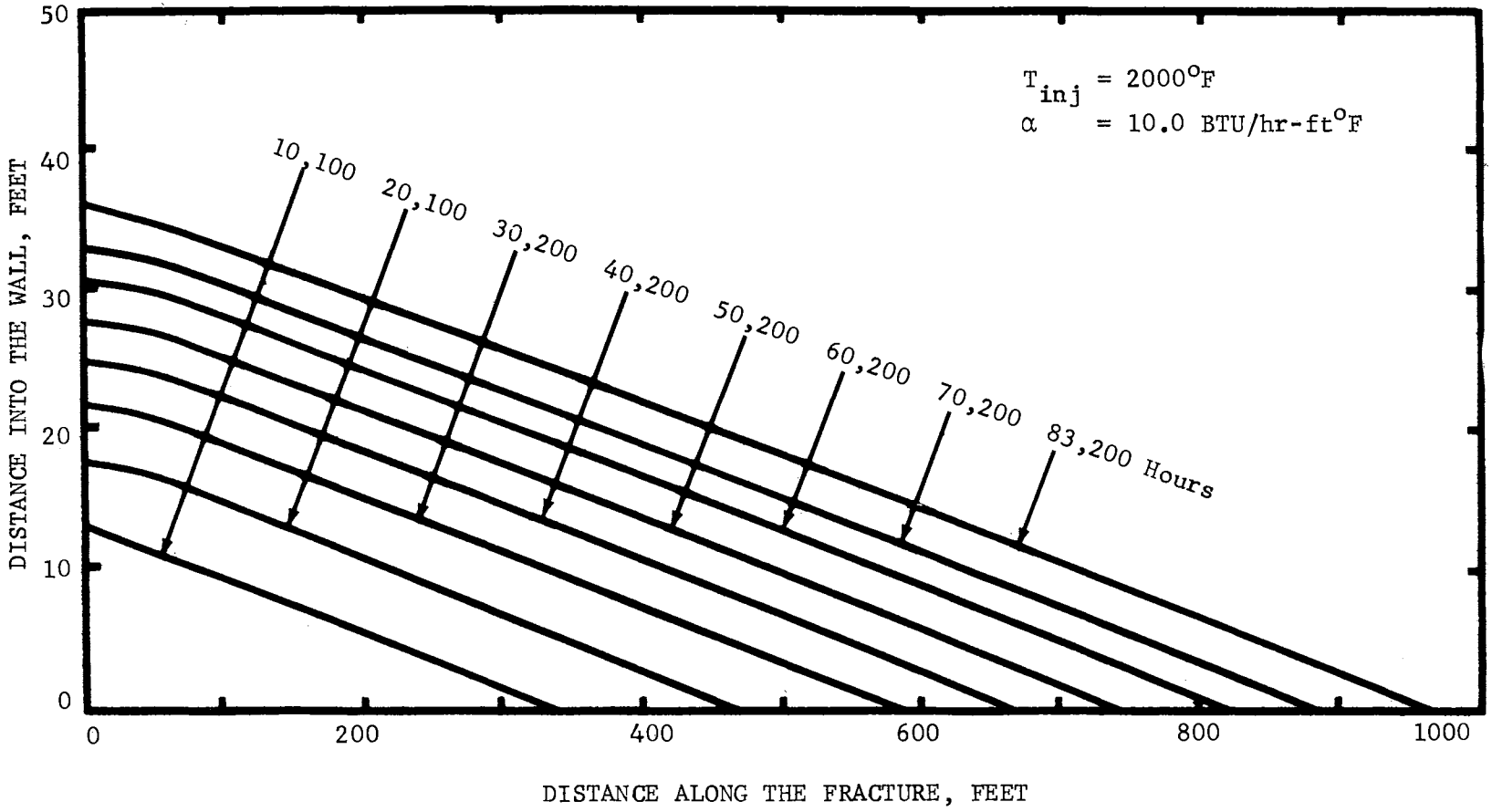


Figure 34. Position of 700°F Isotherm Versus Time -- Run 9

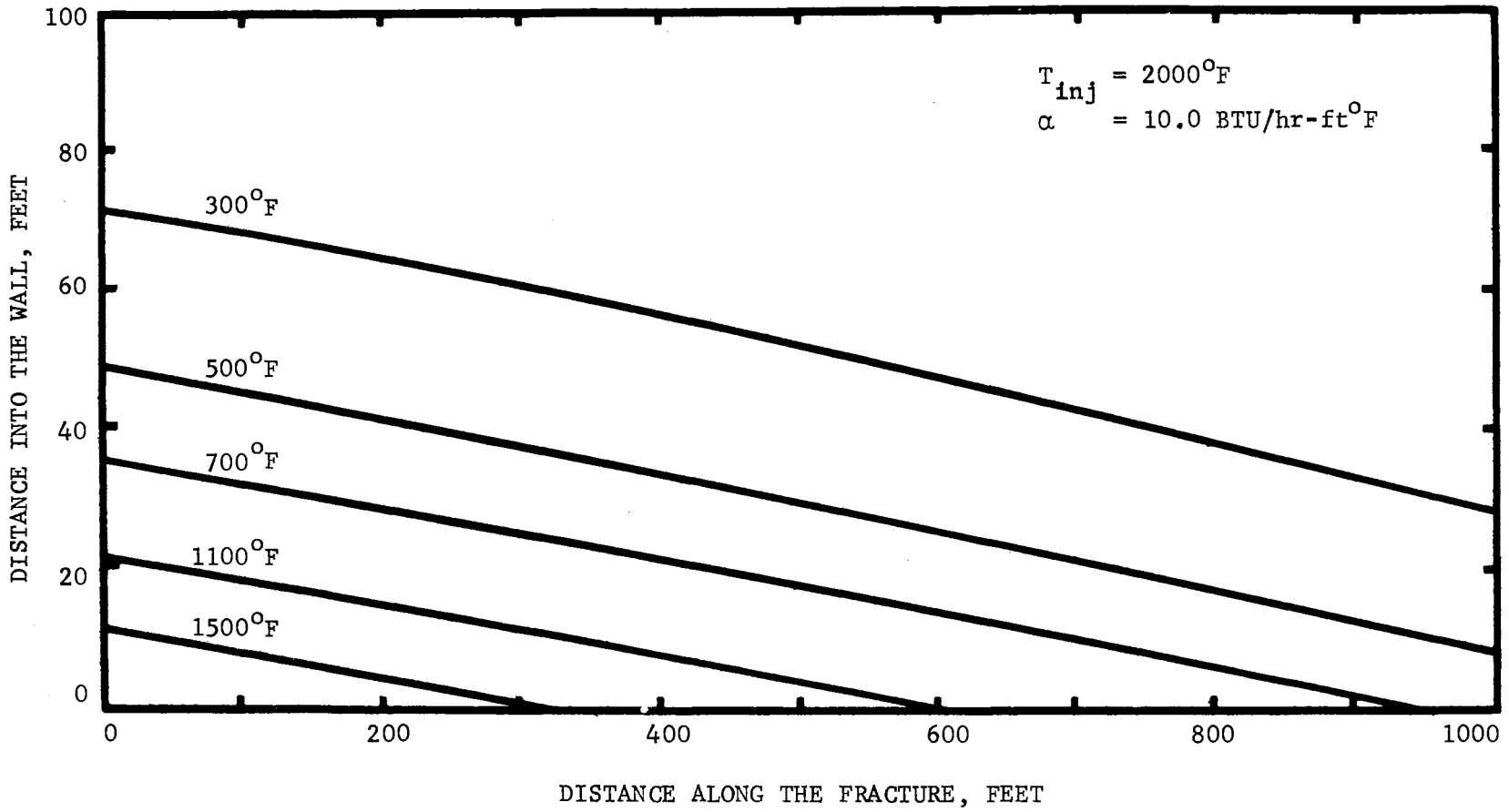


Figure 35. Temperature Distribution After 83,000 Hours -- Run 9

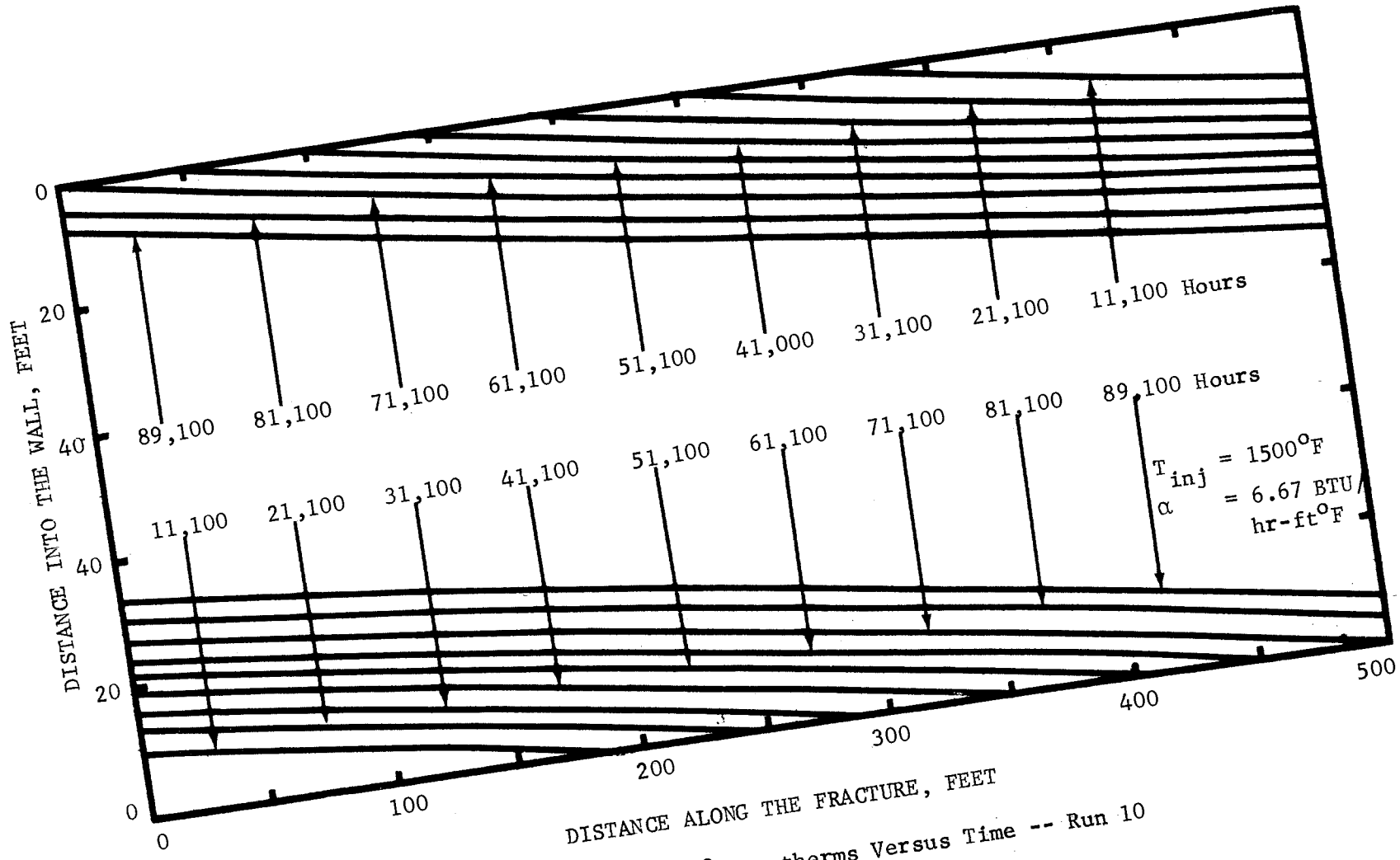


Figure 36. Positions of 700°F Isotherms Versus Time -- Run 10

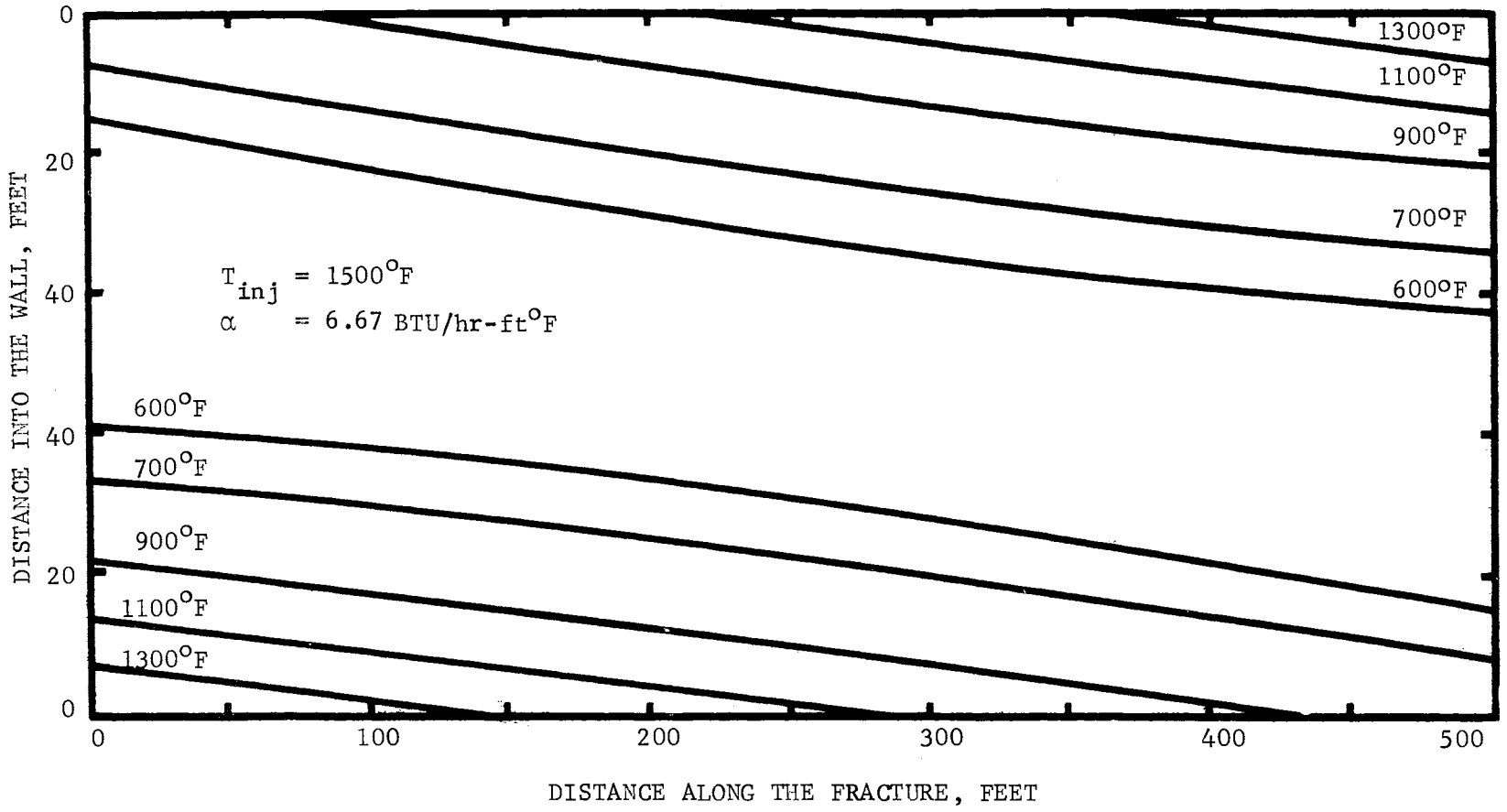


Figure 37. Temperature Distribution After 86,100 Hours -- Run 10

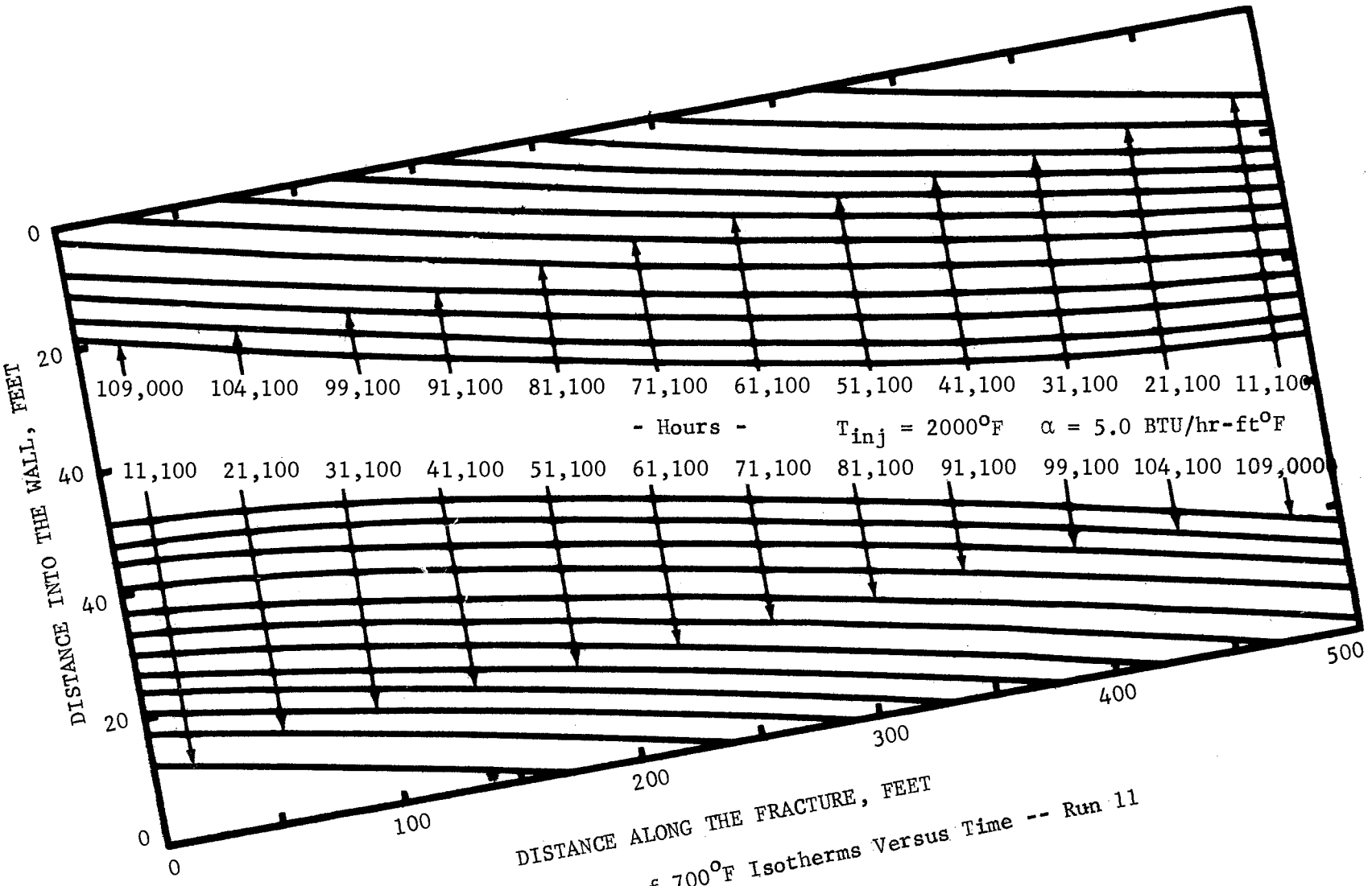


Figure 38. Positions of 700°F Isotherms Versus Time -- Run 11

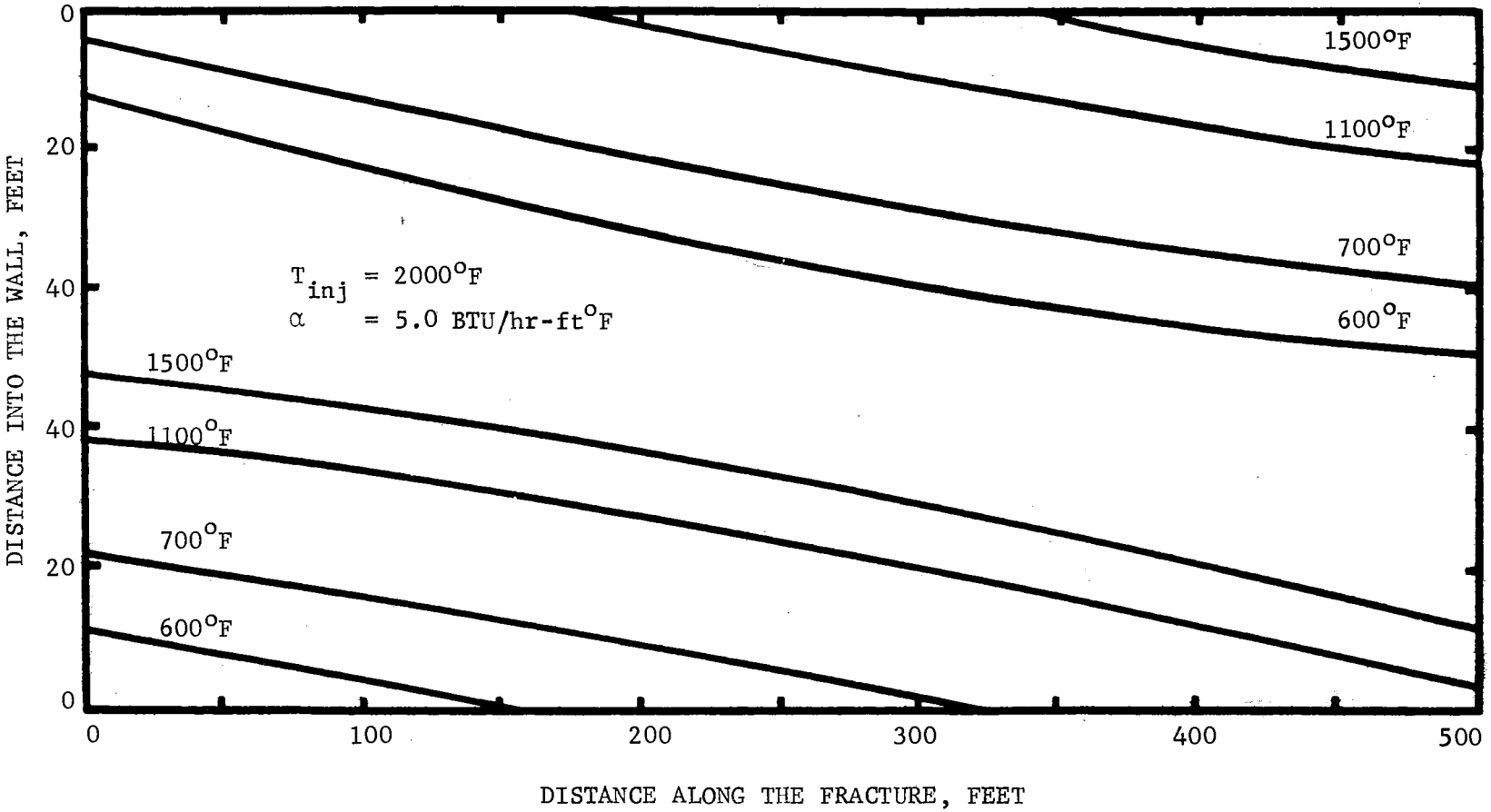


Figure 39. Temperature Distribution After 86,100 Hours -- Run 11

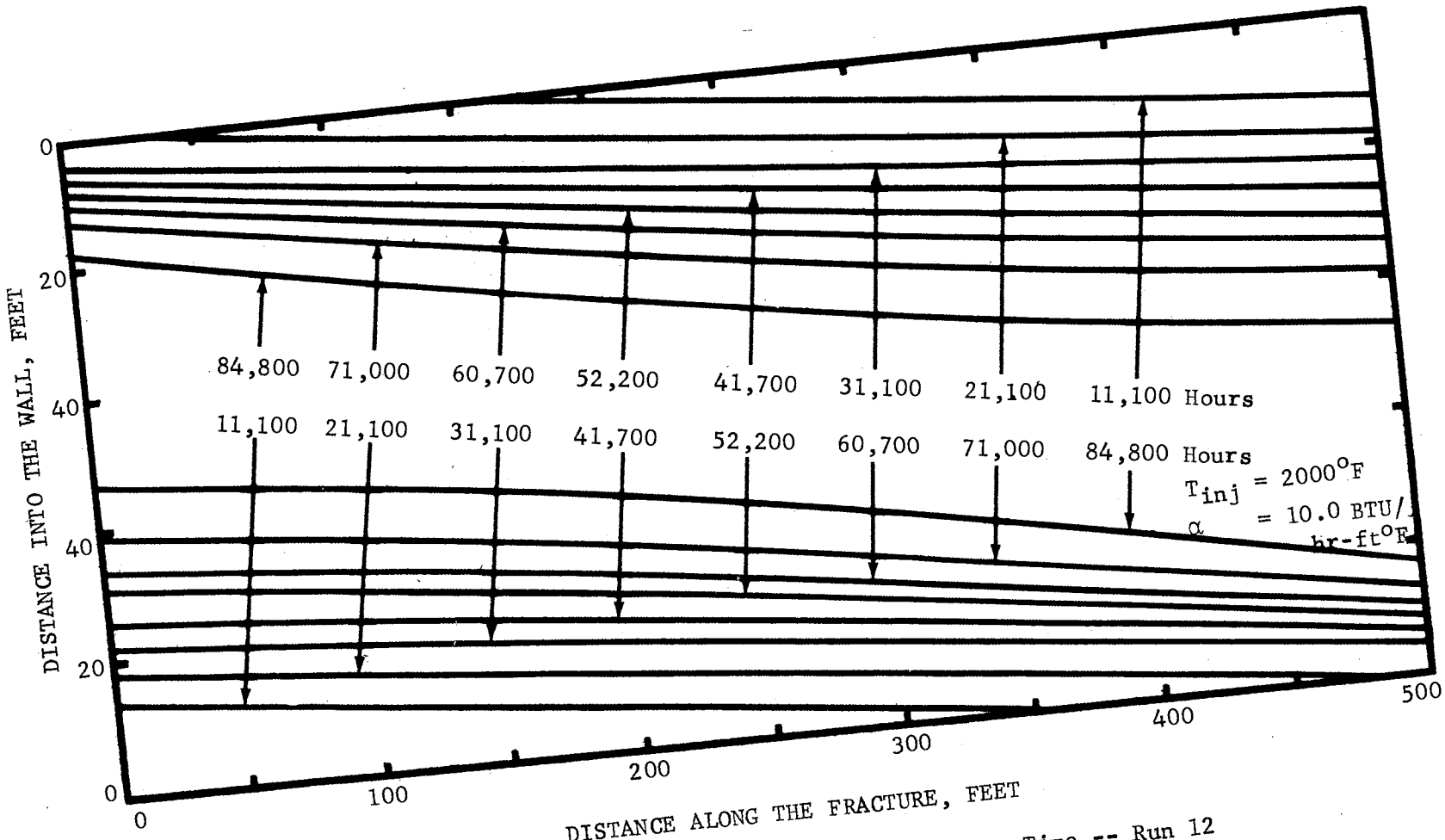


Figure 40. Positions of 700°F Isotherms Versus Time -- Run 12

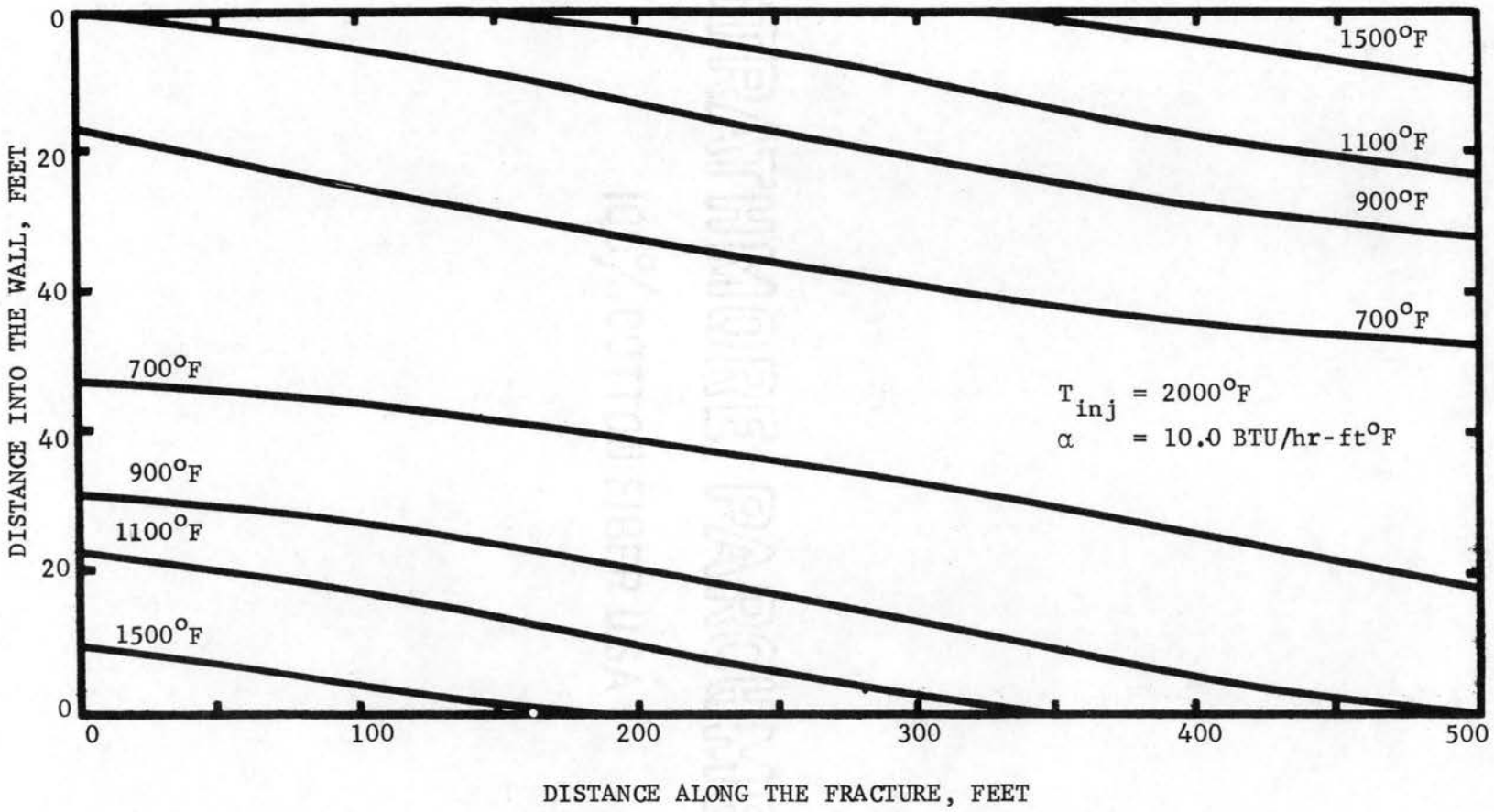


Figure 41. Temperature Distribution After 84,800 Hours -- Run 12

isotherm versus time. The second is the temperature distribution at the end of 86,100 hours. Notice that in the case of the Model II runs, double countercurrent plots are presented. The temperature distribution curves are given to serve as reference information. For those runs where the "soak" alternative was studied, positions of the 700°F at the end of selected injection periods were plotted to insure that the farthest advance of the 700°F was recorded. For these runs the final temperature distribution was listed at the end of the injection cycle closest to 86,100 hours.

The plots of the positions of the 700°F isotherms versus time are the basic curves from which all other comparison curves were constructed. The areas under the 700°F isotherm at any time represent the volumes of oil shale retorted per foot of fracture thickness as calculated by the two-dimensional models. These volumes are compared with the parameters of the process in Figures 42 through 59.

In order to obtain accurate volumes of shale retorted, large plots of the positions of the 700°F isotherms versus time were made. The areas under these large plots were determined with a planimeter. Planimeter error averaged less than 1 per cent.

Fractional cumulative heat balance errors ranged from 0.02 to 0.14; however, heat balance error was found to lie within 0.05 to 0.10 for most cases. Two soak runs had the lowest errors and the 3000°F run had the largest error. It is noted that all heat balance errors were in the same direction, i.e., cumulative stored heat plus produced heat was greater than cumulative injected heat. With this in mind results from these runs can be considered on the optimistic side. However, retorted volumes determined here are not necessarily in error

to the same degree as the heat balance because only the position of the 700°F isotherm is involved.

The previously described iteration parameters developed with THERMAB3 for this work were satisfactory for all runs except for run 9. This was the only case studied with a 1000-foot fracture length. In this case, the time increment was held at 50 hours up to 30,000 hours of history instead of the usual 10,000 hours. This was necessary to get convergence within the prescribed tolerances.

Effect of Temperature

The effect of injection temperature upon retorting rate at several heat injection rates is shown in Figures 42, 43 and 44. These figures show that rates of retorting are all constant up to the time that the 700°F isotherm arrives at the producing well. For the cases shown in Figures 42 and 43 the 700°F isotherms do not reach the producers during the durations of the runs. At the highest injection rate of 20,000 BTU/hr-ft, it can be seen from Figure 44 that the rate of retorting for the 1500°F case decreases at approximately 20,000 hours; whereas, for the 2000°F case, it decreases at 25,000 hours. Figures 24 and 30 show at these times the 700°F isotherms have reached the producers. When the producers reached 900°F the alternating soak-injection routine was used. Figure 44 also shows that for the 2000°F, 1000-foot case, its rate of retorting remained constant, and therefore the 700°F isotherm did not reach the producer during the run.

Figures 42, 43 and 44 clearly show that the rate of retorting increases as the injection temperature is increased when compared at the same heat injection rate. Figure 43, which compares retorting

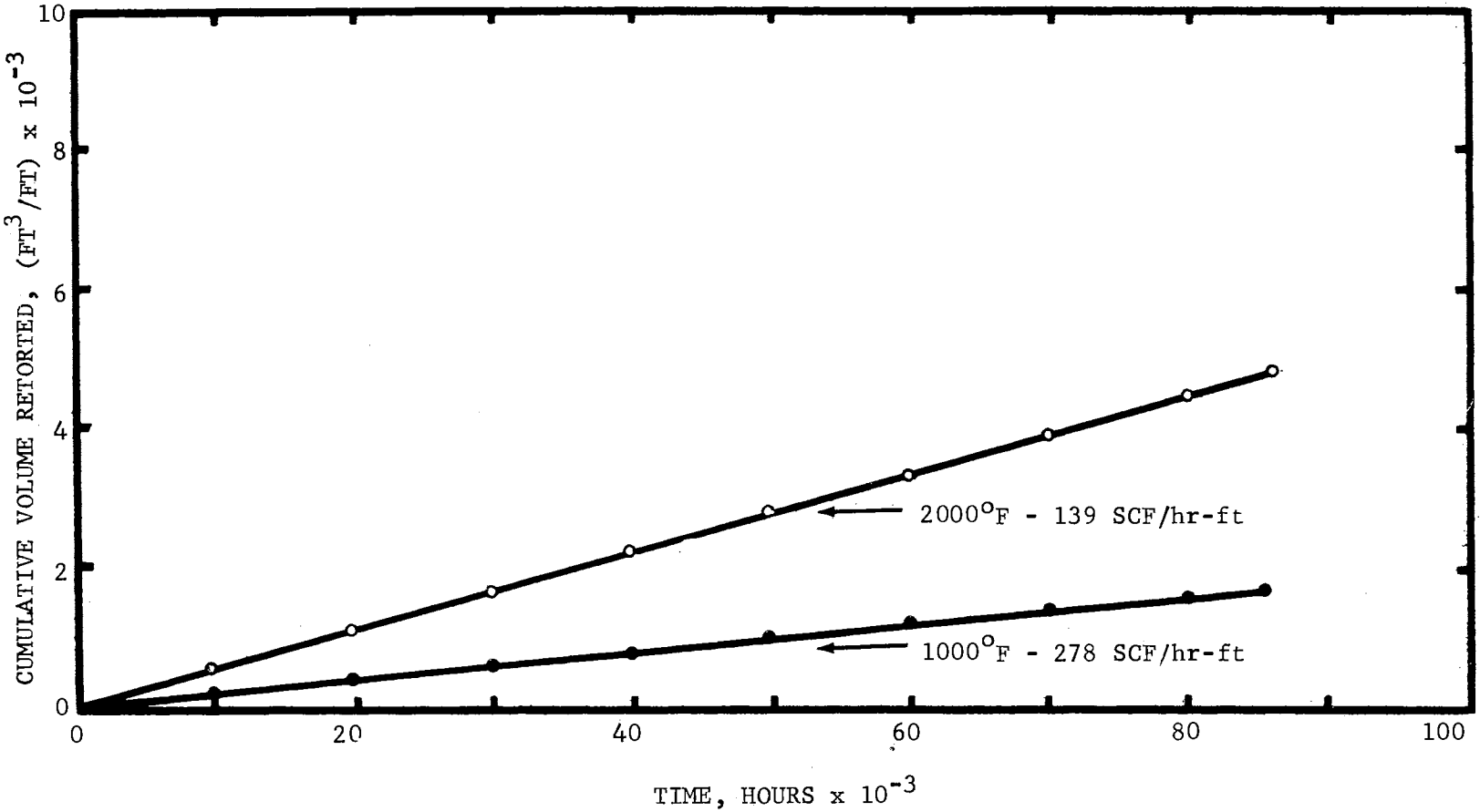


Figure 42. Effect of Temperature upon Retorting Rate with a Heat Injection of 5000 BTU/hr-ft

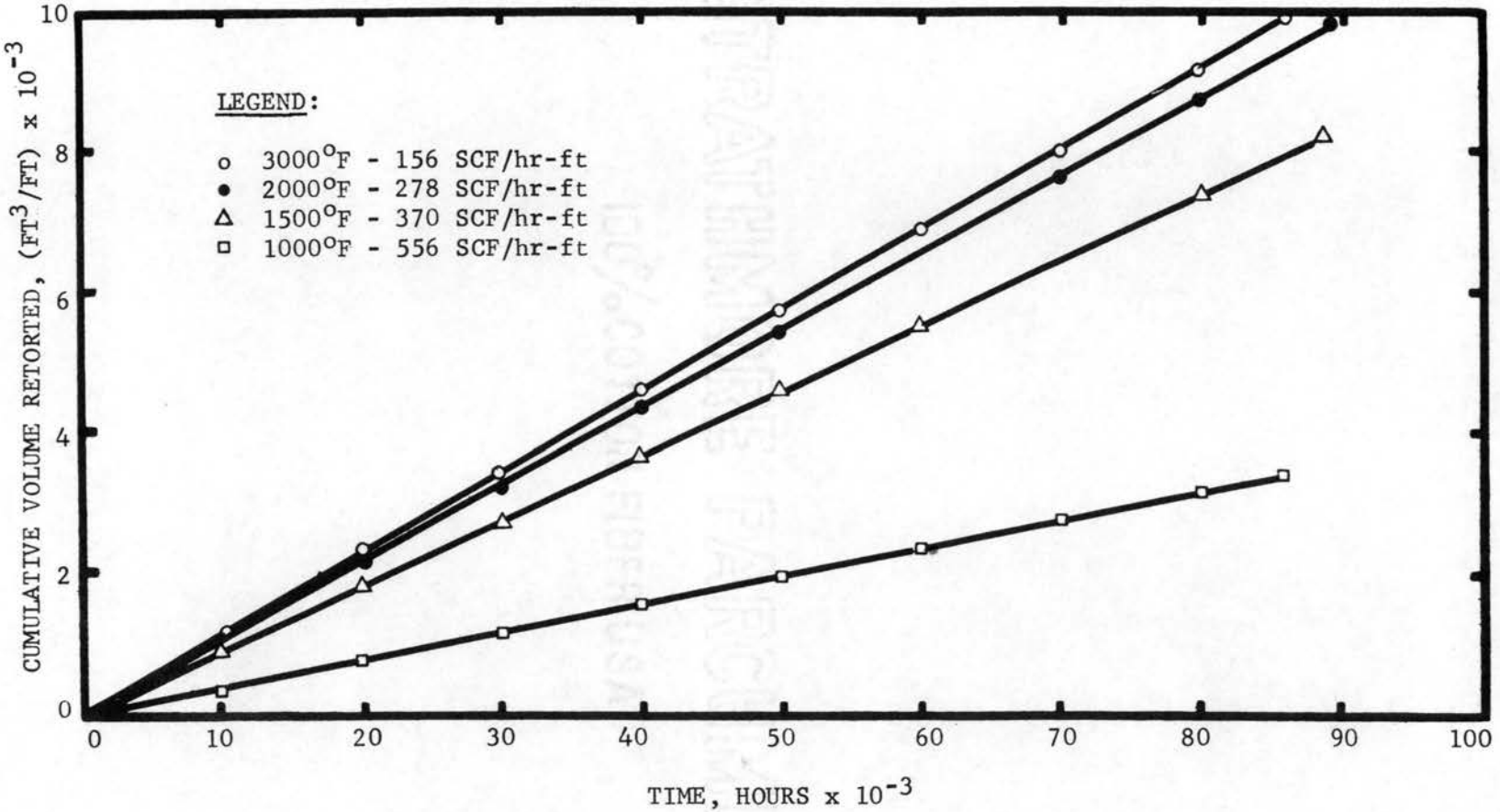


Figure 43. Effect of Temperature upon Retorting Rate with a Heat Injection Rate of 10,000 BTU/hr-ft

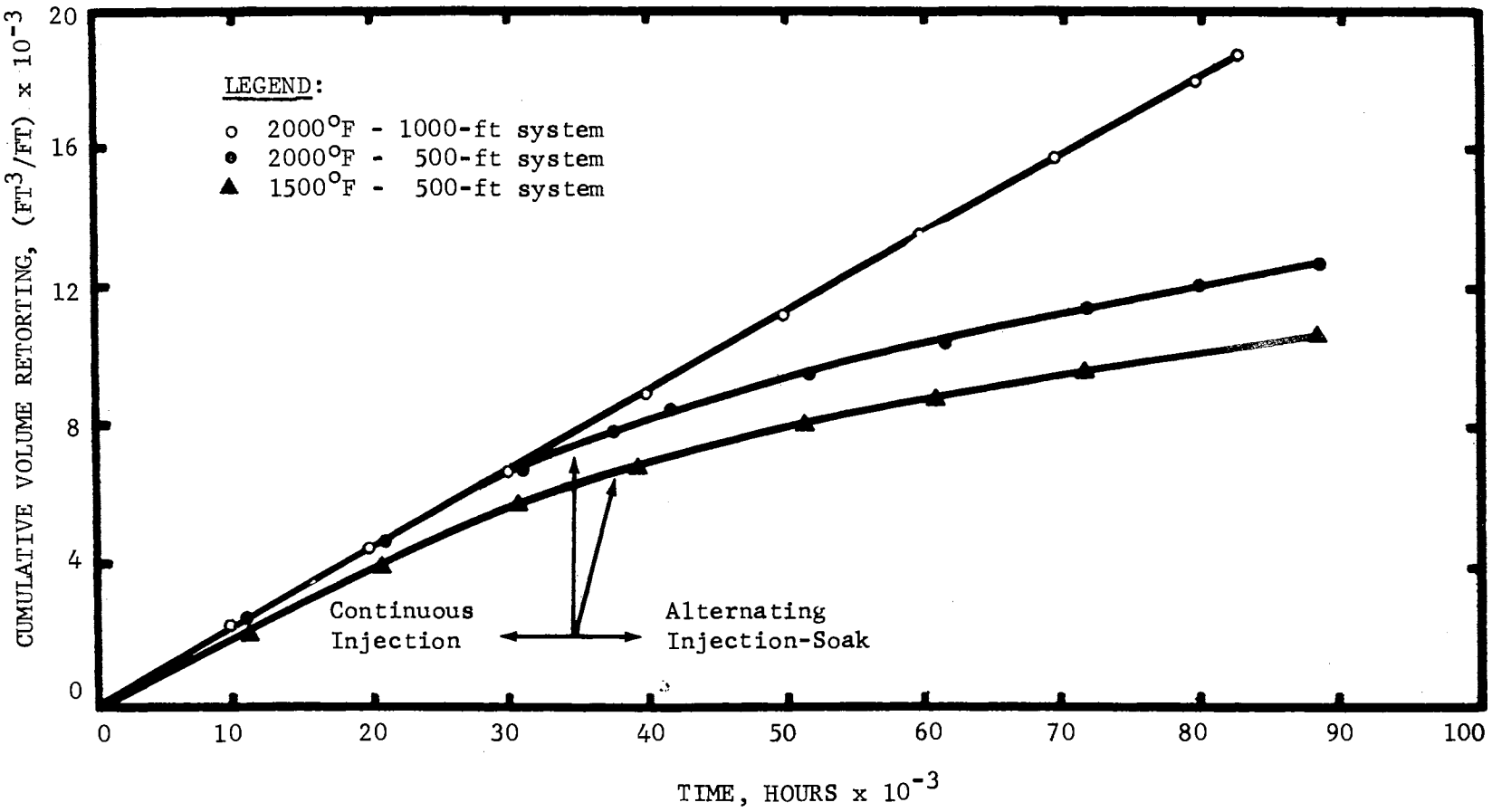


Figure 44. Effect of Temperature upon Retorting Rate with a Heat Injection Rate of 20,000 BTU/hr-ft

rates at four temperatures, shows the rate of retorting is increased to a smaller extent as the injection temperature increases. Rates of retorting at 1500, 2000 and 3000°F are 2.35, 2.80 and 2.94 times the rate at 1000°F.

Results from these three figures are all the more striking since, for equal heat injection rates, it is necessary that the air injection for the 1000°F case be approximately twice that of the 2000°F case. This means that the air compression cost for the 1000°F case would be twice that of the 2000°F case, while the retorting rate is only about 35 per cent of the 2000°F case.

Figures 45 and 46 compare retorting rates at 1000 and 2000°F for two gas injection rates. These comparisons are more significant as compression costs would be the same. These comparisons show the rate of retorting at 2000°F is approximately 5.7 times that at 1000°F when compared at the same gas injection rate.

Effect of Injection Rate

Figures 47, 48, and 49 show the effect of injection rate for a constant gas injection temperature. These figures show that up until the time the 700°F isotherm reaches the producer, the retorting rate is a direct function of injection rate when compared at the same temperature. After "breakthrough" of the 700°F isotherm, retorting efficiency falls off.

Breakthrough of the 700°F isotherm occurs at 90,000 hours for a heat injection rate of 10,000 BTU/hr-ft at 2000°F (see Figure 28). If this injection rate is doubled, using the same 500-foot system, then breakthrough occurs at 25,000 hours (see Figure 30). If the

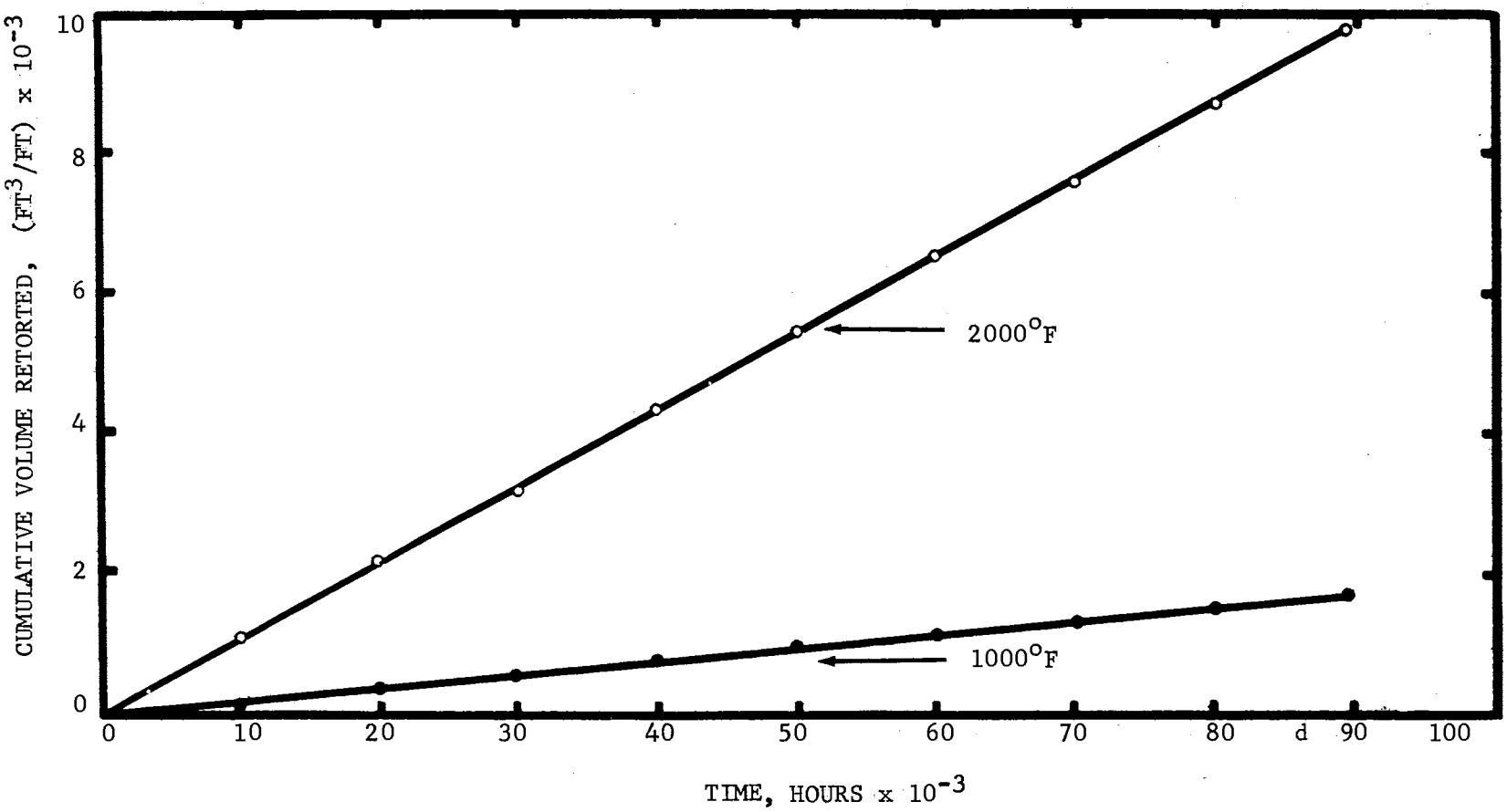


Figure 45. Effect of Injection Temperature upon Retorting Rate with an Air Injection Rate of 278 SCF/hr-ft

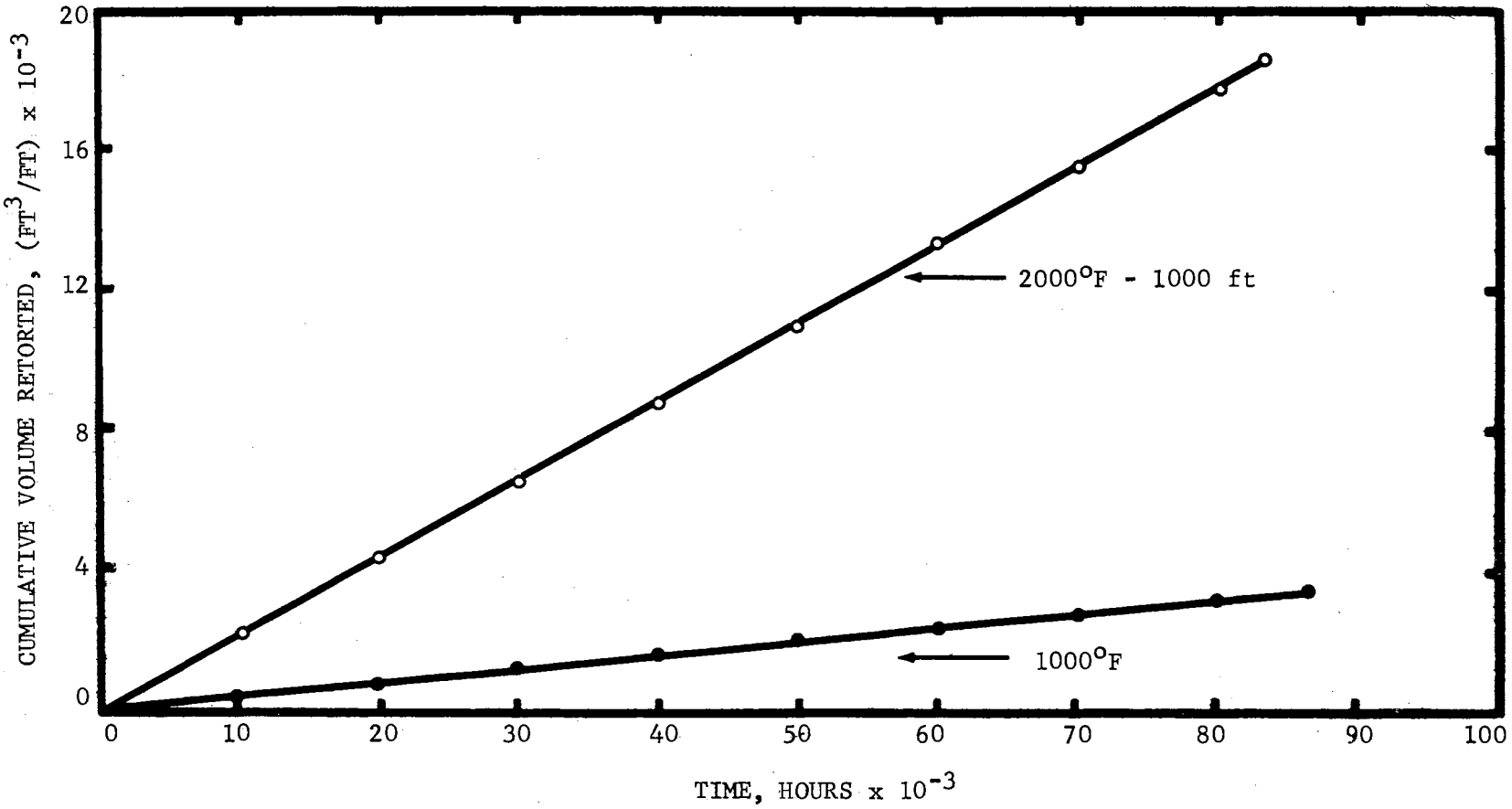


Figure 46. Effect of Injection Temperature upon Retorting Rate with an Air Injection Rate of 556 SCF/hr-ft

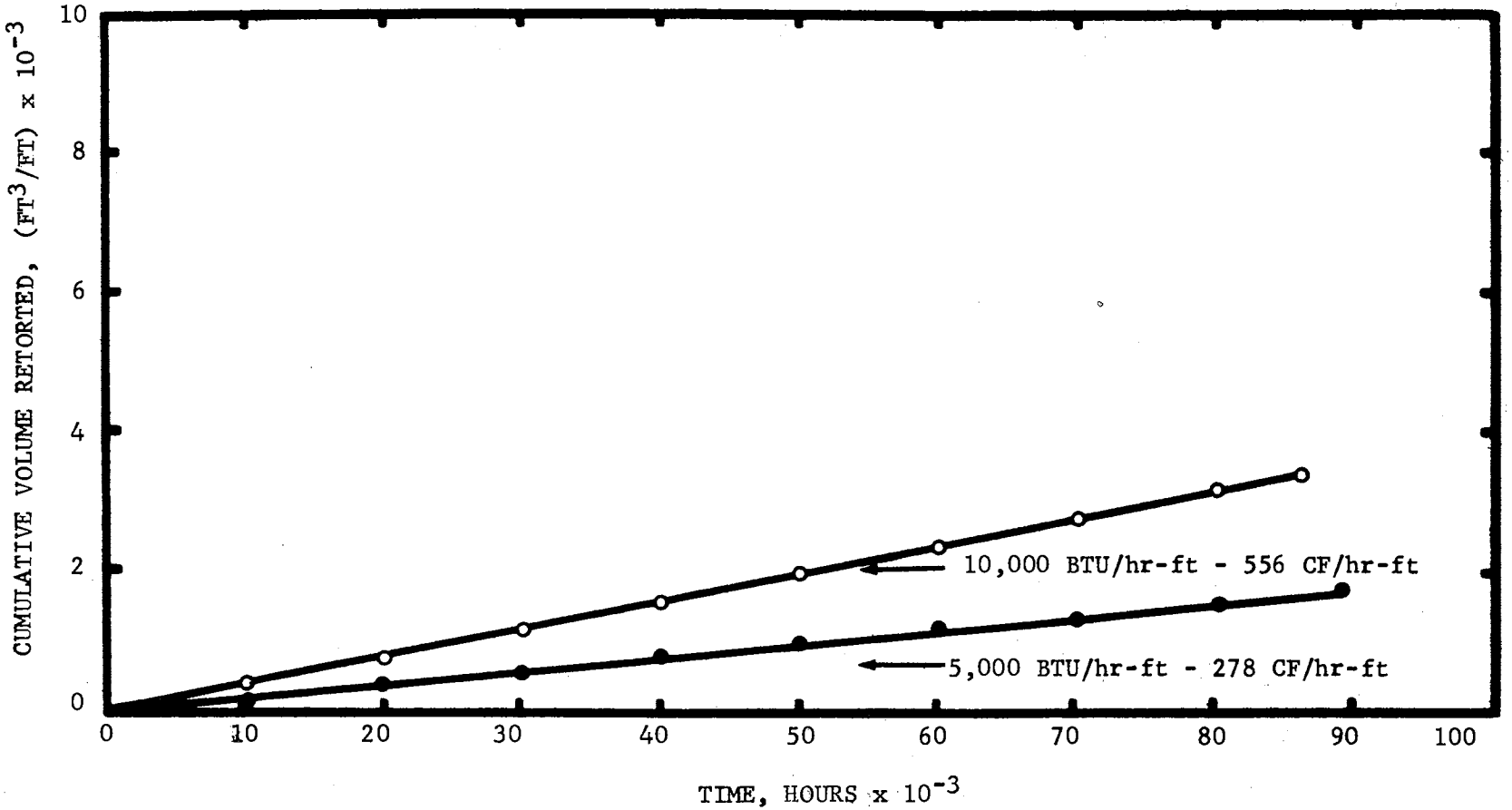


Figure 47. Effect of Injection Rate upon Retorting Rate with an Injection Temperature of 1000°F

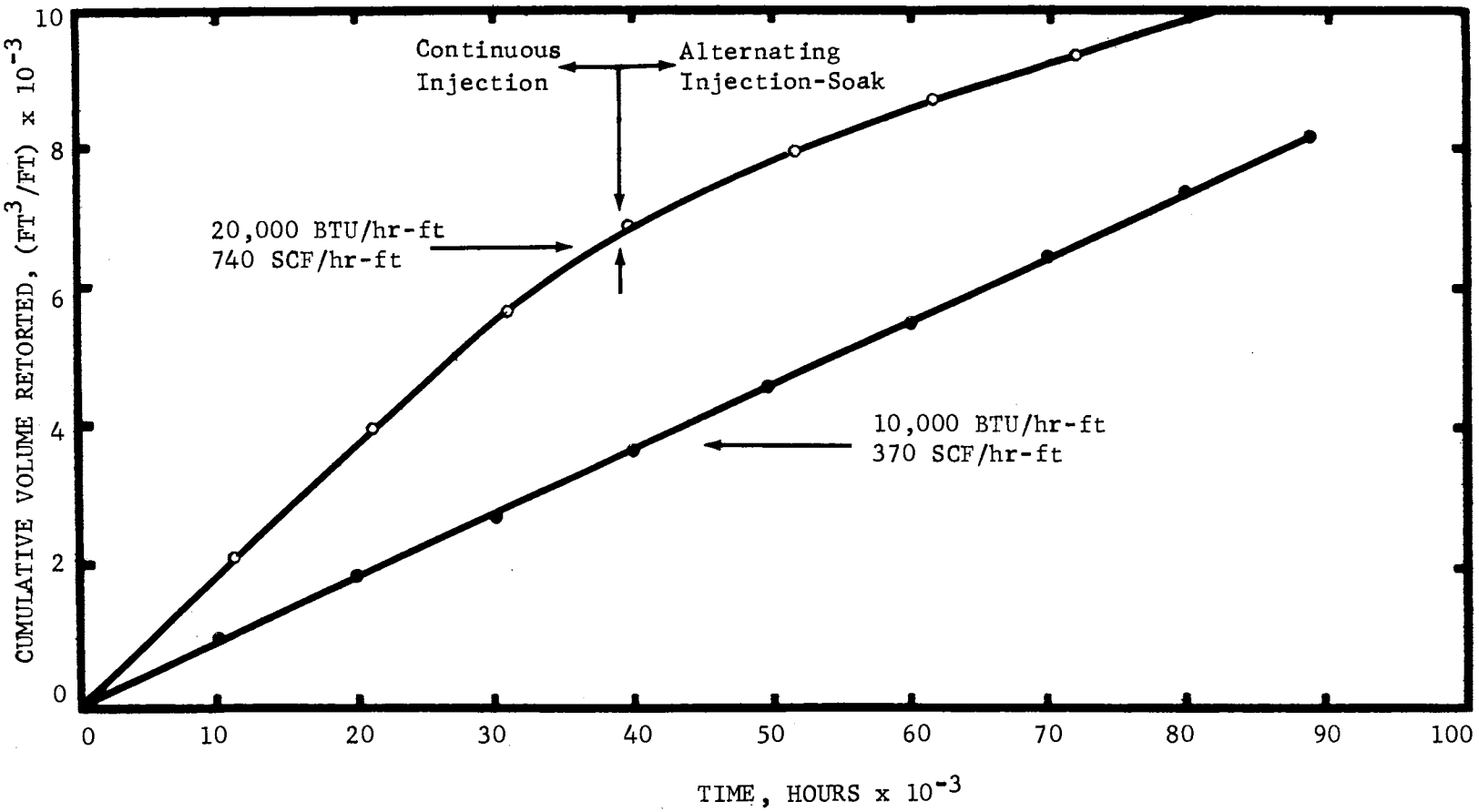


Figure 48. Effect of Injection Rate upon Retorting Rate with an Injection Temperature of 1500°F

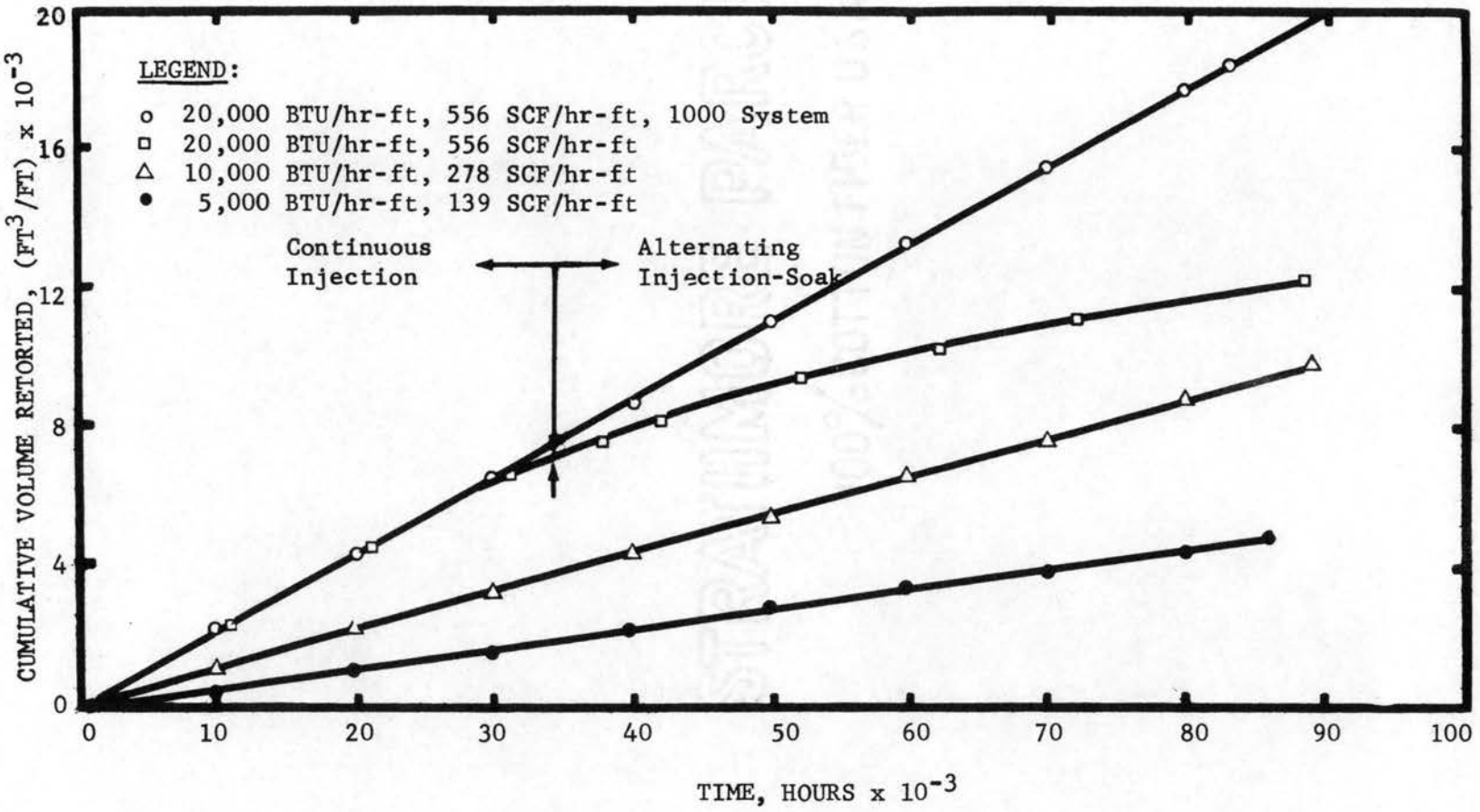


Figure 49. Effect of Injection Rate upon Retorting Rate with an Injection Temperature of 2000°F

injection rate is 20,000 BTU/hr-ft but the system length is doubled to 1000 feet, then breakthrough again occurs beyond 80,000 hours (see Figure 34). These comparisons show that for a specified project life (here taken to be 86,100 hours or approximately 10 years), there is an optimum injection rate-system length combination and they are almost directly related. In other words, doubling the injection rate requires doubling the system's length for 700°F isotherm breakthrough to occur within the same period.

A comparison at a rate of 10,000 BTU/hr-ft shows that after 86,100 hours, the 700°F isotherms are 500, 480 and 410 feet down the fracture, respectively, for the 1500°F, 2000°F and 3000°F injection temperatures (see Figures 22, 28 and 32). This behavior can be understood when it is remembered that the gas injection rate is inversely proportional to the injection temperature for a constant heat injection rate.

Effect of Soaking

For those cases where a producing well reached 900°F, heat injection was discontinued and the system was allowed to "soak." When the producing well temperature had decreased to 800°F, injection was resumed. The efficiency of the alternating soak-injection period is illustrated in Figures 50 and 51. Cumulative volume retorted is plotted versus cumulative heat injected. Prior to the soaking periods all runs show comparable retorting efficiencies at the same injection temperature. After the alternating injection-soak routine starts, the retorting efficiencies of these runs monotonically decrease with cumulative heat injected. The decrease is more dramatic when plotted against time as shown in earlier figures.

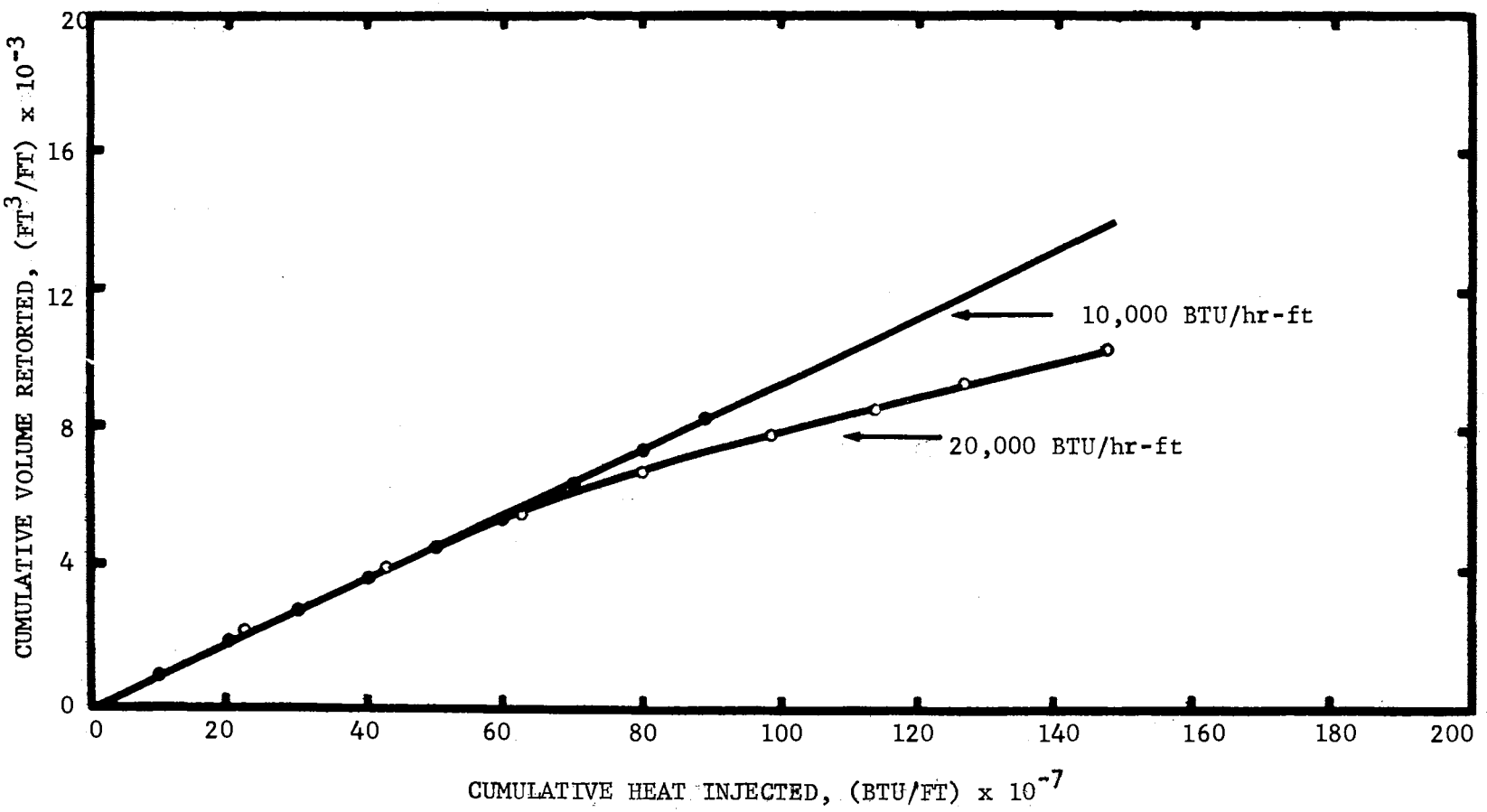


Figure 50. Effectiveness of Soaking with Injection Temperature of 1500°F

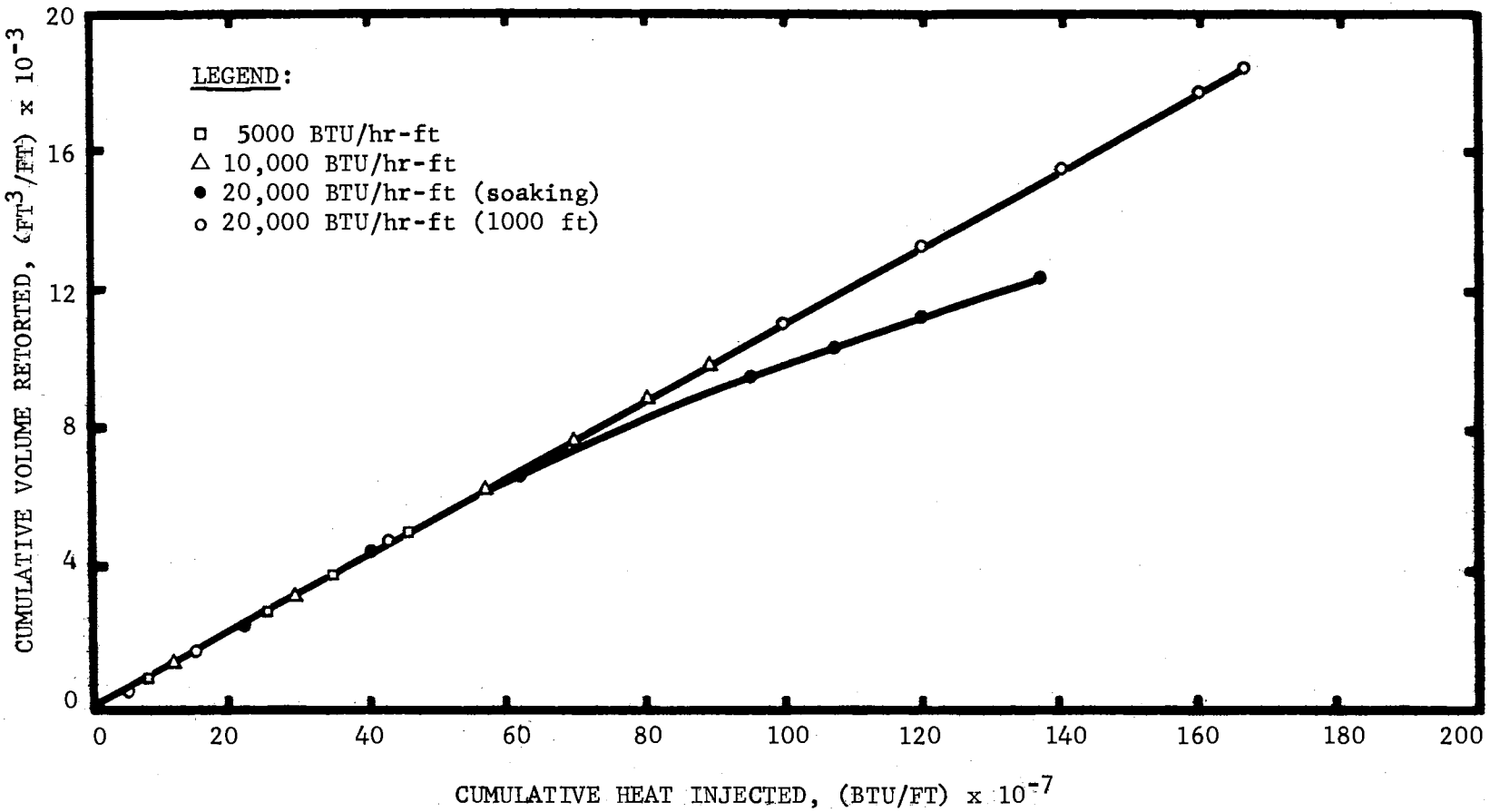


Figure 51. Effectiveness of Soaking with Injection Temperature of 2000°F

A comparison at cumulative heat injections of 120 x 10^7 BTU/ft shows the alternating injection-soak portion of a run to be only 60 per cent as efficient as the continuous injection portion using the 1500°F injection temperature. At 2000°F the same comparison shows an efficiency of 70 per cent. This therefore suggests it is preferable to increase the system's length or decrease the injection rate rather than use the soak technique.

Effect of Bounded Systems

Three runs were made for Model II (THERMAB4). Basic data from these runs are shown in Figures 36 through 41. Figures 52, 53 and 54 compare rates of retorting of bounded and unbounded systems using the same injection rates and temperatures. They show that retorting rates for bounded systems increase monotonically with time compared to the retorting rates for unbounded systems.

Final temperature distributions shown in Figures 37 and 39 after 86,100 hours indicate that the temperatures in the area between the retorted areas range between 500 and 600°F. In fact all temperatures in these areas exceed 550°F. The final temperature distribution shown in Figure 41 indicates all temperatures are greater than 600°F. These results show that with additional time, with or without injection, the entire volumes of the systems would be retorted.

Figure 38 shows the final positions of the 700°F isotherms after 109,100 hours. The isotherms are very close and all temperatures at this time were in excess of 650°F. This volume can be considered essentially completely retorted at this time. This run began the alternating injection-soak routine after 104,000 hours which indicates

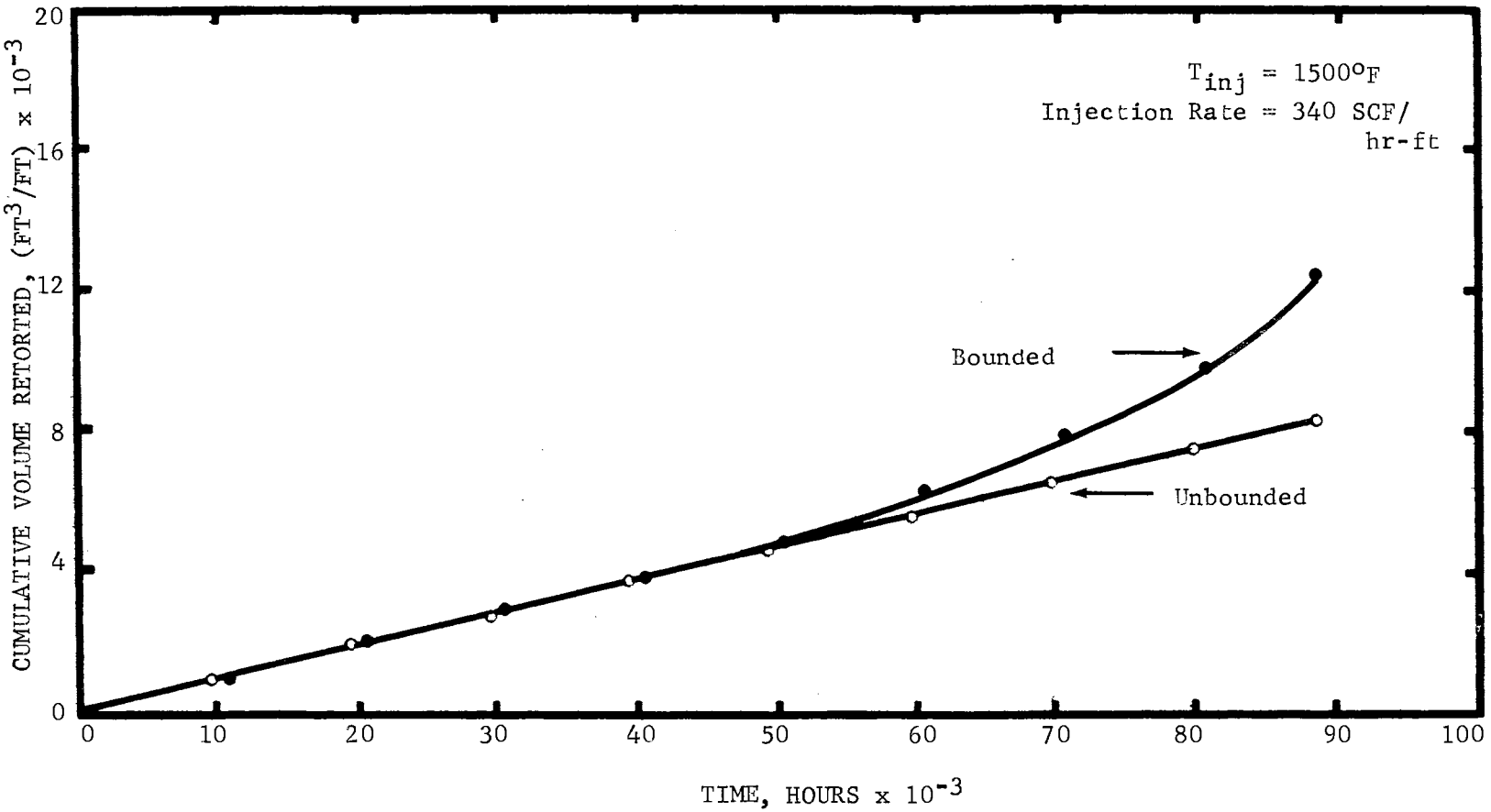


Figure 52. A Comparison of Retorting Rates with Bounded and Unbounded Systems

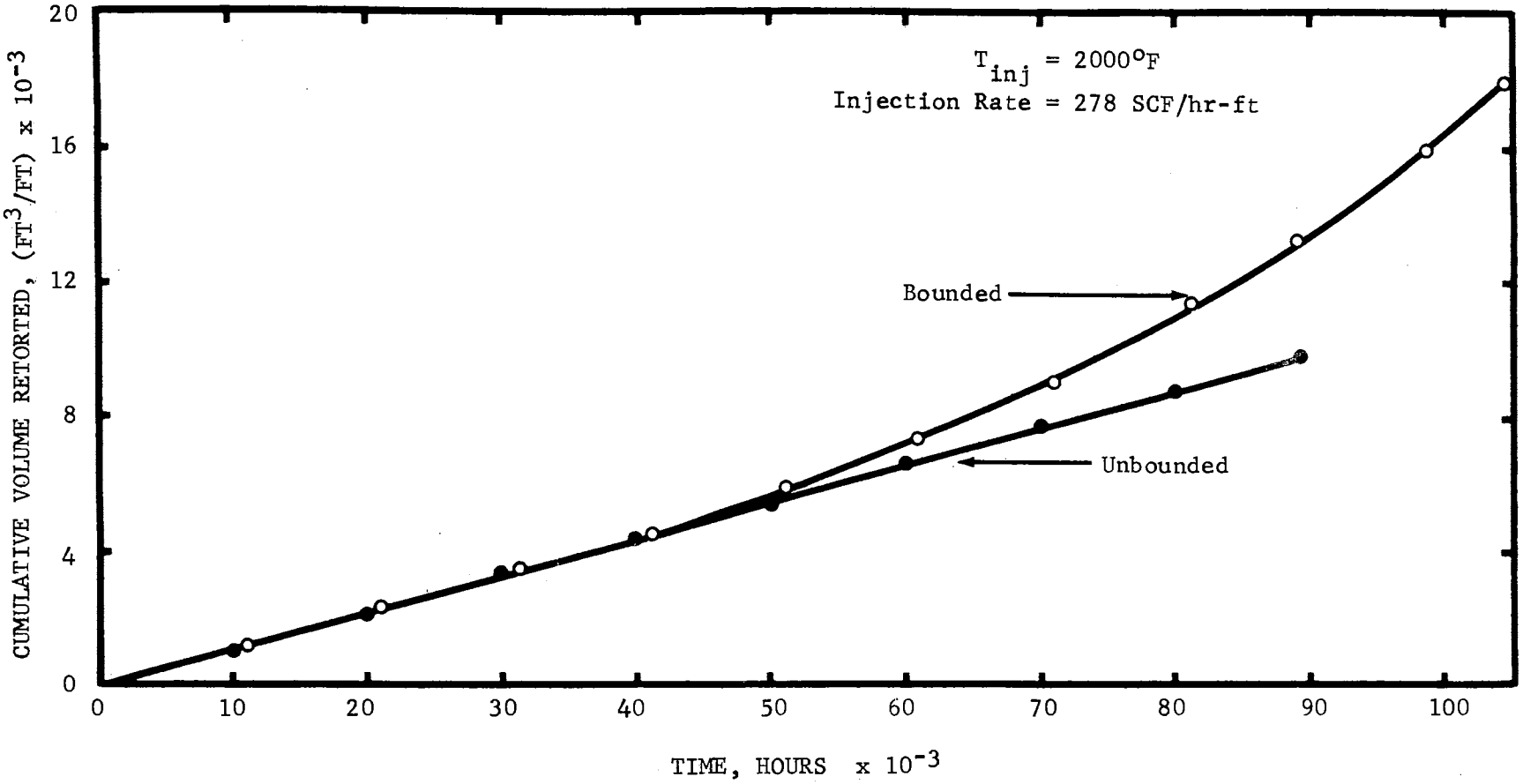


Figure 53. A Comparison of Retorting Rates with Bounded and Unbounded Systems

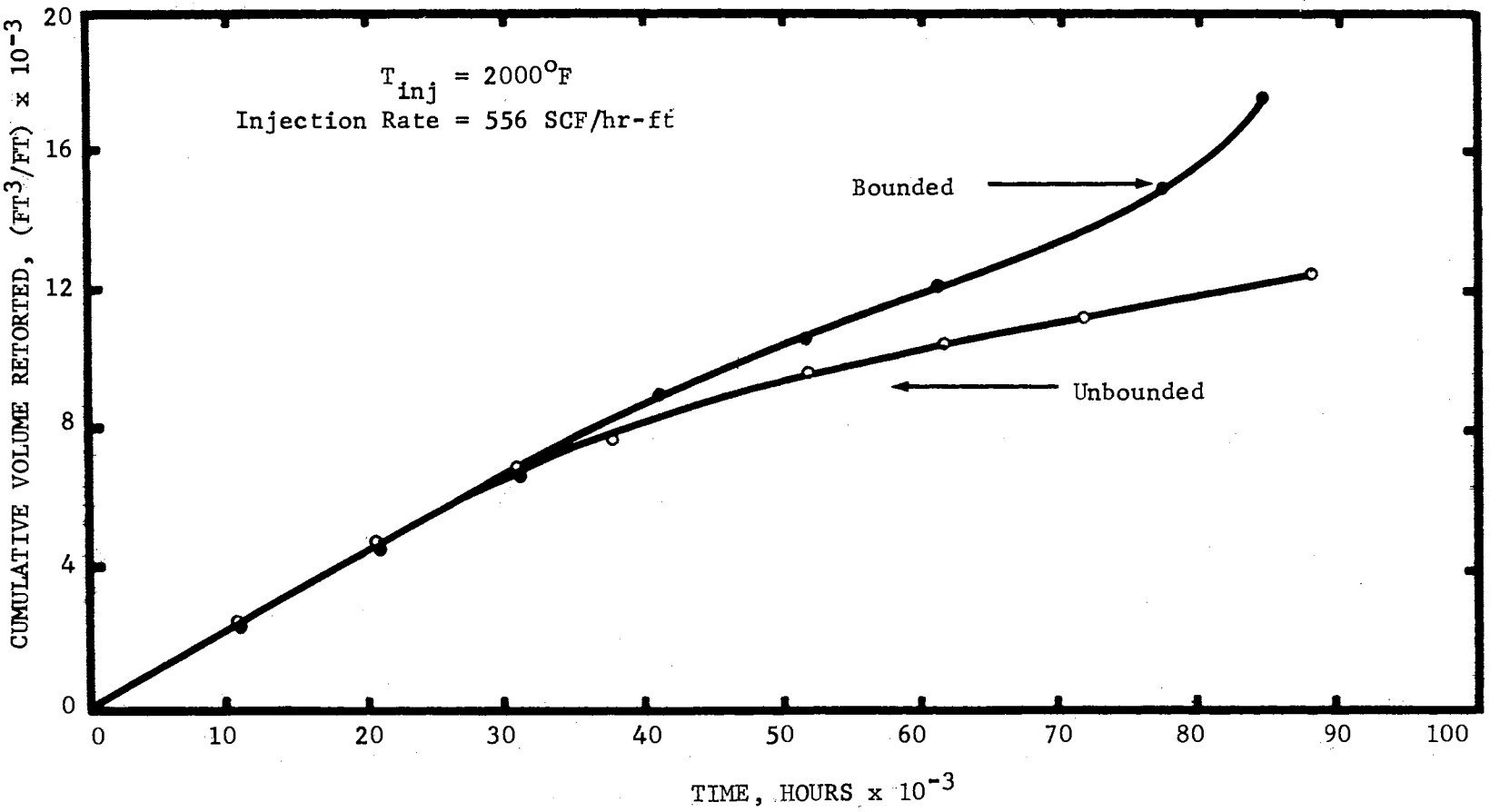


Figure 54. A Comparison of Retorting Rates with Bounded and Unbounded Systems

the producer reached the maximum allowable temperature of 900°F near the end of the run. We can conclude, therefore, at a rate of 278 SCF/hr-ft (1.33 MMSCF/D per well) with a 2000°F injection temperature, it would take approximately 12 years to retort a symmetric element from a pattern array which had wells spaced 500 feet apart along parallel lines spaced 100 feet apart.

Figure 55 compares the retorting rates of the three runs conducted on bounded systems. The most pertinent observation from this figure is a comparison of runs at 2000°F with injection rates of 278 and 556 SCF/hr-ft. In the latter case the run utilized the alternating injection-soak routine throughout much of the time. A comparison of retorted volumes and cumulative injected volumes at the end of these two runs is illustrated. At the end of 85,000 hours the high injection rate case, using the soak routine, retorted 17,500 cubic feet per foot requiring 34.8 MMSCF/ft of injected gas. The slower injection rate case took 104,000 hours to retort the same volume but used only 29.0 MMSCF/ft of injected gas to do the job. This comparison shows it takes 20 per cent longer at the slower rate but requires only 80 per cent as much injected gas. This shows that the slower rate would be superior to the faster rate that resulted in alternating injection-soak during the latter part of the run.

This comparison, along with the unbounded soak runs, indicates the alternating injection-soak routine would be inefficient and that it is preferable to design a project by a combination of geometry, project life and injection rate which would allow complete retorting of a system by the time the producing well temperature reached 900°F. The fact that the retorting rate is constant until the 700°F isotherm

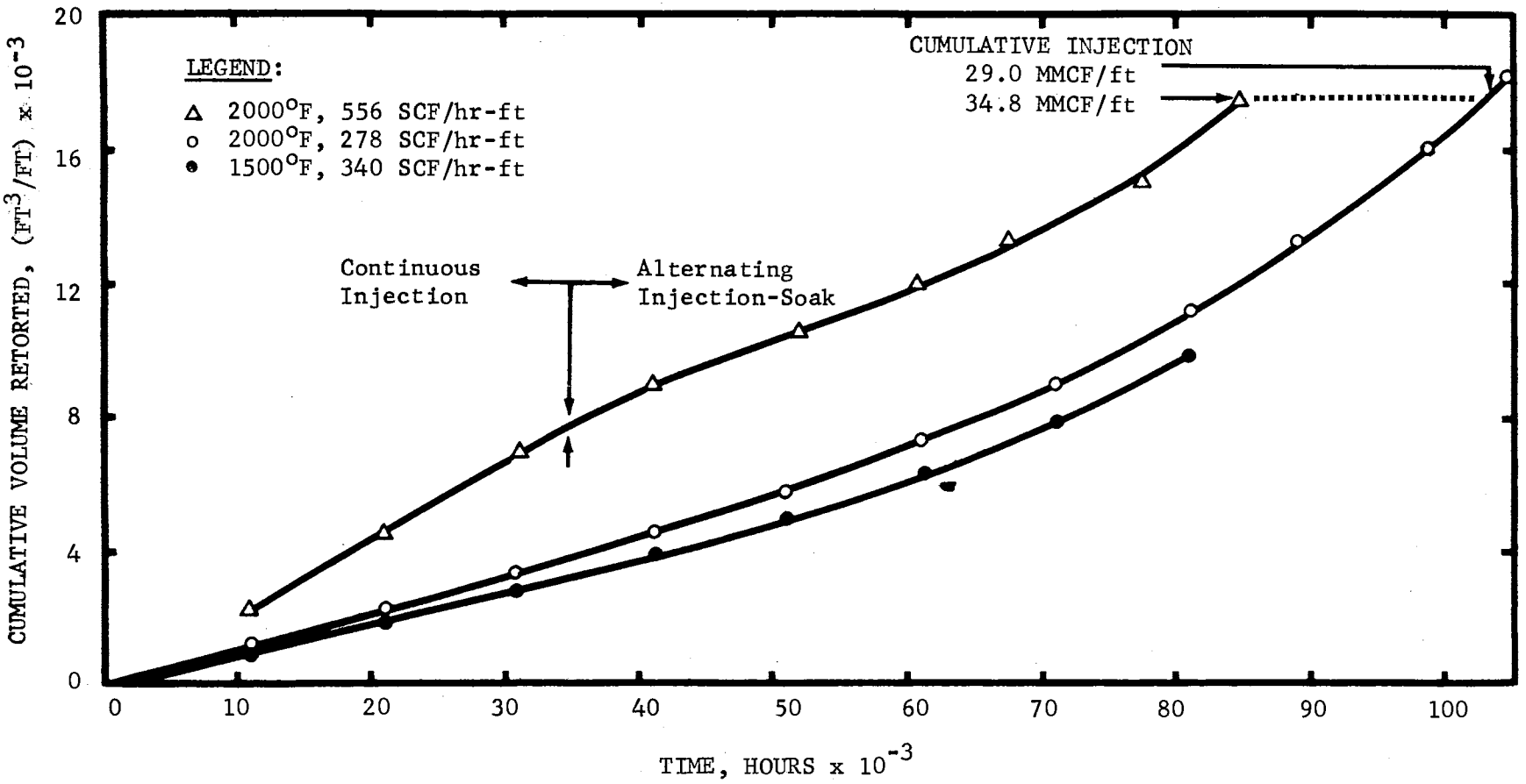


Figure 55. A Comparison of Retorting Rates of Runs Using Bounded Systems

reaches the producer also supports this conclusion. Retorting efficiency starts to drop off between 700°F and 900°F, but a project could be designed to finish when the producing well reaches a temperature somewhere between 700°F and 900°F.

Air-Oil Ratios

Theoretical air-oil ratios can be calculated from the results of all the runs on Model I and II. The air-oil ratio is an excellent yardstick for predicting process feasibility because the cost of air injection would be the principal expense in such a process. The volume of oil shale retorted at any time represents a certain quantity of oil. It has been assumed in this work that the average oil shale richness is 30 gallons per ton. If it is assumed that the oil is 100 per cent recoverable then an optimum air-oil ratio can be calculated by dividing the volume of air-fuel mixture injected by the oil production during a selected period of time. Figures 56, 57, 58 and 59 show air-oil ratios for the 12 runs.

Figure 56 shows that the producing air-oil ratio with an injection temperature of 1000°F would be constant at approximately 300,000 SCF/bbl for an unbounded system. A constant air-oil ratio would be expected prior to breakthrough of the 700°F isotherm since the rate of retorting is constant.

Figure 57 shows that up to the time of breakthrough at the producing well, the producing air-oil ratio with an injection temperature of 1500°F is 80,000 SCF/bbl for an unbounded system. At a high gas injection rate of 740 SCF/hr-ft using the alternating injection-soak routine, the air-oil ratio continued to increase with time and was

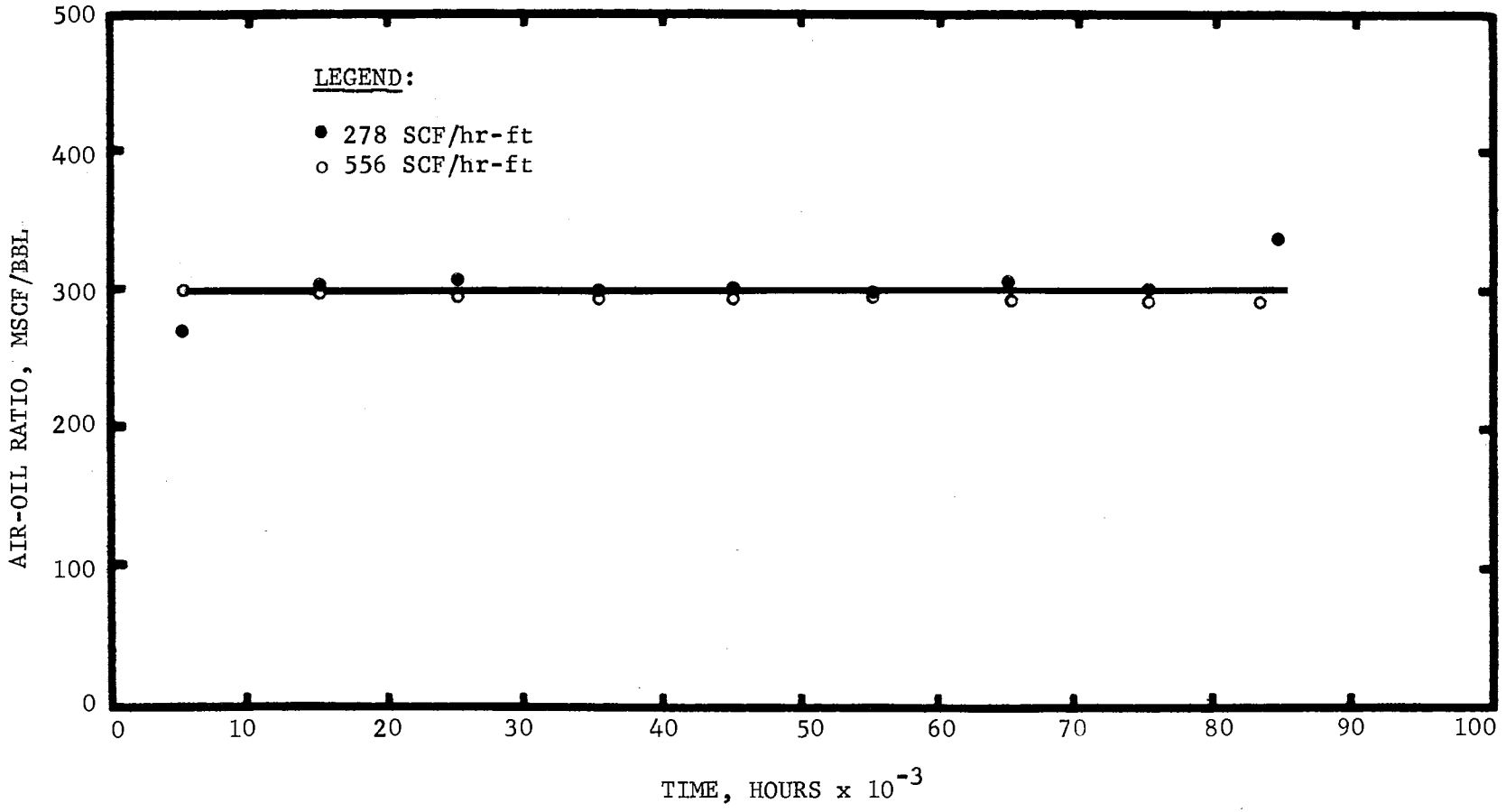


Figure 56. Air-Oil Ratio for Injection Temperature of 1000°F

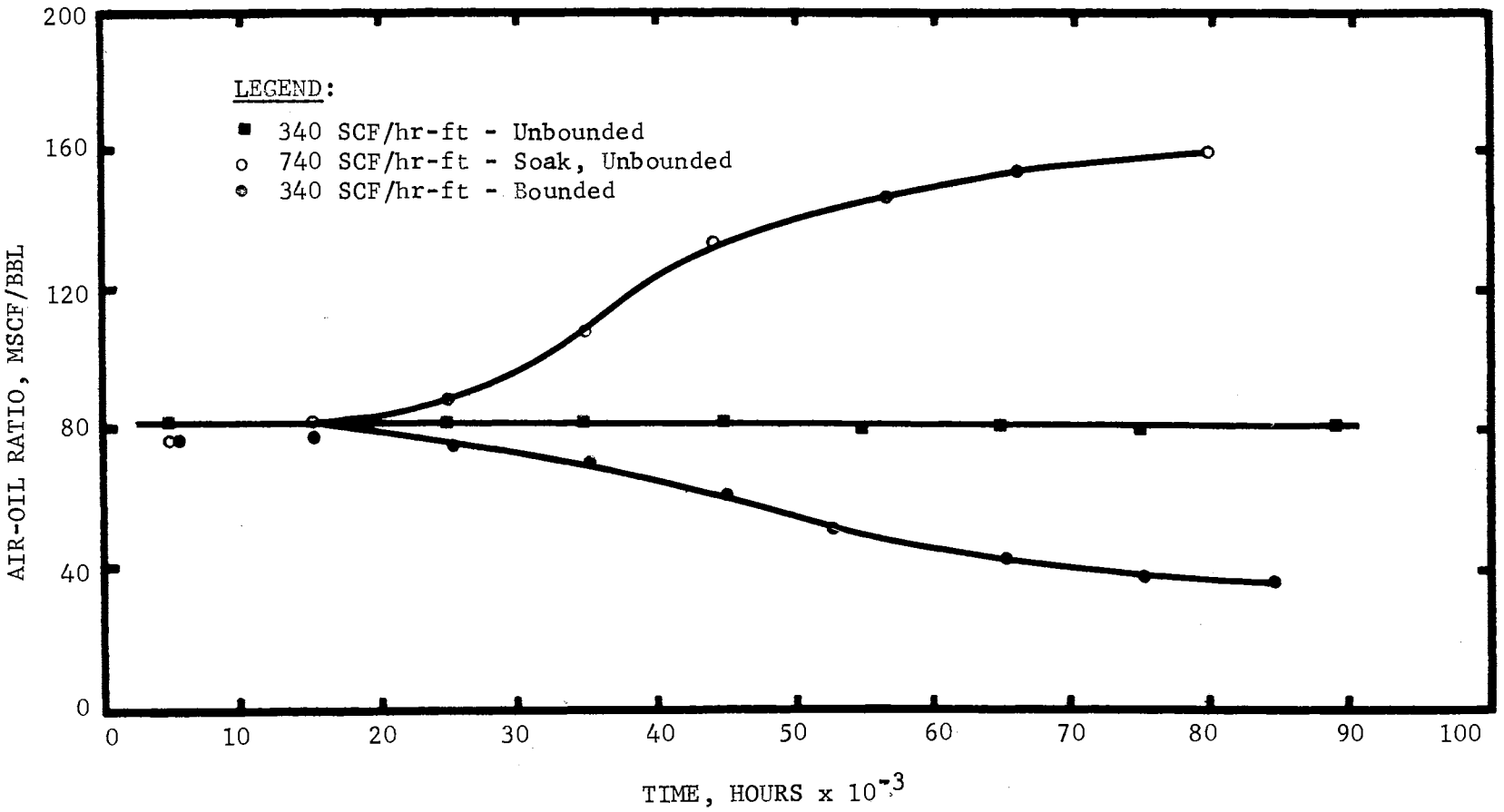


Figure 57 Air-Oil Ratios at Injection Temperature at 1500°F

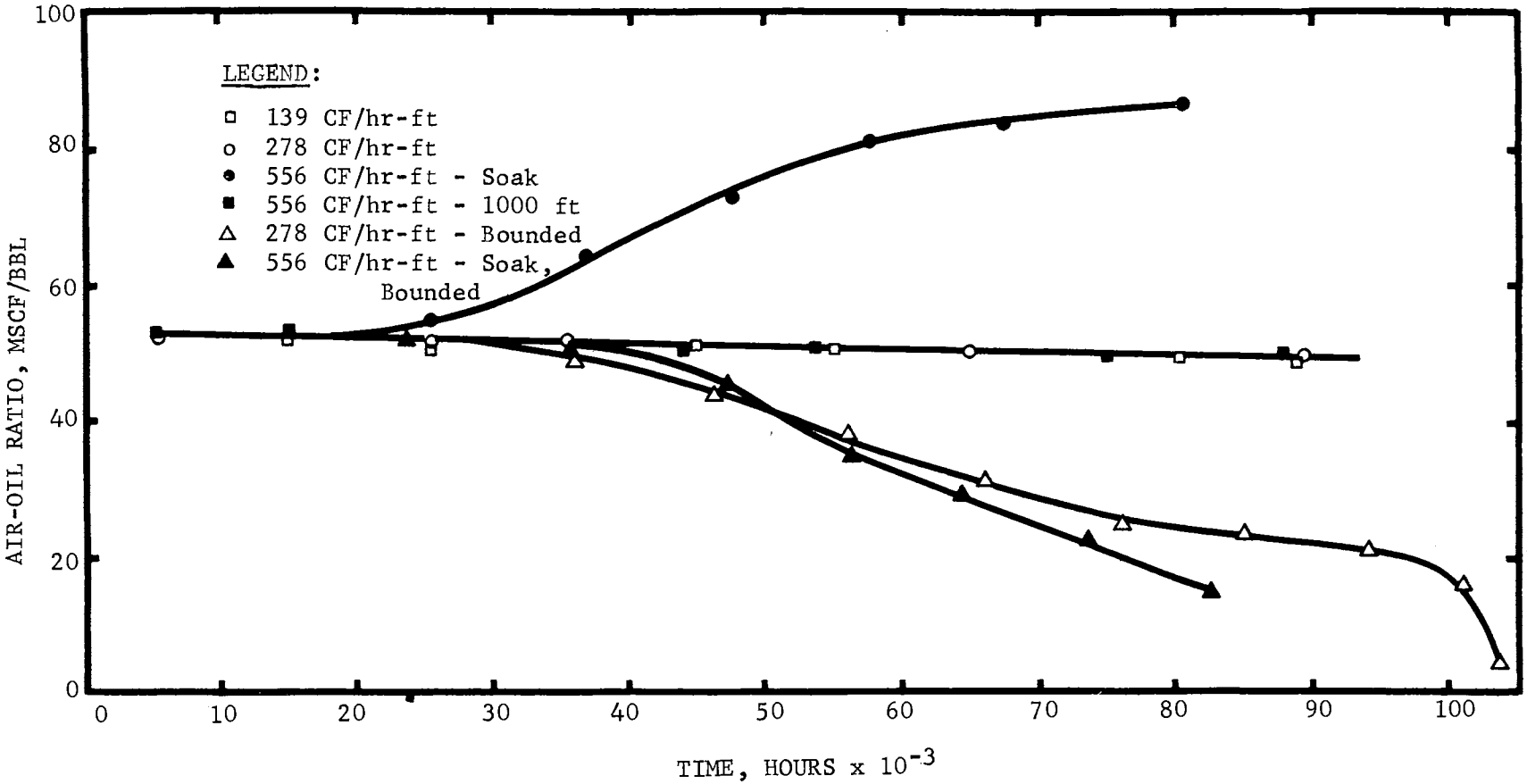


Figure 58. Air-Oil Ratios at Injection Temperature of 2000°F

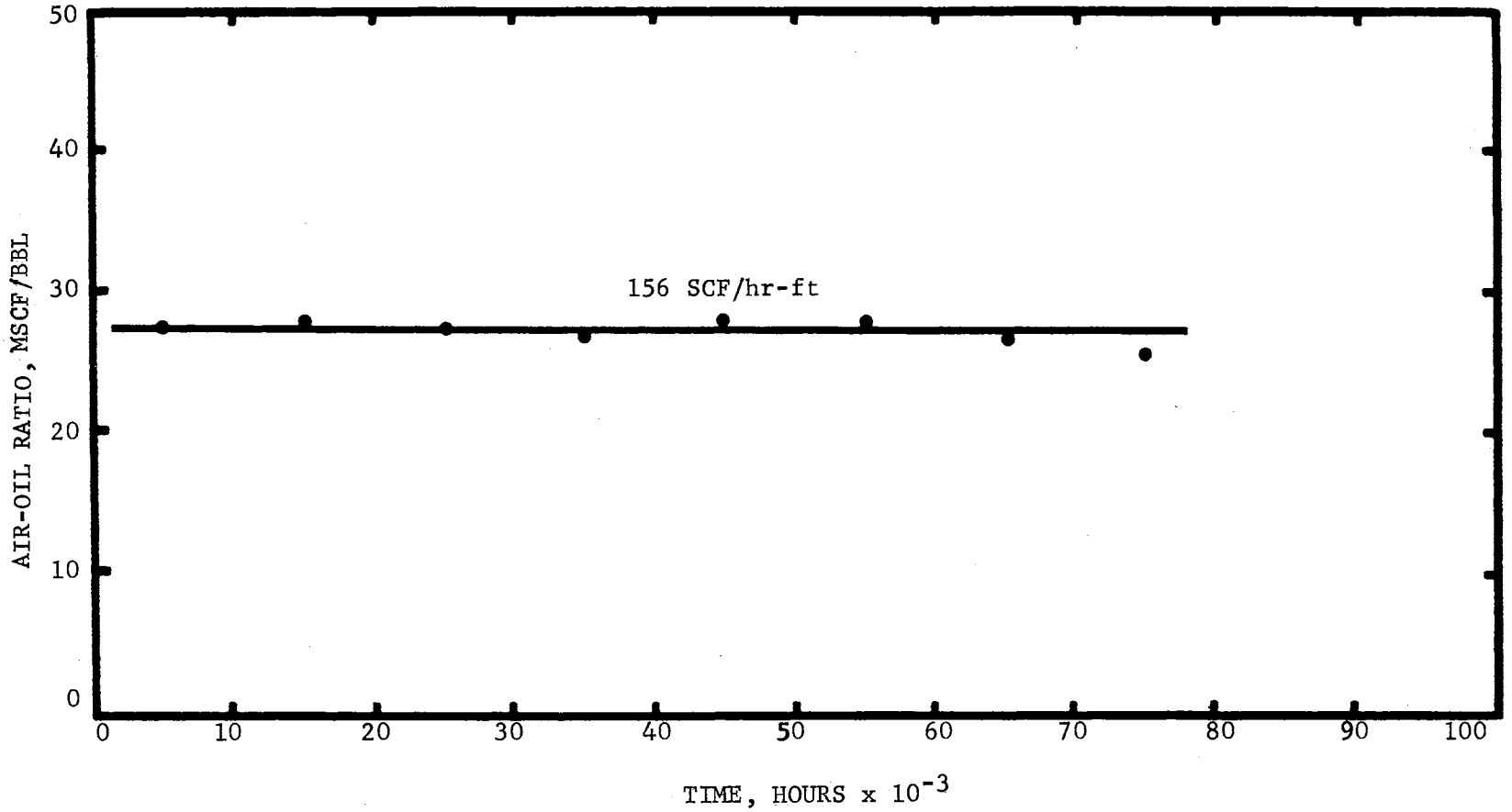


Figure 59. Air-Oil Ratio for Injection Temperature of 3000°F

160,000 SCF/bbl at 80,000 hours' producing history. With a rate of 340 SCF/hr-ft the air-oil ratio decreases for a bounded system from 80,000 to 38,000 SCF/bbl after 80,000 hours.

Figure 58 compares air-oil ratios for six runs using a gas injection temperature of 2000°F. The air-oil ratio for all runs prior to 700°F isotherms breakthrough is 52,000 SCF/bbl. The unbounded soak run shows that the air-oil ratio increases to 86,000 SCF/bbl during the remaining life of the run. The air-oil ratio of the bounded soak run decreased throughout the run to 15,000 SCF/bbl at 82,500 hours. The bounded run with an air injection rate of 278 SCF/hr-ft shows the air-oil ratio decreasing to 5000 SCF/bbl at 104,000 hours. At this time the producing well had reached 900°F but the temperatures of the rest of the unretorted area as illustrated in Figure 38 are greater than 650°F. This means that this area would be retorted with practically no additional air injection, thus making the producing air-oil ratio extremely low.

Figure 59 illustrates that a 3000°F injection temperature will yield a producing air-oil ratio of 27,500 SCF/bbl with an unbounded system.

It can be seen that the producing air-oil ratio is solely a function of the injection temperature prior to breakthrough of the 700°F isotherm at the producing well. The higher the operating temperature the lower the air-oil ratio will be. It is also obvious that the bounded systems will yield the lower average air-oil ratios. Run 11 was conducted 109,100 hours. At that time all temperatures were greater than 650°F, indicating the volume between fractures would be completely retorted. Dividing this volume into the cumulative air

injected gives an average theoretical air-oil ratio of 24,000 SCF/bbl over the 12-year project life. It is possible that air injection could be stopped earlier and the whole area still be retorted over a longer time. This would serve to lower the average air-oil ratio somewhat.

Heat Utilization

The fraction of the cumulative injected heat that is absorbed by oil shale is one measure of efficiency. The following table lists the twelve runs for these two models with their total times and fractional heats stored. These fractional values have been adjusted downward by the percentage heat balance error since the heat absorbed plus the heat produced was higher than the heat injected for each of the runs.

Run Number	Time, Hrs.	Fraction of Cumulative Injected Heat that is Absorbed by Oil Shale
1	89,100	0.83
2	86,100	0.59
3	89,100	0.73
4	88,500	0.47
5	86,100	0.94
6	89,100	0.84
7	85,800	0.67
8	88,100	0.92
9	83,200	0.85
10	89,100	0.72
11	109,100	0.78
12	84,800	0.65

Comparison of these values shows that heat utilization improves at reduced injection rates for the same system length. Also, the higher the injection temperature, the better the heat utilizations. The runs employing the soak routine during part of the time had lower heat

utilizations than other runs. Although Runs 10, 11 and 12 (Model II) indicate lower heat utilizations, it must be remembered that the heat absorbed by bounded systems is used more efficiently as far as retorting is concerned.

Results from Model IV

Results from Model IV will be discussed before those of Model III since Model IV was designed to evaluate the assumption of vertical adiabatic boundaries used in the preceding models. A qualitative estimate of the error involved when two-dimensional horizontal conduction heat transfer is assumed was made with THERMAB5 using results from Run 6 as source data. Vertical cross-section profiles were examined at 0, 100, 200 and 400 feet along the fracture using temperature histories at these positions from Run 6. Four separate runs with THERMAB5 were necessary to determine these profiles. The measured vertical conductivity (traverse to the bedding planes) of 30 gal/ton shale was found by Somerton (13) to be 0.71 of the horizontal conductivity (parallel with bedding planes) at room temperature. This ratio of vertical or horizontal conductivity was used in the variable conductivity subroutine in THERMAB5.

Final cross-sectional positions of the 700°F isotherms as determined with THERMAB5 at these four positions are shown in Figures 60, 61, 62 and 63. The areas encompassed by the 700°F isotherms are compared to the rectangular areas shown. These rectangular areas represent the cross-sectional retorted areas when vertical heat flow is assumed to be zero. At 0 feet down the fracture the cross-sectional profile is 5 per cent larger than the rectangular area. At 100, 200

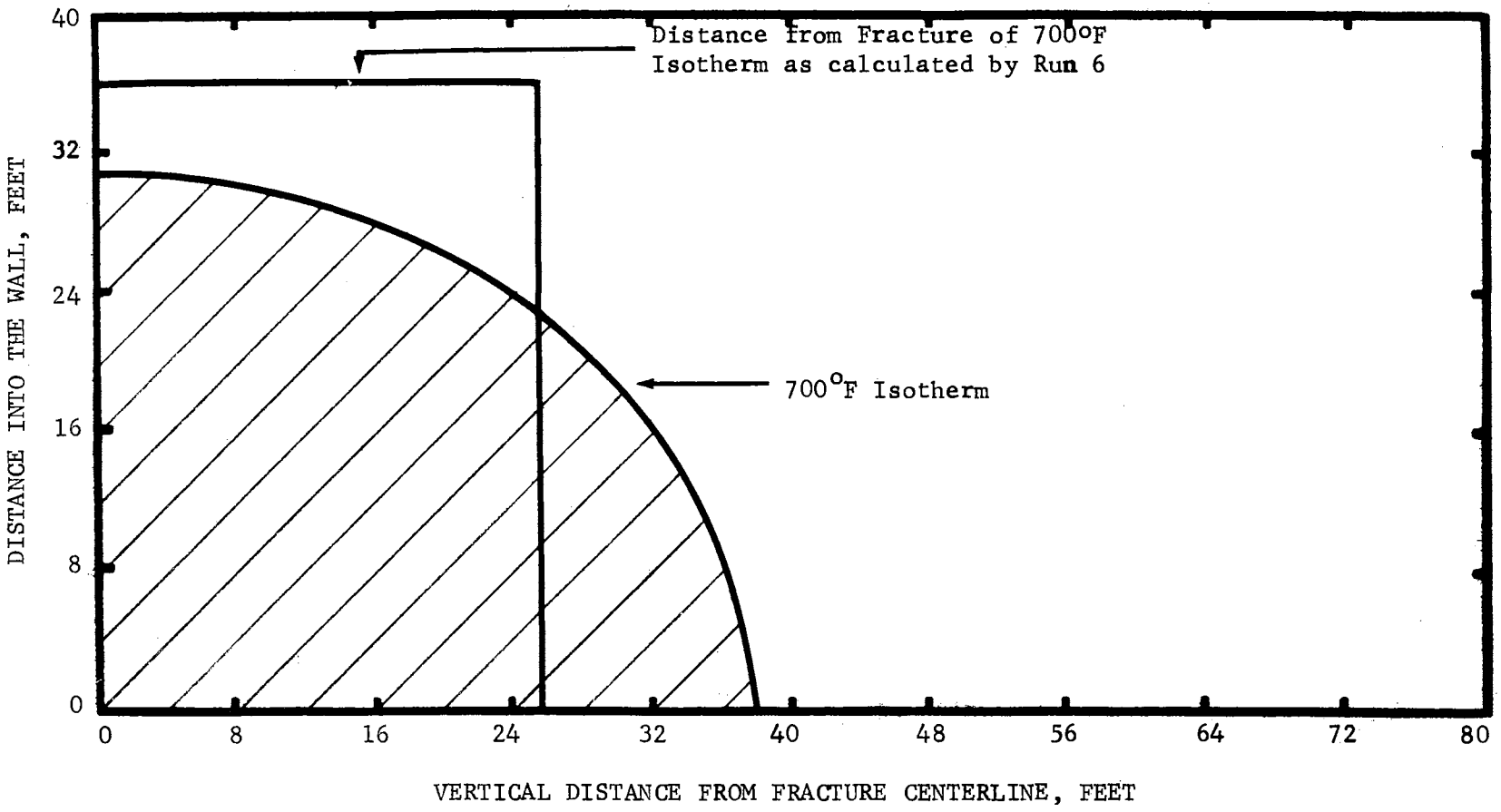


Figure 60. Vertical Profile of 700°F Isotherm at 0 Feet Down Fracture After 89,100 Hours

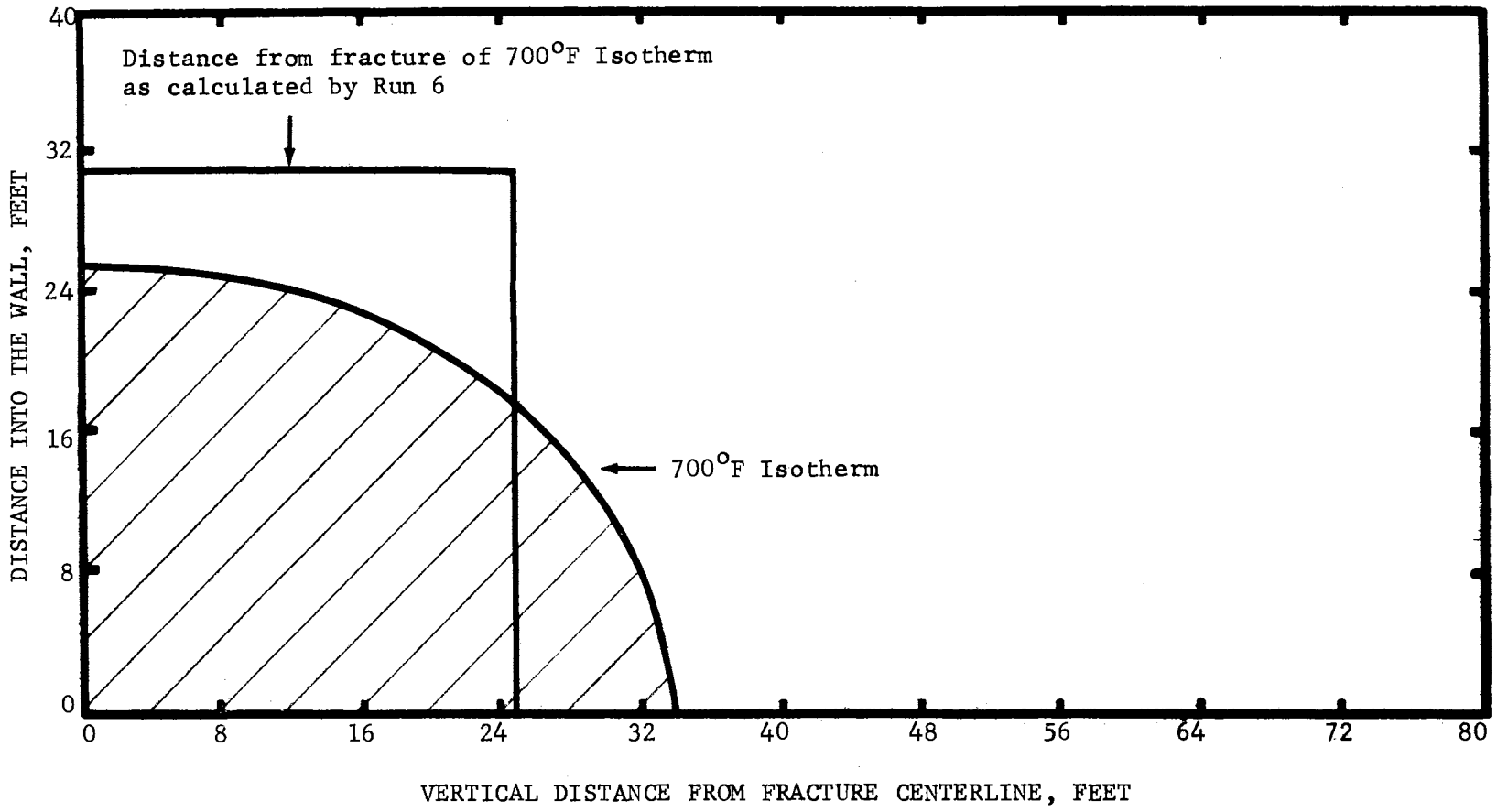


Figure 61. Vertical Profile of 700°F Isotherm at 100 Feet Down Fracture After 89,100 Hours

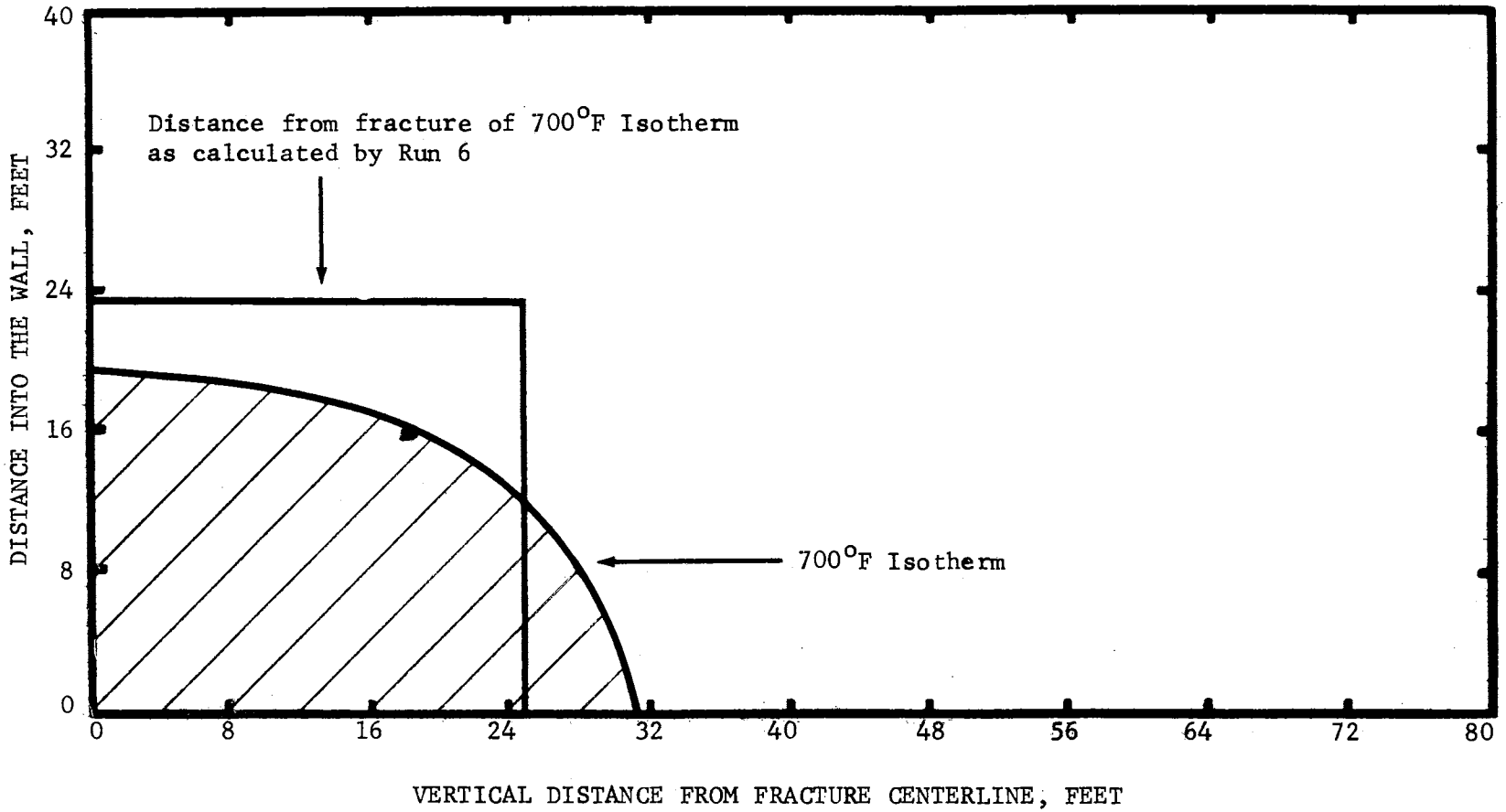


Figure 62. Vertical Profile of 700°F Isotherm at 200 Feet Down Fracture After 89,100 Hours

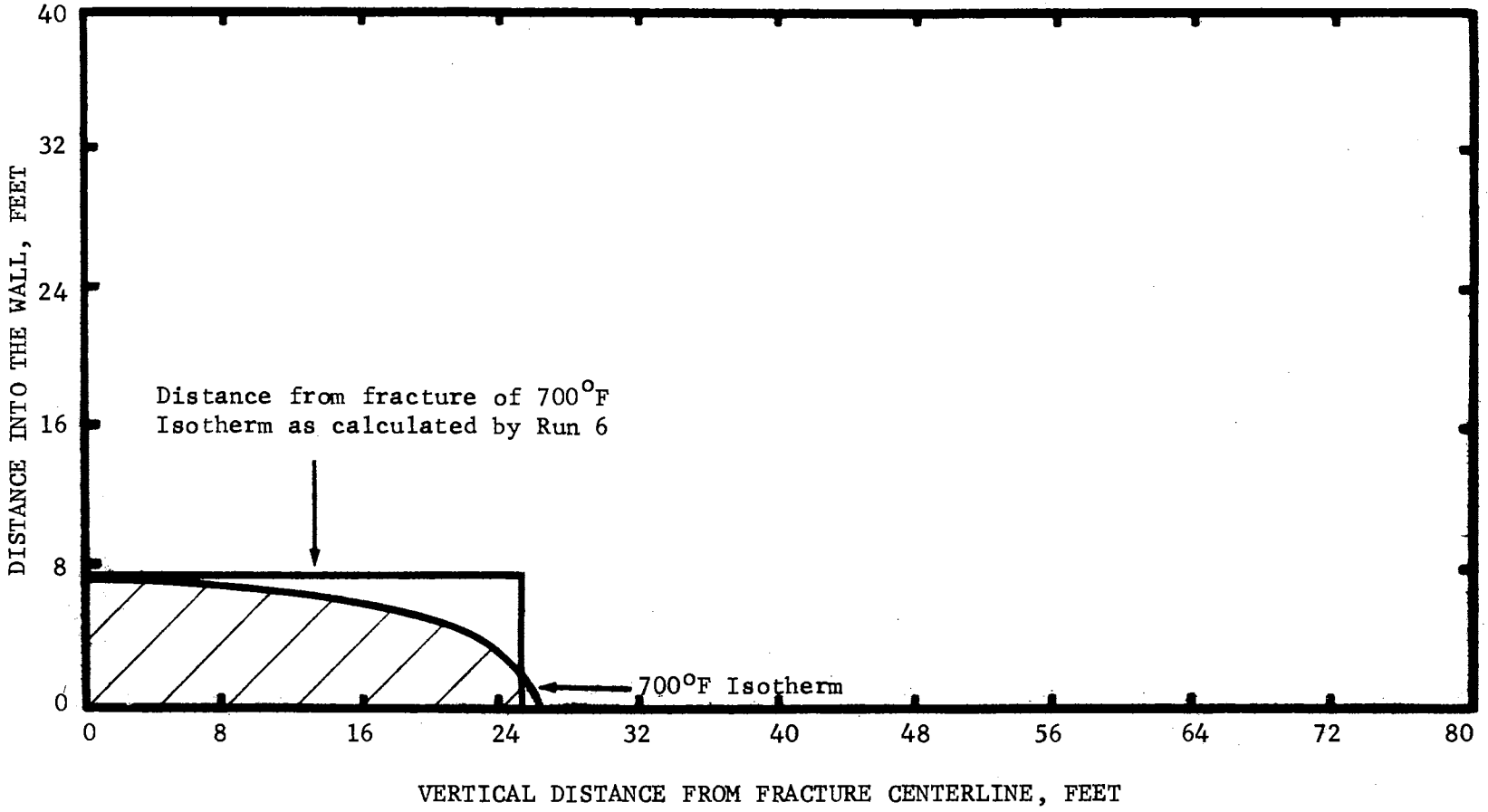


Figure 63. Vertical Profile of 700°F Isotherm at 400 Feet Down Fracture After 89,100 Hours

and 400 feet down the fracture the cross-sectional profiles are 10, 17 and 21 per cent smaller than the rectangular areas. Applying these four errors to the final retorted volume of Run 6 in a weighted average manner, it was determined that the two-dimensional retorted volume determined by THERMAB3 in Run 6 was approximately 12 per cent higher than would be anticipated in an actual three-dimensional case.

It must be remembered that this result is qualitative, since source temperatures for these profiles were taken from the two-dimensional solution of which we are determining error. At large times these two-dimensional source temperatures should be near the average three-dimensional ones. In three dimensions the entire fracture height would not be at the same temperature as was assumed here.

Even though this is a qualitative result, it points out two facts. First, retorted volumes calculated with the two-dimensional unbounded models are high on the order of 10 per cent or more. Since heat balance errors are all on the plus side, errors in retorted volumes can no doubt be considered a few per cent higher than this. Second, results from Model II (bounded systems) would also be optimistic, since interference between heat sources would occur later than predicted by the two-dimensional model.

Results from Model III

Model III was designed to evaluate the effect of partial convective flow through the retorted zone. A combined conduction-convection equation was used with this model instead of separate equations, thus numerical procedures used to solve this model are different from the

numerical procedures used to solve Model I. The relative accuracy of these two models was compared in Runs 5 and 13. Boundary and initial conditions for these two runs were identical. Run 5 was made using THERMAB3, the program written to solve Model I; and Run 13 was made using THERMAB2, the program written to solve Model III. Although cumulative heat error was several per cent higher using Model III than it was using Model I, the resulting final position of the 700°F isotherm coincided with that obtained in Run 5 shown in Figure 26. This check indicates that this method of solution was valid and results from Model III could be compared directly with Model I.

Run 14 was made using the same parameters as Run 13, but it was assumed that behind the leading edge of the 700°F isotherm 10 per cent of the injected gas traveled through the retorted zone, evenly dispersed, while 90 per cent traveled through the fracture. Ahead of the leading edge of this isotherm, all of the gas flow was in the fracture. Figure 64 shows the position of the 700°F isotherm versus time for this run. Figure 65 shows final temperature distribution for this run. A comparison of this figure with Figure 27 shows a considerable shift of the isotherms to the right. This suggests an improvement of heat utilization. Figure 66 compares the final position of the 700°F with and without 10 per cent convection in the retorted matrix. As can be seen, the retorted volume of oil shale is larger assuming limited gas flow in the region. This leads to a reduced air-oil ratio as shown in Figure 67.

Injected gas flow in the matrix is expected to be negligible. However, any convection in the retorted zone that may occur will serve

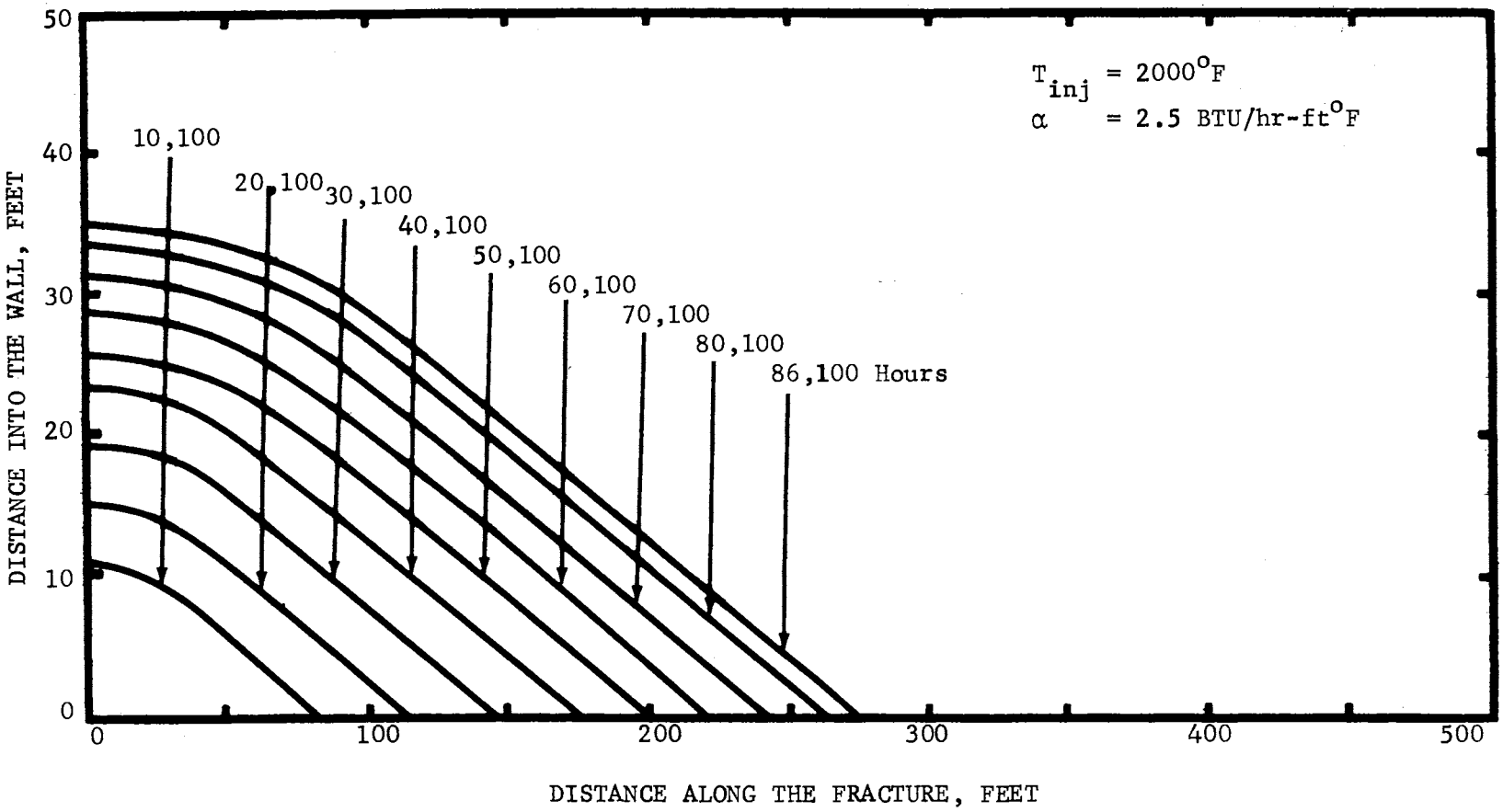


Figure 64. Position of 700°F Isotherm Versus Time -- Run 14

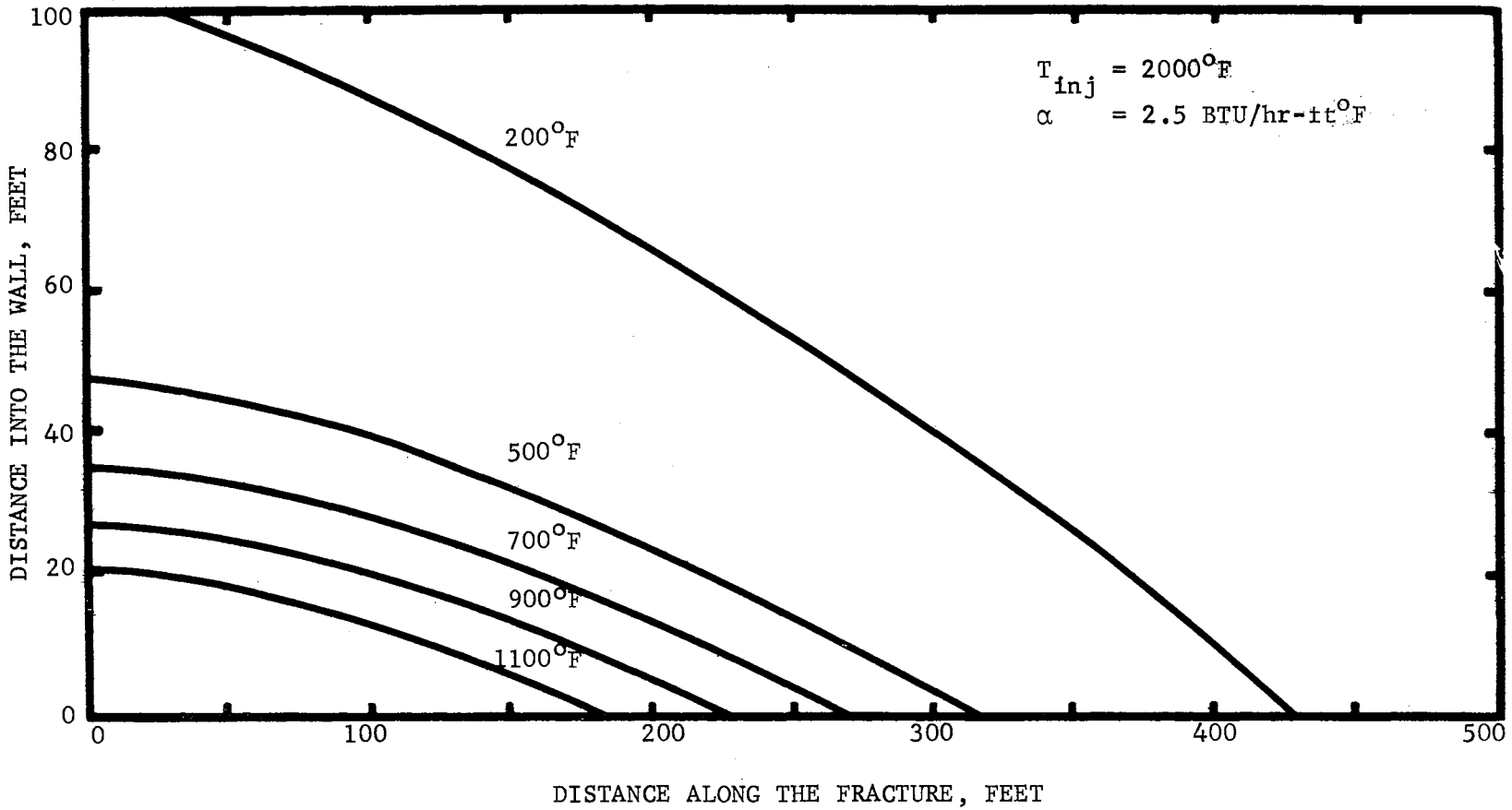


FIGURE 65. Temperature Distribution After 86,100 Hours -- Run 14

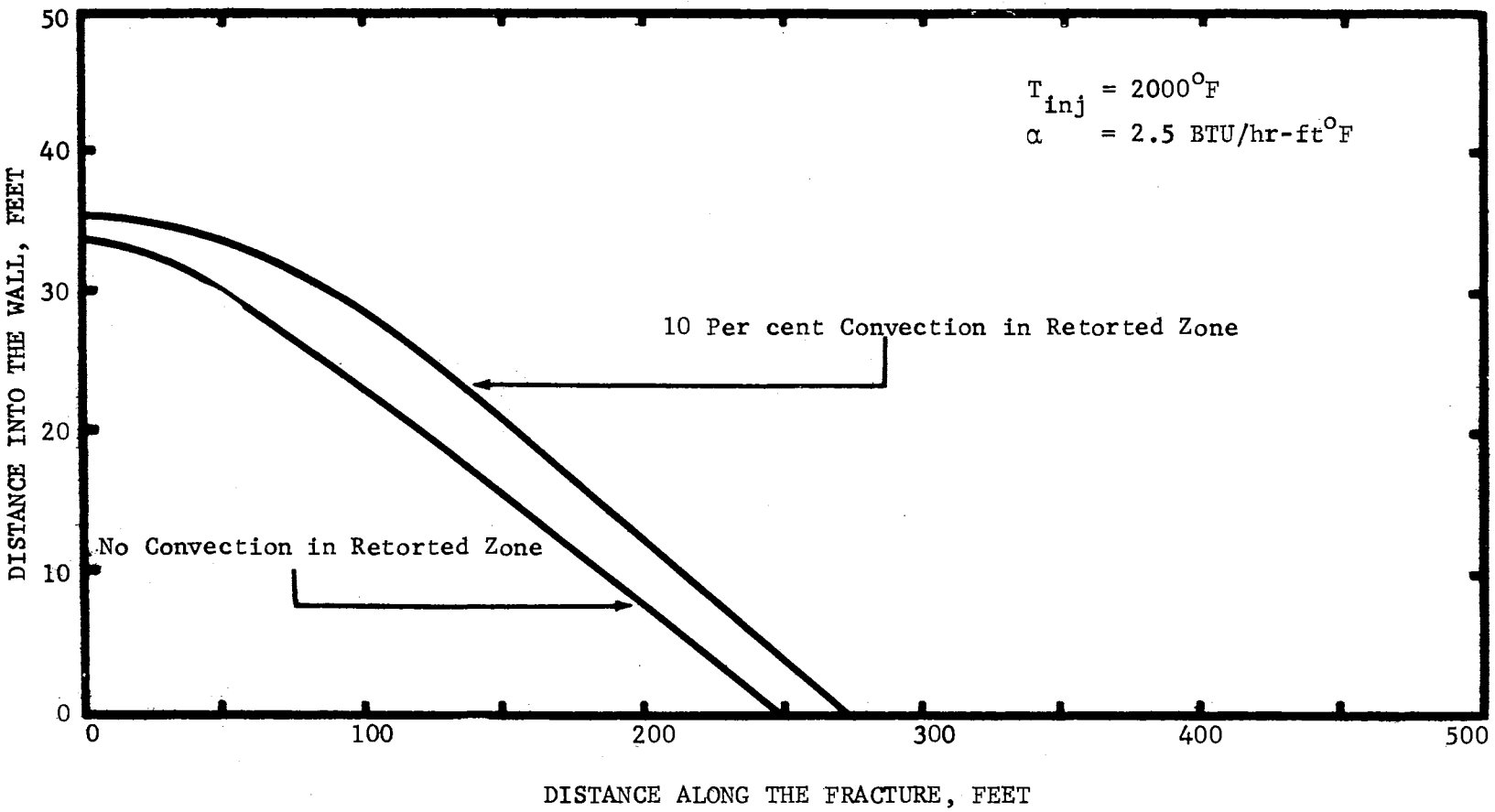


Figure 66. Positions of 700°F Isotherms After 86,100 Hours -- Runs 13 and 14

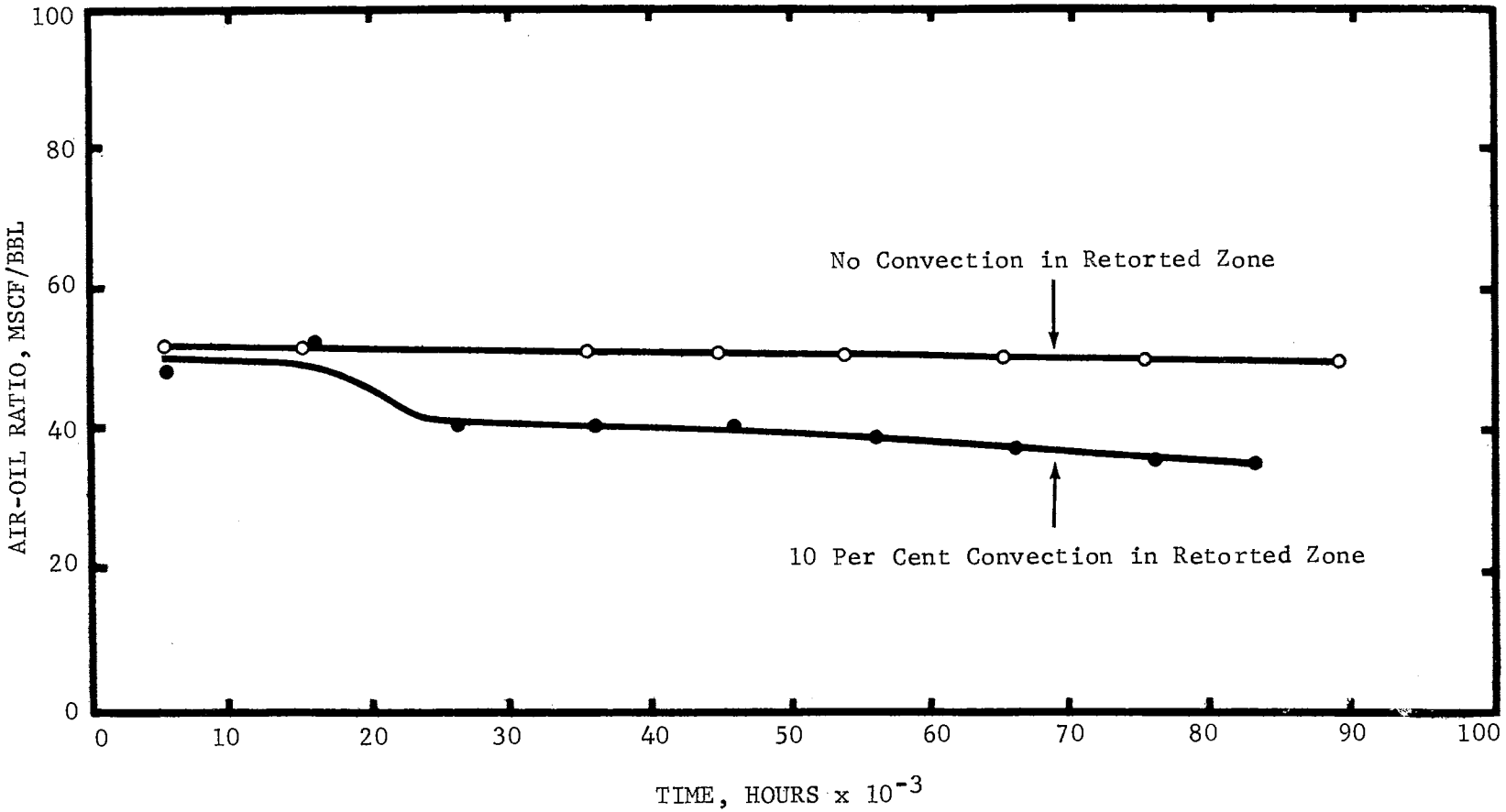


Figure 67. Air-Oil Ratios at Injection Temperature of 2000°F -- Runs 13 and 14

to increase the rate of retorting and improve the process over that predicted by Models I and II.

Significance of Results

General Remarks

Results of this study give a guide as to the feasibility of retorting oil shale by hot gas injection through vertical fractures from a heat transfer standpoint. The resulting theoretical air-oil ratios, which are based on retorted volumes, are the most important yardsticks as to economic feasibility.

In this investigation the values of oil shale thermal capacity, thermal conductivity and endothermic heat loss as functions of temperature were approximated. Also oil shale is a heterogeneous matrix where values of these heat transfer parameters cannot be accurately predicted from point to point. These facts would lead to some difference between the calculated and the real heat distribution of a shale matrix, even if the calculated solution were exact. In this work it has been shown that the numerically calculated retorted volumes in two dimensions are probably 10 per cent or more above a numerical three-dimensional solution. This accuracy is considered sufficient for the type of heterogeneous system being studied. These results should be taken to be semiquantitative. They serve to indicate the range and direction of expected results.

Feasible Air-Oil Ratios

Economic air-oil ratios depend on a number of factors, the most

significant being the injection pressure. Results from this work indicate that retorting efficiency drops off when the 700°F isotherm reaches the producing well. This would indicate that a project should be designed so that the 700°F reaches the producer near the end of the project life. This being the case, it would be expected that high injection pressures (2000 - 2500 psi) would be required during the entire life of a project in order to keep the fracture opened. Pressures would be on the order of 2000 psi which would lead to compression costs on the order of 6 cents per MSCF of air compressed from atmospheric conditions.

High injection pressures are not as discouraging as might be supposed. This is because relatively high-pressured produced gas from the process could be reinjected to control the injection temperature. This reinjected produced gas could represent up to 70 per cent of the total gas injected. The cost of compressing this produced gas would be only a fraction of that to compress an equivalent amount of gas from atmospheric pressure. The reinjected produced gas would probably contain enough fuel so that make-up gas would not be needed after the process had been operating awhile.

Other factors such as project life, well costs, hydrocarbon product quality, etc., would have a bearing on total costs and thus determine what maximum air-oil ratio might be feasible. There have been indications that the quality of oil obtained from an in situ oil shale retorting process would be considerably improved over the quality from surface retorting. This means the oil would be more valuable per barrel and visbreaking would be unnecessary. A higher air-oil ratio could be tolerated if the oil quality should be improved.

With this as a background, it is clear low injection temperatures, i.e., 1000°F, are ruled out. An injection temperature of 2000°F should yield air-oil ratios of about 25,000 SCF/bbl and therefore would be in the feasible range. This assumes liquid recovery equivalent to USBM assay and that a field-wide pattern can be assumed to have the efficiency of the bounded system. This latter assumption would be close for a large development.

The type and quality of the recoverable hydrocarbon is very important. If a large portion of the hydrocarbon product were cracked to gas, the resulting air-oil ratio would increase over that indicated here, and therefore the process economics would become less attractive. The cracked gas would be so diluted with gas volumes of the 25,000 SCF/bbl magnitude that the resulting heating value of the exhaust gas would be too low to be commercially attractive.

If it were possible to operate with a 3000°F injection temperature, this retorting process would definitely look advantageous from a heat transfer standpoint. Although the unbounded 3000°F run showed an air-oil ratio of 27,000 SCF/bbl, a bounded case would give an average gas-oil ratio of about 12,000 SCF/bbl. This is based on the same ratio of retorted volumes as obtained between bounded and unbounded runs at 2000°F. A 12,000 SCF/bbl process would be very attractive. If the hydrocarbon product would be entirely cracked to gas the resulting exhaust gas heating value after scrubbing the carbon dioxide would be approximately 600 BTU/SCF. This would be a marketable product. In this limiting case the retorting process would still be attractive with a 3000°F injection temperature.

Project Design

This study has shown that the alternating injection-soak procedure is not efficient. Because of this and because retorting efficiency drops off after the producing well reaches 700°F, it would be desirable to design a project so that it would be terminated at this time. The feasible way to design an in situ project of this type would be to start with the project life desired. This optimum life is interrelated with many factors, and it would be necessary to analyze several cases to determine it. After deciding on a project length, the distance between injection and producing wells to be used must be selected. The longer the distance the more economical the process because of the need of fewer wells, assuming communication between wells was obtainable with equal ease. Distances between wells would probably be limited because establishing communication between wells would no doubt become more difficult at greater distances. This distance would have to be determined experimentally. Once project life and path length are established, the injection rate can then be set so that the 700°F isotherm will arrive at the producer in the allotted time.

Spacing between lines of wells would be a function of project life and injection temperature and would be easily determined. Injection temperature should be as high as feasible and would have to be determined experimentally in a pilot test.

Experimental Pilot

This heat transfer study has served to indicate whether adequate

heat penetration and utilization could be achieved under various assumed conditions. Results have shown that with high injection temperatures the theoretical air-oil ratios are within range of being economic. Experimental pilot testing would be necessary to develop methods of operation, to verify air-oil ratios and to determine operating costs. The first step in this testing would be to develop techniques to reliably communicate between wells at long distances apart with the assurance of nearly complete gas recovery. The maximum feasible continuous gas injection temperature and the resulting injection pressures are very important parameters to be determined initially with pilot operation. Equipment development, operating technique and resulting costs are important information affecting process economics that are determined from prolonged pilot operation. The resulting hydrocarbon product, whether it be predominately high or low quality oil or gas, must be determined. Finally, the actual producing air-oil ratio possible for the process would be determined from a pilot test.

CHAPTER VI

SUMMARY AND CONCLUSIONS

The objective of this work was to make a preliminary feasibility study of in situ retorting oil shale by hot gas injection through wells interconnected by single vertical fractures. A theoretical heat transfer study was conducted to achieve this objective. A knowledge of the heat distribution and content under various assumed conditions will indicate whether experimental pilot testing of this approach to in situ retorting is justified.

This analysis involved the simultaneous numerical solution of a nonlinear second order partial differential equation, which describes two-dimensional conduction heat transfer in oil shale, and a nonlinear first-order partial differential equation which describes convection heat transfer in the fractures. Three nonlinear temperature-dependent coefficients that were taken into account in this work are the thermal conductivity, thermal capacity and retorting endothermic heat losses of oil shale. Vertical fractures were considered to be of finite height. Vertical conduction heat transfer was not considered; however an estimate of the error resulting from this limitation was made.

The effects of injected gas temperature, injection rate, system geometry, cyclic injection and time upon retorting efficiency were investigated.

The following conclusions have been made as a result of this study:

1. Results from this heat transfer study show that the rate of retorting oil shale is a function of the injection temperature; however, as the injection temperature is raised the retorting rate increases to a lesser extent.

2. The air-oil ratio is an inverse function of the injection temperature at a constant rate of heat injection and it is a stronger inverse function of the injection temperature at a constant volume of air injection.

3. The rate of retorting is constant with a constant injection temperature and rate up until the time when the 700°F isotherm "breaks through" at the producing well. At that time the retorting efficiency decreases.

4. The retorting rate with a constant injection temperature is a direct function of the heat injection rate until breakthrough of the 700°F isotherm at the producer.

5. High injection rates with the use of an alternating injection-soak routine when the producing wells reach 900°F are less efficient than lower continuous injection rates.

6. Retorting efficiencies are improved for bounded systems compared to analogous unbounded systems and likewise air-oil ratios are lower.

7. Retorting efficiency is improved if partial convection is assumed to occur in the retorted zone of the oil shale.

8. It was found that with an injection temperature of 2000°F and an injection rate of 1.33 MMSCF/D per well it would take approximately

12 years to retort a shale section developed with a pattern array of wells spaced 500 feet from producer to injector and 100 feet between rows of wells.

9. The error in the volume of rock retorted caused by the assumption of adiabatic boundaries in the vertical direction was estimated to be approximately 10 per cent high. However, results obtained from these two-dimensional models are adequate in light of the heterogeneous system being studied.

10. This work shows that it would be preferential to design a field project so that the 700°F isotherm arrives at the producing wells at the end of a project's life. The injection rate can be set to accomplish this if the project life and injection to producer well spacing is determined beforehand.

11. Injection temperatures on the order of 2000°F and higher give theoretical air-oil ratios in the economic range.

This heat transfer study shows that adequate heat penetration and utilization are possible. Results have shown that with the use of high injection temperatures the theoretical air-oil ratios are within the economic range. Experimental pilot testing would be necessary to develop methods of operation, to verify air-oil ratios, and to determine operating costs.

A SELECTED BIBLIOGRAPHY

(1) Jaffe', Felix C. "Oil Shale -- Part 1, Nomenclature, Uses, Reserves and Production." Mineral Industries Bulletin, Colorado School of Mines, Vol. 5, No. 2 (March 1962).

(2) Jaffe', Felix C. "Oil Shale -- Part II, Geology and Mineralogy of the Oil Shales of the Green River Formation, Colorado, Utah and Wyoming." Mineral Industries Bulletin, Colorado School of Mines, Vol. 5, No. 3 (May 1962).

(3) Williamson, D. R. "Oil Shales -- Part III, The Natures and Origins of Kerogens." Mineral Industries Bulletin, Colorado School of Mines, Vol. 7, No. 5 (Sept. 1964).

(4) Williamson, D. R. "Oil Shales -- Part IV, Retorting and Other Beneficiation of Oil Shales." Mineral Industries Bulletin, Colorado School of Mines, Vol. 7, No. 6 (Nov. 1964).

(5) Williamson, D. R. "Oil Shales -- Part V, Oil Shale Retorts." Mineral Industries Bulletin, Colorado School of Mines, Vol. 8, No. 2 (March 1965).

(6) "First Symposium on Oil Shale." Quarterly of the Colorado School of Mines, Vol. 59, No. 3 (July 1964).

(7) "Second Symposium on Oil Shale." Quarterly of the Colorado School of Mines, Vol. 60, No. 3 (July 1965).

(8) Hubbard, Arnold B. and Robinson, W. E. "A Thermal Decomposition Study of Colorado Oil Shale." Bureau of Mines Report of Investigations 4744 (1950).

(9) Sohns, H. W., Jukkola, E. E., and Murphy, W. I. R. "Development and Operation of an Experimental Entrained-Solids, Oil-Shale Retort." Bureau of Mines Report of Investigations 5522 (1959).

(10) Sohns, H. W., Mitchell, L. R., Cox, R. J., Barnet, W. I., and Murphy, W. I. R. "Heat Requirements for Retorting Oil Shale." Industrial and Engineering Chemistry, Vol. 43, No. 1 (1950) 33.

(11) Gavin, Martin J. and Sharp, Leslie H. "Some Physical and Chemical Data on Colorado Oil Shale." Bureau of Mines Report of Investigations 2152 (1920).

(12) Somerton, W. H. and Boozer, G. D. "Thermal Characteristics of Porous Rocks at Elevated Temperatures." Journal of Pet. Tech. Vol. 12 (June 1960) 77.

(13) Somerton, Wilbur H. Report to Sinclair Research, Petroleum Engineering Department, University of California (July 1964).

(14) Shaw, R. J. "Specific Heat of Colorado Oil Shales." Bureau of Mines Report of Investigations 4151 (1947).

(15) Somerton, Wilbur H. "Some Thermal Characteristics of Porous Rocks." Journal of Pet. Tech. Vol. 10 (May 1958) 61.

(16) Jukkola, E. E., Denhauler, A. J., Jensen, H. B., Barnet, W. I. and Murphy, W. I. R. "Thermal Decomposition Rates of Carbonates in Oil Shale." Industrial and Engineering Chemistry, Vol. 45, No. 12 (Dec. 1953) 2711.

(17) Feldkirchner, H. L. and Linden, H. R. "Pipeline Gas from Oil Shale by Direct Hydrogasification." I. & E. C. Process Design and Development, Vol. 3, No. 3 (July 1964) 218.

(18) Dannenberg, Raymond O. and Matzick, Arthur. "A Study of the Effects of Process Variables." Bureau of Mines Report of Investigations 5545 (1960).

(19) Matzick, Arthur, Dannenberg, R. O., Ruark, J. R., Phillips, J. E., Lankford, J. C. and Guthrie, Boyd. "Development of the Bureau of Mines Gas-Combustion Oil-Shale Retorting Process." Bureau of Mines Bulletin 635 (1966).

(20) Thomas, G. W. "Some Effects of Overburden Pressure on Oil Shale During Underground Retorting." SPE Jour. Vol. 6, No. 1 (March 1966) 1.

(21) Thomas, G. W. "A Simplified Model of Conduction Heating in Systems of Limited Permeability." SPE Jour. Vol. 4, No. 4 (Dec. 1964) 335.

(22) Lesser, H. F., Bruce, G. H. and Stone, H. L. "Conduction Heating of Formations with Limited Permeability by Condensing Gases." SPE Jour. Vol. 6, No. 4 (Dec. 1966) 372.

(23) Lauwerier, H. A. "The Transport of Heat in an Oil Layer Caused by the Injection of Hot Fluid." Appl. Sci. Res. Vol. 5, Sec. A (1955) 145.

(24) Birkhoff, Garrett. "Partial Difference Methods." SIAM J., Vol. 14 (1966) 17.

(25) Peaceman, D. W. and Rachford, H. D. Jr. "The Numerical Solution of Parabolic and Elliptic Differential Equations." J. Soc. Indus. Appl. Math. No. 3 (1955).

(26) Douglas, Jim Jr. and Peaceman, D. W. "Numerical Solution of Two-Dimensional Heat-Flow Problems." A.I.Ch.E. Jour. Vol. 1, No. 4 (Dec. 1955) 505.

(27) Douglas, J. Jr. "On the Numerical Integration of $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{\partial u}{\partial t}$ by Implicit Methods." J. Soc. Indus. Appl. Math. No. 3 (1955) 42.

(28) Brian, D. L. T. "A Finite-Difference Method of High-Order Accuracy for the Solution of Three-Dimensional Transient Heat Conduction Problems." A.I.Ch.E. Jour. Vol. 7, No. 3 (Sept. 1961) 368.

(29) Bruce, G. H., Peaceman, D. W. and Rachford, H. H. Jr. "Calculations of Unsteady-State Gas Flow through Porous Media." Trans. AIME (1953) Vol. 198, 79.

(30) Blair, P. M. and Peaceman, D. W. "An Experimental Verification of a Two-Dimensional Technique for Computing Performance of Gas-Drive Reservoirs." SPE Jour. Vol. 3, No. 1 (March 1963) 19.

(31) Stone, Herbert L. and Brian, P. L T. "Numerical Solution of Convective Transport Problems." A.I.Ch.E. Jour. Vol. 9, No. 5 (Sept. 1963) 681.

(32) Larkin, Bert K. "Some Stable Explicit Difference Approximations to the Diffusion Equation." Math. of Comp. (1964) Vol. 18, No. 86, 196.

(33) Quon, D., Dranchuk, P. M., Allada, S. R. and Leung, P. K. "Application of the Alternating Direction Explicit Procedure to Two-Dimensional Natural Gas Reservoirs." J. Pet. Tech. Vol. 18 (June 1966) 137.

(34) Coats, K. H. and Terhune, H. M. "Comparison of Alternating Direction Explicit and Implicit Procedures in Two-Dimensional Flow Calculations." SPE Jour. Vol. 6, No. 4 (Dec. 1966) 350.

(35) Schneider, P. J. Conduction Heat Transfer. Addison-Wesley Publishing Company, Inc., Reading, Massachusetts (1955) 242.

(36) Eckert, E. R. G. and Drake, Robert M. Jr. Heat and Mass Transfer. McGraw-Hill Book Company, Inc., New York, New York (1959) 12.

DEFINITION OF TERMINOLOGY

h = fracture thickness

ρ_g = gas density

c_g = gas thermal capacity

v_g = gas velocity

ρ_s = shale density

$c_s(T)$ = shale thermal capacity

$q(T)$ = endothermic heat loss

t = time

Q = rate of heat flow

T = temperature

T_g = temperature of gas

T_s = temperature of shale

T_0 = initial temperature of formation

$(\Delta T)_{max}$ = maximum allowable temperature difference between grid points or between time steps

x = horizontal distance from injection well along fracture

y = horizontal distance from fracture perpendicular to fracture

z = vertical distance from fracture centerline

$K(T)$ = thermal conductivity

$K_x(T)$ = thermal conductivity in x direction

$K_y(T)$ = thermal conductivity in y direction

$K_z(T)$ = thermal conductivity in z direction

I = number of grid points in x direction

J = number of grid points in y direction

M = number of grid points in z direction

i = subscript, refers to point location in x direction

j = subscript, refers to point location in y direction

m = subscript, refers to point location in z direction

n = superscript, refers to time step

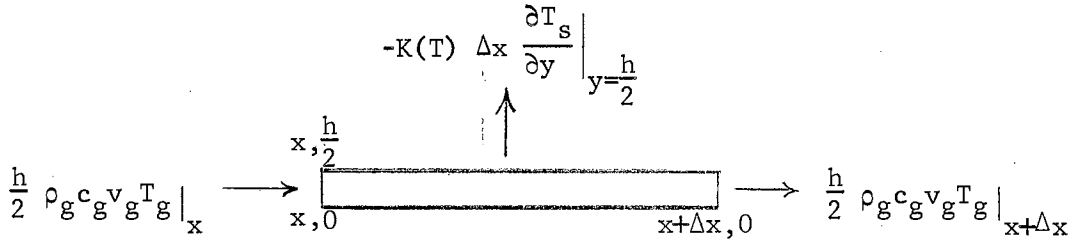
k = superscript, refers to iteration step

APPENDIX A

DIFFERENTIAL EQUATIONS

Derivation of Convection Equation

Consider a differential element in the fracture.



Therefore a differential heat balance results in

$$\frac{h}{2}\ \rho_g c_g v_g T_g\Big|_x - \frac{h}{2}\ \rho_g c_g v_g T_g\Big|_{x+\Delta x} - \left[-K(T)\ \Delta x\ \frac{\partial T_s}{\partial y}\right]\Big|_{y=\frac{h}{2}} = \frac{h}{2}\ \rho_g c_g \Delta x\ \frac{\partial T_g}{\partial t}$$

$$\text{or} \quad -\frac{h}{2}\ \rho_g c_g v_g \left(T_g\Big|_{x+\Delta x} - T_g\Big|_x\right) + K(T)\ \Delta x\ \frac{\partial T_s}{\partial y}\Big|_{y=\frac{h}{2}} = \frac{h}{2}\ \rho_g c_g \Delta x\ \frac{\partial T_g}{\partial t}$$

$$\text{or} \quad -\frac{h}{2}\ \rho_g c_g v_g \frac{\left(T_{x+\Delta x} - T_x\right)_g}{\Delta x} + K(T)\ \frac{\partial T_s}{\partial y}\Big|_{y=\frac{h}{2}} = \frac{h}{2}\ \rho_g c_g\ \frac{\partial T_g}{\partial t}\ .$$

In the limit we have

$$-\frac{h}{2}\ \rho_g c_g v_g\ \frac{\partial T_g}{\partial x} + K(T)\ \frac{\partial T_s}{\partial y}\Big|_{y=\frac{h}{2}} = \frac{h}{2}\ \rho_g c_g\ \frac{\partial T_g}{\partial t}$$

or

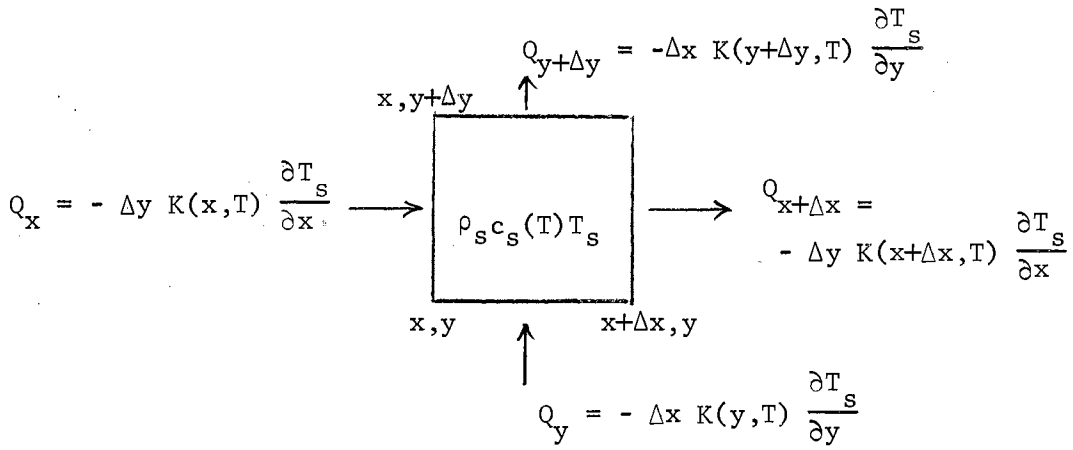
$$\frac{h}{2}\,\rho_g c_g v_g \frac{\partial T_g}{\partial x} + \frac{h}{2}\,\rho_g c_g \frac{\partial T_g}{\partial t} = K(T)\left.\frac{\partial T_s}{\partial y}\right|_{y=\frac{h}{2}} .$$

Let $\alpha = \frac{h}{2}\,\rho_g c_g v_g$.

Then: $$\frac{\partial T_g}{\partial x} + \frac{1}{v_g}\frac{\partial T_g}{\partial t} = \frac{K(T)}{\alpha}\left.\frac{\partial T_s}{\partial y}\right|_{y=\frac{h}{2}} . \qquad \text{(A-1)}$$

Derivation of Conduction Equation

Consider a differential element in the matrix where no mass flow occurs within or out of it.



A differential heat balance gives

heat in - heat out - endothermic heat loss = change in heat content.

$$\Delta t\,[Q_x + Q_y - Q_{x+\Delta x} - Q_{y+\Delta y} - \Delta x \Delta y\, q(T)] = \Delta x \Delta y$$

$$\{[\rho_s c_s(T) T_s]_{t+\Delta t} - [\rho_s c_s(T) T_s]_t\} .$$

Then

$$-\frac{Q_{x+\Delta x}-Q_x}{\Delta x\Delta y}-\frac{Q_{y+\Delta y}-Q_y}{\Delta x\Delta y}-q(T)=\frac{\left[\rho_s c_s(T)T_s\right]_{t+\Delta t}-\left[\rho_s c_s(T)T_s\right]_t}{\Delta t}.$$

Substituting in Fourier Law:

$$\frac{\left[K(x+\Delta x,T)\frac{\partial T_s}{\partial x}\right]_{x+\Delta x}-\left[K(x,T)\frac{\partial T_s}{\partial x}\right]_x}{\Delta x}+$$

$$\frac{\left[K(y+\Delta y,T)\frac{\partial T_s}{\partial y}\right]_{y+\Delta y}-\left[K(y,T)\frac{\partial T_s}{\partial y}\right]_y}{\Delta y}-q(T)=$$

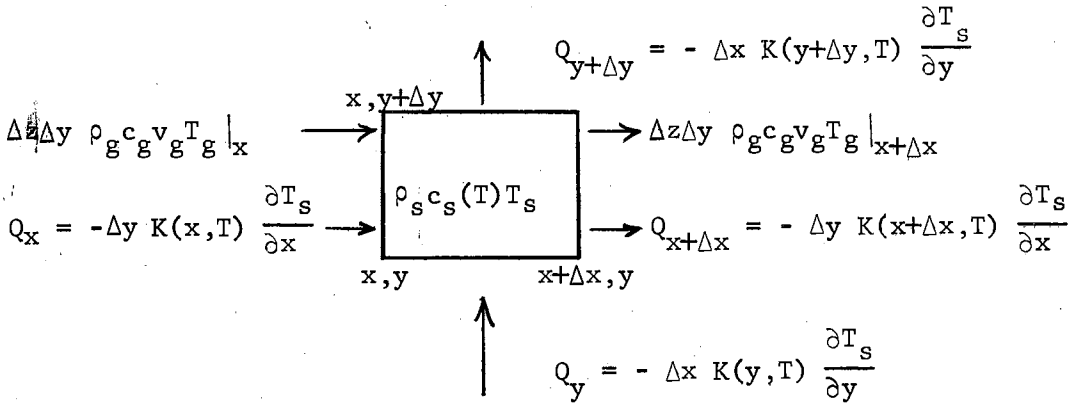
$$\frac{[\rho_s c_s(T)T_s]_{t+\Delta t}-[\rho_s c_s(T)T_s]_t}{\Delta t}.$$

In the limit we have

$$\frac{\partial}{\partial x}\left[K_x(T)\frac{\partial T_s}{\partial x}\right]+\frac{\partial}{\partial y}\left[K_y(T)\frac{\partial T_s}{\partial y}\right]-q(T)=\frac{\partial}{\partial t}\left[\rho_s c_s(T)T_s\right]. \qquad (A\text{-}2)$$

Derivation of Combined Conduction-Convection Equation

Consider a differential element with constant density and a thickness of z.



A differential heat balance gives:

heat in - heat out - endothermic heat loss = change in heat content.

$$\Delta t \left[Q_y + Q_x + \Delta z \Delta y \, \rho_g c_g v_g T_g \Big|_x - Q_{y+\Delta y} - Q_{x+\Delta x} \right.$$

$$\left. - \Delta z \Delta y \, \rho_g c_g v_g T_g \Big|_{x+\Delta x} - \Delta x \Delta y \, q(T) \right] = \Delta x \Delta y$$

$$\left\{ \left[\rho_s c_s(T) T_s \right]_{t+\Delta t} - \left[\rho_s c_s(T) T_s \right]_t \right\} .$$

Assume $T_g = T_s$.

Rearranging

$$- \frac{Q_{x+\Delta x} - Q_x}{\Delta x \Delta y} - \frac{Q_{y+\Delta y} - Q_y}{\Delta x \Delta y} - \Delta z \frac{\rho_g c_g v_g \left(T_s \Big|_{x+\Delta x} - T_s \Big|_x \right)}{\Delta x} - q(T) =$$

$$\frac{\left[\rho_s c_s(T) T_s \right]_{t+\Delta t} - \left[\rho_s c_s(T) T_s \right]_t}{\Delta t} .$$

Substituting in Fourier Law

$$- \frac{\left[- \Delta y \, K(x+\Delta x, T) \frac{\partial T_s}{\partial x} \right]_{x+\Delta x} - \left[- \Delta y \, K(x,T) \frac{\partial T_s}{\partial x} \right]_x}{\Delta x \Delta y} -$$

$$\frac{\left[- \Delta x \, K(y+\Delta y, T) \frac{\partial T_s}{\partial y} \right]_{y+\Delta y} - \left[- \Delta y \, K(y,T) \frac{\partial T_s}{\partial y} \right]_y}{\Delta x \Delta y} - q(T)$$

$$- \Delta z \frac{\rho_g c_g v \left(T_s \Big|_{x+\Delta x} - T_s \Big|_x \right)}{\Delta x} = \frac{\left[\rho_s c_s(T) T_s \right]_{t+\Delta t} - \left[\rho_s c_s(T) T_s \right]_t}{\Delta t} .$$

In the limit

$$\frac{\partial}{\partial x} \left[K_x(T) \frac{\partial T_s}{\partial x} \right] + \frac{\partial}{\partial y} \left[K_y(T) \frac{\partial T_s}{\partial y} \right] - q(T)$$

$$- \Delta z \, \rho_g c_g v_g \frac{\partial T_s}{\partial x} = \frac{\partial}{\partial t} \left[\rho_s c_s(T) T_s \right] .$$

Let $2\alpha = \Delta z \, \rho_g c_g v_g$.

Then:

$$\frac{\partial}{\partial x} \left[K_x(T) \frac{\partial T_s}{\partial x} \right] + \frac{\partial}{\partial y} \left[K_y(T) \frac{\partial T_s}{\partial y} \right] - q(T) - 2\alpha \frac{\partial T_s}{\partial x} =$$

$$\frac{\partial}{\partial t} \left[\rho_s c_s(T) T_s \right] . \qquad \text{(A-3)}$$

APPENDIX B

DIFFERENCE EQUATIONS

Variable Spacing Scheme

The interval $0 \leq x \leq x_I$ is divided into I intervals of arbitrary length, $\Delta x_{i+\frac{1}{2}}$, $0 \leq i \leq I-1$, so that

$$x_i = \sum_{r=0}^{r=i-1} \Delta x_{r+\frac{1}{2}} \quad 1 \leq i \leq I$$

with $x_0 = 0$, x_I = total distance in x direction.

Similarly, the interval $0 \leq y \leq y_J$ is divided into J intervals of arbitrary length, $\Delta y_{j+\frac{1}{2}}$, $0 \leq j \leq J-1$ so that

$$y_j = \sum_{r=0}^{r=j-1} \Delta y_{r+\frac{1}{2}} \quad 1 \leq j \leq J$$

with $y_0 = 0$, y_J = total distance in y direction.

Similarly, the interval $0 \leq z \leq z_M$ is divided into M intervals of arbitrary length, $\Delta z_{m+\frac{1}{2}}$, $0 \leq m \leq M-1$ so that

$$z_m = \sum_{r=0}^{r=m-1} \Delta z_{r+\frac{1}{2}} \quad 1 \leq m \leq M$$

with $z_0 = 0$, z_M = total distance in z direction.

For the points on the boundary, the above definitions can be extended as follows:

$$\Delta x_{-\frac{1}{2}} = \Delta x_{\frac{1}{2}} \qquad \Delta x_{I+\frac{1}{2}} = \Delta x_{I-\frac{1}{2}}$$

$$\Delta y_{-\frac{1}{2}} = \Delta y_{\frac{1}{2}} \qquad \Delta y_{J+\frac{1}{2}} = \Delta y_{J-\frac{1}{2}}$$

$$\Delta z_{-\frac{1}{2}} = \Delta z_{\frac{1}{2}} \qquad \Delta z_{K+\frac{1}{2}} = \Delta z_{K-\frac{1}{2}} .$$

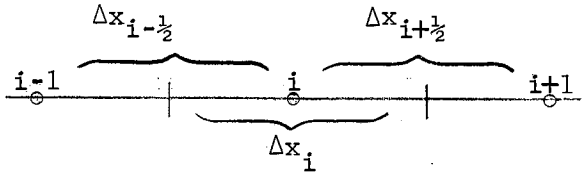
For ease of expression in difference form we can write:

$$\Delta x_i = \tfrac{1}{2} (\Delta x_{i-\frac{1}{2}} + \Delta x_{i+\frac{1}{2}}). \tag{B-1}$$

$$\Delta y_j = \tfrac{1}{2} (\Delta y_{j-\frac{1}{2}} + \Delta y_{i+\frac{1}{2}}). \tag{B-2}$$

$$\Delta z_m = \tfrac{1}{2} (\Delta z_{m-\frac{1}{2}} + \Delta z_{m+\frac{1}{2}}). \tag{B-3}$$

These relations are illustrated as follows:



Also we define:

$$(K_x)_{i+\frac{1}{2}} = 1/2 \left[(K_x)_i + (K_x)_{i+1}\right] . \tag{B-4}$$

$$(K_y)_{j+\frac{1}{2}} = 1/2 \left[(K_y)_j + (K_y)_{j+1}\right] . \tag{B-5}$$

Convection Difference Equations for THERMAB3 and THERMAB4

$$\frac{\partial T_g}{\partial x} + \frac{1}{v_g} \frac{\partial T_g}{\partial t} = \frac{K(T)}{\alpha} \frac{\partial T_s}{\partial y}\bigg|_{y=\frac{h}{2}} . \tag{B-6}$$

Use backward difference for $\frac{\partial T_g}{\partial x}$. Use forward difference for $\frac{\partial T_s}{\partial y}$.

$$\frac{\left(T_{i,1}^{n+1} - T_{i-1,1}^{n+1}\right)^{k+\frac{1}{2}}}{\Delta x_i} + \frac{1}{v_i} \frac{\left(T_{i,1}^{n+1}\right)^{k+\frac{1}{2}} - T_{i,1}^{n}}{\Delta t} =$$

$$\frac{3 \left(K_{i,3/2}^{n+1}\right)^k \left[\left(T_{i,2}^{n+1}\right)^k - \left(T_{i,1}^{n+1}\right)^{k+\frac{1}{2}}\right] - \left(K_{i,5/2}^{n+1}\right)^k \left[\left(T_{i,3}^{n+1}\right)^k - \left(T_{i,2}^{n+1}\right)^k\right]}{2 \Delta y \alpha} \tag{B-7}$$

Rearranging:

$$\left(T_{i,1}^{n+1} - T_{i-1,1}^{n+1}\right)^{k+\frac{1}{2}} + \frac{\Delta x_i}{\Delta t v_i}\left(T_{i,1}^{n+1}\right)^{k+\frac{1}{2}} =$$

$$\frac{\Delta x_i}{\Delta t\ v_i}\left(T_{i,1}^{n}\right) - \frac{3\ \Delta x_i \left(K_{i,3/2}^{n+1}\right)^k}{2\ \Delta y\ \alpha}\left(T_{i,1}^{n+1}\right)^{k+\frac{1}{2}} +$$

$$\frac{3\ \Delta x_i \left(K_{i,3/2}^{n+1}\right)^k}{2\ \Delta y\ \alpha}\left(T_{i,2}^{n+1}\right)^{k} + \frac{\Delta x_i \left(K_{i,5/2}^{n+1}\right)^k}{2\ \Delta y\ \alpha}\left[\left(T_{i,2}^{n+1}\right)^{k} - \left(T_{i,3}^{n+1}\right)^{k}\right] .$$

Rearranging again:

$$\left(T_{i,1}^{n+1}\right)^{k+\frac{1}{2}}\left[1 + \frac{\Delta x_i}{\Delta t\ v_i} + \frac{3\ \Delta x_i \left(K_{i,3/2}^{n+1}\right)^k}{2\ \Delta y\ \alpha}\right] = \frac{\Delta x_i}{\Delta t\ v_i}\left(T_{i,1}^{n}\right)$$

$$\left(T_{i-1,1}^{n+1}\right)^{k+\frac{1}{2}} + \frac{3\ \Delta x_i \left(K_{i,3/2}^{n+1}\right)^k}{2\ \Delta y\ \alpha}\left(T_{i,2}^{n+1}\right)^{k} +$$

$$\frac{\Delta x_i \left(K_{i,5/2}^{n+1}\right)^k}{2\ \Delta y\ \alpha}\left[\left(T_{i,2}^{n+1}\right)^{k} - \left(T_{i,3}^{n+1}\right)^{k}\right] .$$

Resulting in:

$$\left(T_{i,1}^{n+1}\right)^{k+\frac{1}{2}} = \frac{\left(T_{i-1,1}^{n+1}\right)^{k+\frac{1}{2}} + (D13) + (D14)^k + (D15)^k}{D16 + (D17)^k} , \qquad (B\text{-}8)$$

where:

$$(D13) = \frac{\Delta x_i\ T_{i,1}^{n}}{\Delta t\ v_i} .$$

$$(D14)^k = \frac{3\ \Delta x_i \left(K_{i,3/2}^{n+1}\right)^k \left(T_{i,2}^{n+1}\right)^k}{2\ \Delta y\ \alpha} .$$

$$(D15)^k = \frac{\Delta x_i \left(K^{n+1}_{i,5/2}\right)^k \left[\left(T^{n+1}_{i,2}\right)^k - \left(T^{n+1}_{i,3}\right)^k\right]}{2\ \Delta y\ \alpha} .$$

$$(D16) = 1 + \frac{\Delta x_i}{\Delta t\ v_i} .$$

$$(D17)^k = \frac{3\ \Delta x_i \left(K^{n+1}_{i,3/2}\right)^k}{2\ \Delta y\ \alpha}$$

Or also:

$$\left(T^{n+1}_{i,1}\right)^{k+1} = \frac{\left(T^{n+1}_{i-1,1}\right)^{k+1} + (D13) + (D14)^{k+\frac{1}{2}} + (D15)^{k+\frac{1}{2}}}{D16 + (D17)^{k+\frac{1}{2}}} . \qquad (B\text{-}9)$$

Boundary Condition:

$$\left(T^{n+1}_{1,1}\right)^{k+\frac{1}{2}} = \text{Constant Temperature}$$

$$\left(T^{n+1}_{1,1}\right)^{k+1} = \text{Constant Temperature}$$

Also, for THERMAB4 the countercurrent flow equations at y equal y_J are:

$$\left(T^{n+1}_{i,J}\right)^{k+\frac{1}{2}} = \frac{\left(T^{n+1}_{i+1,J}\right)^{k+\frac{1}{2}} + (D13) + (D14)^k + (D15)^k}{D16 + (D17)^k} , \qquad (B\text{-}10)$$

where:

$$(D13) = \frac{\Delta x_i\ T^n_{i,J}}{\Delta t\ v_i} .$$

$$(D14)^k = \frac{3\Delta x_i \left(K^{n+1}_{i,J-\frac{1}{2}}\right)^k \left(T^{n+1}_{i,J-1}\right)^k}{2\ \Delta y\ \alpha} .$$

$$(D15)^k = \frac{\Delta x_i \left(K^{n+1}_{i,J-3/2}\right) \left[\left(T^{n+1}_{i,J-1}\right)^k - \left(T^{n+1}_{i,J-2}\right)^k\right]}{2\ \Delta y\ \alpha} .$$

$$D16 = 1 + \frac{\Delta x_i}{\Delta t\ v_i} .$$

$$(D17)^k = \frac{3\ \Delta x_i \left(K^{n+1}_{i,J-\frac{1}{2}}\right)^k}{2\ \Delta y\ \alpha} .$$

And,

$$\left(T^{n+1}_{k,J}\right)^{k+1} = \frac{\left(T^{n+1}_{i+1,J}\right)^{k+1} + D13 + (D14)^{k+\frac{1}{2}} + (D15)^{k+\frac{1}{2}}}{D16 + (D17)^{k+\frac{1}{2}}} . \qquad (B\text{-}11)$$

Boundary Condition:

$$\left(T^{n+1}_{I,J}\right)^{k+\frac{1}{2}} = \text{Constant Temperature}$$

$$\left(T^{n+1}_{I,J}\right)^{k+1} = \text{Constant Temperature.}$$

Conduction Difference Equations

for THERMAB3 and THERMAB4

$$\frac{\partial}{\partial x}\left(K_x(T)\ \frac{\partial T_s}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_y(T)\ \frac{\partial T_s}{\partial y}\right) - q(T) = \frac{\partial}{\partial t}\left(\rho_s c_s(T) T_s\right) . \qquad (B\text{-}12)$$

Where $q(T)$ can be expressed as $C'(T)\ \frac{\partial T}{\partial t}$,

let $C(T) = C'(T) + \rho_s\ c_s(T)$,

let $(K_x) = (K_y) = K$.

General Difference Representation -- Using Central Differences, Implicit in x, Explicit in y.

$$\frac{1}{\Delta x_i}\left(K_{i+\frac{1}{2}}\frac{T^{n+1}_{i+1,j}-T^{n+1}_{i,j}}{\Delta x_{i+\frac{1}{2}}}-K_{i-\frac{1}{2}}\frac{T^{n+1}_{i,j}-T^{n+1}_{i-1,j}}{\Delta x_{i-\frac{1}{2}}}\right)^{k+\frac{1}{2}}+$$

$$\frac{1}{\Delta y_j}\left(K_{j+\frac{1}{2}}\frac{T^{n+1}_{i,j+1}-T^{n+1}_{i,j}}{\Delta y_{j+\frac{1}{2}}}-K_{j-\frac{1}{2}}\frac{T^{n+1}_{i,j}-T^{n+1}_{i,j-1}}{\Delta y_{j-\frac{1}{2}}}\right)^{k}=$$

$$\left(C^{n+1}_{i,j}\right)^{k}\frac{\left(T^{n+1}_{i,j}\right)^{k+\frac{1}{2}}-T^{n}_{i,j}}{\Delta t}. \qquad \text{(B-13)}$$

For constant x spacing and regrouping:

$$\left(T^{n+1}_{i+1,j}\right)^{k+\frac{1}{2}}-\left(T^{n+1}_{i,j}\right)^{k+\frac{1}{2}}-\frac{K_{i-\frac{1}{2}}}{K_{i+\frac{1}{2}}}\left(T^{n+1}_{i,j}-T^{n+1}_{i-1,j}\right)^{k+\frac{1}{2}}$$

$$-\frac{(\Delta x)^2\left(C^{n+1}_{i,j}\right)^{k}}{\Delta t\,K_{i+\frac{1}{2}}}\left(T^{n+1}_{i,j}\right)^{k+\frac{1}{2}}=$$

$$-\frac{(\Delta x)^2}{\Delta y_i\,K_{i+\frac{1}{2}}}\left(K_{j+\frac{1}{2}}\frac{T^{n+1}_{i,j+1}-T^{n+1}_{i,j}}{\Delta y_{j+\frac{1}{2}}}-K_{j-\frac{1}{2}}\frac{T^{n+1}_{i,j}-T^{n+1}_{i,j-1}}{\Delta y_{j-\frac{1}{2}}}\right)^{k}$$

$$-\frac{(\Delta x)^2\left(C^{n+1}_{i,j}\right)^{k}}{\Delta t\,K_{i+\frac{1}{2}}}\left(T^{n}_{i,j}\right).$$

Rearranging:

$$\frac{K_{i-\frac{1}{2}}}{K_{i+\frac{1}{2}}}\left(T^{n+1}_{i-1,j}\right)^{k+\frac{1}{2}} - \left[1 + \frac{K_{i-\frac{1}{2}}}{K_{i+\frac{1}{2}}} + \frac{(\Delta x)^2\left(C^{n+1}_{i,j}\right)^k}{\Delta t\ K_{i+\frac{1}{2}}}\right]\left(T^{n+1}_{i,j}\right)^{k+\frac{1}{2}} +$$

$$\left(T^{n-1}_{i+1,j}\right)^{k+\frac{1}{2}} = -\frac{(\Delta x)^2}{\Delta y_i\ K_{i+\frac{1}{2}}}\left(K_{j+\frac{1}{2}}\frac{T^{n+1}_{i,j+1} - T^{n+1}_{i,j}}{\Delta y_{j+\frac{1}{2}}} -\right.$$

$$\left. K_{j-\frac{1}{2}}\frac{T^{n+1}_{i,j} - T^{n+1}_{i,j-1}}{\Delta y_{j-\frac{1}{2}}}\right)^k - \frac{(\Delta x)^2\left(C^{n+1}_{i,j}\right)^k}{\Delta t\ K_{i+\frac{1}{2}}}\left(T^{n}_{i,j}\right).$$

Let $Al_i = \dfrac{K_{i-\frac{1}{2}}}{K_{i+\frac{1}{2}}}$.

Let $Bl_i = -\left[1 + \dfrac{K_{i-\frac{1}{2}}}{K_{i+\frac{1}{2}}} + \dfrac{(\Delta x)^2\left(C^{n+1}_{i,j}\right)^k}{\Delta t\ K_{i+\frac{1}{2}}}\right]$.

Let $Cl_i = 1.$

Let $Dl_i = -\dfrac{(\Delta x)^2}{\Delta y_i\ K_{i+\frac{1}{2}}}\left(K_{j+\frac{1}{2}}\dfrac{T^{n+1}_{i,j+1} - T^{n+1}_{i,j}}{\Delta y_{i+\frac{1}{2}}} - K_{j-\frac{1}{2}}\dfrac{T^{n+1}_{i,j} - T^{n+1}_{i,j-1}}{\Delta y_{j-\frac{1}{2}}}\right)^k$

$$- \frac{(\Delta x)^2\left(C^{n+1}_{i,j}\right)^k}{\Delta t\ K_{i+\frac{1}{2}}}\left(T^{n}_{i,j}\right).$$

Then:

$$Al_i\left(T^{n+1}_{i-1,j}\right)^{k+\frac{1}{2}} + Bl_i\left(T^{n+1}_{i,j}\right)^{k+\frac{1}{2}} + Cl_i\left(T^{n+1}_{i+1,j}\right)^{k+\frac{1}{2}} = Dl_i. \qquad \text{(B-14)}$$

Implicit in y, Explicit in x, Constant x Spacing

$$\frac{1}{(\Delta x)^2}\left[K_{i+\frac{1}{2}}\left(T^{n+1}_{i+1,j} - T^{n+1}_{i,j}\right) - K_{i-\frac{1}{2}}\left(T^{n+1}_{i,j} - T^{n+1}_{i-1,j}\right)\right]^{k+\frac{1}{2}} +$$

$$\frac{1}{\Delta y_i}\left(K_{j+\frac{1}{2}}\frac{T^{n+1}_{i,j+1} - T^{n+1}_{i,j}}{\Delta y_{j+\frac{1}{2}}} - K_{j-\frac{1}{2}}\frac{T^{n+1}_{i,j} - T^{n+1}_{i,j-1}}{\Delta y_{j-\frac{1}{2}}}\right)^{k+1} =$$

$$\left(C^{n+1}_{i,j}\right)^{k+\frac{1}{2}}\frac{\left(T^{n+1}_{i,j}\right)^{k+1} - T^{n}_{i,j}}{\Delta t} \quad . \tag{B-15}$$

Regrouping:

$$\left(T^{n+1}_{i,j+1}\right)^{k+1} - \left(T^{n+1}_{i,j}\right)^{k+1} - \frac{K_{j-\frac{1}{2}}\,\Delta y_{j+\frac{1}{2}}}{K_{j+\frac{1}{2}}\,\Delta y_{j-\frac{1}{2}}}\left(T^{n+1}_{i,j} - T^{n+1}_{i,j-1}\right)^{k+1}$$

$$- \frac{\left(C^{n+1}_{i,j}\right)^{k+\frac{1}{2}}\Delta y_j\,\Delta y_{j+\frac{1}{2}}}{\Delta t\,K_{j+\frac{1}{2}}}\left(T^{n+1}_{i,j}\right)^{k+1} = - \frac{\left(C^{n+1}_{i,j}\right)^{k+\frac{1}{2}}\Delta y_j\,\Delta y_{j+\frac{1}{2}}}{\Delta t\,K_{j+\frac{1}{2}}}\left(T^{n}_{i,j}\right)$$

$$- \frac{\Delta y_j\,\Delta y_{j+\frac{1}{2}}}{(\Delta x)^2\,K_{j+\frac{1}{2}}}\left[K_{i+\frac{1}{2}}\left(T^{n+1}_{i+1,j} - T^{n+1}_{i,j}\right) - K_{i-\frac{1}{2}}\left(T^{n+1}_{i,j} - T^{n+1}_{i-1,j}\right)\right]^{k+\frac{1}{2}} .$$

Let $\quad Al_j = \dfrac{K_{j+\frac{1}{2}}\,\Delta y_{j+\frac{1}{2}}}{K_{j+\frac{1}{2}}\,\Delta y_{j-\frac{1}{2}}} \quad .$

Let $\quad Bl_j = -\left[1 + \dfrac{K_{j-\frac{1}{2}}\,\Delta y_{j+\frac{1}{2}}}{K_{j+\frac{1}{2}}\,\Delta y_{j-\frac{1}{2}}} + \dfrac{\left(C^{n+1}_{i,j}\right)^{k+\frac{1}{2}}\Delta y_i\,\Delta y_{j+\frac{1}{2}}}{\Delta t\,K_{j+\frac{1}{2}}}\right] \quad .$

Let $\quad Cl_j = 1 \quad .$

Let $$D1_j = -\frac{\Delta y_j \, \Delta y_{j+\frac{1}{2}}}{(\Delta x)^2 \, K_{j+\frac{1}{2}}} \left[K_{i+\frac{1}{2}} \left(T^{n+1}_{i+1,j} - T^{n+1}_{i,j} \right) - K_{i-\frac{1}{2}} \left(T^{n+1}_{i,j} - T^{n+1}_{i-1,j} \right) \right]^{k+\frac{1}{2}}$$

$$- \frac{\left(C^{n+1}_{i,j} \right)^{k+\frac{1}{2}} \Delta y_j \, \Delta y_{j+\frac{1}{2}}}{\Delta t \, K_{j+\frac{1}{2}}} \left(T^{n}_{i,j} \right) .$$

Then:

$$A1_j \left(T^{n+1}_{i,j-1} \right)^{k+1} + B1_j \left(T^{n+1}_{i,j} \right)^{k+1} + C1_j \left(T^{n+1}_{i,j+1} \right)^{k+1} = D1_j .$$

(B-16)

Boundary Conditions for THERMAB3 (Heating Period)

1) $\left(T^{n+1}_{0,j} \right)^{k+\frac{1}{2}} = \left(T^{n+1}_{2,j} \right)^{k+\frac{1}{2}}$ for $2 \le j \le J$.

2) $\left(T^{n+1}_{I-1,j} \right)^{k+\frac{1}{2}} = \left(T^{n+1}_{I+1,j} \right)^{k+\frac{1}{2}}$ for $2 \le j \le J$.

3) $\left(T^{n+1}_{i,1} \right)^{k+1} = \left(T^{n+1}_{i,1} \right)^{k+\frac{1}{2}}$ for all i .

4) $\left(T^{n+1}_{i,J-1} \right)^{k+1} = \left(T^{n+1}_{i,J+1} \right)^{k+1}$ for all i .

Boundary Conditions for THERMAB4 (Heating Period)

1) $\left(T^{n+1}_{0,j} \right)^{k+\frac{1}{2}} = \left(T^{n+1}_{2,j} \right)^{k+\frac{1}{2}}$ for $2 \le j \le J-1$.

2) $\left(T^{n+1}_{I-1,j} \right)^{k+\frac{1}{2}} = \left(T^{n+1}_{I+1,j} \right)^{k+\frac{1}{2}}$ for $2 \le j \le J-1$.

3) $\left(T^{n+1}_{i,1} \right)^{k+1} = \left(T^{n+1}_{i,1} \right)^{k+\frac{1}{2}}$ for all i .

4) $\left(T^{n+1}_{i,J} \right)^{k+1} = \left(T^{n+1}_{i,J} \right)^{k+\frac{1}{2}}$ for all i .

Boundary Conditions for THERMAB3 and THERMAB4 (Soaking Period)

1) $\left(T_{0,j}^{n+1}\right)^{k+\frac{1}{2}} = \left(T_{2,j}^{n+1}\right)^{k+\frac{1}{2}}$ for all j .

2) $\left(T_{I-1,j}^{n+1}\right)^{k+\frac{1}{2}} = \left(T_{I+1,j}^{n+1}\right)^{k+\frac{1}{2}}$ for all j .

3) $\left(T_{i,0}^{n+1}\right)^{k+1} = \left(T_{i,2}^{n+1}\right)^{k+1}$ for all i .

4) $\left(T_{i,J-1}^{n+1}\right)^{k+1} = \left(T_{i,J+1}^{n+1}\right)^{k+1}$ for all i .

Conduction Difference Equations for THERMAB1

Using derivations for THERMAB3 and THERMAB4 and defining

$K_x = K_{i-\frac{1}{2}} = K_{i+\frac{1}{2}}$, $\qquad K_y = K_{j-\frac{1}{2}} = K_{j+\frac{1}{2}}$

$C = C_{i,j}^{n+1}$

$q = 0$, no endothermic heat loss.

For Constant Coefficients, Implicit in x, Explicit in y

Therefore:

$$A1_i = 1.$$

$$B1_i = -\left[2 + \frac{(\Delta x)^2 C}{\Delta t\, K_x}\right] .$$

$$C1_i = 1 .$$

$$D1_i = -\frac{(\Delta x)^2 K_y}{\Delta y_i K_x}\left(\frac{T_{i,j+1}-T_{i,j}}{\Delta y_{j+\frac{1}{2}}} - \frac{T_{i,j}-T_{i,j-1}}{\Delta y_{j-\frac{1}{2}}}\right)^k - \frac{(\Delta x)^2 C\, T_{i,j}^n}{\Delta t\, K_x} .$$

Then:

$$A1_i \left(T^{n+1}_{i-1,j}\right)^{k+\frac{1}{2}} + B1_i \left(T^{n+1}_{i,j}\right)^{k+\frac{1}{2}} + C1_i \left(T^{n+1}_{i+1,j}\right)^{k+\frac{1}{2}} = D1_i \ .$$

Implicit in y and Explicit in x

$$A1_j = \frac{\Delta y_{j+\frac{1}{2}}}{\Delta y_{j-\frac{1}{2}}} \ .$$

$$B1_j = -\left[1 + \frac{\Delta y_{j+\frac{1}{2}}}{\Delta y_{j-\frac{1}{2}}} + \frac{C\ \Delta y_j\ \Delta y_{j+\frac{1}{2}}}{\Delta t\ K_y}\right] .$$

$$C1_j = 1 \ .$$

$$D1_j = -\frac{K_x\ \Delta y_j\ \Delta y_{j+\frac{1}{2}}}{K_y\ (\Delta x)^2}\left(T^{n+1}_{i+1,j} - 2\ T^{n+1}_{i,j} + T^{n+1}_{i-1,j}\right)^{k+\frac{1}{2}}$$

$$- \frac{C\ \Delta y_j\ \Delta y_{j+\frac{1}{2}}\ T^n_{i,j}}{\Delta t\ K_y} \ .$$

Then:

$$A1_j \left(T^{n+1}_{i,j-1}\right)^{k+1} + B1_j \left(T^{n+1}_{i,j}\right)^{k+1} + C1_j \left(T^{n+1}_{i,j+1}\right)^{k+1} = D1_j \ .$$

Combined Conduction-Convection Difference

Equations for THERMAB2

$$\frac{\partial}{\partial x}\left[K_x(T)\ \frac{\partial T_s}{\partial x}\right] - 2\alpha\ \frac{\partial T_s}{\partial x} + \frac{\partial}{\partial y}\left[K_y(T)\ \frac{\partial T_s}{\partial y}\right] - q(T) = \frac{\partial}{\partial T}\left[\rho_s c_s(T)\ T_s\right].$$

Let: HH = 1 where convection is present,

HH = 0 where convection is absent,

$q(T) = C'(T)\ \frac{\partial T_s}{\partial t}$,

$C(T) = c_s(T)\ \rho_s(\ ) + C'(T)$ and $K_x = K_y = K$.

$$\frac{\partial}{\partial x}\left[K(T)\,\frac{\partial T_s}{\partial x}\right] - 2\,HH\,\alpha\,\frac{\partial T_s}{\partial x} + \frac{\partial}{\partial y}\left[K(T)\,\frac{\partial T_s}{\partial y}\right] = C(T)\,\frac{\partial T_s}{\partial t}\ . \qquad \text{(B-17)}$$

Using Central Differences Implicit in x, Explicit in y, Constant x Spacing.

$$\frac{1}{\Delta x_i}\left(K_{i+\frac{1}{2}}\,\frac{T^{n+1}_{i+1,j} - T^{n+1}_{i,j}}{\Delta x_i} - K_{i-\frac{1}{2}}\,\frac{T^{n+1}_{i,j} - T^{n+1}_{i-1,j}}{\Delta x_i}\right)^{k+\frac{1}{2}}$$

$$- 2HH\alpha\left(\frac{T^{n+1}_{i+1,j} - T^{n+1}_{i-1,j}}{2\,\Delta x_i}\right)^{k+\frac{1}{2}}$$

$$+ \frac{1}{\Delta y_i}\left(K_{j+\frac{1}{2}}\,\frac{T^{n+1}_{i,j+1} - T^{n+1}_{i,j}}{\Delta y_{i+\frac{1}{2}}} - K_{j-\frac{1}{2}}\,\frac{T^{n+1}_{i,j} - T^{n+1}_{i,j-1}}{\Delta y_{i-\frac{1}{2}}}\right)^{k}$$

$$= \left(C^{n+1}_{i,j}\right)^{k}\,\frac{\left(T^{n+1}_{i,j}\right)^{k+\frac{1}{2}} - T^{n}_{i,j}}{\Delta t}\ . \qquad \text{(B-18)}$$

Rearranging:

$$\left(T^{n+1}_{i,1,j} - T^{n+1}_{i,j}\right)^{k+\frac{1}{2}} - \frac{K_{i-\frac{1}{2}}}{K_{i+\frac{1}{2}}}\left(T^{n+1}_{i,j} - T^{n+1}_{i-1,j}\right)^{k+\frac{1}{2}}$$

$$- \frac{\Delta x_i\ HH\alpha}{K_{i+\frac{1}{2}}}\left(T^{n+1}_{i+1,j} - T^{n+1}_{i-1,j}\right)^{k+\frac{1}{2}}$$

$$+ \frac{(\Delta x_i)^2}{\Delta y_i\ K_{i+\frac{1}{2}}}\left(K_{j+\frac{1}{2}}\,\frac{T^{n+1}_{i,j+1} - T^{n+1}_{i,j}}{\Delta y_{i+\frac{1}{2}}} - K_{j-\frac{1}{2}}\,\frac{T_{i,j} - T_{i,j-1}}{\Delta y_{i-\frac{1}{2}}}\right)^{k}$$

$$= \frac{\left(C^{n+1}_{i,j}\right)^{k}\left(\Delta x_i\right)^2}{K_{i+\frac{1}{2}}\,\Delta t}\left[\left(T^{n+1}_{i,j}\right)^{k+\frac{1}{2}} - T^{n}_{i,j}\right]\ .$$

Rearranging again,

$$\left(T^{n+1}_{i-1,j}\right)^{k+\frac{1}{2}}\left(\frac{K_{i-\frac{1}{2}}}{K_{i+\frac{1}{2}}}+\frac{\Delta x_i\ HH\alpha}{K_{i+\frac{1}{2}}}\right)-\left(T^{n+1}_{i,j}\right)^{k+\frac{1}{2}}\left[1+\frac{K_{i-\frac{1}{2}}}{K_{i+\frac{1}{2}}}+\frac{\left(C^{n+1}_{i,j}\right)^{k}\left(\Delta x_i\right)^{2}}{K_{i+\frac{1}{2}}\,\Delta t}\right]$$

$$-\left(T^{n+1}_{i+1,j}\right)^{k+\frac{1}{2}}\left(\frac{\Delta x_i\ HH\alpha}{K_{i+\frac{1}{2}}}-1\right)=$$

$$-\frac{(\Delta x_i)^2}{\Delta y_i\ K_{i+\frac{1}{2}}}\left(K_{j+\frac{1}{2}}\frac{T^{n+1}_{i,j+1}-T^{n+1}_{i,j}}{\Delta y_{i+\frac{1}{2}}}-K_{j-\frac{1}{2}}\frac{T_{i,j}-T_{i,j-1}}{\Delta y_{i-\frac{1}{2}}}\right)^{k}$$

$$-\frac{\left(C^{n+1}_{i,j}\right)^{k}\left(\Delta x_i\right)^{2}T^{n}_{i,j}}{K_{i+\frac{1}{2}}\,\Delta t}\ .$$

Let: $$A1_i=\left[\frac{K_{i-\frac{1}{2}}}{K_{i+\frac{1}{2}}}+\frac{\Delta x_i\ HH\alpha}{K_{i+\frac{1}{2}}}\right].$$

$$B1_i=-\left[1+\frac{K_{i-\frac{1}{2}}}{K_{i+\frac{1}{2}}}+\frac{\left(C^{n+1}_{i,j}\right)^{k}\left(\Delta x_i\right)^{2}}{K_{i+\frac{1}{2}}\,\Delta t}\right].$$

$$C1_i=-\left(\frac{\Delta x_i\ HH\alpha}{K_{i+\frac{1}{2}}}-1\right).$$

$$D1_i=\frac{\left(\Delta x_i\right)^2}{\Delta y_i\ K_{i+\frac{1}{2}}}\left(K_{j+\frac{1}{2}}\frac{T^{n+1}_{i,j+1}-T^{n+1}_{i,j}}{\Delta y_{i+\frac{1}{2}}}-K_{j-\frac{1}{2}}\frac{T^{n+1}_{i,j}-T^{n+1}_{i,j-1}}{\Delta y_{i-\frac{1}{2}}}\right)^{k}$$

$$-\frac{\left(C^{n+1}_{i,j}\right)^{k}\left(\Delta x_i\right)^{2}T^{n}_{i,j}}{K_{i+\frac{1}{2}}\,\Delta t}\ .$$

Then:

$$A1_i\, T^{n+1}_{i-1,j} + B1_i\, T^{n+1}_{i,j} + C1_i\, T^{n+1}_{i+1,j} = D1_i \quad . \tag{B-19}$$

Implicit in y, Explicit in x, Constant x Spacing

$$\frac{1}{\Delta x_i}\left(K_{i+\frac{1}{2}} \frac{T^{n+1}_{i+1,j} - T^{n+1}_{i,j}}{\Delta x_i} - K_{i-\frac{1}{2}} \frac{T^{n+1}_{i,j} - T^{n+1}_{i-1,j}}{\Delta x_i}\right)^{k+\frac{1}{2}}$$

$$- 2HH\alpha \left(\frac{T^{n+1}_{i+1,j} - T^{n+1}_{i-1,j}}{2\,\Delta x_i}\right)^{k+\frac{1}{2}}$$

$$+ \frac{1}{\Delta y_i}\left(K_{j+\frac{1}{2}} \frac{T^{n+1}_{i,j+1} - T^{n+1}_{i,j}}{\Delta y_{j+\frac{1}{2}}} - K_{j-\frac{1}{2}} \frac{T^{n+1}_{i,j} - T^{n+1}_{i,j-1}}{\Delta y_{j-\frac{1}{2}}}\right)^{k+1}$$

$$= \left(C^{n+1}_{i,j}\right)^{k+\frac{1}{2}} \frac{\left(T^{n+1}_{i,j}\right)^{k+1} - T^{n}_{i,j}}{\Delta t} \quad . \tag{B-20}$$

Rearranging:

$$\left(T^{n+1}_{i,j+1} - T^{n+1}_{i,j}\right)^{k+1} - \frac{K_{j-\frac{1}{2}}\,\Delta y_{j+\frac{1}{2}}}{K_{j+\frac{1}{2}}\,\Delta y_{j-\frac{1}{2}}}\left(T^{n+1}_{i,j} - T^{n+1}_{i,j-1}\right)^{k+1}$$

$$- \frac{\left(C^{n+1}_{i,j}\right)^{k+\frac{1}{2}}\,\Delta y_{j+\frac{1}{2}}\,\Delta y_j}{K_{j+\frac{1}{2}}\,\Delta t}\left(T^{n+1}_{i,j}\right)^{k+1} =$$

$$- \frac{\Delta y_j\,\Delta y_{j+\frac{1}{2}}}{\left(\Delta x_i\right)^2 K_{j+\frac{1}{2}}}\left[K_{i+\frac{1}{2}}\left(T^{n+1}_{i+1,j} - T^{n+1}_{i,j}\right) - K_{i-\frac{1}{2}}\left(T^{n+1}_{i,j} - T^{n+1}_{i-1,j}\right)\right]^{k+\frac{1}{2}}$$

$$+ \frac{2HH\alpha\ \Delta y_j\ \Delta y_{j+\frac{1}{2}}}{2\ \Delta x_i\ K_{j+\frac{1}{2}}} \left(T_{i+1,j}^{n+1} - T_{i-1,j}^{n+1}\right)^{k+\frac{1}{2}}$$

$$- \frac{\left(C_{i,j}^{n+1}\right)^{k+\frac{1}{2}}\ \Delta y_j\ \Delta y_{j+\frac{1}{2}}}{K_{j+\frac{1}{2}}\ \Delta t}\ T_{i,j}^{n} \ .$$

Let: $Al_j = \left(\frac{K_{j-\frac{1}{2}}\ \Delta y_{j+\frac{1}{2}}}{K_{j+\frac{1}{2}}\ \Delta y_{j-\frac{1}{2}}}\right)$.

$$Bl_j = -\left[1 + \frac{K_{j-\frac{1}{2}}\ \Delta y_{j+\frac{1}{2}}}{K_{j+\frac{1}{2}}\ \Delta y_{j-\frac{1}{2}}} + \frac{\left(C_{i,j}^{n+1}\right)^{k+\frac{1}{2}}\ \Delta y_{j+\frac{1}{2}}\ \Delta y_j}{\Delta t\ K_{j+\frac{1}{2}}}\right] .$$

$$Cl_j = 1 \ .$$

$$Dl_j = \frac{\Delta y_j\ \Delta y_{j+\frac{1}{2}}}{\left(\Delta x_i\right)^2 K_{j+\frac{1}{2}}} \left[K_{j+\frac{1}{2}} \left(T_{i+1,j}^{n+1} - T_{i,j}^{n+1}\right) - K_{j-\frac{1}{2}} \left(T_{i,j} - T_{i-1,j}\right)^{k+\frac{1}{2}}\right.$$

$$\left. + \frac{HH\alpha\ \Delta y_i\ \Delta y_{i+\frac{1}{2}}}{\Delta x_i\ K_{j+\frac{1}{2}}} \left(T_{i+1,j}^{n+1} - T_{i-1,j}^{n+1}\right)^{k+\frac{1}{2}}\right] .$$

Therefore:

$$Al_j\ T_{i,j-1}^{n+1} + Bl_j\ T_{i,j}^{n+1} + Cl_j\ T_{i,j+1}^{n+1} = Dl_j \ . \qquad \text{(B-21)}$$

Boundary Conditions for THERMAB2

1) $\left(T_{0,j}^{n+1}\right)^{k+\frac{1}{2}} = \left(T_{2,j}^{n+1}\right)^{k+\frac{1}{2}}$ for $2 \le j \le J$.

2) $\left(T_{1,1}^{n+1}\right)^{k+\frac{1}{2}} = T_{constant}$.

3) $\left(T^{n+1}_{I-1,j}\right)^{k+\frac{1}{2}} = \left(T^{n+1}_{I+1,j}\right)^{k+\frac{1}{2}}$ for $1 \le j \le J$.

4) $\left(T^{n+1}_{i,0}\right)^{k+1} = \left(T^{n+1}_{i,2}\right)^{k+1}$ for $2 \le i \le I$.

5) $\left(T^{n+1}_{1,1}\right)^{k+1} = T_{constant}$.

6) $\left(T^{n+1}_{i,J-1}\right)^{k+1} = \left(T^{n+1}_{i,J+1}\right)^{k+1}$ for $1 \le i \le I$.

Conduction Difference Equations for THERMAB5

$$\frac{\partial}{\partial z}\left(K_z \frac{\partial T}{\partial z}\right) + \frac{\partial}{\partial y}\left(K_y \frac{\partial T}{\partial y}\right) - q(T) = \frac{\partial}{\partial T}\left[\rho_s c_s(T)\, T_s\right] . \qquad \text{(B-22)}$$

Where $q(T) = C'(T) \frac{\partial T}{\partial t}$, let $C(T) = \rho_s c_s(T) + C'(T)$.

Implicit in z, Explicit in y, Variable Grid Spacing

$$\frac{1}{\Delta z_m}\left[\frac{(K_z)_{m+\frac{1}{2}}\left(T^{n+1}_{m+1,j} - T^{n+1}_{m,j}\right)^{k+\frac{1}{2}}}{\Delta z_{m+\frac{1}{2}}} - \frac{(K_z)_{m-\frac{1}{2}}\left(T^{n+1}_{m,j} - T^{n+1}_{m-1,j}\right)^{k+\frac{1}{2}}}{\Delta z_{m-\frac{1}{2}}}\right]$$

$$- \frac{1}{\Delta y_j}\left[\frac{(K_y)_{j+\frac{1}{2}}\left(T^{n+1}_{m,j+1} - T^{n+1}_{m,j}\right)^{k}}{\Delta y_{j+\frac{1}{2}}} - \frac{(K_y)_{j-\frac{1}{2}}\left(T^{n+1}_{m,j} - T^{n+1}_{m,j-1}\right)^{k}}{\Delta y_{j-\frac{1}{2}}}\right]$$

$$= \left(C^{n+1}_{m,j}\right)^{k} \frac{\left(T^{n+1}_{m,j}\right)^{k+\frac{1}{2}} - T^{n}_{m,j}}{\Delta t} . \qquad \text{(B-23)}$$

Rearranging:

$$\frac{\Delta z_{m-\frac{1}{2}}\,(K_z)_{m+\frac{1}{2}}}{\Delta z_{m+\frac{1}{2}}\,(K_z)_{m-\frac{1}{2}}}\left(T^{n+1}_{m+1,j} - T^{n+1}_{m,j}\right)^{k+\frac{1}{2}} - \left(T^{n+1}_{m,j} - T^{n+1}_{m-1,j}\right)^{k+\frac{1}{2}} +$$

$$\frac{\Delta z_{m-\frac{1}{2}}\,\Delta z_m}{\Delta y_j\,(K_z)_{m-\frac{1}{2}}}\left[\frac{(K_y)_{j+\frac{1}{2}}\left(T^{n+1}_{m,j+1} - T^{n+1}_{m,j}\right)^{k}}{\Delta y_{j+\frac{1}{2}}} - \frac{(K_y)_{j-\frac{1}{2}}\left(T^{n+1}_{m,j} - T^{n+1}_{m,j-1}\right)^{k}}{\Delta y_{j-\frac{1}{2}}}\right]$$

$$= \frac{\left(C^{n+1}_{m,j}\right)^{k}\,\Delta z_{m-\frac{1}{2}}\,\Delta z_m}{(K_z)_{m-\frac{1}{2}}\,\Delta t}\left[\left(T^{n+1}_{m,j}\right)^{k+\frac{1}{2}} - T^{n}_{m,j}\right] .$$

Regrouping:

$$\left(T^{n+1}_{m-1,j}\right)^{k+\frac{1}{2}} - \left[\frac{\Delta z_{m-\frac{1}{2}}(K_z)_{m+\frac{1}{2}}}{\Delta z_{m+\frac{1}{2}}(K_z)_{m-\frac{1}{2}}} + 1 + \frac{\left(C^{n+1}_{m,j}\right)^{k}\,\Delta z_{m-\frac{1}{2}}\,\Delta z_m}{(K_z)_{m-\frac{1}{2}}\,\Delta t}\right]\left(T^{n+1}_{m,j}\right)^{k+\frac{1}{2}} +$$

$$\frac{\Delta z_{m-\frac{1}{2}}\,(K_z)_{m+\frac{1}{2}}}{\Delta z_{m+\frac{1}{2}}\,(K_z)_{m-\frac{1}{2}}}\left(T^{n+1}_{m-1,j}\right)^{k+\frac{1}{2}} =$$

$$\frac{\Delta z_{m-\frac{1}{2}}\,\Delta z_m}{\Delta y_j\,(K_z)_{m-\frac{1}{2}}}\left[\frac{(K_y)_{j+\frac{1}{2}}\left(T^{n+1}_{m,j+1} - T^{n+1}_{m,j}\right)^{k}}{\Delta y_{j+\frac{1}{2}}} - \frac{(K_y)_{j-\frac{1}{2}}\left(T^{n+1}_{m,j} - T^{n+1}_{m,j-1}\right)^{k}}{\Delta y_{j-\frac{1}{2}}}\right]$$

$$- \frac{\left(C^{n+1}_{m,j}\right)^{k}\,\Delta z_{m-\frac{1}{2}}\,\Delta z_m}{(K_z)_{m-\frac{1}{2}}\,\Delta t}\left(T^{n}_{m,j}\right) .$$

Let: $A1_m = 1$.

$$B1_m = -\left[1 + \frac{\Delta z_{m-\frac{1}{2}}\,(K_z)_{m+\frac{1}{2}}}{\Delta z_{m+\frac{1}{2}}\,(K_z)_{m-\frac{1}{2}}} + \frac{\left(C^{n+1}_{m,j}\right)^{k}\,\Delta z_{m-\frac{1}{2}}\,\Delta z_m}{(K_z)_{m-\frac{1}{2}}\,\Delta t}\right] .$$

$$C1_m = \frac{\Delta z_{m-\frac{1}{2}}\,(K_z)_{m+\frac{1}{2}}}{\Delta z_{m+\frac{1}{2}}\,(K_z)_{m-\frac{1}{2}}} .$$

$$Dl_m = -\frac{\left(C^{n+1}_{m,j}\right)^k \Delta z_{m-\frac{1}{2}} \Delta z_m}{(K_z)_{m-\frac{1}{2}} \Delta t}\left(T^n_{m,j}\right) +$$

$$\frac{\Delta z_{m-\frac{1}{2}} \Delta z_m}{\Delta y_j (K_z)_{m-\frac{1}{2}}}\left[\frac{(K_y)_{j+\frac{1}{2}}\left(T^{n+1}_{m,j+1} - T^{n+1}_{m,j}\right)^k}{\Delta y_{j+\frac{1}{2}}} - \frac{(K_y)_{j-\frac{1}{2}}\left(T^{n+1}_{m,j} - T^{n+1}_{m,j-1}\right)^k}{\Delta y_{j-\frac{1}{2}}}\right].$$

Then:

$$Al_m \left(T^{n+1}_{m-1,j}\right)^{k+\frac{1}{2}} + Bl_m \left(T^{n+1}_{m,j}\right)^{k+\frac{1}{2}} + Cl_m \left(T^{n+1}_{m+1,j}\right)^{k+\frac{1}{2}} = Dl_m . \quad \text{(B-24)}$$

Implicit in z, Explicit in y, Variable Grid Spacing

$$\frac{1}{\Delta z_m}\left[\frac{(K_z)_{m+\frac{1}{2}}\left(T^{n+1}_{m+1,j} - T^{n+1}_{m,j}\right)^{k+\frac{1}{2}}}{\Delta z_{m+\frac{1}{2}}} - \frac{(K_z)_{m-\frac{1}{2}}\left(T^{n+1}_{m,j} - T^{n+1}_{m-1,j}\right)^{k+\frac{1}{2}}}{\Delta z_{m-\frac{1}{2}}}\right]$$

$$+ \frac{1}{\Delta y_j}\left[\frac{(K_y)_{j+\frac{1}{2}}\left(T^{n+1}_{m,j+1} - T^{n+1}_{m,j}\right)^{k+1}}{\Delta y_{j+\frac{1}{2}}} - \frac{(K_y)_{j-\frac{1}{2}}\left(T^{n+1}_{m,j} - T^{n+1}_{m,j-1}\right)^{k+1}}{\Delta y_{j-\frac{1}{2}}}\right]$$

$$= \left(C^{n+1}_{m,j}\right)^{k+\frac{1}{2}} \frac{\left(T^{n+1}_{m,j}\right)^{k+1} + T^n_{m,j}}{\Delta t} . \quad \text{(B-25)}$$

Rearranging:

$$\frac{\Delta y_{j-\frac{1}{2}} (K_y)_{j+\frac{1}{2}}}{\Delta y_{j+\frac{1}{2}} (K_y)_{j-\frac{1}{2}}}\left[T^{n+1}_{m,j+1} - T^{n+1}_{m,j}\right]^{k+1} - \left(T^{n+1}_{m,j}\right)^{k+1} + \left(T^{n+1}_{m,j-1}\right)^{k+1}$$

$$- \frac{\Delta y_{j-\frac{1}{2}} \Delta y_j \left(C^{n+1}_{m,j}\right)^{k+\frac{1}{2}}}{(K_y)_{j-\frac{1}{2}} \Delta t}\left(T^{n+1}_{m,j}\right)^{k+\frac{1}{2}} =$$

$$- \frac{\Delta y_{j-\frac{1}{2}}\, \Delta y_j}{\Delta z_m\, (K_y)_{j-\frac{1}{2}}} \left[\frac{(K_z)_{m+\frac{1}{2}} \left(T^{n+1}_{m+1,j} - T^{n+1}_{m,j} \right)^{k+\frac{1}{2}}}{\Delta z_{m+\frac{1}{2}}} - \frac{(K_z)_{m-\frac{1}{2}} \left(T^{n+1}_{m,j} - T^{n+1}_{m-1,j} \right)^{k+\frac{1}{2}}}{\Delta z_{m-\frac{1}{2}}} \right]$$

$$- \frac{\Delta y_{j-\frac{1}{2}}\, \Delta y_j \left(C^{n+1}_{m,j} \right)^{k+\frac{1}{2}}}{(K_y)_{j-\frac{1}{2}}\, \Delta t} \left(T^{n}_{m,j} \right) .$$

Let: $Al_j = 1$.

$$Bl_j = - \left[1 + \frac{\Delta y_{j-\frac{1}{2}}\, (K_y)_{j+\frac{1}{2}}}{\Delta y_{j+\frac{1}{2}}\, (K_y)_{j-\frac{1}{2}}} + \frac{\Delta y_{j-\frac{1}{2}}\, \Delta y_j \left(C^{n+1}_{m,j} \right)^{k+\frac{1}{2}}}{(K_y)_{j-\frac{1}{2}}\, \Delta t} \right] .$$

$$Cl_j = \frac{\Delta y_{j-\frac{1}{2}}\, (K_y)_{j+\frac{1}{2}}}{\Delta y_{j+\frac{1}{2}}\, (K_y)_{j-\frac{1}{2}}} .$$

$$Dl_j = - \frac{\Delta y_{j-\frac{1}{2}}\, \Delta y_j}{\Delta z_m\, (K_y)_{j-\frac{1}{2}}} \left[\frac{(K_z)_{m+\frac{1}{2}} \left(T^{n+1}_{m+1,j} - T^{n+1}_{m,j} \right)^{k+\frac{1}{2}}}{\Delta z_{m+\frac{1}{2}}} - \right.$$

$$\left. \frac{(K_z)_{m-\frac{1}{2}} \left(T^{n+1}_{m,j} - T^{n+1}_{m-1,j} \right)^{k+1}}{\Delta z_{m-\frac{1}{2}}} \right] - \frac{\Delta y_{j-\frac{1}{2}} \Delta y_j \left(C^{n+1}_{m,j} \right)^{k+\frac{1}{2}}}{(K_y)_{j-\frac{1}{2}}\, \Delta t} \left(T^{n}_{m,j} \right) .$$

Then:

$$Al_j \left(T^{n+1}_{m,j-1} \right)^{k+1} + Bl_j \left(T^{n+1}_{m,j+1} \right)^{k+1} + Cl_j \left(T^{n+1}_{m,j+1} \right)^{k+1} = Dl_j . \quad \text{(B-26)}$$

Boundary Conditions for THERMAB5

1) $\left(T^{n+1}_{0,j} \right)^{k+\frac{1}{2}} = \left(T^{n+1}_{2,j} \right)^{k+\frac{1}{2}} \qquad 2 \le j \le J .$

2) $\left(T^{n+1}_{M-1,j} \right)^{k+\frac{1}{2}} = \left(T^{n+1}_{M+1,j} \right)^{k+\frac{1}{2}} \qquad 1 \le j \le J .$

3) $\left(T^{n+1}_{6,1} \right)^{k+\frac{1}{2}} = T^{n+1}_{i,1} \qquad j = 1;$ desired i .

4) $\left(T_{m,1}^{n+1}\right)^{k+1} = T_{i,1}^{n+1}$ $\quad 1 \leq m \leq 6$; desired i .

5) $\left(T_{m,0}^{n+1}\right)^{k+1} = \left(T_{m,2}^{n+1}\right)^{k+1}$ $\quad 6 < m \leq M$.

6) $\left(T_{m,J-1}^{n+1}\right)^{k+1} = \left(T_{m,J+1}^{n+1}\right)^{k+1}$ $\quad$ For all m .

APPENDIX C

MATRIX INVERSION TECHNIQUE

As shown in Appendix B all programs have two conduction equations, one implicit in the x direction and explicit in the y direction and the other implicit in the y direction and explicit in the x direction. In case of THERMAB5 we have the y and z directions. These equations are written for each mesh point and must be solved, a line at a time, as dictated by the alternating direction implicit method.

Each line of mesh points leads to a set of simultaneous equations of the form,

$$B_1 T_1 + C_1 T_2 = D_1$$

$$A_r T_{r-1} + B_r T_r + C_r T_{R+1} = D_r$$

$$A_R T_{R-1} + B_R T_R = D_R$$

where r = i, j or m,

which are most conveniently solved by use of a tridiagonal matrix algorithm suggested by Thomas (28). The algorithm is equivalent to a plain Gaussian elimination, but it avoids the error growth associated with the back solution of the elimination method and also minimizes the storage problems in machine computation.

The method may be summarized as follows (25):

Let:

$$w_1 = B_1$$

$$w_r = B_r - A_r\, b_{r-1} \qquad 2 \le r \le R$$

$$b_r = C_r / w_r \qquad 1 \le r \le R-1$$

$$g_1 = D_1 / w_1$$

$$g_r = (D_r - A_r\, g_{r-1}) / w_r \qquad 2 \le r \le R ;$$

The solution is:

$$T_R = g_R$$

$$T_r = g_r - b_r\, T_{r+1} \qquad 1 \le r \le R-1 \ .$$

Thus, w, b, and g are computed in order of increasing r, and T is computed in order of decreasing r.

APPENDIX D

VARIABLE GRID SPACING AND TIME STEPS

Variable Grid Spacing

To minimize error and improve stability considerations, it is desirable to choose Δy_j in equation B-2 so that the temperature difference between any $T_{i,j}$ and $T_{i,j+1}$ can be controlled to be less than a preselected ΔT_{max}. A systematic method of choosing y_j values can be based upon the heat distribution prescribed by the analytical solution of conduction heat transfer in a semi-infinite slab (22).

Consider the one-dimensional transient conduction heat problem described by:

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial y^2}$$

with boundary conditions,

$$\text{at } t = 0, \quad T = T_0, \quad \text{for } y \geq 0$$

$$\text{at } y = 0, \quad T = T_1, \quad \text{for } t \geq 0$$

$$\text{as } y \to \infty, \quad T \to T_0, \quad \text{for } t \geq 0 \ .$$

The solution is

$$\frac{T - T_0}{T_1 - T_0} = \text{erfc}\left(\frac{y}{2\sqrt{\alpha t}}\right) . \tag{D-1}$$

Recalling the general differentiation formula (35),

$$\frac{\partial}{\partial v} \int_0^v f(w)\ dw = f(v) \tag{D-2}$$

and knowing that erfc = 1 - erf, we can write

$$\operatorname{erfc}\left(\frac{y}{2\sqrt{\alpha t}}\right) = 1 - \frac{2}{\sqrt{\pi}} \int_0^{\left(\frac{y}{2\sqrt{\alpha t}}\right)} e^{-\frac{y^2}{4\alpha t}} \, d\left(\frac{y}{2\sqrt{\alpha t}}\right) .$$

Then:

$$\frac{\partial T}{\partial y} = (T_1 - T_0) \frac{\partial T}{\partial \left(\frac{y}{2\sqrt{\alpha t}}\right)} \frac{\partial \left(\frac{y}{2\sqrt{\alpha t}}\right)}{\partial y} ,$$

$$\frac{\partial T}{\partial y} = - (T_1 - T_0) \frac{2}{\sqrt{\pi}} \left(e^{-\frac{y^2}{4\alpha t}}\right) \left(\frac{1}{2\sqrt{\alpha t}}\right) ,$$

$$\frac{\partial T}{\partial y} = - \frac{T_1 - T_0}{\sqrt{\pi \alpha t}} e^{-\frac{y^2}{4\alpha t}} . \qquad \text{(D-3)}$$

To determine the time when this gradient reaches its maximum value at any y we first differentiate the gradient with respect to t.

$$\frac{\partial^2 T}{\partial y \, \partial t} = - \frac{T_1 - T_0}{\sqrt{\pi \alpha}} \left(- 1/2 \; t^{-3/2}\right) e^{-\frac{y^2}{4\alpha t}} -$$

$$\frac{T_1 - T_0}{\sqrt{\pi \alpha t}} \left(- \frac{y^2}{4\alpha}\right) \left(- t^{-2}\right) \left(e^{-\frac{y^2}{4\alpha t}}\right)$$

$$\frac{\partial^2 T}{\partial y \, \partial t} = \frac{T_1 - T_0}{2t\sqrt{\pi \alpha t}} e^{-\frac{y^2}{4\alpha t}} \left(1 - \frac{y^2}{2\alpha t}\right) . \qquad \text{(D-4)}$$

Then setting $\frac{\partial^2 T}{\partial y \, \partial t} = 0$.

$$\frac{T_1 - T_0}{2t\sqrt{\pi \alpha t}} e^{-\frac{y^2}{4\alpha t}} \left(1 - \frac{y^2}{2\alpha t}\right) = 0 ,$$

which results in:

$$t = \frac{y^2}{2\alpha} . \tag{D-5}$$

The temperature gradient at this time is obtained by substituting equation D-5 into equation D-3.

$$\frac{\partial T}{\partial y} = - \frac{\sqrt{2}\ (T_1 - T_0)}{y \sqrt{\pi}\ e^{\frac{1}{2}}} \tag{D-6}$$

which is approximately

$$\frac{\partial T}{\partial y} \approx - \frac{T_1 - T_0}{2y} . \tag{D-7}$$

Then using finite differences, the temperature difference between grid points can be limited to $(\Delta T)_{max}$ by requiring

$$\frac{T_1 - T_0}{2y_j} \leq \frac{(\Delta T)_{max}}{\Delta y_j} \qquad \text{for all } j. \tag{D-8}$$

Then,

$$y_{j+1} - y_j = \Delta y_j \leq \frac{2y_j\ (\Delta T)_{max}}{(T_1 - T_0)} . \tag{D-9}$$

Let

$$\frac{(\Delta T)_{max}}{T_1 - T_0} = Ry .$$

Then $y_{j+1} - y_j \leq 2\ Ry\ y_j$.

If we let $y_{j+1} = y_j\ (1 + 2\ Ry)$, (D-10)

we can begin with some y_P and generate a y_j sequence, such that the temperature difference between points will never exceed a predesignated $(\Delta T)_{max}$.

$$y_j = y_P\ (1 + 2\ Ry)^{j-P} . \tag{D-11}$$

When establishing this sequence, we want

$$\Delta y_{j+1} \geq \Delta y_j \quad \text{for all } j$$

but if P = 2, equation D-7 implies $\Delta y_2 < \Delta y_1 \equiv y_2$ since $y_1 = 0$ and 2 Ry is less than 1.

Therefore we must select the initial Δy_j's until y_j becomes large enough such that:

$$2 \text{ Ry } y_{j-1} \geq y_J - y_{J-1} .$$

We can then set this $y_{j-1} = y_P$ and use equation D-11 to select values of y_j for $j > P$.

The Δy_j and Δz_m distributions for this work were based on this procedure.

Time Step Incrementation

It was necessary to begin with small Δt_n's and systematically increase their values partially because of accuracy but mainly because of stability difficulties. The initial time step incrementation for these programs was based on theory similar to that used for the y_j spacing.

$$\frac{T - T_0}{T_1 - T_0} = 1 - \text{erf}\left(\frac{y}{2\sqrt{\alpha t}}\right) .$$

$$\frac{\partial T}{\partial t} = (T_1 - T_0) \frac{\partial T}{\partial\left(\frac{y}{2\sqrt{\alpha t}}\right)} \frac{\partial\left(\frac{y}{2\sqrt{\alpha t}}\right)}{\partial t} .$$

$$\frac{\partial T}{\partial t} = - (T_1 - T_0) \frac{2}{\sqrt{\pi}} \left(e^{-\frac{y^2}{4\alpha t}}\right) \left(\frac{-y}{4\sqrt{\alpha t}^{3/2}}\right) .$$

$$\frac{\partial T}{\partial t} = \frac{(T_1 - T_0)y}{2\sqrt{\pi\alpha}t^{3/2}} e^{-\frac{y^2}{4\alpha t}} \quad . \tag{D-12}$$

To determine the position where the rate of heating reaches its maximum value at any t, the rate of heating is differentiated with respect to y.

$$\frac{\partial^2 T}{\partial t\, \partial y} = \frac{T_1 - T_0}{2\sqrt{\pi\alpha}t^{3/2}} e^{-\frac{y^2}{4\alpha t}} + \frac{(T_1 - T_0)y}{2\sqrt{\pi\alpha}t^{3/2}} \left(\frac{-2y}{4\alpha t}\right) \left(e^{-\frac{y^2}{4\alpha t}}\right) . \tag{D-13}$$

Setting $\frac{\partial^2 T}{\partial t\, \partial y} = 0$, we have

$$\left(1 - \frac{2y^2}{4\alpha t}\right) \frac{(T_1 - T_0)}{2\sqrt{\pi\alpha}t^{3/2}} e^{-\frac{y^2}{4\alpha t}} = 0,$$

$$\frac{y^2}{2\alpha t} = 1, \text{ or } y = \sqrt{2\alpha t} \quad . \tag{D-14}$$

The temperature gradient at this position is obtained by substituting equation D-14 into D-12.

$$\frac{\partial T}{\partial t} = \frac{(T_1 - T_0)\sqrt{2\alpha t}}{2\sqrt{\pi\alpha}t^{3/2}} e^{-\frac{2\alpha t}{4\alpha t}} .$$

$$\frac{\partial T}{\partial t} = \frac{T_1 - T_0}{r\sqrt{2\pi}} e^{-\frac{1}{2}} . \tag{D-15}$$

Or

$$\frac{\partial T}{\partial t} \approx \frac{T_1 - T_0}{4t} . \tag{D-16}$$

The temperature change per time step can be limited to $(\Delta T)_{max}$ by requiring that:

$$\frac{T_1 - T_0}{4\, t_n} \leq \frac{(\Delta T)_{max}}{\Delta t_n} . \tag{D-17}$$

which implies,

$$t_{n+1} - t_n \leq \frac{4 t_n (\Delta T)_{max}}{(T_1 - T_0)} = 4 R_t t_n$$

where $R_t = \frac{(\Delta T)_{max}}{T_1 - T_0}$.

Then if $t_{n+1} = t_n (1+4 R_t)$, we can begin with a t_P and generate a T_n sequence, such that the temperature change per time step nowhere exceeds $(\Delta T)_{max}$.

$$t_n = T_P (1+4 R_t)^{n-P} . \qquad \text{(D-18)}$$

An initial time step Δt_0 was used until $4 R_t t_n \geq \Delta t_0$, then this t_n was denoted t_P and equation D-18 was used to calculate t_n for $n > P$ until stability of the system imposed a limit on the value of Δt_n.

The Δt_0 for these programs was one hour and R_t was selected to be 0.05. With this criteria t_P was five hours. It was necessary to discontinue the use of this relationship when Δt_n had reached 15 hours because of stability limitations. A Δt_n of 15 hours was maintained until t_n reached 1000 hours. At 1000 hours, Δt_n was raised to 30 hours; at 3000 hours, it was raised to 50 hours; and at 10,000 hours, it was raised to 100 hours.

APPENDIX E

EXAMPLE OUTPUT DATA

Example output data from THERMAB3—Run no. 3

J=	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	1500.0	1465.9	1432.1	1365.5	1300.2	1236.3	1142.2	1050.7	936.2	833.7	714.8	602.8	486.0	391.0	297.8	223.3	164.4	124.0	100.0	100.0	100.0
2	1482.9	1450.7	1418.7	1354.8	1291.5	1229.1	1136.7	1046.2	932.9	831.3	713.1	601.7	485.3	390.6	297.6	223.2	164.3	123.9	100.0	100.0	100.0
3	1466.0	1434.5	1403.1	1340.4	1278.3	1217.0	1126.0	1036.9	925.2	825.2	708.7	598.7	483.3	389.3	296.8	222.7	164.1	123.8	100.0	100.0	100.0
4	1449.4	1418.2	1387.1	1325.1	1263.7	1203.1	1113.0	1025.0	915.1	816.9	702.4	594.1	480.3	387.3	295.6	222.0	163.7	123.6	100.0	100.0	100.0
5	1432.8	1401.8	1370.9	1309.4	1248.5	1188.3	1098.9	1011.8	903.6	807.2	694.9	588.5	476.5	384.7	294.0	221.0	163.2	123.4	100.0	100.0	100.0
6	1416.3	1385.5	1354.7	1293.5	1233.0	1173.0	1084.1	998.0	891.4	796.7	686.7	582.3	472.2	381.7	292.1	219.9	162.5	123.1	100.0	100.0	100.0
7	1399.8	1369.1	1338.4	1277.5	1217.3	1157.6	1069.1	983.8	878.9	785.9	678.0	575.6	467.4	378.3	289.9	218.5	161.7	122.7	100.0	100.0	100.0
8	1383.4	1352.7	1322.1	1261.4	1201.4	1141.9	1053.9	969.5	866.2	774.9	669.2	568.8	462.5	374.8	287.6	217.1	160.9	122.4	100.0	100.0	100.0
9	1367.0	1336.4	1305.9	1245.4	1185.6	1126.2	1038.6	955.2	853.4	763.9	660.2	561.7	457.3	371.1	285.1	215.5	160.0	121.9	100.0	100.0	100.0
10	1350.6	1320.1	1289.7	1229.4	1169.7	1110.4	1023.4	940.9	840.8	752.9	651.3	554.7	452.1	367.3	282.6	213.9	159.0	121.5	100.0	100.0	100.0
11	1334.2	1303.8	1273.5	1213.4	1153.8	1094.6	1008.3	926.8	828.3	742.0	642.4	547.7	446.9	363.4	280.0	212.2	158.0	121.0	100.0	100.0	100.0
12	1317.9	1287.5	1257.3	1197.3	1137.9	1078.9	993.3	912.8	815.9	731.3	633.6	540.7	441.7	359.5	277.3	210.4	156.9	120.5	100.0	100.0	100.0
13	1301.6	1271.3	1241.2	1181.3	1121.9	1063.2	978.3	899.0	803.7	720.7	624.9	533.7	436.5	355.7	274.7	208.7	155.8	119.9	100.0	100.0	100.0
14	1285.3	1255.2	1225.1	1165.3	1106.0	1047.6	963.5	885.3	791.6	710.2	616.3	526.9	431.3	351.8	272.1	206.9	154.7	119.4	100.0	100.0	100.0
15	1269.1	1239.0	1209.0	1149.3	1090.1	1032.1	948.9	871.8	779.8	700.0	607.9	520.1	426.2	347.9	269.4	205.1	153.6	118.8	100.0	100.0	100.0
16	1252.9	1222.9	1193.0	1133.4	1074.3	1016.7	934.5	858.5	768.1	689.8	599.5	513.4	421.1	344.1	266.8	203.3	152.5	118.3	100.0	100.0	100.0
17	1236.7	1206.8	1176.9	1117.4	1058.6	1001.5	920.2	845.4	756.7	679.9	591.3	506.8	416.0	340.3	264.2	201.5	151.3	117.7	100.0	100.0	100.0
18	1220.6	1190.8	1160.9	1101.4	1043.0	986.3	906.1	832.5	745.4	670.1	583.3	500.2	411.1	336.5	261.6	199.7	150.1	117.1	100.0	100.0	100.0
19	1204.5	1174.7	1144.8	1085.5	1027.4	971.3	892.2	819.8	734.3	660.5	575.3	493.8	406.1	332.7	259.0	197.8	149.0	116.5	100.0	100.0	100.0
20	1188.5	1158.7	1128.8	1069.7	1012.0	956.5	878.5	807.3	723.4	651.1	567.5	487.4	401.2	329.0	256.3	196.0	147.8	115.9	100.0	100.0	100.0
21	1172.5	1142.7	1112.9	1054.0	996.8	942.0	865.0	795.1	712.8	641.8	559.8	481.2	396.4	325.3	253.7	194.2	146.7	115.3	100.0	100.0	100.0
22	1156.6	1126.8	1097.0	1038.5	981.8	927.6	851.8	783.0	702.3	632.6	552.2	475.0	391.5	321.5	251.1	192.4	145.5	114.6	100.0	100.0	100.0
23	1140.7	1111.0	1081.3	1023.1	966.9	913.4	838.7	771.2	691.9	623.6	544.7	468.8	386.8	317.9	248.5	190.6	144.4	114.0	100.0	100.0	100.0
24	1124.9	1095.2	1065.7	1007.9	952.3	899.4	825.9	759.6	681.8	614.8	537.3	462.7	382.0	314.2	245.8	188.8	143.2	113.3	100.0	100.0	100.0
25	1109.1	1079.6	1050.1	992.7	937.7	885.6	813.3	748.1	671.8	606.0	529.9	456.7	377.3	310.5	243.2	187.0	142.1	112.6	100.0	100.0	100.0
26	1093.3	1063.9	1034.7	977.8	923.4	871.9	800.8	736.8	661.9	597.4	522.7	450.7	372.6	306.9	240.6	185.2	140.9	111.9	100.0	100.0	100.0
27	1077.6	1048.4	1019.3	962.9	909.2	858.5	788.5	725.7	652.2	588.9	515.5	444.7	368.0	303.3	238.0	183.4	139.8	111.3	100.0	100.0	100.0
28	1062.0	1032.9	1004.0	948.2	895.1	845.2	776.4	714.7	642.6	580.4	508.4	438.9	363.3	299.6	235.4	181.6	138.6	110.6	100.0	100.0	100.0
29	1046.4	1017.5	988.8	933.6	881.2	832.0	764.4	703.8	633.1	572.1	501.3	433.0	358.7	296.0	232.8	179.8	137.5	109.8	100.0	100.0	100.0
30	1030.8	1002.1	973.7	919.1	867.4	819.0	752.5	693.1	623.7	563.8	494.4	427.2	354.1	292.4	230.2	178.0	136.4	109.1	100.0	100.0	100.0
31	1015.3	986.9	958.7	904.7	853.8	806.1	740.8	682.5	614.4	555.6	487.4	421.4	349.5	288.9	227.6	176.3	135.2	108.4	100.0	100.0	100.0
32	999.9	971.6	943.8	890.4	840.2	793.4	729.2	672.0	605.1	547.5	480.6	415.7	345.0	285.3	225.0	174.5	134.1	107.6	100.0	100.0	100.0
33	984.4	956.5	928.9	876.2	826.8	780.7	717.8	661.6	596.0	539.5	473.7	410.0	340.5	281.7	222.4	172.7	133.0	106.8	100.0	100.0	100.0
34	969.1	941.4	914.1	862.2	813.5	768.2	706.4	651.3	587.0	531.5	466.9	404.3	335.9	278.2	219.9	170.9	131.9	106.0	100.0	100.0	100.0
35	953.7	926.4	899.4	848.2	800.3	755.8	695.2	641.1	578.0	523.6	460.2	398.6	331.4	274.6	217.3	169.1	130.8	105.1	100.0	100.0	100.0
36	938.4	911.4	884.8	834.3	787.3	743.5	684.0	631.0	569.1	515.7	453.4	393.0	327.0	271.1	214.7	167.4	129.7	104.3	100.0	100.0	100.0
37	923.2	896.5	870.3	820.6	774.3	731.4	673.0	621.0	560.3	507.9	446.7	387.4	322.5	267.6	212.1	165.6	128.7	103.4	100.0	100.0	100.0
38	908.0	881.7	855.8	806.9	761.4	719.3	662.0	611.0	551.5	500.1	440.1	381.8	318.1	264.1	209.6	163.8	127.7	102.5	100.0	100.0	100.0
39	892.8	866.9	841.4	793.3	748.6	707.3	651.1	601.1	542.8	492.4	433.5	376.3	313.7	260.7	207.0	162.1	126.7	101.6	100.0	100.0	100.0
40	877.7	852.2	827.1	779.8	736.0	695.4	640.4	591.4	534.2	484.7	426.9	370.7	309.3	257.3	204.5	160.3	125.8	100.6	100.0	100.0	100.0
41	862.7	837.5	812.8	766.4	723.4	683.6	629.7	581.7	525.6	477.1	420.4	365.3	305.0	253.9	202.0	158.6	124.9	100.0	100.0	100.0	100.0
42	847.7	822.9	798.7	753.1	710.9	672.0	619.1	572.1	517.1	469.6	414.0	359.9	300.7	250.7	199.6	156.9	124.1	100.0	100.0	100.0	100.0
43	832.7	808.4	784.6	739.9	698.5	660.4	608.6	562.6	508.8	462.2	407.7	354.6	296.5	247.4	197.2	155.3	123.3	100.0	100.0	100.0	100.0
44	817.8	794.0	770.6	726.8	686.3	649.0	598.3	553.2	500.6	454.9	401.5	349.4	292.4	244.3	194.9	153.8	122.5	100.0	100.0	100.0	100.0
45	803.0	779.7	756.8	713.8	674.2	637.7	588.2	544.1	492.5	447.8	395.4	344.4	288.5	241.2	192.7	152.3	121.8	100.0	100.0	100.0	100.0
46	788.3	765.5	743.1	701.1	662.4	626.7	578.3	535.2	484.7	441.0	389.7	339.7	284.8	238.4	190.7	151.0	121.1	100.0	100.0	100.0	100.0
47	773.7	751.4	729.5	688.5	650.8	616.0	568.7	526.6	477.4	434.6	384.4	335.3	281.5	235.9	189.0	149.9	120.6	100.0	100.0	100.0	100.0
48	759.3	737.6	716.2	676.3	639.6	605.7	559.7	518.7	470.6	428.7	379.6	331.5	278.7	233.8	187.5	149.0	120.1	100.0	100.0	100.0	100.0
49	745.1	724.1	703.4	664.7	629.1	596.2	551.5	511.6	464.7	423.8	375.7	328.5	276.4	232.2	186.4	148.3	119.7	100.0	100.0	100.0	100.0
50	731.3	711.1	691.3	654.2	619.9	588.2	544.9	506.1	460.4	420.3	373.0	326.4	275.0	231.2	185.7	147.8	119.5	100.0	100.0	100.0	100.0
51	718.5	700.1	681.7	646.8	614.2	583.7	541.6	503.6	458.5	418.9	372.0	325.7	274.5	230.8	185.5	147.7	119.4	100.0	100.0	100.0	100.0

88117.0 HOURS
0.04894 DISCREPANCY IN HEAT BALANCE
0.70402099E 09 BTU OF HEAT STORED IN RESERVOIR
0.24409128E 09 BTU OF HEAT PRODUCED
0.88157645E 09 BTU OF HEAT INJECTED
0.13594491E 10 HTTOT
0.41133696E 09 HTORG
100.0TIME INCREMENT, HOURS
0.05146 ER1
-0.61596915E-01 ER2
0.01000 TOLERANCE
0.58262092E 00 =INSTANT HEAT STORED

Example output data from THERMAB3—Run no. 3

DISTANCE DOWN FRACTURE FEET	ISOTHERM DEGREES F 300.0 FEET INTO WALL	ISOTHERM DEGREES F 500.0 FEET INTO WALL	ISOTHERM DEGREES F 600.0 FEET INTO WALL	ISOTHERM DEGREES F 700.0 FEET INTO WALL	ISOTHERM DEGREES F 900.0 FEET INTO WALL	ISOTHERM DEGREES F 1100.0 FEET INTO WALL	ISOTHERM DEGREES F 1300.0 FEET INTO WALL	ISOTHERM DEGREEES F 1500.0 FEET INTO WALL
0.	66.68	41.92	34.22	28.36	19.41	12.38	6.01	0.0
10.	66.64	41.86	34.13	28.27	19.29	12.22	5.73	0.0
20.	66.52	41.70	33.92	28.02	19.01	11.87	5.30	0.0
30.	66.33	41.44	33.65	27.65	18.62	11.44	4.82	0.0
40.	66.07	41.11	33.30	27.25	18.15	10.96	4.31	0.0
50.	65.76	40.73	32.90	26.83	17.68	10.46	3.79	0.0
60.	65.40	40.29	32.45	26.38	17.19	9.95	3.26	0.0
70.	65.01	39.82	31.98	25.90	16.69	9.43	2.73	0.0
80.	64.58	39.32	31.47	25.39	16.17	8.90	2.20	0.0
90.	64.12	38.80	30.95	24.86	15.64	8.36	1.66	0.0
100.	63.64	38.26	30.41	24.32	15.09	7.82	1.13	0.0
110.	63.14	37.70	29.85	23.76	14.53	7.28	0.59	0.0
120.	62.62	37.12	29.27	23.19	13.96	6.75	0.05	0.0
130.	62.09	36.53	28.69	22.60	13.44	6.20	0.0	0.0
140.	61.55	35.92	28.08	22.00	12.90	5.67	0.0	0.0
150.	60.99	35.31	27.47	21.48	12.36	5.13	0.0	0.0
160.	60.41	34.67	26.96	20.95	11.81	4.59	0.0	0.0
170.	59.83	34.03	26.44	20.41	11.25	4.05	0.0	0.0
180.	59.21	33.51	25.91	19.86	10.70	3.51	0.0	0.0
190.	58.59	32.98	25.36	19.30	10.17	2.97	0.0	0.0
200.	57.94	32.44	24.80	18.72	9.64	2.44	0.0	0.0
210.	57.28	31.89	24.23	18.13	9.09	1.90	0.0	0.0
220.	56.60	31.33	23.65	17.59	8.54	1.37	0.0	0.0
230.	55.91	30.75	23.05	17.06	7.98	0.84	0.0	0.0
240.	55.19	30.16	22.43	16.52	7.45	0.31	0.0	0.0
250.	54.45	29.55	21.84	15.97	6.91	0.0	0.0	0.0
260.	53.70	28.92	21.30	15.40	6.36	0.0	0.0	0.0
270.	52.94	28.28	20.74	14.81	5.82	0.0	0.0	0.0
280.	52.37	27.63	20.17	14.22	5.28	0.0	0.0	0.0
290.	51.77	27.05	19.58	13.65	4.74	0.0	0.0	0.0
300.	51.16	26.49	18.98	13.10	4.18	0.0	0.0	0.0
310.	50.54	25.90	18.36	12.53	3.64	0.0	0.0	0.0
320.	49.89	25.30	17.76	11.95	3.10	0.0	0.0	0.0
330.	49.22	24.68	17.19	11.35	2.54	0.0	0.0	0.0
340.	48.53	24.04	16.61	10.76	1.98	0.0	0.0	0.0
350.	47.83	23.39	16.00	10.19	1.43	0.0	0.0	0.0
360.	47.10	22.71	15.38	9.61	0.87	0.0	0.0	0.0
370.	46.35	22.01	14.74	9.01	0.30	0.0	0.0	0.0
380.	45.58	21.39	14.08	8.39	0.0	0.0	0.0	0.0
390.	44.79	20.76	13.47	7.77	0.0	0.0	0.0	0.0
400.	43.98	20.11	12.85	7.18	0.0	0.0	0.0	0.0
410.	43.14	19.44	12.22	6.56	0.0	0.0	0.0	0.0
420.	42.46	18.75	11.56	5.93	0.0	0.0	0.0	0.0
430.	41.80	18.05	10.90	5.32	0.0	0.0	0.0	0.0
440.	41.15	17.42	10.28	4.70	0.0	0.0	0.0	0.0
450.	40.51	16.79	9.65	4.06	0.0	0.0	0.0	0.0
460.	39.91	16.16	9.01	3.44	0.0	0.0	0.0	0.0
470.	39.37	15.55	8.37	2.81	0.0	0.0	0.0	0.0
480.	38.92	14.99	7.77	2.17	0.0	0.0	0.0	0.0
490.	38.62	14.54	7.25	1.56	0.0	0.0	0.0	0.0
500.	38.52	14.32	6.93	1.00	0.0	0.0	0.0	0.0

Example output data from THERMAB4—Run no. 11

J=	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	2000.0	1937.6	1877.9	1771.8	1688.4	1622.0	1537.5	1423.3	1276.0	1109.4	945.6	809.1	712.1	661.7	648.9	660.3	686.2	717.1	744.9	766.0	780.8	797.1	816.8	828.5	841.3
2	1955.4	1897.2	1842.1	1746.2	1671.4	1610.7	1529.9	1417.6	1271.6	1106.2	943.5	807.9	711.7	661.8	649.4	661.4	687.8	719.4	747.9	769.9	785.5	803.3	825.0	838.3	852.8
3	1911.3	1855.3	1803.1	1715.3	1647.9	1592.1	1514.6	1404.4	1260.9	1098.3	938.1	804.9	710.5	662.0	650.9	664.1	691.9	724.9	754.7	777.8	794.4	813.1	836.3	850.3	865.8
4	1868.3	1814.4	1765.0	1684.2	1622.4	1570.4	1495.7	1387.6	1246.6	1086.9	930.2	800.4	708.7	662.3	652.9	667.8	697.4	732.1	763.4	787.6	804.8	824.5	848.6	863.3	879.4
5	1826.6	1775.4	1728.9	1654.5	1597.5	1548.5	1475.8	1369.3	1230.4	1073.8	921.0	794.9	706.5	662.7	655.3	672.1	703.6	740.2	773.0	798.3	816.2	836.6	861.6	876.7	893.4
6	1785.6	1735.7	1690.8	1620.4	1567.4	1521.6	1452.3	1348.6	1212.6	1059.3	910.7	788.9	704.0	663.0	657.9	676.8	710.4	748.8	783.2	809.6	828.2	849.2	875.0	890.6	907.8
7	1748.6	1703.7	1663.3	1598.9	1547.6	1501.2	1430.9	1328.2	1194.5	1044.4	900.3	782.5	701.4	663.3	660.6	681.7	717.4	757.8	793.8	821.2	840.5	862.2	888.7	904.8	922.5
8	1712.5	1668.8	1629.9	1568.6	1520.4	1476.5	1408.5	1307.9	1176.3	1029.3	889.5	776.1	698.7	663.6	663.4	686.8	724.7	767.1	804.7	833.2	853.1	875.4	902.7	919.3	937.5
9	1677.6	1635.6	1598.4	1540.6	1495.0	1452.9	1386.5	1287.6	1158.2	1014.3	878.7	769.6	696.0	664.0	666.3	692.0	732.1	776.6	815.8	845.4	865.9	888.9	917.0	934.0	952.9
10	1644.2	1604.3	1569.1	1514.4	1470.8	1429.9	1364.8	1267.6	1140.4	999.5	868.2	763.3	693.4	664.4	669.3	697.3	739.7	786.3	827.1	857.8	879.0	902.7	931.6	949.2	968.5
11	1612.5	1574.7	1541.3	1489.2	1447.2	1407.3	1343.4	1247.8	1122.8	985.1	857.9	757.2	690.9	665.0	672.4	702.8	747.5	796.2	838.8	870.7	892.6	916.9	946.6	964.7	984.6
12	1582.4	1546.4	1514.6	1464.7	1424.0	1385.0	1322.1	1228.1	1105.4	971.0	847.9	751.2	688.5	665.6	675.6	708.5	755.5	806.4	850.8	883.9	906.6	931.6	962.1	980.6	1000.9
13	1553.7	1519.4	1489.0	1440.9	1401.3	1363.1	1301.2	1208.8	1088.4	957.2	838.2	745.5	686.3	666.4	679.0	714.3	763.7	816.9	863.2	897.6	921.0	946.7	977.8	996.7	1017.5
14	1526.4	1493.6	1464.4	1418.1	1379.5	1341.9	1280.9	1189.8	1071.5	943.7	828.8	740.0	684.3	667.3	682.5	720.3	772.2	827.8	876.0	911.6	935.8	962.1	993.9	1013.2	1034.3
15	1500.3	1468.8	1440.7	1395.8	1358.0	1321.0	1261.0	1171.1	1054.9	930.6	819.7	734.7	682.4	668.4	686.1	726.5	780.9	839.0	889.3	926.3	951.1	978.1	1010.4	1030.0	1051.3
16	1475.0	1444.4	1417.1	1373.2	1336.0	1299.6	1240.7	1152.4	1038.6	918.0	811.0	729.7	680.7	669.6	690.0	732.9	789.9	850.5	902.7	941.0	966.6	994.1	1027.0	1046.8	1068.5
17	1450.5	1420.8	1394.2	1351.2	1314.6	1278.9	1220.9	1134.2	1022.8	905.9	802.6	724.9	679.2	670.9	694.0	739.5	799.2	862.2	916.4	955.9	982.2	1010.3	1043.8	1063.9	1085.9
18	1426.7	1397.8	1371.8	1329.6	1293.6	1258.4	1201.5	1116.3	1007.4	894.0	794.5	720.2	677.8	672.5	698.2	746.3	808.7	874.2	930.4	971.1	998.0	1026.7	1060.8	1081.2	1103.5
19	1403.5	1375.3	1349.9	1308.4	1273.0	1238.3	1182.3	1098.6	992.2	882.4	786.6	715.8	676.6	674.1	702.5	753.4	818.6	886.6	944.8	986.7	1014.3	1043.5	1078.0	1098.7	1121.3
20	1380.8	1353.2	1328.3	1287.5	1252.6	1218.5	1163.4	1081.0	977.4	871.1	779.0	711.6	675.5	676.0	707.1	760.8	828.8	899.4	959.5	1002.7	1030.8	1060.5	1095.5	1116.5	1139.4
21	1358.6	1331.5	1307.1	1266.9	1232.5	1198.9	1144.7	1063.9	962.9	860.1	771.6	707.6	674.6	678.0	711.8	768.3	839.2	912.5	974.6	1018.9	1047.6	1077.8	1113.3	1134.5	1157.7
22	1336.9	1310.3	1286.3	1246.7	1212.8	1179.8	1126.3	1047.1	948.9	849.5	764.6	703.8	673.9	680.1	716.8	776.2	850.0	925.9	990.0	1035.4	1064.7	1095.4	1131.3	1152.8	1176.3
23	1315.6	1289.6	1266.0	1227.1	1193.7	1161.0	1108.3	1030.8	935.2	839.2	757.7	700.2	673.3	682.5	722.0	784.3	861.1	939.7	1005.8	1052.5	1082.3	1113.4	1149.8	1171.6	1195.3
24	1294.7	1269.1	1245.9	1207.6	1174.6	1142.3	1090.4	1014.6	922.0	829.3	751.2	696.8	672.9	685.0	727.4	792.7	872.7	954.0	1022.1	1069.9	1100.3	1131.9	1168.6	1190.5	1214.5
25	1274.1	1249.0	1226.2	1188.3	1155.7	1123.8	1072.8	998.9	909.1	819.7	744.9	693.6	672.6	687.7	733.0	801.5	884.5	968.6	1038.7	1087.5	1118.5	1150.4	1187.5	1209.7	1234.1
26	1253.9	1229.2	1206.7	1169.3	1137.0	1105.5	1055.6	983.6	896.7	810.5	738.8	690.5	672.6	690.5	738.8	810.5	896.7	983.6	1055.6	1105.5	1136.9	1169.3	1206.7	1229.2	1253.9
27	1234.1	1209.7	1187.5	1150.4	1118.5	1087.5	1038.7	968.6	884.5	801.5	733.0	687.7	672.6	693.5	744.9	819.7	909.1	998.9	1072.8	1123.8	1155.7	1188.3	1226.2	1249.0	1274.1
28	1214.5	1190.5	1168.6	1131.9	1100.3	1069.9	1022.1	954.0	872.7	792.7	727.4	685.0	672.9	696.8	751.2	829.3	922.0	1014.6	1090.4	1142.3	1174.6	1207.6	1245.9	1269.1	1294.7
29	1195.3	1171.6	1149.8	1113.4	1082.3	1052.5	1005.8	939.8	861.1	784.3	722.0	682.5	673.3	700.2	757.7	839.2	935.2	1030.8	1108.3	1161.0	1193.7	1227.1	1266.0	1289.6	1315.6
30	1176.3	1152.8	1131.3	1095.4	1064.7	1035.4	990.0	925.9	850.0	776.2	716.8	680.1	673.9	703.8	764.6	849.5	948.9	1047.1	1126.3	1179.7	1212.8	1246.7	1286.3	1310.3	1336.9
31	1157.7	1134.5	1113.3	1077.8	1047.6	1018.9	974.6	912.5	839.2	768.3	711.8	678.0	674.6	707.6	771.6	860.1	962.9	1063.9	1144.7	1198.9	1232.5	1266.9	1307.1	1331.5	1358.6
32	1139.4	1116.5	1095.5	1060.5	1030.8	1002.7	959.5	899.4	828.8	760.8	707.1	676.0	675.5	711.6	779.0	871.1	977.4	1081.0	1163.4	1218.5	1252.6	1287.5	1328.3	1353.2	1380.8
33	1121.3	1098.7	1078.0	1043.5	1014.3	986.7	944.8	886.6	818.6	753.4	702.5	674.1	676.6	715.8	786.6	882.4	992.2	1098.5	1182.3	1238.3	1273.0	1308.4	1349.9	1375.3	1403.5
34	1103.5	1081.2	1060.7	1026.7	998.0	971.2	930.4	874.2	808.7	746.3	698.2	672.5	677.8	720.2	794.5	894.0	1007.4	1116.3	1201.5	1258.4	1293.6	1329.6	1371.8	1397.8	1426.7
35	1085.9	1063.9	1043.8	1010.3	982.2	955.9	916.4	862.2	799.2	739.5	694.0	670.9	679.2	724.8	802.6	905.9	1022.8	1134.2	1220.9	1278.8	1314.6	1351.2	1394.2	1420.8	1450.5
36	1068.5	1046.8	1027.0	994.1	966.6	941.0	902.7	850.5	789.9	732.8	690.0	669.6	680.7	729.7	811.0	918.0	1038.6	1152.4	1240.7	1299.6	1336.0	1373.2	1417.1	1444.4	1475.0
37	1051.3	1030.0	1010.4	978.1	951.1	926.3	889.3	839.0	780.9	726.5	686.1	668.4	682.4	734.7	819.7	930.6	1054.9	1171.1	1261.0	1321.0	1358.0	1395.8	1440.7	1468.8	1500.3
38	1034.3	1013.2	993.9	962.1	935.8	911.6	876.0	827.8	772.2	720.3	682.5	667.3	684.3	740.0	828.8	943.7	1071.5	1189.8	1280.9	1341.9	1379.5	1418.1	1464.4	1493.6	1526.4
39	1017.5	996.7	977.8	946.7	921.0	897.6	863.2	816.9	763.7	714.3	679.0	666.4	686.3	745.5	838.2	957.2	1088.4	1208.8	1301.2	1363.1	1401.3	1440.9	1489.0	1519.4	1553.8
40	1000.9	980.6	962.0	931.6	906.6	883.9	850.8	806.4	755.5	708.5	675.6	665.6	688.5	751.2	847.9	970.9	1105.4	1228.1	1322.1	1384.9	1424.0	1464.7	1514.6	1546.4	1582.4
41	984.6	964.7	946.6	916.9	892.6	870.7	838.8	796.2	747.5	702.8	672.4	665.0	690.9	757.1	857.9	985.0	1122.7	1247.7	1343.4	1407.3	1447.2	1489.2	1541.3	1574.7	1612.5
42	968.5	949.2	931.6	902.7	879.0	857.8	827.1	786.3	739.7	697.3	669.3	664.4	693.4	763.3	868.2	999.5	1140.3	1267.6	1364.8	1429.9	1470.8	1514.4	1569.1	1604.4	1644.2
43	952.9	934.0	917.0	888.9	865.9	845.4	815.8	776.6	732.1	692.0	666.3	664.0	696.0	769.6	878.7	1014.3	1158.2	1287.6	1386.6	1452.9	1495.0	1540.6	1598.4	1635.6	1677.6
44	937.5	919.3	902.7	875.4	853.1	833.2	804.7	767.1	724.6	686.8	663.4	663.6	698.7	776.1	889.5	1029.3	1176.3	1307.9	1408.6	1476.5	1520.4	1568.6	1629.9	1668.8	1712.5
45	922.5	904.8	888.7	862.2	840.5	821.2	793.8	757.8	717.4	681.7	660.6	663.3	701.4	782.5	900.3	1044.4	1194.5	1328.2	1431.0	1501.3	1547.7	1598.9	1663.3	1703.7	1748.6
46	907.8	890.6	875.0	849.2	828.2	809.6	783.2	748.8	710.4	676.8	657.9	663.0	704.0	788.9	910.8	1059.3	1212.6	1348.6	1452.3	1521.6	1567.5	1620.4	1690.9	1735.7	1785.6
47	893.4	876.7	861.6	836.6	816.2	798.3	773.0	740.2	703.6	672.1	655.3	662.7	706.5	794.9	921.0	1073.8	1230.4	1369.3	1475.8	1548.5	1597.6	1654.5	1729.0	1775.4	1826.7
48	879.4	863.3	848.6	824.5	804.8	787.6	763.4	732.1	697.4	667.8	652.9	662.3	708.7	800.4	930.2	1086.9	1246.6	1387.6	1495.7	1570.4	1622.5	1684.2	1765.0	1814.4	1868.3
49	865.8	850.3	836.3	813.1	794.4	777.8	754.7	724.9	691.9	664.1	650.9	662.0	710.5	804.9	938.1	1098.3	1260.9	1404.5	1514.7	1592.1	1647.9	1715.3	1803.1	1855.3	1911.3
50	852.8	838.3	825.0	803.3	785.5	769.9	747.9	719.4	687.8	661.4	649.4	661.8	711.7	807.9	943.6	1106.3	1271.6	1417.6	1529.9	1610.7	1671.4	1746.2	1842.1	1897.2	1955.4
51	841.3	828.5	816.8	797.1	780.8	766.0	744.8	717.1	686.2	660.3	648.9	661.7	712.1	809.1	945.6	1109.4	1276.0	1423.3	1537.5	1622.0	1688.4	1771.8	1877.9	1937.6	2000.0

109117.0 HOURS

0.10441 DISCREPANCY IN HEAT BALANCE

0.18929052E 10 BTU OF HEAT STORED IN RESERVOIR

0.48211277E 09 BTU OF HEAT PRODUCED

0.21102226E 10 BTU OF HEAT INJECTED

0.25361380E 10 HTTOT

0.16112005E 09 HTORG

100.0TIME INCREMENT, HOURS

0.11658 ER1

1.07495 ER2

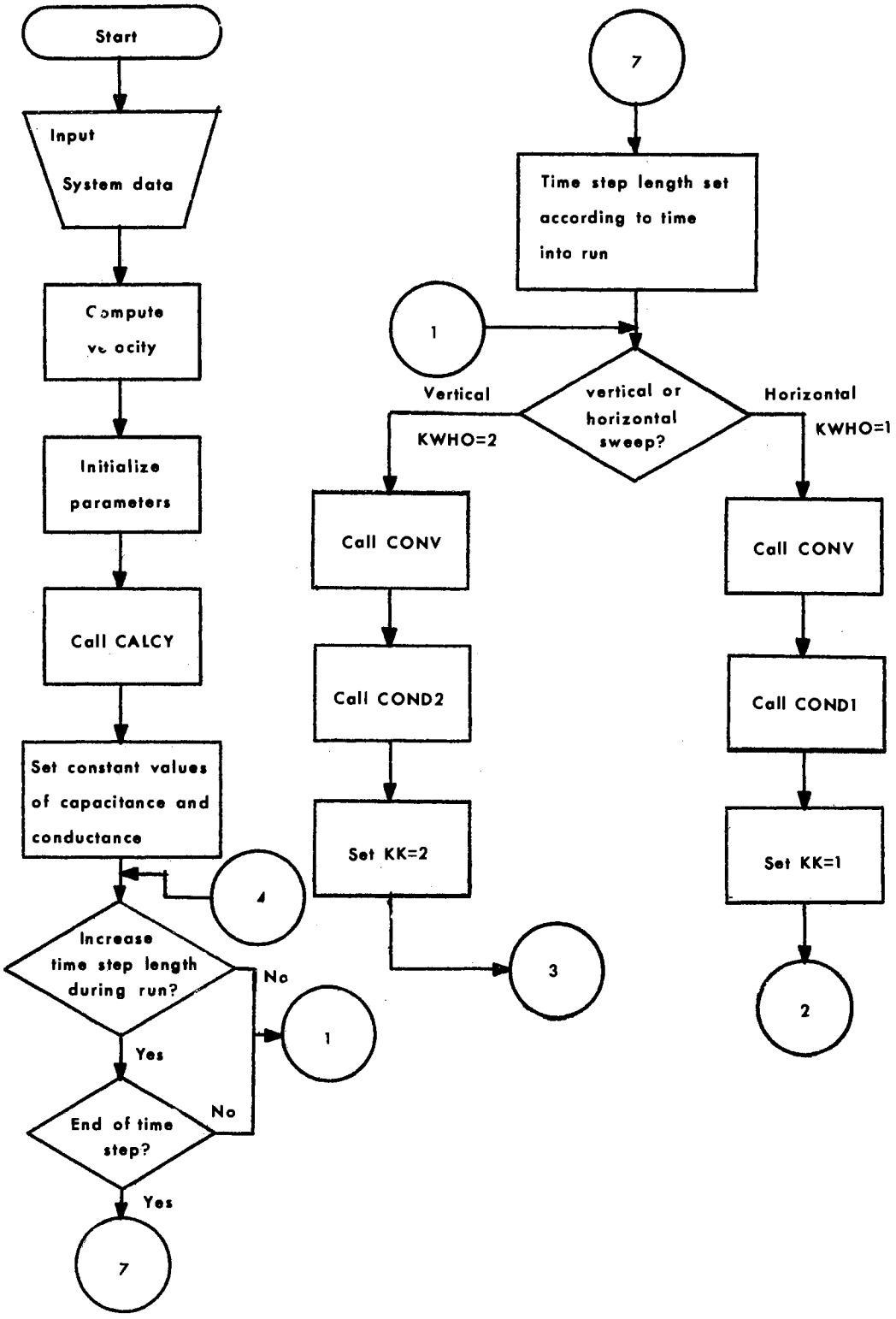
0.01000 TOLERANCE

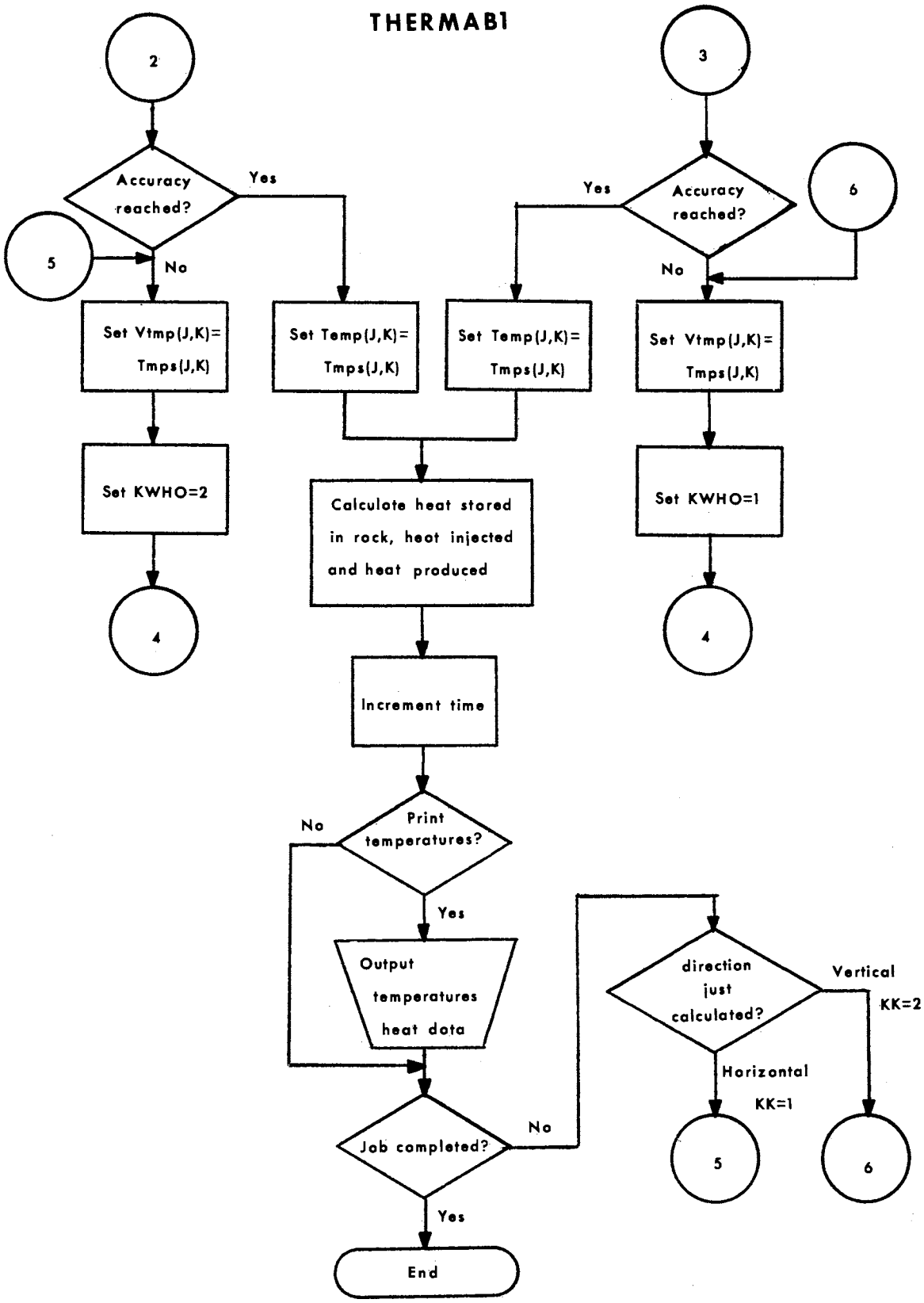
0.65431195E 00 =INSTANT HEAT STORED

APPENDIX F

FLOW SHEET AND PROGRAM LISTINGS

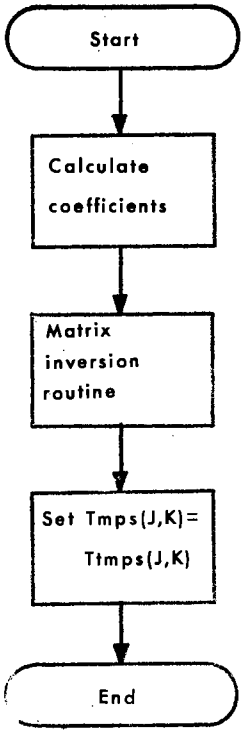
THERMAB1


Start
Input
System data
Compute
velocity
Initialize
parameters
Call CALCY
Set constant values
of capacitance and
conductance
4
Increase
time step length
during run?
No
Yes
End of time
step?
No
Yes
1
7
7
Time step length set
according to time
into run
1
vertical or
horizontal
sweep?
Vertical
KWHO=2
Horizontal
KWHO=1
Call CONV
Call COND2
Set KK=2
3
Call CONV
Call COND1
Set KK=1
2

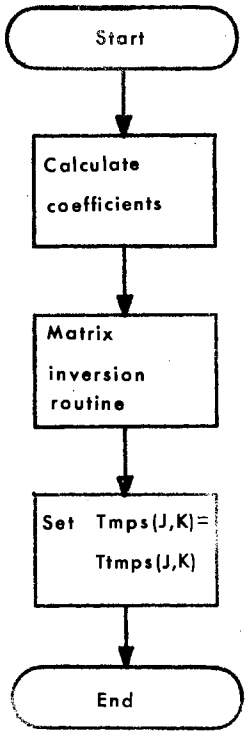

THERMAB1
2
3
Accuracy reached?
Yes
No
5
6
Set Vtmp(J,K)= Tmps(J,K)
Set Temp(J,K)= Tmps(J,K)
Set KWHO=2
Set KWHO=1
4
Calculate heat stored in rock, heat injected and heat produced
Increment time
Print temperatures?
Output temperatures heat data
direction just calculated?
Vertical
KK=2
Horizontal
KK=1
Job completed?
End

THERMAB1

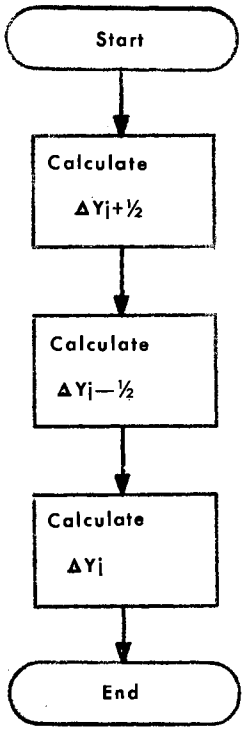
Subroutine COND1 (Horizontal sweep)



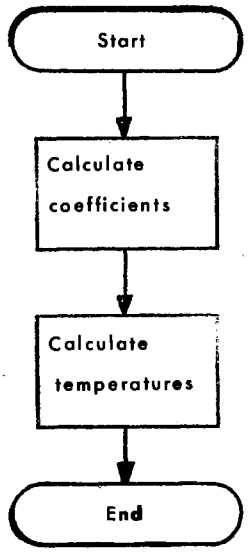
Subroutine COND2 (Vertical sweep)



Subroutine CALCY (Variable grid spacing)



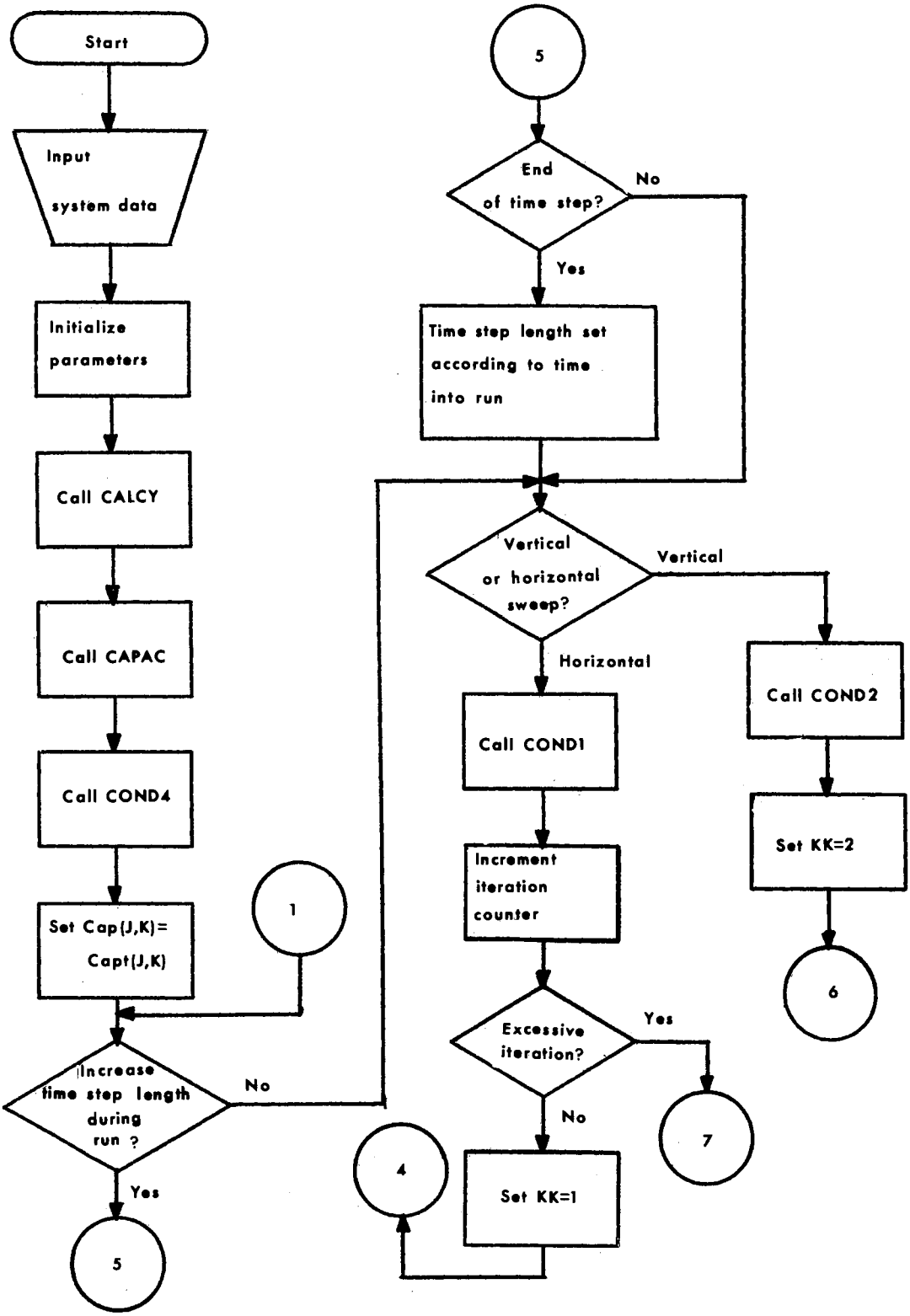
Subroutine CONV

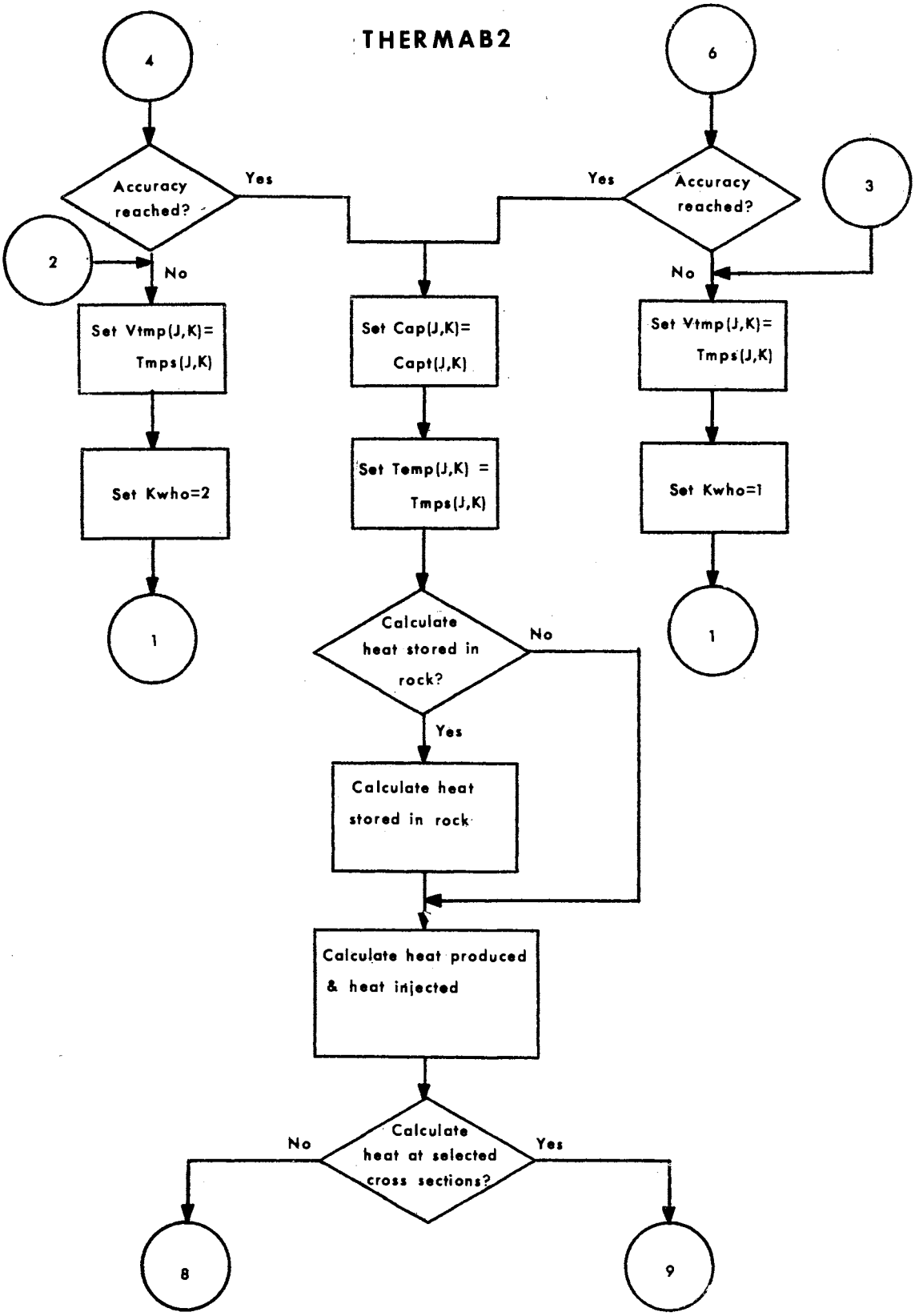


```
C                          THERMAB1-A.L. BARNES
      DIMENSION TMPS(15,51),TEMP(15,51),W(15,51),G(15,51),Y(15),V(51),YP    22241
     1LUS(15),YNEG(15),YMID(15),TTMPS(15,51),AREA(15),H(15),HA(15),VTMP(    22241
     215,51)
      COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH     22241
     1A,DELTX,DENOM,DELTY,TTMPS,CAP,CONDX
C     TEMP=PERMANENT TEMPERATURE ARRAY                                      22241
C     VTMP=TEMPORARY TEMPERATURE ARRAY
C     TMPS=TEMPORARY TEMPERATURE ARRAY                                      22241
  100 READ(5,10)IMAX,JMAX,DELTT,TOL,TIMX,ALPHA,DELTX,DELTY,AA,BB,TT,TTT,    22241
     1CC,KSKIP,NSTRT                                                       22241
C     IMAX=NUMBER OF COLUMNS (K)
C     JMAX=NUMBER OF ROWS (J)
C     DELTT=TIME STEP LENGTH,HOURS
C     TOL=TOLERANCE
C     TIMX=MAXIMUM TIME,HOURS                                               22241
C     ALPHA=SPECIFIC HEAT X VELOCITY X DENSITY(ALL OF GAS)
C     DELTX=DISTANCE BETWEEN POINTS ALONG THE FRACTURE                      22241
C     DELTY=DISTANCE BETWEEN POINTS INTO THE WALL(FIRST 3 POINTS ONLY)
C     AA=CONSTANT IN EQUATION WHICH GENERATES VELOCITY                      22241
C     BB=CONSTANT IN EQUATION WHICH GENERATES VELOCITY                      22241
C     TT=INJECTION TEMPERATURE, DEGREES FAHRENHEIT
C     TTT=RESERVOIR TEMPERATURE, DEGREES FAHRENHEIT
C     CC=CONSTANT IN EQUATION WHICH GENERATES VELOCITY                      22241
C     KSKIP=OPTION TO INCREASE TIME STEP LENGTH DURING RUN
C     NSTRT=SWITCH THAT ALLOWS STARTING RUN WITH CONSTANT OR VARIABLE
C           FIELD TEMPERATURES
      READ(5,20)(Y(I),I=1,15)                                               22241
   20 FORMAT(15F5.0)                                                        22241
C     Y=DISTANCE INTO THE WALL,FT.                                          22241
      PP=0.
      TIME=0.0                                                              22241
      DO 21 K=1,IMAX                                                        22241
      FK=K                                                                  22241
      X=DELTX*FK                                                            22241
      V(K)=AA+BB*X*EXP(CC*X)                                                22241
   21 CONTINUE                                                              22241
      HTOUT=0.                                                              22241
      KWHO=1
      ATMP=0.0
      HTIN=0.                                                               22241
   10 FORMAT(2I5,F5.1,F5.2,F10.1,F10.1,6F5.1,F10.4/2I5)
      IF(NSTRT.EQ.1)GO TO 81
      READ(5,203)TIME,HTIN,HTOUT
  203 FORMAT(F10.0,2E18.8)
C     TIME=TIME, HOURS, ALREADY COMPLETED WHEN RUN RESTARTED
C     HTIN=BTU OF HEAT INJECTED WHEN RUN RESTARTED
C     HTOUT=BTU OF HEAT PRODUCED WHEN RUN RESTARTED
      ATMP=TIME
      DO 74 K=1,IMAX
   74 READ(5,73)(TMPS(J,K),J=1,JMAX)
   73 FORMAT(15F5.0)
      GO TO 204
   81 DO 75 K=1,IMAX
      DO 75 J=1,JMAX
   75 TMPS(J,K)=TTT
  204 DO 205 K=1,IMAX
      DO 205 J=1,JMAX
      TEMP(J,K)=TMPS(J,K)
  205 VTMP(J,K)=TMPS(J,K)
      VTMP(1,1)=TT                                                          22241
      TEMP(1,1)=TT                                                          22241
      TMPS(1,1)=TT
      CALL CALCV                                                            22241
C     CALCV IS A SUBROUTINE THAT CALCULATES VARIABLE Y SPACING VALUES
      CONDX=.0001                                                  THERMAB1
C     CONDX=CONDUCTANCE IN X DIRECTION                             THERMAB1
      COND=0.6
C     COND=CONDUCTANCE IN Y DIRECTION                              THERMAB1
      CAP=35.
   82 GO TO(993,994),KSKIP
  994  IF(ATMP.NE.TIME) GO TO 995
      ATMP=ATMP+1.
  993 IF(TIME.GT.1000.)GO TO 997
      DELTT=5.
      GO TO 995
  997 IF(TIME.GT.3000.)GO TO 992
      DELTT=15.
      GO TO 995
  992 DELTT=50.
C     KWHO IS THE SWITCH TO DETERMINE DIRECTION OF ADIP SWEEP(HORIZONTAL
C          OR VERTICAL)
  995 GO TO(991,53),KWHO
  991 CALL CONV
C     CONV IS A SUBROUTINE THAT SOLVES EXPLICT CONVECTION EQUATIONS
C          ALONG THE FRACTURE
      CALL COND1                                                            22241
C     COND1 IS A SUBROUTINE THAT SOLVES HORIZONTAL ADIP WITH THE CONV
C          SOLUTION AS A BOUNDARY CONDITION
      IF(DENOM.EQ.0.0)GO TO 69                                              22241
      KK=1                                                                  22241
      DO 84 N=1,JMAX
      DO 84 M=1,IMAX                                                        22241
      IF(ABS((VTMP(N,M)-TMPS(N,M))/TMPS(N,M)).GT.TOL)GO TO 86               22241
   84 CONTINUE                                                              22241
      DO 87 J=1,JMAX                                                        22241
      DO 87 K=1,IMAX                                                        22241
   87 TEMP(J,K)=TMPS(J,K)                                                   22241
      GO TO 67                                                              22241
   86 DO 88 J=1,JMAX                                                        22241
      DO 88 K=1,IMAX                                                        22241
   88 VTMP(J,K)=TMPS(J,K)                                                   22241
      KWHO=2
      GO TO 82
   53 CALL CONV                                                             22241
      CALL COND2                                                            22241
C     COND2 IS A SUBROUTINE THAT SOLVES VERTICAL ADIP WITH THE CONV
C          SOLUTION AS A BOUNDARY CONDITION
      IF(DENOM.EQ.0.0)GO TO 69                                              22241
      KK=2                                                                  22241
      DO 64 N=1,JMAX
      DO 64 M=1,IMAX                                                        22241
      IF(ABS((VTMP(N,M)-TMPS(N,M))/TMPS(N,M)).GT.TOL)GO TO 65               22241
   64 CONTINUE                                                              22241
      DO 66 J=1,JMAX                                                        22241
      DO 66 K=1,IMAX                                                        22241
   66 TEMP(J,K)=TMPS(J,K)                                                   22241
      GO TO 67                                                              22241
   65 DO 83 J=1,JMAX                                                        22241
      DO 83 K=1,IMAX                                                        22241
   83 VTMP(J,K)=TMPS(J,K)                                                   22241
      KWHO=1
      GO TO 82                                                              22241
   67 JBMAX=JMAX-2                                                          22241
      IBMAX=IMAX-2                                                          22241
C     THIS ACCOUNTS FOR THE HEAT STORED IN ROCK                             22241
      FIMAX=IMAX                                                            22241
      DO 40 J=1,JMAX                                                        22241
      SUM=0.                                                                22241
      DO 80 K=1,IBMAX,2                                                     22241
   80 SUM=SUM+TEMP(J,K)+4.*TEMP(J,K+1)+TEMP(J,K+2)                          22241
   40 AREA(J)=SUM*(DELTX/3.)                                                22241
      BB2=0.                                                                22241
      DO 90 J=1,JBMAX,2                                                     22241
      H(J)=Y(J+1)-Y(J)                                                      22241
      HA(J)=(Y(J+2)-Y(J+1))/H(J)                                            22241
   90 BB2=BB2+(H(J)*(HA(J)+1.)/(6.*HA(J)))*((2.*HA(J)-1.)*AREA(J+2)+(HA(    22241
     1J)+1.)**2*AREA(J+1)+HA(J)*(2.-HA(J))*AREA(J))                         22241
      HEAT=(CAP*BB2)-(TTT*Y(JMAX)*DELTX*CAP*FIMAX+((TT-TTT)*CAP*DELTX*.2
     15))
C     THIS ACCOUNTS FOR HEAT OUT OF PRODUCER                                22241
      HTOUT=HTOUT+TEMP(1,IMAX)*DELTT*ALPHA                                  22241
      HTTOT=HTOUT+HEAT                                                      22241
      HTIN=HTIN*ALPHA*TT*DELTT                                              22241
      ERR=ABS((HTIN-HTTOT)/HTIN)
   37 FORMAT(I4,15F7.1)                                                     22241
      TIME=TIME+DELTT                                                       22241
      ATMP=TIME
      PP=PP+1.
      IF(PP.NE.2.)GO TO 891
      PP=0.
      WRITE(8,33)
   33 FORMAT(1H1,2X2HJ=5X2H 15X2H 25X2H 35X2H 45X2H 55X2H 65X2H 75X2H 85    22241
     1X2H 95X2H105X2H115X2H125X2H135X2H145X2H15)                            22241
      DO 36 K=1,IMAX                                                        22241
```

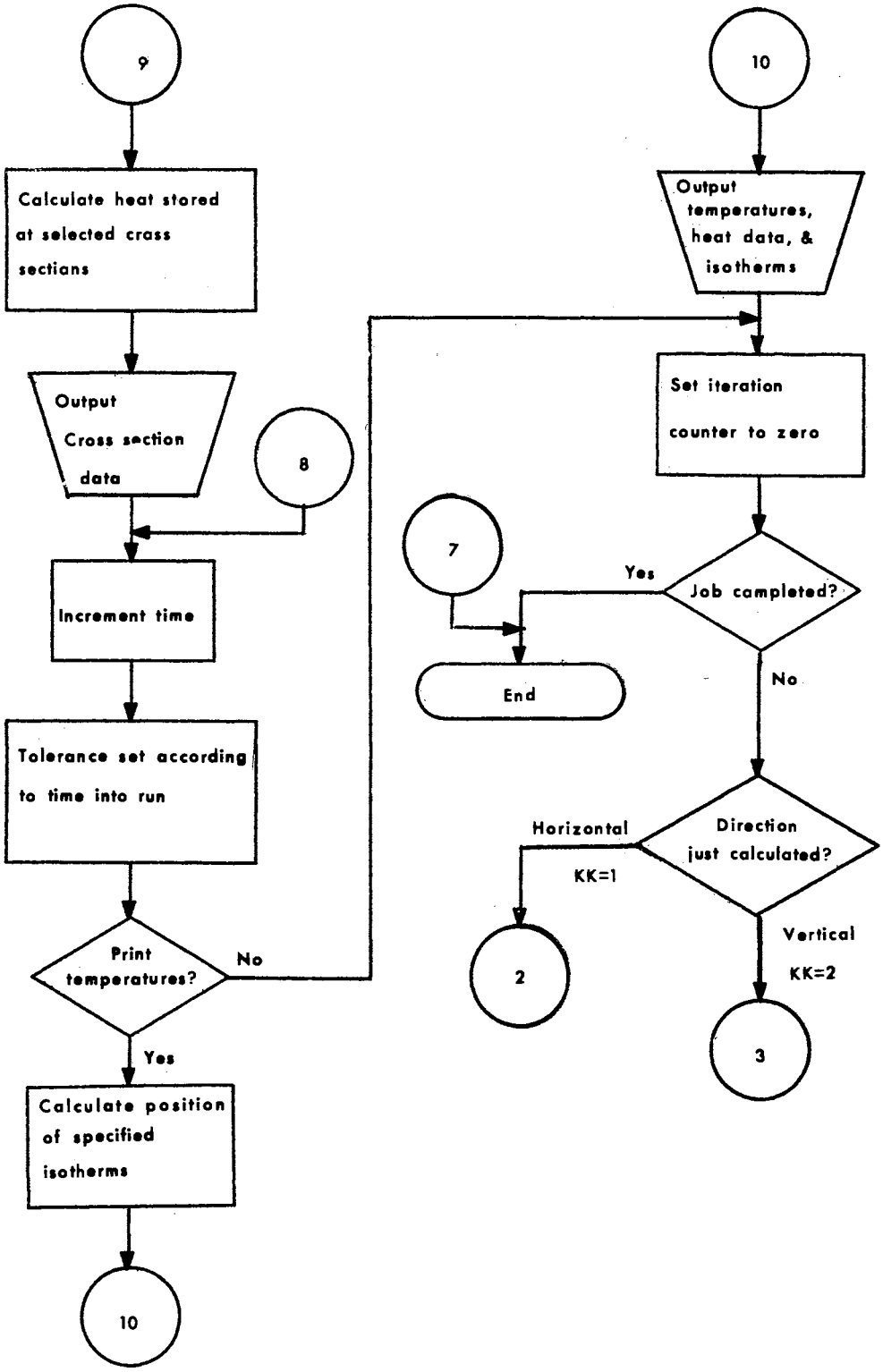
```
     WRITE(8,37)K,(TEMP(J,K),J=1,JMAX)
887  WRITE(8,30)TIME,ERR,HEAT,HTOUT,HTIN
  30 FORMAT(F10.1,2X,5HHOURS/F10.5,2X,27HDISCREPANCY IN HEAT BALANCE/E1   22241
     18.8,2X,31HBTU OF HEAT STORED IN RESERVOIR/E18.8,2X,20HBTU OF HEAT   22241
     2PRODUCED/E18.8,2X,20HBTU OF HEAT INJECTED)                          22241
891  IF(TIME.GT.TIMX)GO TO 100
     GO TO(86,65),KK
  69 STOP                                                                 22241
     END                                                                  22241
     SUBROUTINE CALCV                                                     22241
     DIMENSION TMPS(15,51),TEMP(15,51),W(15,51),G(15,51),Y(15),V(51),YP   22241
     1LUS(15),YNEG(15),YMID(15),TTMPS(15,51),AREA(15),H(15),HA(15),VTMP(  22241
     215,51)
     COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH    22241
     1A,DELTX,DENOM,DELTY,TTMPS,CAP,CONDX
     NMAX=JMAX-1                                                          22241
     DO 2 J=1,NMAX                                                        22241
   2 YPLUS(J)=Y(J+1)-Y(J)                                                 22241
     DO 3 J=2,JMAX                                                        22241
   3 YNEG(J)=Y(J)-Y(J-1)                                                  22241
     YNEG(1)=YPLUS(1)                                                     22241
     YPLUS(JMAX)=YNEG(JMAX)                                               22241
     DO 4 J=1,JMAX                                                        22241
   4 YMID(J)=0.5*(YPLUS(J)+YNEG(J))                                       22241
     RETURN                                                               22241
     END                                                                  22241
     SUBROUTINE CONV                                                      22241
     DIMENSION TMPS(15,51),TEMP(15,51),W(15,51),G(15,51),Y(15),V(51),YP   22241
     1LUS(15),YNEG(15),YMID(15),TTMPS(15,51),AREA(15),H(15),HA(15),VTMP(  22241
     215,51)
     COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH    22241
     1A,DELTX,DENOM,DELTY,TTMPS,CAP,CONDX
     DO 40 K=2,IMAX                                                       22241
     CONMIN=COND
     CONPUS=COND
     D13=DELTX*TEMP(1,K)/(DELTT*V(K))+2.
     D14=(3.*DELTX*CONMIN*TMPS(2,K))/(2.*DELTY*ALPHA)
     D15=(DELTX*CONPUS*(TMPS(2,K)-TMPS(3,K)))/(2.*DELTY*ALPHA)
     D16=1.+(DELTX/(DELTT*V(K)))
     D17=(3.*DELTX*CONMIN)/(2.*DELTY*ALPHA)
     TMPS(1,K)=(TMPS(1,K-1)+D13+D14+D15)/(D16+D17)
     IF(TMPS(1,K).GE.0.0) GO TO 40
     TMPS(1,K)=0.0
  40 CONTINUE                                                             22241
     RETURN                                                               22241
     END                                                                  22241
     SUBROUTINE COND1                                                     22241
     DIMENSION TMPS(15,51),TEMP(15,51),W(15,51),G(15,51),Y(15),V(51),YP   22241
     1LUS(15),YNEG(15),YMID(15),TTMPS(15,51),AREA(15),H(15),HA(15),VTMP(  22241
     215,51)
     COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH    22241
     1A,DELTX,DENOM,DELTY,TTMPS,CAP,CONDX
     B1=-(2.+(DELTX**2/DELTT)*(CAP/CONDX))                                THERMAB1
     DO 1 J=2,JMAX                                                        22241
     DO 20 K=1,IMAX                                                       22241
     D14=-((DELTX**2/YMID(J))*(COND/CONDX))                               THERMAB1
     IF(J.LT.JMAX)GO TO 22                                                22241
     D15=(TMPS(J-1,K)-TMPS(J,K))/YPLUS(J)                                 22241
     GO TO 90                                                             22241
  22 D15=(TMPS(J+1,K)-TMPS(J,K))/YPLUS(J)                                 22241
  90 D16=(TMPS(J,K)-TMPS(J-1,K))/YNEG(J)                                  22241
     D17=(DELTX**2*CAP*TEMP(J,K))/(DELTT*CONDX)
     O1=D14*(D15-D16)-D17                                                 22241
     IF(K.NE.1)GO TO 3                                                    22241
     A1=0.0                                                               22241
     C1=2.0                                                               22241
     W(J,1)=C1/B1                                                         22241
     G(J,1)=D1/B1                                                         22241
     GO TO 20                                                             22241
   3 IF(K.NE.IMAX)GO TO 5                                                 22241
     C1=0.0                                                               22241
     A1=2.0                                                               22241
     GO TO 10                                                             22241
   5 A1=1.0                                                               22241
     C1=1.0                                                               22241
  10 DENOM=B1-A1*W(J,K-1)                                                 22241
     IF(DENOM.NE.0.)GO TO 18                                              22241
     WRITE(8,50)
  50 FORMAT(18H ZERO DENOMINATOR.)                                        22241
     GO TO 800                                                            22241
  18 W(J,K)=C1/DENOM                                                      22241
     G(J,K)=(D1-A1*G(J,K-1))/DENOM                                        22241
  20 CONTINUE                                                             22241
     TTMPS(J,IMAX)=G(J,IMAX)
     DO 30 I=2,IMAX                                                       22241
     II=IMAX+1-I                                                          22241
  30 TTMPS(J,II)=G(J,II)-W(J,II)*TTMPS(J,II+1)
   1 CONTINUE                                                             22241
     DO 95 J=2,JMAX
     DO 95 K=1,IMAX
     TMPS(J,K)=TTMPS(J,K)
  95 CONTINUE
 800 RETURN                                                               22241
     END                                                                  22241
     SUBROUTINE COND2                                                     22241
     DIMENSION TMPS(15,51),TEMP(15,51),W(15,51),G(15,51),Y(15),V(51),YP   22241
     1LUS(15),YNEG(15),YMID(15),TTMPS(15,51),AREA(15),H(15),HA(15),VTMP(  22241
     215,51)
     COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH    22241
     1A,DELTX,DENOM,DELTY,TTMPS,CAP,CONDX
     DO 1 K=1,IMAX                                                        22240
     DO 20 J=2,JMAX                                                       22240
     A1=YPLUS(J)/YNEG(J)                                                  22241
     B1=-(1.0+YPLUS(J)/YNEG(J)+(CAP*YMID(J)*YPLUS(J))/(DELTT*COND))       22241
     O14=-((YMID(J)*YPLUS(J)/DELTX**2)*(CONDX/COND))                      THERMAB1
     IF(K.LT.IMAX) GO TO 24
     D15=TMPS(J,K-1)-TMPS(J,K)
     GO TO 25
  24 D15=TMPS(J,K+1)-TMPS(J,K)
  25 IF(K.GT.1) GO TO 26
     D16=TMPS(J,K)-TMPS(J,K+1)
     GO TO 27
  26 D16=TMPS(J,K)-TMPS(J,K-1)                                            22241
  27 D17=(CAP*YMID(J)*YPLUS(J)*TEMP(J,K))/(DELTT*COND)                    22241
     D1=D14*(D15-D16)-D17                                                 22241
     IF(J.NE.2)GO TO 3                                                    22241
     C1=1.0                                                               22241
     W(2,K)=C1/B1                                                         22241
     G(2,K)=(D1-A1*TMPS(1,K))/B1                                          22241
     GO TO 20                                                             22241
   3 IF(J.NE.JMAX)GO TO 10                                                22241
     A1=A1+C1                                                             22241
     C1=0.0                                                               22241
  10 DENOM=B1-A1*W(J-1,K)                                                 22241
     IF(DENOM.NE.0.)GO TO 18                                              22241
     WRITE(8,50)
  50 FORMAT(18H ZERO DENOMINATOR.)                                        22241
  18 W(J,K)=C1/DENOM                                                      22241
     G(J,K)=(D1-A1*G(J-1,K))/DENOM                                        22241
  20 CONTINUE                                                             22241
     TTMPS(JMAX,K)=G(JMAX,K)
     NMAX=JMAX-2                                                          22241
     DO 30 I=1,NMAX                                                       22241
     II=JMAX-I                                                            22241
  30 TTMPS(II,K)=G(II,K)-W(II,K)*TTMPS(II+1,K)
   1 CONTINUE                                                             22241
     DO 95 J=2,JMAX
     DO 95 K=1,IMAX
     TMPS(J,K)=TTMPS(J,K)
  95 CONTINUE
 800 RETURN                                                               22241
     END
```

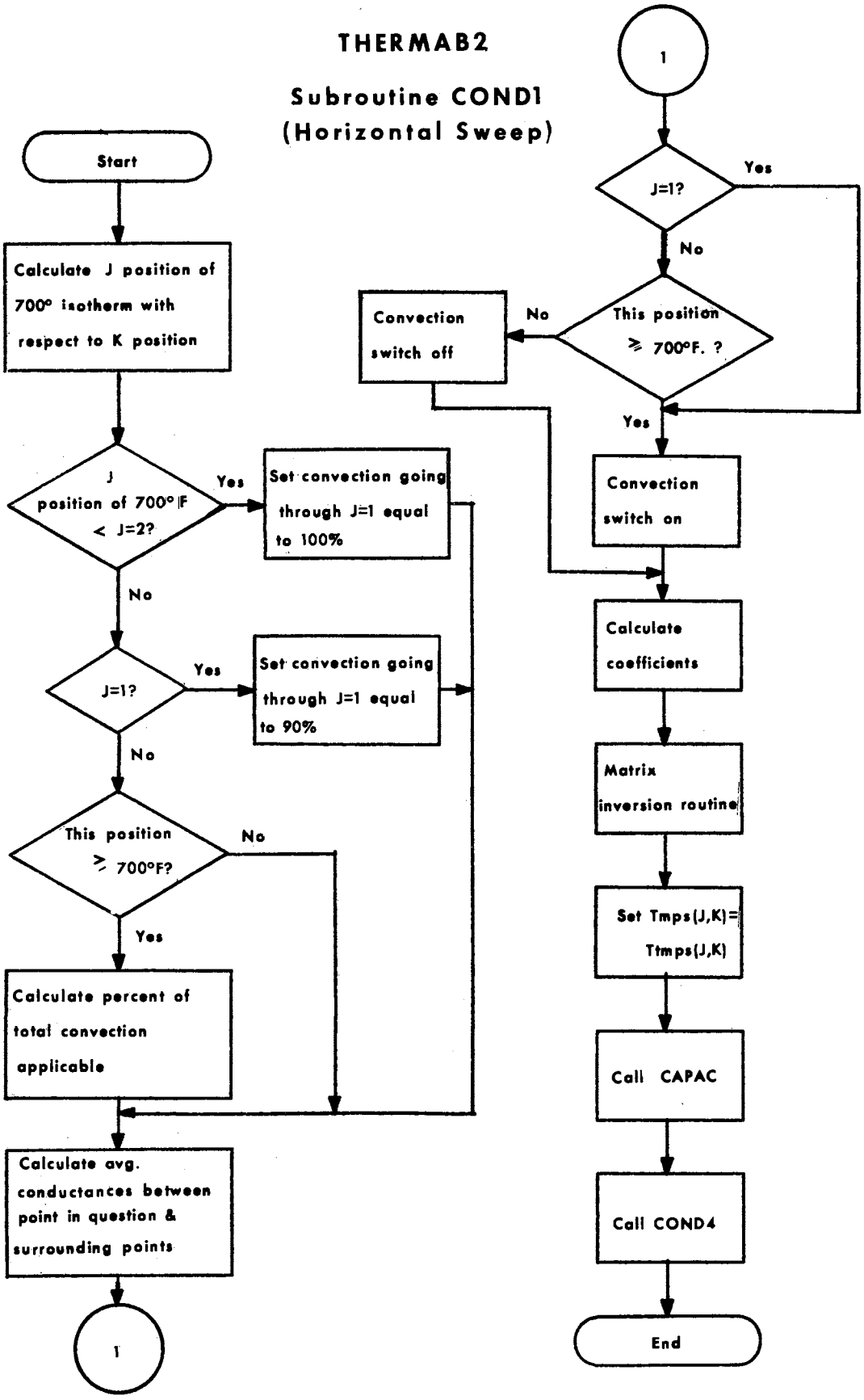
THERMAB2

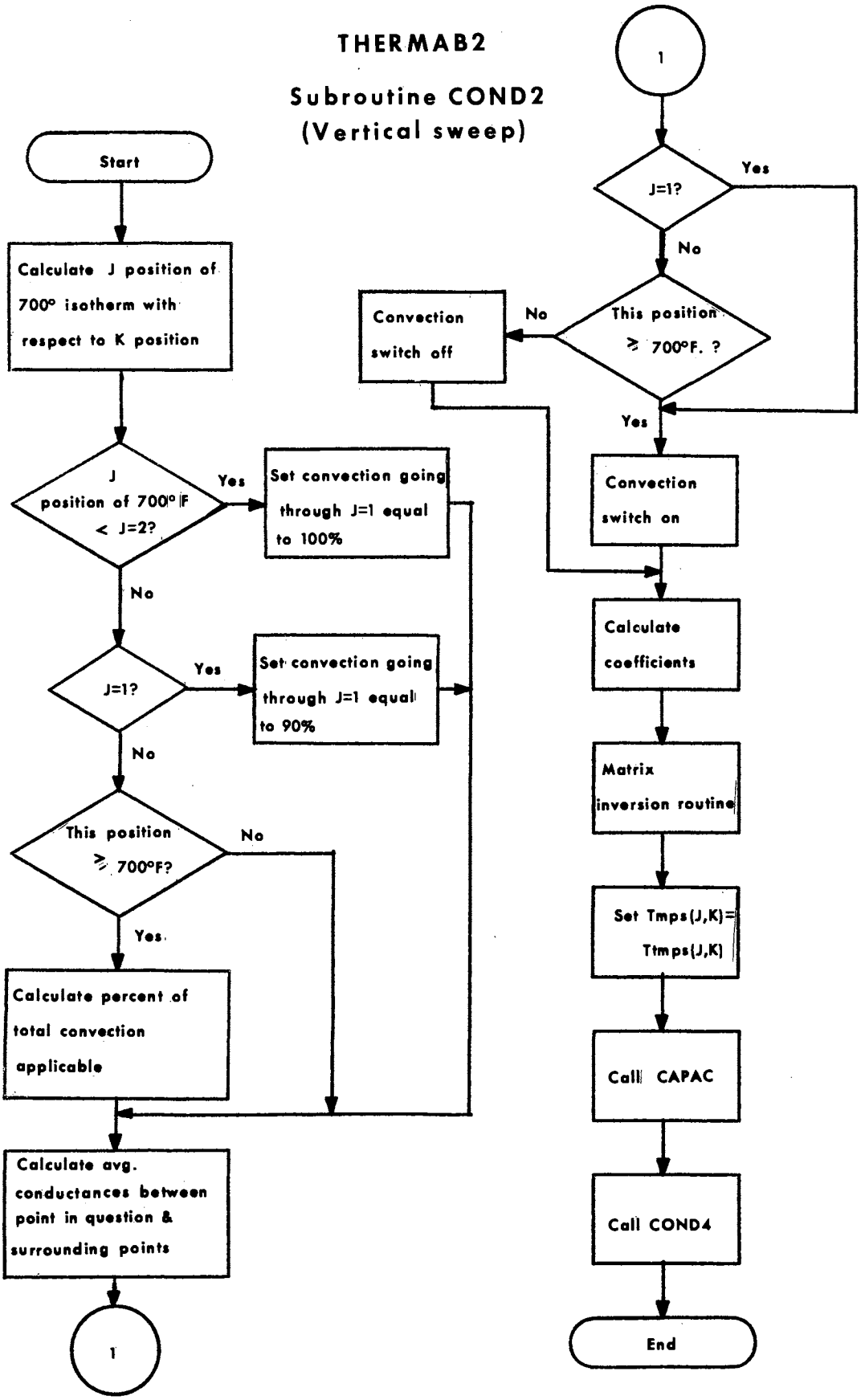

Start
Input system data
Initialize parameters
Call CALCY
Call CAPAC
Call COND4
Set Cap(J,K)= Capt(J,K)
1
Increase time step length during run ?
No
Yes
5
5
End of time step?
No
Yes
Time step length set according to time into run
Vertical or horizontal sweep?
Vertical
Horizontal
Call COND1
Increment iteration counter
Excessive iteration?
Yes
No
Set KK=1
4
7
Call COND2
Set KK=2
6


THERMAB2
4
6
Accuracy reached?
Yes
Yes
Accuracy reached?
3
2
No
No
Set Vtmp(J,K)= Tmps(J,K)
Set Cap(J,K)= Capt(J,K)
Set Vtmp(J,K)= Tmps(J,K)
Set Kwho=2
Set Temp(J,K) = Tmps(J,K)
Set Kwho=1
1
Calculate heat stored in rock?
No
1
Yes
Calculate heat stored in rock
Calculate heat produced & heat injected
Calculate heat at selected cross sections?
No
Yes
8
9

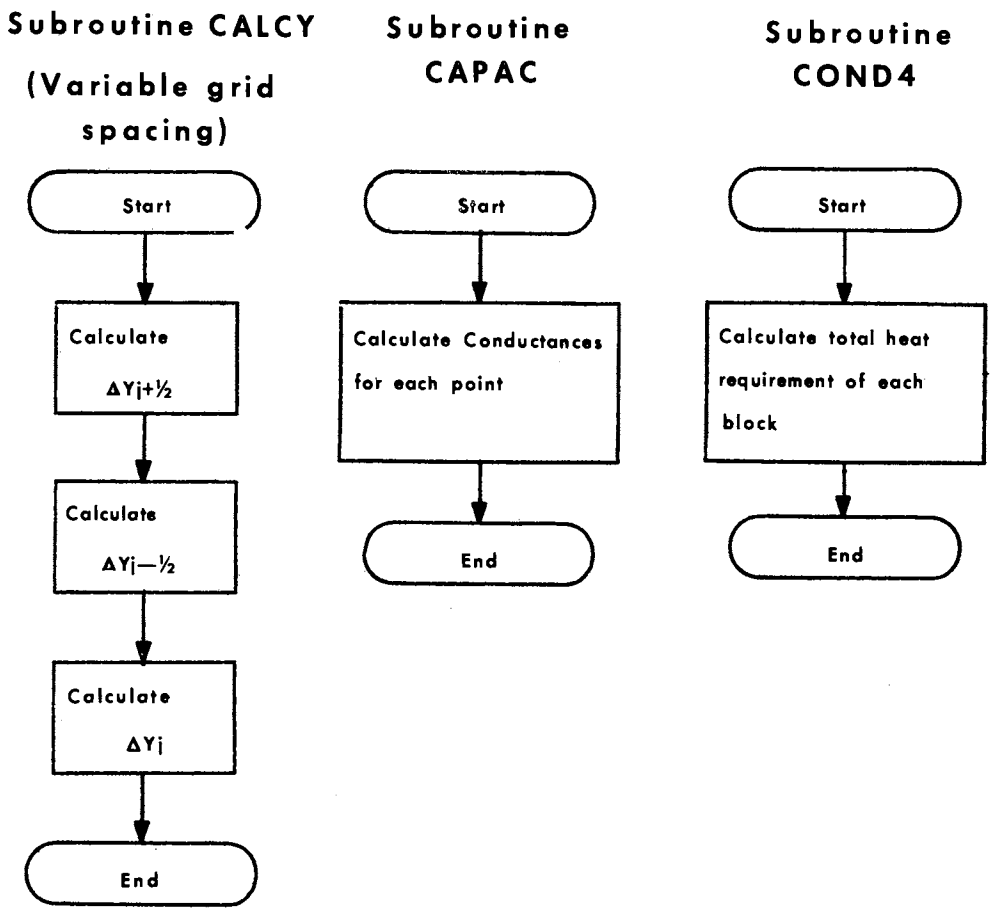
THERMAB2


9
Calculate heat stored at selected crass sectians
Output Cross section data
8
Increment time
Tolerance set according to time into run
Print temperatures?
No
Yes
Calculate position of specified isotherms
10
10
Output temperatures, heat data, & isotherms
Set iteration counter to zero
7
Yes
Job completed?
End
No
Horizontal
KK=1
Direction just calculated?
Vertical
KK=2
2
3


THERMAB2
Subroutine COND1
(Horizontal Sweep)
Start
Calculate J position of 700° isotherm with respect to K position
J position of 700°F < J=2?
Yes
Set convection going through J=1 equal to 100%
No
J=1?
Yes
Set convection going through J=1 equal to 90%
No
This position ≥ 700°F?
No
Yes
Calculate percent of total convection applicable
Calculate avg. conductances between point in question & surrounding points
1
1
J=1?
Yes
No
This position ≥ 700°F. ?
No
Convection switch off
Yes
Convection switch on
Calculate coefficients
Matrix inversion routine
Set Tmps(J,K)= Ttmps(J,K)
Call CAPAC
Call COND4
End


THERMAB2
Subroutine COND2
(Vertical sweep)
Start
Calculate J position of 700° isotherm with respect to K position
J position of 700° F < J=2?
Yes
Set convection going through J=1 equal to 100%
No
J=1?
Yes
Set convection going through J=1 equal to 90%
No
This position ≥ 700°F?
No
Yes
Calculate percent of total convection applicable
Calculate avg. conductances between point in question & surrounding points
1
1
J=1?
Yes
No
This position ≥ 700°F. ?
No
Convection switch off
Yes
Convection switch on
Calculate coefficients
Matrix inversion routine
Set Tmps(J,K)= Ttmps(J,K)
Call CAPAC
Call COND4
End

THERMAB2


Subroutine CALCY
(Variable grid spacing)
Start
Calculate ΔYj+½
Calculate ΔYj—½
Calculate ΔYj
End
Subroutine CAPAC
Start
Calculate Conductances for each point
End
Subroutine COND4
Start
Calculate total heat requirement of each block
End

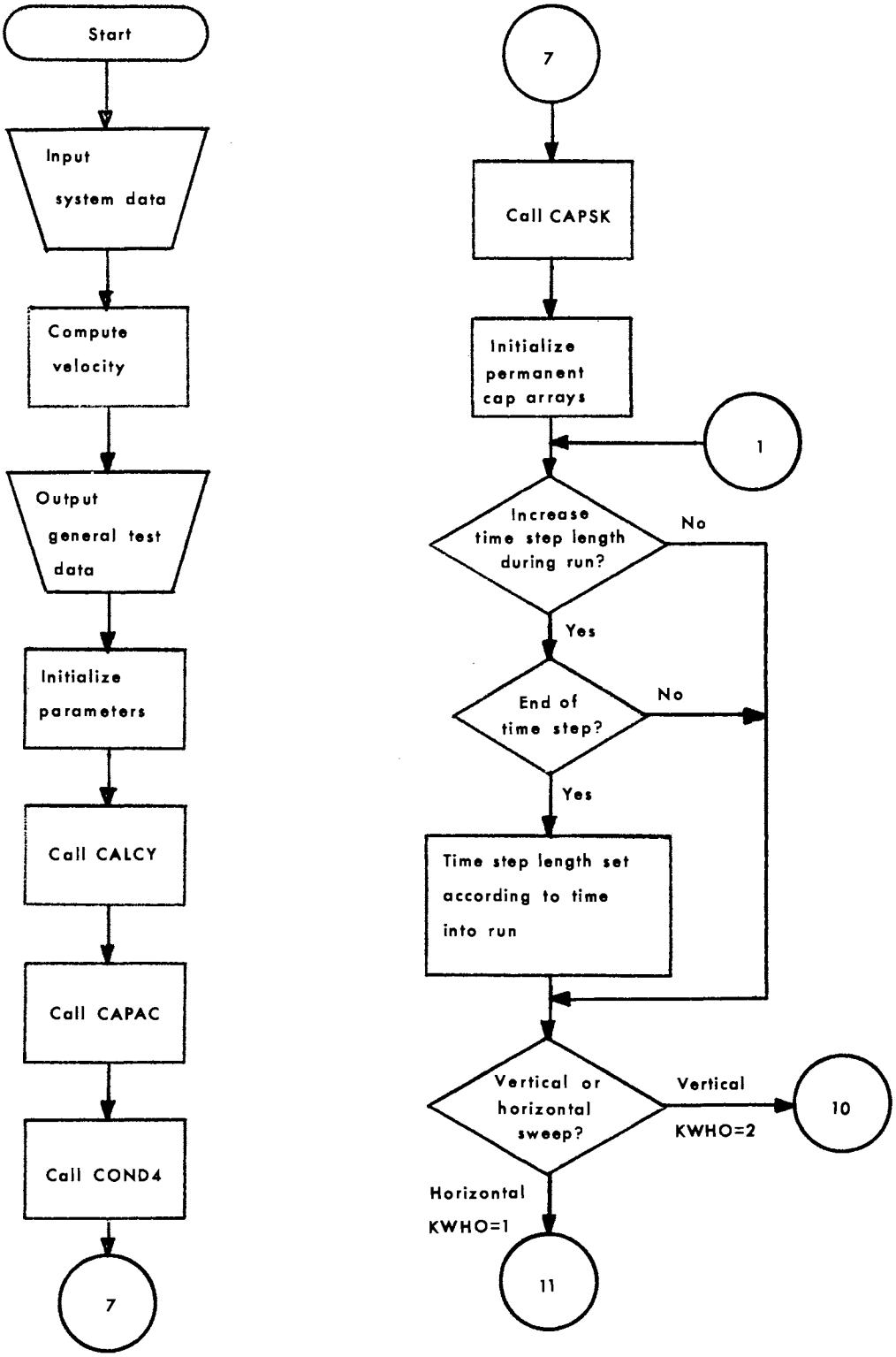
```
C                    THERMAB2-A.L. BARNES
      DIMENSION TMPS(17,51),TEMP(17,51),W(17,51),G(17,51),COND(17,51),Y(  22242
     117),YPLUS(17),YNEG(17),YMID(17),TTMPS(17,51),AREA(17),H(17),HA(17)
     2,VTMP(17,51),CAPT(17,51),CAP(17,51),IS(10),SLTMP(10),DIST(8,51)
      COMMON TMPS,TEMP,W,G,Y,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPHA,  22242
     1DELTX,DENOM,PO,P1,P2,P3,TTMPS,CAPT,CAP,TIME
C     TEMP=PERMANENT TEMPERATURE ARRAY                                    22242
C     TMPS=TEMPORARY TEMPERATURE ARRAY                                    22241
C     VTMP=TEMPORARY TEMPERATURE ARRAY
  105 READ(5,10)IMAX,JMAX,DELTT,TTT,TIMX,ALPHA,DELTX,DELTY,TT,KSKIP,TOL,
     1NRUN,KSLCE,ISSN,LMAX,(IS(I),I=1,5),(SLTMP(I),I=1,LMAX)
   10 FORMAT(2I5,F5.1,F5.1,F10.1,F10.1,3F5.1,15/F5.2,9I5/8F10.1)
C     IMAX=NUMBER OF COLUMNS (K)
C     JMAX=NUMBER OF ROWS (J)
C     DELTT=TIME STEP LENGTH,HOURS
C     TTT=RESERVOIR TEMPERATURE, DEGREES FAHRENHEIT
C     TIMX=MAXIMUM TIME, HOURS
C     ALPHA=SPECIFIC HEAT X VELOCITY X DENSITY(ALL OF GAS)
C     DELTX=DISTANCE BETWEEN POINTS ALONG THE FRACTURE
C     DELTY=DISTANCE BETWEEN POINTS INTO THE WALL(FIRST 3 POINTS ONLY)
C     TT=INJECTION TEMPERATURE, DEGREES FAHRENHEIT
C     KSKIP=OPTION TO INCREASE TIME STEP LENGTH DURING RUN
C     TOL=TOLERANCE
C     NRUN=RUN NUMBER
C     KSLCE=OPTION TO CALCULATE TOTAL HEAT CONTENT AND PRINT TEMPERATURE
C           AND TOTAL HEAT CONTENT
C     ISSN=MAXIMUM NUMBER OF DESIRED SLICES FOR CROSS SECTION STUDY
C     LMAX=NUMBER OF ISOTHERMS DESIRED
C     IS(M)=VALUES OF DESIRED SLICES FOR CROSS SECTION STUDY
C     SLTMP(I)=TEMPERATURE OF DESIRED ISOTHERM
      READ(5,20)(Y(I),I=1,17)
   20 FORMAT(16F5.0/F5.0)
C     Y=DISTANCE INTO THE WALL, FEET
      TIME=0.0                                                           22242
      KIT=0
      HEAT=0.0
      DO 81 J=1,JMAX                                                     22242
      DO 81 K=1,IMAX                                                     22242
      TEMP(J,K)=TTT                                                      22242
      VTMP(J,K)=TTT
   81 TMPS(J,K)=TEMP(J,K)                                                22242
      TEMP(1,1)=TT
      TMPS(1,1)=TT
      VTMP(1,1)=TT                                                       22241
      TTMPS(1,1)=TT                                                     THERMAB2
      ATMP=0.0
      KWHO=1
      DD=1.
      QQ=0.0                                                             22242
      PP=0.
      PPP=0.0
      CALL CALCY                                                         22242
C     CALCY IS A SUBROUTINE THAT CALCULATES VARIABLE Y SPACING VALUES
      CALL COND4
C     COND4 IS A SUBROUTINE THAT CALCULATES CONDUCTANCES
      CALL CAPAC
C     CAPAC IS A SUBROUTINE THAT CALCULATES TOTAL HEAT REQUIREMENT
C           VALUES.
      DO 232 J=1,JMAX
      DO 232 K=1,IMAX
  232 CAP(J,K)=CAPT(J,K)
   82 GO TO(995,994),KSKIP
  994  IF(ATMP.NE.TIME) GO TO 995
      ATMP=ATMP+1.
      IF(TIME.GT.5) GO TO 318
      DELTT=1.
      GO TO 995
  318 IF(DELTT.GE.15.)GO TO 993
      UU=DD-5.
      TME=5.*((1.2)**UU)
      DELTT=TME-TIME
      IF(DELTT.LT.15.)GO TO 995
      DELTT=15.
  993 IF(TIME.GT.1000.)GO TO 996
      DELTT=15.0
      GO TO 995
  996 IF(TIME.GT.3000.)GO TO 997
      DELTT=30.0
      GO TO 995
  997 IF(TIME.GT.10000.)GO TO 998
      DELTT=50.0
      GO TO 995
  998 DELTT=100.
C     KWHO IS THE SWITCH TO DETERMINE DIRECTION OF ADIP SWEEP(HORIZONTAL
C          OR VERTICAL)
  995 GO TO(991,53),KWHO
  991 CALL COND1
C     COND1 IS A SUBROUTINE THAT SOLVES HORIZONTAL ADIP
      IF(DENOM.EQ.0.0)GO TO 69                                           22242
      KIT=KIT+1
      IF(KIT.GE.15)GO TO 222
      KK=1                                                               22242
      DO 84 N=1,JMAX
      DO 84 M=1,IMAX                                                     22241
      IF(TMPS(N,M).LT.100.10) GO TO 84
      IF(ABS((VTMP(N,M)-TMPS(N,M))/TMPS(N,M)).GT.TOL)GO TO 86            22241
   84 CONTINUE                                                           22241
      GO TO 67                                                           22241
   86 DO 88 J=1,JMAX                                                     22241
      DO 88 K=1,IMAX                                                     22241
   88 VTMP(J,K)=TMPS(J,K)                                                22241
      KWHO=2
      GO TO 82
   53 CALL COND2                                                         22242
C     COND2 IS A SUBROUTINE THAT SOLVES VERTICAL ADIP
      IF(DENOM.EQ.0.0)GO TO 69                                           22242
      KK=2                                                               22242
      DO 64 N=1,JMAX
      DO 64 M=1,IMAX                                                     22241
      IF(TMPS(N,M).LT.100.10) GO TO 64
      IF(ABS((VTMP(N,M)-TMPS(N,M))/TMPS(N,M)).GT.TOL)GO TO 65            22241
   64 CONTINUE                                                           22241
      GO TO 67                                                           22241
   65 DO 83 J=1,JMAX                                                     22241
      DO 83 K=1,IMAX                                                     22241
   83 VTMP(J,K)=TMPS(J,K)                                                22241
      KWHO=1
      GO TO 82                                                           22241
   67 DO 66 J=1,JMAX                                                     22241
      DO 66 K=1,IMAX                                                     22241
      IF(TMPS(J,K).LT.100.10) GO TO 66
      CAP(J,K)=CAPT(J,K)
      TEMP(J,K)=TMPS(J,K)
   66 CONTINUE
      PPP=PPP+1D.0
      IF(PPP.NE.10.)GO TO 741
      PPP=0.0
      JBMAX=JMAX-2                                                       22241
      IBMAX=IMAX-2                                                       22241
      FIMAX=IMAX                                                         22241
C     THIS ACCOUNTS FOR THE HEAT STORED IN ROCK                          22241
      DO 40 J=1,JMAX                                                     22241
      SUM=0.                                                             22241
      DO 80 K=1,IBMAX,2                                                  22241
   80 SUM=SUM+CAP(J,K)+4.*CAP(J,K+1)+CAP(J,K+2)
   40 AREA(J)=SUM*(DELTX/3.)                                             22241
      BB2=0.                                                             22241
      DO 90 J=1,JBMAX,2                                                  22241
      H(J)=Y(J+1)-Y(J)                                                   22241
      HA(J)=(Y(J+2)-Y(J+1))/H(J)                                         22241
   90 BB2=BB2+(H(J)*(HA(J)+1.)/(6.*HA(J)))*((2.*HA(J)-1.)*AREA(J+2)+(HA(  22241
     1J)+1.)**2*AREA(J+1)+HA(J)*(2.-HA(J))*AREA(J))                      22241
      HTORG=Y(JMAX)*DELTX*FIMAX*3149.+(CAP(I,1)-3149.)*DELTX*.25
C     HTORG=INITIAL HEAT CONTENT OF ROCK AT 100 DEGREES FAHRENHEIT
      HEAT=BB2-HTORG
C     THIS ACCOUNTS FOR HEAT OUT OF PRODUCER                             22241
  741 HTOUT=HTOUT+TEMP(1,IMAX)*DELTT*ALPHA                               22241
      HTTOT=HTOUT+HEAT+HTORG
      HTIN=HTIN+ALPHA*TT*DELTT                                           22241
      HTINN=HTIN+HTORG
      ERR=ABS((HTINN-HTTOT)/HTTOT)
```

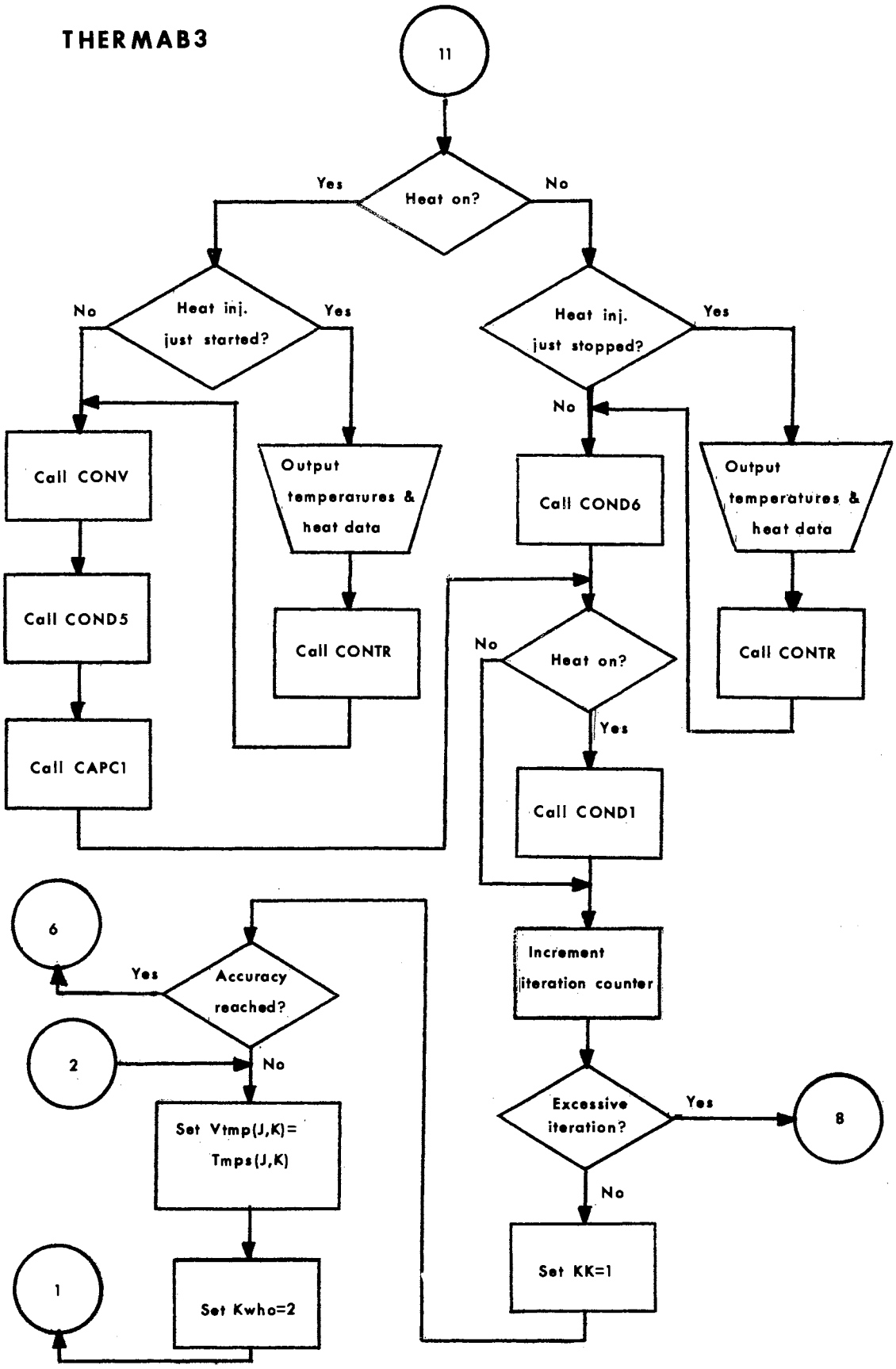
```
      GO TO(4,5),KSLCE                                                  THERMAB3
C     THIS ACCOUNTS FOR THE TEMPERATURE AND CULULATIVE HEAT PRINT OUT AT
C     VARIOUS SELECTED POSITIONS ALONG THE FRACTURE.
    4 WRITE(6,26)                                                       THERMAB3
   26 FORMAT(1H1,30X,'INPUT DATA FOR CROSS SECTION STUDY'//9X,'LOCATION(THERMAB3
     1FEET DOWN FRACTURE)',2X,'TEMPERATURE',9X,'TIME(HOURS)',9X,'CUMULATTHERMAB3
     2IVE HEAT'//)                                                      THERMAB3
      JBMAX=JMAX-2                                                      THERMAB3
      DO 70 K=1,IMAX                                                    THERMAB3
      DO 70 M=1,ISSN                                                    THERMAB3
      IF(K.NE.IS(M+1))GO TO 70
      BB3=0.0                                                           THERMAB3
      DO 71 J=1,JBMAX,2                                                 THERMAB3
      H(J)=Y(J+1)-Y(J)                                                  THERMAB3
      HA(J)=(Y(J+2)-Y(J+1))/H(J)                                          22241
   71 BB3=BB3+((H(J)*(HA(J)+1.)/(6.*HA(J)))*((2.*HA(J)-1.)*CAP(J+2,K)+(H
     1A(J)+1.)**2*CAP(J+1,K)+HA(J)*(2.-HA(J))*CAP(J,K)))*DELTX
      AIS=IS(M)
      ALOCF=DELTX*AIS
      WRITE(6,25)ALOCF,TEMP(1,K),TIME,BB3                               THERMAB3
   25 FORMAT(9X,F10.1,20X,F10.1,10X,F10.1,7X,E18.8)                     THERMAB3
   70 CONTINUE                                                          THERMAB3
    5 TIME=TIME+DELTT                                                     22241
      IF(TIME.GT.3000.)GO TO 56
      TOL=.05
      GO TO 55
   56 IF(TIME.GT.4000.)GO TO 57
      TOL=.04
      GO TO 55
   57 IF(TIME.GT.5000.)GO TO 58
      TOL=.03
      GO TO 55
   58 IF(TIME.GT.15000.)GO TO 59
      TOL=.02
      GO TO 55
   59 TOL=.01
   55 ATMP=TIME
      DD=DD+1.
      PP=PP+1.
      IF(PP.NE.10.) GO TO 891
      PP=0.
      WRITE(6,33)                                                         22242
   33 FORMAT(1H1,2X2HJ=5X2H 15X2H 25X2H 35X2H 45X2H 55X2H 65X2H 75X2H 85  22242
     1X2H 95X2H105X2H115X2H125X2H135X2H145X2H15)                          22242
      MCONT=0
      DO 36 K=1,42
   36 WRITE(6,37)K,(TEMP(J,K),J=1,JMAX)                                   22241
   37 FORMAT(I4,17F7.1)
      WRITE(6,33)                                                       THERMAB3
      DO 24 K=43,IMAX                                                   THERMAB3
   24 WRITE(6,37)K,(TEMP(J,K),J=1,JMAX)                                 THERMAB3
  887 WRITE(6,30)TIME,ERR,HEAT,HTOUT,HTIN,HTTOT,HTORG,DELTT
   30 FORMAT(F10.1,2X,5HHOURS/F10.5,2X,27HDISCREPANCY IN HEAT BALANCE/E1    22241
     18.8,2X,31HBTU OF HEAT STORED IN RESERVOIR/E18.8,2X,20HBTU OF HEAT     22241
     2PRODUCED/E18.8,2X,20HBTU OF HEAT INJECTED/E18.8,2X,5HHTTOT/E18.8,2    22241
     3X,5HHTORG/F10.1,21HTIME INCREMENT, HOURS)
C     INTERPOLATING ROUTINE FOR DETERMING POSITION OF SELECTED ISOTHERMS
C     LET LMAX EQUAL NUMBER OF ISOTHERMS DESIRED
C     SLTMP(I) REFERS TO TEMPERATURE OF DESIRED ISOTHERM
      DO 263 I=1,LMAX
      DO 260 K=1,IMAX
      DO 261 J=1,JMAX
      IF(TEMP(J,K).LT.SLTMP(I))GO TO 262
  261 CONTINUE
  262 IF(J.EQ.1)GO TO 260
      DIST(I,K)=Y(J-1)+((TEMP(J-1,K)-SLTMP(I))/(TEMP(J-1,K)-TEMP(J,K)))*
     1(Y(J)-Y(J-1))
  260 CONTINUE
  263 CONTINUE
      WRITE(6,264)(SLTMP(I),I=1,LMAX)
  264 FORMAT(1H1,6X,'DISTANCE',7X,'ISOTHERM  ISOTHERM  ISOTHERM  ISOTHERTHERMAB6
     1M  ISOTHERM  ISOTHERM  ISOTHERM  ISOTHERM'/8X,'DOWN',9X,'DEGREES FTHERMAB6
     2 DEGREES F DEGREES F DEGREES F DEGREES F DEGREES F DEGREES F DEGRETHERMAB6
     3EES F'/7X,'FRACTURE',4X,8F10.1/9X,'FEET'/20X,'FEET INTO FEET INTO THERMAB6
     4FEET INTO FEET INTO FEET INTO FEET INTO FEET INTO FEET INTO'/22X,'THERMAB6
     5WALL',6X,'WALL',6X,'WALL',6X,'WALL',6X,'WALL',6X,'WALL',6X,'WALL',THERMAB6
     66X,'WALL')                                                        THERMAB6
      DO 266 K=1,40                                                     THERMAB6
      FK2=K-1.                                                          THERMAB6
      XFT=DELTX*FK2                                                     THERMAB6
  266 WRITE(6,267)XFT,(DIST(I,K),I=1,LMAX)                              THERMAB6
  267 FORMAT(7X,F6.0,4X,8F10.2)
      WRITE(6,264)(SLTMP(I),I=1,LMAX)
      DO 265 K=41,IMAX
      FK2=K-1
      XFT=DELTX*FK2
  265 WRITE(6,267)XFT,(DIST(I,K),I=1,LMAX)                              THERMAB6
      WRITE(6,223)DD,KIT
  223 FORMAT(3X,F10.0,I5)
  891 KIT=0
      IF(TIME.GT.TIMX)GO TO 105
      GO TO(86,65),KK                                                     22241
  222 WRITE(6,223)DD,KIT
   69 STOP                                                                22242
      END                                                                 22242
      SUBROUTINE CALCY                                                    22242
      DIMENSION TMPS(17,51),TEMP(17,51),W(17,51),G(17,51),COND(17,51),Y(  22242
     117),YPLUS(17),YNEG(17),YMID(17),TTMPS(17,51),AREA(17),H(17),HA(17)
     2,VTMP(17,51),CAPT(17,51),CAP(17,51),IS(10),SLTMP(10),DIST(8,51)
      COMMON TMPS,TEMP,W,G,Y,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPHA,   22242
     1DELTX,DENOM,PO,P1,P2,P3,TTMPS,CAPT,CAP,TIME
      NMAX=JMAX-1                                                         22242
      DO 2 J=1,NMAX                                                       22242
    2 YPLUS(J)=Y(J+1)-Y(J)                                                22242
      DO 3 J=2,JMAX                                                       22242
    3 YNEG(J)=Y(J)-Y(J-1)                                                 22242
      YNEG(1)=YPLUS(1)                                                    22242
      YPLUS(JMAX)=YNEG(JMAX)                                              22242
      DO 4 J=1,JMAX                                                       22242
    4 YMID(J)=0.5*(YPLUS(J)+YNEG(J))                                      22242
      RETURN                                                              22242
      END                                                                 22242
      SUBROUTINE COND1                                                    22242
      DIMENSION TMPS(17,51),TEMP(17,51),W(17,51),G(17,51),COND(17,51),Y(  22242
     117),YPLUS(17),YNEG(17),YMID(17),TTMPS(17,51),AREA(17),H(17),HA(17)
     2,VTMP(17,51),CAPT(17,51),CAP(17,51),IS(10),SLTMP(10),DIST(8,51)
      COMMON TMPS,TEMP,W,G,Y,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPHA,   22242
     1DELTX,DENOM,PO,P1,P2,P3,TTMPS,CAPT,CAP,TIME
      DO 1 J=1,JMAX                                                       22242
      DO 20 K=1,IMAX                                                      22242
      IF(J.NE.1)GO TO 35                                                THERMAB2
      IF(K.NE.1)GO TO 35                                                THERMAB2
      GO TO 20                                                          THERMAB2
   35 FMAX=JMAX
      NDD=0
      DD=0.0
      DO 2 I=1,JMAX
      IF(TMPS(I,K).LT.700.)GO TO 4
      DD=DD+1.0
      NDD=DD
    2 CONTINUE
    4 IF(DD.LT.2.)GO TO 7
      IF(J.NE.1) GO TO 101
      GG=0.9
      GO TO 8
  101 FFF=.1
      IF(K.NE.2) GO TO 102
      ALPA1=(YMID(J)*ALPHA*FFF*(TMPS(1,1)-TMPS(J,K+1)))/((Y(NDD)+0.5*YPL
     1US(NDD)-0.5)*(TMPS(J,K-1)-TMPS(J,K+1)))
      GO TO 9
  102 ALPA1=(YMID(J)/(Y(NDD)+0.5*YPLUS(NDD)-.5))*ALPHA*FFF
      GO TO 9
    7 GG=1.
    8 ALPA1=ALPHA*GG
    9 IF(J.EQ.1)GO TO 78                                                  22242
   51 IF(J.EQ.JMAX)GO TO 70                                               22242
      IF(K.EQ.1)GO TO 79                                                  22242
      IF(K.EQ.IMAX)GO TO 80                                               22242
      CXPUS=(COND(J,K)+COND(J,K+1))/2.                                    22241
      CXMIN=(COND(J,K)+COND(J,K-1))/2.                                    22241
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                    22241
```

```
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22241
      GO TO 77                                                          22241
   79 CXPUS=(COND(J,K)+COND(J,K+1))/2.                                  22241
      CXMIN=CXPUS                                                       22241
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                  22241
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22241
      GO TO 77                                                          22241
   80 CXMIN=(COND(J,K)+COND(J,K-1))/2.                                  22242
      CXPUS=CXMIN                                                       22241
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                  22241
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22241
      GO TO 77                                                          22241
   78 IF(K.EQ.1)GO TO 75                                                22242
      IF(K.EQ.IMAX)GO TO 76                                             22242
      CXPUS=(COND(J,K)+COND(J,K+1))/2.                                  22242
      CXMIN=(COND(J,K)+COND(J,K-1))/2.                                  22242
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                  22242
      CYMIN=CYPUS                                                       22242
      GO TO 77                                                          22242
   70 IF(K.EQ.1)GO TO 72                                                22242
      IF(K.EQ.IMAX)GO TO 73                                             22242
      CXPUS=(COND(J,K)+COND(J,K+1))/2.                                  22242
      CXMIN=(COND(J,K)+COND(J,K-1))/2.                                  22242
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22242
      CYPUS=CYMIN                                                       22242
      GO TO 77                                                          22242
   72 CXPUS=(COND(J,K)+COND(J,K+1))/2.                                  22242
      CXMIN=CXPUS                                                       22242
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22242
      CYPUS=CYMIN                                                       22242
      GO TO 77                                                          22242
   73 CXMIN=(COND(J,K)+COND(J,K-1))/2.                                  22242
      CXPUS=CXMIN                                                       22242
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22242
      CYPUS=CYMIN                                                       22242
      GO TO 77                                                          22242
   75 CXPUS=(COND(J,K)+COND(J,K+1))/2.                                  22242
      CXMIN=CXPUS                                                       22242
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                  22242
      CYMIN=CYPUS                                                       22242
      GO TO 77                                                          22242
   76 CXMIN=(COND(J,K)+COND(J,K-1))/2.                                  22242
      CXPUS=CXMIN                                                       22242
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                  22242
      CYMIN=CYPUS                                                       22242
   77 CONTINUE                                                          22242
      IF(J.EQ.1)GO TO 89                                                22242
      IF(TMPS(J,K).GT.700.)GO TO 89
      HH=0.                                                             22242
      GO TO 641
   89 HH=1.                                                             22240
  641 A1=CXMIN/CXPUS+(HH*ALPA1*DELTX)/(2.*CXPUS)                        22240
      B1=-(1.+CXMIN/CXPUS+(DELTX**2/DELTT)*(CAPT(J,K)/(TMPS(J,K)*CXPUS))
     1)
      C1=-((ALPA1*HH*DELTX)/(2.*CXPUS)-1.)
  642 D14=-(DELTX**2/(YMID(J)*CXPUS))                                   22242
      IF(J.LT.JMAX)GO TO 24                                             22242
      D15=CYPUS*(TMPS(J-1,K)-TMPS(J,K))/YPLUS(J)                        22242
      GO TO 25                                                          22242
   24 D15=(CYPUS*(TMPS(J+1,K)-TMPS(J,K)))/YPLUS(J)                      22242
   25 IF(J.GT.1)GO TO 26                                                22242
      D16=(CYMIN*(TMPS(J,K)-TMPS(J+1,K)))/YNEG(J)                       22242
      GO TO 27                                                          22242
   26 D16=(CYMIN*(TMPS(J,K)-TMPS(J-1,K)))/YNEG(J)                       22242
   27 D17=(DELTX**2*CAP(J,K))/(DELTT*CXPUS)
   99 D1=D14*(D15-D16)-D17
  556 IF(J.EQ.1)GO TO 40                                                THERMAB2
      IF(K.NE.1)GO TO 3                                                 THERMAB2
      C1=A1+C1                                                          THERMAB2
      W(J,1)=C1/B1                                                      THERMAB2
      G(J,1)=D1/B1                                                      THERMAB2
      GO TO 20                                                          THERMAB2
   40 IF(K.NE.2)GO TO 3                                                 THERMAB2
      W(1,2)=C1/B1                                                      THERMAB2
      G(1,2)=(D1-A1*TEMP(1,1))/B1                                       THERMAB2
      GO TO 20                                                          THERMAB2
    3 IF(K.NE.IMAX)GO TO 11                                             THERMAB2
      A1=A1+C1
      C1=0.0
      GO TO 11                                                          22242
   11 DENOM=B1-A1*W(J,K-1)                                              22242
      IF(DENOM.NE.0.)GO TO 18                                           22242
      WRITE(6,50)                                                       22242
   50 FORMAT(18H ZERO DENOMINATOR.)                                     22242
      GO TO 800                                                         22242
   18 W(J,K)=C1/DENOM                                                   22242
      G(J,K)=(D1-A1*G(J,K-1))/DENOM                                     22242
   20 CONTINUE                                                          22242
      TTMPS(J,IMAX)=G(J,IMAX)
      NMAX=IMAX-2                                                       THERMAB2
      IF(J.NE.1)GO TO 41                                                THERMAB2
      DO 30 I=1,NMAX                                                    THERMAB2
      II=IMAX-I                                                         THERMAB2
   30 TTMPS(J,II)=G(J,II)-W(J,II)*TTMPS(J,II+1)
      GO TO 1                                                           THERMAB2
   41 DO 42 I=2,IMAX                                                    THERMAB2
      II=IMAX+1-I                                                       THERMAB2
   42 TTMPS(J,II)=G(J,II)-W(J,II)*TTMPS(J,II+1)
    1 CONTINUE                                                          22242
      DO 95 J=1,JMAX
      DO 95 K=1,IMAX
      TMPS(J,K)=TTMPS(J,K)
   95 CONTINUE
      IF(TIME.GT.5.)GO TO 300
      DO 301 K=1,42
  301 WRITE(6,302)(TMPS(J,K),J=1,JMAX)
  302 FORMAT(17F7.1)
  300 CALL CAPAC
      CALL COND4
  800 RETURN                                                            22242
      END                                                               22242
      SUBROUTINE COND2                                                  22242
      DIMENSION TMPS(17,51),TEMP(17,51),W(17,51),G(17,51),COND(17,51),Y( 22242
     117),YPLUS(17),YNEG(17),YMID(17),TTMPS(17,51),AREA(17),H(17),HA(17)
     2,VTMP(17,51),CAPT(17,51),CAP(17,51),IS(10),SLTMP(10),DIST(8,51)
      COMMON TMPS,TEMP,W,G,Y,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPHA,  22242
     1DELTX,DENOM,P0,P1,P2,P3,TTMPS,CAPT,CAP,TIME
      C1=1.0                                                            22242
      DO 1 K=1,IMAX                                                     22242
      DO 20 J=1,JMAX                                                    22242
      IF(K.NE.1)GO TO 35                                                THERMAB2
      IF(J.NE.1)GO TO 35                                                THERMAB2
      GO TO 20                                                          THERMAB2
   35 FMAX=JMAX
      NDD=0
      DD=0.0
      DO 2 I=1,JMAX
      IF(TMPS(I,K).LT.700.)GO TO 4
      DD=DD+1.0
      NDD=DD
    2 CONTINUE
    4 IF(DD.LT.2.)GO TO 7
      IF(J.NE.1) GO TO 101
      GG=0.9
      GO TO 8
  101 FFF=.1
      IF(K.NE.2) GO TO 102
      ALPA1=(YMID(J)*ALPHA*FFF*(TMPS(1,1)-TMPS(J,K+1)))/((Y(NDD)+0.5*YPL
     1US(NDD)-0.5)*(TMPS(J,K-1)-TMPS(J,K+1)))
      GO TO 9
  102 ALPA1=(YMID(J)/(Y(NDD)+0.5*YPLUS(NDD)-.5))*ALPHA*FFF
      GO TO 9
    7 GG=1.0
    8 ALPA1=ALPHA*GG
    9 IF(J.EQ.1)GO TO 78                                                22242
   61 IF(J.EQ.JMAX)GO TO 70                                             22242
      IF(K.EQ.1)GO TO 79                                                22242
      IF(K.EQ.IMAX)GO TO 80                                             22242
      CXPUS=(COND(J,K)+COND(J,K+1))/2.                                  22241
```

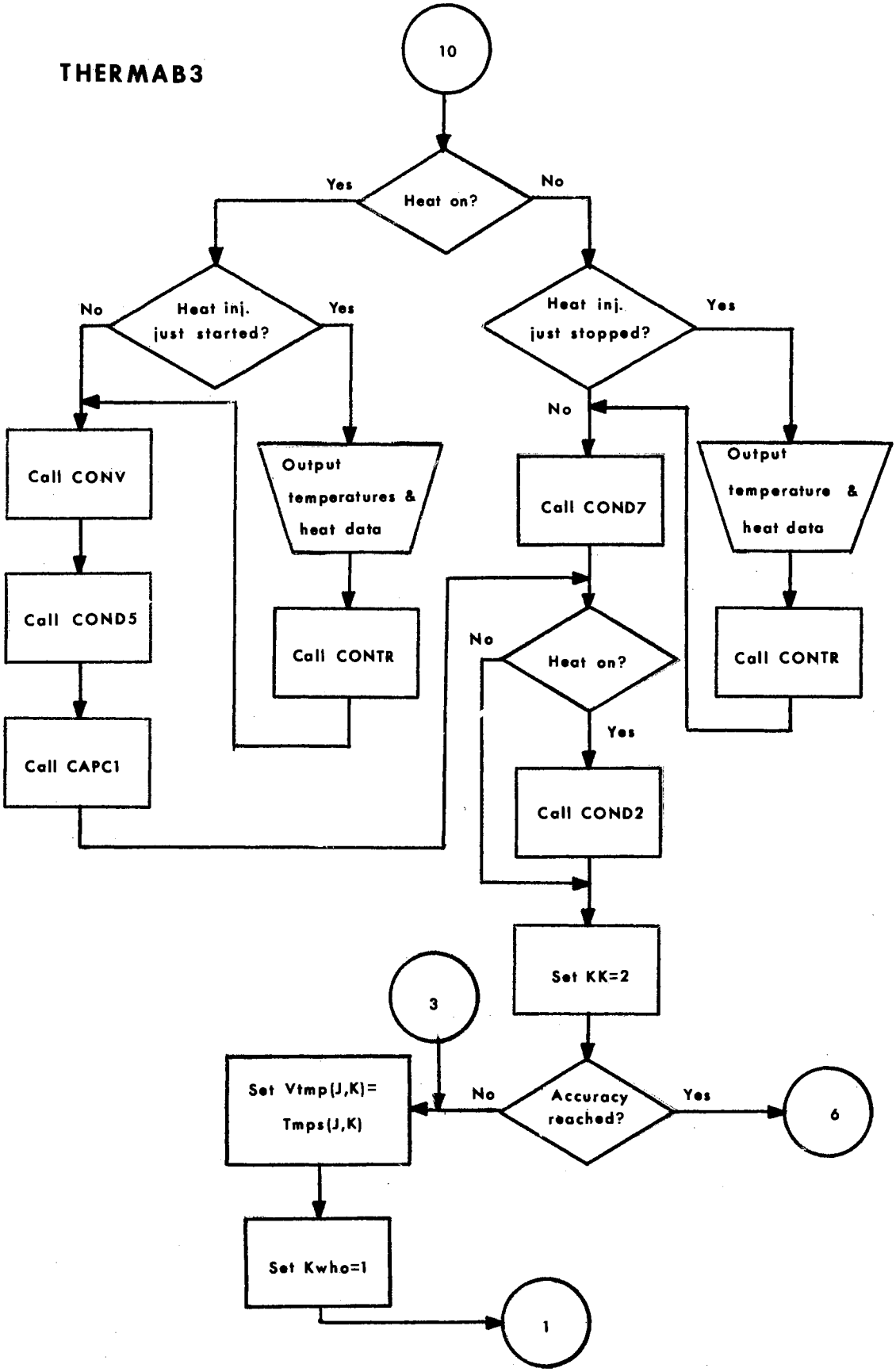
```
      CXMIN=(COND(J,K)+COND(J,K-1))/2.                                  22241
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                  22241
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22241
      GO TO 77                                                          22241
   79 CXPUS=(COND(J,K)+COND(J,K+1))/2.                                  22242
      CXMIN=CXPUS                                                       22242
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                  22241
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22241
      GO TO 77                                                          22241
   80 CXMIN=(COND(J,K)+COND(J,K-1))/2.                                  22242
      CXPUS=CXMIN                                                       22242
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                  22241
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22242
      GO TO 77                                                          22241
   78 IF(K.EQ.1)GO TO 75                                                22242
      IF(K.EQ.IMAX)GO TO 76                                             22242
      CXPUS=(COND(J,K)+COND(J,K+1))/2.                                  22242
      CXMIN=(COND(J,K)+COND(J,K-1))/2.                                  22242
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                  22242
      CYMIN=CYPUS                                                       22242
      GO TO 77                                                          22242
   70 IF(K.EQ.1)GO TO 72                                                22242
      IF(K.EQ.IMAX)GO TO 73                                             22242
      CXPUS=(COND(J,K)+COND(J,K+1))/2.                                  22242
      CXMIN=(COND(J,K)+COND(J,K-1))/2.                                  22242
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22242
      CYPUS=CYMIN                                                       22242
      GO TO 77                                                          22242
   72 CXPUS=(COND(J,K)+COND(J,K+1))/2.                                  22242
      CXMIN=CXPUS                                                       22242
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22242
      CYPUS=CYMIN                                                       22242
      GO TO 77                                                          22242
   73 CXMIN=(COND(J,K)+COND(J,K-1))/2.                                  22242
      CXPUS=CXMIN                                                       22242
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22242
      CYPUS=CYMIN                                                       22242
      GO TO 77                                                          22242
   75 CXPUS=(COND(J,K)+COND(J,K+1))/2.                                  22242
      CXMIN=CXPUS                                                       22242
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                  22242
      CYMIN=CYPUS                                                       22242
      GO TO 77                                                          22242
   76 CXMIN=(COND(J,K)+COND(J,K-1))/2.                                  22242
      CXPUS=CXMIN                                                       22242
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                  22242
      CYMIN=CYPUS                                                       22242
   77 CONTINUE                                                          22242
      IF(J.EQ.1)GO TO 89                                                22242
      IF(TMPS(J,K).GT.700.)GO TO 89
      HH=0.                                                             22240
      GO TO 91                                                          22240
   89 HH=1.                                                             22240
   91 A1=+(YPLUS(J)*CYMIN)/(YNEG(J)*CYPUS)                              22242
      B1=-(1.0+(YPLUS(J)*CYMIN)/(YNEG(J)*CYPUS)+(CAPT(J,K)*YMID(J)*YPLUS
     1(J))/(DELTT*CYPUS*TMPS(J,K)))
      C1=1.                                                             22240
      D14=-(YPLUS(J)*YMID(J))/(DELTX*CYPUS)
      IF(K.LT.IMAX)GO TO 24                                             22242
      D15=CXPUS*(TMPS(J,K-1)-TMPS(J,K))/DELTX                           22242
      D17=0.0
      D16=CXMIN*(TMPS(J,K)-TMPS(J,K-1))/DELTX                           22242
      GO TO 26                                                          22242
   24 IF(K.GT.1)GO TO 25                                                THERMAB2
      D15=CXPUS*(TMPS(J,K+1)-TMPS(J,K))/DELTX                           22242
      D16=CXMIN*(TMPS(J,K)-TMPS(J,K+1))/DELTX                           22242
   27 D17=0.
      GO TO 26                                                          THERMAB2
   25 D15=CXPUS*(TMPS(J,K+1)-TMPS(J,K))/DELTX                           22242
      D16=CXMIN*(TMPS(J,K)-TMPS(J,K-1))/DELTX                           22242
      D17=(ALPA1*HH*YPLUS(J)*YMID(J)*(TMPS(J,K+1)-TMPS(J,K-1)))/(2.*CYPU  22242
     1S*DELTX)                                                          22242
   26 D18=(CAPT(J,K)*YMID(J)*YPLUS(J))/(DELTT*CYPUS)
   99 D1=D14*(D15-D16)+D17-D18
  556 IF(K.NE.1)GO TO 10                                                THERMAB2
      IF(J.GT.2)GO TO 3
      W(2,1)=C1/B1                                                      THERMAB2
      G(2,1)=(D1-A1*TEMP(1,1))/B1
      GO TO 20                                                          THERMAB2
   10 IF(J.NE.1)GO TO 3                                                 THERMAB2
      C1=C1*A1                                                          THERMAB2
      W(1,K)=C1/B1
      G(1,K)=D1/B1                                                      22242
      GO TO 20
    3 IF(J.NE.JMAX)GO TO 11                                             22242
      A1=A1+C1
      CI=0.0
   11 DENOM=B1-A1*W(J-1,K)                                              22242
      IF(DENOM.NE.0.)GO TO 18                                           22242
      WRITE(6,50)                                                       22242
   50 FORMAT(18H ZERO DENOMINATOR.)                                     22242
   18 W(J,K)=C1/DENOM                                                   22242
      G(J,K)=(D1-A1*G(J-1,K))/DENOM                                     22242
   20 CONTINUE
      TTMPS(JMAX,K)=G(JMAX,K)
      NMAX=JMAX-2                                                       THERMAB2
      IF(K.NE.1)GO TO 41                                                THERMAB2
      DO 30 I=1,NMAX                                                    THERMAB2
      II=JMAX-I                                                         THERMAB2
   30 TTMPS(II,K)=G(II,K)-W(II,K)*TTMPS(II+1,K)
      GO TO 1                                                           THERMAB2
   41 DO 42 I=2,JMAX                                                    THERMAB2
      II=JMAX+1-I                                                       THERMAB2
   42 TTMPS(II,K)=G(II,K)-W(II,K)*TTMPS(II+1,K)
    1 CONTINUE                                                          22242
      DO 95 J=1,JMAX
      DO 95 K=1,IMAX
      TMPS(J,K)=TTMPS(J,K)
   95 CONTINUE
      IF(TIME.GT.5.)GO TO 300
      DO 301 K=1,42
  301 WRITE(6,302)(TMPS(J,K),J=1,JMAX)
  302 FORMAT(17F7.1)
  300 CALL CAPAC
      CALL COND4
  800 RETURN                                                            22242
      END                                                               22242
      SUBROUTINE COND4
      DIMENSION TMPS(17,51),TEMP(17,51),W(17,51),G(17,51),COND(17,51),Y(  22242
     117),YPLUS(17),YNEG(17),YMID(17),TTMPS(17,51),AREA(17),H(17),HA(17)
     2,VTMP(17,51),CAPT(17,51),CAP(17,51),IS(10),SLTMP(10),DIST(8,51)
      COMMON TMPS,TEMP,W,G,Y,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPHA,   22242
     1DELTX,DENOM,PO,P1,P2,P3,TTMPS,CAPT,CAP,TIME
      DO 5 J=1,JMAX                                                     22242
      DO 5 K=1,IMAX                                                     22242
      IF(TIME.EQ.0.0)GO TO 9
      IF(TMPS(J,K).LT.200.) GO TO 5
    9 IF(TMPS(J,K).GE.200.)GO TO 1
      COND(J,K)=0.9
      GO TO 5
    1 IF(TMPS(J,K).GT.1200.)GO TO 7
      TSQ=TMPS(J,K)*TMPS(J,K)
      TCUB=TMPS(J,K)*TSQ
      COND(J,K)=.12561573E+1-.18377799E-2*TMPS(J,K)+.10902421E-5*TSQ-.15
     1084869E-9*TCUB
      GO TO 5
    7 COND(J,K)=0.36
    5 CONTINUE
      RETURN                                                            22242
      END                                                               22242
      SUBROUTINE CAPAC
      DIMENSION TMPS(17,51),TEMP(17,51),W(17,51),G(17,51),COND(17,51),Y(  22242
     117),YPLUS(17),YNEG(17),YMID(17),TTMPS(17,51),AREA(17),H(17),HA(17)
     2,VTMP(17,51),CAPT(17,51),CAP(17,51),IS(10),SLTMP(10),DIST(8,51)
      COMMON TMPS,TEMP,W,G,Y,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPHA,   22242
     1DELTX,DENOM,PO,P1,P2,P3,TTMPS,CAPT,CAP,TIME
      DO 232 J=1,JMAX
      DO 232 K=1,IMAX
      IF(TIME.EQ.0.0)GO TO 9
      IF(TMPS(J,K).LT.100.01) GO TO 232
    9 IF(TMPS(J,K).GT.450.)GO TO 233
      CAPT(J,K)=(23.5+.307*(TMPS(J,K)-100.))*134.0
      GO TO 232
  233 IF(TMPS(J,K).GT.900.)GO TO 234
      CAPT(J,K)=(130.0+.344*(TMPS(J,K)-450.))*134.0
      GO TO 232
  234 IF(TMPS(J,K).GT.1100.)GO TO 235
      CAPT(J,K)=134.0*(285.0+.485*(TMPS(J,K)-900.))
      GO TO 232
  235 IF(TMPS(J,K).GT.1600.)GO TO 236
      CAPT(J,K)=134.0*(382.0+.6240*(TMPS(J,K)-1100.))
      GO TO 232
  236 CAPT(J,K)=134.0*(694.0+.247*(TMPS(J,K)-1600.))
  232 CONTINUE
      RETURN
      END
```

Thermab3

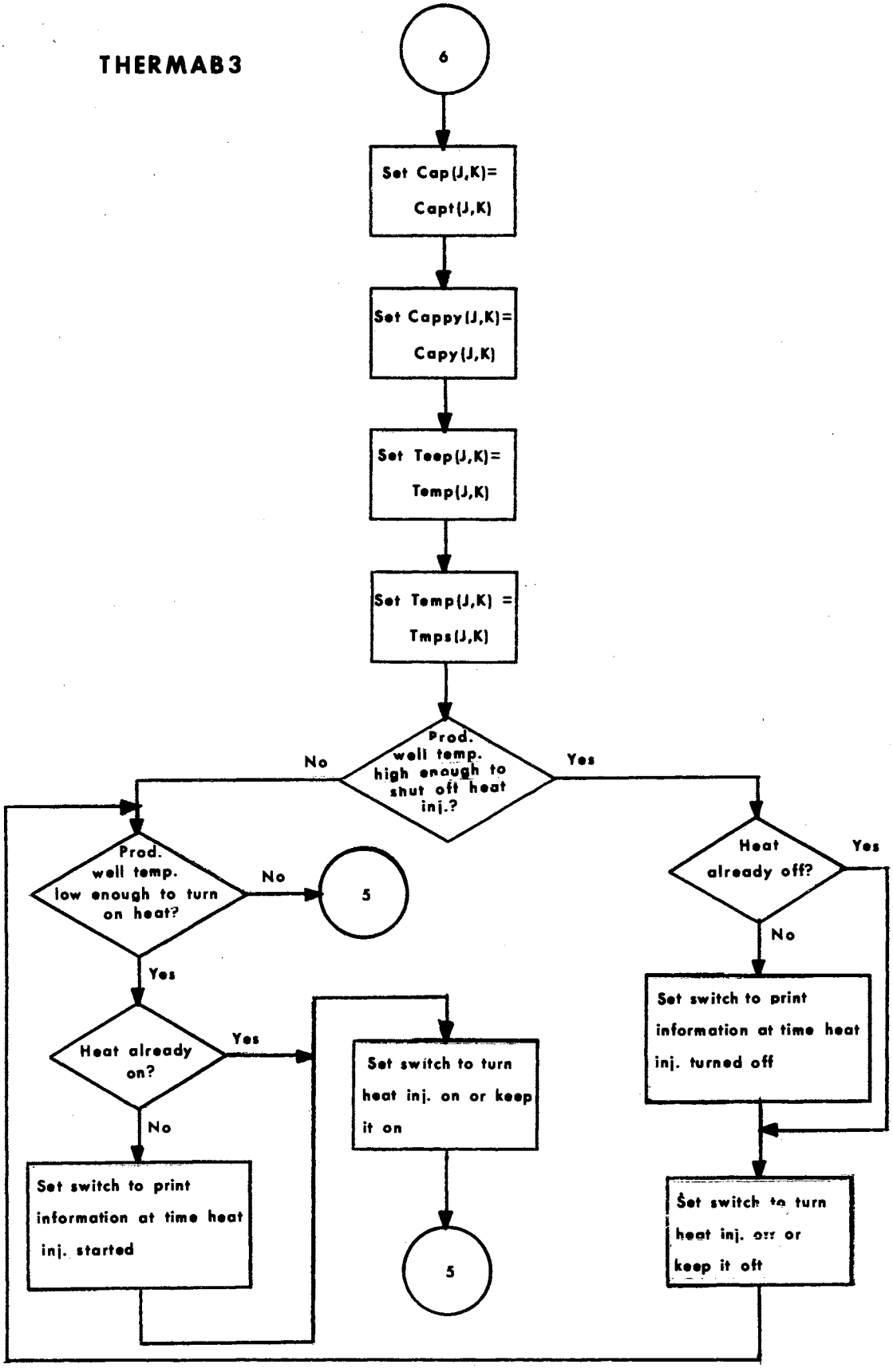

Start
Input system data
Compute velocity
Output general test data
Initialize parameters
Call CALCY
Call CAPAC
Call COND4
7
7
Call CAPSK
Initialize permanent cap arrays
1
Increase time step length during run?
No
Yes
End of time step?
No
Yes
Time step length set according to time into run
Vertical or horizontal sweep?
Vertical
KWHO=2
10
Horizontal
KWHO=1
11

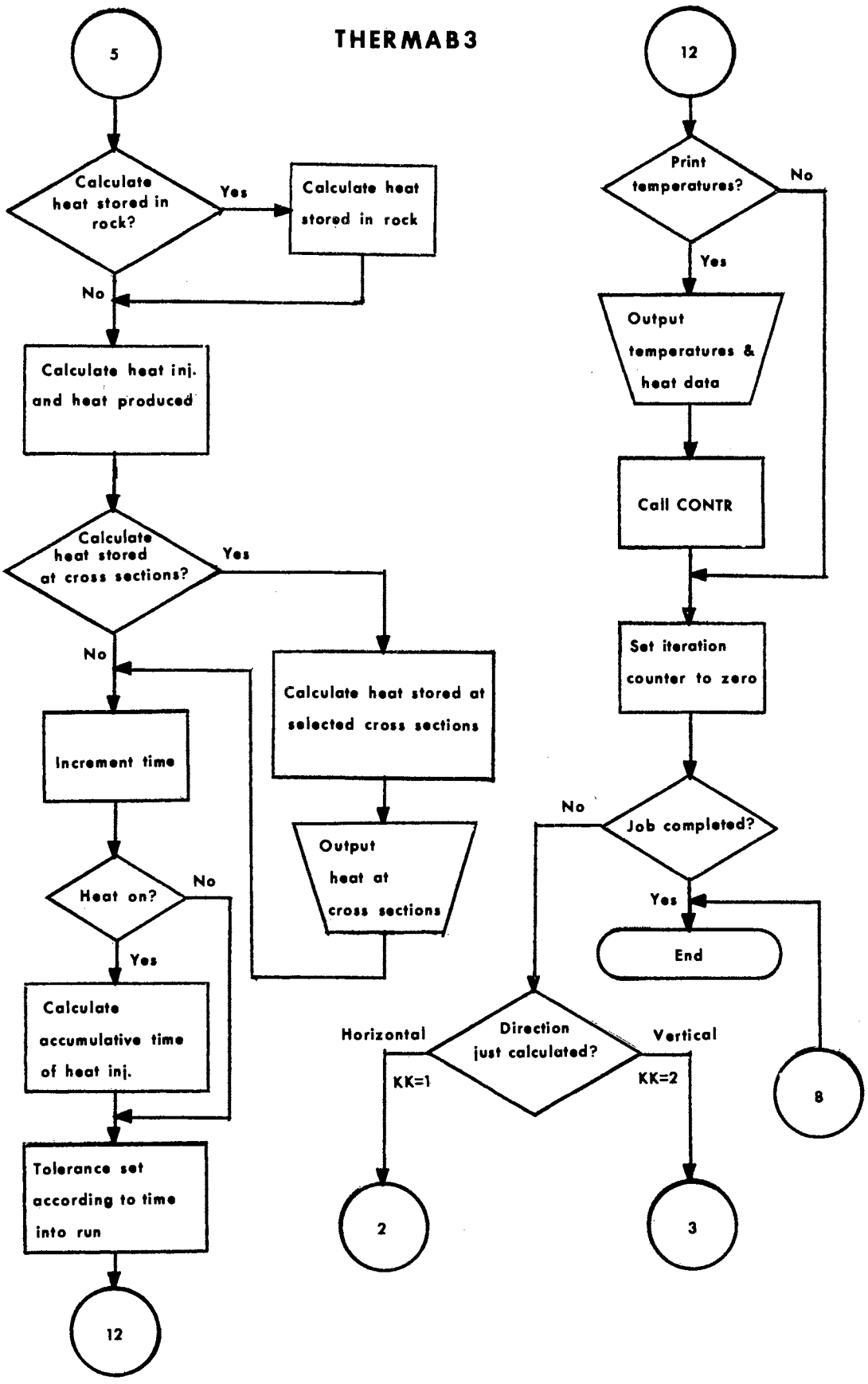

THERMAB3
11
Heat on?
Yes
No
Heat inj. just started?
No
Yes
Heat inj. just stopped?
Yes
No
Call CONV
Output temperatures & heat data
Call COND6
Output temperatures & heat data
Call COND5
Call CONTR
Heat on?
No
Call CONTR
Yes
Call CAPC1
Call COND1
6
Yes
Accuracy reached?
Increment iteration counter
2
No
Excessive iteration?
Yes
8
Set Vtmp(J,K)= Tmps(J,K)
No
Set KK=1
1
Set Kwho=2

THERMAB3

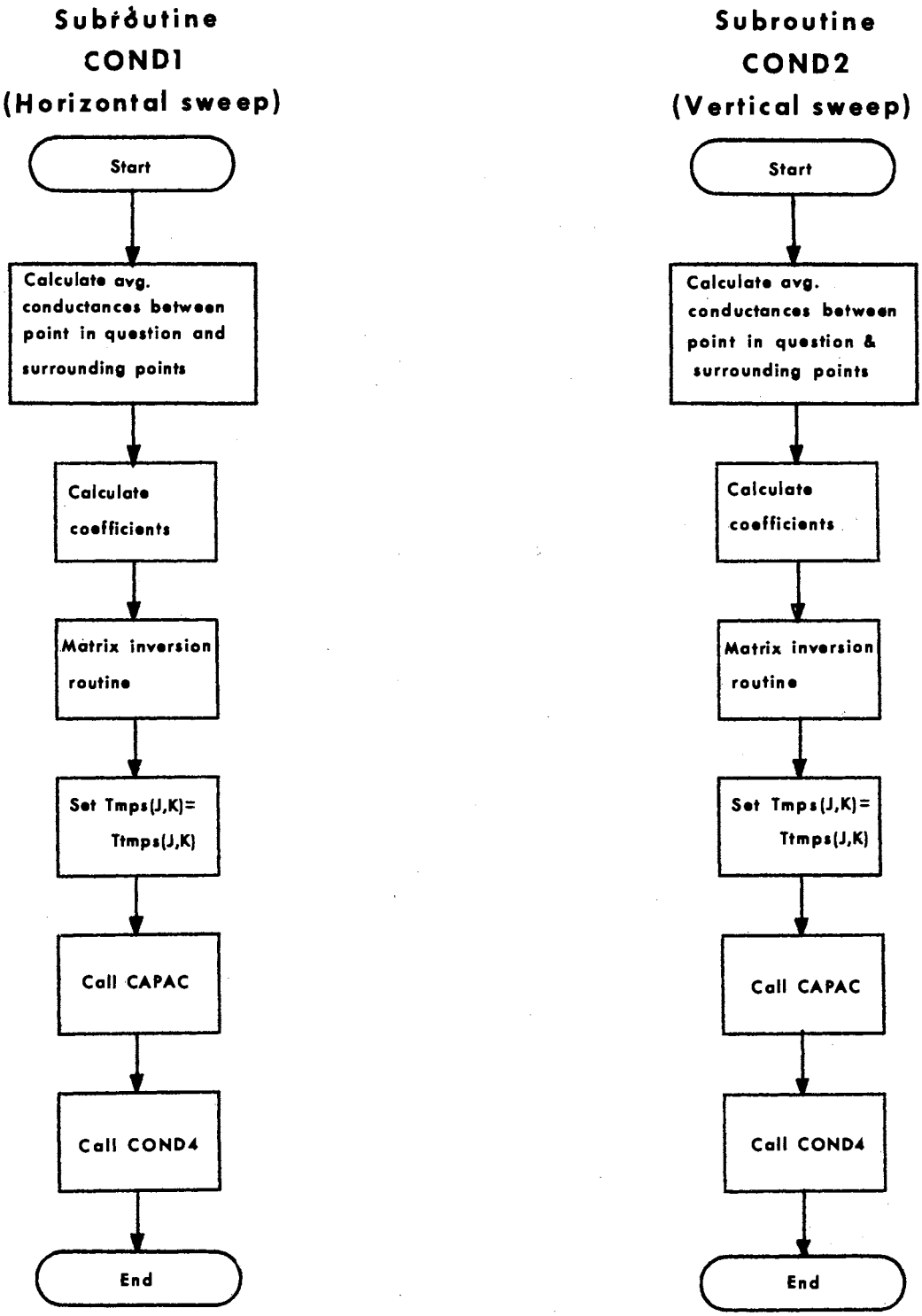

10
Heat on?
Yes
No
Heat inj. just started?
No
Yes
Heat inj. just stopped?
Yes
No
Call CONV
Call COND5
Call CAPC1
Output temperatures & heat data
Call CONTR
Call COND7
Output temperature & heat data
Call CONTR
No
Heat on?
Yes
Call COND2
Set KK=2
3
Set Vtmp(J,K)= Tmps(J,K)
No
Accuracy reached?
Yes
6
Set Kwho=1
1

THERMAB3

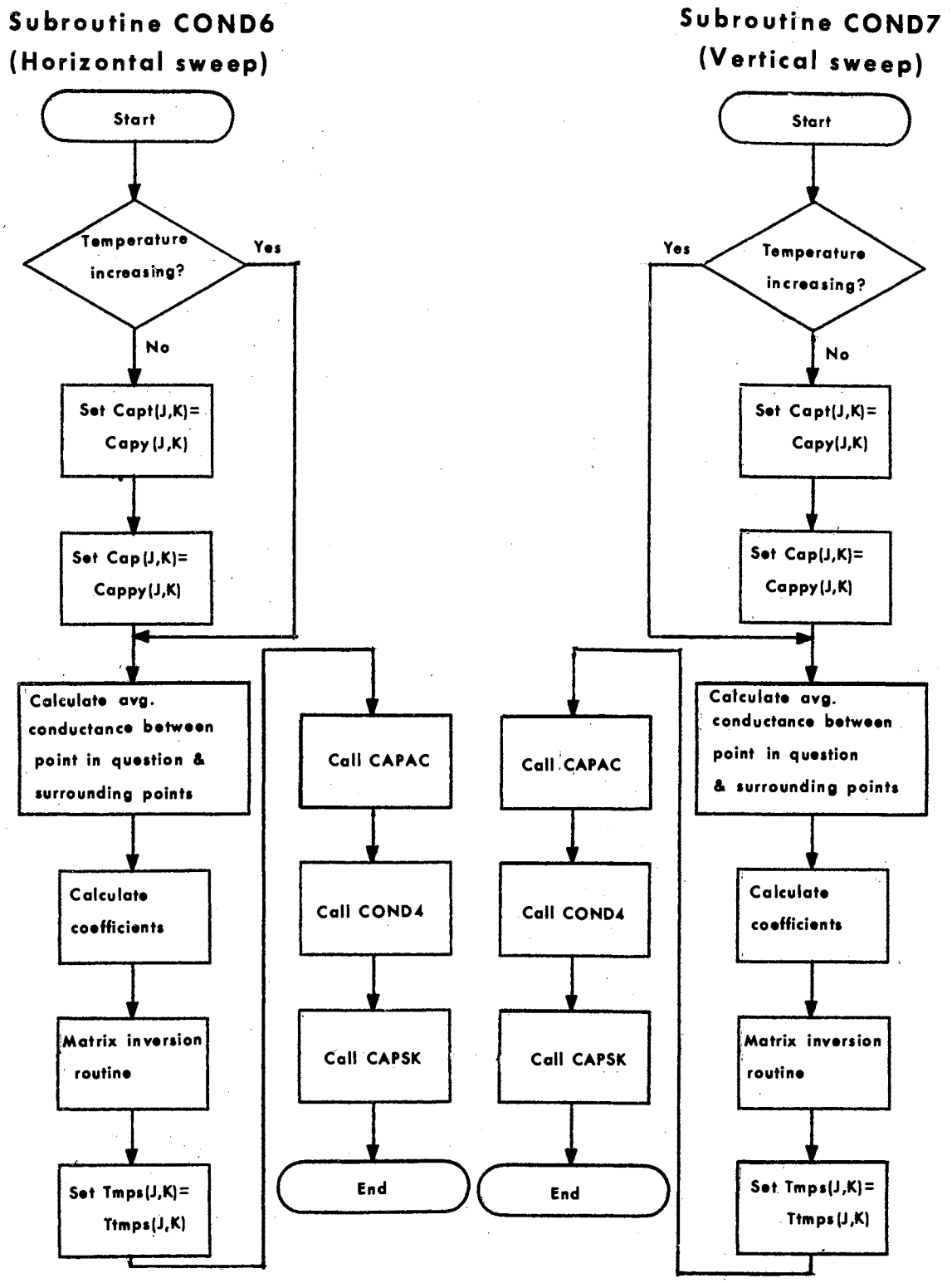

6
Set Cap(J,K)= Capt(J,K)
Set Cappy(J,K)= Capy(J,K)
Set Teep(J,K)= Temp(J,K)
Set Temp(J,K) = Tmps(J,K)
Prod. well temp. high enough to shut off heat inj.?
No
Yes
Prod. well temp. low enough to turn on heat?
No
5
Yes
Heat already on?
Yes
No
Set switch to print information at time heat inj. started
Set switch to turn heat inj. on or keep it on
5
Heat already off?
Yes
No
Set switch to print information at time heat inj. turned off
Set switch to turn heat inj. off or keep it off


THERMAB3
5
Calculate heat stored in rock?
Yes
Calculate heat stored in rock
No
Calculate heat inj. and heat produced
Calculate heat stored at cross sections?
Yes
Calculate heat stored at selected cross sections
Output heat at cross sections
No
Increment time
Heat on?
No
Yes
Calculate accumulative time of heat inj.
Tolerance set according to time into run
12
12
Print temperatures?
No
Yes
Output temperatures & heat data
Call CONTR
Set iteration counter to zero
Job completed?
No
Yes
End
Direction just calculated?
Horizontal
KK=1
Vertical
KK=2
2
3
8

THERMAB3

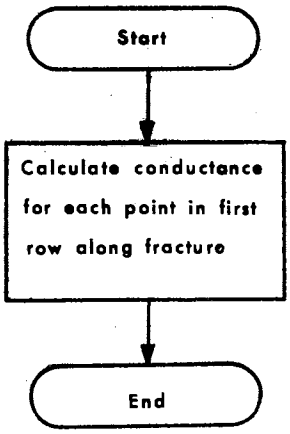

Subroutine
COND1
(Horizontal sweep)
Start
Calculate avg. conductances between point in question and surrounding points
Calculate coefficients
Matrix inversion routine
Set Tmps(J,K)= Ttmps(J,K)
Call CAPAC
Call COND4
End
Subroutine
COND2
(Vertical sweep)
Start
Calculate avg. conductances between point in question & surrounding points
Calculate coefficients
Matrix inversion routine
Set Tmps(J,K)= Ttmps(J,K)
Call CAPAC
Call COND4
End

THERMAB3

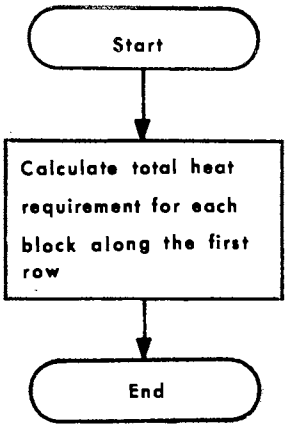

Subroutine COND6
(Horizontal sweep)
Start
Temperature increasing?
Yes
No
Set Capt(J,K)= Capy(J,K)
Set Cap(J,K)= Cappy(J,K)
Calculate avg. conductance between point in question & surrounding points
Calculate coefficients
Matrix inversion routine
Set Tmps(J,K)= Ttmps(J,K)
Call CAPAC
Call COND4
Call CAPSK
End
Subroutine COND7
(Vertical sweep)
Start
Yes
Temperature increasing?
No
Set Capt(J,K)= Capy(J,K)
Set Cap(J,K)= Cappy(J,K)
Call CAPAC
Call COND4
Call CAPSK
End
Calculate avg. conductance between point in question & surrounding points
Calculate coefficients
Matrix inversion routine
Set Tmps(J,K)= Ttmps(J,K)

THERMAB3

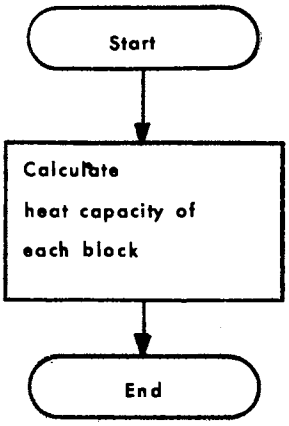
Subroutine COND5



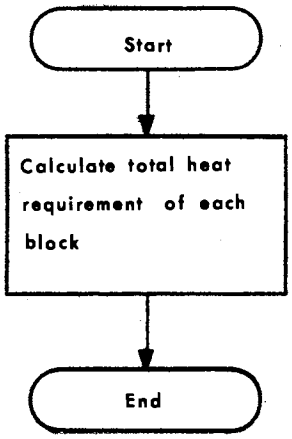
Subroutine CAPC1



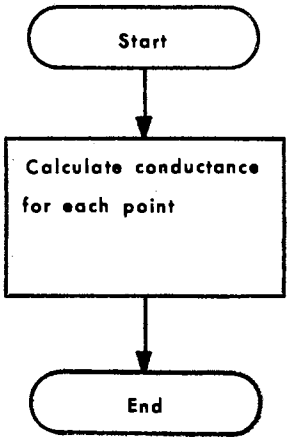
Subroutine CAPSK



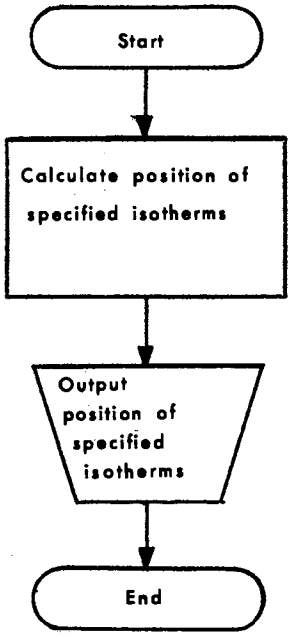
Subroutine CAPAC



Subroutine COND4

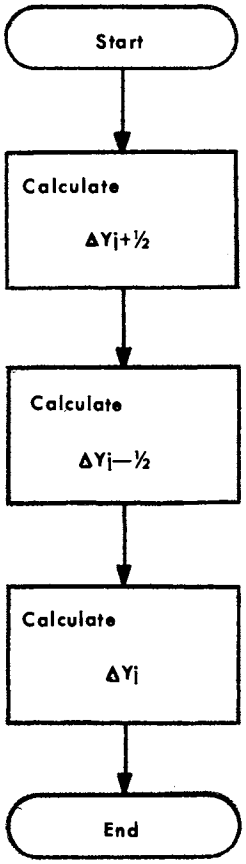


Subroutine CONTR

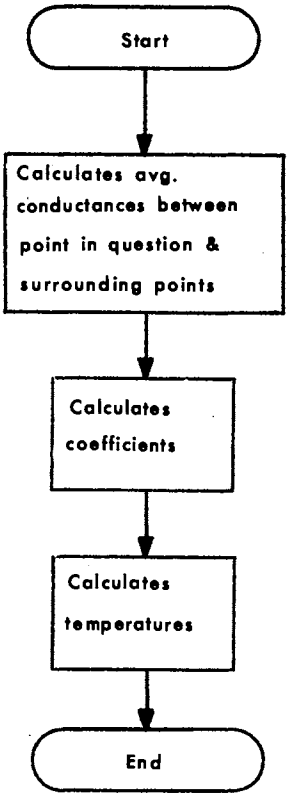


THERMAB3

Subroutine CALCY (Variable grid spacing)



Subroutine CONV



```
C                        THERMAB3-A.L. BARNES
      DIMENSION TMPS(21,51),TEMP(21,51),W(21,51),G(21,51),Y(21),V(51),YP    22241
     1LUS(21),YNEG(21),YMID(21),TTMPS(21,51),AREA(21),H(21),HA(21),VTMP(    22241
     221,51),CAPT(21,51),CAP(21,51),COND(21,51),IS(10),SLTMP(10),DIST(8,
     351),TEEP(21,51),CAPY(21,51),CAPPY(21,51)                          THERMAB3
      COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH    22241
     1A,DELTX,DENOM,DELTY,TTMPS,CAP,CAPT,TIME,LMAX,SLTMP,DIST,TEEP,CAPY,
     2CAPPY
C     TEMP=PERMANENT TEMPERATURE ARRAY                                     22241
C     VTMP=TEMPORARY TEMPERATURE ARRAY
C     TMPS=TEMPORARY TEMPERATURE ARRAY                                     22241
C     KHT IS THE SWITCH TO TURN HEAT ON OR OFF                           THERMAB6
C     KWTSP IS THE SWITCH TO PRINT INFORMATION AT THE TIME HEAT IS SHUT  THERMAB6
C          OFF                                                           THERMAB6
C     KWTST IS THE SWITCH TO PRINT INFORMATION AT THE TIME HEAT IS       THERMAB6
C          TURNED ON                                                     THERMAB6
C     TIMHT=CUMULATIVE TIME OF HEAT INJECTION                            THERMAB6
  100 READ(5,10)IMAX,JMAX,DELTT,TOL,TIMX,ALPHA,DELTX,DELTY,AA,BB,TT,TTT,    22241
     1CC,KSKIP,NRUN,KSLCE,ISSN,(IS(M),M=1,5),LMAX,(SLTMP(I),I=1,8),KSTRT
C     IMAX=NUMBER OF COLUMNS (K)
C     JMAX=NUMBER OF ROWS (J)
C     DELTT=TIME STEP LENGTH,HOURS
C     TOL=TOLERANCE
C     TIMX=MAXIMUM TIME, HOURS
C     ALPHA=SPECIFIC HEAT X VELOCITY X DENSITY(ALL OF GAS)
C     DELTX=DISTANCE BETWEEN POINTS ALONG THE FRACTURE
C     DELTY=DISTANCE BETWEEN POINTS INTO THE WALL(FIRST 3 POINTS ONLY)
C     AA=CONSTANT IN EQUATION WHICH GENERATES VELOCITY
C     BB=CONSTANT IN EQUATION WHICH GENERATES VELOCITY
C     TT=INJECTION TEMPERATURE, DEGREES FAHRENHEIT
C     TTT=RESERVOIR TEMPERATURE, DEGREES FAHRENHEIT
C     CC=CONSTANT IN EQUATION WHICH GENERATES VELOCITY
C     KSKIP=OPTION TO INCREASE TIME STEP LENGTH DURING RUN
C     NRUN=RUN NUMBER
C     KSLCE=OPTION TO CALCULATE TOTAL HEAT CONTENT AND PRINT TEMPERATURE
C          AND TOTAL HEAT CONTENT
C     ISSN=MAXIMUM NUMBER OF DESIRED SLICES FOR CROSS SECTION STUDY
C     IS(M)=VALUES OF DESIRED SLICES FOR CROSS SECTION STUDY
C     LMAX=NUMBER OF ISOTHERMS DESIRED
C     SLTMP(I)=TEMPERATURE OF DESIRED ISOTHERM
C     KSTRT=SWITCH THAT ALLOWS STARTING RUN WITH CONSTANT OR VARIABLE
C          FIELD TEMPERATURES
      READ(5,20)(Y(I),I=1,JMAX)
   20 FORMAT(16F5.0/5F5.0)
C     Y=DISTANCE INTO THE WALL, FEET
      KHT=1                                                             THERMAB6
      KWTSP=1                                                           THERMAB6
      TIMHT=0.0                                                         THERMAB6
      KWTST=1                                                           THERMAB6
      PP=1.
      PPP=1.0
      TIME=0.0                                                             22241
      SMAX=IMAX
      IIT=0
      CM=2500.
      FTSYT=(SMAX-1.)*DELTX
      ATMP=0.0
      DD=1.0
      KWHO=1
      KIT=0
      HTOUT=0.
      HTIN=0.
      DO 21 K=1,IMAX                                                       22241
      FK=K                                                                 22241
      X=DELTX*FK                                                           22241
      V(K)=(AA+BB*X+EXP(CC*X))*2.                                       THERMAB3
   21 CONTINUE                                                             22241
      DO 76 I=1,LMAX
      DO 76 K=1,IMAX
   76 DIST(I,K)=0.0
      WRITE(6,9)NRUN,TT,TTT,TOL,KSKIP,ALPHA,DELTX,(Y(I),I=1,21),AA,BB,CC
     1,TIMX,IMAX,JMAX,FTSYT
    9 FORMAT(1H1,5X,'INPUT DATA FOR THERMAB3 RUN NUMBER',2X,I5//5X,F10.1
     1,2X,'=INJECTION TEMPERATURE, DEGREES FAHRENHEIT'/5X,F10.1,2X,'=RES
     2ERVOIR TEMPERATURE, DEGREES FAHRENHEIT'/5X,F10.4,2X,'=TOLERANCE(PE
     3R CENT'/10X,I5,2X,'=OPTION TO INCREASE TIME STEP LENGTH DURING RUN
     4'/20X,'(1) INDICATES TIME STEP LENGTH REMAINED CONSTANT DURING RUN
     5'/20X,'(2) INDICATES TIME STEP LENGTH WAS INCREASED DURING RUN'/5X
     6,F10.4,2X,'=MASS X VELOCITY X DENSITY(ALL OF GAS)                 '
     7/5X,F10.1,2X,'=DISTANCE BETWEEN POINTS ALONG THE FRACTURE, FEET'//
     85X,'THE DISTANCE INTO THE WALL, FEET, IS GIVEN BELOW',/3X,21F6.1//
     95X,'THE CONSTANTS IN THE EQUATION WHICH GENERATES VELOCITY ARE'/10
     1X,F10.3,2X,'=AA'/10X,F10.3,2X,'=BB'/10X,F10.3,2X,'=CC'//5X,F10.1,2
     2X,'=MAXIMUM TIME, HOURS'/10X,I5,2X,'=IMAX'/10X,I5,2X,'=JMAX'/5X,F1
     30.1,2X,'LENGTH OF SYSTEM, FEET')
      TENTH=0.0
   10 FORMAT(2I5,F5.1,F5.2,F10.1,F10.1,6F5.1,F10.4/1015/8F10.1/I5)
      IF(KSTRT.EQ.1)GO TO 81
      READ(5,203)TIME,HTIN,HTOUT
  203 FORMAT(F10.0,2E18.8)
C     TIME=TIME, HOURS, ALREADY COMPLETED WHEN RUN RESTARTED
C     HTIN=BTU OF HEAT INJECTED WHEN RUN RESTARTED
C     HTOUT=BTU OF HEAT PRODUCED WHEN RUN RESTARTED
      TIMHT=TIME
      ATMP=TIME
      DO 74 K=1,IMAX
   74 READ(5,73)(TMPS(J,K),J=1,JMAX)
   73 FORMAT(16F5.0/5F5.0)
      GO TO 204
   81 DO 75 K=1,IMAX
      DO 75 J=1,JMAX
   75 TMPS(J,K)=TTT
  204 DO 205 K=1,IMAX
      DO 205 J=1,JMAX
      TEMP(J,K)=TMPS(J,K)
      TEEP(J,K)=TMPS(J,K)                                               THERMAB3
  205 VTMP(J,K)=TMPS(J,K)
      TMPS(1,1)=TT
      TEMP(1,1)=TT                                                         22241
      VTMP(1,1)=TT                                                         22241
      CALL CALCY                                                           22241
C     CALCY IS A SUBROUTINE THAT CALCULATES VARIABLE Y SPACING VALUES
      CALL CAPAC
C     CAPAC IS A SUBROUTINE THAT CALCULATES TOTAL HEAT REQUIREMENT
C          VALUES.
      CALL CAPSK
C     CAPSK IS A SUBROUTINE THAT CALCULATES HEAT CAPACITY VALUES
      CALL COND4
C     COND4 IS A SUBROUTINE THAT CALCULATES CONDUCTANCES
      DO 232 J=1,JMAX
      DO 232 K=1,IMAX
      CAPPY(J,K)=CAPY(J,K)
  232 CAP(J,K)=CAPT(J,K)
      FIMAX=IMAX                                                           22241
      HTORG=Y(JMAX)*DELTX*FIMAX*3149.+(CAP(1,1)-3149.)*DELTX*.25
C     HTORG=INITIAL HEAT CONTENT OF ROCK AT 100 DEGREES FAHRENHEIT
   82 GO TO(995,994),KSKIP
  994 IF(ATMP.NE.TIME) GO TO 995
      ATMP=ATMP+1.
      IF(TIME.GT.5.0)GO TO 318
      DELTT=1.
      GO TO 995
  318 IF(DELTT.GE.15.0) GO TO 993
      UU=DD-5.0
      TME=5.0*((1.2)**UU)
      DELTT=TME-TIME
      IF(DELTT.LT.15.)GO TO 995
      DELTT=15.
  993 IF(TIME.GT.1000.)GO TO 996
      DELTT=15.0
      GO TO 995
  996 IF(TIME.GT.3000.)GO TO 997
      DELTT=30.0
      GO TO 995
  997 IF(TIME.GT.10000.) GO TO 998
      DELTT=50.0
      GO TO 995
  998 DELTT=100.
C     KWHO IS THE SWITCH TO DETERMINE DIRECTION OF ADIP SWEEP(HORIZONTAL
C          OR VERTICAL)
```

```
  995 GO TO(991,53),KWHO
  991 GO TO(142,140),KHT                                                    THERMAB6
  140 GO TO(158,141),KWTSP                                                  THERMAB6
  141 KWTSP=1
C     TEMPERATURES AND HEAT DATA ARE PRINTED IF HEAT INJECTION HAS JUST
C          STOPPED
      WRITE(6,33)
      DO 46 K=1,42                                                          THERMAB6
   46 WRITE(6,37)K,(TEMP(J,K),J=1,17)                                       THERMAB6
      WRITE(6,250)                                                          THERMAB6
  250 FORMAT(1H1,2X,'J=    18',5X,'19',5X,'20',5X,'21')                     THERMAB6
      DO 251 K=1,42                                                         THERMAB6
  251 WRITE(6,252)K,(TEMP(J,K),J=18,JMAX)                                   THERMAB6
  252 FORMAT(I4,4F7.1)                                                      THERMAB6
      WRITE(6,33)                                                           THERMAB6
      DO 44 K=43,IMAX                                                       THERMAB6
   44 WRITE(6,37)K,(TEMP(J,K),J=1,17)                                       THERMAB6
      WRITE(6,250)                                                          THERMAB6
      DO 253 K=43,IMAX                                                      THERMAB6
  253 WRITE(6,252)K,(TEMP(J,K),J=18,JMAX)                                   THERMAB6
      WRITE(6,30)TIME,ERR,HEAT,HTOUT,HTIN,HTTOT,HTORG,DELTT,ER1,ER2,TOL THERMAB6
      WRITE(6,150)TIMHT,KWHO
  150 FORMAT(1H-,2X,'HEAT INJECTION STOPS, PRODUCING WELL TEMPERATURE   THERMAB6
     1EXCEEDS 800 DEGREES F.'/2X,'CUMULATIVE TIME OF HEAT INJECTION EQUATHERMAB6
     2LS',2X,F10.1,I5)
      CALL CONTR
C     CONTR IS A SUBROUTINE THAT CALCULATES THE POSITION OF SELECTED
C          ISOTHERMS
  158 CALL COND6
C     COND6 IS A SUBROUTINE FOR THE HORIZONTAL ADIP SOLUTION WITH
C          ADIABATIC BOUNDARY CONDITIONS
      GO TO 45
  142 GO TO(143,159),KWTST                                                  THERMAB6
  159 KWTST=1                                                               THERMAB6
C     TEMPERATURES AND HEAT DATA ARE PRINTED IF HEAT INJECTION HAS JUST
C          STARTED
      WRITE(6,33)
      DO 32 K=1,42
   32 WRITE(6,37)K,(TEMP(J,K),J=1,17)
      WRITE(6,250)
      DO 275 K=1,42
  275 WRITE(6,252)K,(TEMP(J,K),J=18,JMAX)
      WRITE(6,33)
      DO 36 K=43,IMAX
   36 WRITE(6,37)K,(TEMP(J,K),J=1,17)
      WRITE(6,250)
      DO 276 K=43,IMAX
  276 WRITE(6,252)K,(TEMP(J,K),J=18,JMAX)
      WRITE(6,30)TIME,ERR,HEAT,HTOUT,HTIN,HTTOT,HTORG,DELTT,ER1,ER2,TOL THERMAB6
      WRITE(6,255)
      CALL CONTR
      GO TO 143
   53 GO TO(146,145),KHT                                                    THERMAB6
  145 GO TO(170,261),KWTSP
  261 KWTSP=1
C     TEMPERATURES AND HEAT DATA ARE PRINTED IF HEAT INJECTION HAS JUST
C          STOPPED
      WRITE(6,33)
      DO 31 K=1,42
   31 WRITE(6,37)K,(TEMP(J,K),J=1,17)
      WRITE(6,250)
      DO 271 K=1,42
  271 WRITE(6,252)K,(TEMP(J,K),J=18,JMAX)
      WRITE(6,33)
      DO 34 K=43,IMAX
   34 WRITE(6,37)K,(TEMP(J,K),J=1,17)
      WRITE(6,250)
      DO 272 K=43,IMAX
  272 WRITE(6,252)K,(TEMP(J,K),J=18,JMAX)
      WRITE(6,30)TIME,ERR,HEAT,HTOUT,HTIN,HTTOT,HTORG,DELTT,ER1,ER2,TOL THERMAB6
      WRITE(6,150)TIMHT                                                     THERMAB6
      CALL CONTR
  170 CALL COND7
C     COND7 IS A SUBROUTINE FOR THE VERTICAL ADIP SOLUTION WITH
C          ADIABATIC BOUNDARY CONDITIONS
      GO TO 171
  146 GO TO(43,157),KWTST
  157 KWTST=1                                                               THERMAB6
C     TEMPERATURES AND HEAT DATA ARE PRINTED IF HEAT INJECTION HAS JUST
C          STARTED
      WRITE(6,33)
      DO 38 K=1,42
   38 WRITE(6,37)K,(TEMP(J,K),J=1,17)
      WRITE(6,250)
      DO 278 K=1,42
  278 WRITE(6,252)K,(TEMP(J,K),J=18,JMAX)
      WRITE(6,33)
      DO 39 K=43,IMAX
   39 WRITE(6,37)K,(TEMP(J,K),J=1,17)
      WRITE(6,250)
      DO 279 K=43,IMAX
  279 WRITE(6,252)K,(TEMP(J,K),J=18,JMAX)
      WRITE(6,30)TIME,ERR,HEAT,HTOUT,HTIN,HTTOT,HTORG,DELTT,ER1,ER2,TOL THERMAB6
      WRITE(6,255)KWHO
  255 FORMAT(1H-,2X,'HEAT INJECTION STARTS. PRODUCING WELL TEMPERATURE ITHERMAB6
     1S LESS THAN 700 DEGREES F.',I5)
      CALL CONTR
      GO TO 43                                                              THERMAB6
  143 CALL CONV
C     CONV IS A SUBROUTINE THAT SOLVES EXPLICIT CONVECTION EQUATIONS
C          ALONG THE FRACTURE
      CALL COND5
C     COND5 IS A SUBROUTINE THAT CALCULATES NEW CONDUCTANCE VALUES
C          ALONG THE J=1 ROW
      CALL CAPC1
C     CAPC1 IS A SUBROUTINE THAT CALCULATES NEW TOTAL HEAT REQUIREMENT
C          VALUES ALONG THE J=1 ROW
   45 GO TO(144,47),KHT
  144 CALL COND1                                                            22241
C     COND1 IS A SUBROUTINE THAT SOLVES HORIZONTAL ADIP WITH THE CONV
C          SOLUTION AS A BOUNDARY CONDITION
   47 IF(DENOM.EQ.0.0)GO TO 69                                              22241
      IIT=IIT+1
      KIT=KIT+1
      IF(IIT.EQ.5)GO TO 296
      IF(KIT.GT.16)GO TO 222
      KK=1
      DO 84 N=1,JMAX
      DO 84 M=1,IMAX                                                        22241
      IF(TMPS(N,M).LT.100.10) GO TO 84
      IF(ABS((VTMP(N,M)-TMPS(N,M))/TMPS(N,M)).GT.TOL)GO TO 86               22241
   84 CONTINUE                                                              22241
      GO TO 67                                                              22241
   86 DO 88 J=1,JMAX                                                        22241
      DO 88 K=1,IMAX                                                        22241
   88 VTMP(J,K)=TMPS(J,K)                                                   22241
      KWHO=2
      GO TO 82
  296 DO 297 K=1,IMAX
      DO 297 J=1,JMAX
      TMPS(J,K)=TEMP(J,K)
  297 VTMP(J,K)=TEMP(J,K)
      IIT=0
      CM=CM+5000.
      TOL=TOL+.01
      GO TO 82
   43 CALL CONV
  147 CALL COND5
      CALL CAPC1
  171 GO TO(149,148),KHT
  149 CALL COND2                                                            22241
C     COND2 IS A SUBROUTINE THAT SOLVES VERTICAL ADIP WITH THE CONV
C          SOLUTION AS A BOUNDARY CONDITION
  148 IF(DENOM.EQ.0.0)GO TO 69                                              22241
      KK=2                                                                  22241
      DO 64 N=1,JMAX
      DO 64 M=1,IMAX                                                        22241
      IF(TMPS(N,M).LT.100.10) GO TO 64
      IF(ABS((VTMP(N,M)-TMPS(N,M))/TMPS(N,M)).GT.TOL)GO TO 65               22241
   64 CONTINUE                                                              22241
      GO TO 67                                                              22241
   65 DO 83 J=1,JMAX                                                        22241
      DO 83 K=1,IMAX                                                        22241
   83 VTMP(J,K)=TMPS(J,K)                                                   22241
      KWHO=1
      GO TO 82                                                              22241
   67 DO 66 J=1,JMAX                                                        22241
      DO 66 K=1,IMAX                                                        22241
      IF(TMPS(J,K).LT.100.10) GO TO 66
      CAP(J,K)=CAPT(J,K)
      TEEP(J,K)=TEMP(J,K)
```

```
      CAPPY(J,K)=CAPY(J,K)
      TEMP(J,K)=TMPS(J,K)
   66 CONTINUE
      GO TO(152,153),KHT
  152 TIMHT=TIMHT+DELTT
      HTOUT=HTOUT+TEMP(1,IMAX)*DELTT*ALPHA
      HTIN=HTIN+ALPHA*TT*DELTT
  153 IF(TEMP(1,IMAX).LE.900.)GO TO 48
      IF(KHT.EQ.2) GO TO 174                                          THERMAB6
      KWTSP=2                                                         THERMAB6
  174 KHT=2                                                           THERMAB6
   48 IF(TEMP(1,IMAX).GE.800.)GO TO 49
      TEMP(1,1)=TT
      TMPS(1,1)=TT
      VTMP(1,1)=TT
      IF(KHT.EQ.1) GO TO 175
      KWTST=2                                                         THERMAB6
  175 KHT=1                                                           THERMAB6
      IF(KHT.EQ.2)GO TO 183
   49 PPP=PPP+1.
      IF(PPP.LT.19.)GO TO 741
      IF(PPP.EQ.19.)GO TO 381
      IF(PPP.NE.20.)GO TO 741
      PPP=0.0
  381 JBMAX=JMAX-2                                                    22241
      IBMAX=IMAX-2                                                    22241
C     THIS ACCOUNTS FOR THE HEAT STORED IN ROCK                       22241
      DO 40 J=1,JMAX                                                  22241
      SUM=0.                                                          22241
      DO 80 K=1,IBMAX,2                                               22241
   80 SUM=SUM+CAP(J,K)+4.*CAP(J,K+1)+CAP(J,K+2)
   40 AREA(J)=SUM*(DELTX/3.)                                          22241
      BB2=0.                                                          22241
      DO 90 J=1,JBMAX,2                                               22241
      H(J)=Y(J+1)-Y(J)                                                22241
      HA(J)=(Y(J+2)-Y(J+1))/H(J)                                      22241
   90 BB2=BB2+(H(J)*(HA(J)+1.)/(6.*HA(J)))*((2.*HA(J)-1.)*AREA(J+2)+(HA(
     1J)+1.)**2*AREA(J+1)+HA(J)*(2.-HA(J))*AREA(J))                   22241
      ER2=((BB2-HEAT-HTORG)+(TEMP(1,IMAX)*DELTT*ALPHA))/(ALPHA*TT*DELTT)
      HETIN=(BB2-HEAT-HTORG)/(ALPHA*TT*DELTT)
      HEAT=BB2-HTORG
C     THIS ACCOUNTS FOR HEAT OUT OF PRODUCER                          22241
  741 HTTOT=HTOUT+HEAT+HTORG
      HTINN=HTIN+HTORG
      ERR=ABS((HTINN-HTTOT)/HTTOT)
      ER1=ABS((HTINN-HTTOT)/HTINN)
  183 GO TO(4,5),KSLCE                                                THERMAB3
C     THIS ACCOUNTS FOR THE TEMPERATURE AND CULULATIVE HEAT PRINT OUT AT
C     VARIOUS SELECTED POSITIONS ALONG THE FRACTURE.
    4 WRITE(6,26)                                                     THERMAB3
   26 FORMAT(1H1,30X,'INPUT DATA FOR CROSS SECTION STUDY'//9X,'LOCATION(THERMAB3
     1FEET DOWN FRACTURE)',2X,'TEMPERATURE',9X,'TIME(HOURS)',9X,'CUMULATTHERMAB3
     2IVE HEAT'//)                                                    THERMAB3
      JBMAX=JMAX-2                                                    THERMAB3
      DO 70 K=1,IMAX                                                  THERMAB3
      DO 70 M=1,ISSN                                                  THERMAB3
      IF(K.NE.IS(M+1))GO TO 70
      BB3=0.0                                                         THERMAB3
      DO 71 J=1,JBMAX,2                                               THERMAB3
      H(J)=Y(J+1)-Y(J)                                                THERMAB3
      HA(J)=(Y(J+2)-Y(J+1))/H(J)                                      22241
   71 BB3=BB3+(H(J)*(HA(J)+1.)/(6.*HA(J)))*((2.*HA(J)-1.)*CAP(J+2,K)+(H
     1A(J)+1.)**2*CAP(J+1,K)+HA(J)*(2.-HA(J))*CAP(J,K)))*DELTX
      AIS=IS(M)
      ALOCF=DELTX*AIS
      WRITE(6,25)ALOCF,TEMP(1,K),TIME,BB3                             THERMAB3
   25 FORMAT(9X,F10.1,20X,F10.1,10X,F10.1,7X,E18.8)                   THERMAB3
   70 CONTINUE                                                        THERMAB3
    5 TIME=TIME+DELTT                                                 22241
      IF(TIME.GT.3000.)GO TO 56
      TOL=.05
      GO TO 55
   56 IF(TIME.GT.4000.)GO TO 57
      TOL=.04
      GO TO 55
   57 IF(TIME.GT.5000.)GO TO 58
      TOL=.03
      GO TO 55
   58 IF(TIME.GT.CM)GO TO 59
      TOL=.02
      GO TO 55
   59 TOL=.01
   55 ATMP=TIME
      DD=DD+1.
      PP=PP+1.
      IF(PP.NE.20.) GO TO 891
      PP=0.
      WRITE(6,33)                                                     22241
   33 FORMAT(1H1,2X2HJ=4X2H 15X2H 25X2H 35X2H 45X2H 55X2H 65X2H 75X2H 85   22241
     1X2H 95X2H105X2H115X2H125X2H135X2H145X2H15)                      22241
      DO 92 K=1,42
   92 WRITE(6,37)K,(TEMP(J,K),J=1,17)
   37 FORMAT(I4,17F7.1)                                               22241
      IF(TEMP(18,1).LT.100.1)GO TO 751
      WRITE(6,250)
      DO 290 K=1,42
  290 WRITE(6,252)K,(TEMP(J,K),J=18,JMAX)
  751 IF(TEMP(1,43).LT.100.1)GO TO 752
      WRITE(6,33)                                                     THERMAB3
      DO 24 K=43,IMAX                                                 THERMAB3
   24 WRITE(6,37)K,(TEMP(J,K),J=1,17)
  752 IF(TEMP(18,43).LT.100.1)GO TO 887
      WRITE(6,250)
      DO 299 K=43,IMAX
  299 WRITE(6,252)K,(TEMP(J,K),J=18,JMAX)
  887 WRITE(6,30)TIME,ERR,HEAT,HTOUT,HTIN,HTTOT,HTORG,DELTT,ER1,ER2,TOL,
     1HETIN
   30 FORMAT(F10.1,2X,5HHOURS/F10.5,2X,27HDISCREPANCY IN HEAT BALANCE/E1   22241
     18.8,2X,31HBTU OF HEAT STORED IN RESERVOIR/E18.8,2X,20HBTU OF HEAT    22241
     2PRODUCED/E18.8,2X,20HBTU OF HEAT INJECTED/E18.8,2X,5HHTTOT/E18.8,2   22241
     3X,5HHTORG/F10.1,21HTIME INCREMENT, HOURS/F10.5,3X,3HER1/F10.5,3X,3
     4HER2/F10.5,3X,9HTOLERANCE/E18.8,2X,'*=INSTANTEOUS HEAT STORED')
      WRITE(6,223)DD,KIT
C     INTERPOLATING ROUTINE FOR DETERMING POSITION OF SELECTED ISOTHERMS
      CALL CONTR
  891 KIT=0
      IIT=0
      IF(TIME.GT.TIMX)GO TO 100
      GO TO(86,65),KK                                                 22241
  222 WRITE(6,223)DD,KIT
  223 FORMAT(3X,F10.0,I5)
   69 STOP                                                            22241
      END                                                             22241
      SUBROUTINE CALCV
      DIMENSION TMPS(21,51),TEMP(21,51),W(21,51),G(21,51),Y(21),V(51),YP   22241
     1LUS(21),YNEG(21),YMID(21),TTMPS(21,51),AREA(21),H(21),HA(21),VTMP(   22241
     221,51),CAPT(21,51),CAP(21,51),COND(21,51),IS(10),SLTMP(10),DIST(8,
     351),TEEP(21,51),CAPY(21,51),CAPPY(21,51)                        THERMAB3
      COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH    22241
     1A,DELTX,DENOM,DELTY,TTMPS,CAP,CAPT,TIME,LMAX,SLTMP,DIST,TEEP,CAPY,
     2CAPPY
      NMAX=JMAX-1                                                     22241
      DO 2 J=1,NMAX                                                   22241
    2 YPLUS(J)=Y(J+1)-Y(J)                                            22241
      DO 3 J=2,JMAX                                                   22241
    3 YNEG(J)=Y(J)-Y(J-1)                                             22241
      YNEG(1)=YPLUS(1)                                                22241
      YPLUS(JMAX)=YNEG(JMAX)                                          22241
      DO 4 J=1,JMAX                                                   22241
    4 YMID(J)=0.5*(YPLUS(J)+YNEG(J))                                  22241
      RETURN                                                          22241
      END                                                             22241
      SUBROUTINE CONV
      DIMENSION TMPS(21,51),TEMP(21,51),W(21,51),G(21,51),Y(21),V(51),YP   22241
     1LUS(21),YNEG(21),YMID(21),TTMPS(21,51),AREA(21),H(21),HA(21),VTMP(   22241
     221,51),CAPT(21,51),CAP(21,51),COND(21,51),IS(10),SLTMP(10),DIST(8,
     351),TEEP(21,51),CAPY(21,51),CAPPY(21,51)                        THERMAB3
      COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH    22241
     1A,DELTX,DENOM,DELTY,TTMPS,CAP,CAPT,TIME,LMAX,SLTMP,DIST,TEEP,CAPY,
     2CAPPY
```

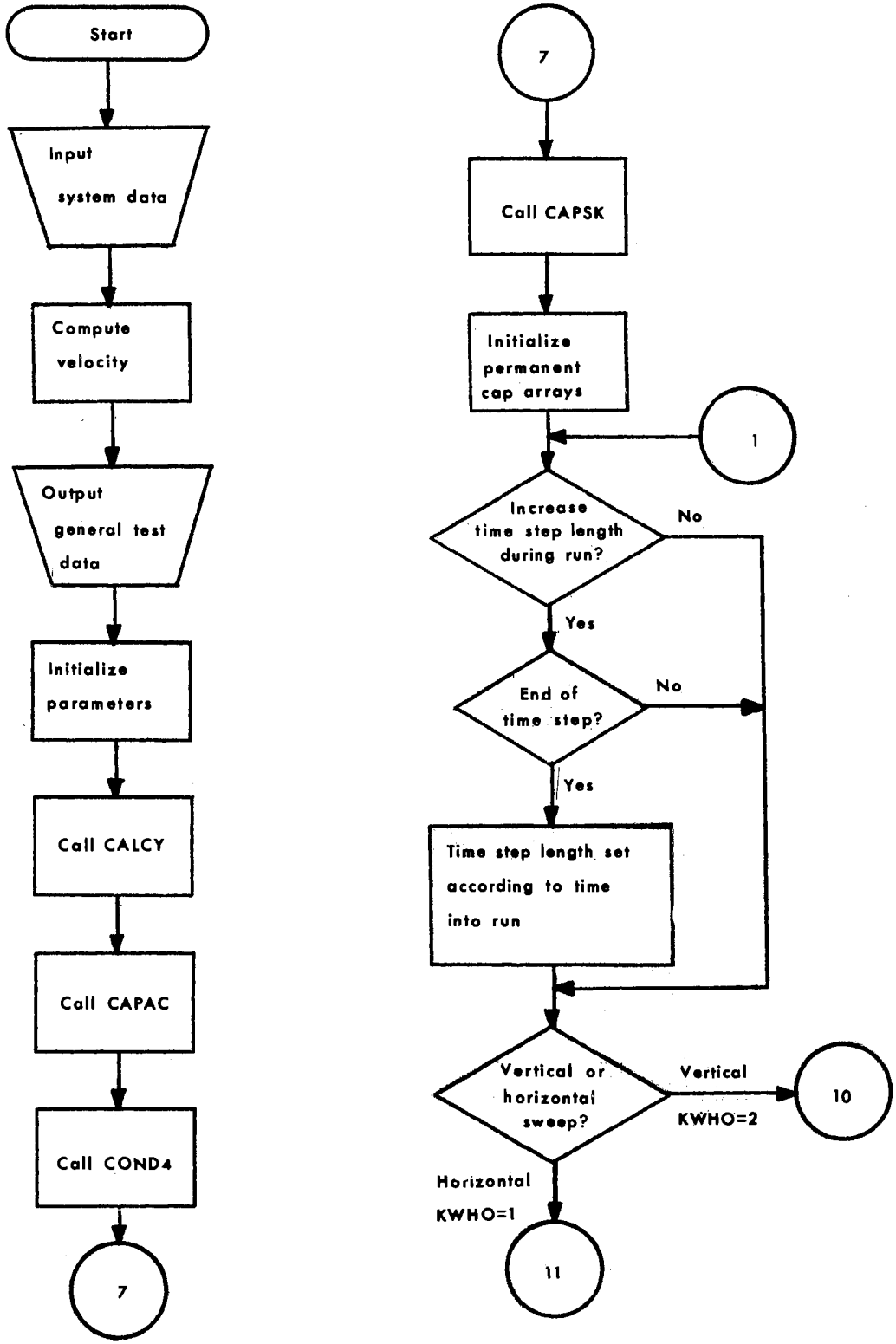
```
      DO 40 K=2,IMAX                                                    22241
      CONMIN=(COND(1,K)+COND(2,K))/2.
      CONPUS=(COND(2,K)+COND(3,K))/2.
      D13=DELTX*TEMP(1,K)/(DELTT*V(K))
      D14=(3.*DELTX*CONMIN*TMPS(2,K))/(2.*DELTY*ALPHA)
      D15=(DELTX*CONPUS*(TMPS(2,K)-TMPS(3,K)))/(2.*DELTY*ALPHA)
      D16=1.+(DELTX/(DELTT*V(K)))
      D17=(3.*DELTX*CONMIN)/(2.*DELTY*ALPHA)
      TMPS(1,K)=(TMPS(1,K-1)+D13+D14+D15)/(D16+D17)
      IF(TMPS(1,K).GE.100.)GO TO 40
      TMPS(1,K)=100.
   40 CONTINUE                                                          22241
      RETURN                                                            22241
      END                                                               22241
      SUBROUTINE COND1
      DIMENSION TMPS(21,51),TEMP(21,51),W(21,51),G(21,51),Y(21),V(51),YP  22241
     1LUS(21),YNEG(21),YMID(21),TTMPS(21,51),AREA(21),H(21),HA(21),VTMP(  22241
     221,51),CAPT(21,51),CAP(21,51),COND(21,51),IS(10),SLTMP(10),DIST(8,
     351),TEEP(21,51),CAPY(21,51),CAPPY(21,51)                      THERMAB3
      COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH   22241
     1A,DELTX,DENOM,DELTY,TTMPS,CAP,CAPT,TIME,LMAX,SLTMP,DIST,TEEP,CAPY,
     2CAPPY
      DO 1 J=2,JMAX                                                     22241
      DO 20 K=1,IMAX                                                    22241
   51 IF(J.EQ.JMAX)GO TO 70                                             22240
      IF(K.EQ.1)GO TO 75                                                22240
      IF(K.EQ.IMAX)GO TO 76                                             22240
      CXPUS=(COND(J,K)+COND(J,K+1))/2.                                  22240
      CXMIN=(COND(J,K)+COND(J,K-1))/2.                                  22240
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                  22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22240
      GO TO 77                                                          22240
   70 IF(K.EQ.1)GO TO 72                                                22240
      IF(K.EQ.IMAX)GO TO 73                                             22240
      CXPUS=(COND(J,K)+COND(J,K+1))/2.                                  22240
      CXMIN=(COND(J,K)+COND(J,K-1))/2.                                  22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22240
      CYPUS=CYMIN                                                       22240
      GO TO 77                                                          22240
   72 CXPUS=(COND(J,K)+COND(J,K+1))/2.                                  22240
      CXMIN=CXPUS                                                       22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22240
      CYPUS=CYMIN                                                       22240
      GO TO 77                                                          22240
   73 CXMIN=(COND(J,K)+COND(J,K-1))/2.                                  22240
      CXPUS=CXMIN                                                       22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22240
      CYPUS=CYMIN                                                       22240
      GO TO 77                                                          22240
   75 CXPUS=(COND(J,K)+COND(J,K+1))/2.                                  22240
      CXMIN=CXPUS                                                       22240
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                  22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22240
      GO TO 77                                                          22240
   76 CXMIN=(COND(J,K)+COND(J,K-1))/2.                                  22240
      CXPUS=CXMIN                                                       22240
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                  22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22240
   77 CONTINUE                                                          22240
      B1=-(1.+CXMIN/CXPUS+(DELTX**2/DELTT)*(CAPT(J,K)/(TMPS(J,K)*CXPUS))
     1)
      D14=-(DELTX**2/(YMID(J)*CXPUS))                                   22240
      IF(J.LT.JMAX)GO TO 22                                             22241
      D15=CYPUS*((TMPS(J-1,K)-TMPS(J,K))/YPLUS(J))                      22240
      GO TO 90                                                          22241
   22 D15=CYPUS*((TMPS(J+1,K)-TMPS(J,K))/YPLUS(J))                      22240
   90 D16=CYMIN*((TMPS(J,K)-TMPS(J-1,K))/YNEG(J))                       22240
      D17=(DELTX**2*CAP(J,K))/(DELTT*CXPUS)
      D1=D14*(D15-D16)-D17
      IF(K.NE.1)GO TO 3
      A1=0.0
      C1=1.+CXMIN/CXPUS                                                 22240
      W(J,1)=C1/B1
      G(J,1)=D1/B1
      GO TO 20
    3 IF(K.NE.IMAX)GO TO 5                                              22241
      C1=0.0
      A1=1.+CXMIN/CXPUS                                                 22240
      GO TO 10
    5 A1=CXMIN/CXPUS                                                    22240
      C1=1.0
   10 DENOM=B1-A1*W(J,K-1)                                              22241
      IF(DENOM.NE.0.)GO TO 18                                           22241
      WRITE(6,50)
   50 FORMAT(18H ZERO DENOMINATOR.)                                     22241
      GO TO 800
   18 W(J,K)=C1/DENOM
      G(J,K)=(D1-A1*G(J,K-1))/DENOM                                     22241
   20 CONTINUE
      TTMPS(J,IMAX)=G(J,IMAX)
      DO 30 I=2,IMAX
      II=IMAX+1-I
   30 TTMPS(J,II)=G(J,II)-W(J,II)*TTMPS(J,II+1)
    1 CONTINUE
      DO 95 J=2,JMAX
      DO 95 K=1,IMAX
      TMPS(J,K)=TTMPS(J,K)
   95 CONTINUE
      CALL CAPAC
      CALL COND4
  800 RETURN
      END
      SUBROUTINE COND2
      DIMENSION TMPS(21,51),TEMP(21,51),W(21,51),G(21,51),Y(21),V(51),YP   22241
     1LUS(21),YNEG(21),YMID(21),TTMPS(21,51),AREA(21),H(21),HA(21),VTMP(   22241
     221,51),CAPT(21,51),CAP(21,51),COND(21,51),IS(10),SLTMP(10),DIST(8,
     351),TEEP(21,51),CAPY(21,51),CAPPY(21,51)                      THERMAB3
      COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH    22241
     1A,DELTX,DENOM,DELTY,TTMPS,CAP,CAPT,TIME,LMAX,SLTMP,DIST,TEEP,CAPY,
     2CAPPY
      DO 1 K=1,IMAX
      DO 20 J=2,JMAX
   61 IF(J.EQ.JMAX)GO TO 70                                             22240
      IF(K.EQ.1)GO TO 75                                                22240
      IF(K.EQ.IMAX)GO TO 76                                             22240
      CXPUS=(COND(J,K)+COND(J,K+1))/2.                                  22240
      CXMIN=(COND(J,K)+COND(J,K-1))/2.                                  22240
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                  22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22240
      GO TO 77                                                          22240
   70 IF(K.EQ.1)GO TO 72                                                22240
      IF(K.EQ.IMAX)GO TO 73                                             22240
      CXPUS=(COND(J,K)+COND(J,K+1))/2.                                  22240
      CXMIN=(COND(J,K)+COND(J,K-1))/2.                                  22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22240
      CYPUS=CYMIN                                                       22240
      GO TO 77                                                          22240
   72 CXPUS=(COND(J,K)+COND(J,K+1))/2.                                  22240
      CXMIN=CXPUS                                                       22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22240
      CYPUS=CYMIN                                                       22240
      GO TO 77                                                          22240
   73 CXMIN=(COND(J,K)+COND(J,K-1))/2.                                  22240
      CXPUS=CXMIN                                                       22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22240
      CYPUS=CYMIN                                                       22240
      GO TO 77                                                          22240
   75 CXPUS=(COND(J,K)+COND(J,K+1))/2.                                  22240
      CXMIN=CXPUS                                                       22240
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                  22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22240
      GO TO 77                                                          22240
   76 CXMIN=(COND(J,K)+COND(J,K-1))/2.                                  22240
      CXPUS=CXMIN                                                       22240
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                  22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22240
   77 CONTINUE                                                          22240
      A1=(YPLUS(J)*CYMIN)/(CYPUS*YNEG(J))                               22240
      B1=-(1.0+(YPLUS(J)*CYMIN)/(YNEG(J)*CYPUS)+(CAPT(J,K)*YMID(J)*YPLUS
     1(J))/(DELTT*CYPUS*TMPS(J,K)))
      D14=-(YMID(J)*YPLUS(J)/(DELTX**2*CYPUS))
      IF(K.LT.IMAX) GO TO 24
      D15=(TMPS(J,K-1)-TMPS(J,K))*CXPUS
      GO TO 25
```

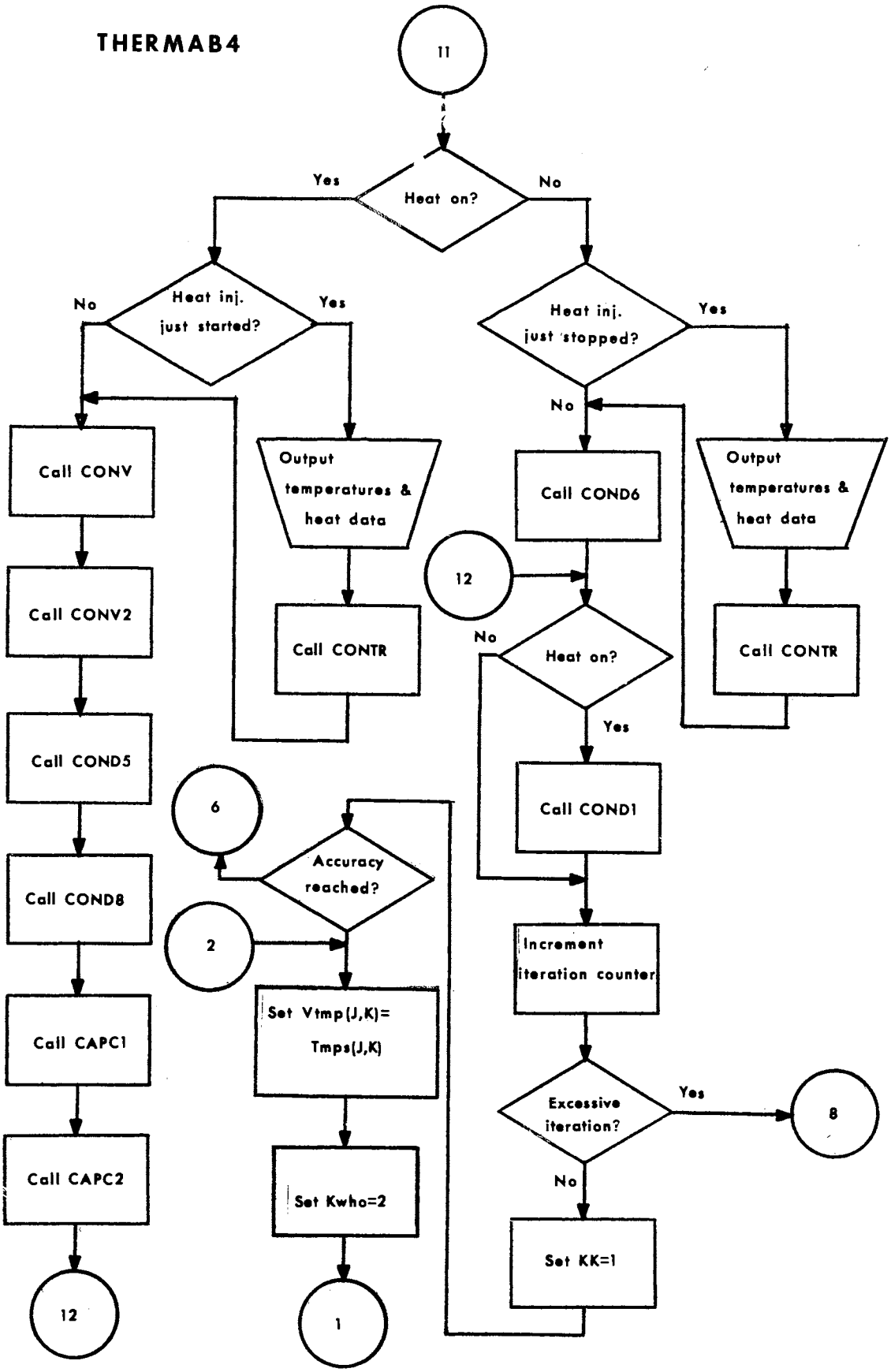
```
   D15=(TMPS(J,K+1)-TMPS(J,K))*CXPUS
   IF(K.GT.1) GO TO 26
   D16=(TMPS(J,K)-TMPS(J,K+1))*CXMIN
   GO TO 27
26 D16=(TMPS(J,K)-TMPS(J,K-1))*CXMIN
27 D17=(CAP(J,K)*YMID(J)*YPLUS(J))/(DELTT*CYPUS)
   D1=D14*(D15-D16)-D17                                                  22241
   IF(J.NE.2)GO TO 3
   W(2,K)=C1/B1
   G(2,K)=(D1-A1*TMPS(1,K))/B1
   GO TO 20
 3 IF(J.NE.JMAX)GO TO 10                                                 22241
   A1=A1+C1
   C1=0.0
10 DENOM=B1-A1*W(J-1,K)
   IF(DENOM.NE.0.)GO TO 18
   WRITE(6,50)
50 FORMAT(18H ZERO DENOMINATOR.)                                         22241
   GO TO 800
18 W(J,K)=C1/DENOM                                                       22241
   G(J,K)=(D1-A1*G(J-1,K))/DENOM                                         22241
20 CONTINUE
   TTMPS(JMAX,K)=G(JMAX,K)
   NMAX=JMAX-2
   DO 30 I=1,NMAX
   II=JMAX-I
30 TTMPS(II,K)=G(II,K)-W(II,K)*TTMPS(II+1,K)
 1 CONTINUE
   DO 95 J=2,JMAX
   DO 95 K=1,IMAX
   TMPS(J,K)=TTMPS(J,K)
95 CONTINUE
   CALL CAPAC
   CALL COND4
800 RETURN
   END
   SUBROUTINE CAPAC
   DIMENSION TMPS(21,51),TEMP(21,51),W(21,51),G(21,51),Y(21),V(51),YP    22241
  1LUS(21),YNEG(21),YMID(21),TTMPS(21,51),AREA(21),H(21),HA(21),VTMP(    22241
  221,51),CAPT(21,51),CAP(21,51),COND(21,51),IS(10),SLTMP(10),DIST(8,
  351),TEEP(21,51),CAPY(21,51),CAPPY(21,51)                             THERMAB3
   COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH    22241
  1A,DELTX,DENOM,DELTY,TTMPS,CAP,CAPT,TIME,LMAX,SLTMP,DIST,TEEP,CAPY,
  2CAPPY
   DO 232 J=1,JMAX
   DO 232 K=1,IMAX
 9 IF(TMPS(J,K).GT.450.)GO TO 233
   CAPT(J,K)=(23.5+.307*(TMPS(J,K)-100.))*134.0
   GO TO 232
233 IF(TMPS(J,K).GT.900.)GO TO 234
   CAPT(J,K)=(130.0+.344*(TMPS(J,K)-450.))*134.0
   GO TO 232
234 IF(TMPS(J,K).GT.1100.)GO TO 235
   CAPT(J,K)=134.0*(285.0+.485*(TMPS(J,K)-900.))
   GO TO 232
235 IF(TMPS(J,K).GT.1600.)GO TO 236
   CAPT(J,K)=134.0*(382.0+.650D*(TMPS(J,K)-1100.))
   GO TO 232
236 CAPT(J,K)=134.0*(694.0+.268*(TMPS(J,K)-1600.))
232 CONTINUE
   RETURN
   END
   SUBROUTINE COND4
   DIMENSION TMPS(21,51),TEMP(21,51),W(21,51),G(21,51),Y(21),V(51),YP    22241
  1LUS(21),YNEG(21),YMID(21),TTMPS(21,51),AREA(21),H(21),HA(21),VTMP(    22241
  221,51),CAPT(21,51),CAP(21,51),COND(21,51),IS(10),SLTMP(10),DIST(8,
  351),TEEP(21,51),CAPY(21,51),CAPPY(21,51)                             THERMAB3
   COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH    22241
  1A,DELTX,DENOM,DELTY,TTMPS,CAP,CAPT,TIME,LMAX,SLTMP,DIST,TEEP,CAPY,
  2CAPPY
   DO 5 J=1,JMAX                                                        22242
   DO 5 K=1,IMAX                                                        22242
 9 IF(TMPS(J,K).GE.200.)GO TO 1
   COND(J,K)=0.9
   GO TO 5
 1 IF(TMPS(J,K).GT.1200.)GO TO 7
   TSQ=TMPS(J,K)*TMPS(J,K)
   TCUB=TMPS(J,K)*TSQ
   COND(J,K)=.12561573E+1-.18377799E-2*TMPS(J,K)+.10902421E-5*TSQ-.15
  1084869E-9*TCUB
   GO TO 5
 7 COND(J,K)=0.36
 5 CONTINUE
   RETURN                                                               22242
   END                                                                  22242
   SUBROUTINE COND5
   DIMENSION TMPS(21,51),TEMP(21,51),W(21,51),G(21,51),Y(21),V(51),YP    22241
  1LUS(21),YNEG(21),YMID(21),TTMPS(21,51),AREA(21),H(21),HA(21),VTMP(    22241
  221,51),CAPT(21,51),CAP(21,51),COND(21,51),IS(10),SLTMP(10),DIST(8,
  351),TEEP(21,51),CAPY(21,51),CAPPY(21,51)                             THERMAB3
   COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH    22241
  1A,DELTX,DENOM,DELTY,TTMPS,CAP,CAPT,TIME,LMAX,SLTMP,DIST,TEEP,CAPY,
  2CAPPY
   DO 5 K=1,IMAX                                                        22242
   IF(TMPS(1,K).GE.200.)GO TO 1
   COND(1,K)=0.9
   GO TO 5
 1 IF(TMPS(1,K).GT.1200.)GO TO 7
   TSQ=TMPS(1,K)*TMPS(1,K)
   TCUB=TMPS(1,K)*TSQ
   COND(1,K)=.12561573E+1-.18377799E-2*TMPS(1,K)+.10902421E-5*TSQ-.15
  1084869E-9*TCUB
   GO TO 5
 7 COND(1,K)=0.36
 5 CONTINUE
   RETURN                                                               22242
   END                                                                  22242
   SUBROUTINE CAPC1
   DIMENSION TMPS(21,51),TEMP(21,51),W(21,51),G(21,51),Y(21),V(51),YP    22241
  1LUS(21),YNEG(21),YMID(21),TTMPS(21,51),AREA(21),H(21),HA(21),VTMP(    22241
  221,51),CAPT(21,51),CAP(21,51),COND(21,51),IS(10),SLTMP(10),DIST(8,
  351),TEEP(21,51),CAPY(21,51),CAPPY(21,51)                             THERMAB3
   COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH    22241
  1A,DELTX,DENOM,DELTY,TTMPS,CAP,CAPT,TIME,LMAX,SLTMP,DIST,TEEP,CAPY,
  2CAPPY
   DO 232 K=1,IMAX
   IF(TMPS(1,K).GT.450.)GO TO 233
   CAPT(1,K)=(23.5+.307*(TMPS(1,K)-100.))*134.0
   GO TO 232
233 IF(TMPS(1,K).GT.900.)GO TO 234
   CAPT(1,K)=(130.0+.344*(TMPS(1,K)-450.))*134.0
   GO TO 232
234 IF(TMPS(1,K).GT.1100.)GO TO 235
   CAPT(1,K)=134.0*(285.0+.485*(TMPS(1,K)-900.))
   GO TO 232
235 IF(TMPS(1,K).GT.1600.)GO TO 236
   CAPT(1,K)=134.0*(382.0+.650D*(TMPS(1,K)-1100.))
   GO TO 232
236 CAPT(1,K)=134.0*(694.0+.268*(TMPS(1,K)-1600.))
232 CONTINUE
   RETURN
   END
   SUBROUTINE COND7
   DIMENSION TMPS(21,51),TEMP(21,51),W(21,51),G(21,51),Y(21),V(51),YP    22241
  1LUS(21),YNEG(21),YMID(21),TTMPS(21,51),AREA(21),H(21),HA(21),VTMP(    22241
  221,51),CAPT(21,51),CAP(21,51),COND(21,51),IS(10),SLTMP(10),DIST(8,
  351),TEEP(21,51),CAPY(21,51),CAPPY(21,51)                             THERMAB3
   COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH    22241
  1A,DELTX,DENOM,DELTY,TTMPS,CAP,CAPT,TIME,LMAX,SLTMP,DIST,TEEP,CAPY,
  2CAPPY
   DO 1 K=1,IMAX
   DO 20 J=1,JMAX
   IF(TMPS(J,K).LT.600.)GO TO 9                                         THERMAB3
   IF(TEEP(J,K).LT.TEMP(J,K))GO TO 9
   CAPT(J,K)=CAPY(J,K)                                                  THERMAB3
   CAP(J,K)=CAPPY(J,K)                                                  THERMAB3
 9 IF(J.EQ.1)GO TO 78
61 IF(J.EQ.JMAX)GO TO 70
```

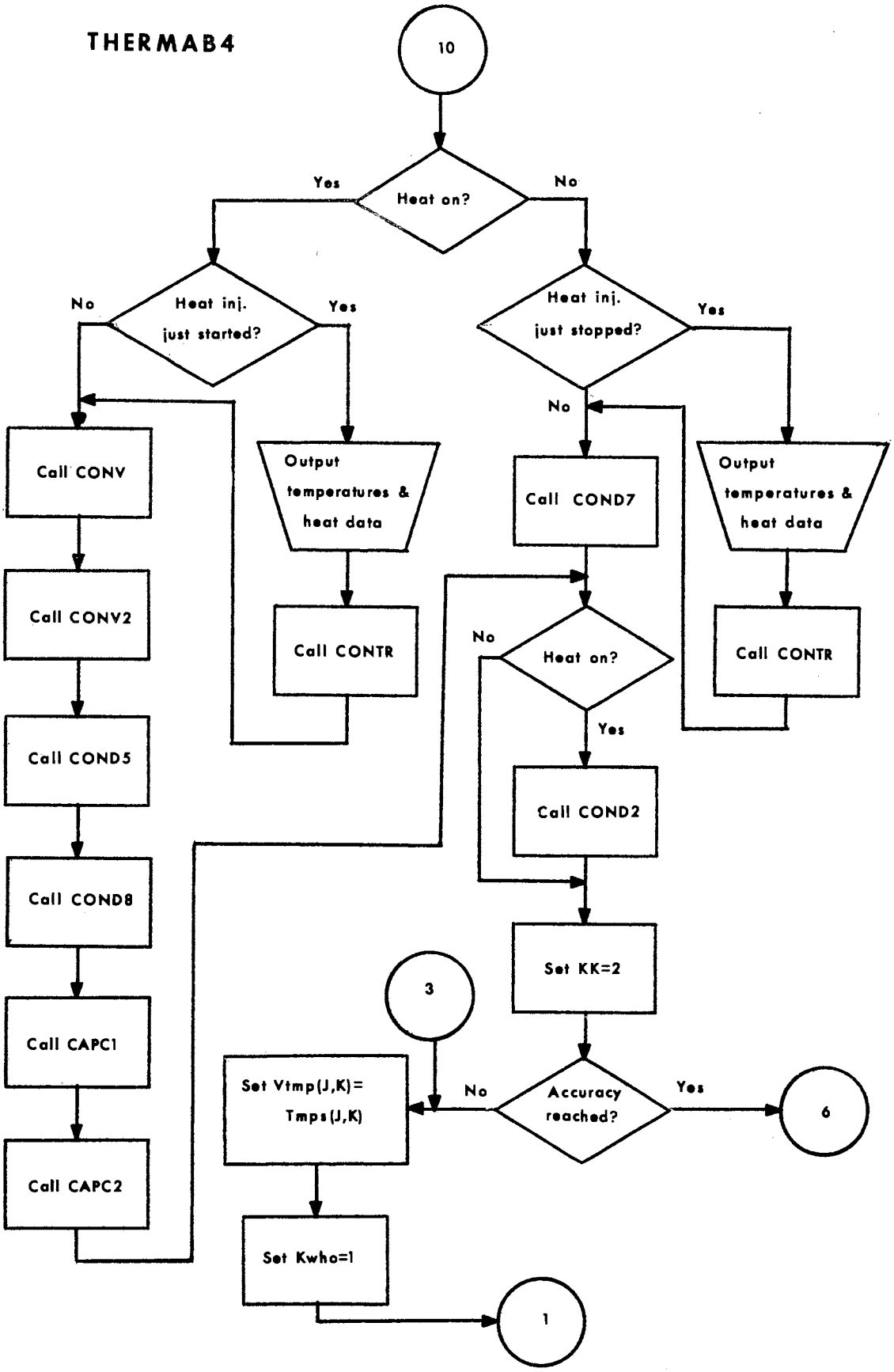
```
      IF(K.EQ.1)GO TO 79
      IF(K.EQ.IMAX)GO TO 80
      CXPUS=(COND(J,K)+COND(J,K+1))/2.                                  22240
      CXMIN=(COND(J,K)+COND(J,K-1))/2.                                  22240
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                  22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22240
      GO TO 77                                                          22240
   79 CXPUS=(COND(J,K)+COND(J,K+1))/2.
      CXMIN=CXPUS                                                       22240
      CYPUS=(COND(J,K)+COND(J+1,K))/2.
      CYMIN=(COND(J,K)+COND(J-1,K))/2.
      GO TO 77                                                          22240
   80 CXMIN=(COND(J,K)+COND(J,K-1))/2.
      CXPUS=CXMIN                                                       22240
      CYPUS=(COND(J,K)+COND(J+1,K))/2.
      CYMIN=(COND(J,K)+COND(J-1,K))/2.
      GO TO 77                                                          22240
   78 IF(K.EQ.1)GO TO 75
      IF(K.EQ.IMAX)GO TO 76
      CXPUS=(COND(J,K)+COND(J,K+1))/2.
      CXMIN=(COND(J,K)+COND(J,K-1))/2.
      CYPUS=(COND(J,K)+COND(J+1,K))/2.
      CYMIN=CYPUS
      GO TO 77                                                          22240
   70 IF(K.EQ.1)GO TO 72
      IF(K.EQ.IMAX)GO TO 73
      CXPUS=(COND(J,K)+COND(J,K+1))/2.
      CXMIN=(COND(J,K)+COND(J,K-1))/2.
      CYMIN=(COND(J,K)+COND(J-1,K))/2.
      CYPUS=CYMIN                                                       22240
      GO TO 77                                                          22240
   72 CXPUS=(COND(J,K)+COND(J,K+1))/2.
      CXMIN=CXPUS                                                       22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.
      CYPUS=CYMIN                                                       22240
      GO TO 77
   73 CXMIN=(COND(J,K)+COND(J,K-1))/2.
      CXPUS=CXMIN                                                       22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.
      CYPUS=CYMIN                                                       22240
      GO TO 77
   75 CXPUS=(COND(J,K)+COND(J,K+1))/2.
      CXMIN=CXPUS
      CYPUS=(COND(J,K)+COND(J+1,K))/2.
      CYMIN=CYPUS
      GO TO 77
   76 CXMIN=(COND(J,K)+COND(J,K-1))/2.
      CXPUS=CXMIN
      CYPUS=(COND(J,K)+COND(J+1,K))/2.
      CYMIN=CYPUS
   77 CONTINUE                                                          22240
      A1=(YPLUS(J)*CYMIN)/(CYPUS*YNEG(J))                               22240
      C1=1.
      B1=-(1.0+(YPLUS(J)*CYMIN)/(YNEG(J)*CYPUS)+(CAPT(J,K)*YMID(J)*YPLUS
     1(J))/(DELTT*CYPUS*TMPS(J,K)))
      D14=-(YMID(J)*YPLUS(J)/(DELTX**2*CYPUS))
      IF(K.LT.IMAX) GO TO 24
      D15=(TMPS(J,K-1)-TMPS(J,K))*CXPUS
      GO TO 25
   24 D15=(TMPS(J,K+1)-TMPS(J,K))*CXPUS
   25 IF(K.GT.1) GO TO 26
      D16=(TMPS(J,K)-TMPS(J,K+1))*CXMIN
      GO TO 27
   26 D16=(TMPS(J,K)-TMPS(J,K-1))*CXMIN
   27 D17=(CAP(J,K)*YMID(J)*YPLUS(J))/(DELTT*CYPUS)
      D1=D14*(015-016)-017                                              22241
      IF(J.NE.1)GO TO 3
      C1=C1+A1
      A1=0.0
      W(1,K)=C1/B1
      G(1,K)=D1/B1
      GO TO 20
    3 IF(J.NE.JMAX)GO TO 10                                             22241
      A1=A1+C1
      C1=0.0
   10 DENOM=B1-A1*W(J-1,K)
      IF(DENOM.NE.0.)GO TO 18
      WRITE(6,50)
   50 FORMAT(18H ZERO DENOMINATOR.)                                     22241
      GO TO 800
   18 W(J,K)=C1/DENOM                                                   22241
      G(J,K)=(D1-A1*G(J-1,K))/DENOM                                     22241
   20 CONTINUE
      TTMPS(JMAX,K)=G(JMAX,K)
      NMAX=JMAX-1
      DO 30 I=1,NMAX
      II=JMAX-I
   30 TTMPS(II,K)=G(II,K)-W(II,K)*TTMPS(II+1,K)
    1 CONTINUE
      DO 95 J=1,JMAX
      DO 95 K=1,IMAX
      TMPS(J,K)=TTMPS(J,K)
   95 CONTINUE
      CALL CAPAC
      CALL COND4
      CALL CAPSK                                                        THERMAB3
  800 RETURN
      END
      SUBROUTINE COND6
      DIMENSION TMPS(21,51),TEMP(21,51),W(21,51),G(21,51),Y(21),V(51),YP   22241
     1LUS(21),YNEG(21),YMID(21),TTMPS(21,51),AREA(21),H(21),HA(21),VTMP(   22241
     221,51),CAPT(21,51),CAP(21,51),COND(21,51),IS(10),SLTMP(10),DIST(8,
     351),TEEP(21,51),CAPY(21,51),CAPPY(21,51)                          THERMAB3
      COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH    22241
     1A,DELTX,DENOM,DELTY,TTMPS,CAP,CAPT,TIME,LMAX,SLTMP,DIST,TEEP,CAPY,
     2CAPPY
      DO 1 J=1,JMAX                                                     22242
      DO 20 K=1,IMAX                                                    22241
      IF(TMPS(J,K).LT.600.)GO TO 9                                      THERMAB3
      IF(TEEP(J,K).LT.TEMP(J,K))GO TO 9
      CAPT(J,K)=CAPY(J,K)                                               THERMAB3
      CAP(J,K)=CAPPY(J,K)                                               THERMAB3
    9 IF(J.EQ.1)GO TO 78                                                22242
   51 IF(J.EQ.JMAX)GO TO 70                                             22242
      IF(K.EQ.1)GO TO 79                                                22242
      IF(K.EQ.IMAX)GO TO 80                                             22242
      CXPUS=(COND(J,K)+COND(J,K+1))/2.                                  22240
      CXMIN=(COND(J,K)+COND(J,K-1))/2.                                  22240
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                  22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22240
      GO TO 77                                                          22240
   79 CXPUS=(COND(J,K)+COND(J,K+1))/2.                                  22241
      CXMIN=CXPUS                                                       22241
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                  22241
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22241
      GO TO 77
   80 CXMIN=(COND(J,K)+COND(J,K-1))/2.
      CXPUS=CXMIN
      CYPUS=(COND(J,K)+COND(J+1,K))/2.
      CYMIN=(COND(J,K)+COND(J-1,K))/2.
      GO TO 77                                                          22240
   78 IF(K.EQ.1)GO TO 75
      IF(K.EQ.IMAX)GO TO 76
      CXPUS=(COND(J,K)+COND(J,K+1))/2.
      CXMIN=(COND(J,K)+COND(J,K-1))/2.
      CYPUS=(COND(J,K)+COND(J+1,K))/2.
      CYMIN=CYPUS
      GO TO 77                                                          22240
   70 IF(K.EQ.1)GO TO 72
      IF(K.EQ.IMAX)GO TO 73
      CXPUS=(COND(J,K)+COND(J,K+1))/2.
      CXMIN=(COND(J,K)+COND(J,K-1))/2.
      CYMIN=(COND(J,K)+COND(J-1,K))/2.
      CYPUS=CYMIN
      GO TO 77                                                          22240
   72 CXPUS=(COND(J,K)+COND(J,K+1))/2.
      CXMIN=CXPUS                                                       22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.
      CYPUS=CYMIN                                                       22240
```

```
      GO TO 77                                                          22240
   73 CXMIN=(COND(J,K)+COND(J,K-1))/2.
      CXPUS=CXMIN                                                       22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.
      CYPUS=CYMIN
      GO TO 77
   75 CXPUS=(COND(J,K)+COND(J,K+1))/2.
      CXMIN=CXPUS
      CYPUS=(COND(J,K)+COND(J+1,K))/2.
      CYMIN=CYPUS
      GO TO 77
   76 CXMIN=(COND(J,K)+COND(J,K-1))/2.
      CXPUS=CXMIN
      CYPUS=(COND(J,K)+COND(J+1,K))/2.
      CYMIN=CYPUS
   77 CONTINUE                                                          22240
      B1=-(1.+CXMIN/CXPUS+(DELTX**2/DELTT)*(CAPT(J,K)/(TMPS(J,K)*CXPUS))
     1)
      D14=-(DELTX**2/(YMID(J)*CXPUS))                                   22240
      IF(J.LT.JMAX)GO TO 22                                             22241
      D15=CYPUS*((TMPS(J-1,K)-TMPS(J,K))/YPLUS(J))                      22240
      GO TO 25
   22 D15=CYPUS*((TMPS(J+1,K)-TMPS(J,K))/YPLUS(J))                      22240
   25 IF(J.GT.1) GO TO 90
      D16=(TMPS(J,K)-TMPS(J+1,K))*CYMIN/YNEG(J)
      GO TO 23
   90 D16=CYMIN*((TMPS(J,K)-TMPS(J-1,K))/YNEG(J))                       22240
   23 D17=(DELTX**2*CAP(J,K))/(DELTT*CXPUS)
      D1=D14*(D15-D16)-D17
      IF(K.NE.1)GO TO 3
      A1=0.0
      C1=1.+CXMIN/CXPUS                                                 22240
      W(J,1)=C1/B1
      G(J,1)=D1/B1
      GO TO 20
    3 IF(K.NE.IMAX)GO TO 5                                              22241
      C1=0.0
      A1=1.+CXMIN/CXPUS                                                 22240
      GO TO 10
    5 A1=CXMIN/CXPUS                                                    22240
      C1=1.0
   10 DENOM=B1-A1*W(J,K-1)                                              22241
      IF(DENOM.NE.0.)GO TO 18                                           22241
      WRITE(6,50)
   50 FORMAT(18H ZERO DENOMINATOR.)                                     22241
      GO TO 800
   18 W(J,K)=C1/DENOM
      G(J,K)=(D1-A1*G(J,K-1))/DENOM                                     22241
   20 CONTINUE
      TTMPS(J,IMAX)=G(J,IMAX)
      DO 30 I=2,IMAX
      II=IMAX+1-I
   30 TTMPS(J,II)=G(J,II)-W(J,II)*TTMPS(J,II+1)
    1 CONTINUE
      DO 95 J=1,JMAX
      DO 95 K=1,IMAX
      TMPS(J,K)=TTMPS(J,K)
   95 CONTINUE
      CALL CAPAC
      CALL COND4
      CALL CAPSK                                                        THERMAB3
  800 RETURN
      END
      SUBROUTINE CONTR
      DIMENSION TMPS(21,51),TEMP(21,51),W(21,51),G(21,51),Y(21),V(51),YP  22241
     1LUS(21),YNEG(21),YMID(21),TTMPS(21,51),AREA(21),H(21),HA(21),VTMP(  22241
     221,51),CAPT(21,51),CAP(21,51),COND(21,51),IS(10),SLTMP(10),DIST(8,
     351),TEEP(21,51),CAPY(21,51),CAPPY(21,51)                          THERMAB3
      COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH  22241
     1A,DELTX,DENOM,DELTY,TTMPS,CAP,CAPT,TIME,LMAX,SLTMP,DIST,TEEP,CAPY,
     2CAPPY
      DO 263 I=1,LMAX
      DO 260 K=1,IMAX
      DO 261 J=1,JMAX
      IF(TEMP(J,K).LT.SLTMP(I))GO TO 262
  261 CONTINUE
  262 IF(J.EQ.1)GO TO 260
      DIST(I,K)=Y(J-1)+((TEMP(J-1,K)-SLTMP(I))/(TEMP(J-1,K)-TEMP(J,K)))*
     1(Y(J)-Y(J-1))
  260 CONTINUE
  263 CONTINUE
      WRITE(6,264)(SLTMP(I),I=1,LMAX)
  264 FORMAT(1H1,6X,'DISTANCE',7X,'ISOTHERM  ISOTHERM  ISOTHERM  ISOTHERTHERMAB6
     1M  ISOTHERM  ISOTHERM  ISOTHERM  ISOTHERM'/8X,'DOWN',9X,'DEGREES FTHERMAB6
     2 DEGREES F DEGREES F DEGREES F DEGREES F DEGREES F DEGREES F DEGRETHERMAB6
     3EES F'/7X,'FRACTURE',4X,8F10.1/9X,'FEET'/20X,'FEET  INTO FEET  INTO THERMAB6
     4FEET INTO FEET  INTO FEET  INTO FEET  INTO FEET  INTO FEET  INTO'/22X,'THERMAB6
     5WALL',6X,'WALL',6X,'WALL',6X,'WALL',6X,'WALL',6X,'WALL',6X,'WALL',6X,'WALL',THERMAB6
     66X,'WALL')                                                        THERMAB6
      DO 266 K=1,40                                                     THERMAB6
      FK2=K-1                                                           THERMAB6
      XFT=DELTX*FK2                                                     THERMAB6
  266 WRITE(6,267)XFT,(DIST(I,K),I=1,LMAX)                              THERMAB6
  267 FORMAT(7X,F6.0,4X,8F10.2)
      WRITE(6,264)(SLTMP(I),I=1,LMAX)
      DO 265 K=41,IMAX                                                  THERMAB6
      FK2=K-1                                                           THERMAB6
      XFT=DELTX*FK2                                                     THERMAB6
  265 WRITE(6,267)XFT,(DIST(I,K),I=1,LMAX)                              THERMAB6
      RETURN
      END
      SUBROUTINE CAPSK                                                  THERMAB3
      DIMENSION TMPS(21,51),TEMP(21,51),W(21,51),G(21,51),Y(21),V(51),YP  22241
     1LUS(21),YNEG(21),YMID(21),TTMPS(21,51),AREA(21),H(21),HA(21),VTMP(  22241
     221,51),CAPT(21,51),CAP(21,51),COND(21,51),IS(10),SLTMP(10),DIST(8,
     351),TEEP(21,51),CAPY(21,51),CAPPY(21,51)                          THERMAB3
      COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH  22241
     1A,DELTX,DENOM,DELTY,TTMPS,CAP,CAPT,TIME,LMAX,SLTMP,DIST,TEEP,CAPY,
     2CAPPY
      DO 232 J=1,JMAX                                                   THERMAB3
      DO 232 K=1,IMAX                                                   THERMAB3
      IF(TMPS(J,K).GT.450.)GO TO 233                                    THERMAB3
      CAPY(J,K)=(23.5+.307*(TMPS(J,K)-100.))*134.0                      THERMAB3
      GO TO 232                                                         THERMAB3
  233 IF(TMPS(J,K).GT.600.)GO TO 234                                    THERMAB3
      CAPY(J,K)=(130.0+.344*(TMPS(J,K)-450.))*134.0                     THERMAB3
      GO TO 232                                                         THERMAB3
  234 IF(TMPS(J,K).GT.1100.)GO TO 236                                   THERMAB3
      CAPY(J,K)=134.0*(180.0+.230*(TMPS(J,K)-600.))                     THERMAB3
      GO TO 232                                                         THERMAB3
  236 CAPY(J,K)=134.0*(295.0+.258*(TMPS(J,K)-1100.))                    THERMAB3
  232 CONTINUE                                                          THERMAB3
      RETURN                                                            THERMAB3
      END
```

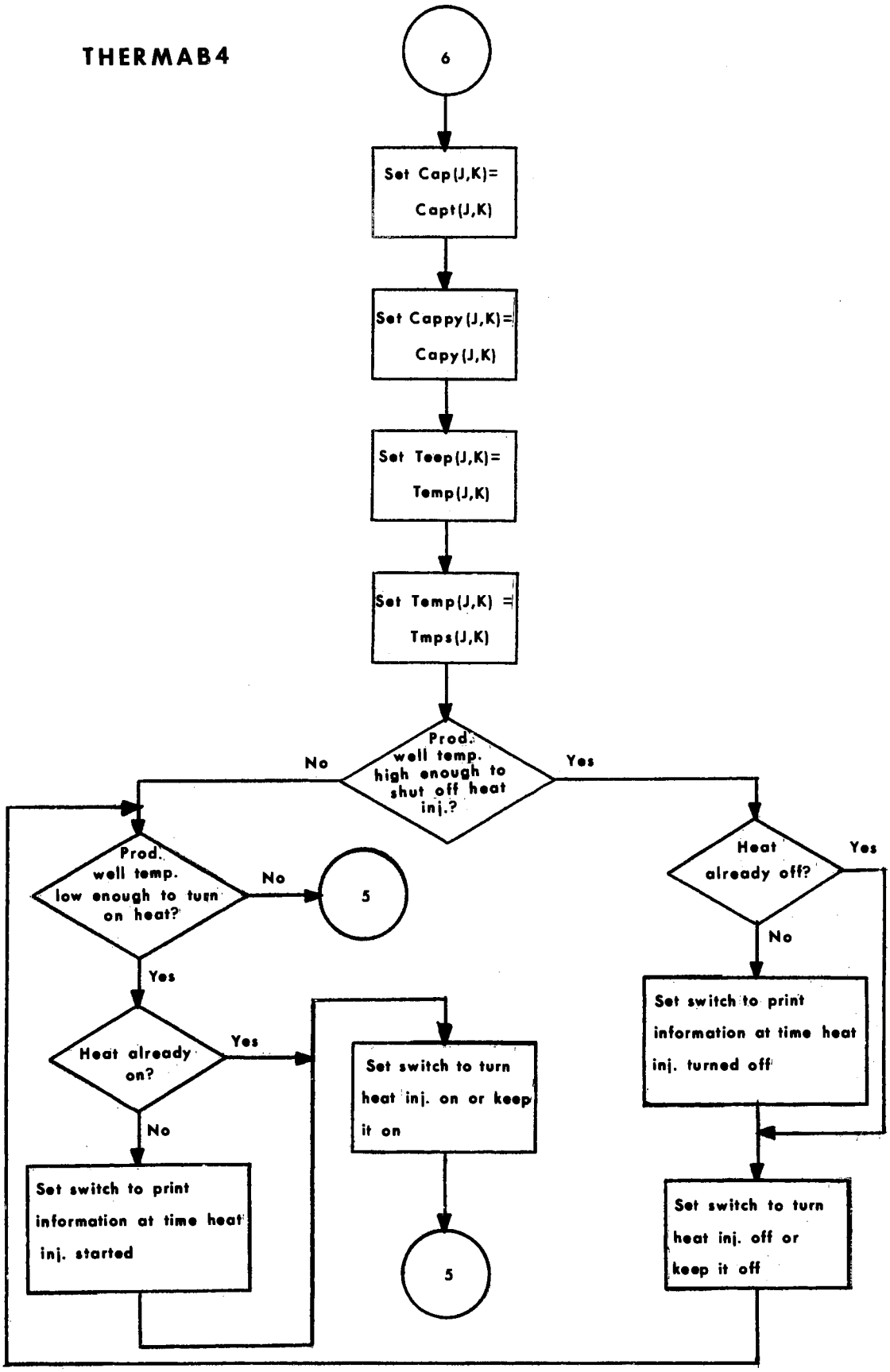
THERMAB4

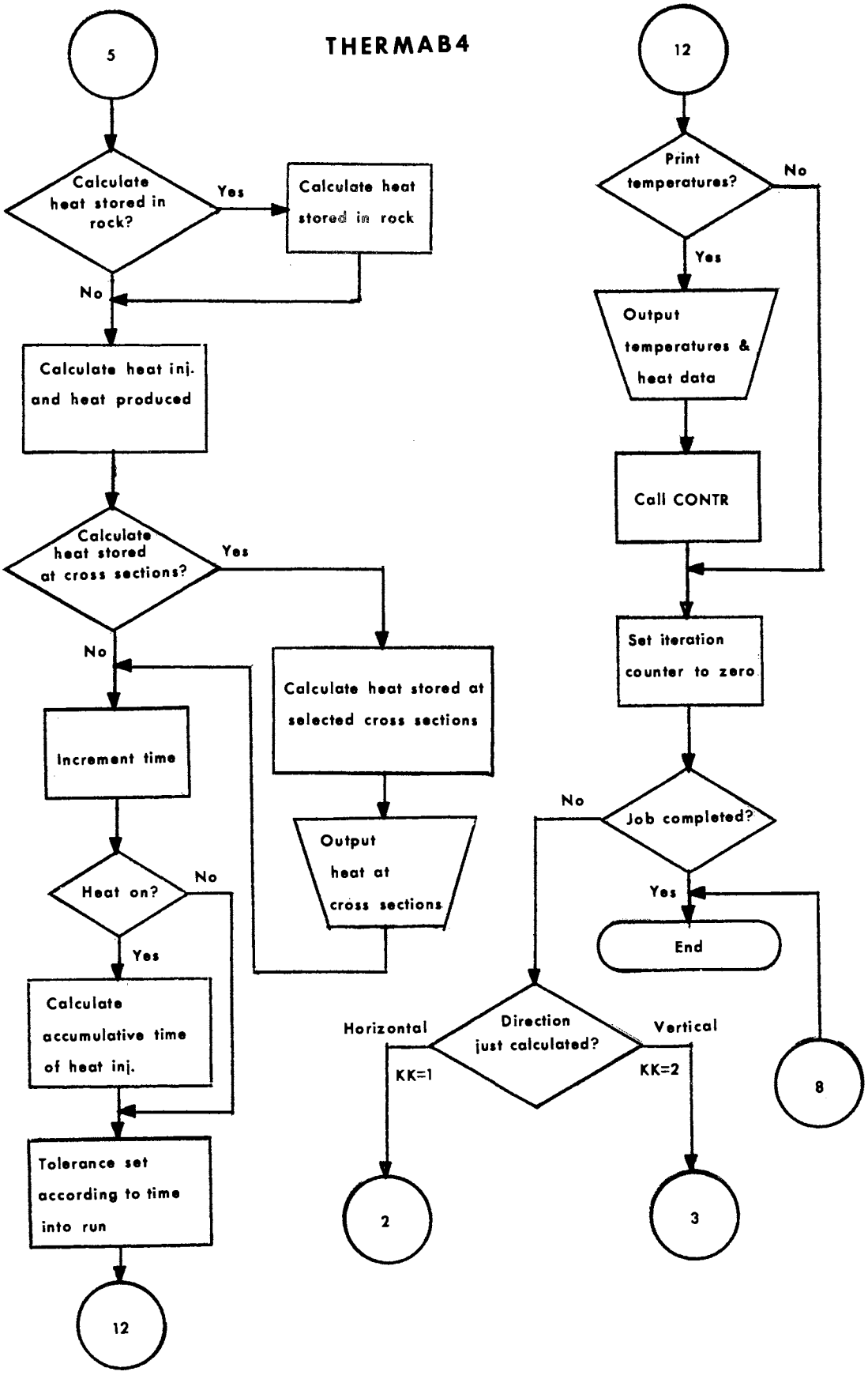

Start
Input system data
Compute velocity
Output general test data
Initialize parameters
Call CALCY
Call CAPAC
Call COND4
7
7
Call CAPSK
Initialize permanent cap arrays
1
Increase time step length during run?
No
Yes
End of time step?
No
Yes
Time step length set according to time into run
Vertical or horizontal sweep?
Vertical
KWHO=2
10
Horizontal
KWHO=1
11


THERMAB4
11
Heat on?
Yes
No
Heat inj. just started?
No
Yes
Heat inj. just stopped?
Yes
No
Call CONV
Output temperatures & heat data
Call COND6
Output temperatures & heat data
Call CONV2
Call CONTR
12
Call CONTR
Heat on?
No
Call COND5
Yes
Call COND1
6
Accuracy reached?
Call COND8
2
Increment iteration counter
Call CAPC1
Set Vtmp(J,K)= Tmps(J,K)
Excessive iteration?
Yes
8
Call CAPC2
Set Kwho=2
No
Set KK=1
12
1


THERMAB4
10
Heat on?
Yes
No
Heat inj. just started?
No
Yes
Heat inj. just stopped?
Yes
No
Call CONV
Output temperatures & heat data
Call COND7
Output temperatures & heat data
Call CONV2
Call CONTR
Heat on?
Call CONTR
No
Yes
Call COND5
Call COND2
Call COND8
Set KK=2
Call CAPC1
3
Set Vtmp(J,K)= Tmps(J,K)
No
Accuracy reached?
Yes
6
Call CAPC2
Set Kwho=1
1

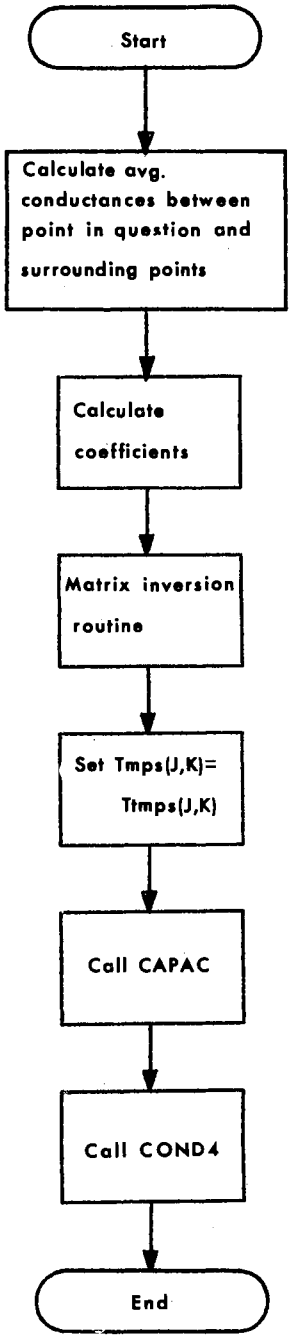
THERMAB4


6
Set Cap(J,K)= Capt(J,K)
Set Cappy(J,K)= Capy(J,K)
Set Teep(J,K)= Temp(J,K)
Set Temp(J,K) = Tmps(J,K)
Prod. well temp. high enough to shut off heat inj.?
No
Yes
Prod. well temp. low enough to turn on heat?
No
5
Yes
Heat already on?
Yes
No
Set switch to print information at time heat inj. started
Set switch to turn heat inj. on or keep it on
5
Heat already off?
Yes
No
Set switch to print information at time heat inj. turned off
Set switch to turn heat inj. off or keep it off

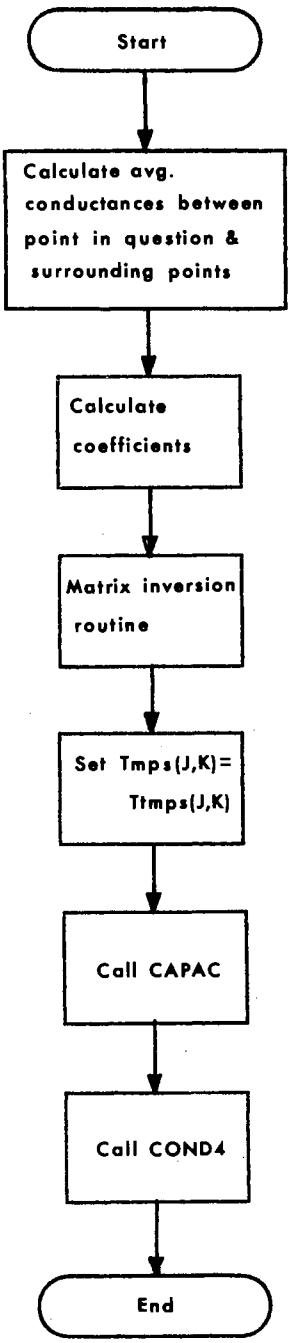

THERMAB4
5
Calculate heat stored in rock?
Yes
Calculate heat stored in rock
No
Calculate heat inj. and heat produced
Calculate heat stored at cross sections?
Yes
Calculate heat stored at selected cross sections
Output heat at cross sections
No
Increment time
Heat on?
No
Yes
Calculate accumulative time of heat inj.
Tolerance set according to time into run
12
12
Print temperatures?
No
Yes
Output temperatures & heat data
Call CONTR
Set iteration counter to zero
Job completed?
No
Yes
End
Direction just calculated?
Horizontal
KK=1
Vertical
KK=2
2
3
8

THERMAB4

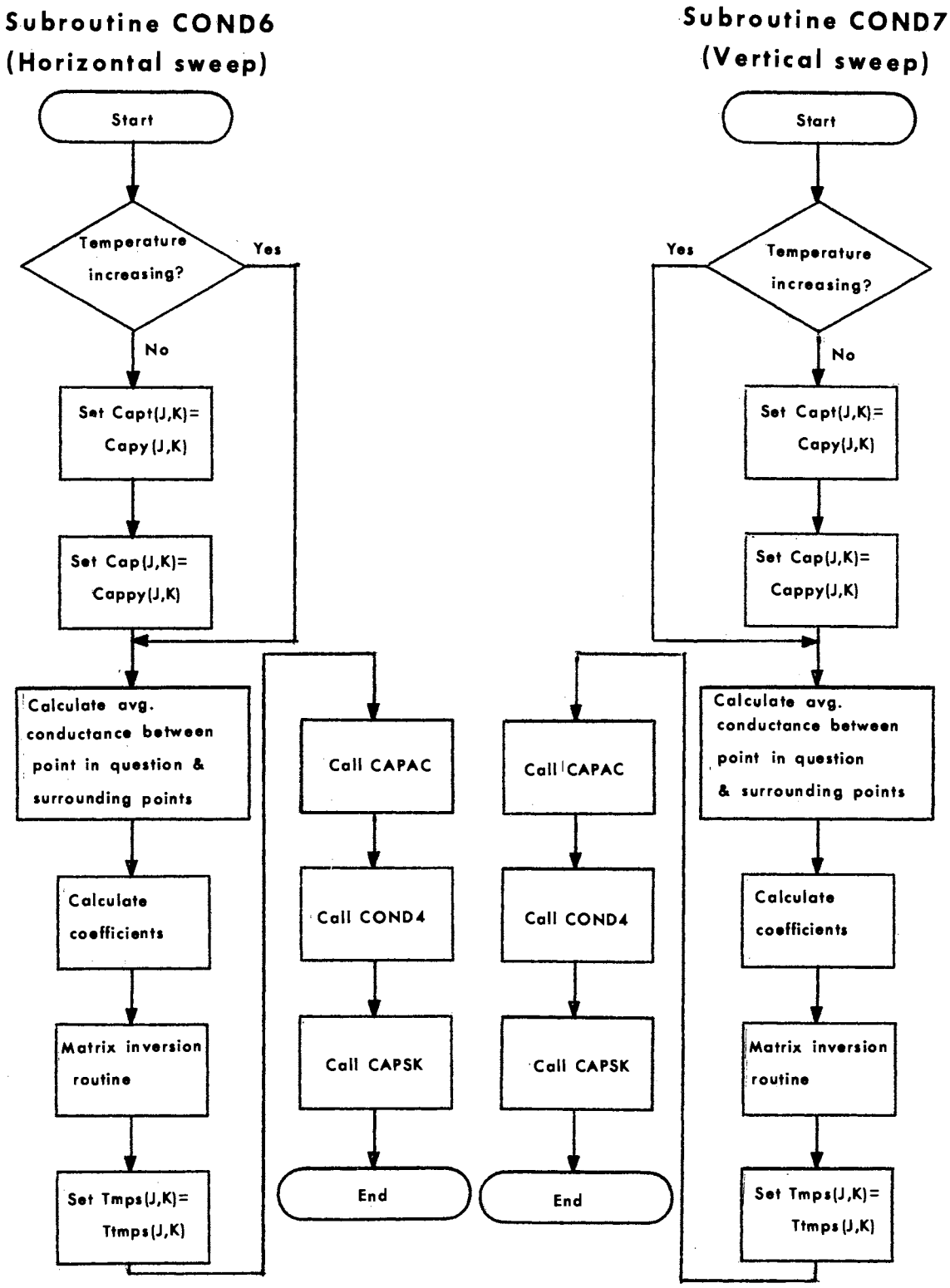
Subroutine COND1 (Horizontal sweep)



Subroutine COND2 (Vertical sweep)

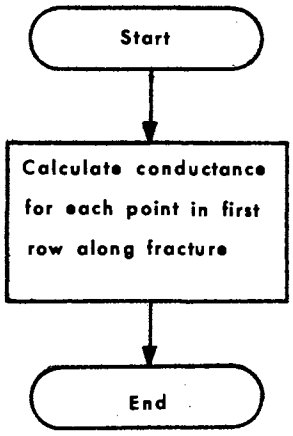


THERMAB4

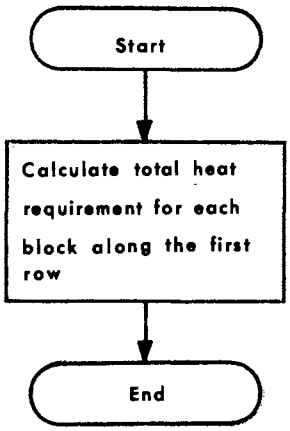

Subroutine COND6
(Horizontal sweep)
Start
Temperature increasing?
Yes
No
Set Capt(J,K)= Capy(J,K)
Set Cap(J,K)= Cappy(J,K)
Calculate avg. conductance between point in question & surrounding points
Calculate coefficients
Matrix inversion routine
Set Tmps(J,K)= Ttmps(J,K)
Call CAPAC
Call COND4
Call CAPSK
End
Subroutine COND7
(Vertical sweep)
Start
Yes
Temperature increasing?
No
Set Capt(J,K)= Capy(J,K)
Set Cap(J,K)= Cappy(J,K)
Call CAPAC
Call COND4
Call CAPSK
End
Calculate avg. conductance between point in question & surrounding points
Calculate coefficients
Matrix inversion routine
Set Tmps(J,K)= Ttmps(J,K)

THERMAB4

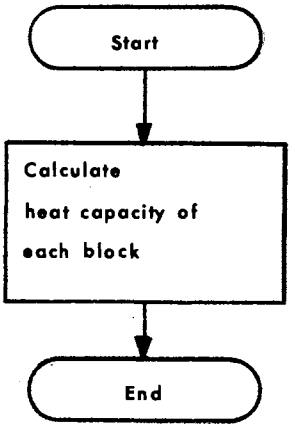
Subroutine COND5



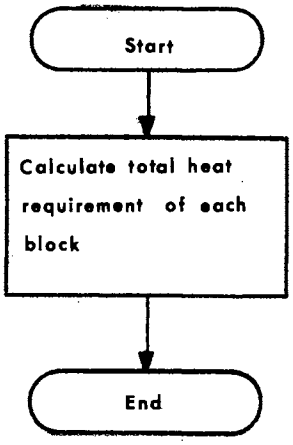
Subroutine CAPC1



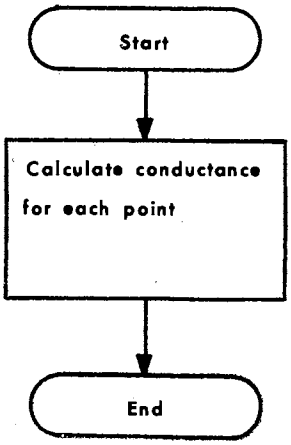
Subroutine CAPSK



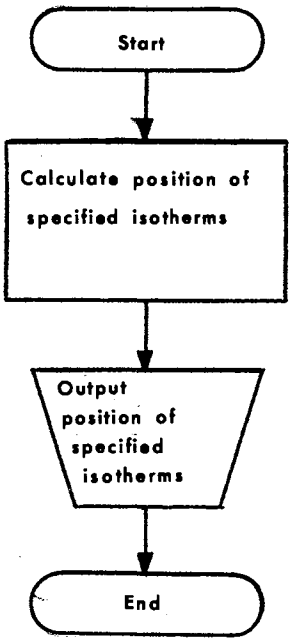
Subroutine CAPAC



Subroutine COND4

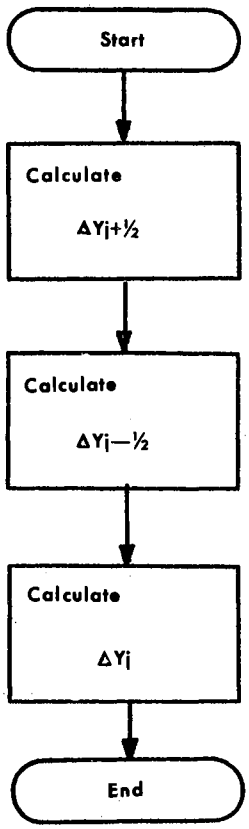


Subroutine CONTR

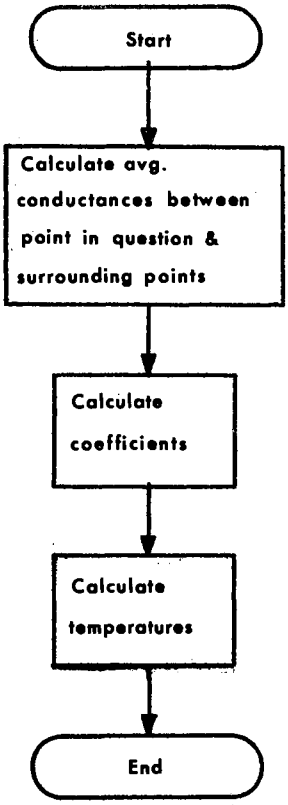


THERMAB4

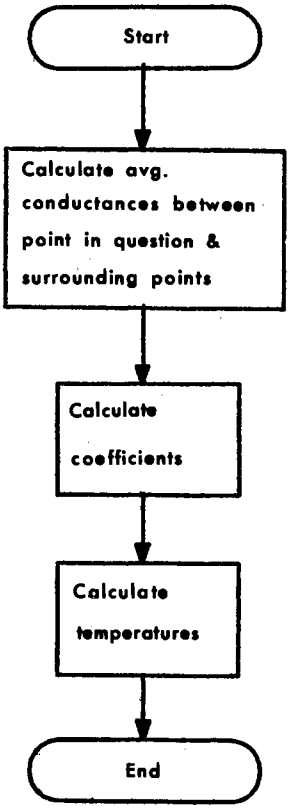
Subroutine CALCY (Variable grid spacing)



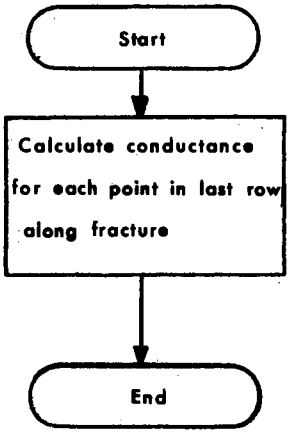
Subroutine CONV



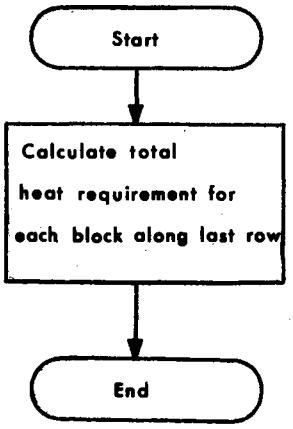
Subroutine CONV2



Subroutine COND8



Subroutine CAPC2



```
C                    THERMAB4-A.L. BARNES
      DIMENSION TMPS(25,51),TEMP(25,51),W(25,51),G(25,51),Y(25),V(51),YP
     1LUS(25),YNEG(25),YMID(25),TTMPS(25,51),AREA(25),H(25),HA(25),VTMP(
     225,51),CAPT(25,51),CAP(25,51),COND(25,51),IS(10),SLTMP(10),DIST(8,
     351),TEEP(25,51),CAPY(25,51),CAPPY(25,51)
      COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH
     1A,DELTX,DENOM,DELTY,TTMPS,CAP,CAPT,TIME,LMAX,SLTMP,DIST,TEEP,CAPY,
     2CAPPY
C     TEMP=PERMANENT TEMPERATURE ARRAY                                  22241
C     VTMP=TEMPORARY TEMPERATURE ARRAY
C     TMPS=TEMPORARY TEMPERATURE ARRAY                                  22241
C     KHT IS THE SWITCH TO TURN HEAT ON OR OFF                         THERMAB6
C     KWTSP IS THE SWITCH TO PRINT INFORMATION AT THE TIME HEAT IS SHUT THERMAB6
C          OFF                                                         THERMAB6
C     KWTST IS THE SWITCH TO PRINT INFORMATION AT THE TIME HEAT IS      THERMAB6
C          TURNED ON                                                   THERMAB6
C     TIMHT=CUMULATIVE TIME OF HEAT INJECTION                          THERMAB6
  100 READ(5,10)IMAX,JMAX,DELTT,TOL,TIMX,ALPHA,DELTX,DELTY,AA,BB,TT,TTT,  22241
     1CC,KSKIP,NRUN,KSLCE,ISSN,(IS(M),M=1,5),LMAX,(SLTMP(I),I=1,8),KSTRT
     2,KR,TTTT
C     IMAX=NUMBER OF COLUMNS (K)
C     JMAX=NUMBER OF ROWS (J)
C     DELTT=TIME STEP LENGTH,HOURS
C     TOL=TOLERANCE
C     TIMX=MAXIMUM TIME, HOURS
C     ALPHA=SPECIFIC HEAT X VELOCITY X DENSITY(ALL OF GAS)
C     DELTX=DISTANCE BETWEEN POINTS ALONG THE FRACTURE
C     DELTY=DISTANCE BETWEEN POINTS INTO THE WALL(FIRST 3 POINTS ONLY)
C     AA=CONSTANT IN EQUATION WHICH GENERATES VELOCITY
C     BB=CUNSTANT IN EQUATION WHICH GENERATES VELOCITY
C     TT=INJECTION TEMPERATURE, DEGREES FAHRENHEIT
C     TTT=RESERVOIR TEMPERATURE, DEGREES FAHRENHEIT
C     CC=CONSTANT IN EQUATION WHICH GENERATES VELOCITY
C     KSKIP=OPTION TO INCREASE TIME STEP LENGTH DURING RUN
C     NRUN=RUN NUMBER
C     KSLCE=OPTION TO CALCULATE TUTAL HEAT CONTENT AND PRINT TEMPERATURE
C          AND TOTAL HEAT CONTENT
C     ISSN=MAXIMUM NUMBER OF DESIRED SLICES FOR CROSS SECTION STUDY
C     IS(M)=VALUES OF DESIRED SLICES FOR CROSS SECTION STUDY
C     LMAX=NUMBER OF ISOTHERMS DESIRED
C     SLTMP(I)=TEMPERATURE OF DESIRED ISOTHERM
C     KSTRT=SWITCH THAT ALLOWS STARTING RUN WITH CONSTANT OR VARIABLE
C          FIELD TEMPERATURES
C     KR=OPTION TO SET INJECTION TEMPERATURE TO TTTT OR TEMP(1,IMAX)
C     TTTT=INJECTION TEMPERATURE AT JMAX,IMAX
      READ(5,20)(Y(I),I=1,JMAX)
   20 FORMAT(16F5.0/9F5.0)
C     Y=DISTANCE INTO THE WALL, FEET
      KHT=1                                                            THERMAB6
      KWTSP=1                                                          THERMAB6
      TIMHT=0.0                                                        THERMAB6
      KWTST=1                                                          THERMAB6
      PPP=0.0
      PP=0.
      TIME=0.0                                                          22241
      SMAX=IMAX
      FTSYT=(SMAX-1.)*DELTX
      ATMP=0.0
      OD=1.0
      KWHO=1
      CM=25000.
      IIT=0
      KIT=0
      HTOUT=0.
      HTIN=0.
      DO 21 K=1,IMAX                                                    22241
      FK=K                                                              22241
      X=DELTX*FK                                                        22241
      V(K)=(AA+BB*X+EXP(CC*X))*2.                                      THERMAB3
   21 CONTINUE                                                          22241
      DO 76 I=1,LMAX
      DO 76 K=1,IMAX
   76 DIST(I,K)=0.0
      WRITE(6,9)NRUN,TT,TTT,TCL,KSKIP,ALPHA,DELTX,(Y(I),I=1,25),AA,BB,CC
     1,TIMX,IMAX,JMAX,FTSYT
    9 FORMAT(1H1,5X,'INPUT DATA FOR THERMAB6 RUN NUMBER',2X,I5//5X,F10.1
     1,2X,'=INJECTION TEMPERATURE, DEGREES FAHRENHEIT'/5X,F10.1,2X,'=RES
     2ERVOIR TEMPERATURE, DEGREES FAHRENHEIT'/5X,F10.4,2X,'=TOLERANCE(PE
     3R CENT'/10X,I5,2X,'=OPTION TO INCREASE TIME STEP LENGTH DURING RUN
     4'/20X,'(1) INDICATES TIME STEP LENGTH REMAINED CONSTANT DURING RUN
     5'/20X,'(2) INDICATES TIME STEP LENGTH WAS INCREASED DURING RUN'/5X
     6,F10.4,2X,'=MASS X VELOCITY X DENSITY(ALL OF GAS)
     7/5X,F10.1,2X,'=DISTANCE BETWEEN POINTS ALONG THE FRACTURE, FEET'//
     85X,'THE DISTANCE INTO THE WALL, FEET, IS GIVEN BELOW',/3X,25F5.1//
     95X,'THE CONSTANTS IN THE EQUATION WHICH GENERATES VELOCITY ARE'/10
     1X,F10.3,2X,'=AA'/10X,F10.3,2X,'=BB'/10X,F10.3,2X,'=CC'//5X,F10.1,2
     2X,'=MAXIMUM TIME, HOURS'/10X,I5,2X,'=IMAX'/10X,I5,2X,'=JMAX'/5X,F1
     30.1,2X,'LENGTH OF SYSTEM, FEET')
      TENTH=0.0
   10 FORMAT(2I5,F5.1,F5.2,F10.1,F10.1,6F5.1,F10.4/10I5/8F10.1/2I5,F10.1
     1)
      IF(KSTRT.EQ.1)GO TO 81
      READ(5,203)TIME,HTIN,HTOUT
C     TIME=TIME, HOURS, ALREADY COMPLETED WHEN RUN RESTARTED
C     HTIN=BTU OF HEAT INJECTED WHEN RUN RESTARTED
C     HTOUT=BTU OF HEAT PRODUCED WHEN RUN RESTARTED
  203 FORMAT(F10.0,2E18.8)
      ATMP=TIME
      DO 74 K=1,IMAX
   74 READ(5,73)(TMPS(J,K),J=1,JMAX)
   73 FORMAT(16F5.0/9F5.0)
      GO TO 204
   81 DO 75 K=1,IMAX
      DO 75 J=1,JMAX
   75 TMPS(J,K)=TTT
  204 DO 205 K=1,IMAX
      DO 205 J=1,JMAX
      TEMP(J,K)=TMPS(J,K)
      TEEP(J,K)=TMPS(J,K)                                              THERMAB3
  205 VTMP(J,K)=TMPS(J,K)
      TMPS(1,1)=TT
      TEMP(1,1)=TT                                                      22241
      VTMP(1,1)=TT                                                      22241
      TEMP(JMAX,IMAX)=TTTT                                              22241
      TMPS(JMAX,IMAX)=TTTT
      VTMP(JMAX,IMAX)=TTTT                                              22241
      IF(KR.EQ.1)GO TO 181                                              22241
      S=0.0                                                             22241
      TEMP(JMAX,IMAX)=TEMP(1,IMAX)                                      22241
      TMPS(JMAX,IMAX)=TMPS(1,IMAX)                                      22241
      VTMP(JMAX,IMAX)=VTMP(1,IMAX)                                      22241
  181 CALL CALCY                                                        22241
C     CALCY IS A SUBROUTINE THAT CALCULATES VARIABLE Y SPACING VALUES
      CALL CAPAC
C     CAPAC IS A SUBROUTINE THAT CALCULATES TOTAL HEAT REQUIREMENT
C          VALUES.
      CALL CAPSK
C     CAPSK IS A SUBROUTINE THAT CALCULATES HEAT CAPACITY VALUES
      CALL COND4
C     COND4 IS A SUBROUTINE THAT CALCULATES CONDUCTANCES
      DO 232 J=1,JMAX
      DO 232 K=1,IMAX
      CAPPY(J,K)=CAPY(J,K)
  232 CAP(J,K)=CAPT(J,K)
      FIMAX=IMAX                                                        22241
      HTORG=Y(JMAX)*DELTX*FIMAX*3149.+(CAP(1,1)-3149.)*DELTX*.50
C     HTORG=INITIAL HEAT CONTENT OF ROCK AT 100 DEGREES FAHRENHEIT
   82 GO TO(995,994),KSKIP
  994  IF(ATMP.NE.TIME) GO TO 995
      ATMP=ATMP+1.
      IF(TIME.GT.5.0)GO TO 318
      DELTT=1.
      GO TO 995
  318 IF(DELTT.GE.15.0) GO TO 993
      UU=DD-5.0
      TME=5.0*((1.2)**UU)
      DELTT=TME-TIME
      IF(DELTT.LT.15.)GO TO 995
      DELTT=15.
```

```
  993 IF(TIME.GT.1000.)GO TO 996
      DELTT=15.0
      GO TO 995
  996 IF(TIME.GT.3000.)GO TO 997
      DELTT=30.0
      GO TO 995
  997 IF(TIME.GT.10000.)GO TO 998
      DELTT=50.0
      GO TO 995
  998 DELTT=100.
C     KWHO IS THE SWITCH TO DETERMINE DIRECTION OF ADIP SWEEP(HORIZONTAL
C          OR VERTICAL)
  995 GO TO(991,53),KWHO
  991 GO TO(142,140),KHT                                              THERMAB6
  140 GO TO(158,141),KWTSP                                            THERMAB6
  141 KWTSP=1
C     TEMPERATURES AND HEAT DATA ARE PRINTED IF HEAT INJECTION HAS JUST
C          STOPPED
      WRITE(6,33)
      DO 46 K=1,42                                                    THERMAB6
   46 WRITE(6,37)K,(TEMP(J,K),J=1,17)                                 THERMAB6
      WRITE(6,250)                                                    THERMAB6
  250 FORMAT(1H1,2X,'J=    18',5X,'19',5X,'20',5X,'21',5X,'22',5X,'23',5
     1X,'24',5X,'25')
      DO 251 K=1,42                                                   THERMAB6
  251 WRITE(6,252)K,(TEMP(J,K),J=18,JMAX)                             THERMAB6
  252 FORMAT(I4,8F7.1)
      WRITE(6,33)                                                     THERMAB6
      DO 44 K=43,IMAX                                                 THERMAB6
   44 WRITE(6,37)K,(TEMP(J,K),J=1,17)                                 THERMAB6
      WRITE(6,250)                                                    THERMAB6
      DO 253 K=43,IMAX                                                THERMAB6
  253 WRITE(6,252)K,(TEMP(J,K),J=18,JMAX)                             THERMAB6
      WRITE(6,30)TIME,ERR,HEAT,HTOUT,HTIN,HTTOT,HTORG,DELTT,ER1,ER2,TOL THERMAB6
      WRITE(6,150)TIMHT                                               THERMAB6
  150 FORMAT(1H-,2X,'HEAT INJECTION STOPS, PRODUCING WELL TEMPERATURE  THERMAB6
     1EXCEEDS 800 DEGREES F.'/2X,'CUMULATIVE TIME OF HEAT INJECTION EQUATHERMAB6
     2LS',2X,F10.1)                                                   THERMAB6
      CALL CONTR
C     CONTR IS A SUBROUTINE THAT CALCULATES THE POSITION OF SELECTED
C          ISOTHERMS
  158 CALL COND6
C     COND6 IS A SUBROUTINE FOR THE HORIZONTAL ADIP SOLUTION WITH
C          ADIABATIC BOUNDARY CONDITIONS
      GO TO 45
  142 GO TO(143,159),KWTST                                            THERMAB6
  159 KWTST=1                                                         THERMAB6
C     TEMPERATURES AND HEAT DATA ARE PRINTED IF HEAT INJECTION HAS JUST
C          STARTED
      WRITE(6,33)
      DO 32 K=1,42
   32 WRITE(6,37)K,(TEMP(J,K),J=1,17)
      WRITE(6,250)
      DO 275 K=1,42
  275 WRITE(6,252)K,(TEMP(J,K),J=18,JMAX)
      WRITE(6,33)
      DO 36 K=43,IMAX
   36 WRITE(6,37)K,(TEMP(J,K),J=1,17)
      WRITE(6,250)
      DO 276 K=43,IMAX
  276 WRITE(6,252)K,(TEMP(J,K),J=18,JMAX)
      WRITE(6,30)TIME,ERR,HEAT,HTOUT,HTIN,HTTOT,HTORG,DELTT,ER1,ER2,TOL THERMAB6
      WRITE(6,255)
      CALL CONTR
      GO TO 143
   53 GO TO(146,145),KHT                                              THERMAB6
  145 GO TO(170,261),KWTSP
  261 KWTSP=1
      WRITE(6,33)
      DO 31 K=1,42
   31 WRITE(6,37)K,(TEMP(J,K),J=1,17)
      WRITE(6,250)
      DO 271 K=1,42
  271 WRITE(6,252)K,(TEMP(J,K),J=18,JMAX)
      WRITE(6,33)
      DO 34 K=43,IMAX
   34 WRITE(6,37)K,(TEMP(J,K),J=1,17)
      WRITE(6,250)
      DO 272 K=43,IMAX
  272 WRITE(6,252)K,(TEMP(J,K),J=18,JMAX)
      WRITE(6,30)TIME,ERR,HEAT,HTOUT,HTIN,HTTOT,HTORG,DELTT,ER1,ER2,TOL THERMAB6
      WRITE(6,150)TIMHT                                               THERMAB6
      CALL CONTR
  170 CALL COND7
C     COND7 IS A SUBROUTINE FOR THE VERTICAL ADIP SOLUTION WITH
C          ADIABATIC BOUNDARY CONDITIONS
      GO TO 171
  146 GO TO(43,157),KWTST
  157 KWTST=1                                                         THERMAB6
      WRITE(6,33)
      DO 38 K=1,42
   38 WRITE(6,37)K,(TEMP(J,K),J=1,17)
      WRITE(6,250)
      DO 278 K=1,42
  278 WRITE(6,252)K,(TEMP(J,K),J=18,JMAX)
      WRITE(6,33)
      DO 39 K=43,IMAX
   39 WRITE(6,37)K,(TEMP(J,K),J=1,17)
      WRITE(6,250)
      DO 279 K=43,IMAX
  279 WRITE(6,252)K,(TEMP(J,K),J=18,JMAX)
      WRITE(6,30)TIME,ERR,HEAT,HTOUT,HTIN,HTTOT,HTORG,DELTT,ER1,ER2,TOL THERMAB6
      WRITE(6,255)                                                    THERMAB6
  255 FORMAT(1H-,2X,'HEAT INJECTION STARTS. PRODUCING WELL TEMPERATURE ITHERMAB6
     1S LESS THAN 700 DEGREES F.')                                    THERMAB6
      CALL CONTR
      GO TO 43                                                        THERMAB6
  143 CALL CONV
C     CONV IS A SUBROUTINE THAT SOLVES EXPLICIT CONVECTION EQUATIONS
C          ALONG THE FRACTURE
      CALL CONV2
C     CONV2 IS A SUBROUTINE THAT SOLVES EXPLICIT CONVECTION EQUATIONS
C          ALONG THE J=JMAX ROW
      CALL COND5
C     COND5 IS A SUBROUTINE THAT CALCULATES NEW CONDUCTANCE VALUES
C          ALONG THE J=1 ROW
      CALL COND8
C     COND8 IS A SUBROUTINE THAT CALCULATES NEW CONDUCTANCE VALUES
C          ALONG THE J=JMAX ROW
      CALL CAPC1
C     CAPC1 IS A SUBROUTINE THAT CALCULATES NEW TOTAL HEAT REQUIREMENT
C          VALUES ALONG THE J=1 ROW
      CALL CAPC2
C     CAPC2 IS A SUBROUTINE THAT CALCULATES NEW TOTAL HEAT REQUIREMENT
C          VALUES ALONG THE J=JMAX ROW
   45 GO TO(144,47),KHT
  144 CALL COND1                                                      22241
C     COND1 IS A SUBROUTINE THAT SOLVES HORIZONTAL ADIP WITH THE CONV
C          SOLUTION AS A BOUNDARY CONDITION
   47 IF(DENOM.EQ.0.0)GO TO 69                                        22241
      KIT=KIT+1
      IIT=IIT+1
      IF(IIT.EQ.5)GO TO 296
      IF(KIT.GT.10)GO TO 222
      KK=1                                                            22241
      DO 84 N=1,JMAX
      DO 84 M=1,IMAX                                                  22241
      IF(TMPS(N,M).LT.100.10) GO TO 84
      IF(ABS((VTMP(N,M)-TMPS(N,M))/TMPS(N,M)).GT.TOL)GO TO 86         22241
   84 CONTINUE                                                        22241
      GO TO 67                                                        22241
   86 DO 88 J=1,JMAX                                                  22241
      DO 88 K=1,IMAX                                                  22241
   88 VTMP(J,K)=TMPS(J,K)                                             22241
      KWHO=2
      GO TO 82
  296 DO 297 K=1,IMAX
      DO 297 J=1,JMAX
      TMPS(J,K)=TEMP(J,K)
```

```
  297 VTMP(J,K)=TEMP(J,K)
      IIT=0
      CM=CM+5000.
      TOL=TOL+.01
      GO TO 82
   43 CALL CONV
      CALL CONV2
  147 CALL COND5
      CALL COND8
      CALL CAPC1
      CALL CAPC2
  171 GO TO(149,148),KHT
  149 CALL COND2                                                        22241
C     COND2 IS A SUBROUTINE THAT SOLVES VERTICAL ADIP WITH THE CONV
C           SOLUTION AS A BOUNDARY CONDITION
  148 IF(DEMOM.EQ.0.0)GO TO 69                                          22241
  927 KK=2                                                              22241
      DO 64 N=1,JMAX
      DO 64 M=1,IMAX                                                    22241
      IF(TMPS(N,M).LT.100.10) GO TO 64
      IF(ABS((VTMP(N,M)-TMPS(N,M))/TMPS(N,M)).GT.TOL)GO TO 65           22241
   64 CONTINUE                                                          22241
      GO TO 67                                                          22241
   65 DO 83 J=1,JMAX                                                    22241
      DO 83 K=1,IMAX                                                    22241
   83 VTMP(J,K)=TMPS(J,K)                                               22241
      KWHO=1
      GO TO 82                                                          22241
   67 DO 66 J=1,JMAX                                                    22241
      DO 66 K=1,IMAX                                                    22241
      IF(TMPS(J,K).LT.100.10) GO TO 66
      CAP(J,K)=CAPT(J,K)
      TEEP(J,K)=TEMP(J,K)
      CAPPY(J,K)=CAPY(J,K)
      TEMP(J,K)=TMPS(J,K)
   66 CONTINUE
      GO TO(152,153),KHT                                                THERMAB6
  152 TIMHT=TIMHT+DELTT                                                 THERMAB6
      HTIN=HTIN+ALPHA*TT*DELTT*2.
      HTOUT=HTOUT+TEMP(1,IMAX)*DELTT*ALPHA*2.
C     THIS ACCOUNTS FOR HEAT OUT OF PRODUCER                            22241
  153 IF(TEMP(1,IMAX).LE.900.)GO TO 48
      IF(KHT.EQ.2) GO TO 174
      KWTSP=2                                                           THERMAB6
  174 KHT=2                                                             THERMAB6
   48 IF(TEMP(1,IMAX).GE.800.)GO TO 49
      TMPS(1,1)=TT
      TEMP(1,1)=TT                                                      22241
      VTMP(1,1)=TT                                                      22241
      TEMP(JMAX,IMAX)=TTTT                                              22241
      TMPS(JMAX,IMAX)=TTTT
      VTMP(JMAX,IMAX)=TTTT
      IF(KHT.EQ.1) GO TO 175
      KWTSP=2                                                           THERMAB6
  175 KHT=1                                                             THERMAB6
      IF(KHT.EQ.2)GO TO 183
   49 PPP=PPP+1.0
      IF(PPP.LT.19.)GO TO 741
      IF(PPP.EQ.19.)GO TO 381
      IF(PPP.NE.20.)GO TO 741
      PPP=0.0
  381 JMAX=JMAX-2                                                       22241
      IMAX=IMAX-2                                                       22241
C     THIS ACCOUNTS FOR THE HEAT STORED IN ROCK                         22241
      DO 40 J=1,JMAX                                                    22241
      SUM=0.                                                            22241
      DO 80 K=1,IMAX,2                                                  22241
   80 SUM=SUM+CAP(J,K)+4.*CAP(J,K+1)+CAP(J,K+2)
   40 AREA(J)=SUM*(DELTX/3.)                                            22241
      BB2=0.                                                            22241
      DO 90 J=1,JBMAX,2                                                 22241
      H(J)=Y(J+1)-Y(J)                                                  22241
      HA(J)=(Y(J+2)-Y(J+1))/H(J)                                        22241
   90 BB2=BB2+(H(J)*(HA(J)+1.)/(6.*HA(J)))*((2.*HA(J)-1.)*AREA(J+2)+(HA( 22241
     1J)+1.)**2*AREA(J+1)+HA(J)*(2.-HA(J))*AREA(J))                     22241
      ER2=((BB2-HEAT-HTORG)+2.*(TEMP(1,IMAX)*DELTT*ALPHA))/(ALPHA*TT*
     1DELTT*2.)
      HETIN=(BB2-HEAT-HTORG)/(ALPHA*TT*DELTT*2.)
      HEAT=BB2-HTORG
  741 HTTUT=HTOUT+HEAT+HTORG
      HTINN=HTIN+HTORG
      ERR=ABS((HTINN-HTTOT)/HTTOT)
      ER1=ABS((HTINN-HTTOT)/HTINN)
  183 GO TO(4,5),KSLCE                                                  THERMAB3
C     THIS ACCOUNTS FOR THE TEMPERATURE AND CULULATIVE HEAT PRINT OUT AT
C     VARIOUS SELECTED POSITIONS ALONG THE FRACTURE.
    4 WRITE(6,26)                                                       THERMAB3
   26 FORMAT(1H1,30X,'INPUT DATA FOR CROSS SECTION STUDY'//9X,'LOCATION(THERMAB3
     1FEET DOWN FRACTURE)',2X,'TEMPERATURE',9X,'TIME(HOURS)',9X,'CUMULATTHERMAB3
     2IVE HEAT'//)                                                      THERMAB3
      JBMAX=JMAX-2                                                      THERMAB3
      DO 70 K=1,IMAX                                                    THERMAB3
      DO 70 M=1,ISSN                                                    THERMAB3
      IF(K.NE.IS(M+1))GO TO 70
      BB3=0.0                                                           THERMAB3
      DO 71 J=1,JBMAX,2                                                 THERMAB3
      H(J)=Y(J+1)-Y(J)                                                  THERMAB3
      HA(J)=(Y(J+2)-Y(J+1))/H(J)                                        22241
   71 BB3=BB3+(H(J)*(HA(J)+1.)/(6.*HA(J)))*((2.*HA(J)-1.)*CAP(J+2,K)+(H
     1A(J)+1.)**2*CAP(J+1,K)+HA(J)*(2.-HA(J))*CAP(J,K)))*DELTX
      AIS=IS(M)
      ALOCF=DELTX*AIS
      WRITE(6,25)ALOCF,TEMP(1,K),TIME,BB3                               THERMAB3
   25 FORMAT(9X,F10.1,20X,F10.1,10X,F10.1,7X,E18.8)                     THERMAB3
   70 CONTINUE                                                          THERMAB3
    5 TIME=TIME+DELTT                                                   22241
      IF(TIME.GT.3000.)GO TO 56
      TOL=.05
      GO TO 55
   56 IF(TIME.GT.4000.)GO TO 57
      TOL=.04
      GO TO 55
   57 IF(TIME.GT.5000.)GO TO 58
      TOL=.03
      GO TO 55
   58 IF(TIME.GT.CM)GO TO 59
      TOL=.02
      GO TO 55
   59 TOL=.01
   55 ATMP=TIME
      OD=OD+1.
      PP=PP+1.0
      IF(PP.NE.20.) GO TO 891
      PP=0.
      WRITE(6,33)                                                       22241
   33 FORMAT(1H1,2X2HJ=4X2H 15X2H 25X2H 35X2H 45X2H 55X2H 65X2H 75X2H 85  22241
     1X2H 95X2H105X2H115X2H125X2H135X2H145X2H15)                        22241
      DO 92 K=1,42
   92 WRITE(6,37)K,(TEMP(J,K),J=1,17)
   37 FORMAT(I4,17F7.1)
      WRITE(6,250)
      DO 290 K=1,42
  290 WRITE(6,252)K,(TEMP(J,K),J=18,JMAX)
      WRITE(6,33)                                                       THERMAB3
      DO 24 K=43,IMAX                                                   THERMAB3
   24 WRITE(6,37)K,(TEMP(J,K),J=1,17)
      WRITE(6,250)
      DO 299 K=43,IMAX
  299 WRITE(6,252)K,(TEMP(J,K),J=18,JMAX)
  887 WRITE(6,30)TIME,ERR,HEAT,HTUUT,HTIN,HTTOT,HTORG,DELTT,ER1,ER2,TOL,
     1HETIN
   30 FORMAT(F10.1,2X,5HHOURS/F10.5,2X,27HDISCREPANCY IN HEAT BALANCE/E1  22241
     18.8,2X,31HBTU OF HEAT STORED IN RESERVOIR/E18.8,2X,20HBTU OF HEAT   22241
     2PRODUCED/E18.8,2X,20HBTU OF HEAT INJECTED/E18.8,2X,5HHTTOT/E18.8,2  22241
     3X,5HHTORG/F10.1,2HTIME INCREMENT, HOURS/F10.5,3X,3HER1/F10.5,3X,3
     4HER2/F10.5,3X,9HTOLERANCE/E18.8,2X,'=INSTANTEOUS HEAT STORED')
      WRITE(6,223)DD,KIT
      CALL CONTR
  891 KIT=0
      IIT=0
      IF(TIME.GT.TIMX)GO TO 100
      GO TO(86,65),KK                                                   22241
  222 WRITE(6,223)DD,KIT
  223 FORMAT(3X,F10.0,I5)
   69 STOP                                                              22241
```

```
      END                                                                   22241
      SUBROUTINE CALCY                                                      22241
      DIMENSION TMPS(25,51),TEMP(25,51),W(25,51),G(25,51),Y(25),V(51),YP
     1LUS(25),YNEG(25),YMID(25),TTMPS(25,51),AREA(25),H(25),HA(25),VTMP(
     225,51),CAPT(25,51),CAP(25,51),COND(25,51),IS(10),SLTMP(10),DIST(8,
     351),TEEP(25,51),CAPY(25,51),CAPPY(25,51)
      COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH
     1A,DELTX,DENOM,DELTY,TTMPS,CAP,CAPT,TIME,LMAX,SLTMP,DIST,TEEP,CAPY,
     2CAPPY
      NMAX=JMAX-1                                                           22241
      DO 2 J=1,NMAX                                                         22241
    2 YPLUS(J)=Y(J+1)-Y(J)                                                  22241
      DO 3 J=2,JMAX                                                         22241
    3 YNEG(J)=Y(J)-Y(J-1)                                                   22241
      YNEG(1)=YPLUS(1)                                                      22241
      YPLUS(JMAX)=YNEG(JMAX)                                                22241
      DO 4 J=1,JMAX                                                         22241
    4 YMID(J)=0.5*(YPLUS(J)+YNEG(J))                                        22241
      RETURN                                                                22241
      END                                                                   22241
      SUBROUTINE CONV                                                       22241
      DIMENSION TMPS(25,51),TEMP(25,51),W(25,51),G(25,51),Y(25),V(51),YP
     1LUS(25),YNEG(25),YMID(25),TTMPS(25,51),AREA(25),H(25),HA(25),VTMP(
     225,51),CAPT(25,51),CAP(25,51),COND(25,51),IS(10),SLTMP(10),DIST(8,
     351),TEEP(25,51),CAPY(25,51),CAPPY(25,51)
      COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH
     1A,DELTX,DENOM,DELTY,TTMPS,CAP,CAPT,TIME,LMAX,SLTMP,DIST,TEEP,CAPY,
     2CAPPY
      DO 40 K=2,IMAX                                                        22241
      CONMIN=(COND(1,K)+COND(2,K))/2.
      CONPUS=(COND(2,K)+COND(3,K))/2.
      D13=DELTX*TEMP(1,K)/(DELTT*V(K))
      D14=(3.*DELTX*CONMIN*TMPS(2,K))/(2.*DELTY*ALPHA)
      D15=(DELTX*CONPUS*(TMPS(2,K)-TMPS(3,K)))/(2.*DELTY*ALPHA)
      D16=1.+(DELTX/(DELTT*V(K)))
      D17=(3.*DELTX*CONMIN)/(2.*DELTY*ALPHA)
      TMPS(1,K)=(TMPS(1,K-1)+D13+D14+D15)/(D16+D17)
      IF(TMPS(1,K).GE.100.)GO TO 40
      TMPS(1,K)=100.
   40 CONTINUE                                                              22241
      RETURN                                                                22241
      END                                                                   22241
      SUBROUTINE COND1                                                      22241
      DIMENSION TMPS(25,51),TEMP(25,51),W(25,51),G(25,51),Y(25),V(51),YP
     1LUS(25),YNEG(25),YMID(25),TTMPS(25,51),AREA(25),H(25),HA(25),VTMP(
     225,51),CAPT(25,51),CAP(25,51),COND(25,51),IS(10),SLTMP(10),DIST(8,
     351),TEEP(25,51),CAPY(25,51),CAPPY(25,51)
      COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH
     1A,DELTX,DENOM,DELTY,TTMPS,CAP,CAPT,TIME,LMAX,SLTMP,DIST,TEEP,CAPY,
     2CAPPY
      MMAX=JMAX-1
      DO 1 J=2,MMAX
      DO 20 K=1,IMAX                                                        22241
   51 IF(J.EQ.JMAX)GO TO 70                                                 22240
      IF(K.EQ.1)GO TO 75                                                    22240
      IF(K.EQ.IMAX)GO TO 76                                                 22240
      CXPUS=(COND(J,K)+COND(J,K+1))/2.                                      22240
      CXMIN=(COND(J,K)+COND(J,K-1))/2.                                      22240
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                      22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                      22240
      GO TO 77                                                              22240
   70 IF(K.EQ.1)GO TO 72                                                    22240
      IF(K.EQ.IMAX)GO TO 73                                                 22240
      CXPUS=(COND(J,K)+COND(J,K+1))/2.                                      22240
      CXMIN=(COND(J,K)+COND(J,K-1))/2.                                      22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                      22240
      CYPUS=CYMIN                                                           22240
      GO TO 77                                                              22240
   72 CXPUS=(COND(J,K)+COND(J,K+1))/2.                                      22240
      CXMIN=CXPUS                                                           22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                      22240
      CYPUS=CYMIN                                                           22240
      GO TO 77                                                              22240
   73 CXMIN=(COND(J,K)+COND(J,K-1))/2.                                      22240
      CXPUS=CXMIN                                                           22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                      22240
      CYPUS=CYMIN                                                           22240
      GO TO 77                                                              22240
   75 CXPUS=(COND(J,K)+COND(J,K+1))/2.                                      22240
      CXMIN=CXPUS                                                           22240
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                      22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                      22240
      GO TO 77                                                              22240
   76 CXMIN=(COND(J,K)+COND(J,K-1))/2.                                      22240
      CXPUS=CXMIN                                                           22240
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                      22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                      22240
   77 CONTINUE                                                              22240
      B1=-(1.+CXMIN/CXPUS+(DELTX**2/DELTT)*(CAPT(J,K)/(TMPS(J,K)*CXPUS))
     1)
      D14=-(DELTX**2/(YMID(J)*CXPUS))                                       22240
   22 D15=CYPUS*((TMPS(J+1,K)-TMPS(J,K))/YPLUS(J))                          22240
   90 D16=CYMIN*((TMPS(J,K)-TMPS(J-1,K))/YNEG(J))                           22240
      D17=(DELTX**2*CAP(J,K))/(DELTT*CXPUS)
      D1=D14*(D15-D16)-D17
      IF(K.NE.1)GO TO 3
      A1=0.0
      C1=1.+CXMIN/CXPUS                                                     22240
      W(J,1)=C1/B1
      G(J,1)=D1/B1
      GO TO 20
    3 IF(K.NE.IMAX)GO TO 5                                                  22241
      C1=0.0
      A1=1.+CXMIN/CXPUS                                                     22240
      GO TO 10
    5 A1=CXMIN/CXPUS                                                        22240
      C1=1.0
   10 DENOM=B1-A1*W(J,K-1)                                                  22241
      IF(DENOM.NE.0.)GO TO 18                                               22241
      WRITE(6,50)
   50 FORMAT(18H ZERO DENOMINATOR.)                                         22241
      GO TO 800
   18 W(J,K)=C1/DENOM
      G(J,K)=(D1-A1*G(J,K-1))/DENOM                                         22241
   20 CONTINUE
      TTMPS(J,IMAX)=G(J,IMAX)
      DO 30 I=2,IMAX
      II=IMAX+1-I
   30 TTMPS(J,II)=G(J,II)-W(J,II)*TTMPS(J,II+1)
    1 CONTINUE
      DO 95 J=2,MMAX
      DO 95 K=1,IMAX
      TMPS(J,K)=TTMPS(J,K)
   95 CONTINUE
      CALL CAPAC
      CALL COND4
  800 RETURN
      END
      SUBROUTINE COND2
      DIMENSION TMPS(25,51),TEMP(25,51),W(25,51),G(25,51),Y(25),V(51),YP
     1LUS(25),YNEG(25),YMID(25),TTMPS(25,51),AREA(25),H(25),HA(25),VTMP(
     225,51),CAPT(25,51),CAP(25,51),COND(25,51),IS(10),SLTMP(10),DIST(8,
     351),TEEP(25,51),CAPY(25,51),CAPPY(25,51)
      COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH
     1A,DELTX,DENOM,DELTY,TTMPS,CAP,CAPT,TIME,LMAX,SLTMP,DIST,TEEP,CAPY,
     2CAPPY
      MMAX=JMAX-1
      NMAX=MMAX-2
      DO 1 K=1,IMAX
      DO 20 J=2,MMAX
   61 IF(J.EQ.JMAX)GO TO 70                                                 22240
      IF(K.EQ.1)GO TO 75                                                    22240
      IF(K.EQ.IMAX)GO TO 76                                                 22240
      CXPUS=(COND(J,K)+COND(J,K+1))/2.                                      22240
      CXMIN=(COND(J,K)+COND(J,K-1))/2.                                      22240
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                      22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                      22240
      GO TO 77                                                              22240
   70 IF(K.EQ.1)GO TO 72                                                    22240
```

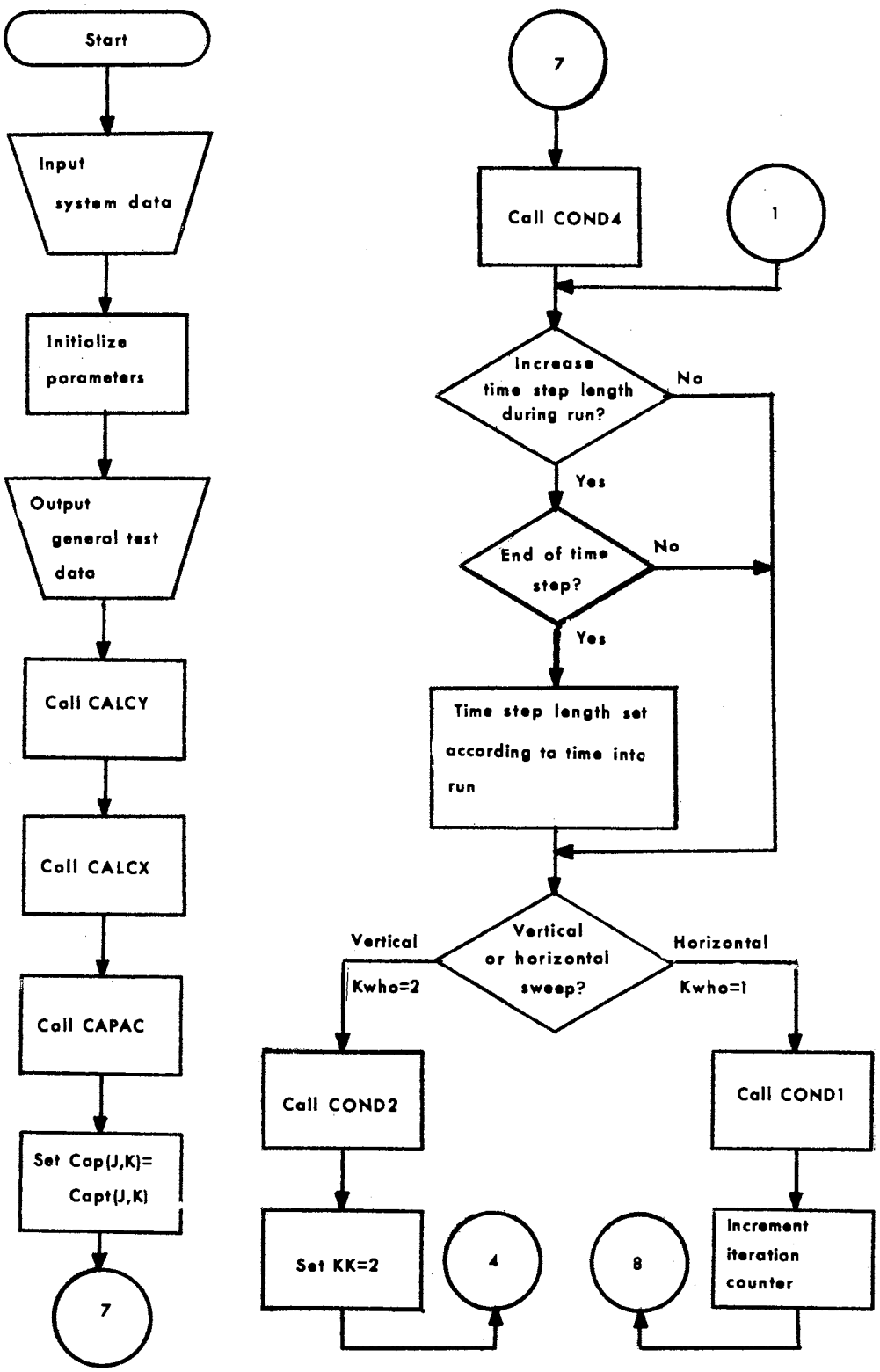
```
      IF(K.EQ.IMAX)GO TO 73                                              22240
      CXPUS=(COND(J,K)+COND(J,K+1))/2.                                   22240
      CXMIN=(COND(J,K)+COND(J,K-1))/2.                                   22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                   22240
      CYPUS=CYMIN                                                        22240
      GO TO 77                                                           22240
   72 CXPUS=(COND(J,K)+COND(J,K+1))/2.                                   22240
      CXMIN=CXPUS                                                        22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                   22240
      CYPUS=CYMIN                                                        22240
      GO TO 77                                                           22240
   73 CXMIN=(COND(J,K)+COND(J,K-1))/2.                                   22240
      CXPUS=CXMIN                                                        22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                   22240
      CYPUS=CYMIN                                                        22240
      GO TO 77                                                           22240
   75 CXPUS=(COND(J,K)+COND(J,K+1))/2.                                   22240
      CXMIN=CXPUS                                                        22240
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                   22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                   22240
      GO TO 77                                                           22240
   76 CXMIN=(COND(J,K)+COND(J,K-1))/2.                                   22240
      CXPUS=CXMIN                                                        22240
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                   22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                   22240
   77 CONTINUE                                                           22240
      C1=1.0
      A1=(YPLUS(J)*CYMIN)/(CYPUS*YNEG(J))
      B1=-(1.0+(YPLUS(J)*CYMIN)/(YNEG(J)*CYPUS)+(CAPT(J,K)*YMID(J)*YPLUS
     1(J))/(DELTT*CYPUS*TMPS(J,K)))
      D14=-(YMID(J)*YPLUS(J)/(DELTX**2*CYPUS))
      IF(K.LT.IMAX) GO TO 24
      D15=(TMPS(J,K-1)-TMPS(J,K))*CXPUS
      GO TO 25
   24 D15=(TMPS(J,K+1)-TMPS(J,K))*CXPUS
   25 IF(K.GT.1) GO TO 26
      D16=(TMPS(J,K)-TMPS(J,K+1))*CXMIN
      GO TO 27
   26 D16=(TMPS(J,K)-TMPS(J,K-1))*CXMIN
   27 D17=(CAP(J,K)*YMID(J)*YPLUS(J))/(DELTT*CYPUS)
      D1=D14*(D15-D16)-D17                                               22241
      IF(J.NE.2)GO TO 10
      W(2,K)=C1/B1
      G(2,K)=(D1-A1*TMPS(1,K))/B1
      GO TO 20
   10 DENOM=B1-A1*W(J-1,K)
      IF(DENOM.NE.0.)GO TO 18
      WRITE(6,50)
   50 FORMAT(18H ZERO DENOMINATOR.)                                      22241
      GO TO 800
   18 W(J,K)=C1/DENOM                                                    22241
      IF(J.NE.MMAX)GO TO 19
      D1=D1-C1*TMPS(JMAX,K)
   19 G(J,K)=(D1-A1*G(J-1,K))/DENOM                                      22241
   20 CONTINUE
      TTMPS(MMAX,K)=G(MMAX,K)
      DO 30 I=1,NMAX
      II=MMAX-I
   30 TTMPS(II,K)=G(II,K)-W(II,K)*TTMPS(II+1,K)
    1 CONTINUE
      DO 95 J=2,MMAX
      DO 95 K=1,IMAX
      TMPS(J,K)=TTMPS(J,K)
   95 CONTINUE
      CALL CAPAC
      CALL COND4
  800 RETURN
      END
      SUBROUTINE CAPAC
      DIMENSION TMPS(25,51),TEMP(25,51),W(25,51),G(25,51),Y(25),V(51),YP
     1LUS(25),YNEG(25),YMID(25),TTMPS(25,51),AREA(25),H(25),HA(25),VTMP(
     225,51),CAPT(25,51),CAP(25,51),COND(25,51),IS(10),SLTMP(10),DIST(8,
     351),TEEP(25,51),CAPY(25,51),CAPPY(25,51)
      COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH
     1A,DELTX,DENOM,DELTY,TTMPS,CAP,CAPT,TIME,LMAX,SLTMP,DIST,TEEP,CAPY,
     2CAPPY
      DO 232 J=1,JMAX
      DO 232 K=1,IMAX
    9 IF(TMPS(J,K).GT.450.)GO TO 233
      CAPT(J,K)=(23.5+.307*(TMPS(J,K)-100.))*134.0
      GO TO 232
  233 IF(TMPS(J,K).GT.900.)GO TO 234
      CAPT(J,K)=(130.0+.344*(TMPS(J,K)-450.))*134.0
      GO TO 232
  234 IF(TMPS(J,K).GT.1100.)GO TO 235
      CAPT(J,K)=134.0*(285.0+.485*(TMPS(J,K)-900.))
      GO TO 232
  235 IF(TMPS(J,K).GT.1600.)GO TO 236
      CAPT(J,K)=134.0*(382.0+.6500*(TMPS(J,K)-1100.))
      GO TO 232
  236 CAPT(J,K)=134.0*(694.0+.268*(TMPS(J,K)-1600.))
  232 CONTINUE
      RETURN
      END
      SUBROUTINE COND4                                                   22242
      DIMENSION TMPS(25,51),TEMP(25,51),W(25,51),G(25,51),Y(25),V(51),YP
     1LUS(25),YNEG(25),YMID(25),TTMPS(25,51),AREA(25),H(25),HA(25),VTMP(
     225,51),CAPT(25,51),CAP(25,51),COND(25,51),IS(10),SLTMP(10),DIST(8,
     351),TEEP(25,51),CAPY(25,51),CAPPY(25,51)
      COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH
     1A,DELTX,DENOM,DELTY,TTMPS,CAP,CAPT,TIME,LMAX,SLTMP,DIST,TEEP,CAPY,
     2CAPPY
      DO 5 J=1,JMAX                                                      22242
      DO 5 K=1,IMAX                                                      22242
    9 IF(TMPS(J,K).GE.200.)GO TO 1
      COND(J,K)=0.9
      GO TO 5
    1 IF(TMPS(J,K).GT.1200.)GO TO 7
      TSQ=TMPS(J,K)*TMPS(J,K)
      TCUB=TMPS(J,K)*TSQ
      COND(J,K)=.12561573E+1-.18377799E-2*TMPS(J,K)+.10902421E-5*TSQ-.15
     1084869E-9*TCUB
      GO TO 5
    7 COND(J,K)=0.36
    5 CONTINUE
      RETURN                                                             22242
      END                                                                22242
      SUBROUTINE COND5
      DIMENSION TMPS(25,51),TEMP(25,51),W(25,51),G(25,51),Y(25),V(51),YP
     1LUS(25),YNEG(25),YMID(25),TTMPS(25,51),AREA(25),H(25),HA(25),VTMP(
     225,51),CAPT(25,51),CAP(25,51),COND(25,51),IS(10),SLTMP(10),DIST(8,
     351),TEEP(25,51),CAPY(25,51),CAPPY(25,51)
      COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH
     1A,DELTX,DENOM,DELTY,TTMPS,CAP,CAPT,TIME,LMAX,SLTMP,DIST,TEEP,CAPY,
     2CAPPY
      DO 5 K=1,IMAX                                                      22242
      IF(TMPS(1,K).GE.200.)GO TO 1
      COND(1,K)=0.9
      GO TO 5
    1 IF(TMPS(1,K).GT.1200.)GO TO 7
      TSQ=TMPS(1,K)*TMPS(1,K)
      TCUB=TMPS(1,K)*TSQ
      COND(1,K)=.12561573E+1-.18377799E-2*TMPS(1,K)+.10902421E-5*TSQ-.15
     1084869E-9*TCUB
      GO TO 5
    7 COND(1,K)=0.36
    5 CONTINUE
      RETURN                                                             22242
      END                                                                22242
      SUBROUTINE CAPC1
      DIMENSION TMPS(25,51),TEMP(25,51),W(25,51),G(25,51),Y(25),V(51),YP
     1LUS(25),YNEG(25),YMID(25),TTMPS(25,51),AREA(25),H(25),HA(25),VTMP(
     225,51),CAPT(25,51),CAP(25,51),COND(25,51),IS(10),SLTMP(10),DIST(8,
     351),TEEP(25,51),CAPY(25,51),CAPPY(25,51)
      COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH
     1A,DELTX,DENOM,DELTY,TTMPS,CAP,CAPT,TIME,LMAX,SLTMP,DIST,TEEP,CAPY,
     2CAPPY
      DO 232 K=1,IMAX
      IF(TMPS(1,K).GT.450.)GO TO 233
```

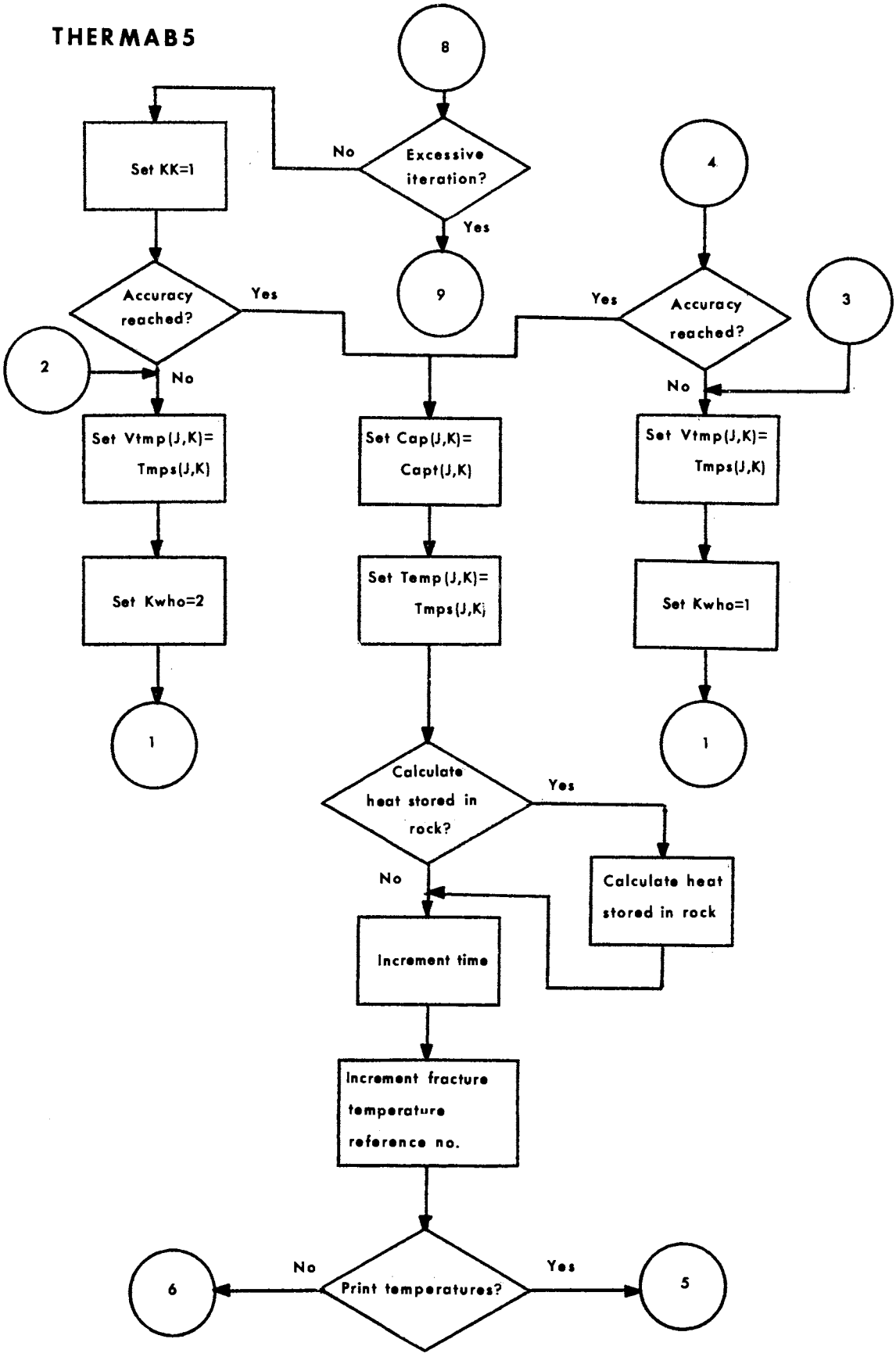
```
      CAPT(1,K)=(23.5+.307*(TMPS(1,K)-100.))*134.0
      GO TO 232
  233 IF(TMPS(1,K).GT.900.)GO TO 234
      CAPT(1,K)=(130.0+.344*(TMPS(1,K)-450.))*134.0
      GO TO 232
  234 IF(TMPS(1,K).GT.1100.)GO TO 235
      CAPT(1,K)=134.0*(285.0+.485*(TMPS(1,K)-900.))
      GO TO 232
  235 IF(TMPS(1,K).GT.1600.)GO TO 236
      CAPT(1,K)=134.0*(382.0+.6500*(TMPS(1,K)-1100.))
      GO TO 232
  236 CAPT(1,K)=134.0*(694.0+.268*(TMPS(1,K)-1600.))
  232 CONTINUE
      RETURN
      END
      SUBROUTINE COND7
      DIMENSION TMPS(25,51),TEMP(25,51),W(25,51),G(25,51),Y(25),V(51),YP
     1LUS(25),YNEG(25),YMID(25),TTMPS(25,51),AREA(25),H(25),HA(25),VTMP(
     225,51),CAPT(25,51),CAP(25,51),COND(25,51),IS(10),SLTMP(10),DIST(8,
     351),TEEP(25,51),CAPY(25,51),CAPPY(25,51)
      COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH
     1A,DELTX,DENOM,DELTY,TTMPS,CAP,CAPT,TIME,LMAX,SLTMP,DIST,TEEP,CAPY,
     2CAPPY
      DO 1 K=1,IMAX
      DO 20 J=1,JMAX
      IF(TMPS(J,K).LT.600.)GO TO 9                                  THERMAB3
      IF(TEEP(J,K).LT.TEMP(J,K))GO TO 9
      CAPT(J,K)=CAPY(J,K)                                           THERMAB3
      CAP(J,K)=CAPPY(J,K)                                           THERMAB3
    9 IF(J.EQ.1)GO TO 78
   61 IF(J.EQ.JMAX)GO TO 70
      IF(K.EQ.1)GO TO 79
      IF(K.EQ.IMAX)GO TO 80
      CXPUS=(COND(J,K)+COND(J,K+1))/2.                              22240
      CXMIN=(COND(J,K)+COND(J,K-1))/2.                              22240
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                              22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                              22240
      GO TO 77                                                      22240
   79 CXPUS=(COND(J,K)+COND(J,K+1))/2.
      CXMIN=CXPUS                                                   22240
      CYPUS=(COND(J,K)+COND(J+1,K))/2.
      CYMIN=(COND(J,K)+COND(J-1,K))/2.
      GO TO 77                                                      22240
   80 CXMIN=(COND(J,K)+COND(J,K-1))/2.
      CXPUS=CXMIN                                                   22240
      CYPUS=(COND(J,K)+COND(J+1,K))/2.
      CYMIN=(COND(J,K)+COND(J-1,K))/2.
      GO TO 77                                                      22240
   78 IF(K.EQ.1)GO TO 75
      IF(K.EQ.IMAX)GO TO 76
      CXPUS=(COND(J,K)+COND(J,K+1))/2.
      CXMIN=(COND(J,K)+COND(J,K-1))/2.
      CYPUS=(COND(J,K)+COND(J+1,K))/2.
      CYMIN=CYPUS
      GO TO 77                                                      22240
   70 IF(K.EQ.1)GO TO 72
      IF(K.EQ.IMAX)GO TO 73
      CXPUS=(COND(J,K)+COND(J,K+1))/2.
      CXMIN=(COND(J,K)+COND(J,K-1))/2.
      CYMIN=(COND(J,K)+COND(J-1,K))/2.
      CYPUS=CYMIN                                                   22240
      GO TO 77                                                      22240
   72 CXPUS=(COND(J,K)+COND(J,K+1))/2.
      CXMIN=CXPUS                                                   22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.
      CYPUS=CYMIN                                                   22240
      GO TO 77
   73 CXMIN=(COND(J,K)+COND(J,K-1))/2.
      CXPUS=CXMIN                                                   22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.
      CYPUS=CYMIN                                                   22240
      GO TO 77
   75 CXPUS=(COND(J,K)+COND(J,K+1))/2.
      CXMIN=CXPUS
      CYPUS=(COND(J,K)+COND(J+1,K))/2.
      CYMIN=CYPUS
      GO TO 77
   76 CXMIN=(COND(J,K)+COND(J,K-1))/2.
      CXPUS=CXMIN
      CYPUS=(COND(J,K)+COND(J+1,K))/2.
      CYMIN=CYPUS
   77 CONTINUE                                                      22240
      A1=(YPLUS(J)*CYMIN)/(CYPUS*YNEG(J))                           22240
      C1=1.
      B1=-(1.0+(YPLUS(J)*CYMIN)/(YNEG(J)*CYPUS)+(CAPT(J,K)*YMID(J)*YPLUS
     1(J))/(DELTT*CYPUS*TMPS(J,K)))
      D14=-(YMID(J)*YPLUS(J)/(DELTX**2*CYPUS))
      IF(K.LT.IMAX) GO TO 24
      D15=(TMPS(J,K-1)-TMPS(J,K))*CXPUS
      GO TO 25
   24 D15=(TMPS(J,K+1)-TMPS(J,K))*CXPUS
   25 IF(K.GT.1) GO TO 26
      D16=(TMPS(J,K)-TMPS(J,K+1))*CXMIN
      GO TO 27
   26 D16=(TMPS(J,K)-TMPS(J,K-1))*CXMIN
   27 D17=(CAP(J,K)*YMID(J)*YPLUS(J))/(DELTT*CYPUS)
      D1=D14*(D15-D16)-D17                                          22241
      IF(J.NE.1)GO TO 3
      C1=C1+A1
      A1=0.0
      W(1,K)=C1/B1
      G(1,K)=D1/B1
      GO TO 20
    3 IF(J.NE.JMAX)GO TO 10                                         22241
      A1=A1+C1
      C1=0.0
   10 DENOM=B1-A1*W(J-1,K)
      IF(DENOM.NE.0.)GO TO 18
      WRITE(6,50)
   50 FORMAT(18H ZERO DENOMINATOR.)                                 22241
      GO TO 800
   18 W(J,K)=C1/DENOM                                               22241
      G(J,K)=(D1-A1*G(J-1,K))/DENOM                                 22241
   20 CONTINUE
      TTMPS(JMAX,K)=G(JMAX,K)
      NMAX=JMAX-1
      DO 30 I=1,NMAX
      II=JMAX-I
   30 TTMPS(II,K)=G(II,K)-W(II,K)*TTMPS(II+1,K)
    1 CONTINUE
      DO 95 J=1,JMAX
      DO 95 K=1,IMAX
      TMPS(J,K)=TTMPS(J,K)
   95 CONTINUE
      CALL CAPAC
      CALL COND4
      CALL CAPSK                                                    THERMAB3
  800 RETURN
      END
      SUBROUTINE COND6
      DIMENSION TMPS(25,51),TEMP(25,51),W(25,51),G(25,51),Y(25),V(51),YP  22241
     1LUS(25),YNEG(25),YMID(25),TTMPS(25,51),AREA(25),H(25),HA(25),VTMP(
     225,51),CAPT(25,51),CAP(25,51),COND(25,51),IS(10),SLTMP(10),DIST(8,
     351),TEEP(25,51),CAPY(25,51),CAPPY(25,51)
      COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH
     1A,DELTX,DENOM,DELTY,TTMPS,CAP,CAPT,TIME,LMAX,SLTMP,DIST,TEEP,CAPY,
     2CAPPY
      DO 1 J=1,JMAX                                                 22242
      DO 20 K=1,IMAX                                                22241
      IF(TMPS(J,K).LT.600.)GO TO 9                                  THERMAB3
      IF(TEEP(J,K).LT.TEMP(J,K))GO TO 9
      CAPT(J,K)=CAPY(J,K)                                           THERMAB3
      CAP(J,K)=CAPPY(J,K)                                           THERMAB3
    9 IF(J.EQ.1)GO TO 78                                            22242
   51 IF(J.EQ.JMAX)GO TO 70                                         22242
      IF(K.EQ.1)GO TO 79                                            22242
      IF(K.EQ.IMAX)GO TO 80                                         22242
      CXPUS=(COND(J,K)+COND(J,K+1))/2.                              22240
      CXMIN=(COND(J,K)+COND(J,K-1))/2.                              22240
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                              22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                              22240
      GO TO 77                                                      22240
   79 CXPUS=(COND(J,K)+COND(J,K+1))/2.                              22241
      CXMIN=CXPUS                                                   22241
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                              22241
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                              22241
      GO TO 77
   80 CXMIN=(COND(J,K)+COND(J,K-1))/2.
      CXPUS=CXMIN
      CYPUS=(COND(J,K)+COND(J+1,K))/2.
      CYMIN=(COND(J,K)+COND(J-1,K))/2.
      GO TO 77                                                      22240
   78 IF(K.EQ.1)GO TO 75
      IF(K.EQ.IMAX)GO TO 76
      CXPUS=(COND(J,K)+COND(J,K+1))/2.
```

```
      CXMIN=(COND(J,K)+COND(J,K-1))/2.
      CYPUS=(COND(J,K)+COND(J+1,K))/2.
      CYMIN=CYPUS
      GO TO 77                                                    22240
   70 IF(K.EQ.1)GO TO 72
      IF(K.EQ.IMAX)GO TO 73
      CXPUS=(COND(J,K)+COND(J,K+1))/2.
      CXMIN=(COND(J,K)+COND(J,K-1))/2.
      CYMIN=(COND(J,K)+COND(J-1,K))/2.
      CYPUS=CYMIN
      GO TO 77                                                    22240
   72 CXPUS=(COND(J,K)+COND(J,K+1))/2.
      CXMIN=CXPUS                                                 22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.
      CYPUS=CYMIN                                                 22240
      GO TO 77                                                    22240
   73 CXMIN=(COND(J,K)+COND(J,K-1))/2.
      CXPUS=CXMIN                                                 22240
      CYMIN=(COND(J,K)+COND(J-1,K))/2.
      CYPUS=CYMIN
      GO TO 77
   75 CXPUS=(COND(J,K)+COND(J,K+1))/2.
      CXMIN=CXPUS
      CYPUS=(COND(J,K)+COND(J+1,K))/2.
      CYMIN=CYPUS
      GO TO 77
   76 CXMIN=(COND(J,K)+COND(J,K-1))/2.
      CXPUS=CXMIN
      CYPUS=(COND(J,K)+COND(J+1,K))/2.
      CYMIN=CYPUS
   77 CONTINUE                                                    22240
      B1=-(1.+CXMIN/CXPUS+(DELTX**2/DELTT)*(CAPT(J,K)/(TMPS(J,K)*CXPUS))
     1)
      D14=-(DELTX**2/(YMID(J)*CXPUS))                             22240
      IF(J.LT.JMAX)GO TO 22                                       22241
      D15=CYPUS*((TMPS(J-1,K)-TMPS(J,K))/YPLUS(J))                22240
      GO TO 25
   22 D15=CYPUS*((TMPS(J+1,K)-TMPS(J,K))/YPLUS(J))                22240
   25 IF(J.GT.1) GO TO 90
      D16=(TMPS(J,K)-TMPS(J+1,K))*CYMIN/YNEG(J)
      GO TO 23
   90 D16=CYMIN*((TMPS(J,K)-TMPS(J-1,K))/YNEG(J))                 22240
   23 D17=(DELTX**2*CAP(J,K))/(DELTT*CXPUS)
      D1=D14*(D15-D16)-D17
      IF(K.NE.1)GO TO 3
      A1=0.0
      C1=1.+CXMIN/CXPUS                                           22240
      W(J,1)=C1/B1
      G(J,1)=D1/B1
      GO TO 20
    3 IF(K.NE.IMAX)GO TO 5                                        22241
      C1=0.0
      A1=1.+CXMIN/CXPUS                                           22240
      GO TO 10
    5 A1=CXMIN/CXPUS                                              22240
      C1=1.0
   10 DENOM=B1-A1*W(J,K-1)                                        22241
      IF(DENOM.NE.0.)GO TO 18                                     22241
      WRITE(6,50)
   50 FORMAT(18H ZERO DENOMINATOR.)                               22241
      GO TO 800
   18 W(J,K)=C1/DENOM
      G(J,K)=(D1-A1*G(J,K-1))/DENOM                               22241
   20 CONTINUE
      TTMPS(J,IMAX)=G(J,IMAX)
      DO 30 I=2,IMAX
      II=IMAX+1-I
   30 TTMPS(J,II)=G(J,II)-W(J,II)*TTMPS(J,II+1)
    1 CONTINUE
      DO 95 J=1,JMAX
      DO 95 K=1,IMAX
      TMPS(J,K)=TTMPS(J,K)
   95 CONTINUE
      CALL CAPAC
      CALL COND4
      CALL CAPSK                                                  THERMAB3
  800 RETURN
      END
      SUBROUTINE CONTR
      DIMENSION TMPS(25,51),TEMP(25,51),W(25,51),G(25,51),Y(25),V(51),YP
     1LUS(25),YNEG(25),YMID(25),TTMPS(25,51),AREA(25),H(25),HA(25),VTMP(
     225,51),CAPT(25,51),CAP(25,51),COND(25,51),IS(10),SLTMP(10),DIST(8,
     351),TEEP(25,51),CAPY(25,51),CAPPY(25,51)
      COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH
     1A,DELTX,DENOM,DELTY,TTMPS,CAP,CAPT,TIME,LMAX,SLTMP,DIST,TEEP,CAPY,
     2CAPPY
      DO 263 I=1,LMAX
      DO 260 K=1,IMAX
      DO 261 J=1,JMAX
      IF(TEMP(J,K).LT.SLTMP(I))GO TO 262
  261 CONTINUE
      J=JMAX
  262 IF(J.EQ.1)GO TO 260
      DIST(I,K)=Y(J-1)+((TEMP(J-1,K)-SLTMP(I))/(TEMP(J-1,K)-TEMP(J,K)))*
     1(Y(J)-Y(J-1))
  260 CONTINUE
  263 CONTINUE
      WRITE(6,264)(SLTMP(I),I=1,LMAX)
  264 FORMAT(1H1,6X,'DISTANCE',7X,'ISOTHERM  ISOTHERM  ISOTHERM  ISOTHERTHERMAB6
     1M  ISOTHERM  ISOTHERM  ISOTHERM  ISOTHERM'/8X,'DOWN',9X,'DEGREES FTHERMAB6
     2 DEGREES F DEGREES F DEGREES F DEGREES F DEGREES F DEGREES F DEGRETHERMAB6
     3EES F'/7X,'FRACTURE',4X,8F10.1/9X,'FEET'/20X,'FEET INTO FEET INTO THERMAB6
     4FEET INTO FEET INTO FEET INTO FEET INTO FEET INTO FEET INTO'/22X,'THERMAB6
     5WALL',6X,'WALL',6X,'WALL',6X,'WALL',6X,'WALL',6X,'WALL',6X,'WALL',THERMAB6
     66X,'WALL')                                                  THERMAB6
      DO 266 K=1,40                                               THERMAB6
      FK2=K-1                                                     THERMAB6
      XFT=DELTX*FK2                                               THERMAB6
  266 WRITE(6,267)XFT,(DIST(I,K),I=1,LMAX)                        THERMAB6
  267 FORMAT(7X,F6.0,4X,8F10.2)
      WRITE(6,264)(SLTMP(I),I=1,LMAX)
      DO 265 K=41,IMAX                                            THERMAB6
      FK2=K-1                                                     THERMAB6
      XFT=DELTX*FK2                                               THERMAB6
  265 WRITE(6,267)XFT,(DIST(I,K),I=1,LMAX)                        THERMAB6
      RETURN
      END
      SUBROUTINE CAPSK                                            THERMAB3
      DIMENSION TMPS(25,51),TEMP(25,51),W(25,51),G(25,51),Y(25),V(51),YP   22241
     1LUS(25),YNEG(25),YMID(25),TTMPS(25,51),AREA(25),H(25),HA(25),VTMP(   22241
     225,51),CAPT(25,51),CAP(25,51),COND(25,51),IS(10),SLTMP(10),DIST(8,
     351),TEEP(25,51),CAPY(25,51),CAPPY(25,51)
      COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH   22241
     1A,DELTX,DENOM,DELTY,TTMPS,CAP,CAPT,TIME,LMAX,SLTMP,DIST,TEEP,CAPY,
     2CAPPY
      DO 232 J=1,JMAX
      DO 232 K=1,IMAX
      IF(TMPS(J,K).GT.450.)GO TO 233
      CAPY(J,K)=(23.5+.307*(TMPS(J,K)-100.))*134.0
      GO TO 232
  233 IF(TMPS(J,K).GT.600.)GO TO 234
      CAPY(J,K)=(130.0+.344*(TMPS(J,K)-450.))*134.0
      GO TO 232
  234 IF(TMPS(J,K).GT.1100.)GO TO 236
      CAPY(J,K)=134.0*(180.0+.230*(TMPS(J,K)-600.))
      GO TO 232
  236 CAPY(J,K)=134.0*(295.0+.258*(TMPS(J,K)-1100.))
  232 CONTINUE
      RETURN
      END
      SUBROUTINE CAPC2
      DIMENSION TMPS(25,51),TEMP(25,51),W(25,51),G(25,51),Y(25),V(51),YP
     1LUS(25),YNEG(25),YMID(25),TTMPS(25,51),AREA(25),H(25),HA(25),VTMP(
     225,51),CAPT(25,51),CAP(25,51),COND(25,51),IS(10),SLTMP(10),DIST(8,
     351),TEEP(25,51),CAPY(25,51),CAPPY(25,51)
      COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH
     1A,DELTX,DENOM,DELTY,TTMPS,CAP,CAPT,TIME,LMAX,SLTMP,DIST,TEEP,CAPY,
     2CAPPY
      DO 232 I=1,IMAX
```

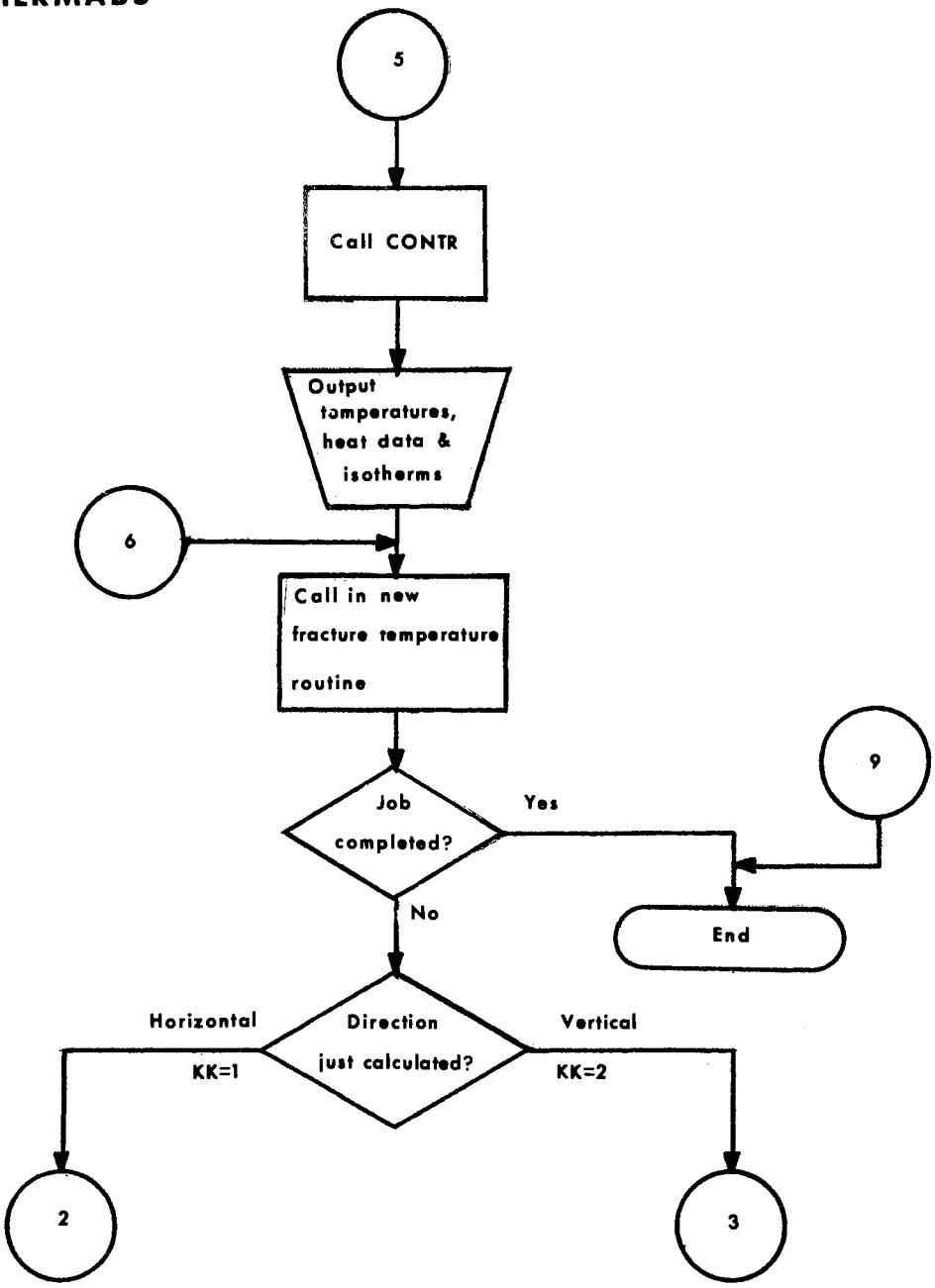
```
      K=IMAX+1-I
      IF(TMPS(JMAX,K).GT.450.) GO TO 233
      CAPT(JMAX,K)=(23.5+.307*(TMPS(JMAX,K)-100.))*134.0
      GO TO 232
  233 IF(TMPS(JMAX,K).GT.900.) GO TO 234
      CAPT(JMAX,K)=(130.0+.344*(TMPS(JMAX,K)-450.))*134.0
      GO TO 232
  234 IF(TMPS(JMAX,K).GT.1100.) GO TO 235
      CAPT(JMAX,K)=134.0*(285.0+.485*(TMPS(JMAX,K)-900.))
      GO TO 232
  235 IF(TMPS(JMAX,K).GT.1600.) GO TO 236
      CAPT(JMAX,K)=134.0*(382.0+.6500*(TMPS(JMAX,K)-1100.))
      GO TO 232
  236 CAPT(JMAX,K)=134.0*(694.0+.268*(TMPS(JMAX,K)-1600.))
  232 CONTINUE
      RETURN
      END
      SUBROUTINE COND8
      DIMENSION TMPS(25,51),TEMP(25,51),W(25,51),G(25,51),Y(25),V(51),YP
     1LUS(25),YNEG(25),YMID(25),TTMPS(25,51),AREA(25),H(25),HA(25),VTMP(
     225,51),CAPT(25,51),CAP(25,51),COND(25,51),IS(10),SLTMP(10),DIST(8,
     351),TEEP(25,51),CAPY(25,51),CAPPY(25,51)
      COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH
     1A,DELTX,DENOM,DELTY,TTMPS,CAP,CAPT,TIME,LMAX,SLTMP,DIST,TEEP,CAPY,
     2CAPPY
      DO 5 I=1,IMAX
      K=IMAX+1-I
      IF(TMPS(JMAX,K).GE.200.) GO TO 1
      COND(JMAX,K)=0.9
      GO TO 5
    1 IF(TMPS(JMAX,K).GT.1200.) GO TO 7
      TSQ=TMPS(JMAX,K)*TMPS(JMAX,K)
      TCUB=TMPS(JMAX,K)*TSQ
      COND(JMAX,K)=.12561573E+1-.18377799E-2*TMPS(JMAX,K)+.10902421E-5*T
     1SQ-.15084869E-9*TCUB
      GO TO 5
    7 COND(JMAX,K)=0.36
    5 CONTINUE
      RETURN
      END
      SUBROUTINE CONV2
      DIMENSION TMPS(25,51),TEMP(25,51),W(25,51),G(25,51),Y(25),V(51),YP   22241
     1LUS(25),YNEG(25),YMID(25),TTMPS(25,51),AREA(25),H(25),HA(25),VTMP(   22241
     225,51),CAPT(25,51),CAP(25,51),COND(25,51),IS(10),SLTMP(10),DIST(8,
     351),TEEP(25,51),CAPY(25,51),CAPPY(25,51)
      COMMON TMPS,TEMP,W,G,Y,V,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,ALPH   22241
     1A,DELTX,DENOM,DELTY,TTMPS,CAP,CAPT,TIME,LMAX,SLTMP,DIST,TEEP,CAPY,
     2CAPPY
      DO 40 I=2,IMAX
      K=IMAX+1-I
      CONMIN=(COND(JMAX,K)+COND(JMAX-1,K))/2.
      CONPUS=(COND(JMAX-1,K)+COND(JMAX-2,K))/2.
      D13=DELTX*TEMP(JMAX,K)/(DELTT*V(I))
      D14=(3.*DELTX*CONMIN*TMPS(JMAX-1,K))/(2.*DELTY*ALPHA)
      D15=(DELTX*CONPUS*(TMPS(JMAX-1,K)-TMPS(JMAX-2,K)))/(2.*DELTY*ALPHA
     1)
      D16=1.+(DELTX/(DELTT*V(I)))
      D17=(3.*DELTX*CONMIN)/(2.*DELTY*ALPHA)
      TMPS(JMAX,K)=(TMPS(JMAX,K+1)+D13+D14+D15)/(D16+D17)
      IF(TMPS(JMAX,K).GE.100.) GO TO 40
      TMPS(JMAX,K)=100.0
   40 CONTINUE
      RETURN
      END
```

THERMAB5


Start
Input system data
Initialize parameters
Output general test data
Call CALCY
Call CALCX
Call CAPAC
Set Cap(J,K)= Capt(J,K)
7
7
Call COND4
1
Increase time step length during run?
No
Yes
End of time step?
No
Yes
Time step length set according to time into run
Vertical or horizontal sweep?
Vertical
Kwho=2
Horizontal
Kwho=1
Call COND2
Call COND1
Set KK=2
4
8
Increment iteration counter

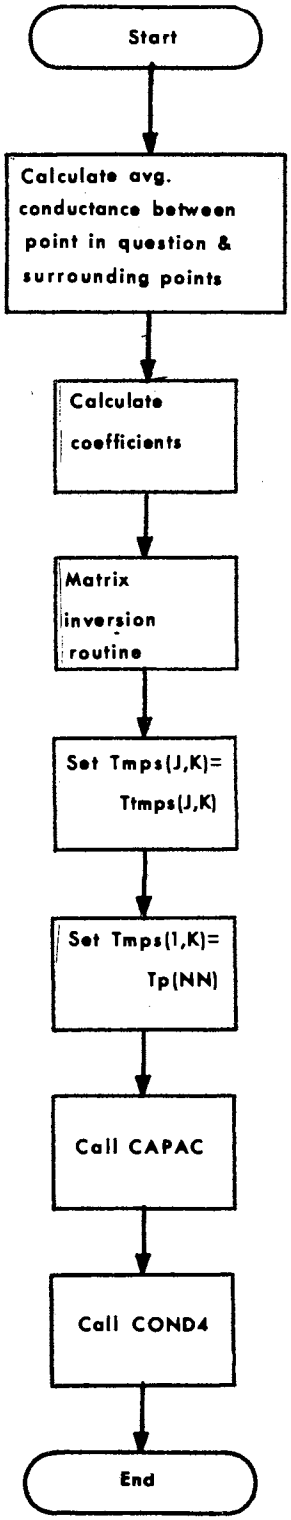

THERMAB5
8
Set KK=1
No
Excessive iteration?
Yes
9
4
Accuracy reached?
Yes
3
Yes
Accuracy reached?
2
No
No
Set Vtmp(J,K)= Tmps(J,K)
Set Cap(J,K)= Capt(J,K)
Set Vtmp(J,K)= Tmps(J,K)
Set Kwho=2
Set Temp(J,K)= Tmps(J,K)
Set Kwho=1
1
1
Calculate heat stored in rock?
Yes
Calculate heat stored in rock
No
Increment time
Increment fracture temperature reference no.
No
Print temperatures?
Yes
6
5

THERMAB5

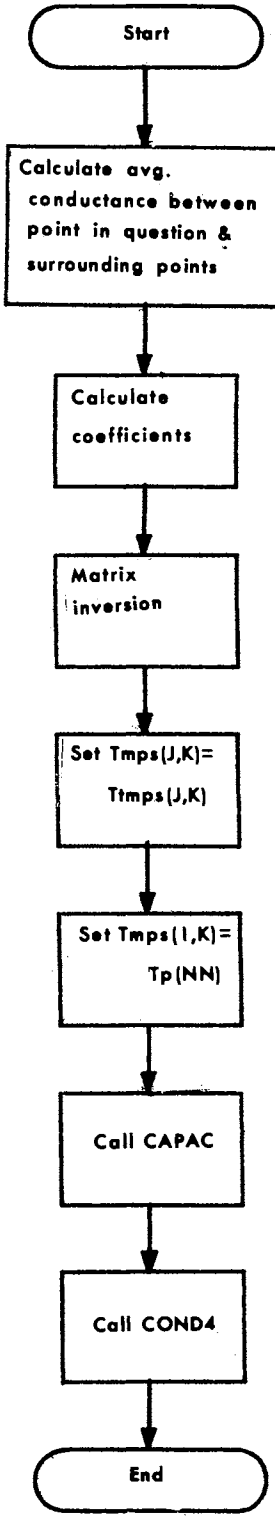

5
Call CONTR
Output temperatures, heat data & isotherms
6
Call in new fracture temperature routine
9
Job completed?
Yes
No
End
Horizontal
KK=1
Direction just calculated?
Vertical
KK=2
2
3

THERMAB5

Subroutine COND1 (Horizontal sweep)

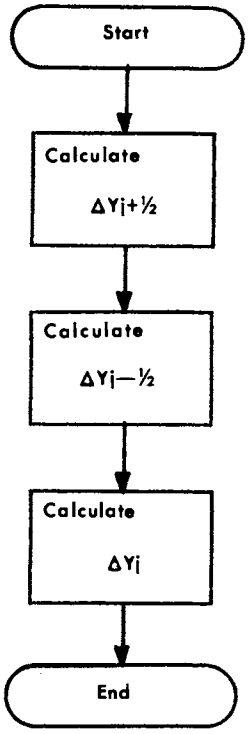


Subroutine COND2 (Vertical sweep)

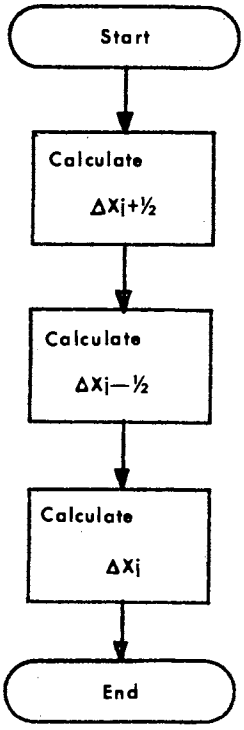


THERMAB5

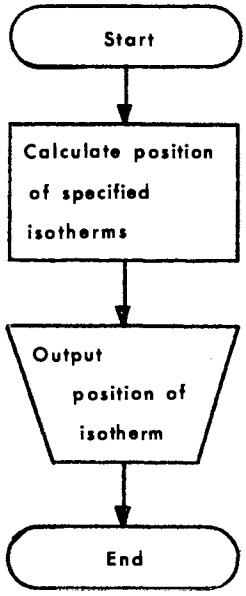
Subroutine CALCY (Variable grid spacing)



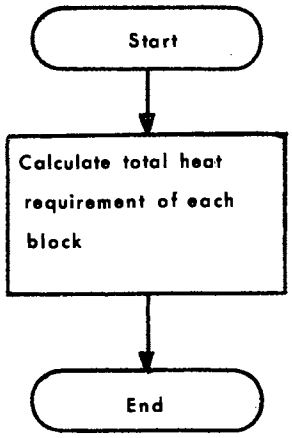
Subroutine CALCX



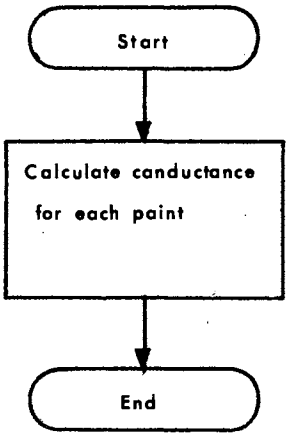
Subroutine CONTR



Subroutine CAPAC



Subroutine COND4



```
C                          THERMAB5-A.L. BARNES
      DIMENSION TMPS(19,23),TEMP(19,23),W(19,23),G(19,23),Y(19),YPLUS(19THERMAB5
     1),YNEG(19),YMID(19),TTMPS(19,23),AREA(19),H(19),HA(19),VTMP(19,23)THERMAB5
     2,CAPT(19,23),CAP(19,23),X(23),XPLUS(23),XNEG(23),XMID(23),COND(19,THERMAB5
     323),TP(1350),E(23),EA(23),SLTMP(8),DIST(8,23)
      COMMON TMPS,TEMP,W,G,Y,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,      THERMAB5
     1DENOM,TTMPS,CAP,CAPT,X,XPLUS,XNEG,XMID,TP,NN,TIME,LMAX,SLTMP,DIST THERMAB5
C      TEMP=PERMANENT TEMPERATURE ARRAY                                  22241
C      TMPS=TEMPORARY TEMPERATURE ARRAY
C      VTMP=TEMPORARY TEMPERATURE ARRAY
       KIT=0
       IIT=0
       CM=30000.
  100 READ(5,10)IMAX,JMAX,DELTT,TOL,TIMX,TTT,KSKIP,NRUN,NNTP,LMAX,(SLTMP
     1(I),I=1,LMAX)
   10 FORMAT(2I5,F5.1,F5.2,F10.1,F5.0,I2,3I5/8F10.1)
C      IMAX=NUMBER OF COLUMNS (K)
C      JMAX=NUMBER OF ROWS (J)
C      DELTT=TIME STEP LENGTH,HOURS
C      TOL=TOLERANCE
C      TIMX=MAXIMUM TIME, HOURS
C      TTT=RESERVOIR TEMPERATURE, DEGREES FAHRENHEIT
C      KSKIP=OPTION TO INCREASE TIME STEP LENGTH DURING RUN
C      NRUN=RUN NUMBER
C      NNTP=NUMBER OF TIME STEPS
C      LMAX=NUMBER OF ISOTHERMS DESIRED
C      SLTMP(I)=TEMPERATURE OF DESIRED ISOTHERM
       READ(5,20)(Y(I),I=1,JMAX)
   20 FORMAT(16F5.0/3F5.0)
C      Y=DISTANCE INTO THE WALL, FEET
       READ(5,31)(X(J),J=1,IMAX)
   31 FORMAT(16F5.0/7F5.0)
C      X=THE VERTICAL DISTANCE PERPENDICULAR TO THE DIRECTION OF THE
C            OF THE FRACTURE
       WRITE(6,9)NRUN,TTT,TOL,KSKIP,(Y(I),I=1,19),TIMX,IMAX,JMAX,(X(I),I=THERMAB5
      11,23)
    9 FORMAT(1H1,5X,'INPUT DATA FOR THERMAB5 RUN NUMBER',2X,I5//5X,F10.1THERMAB5
      1,2X,'=RESERVOIR TEMPERATURE, DEGREES FAHRENHEIT'/5X,F10.4,2X,'=TOLTHERMAB5
      2ERANCE(PER CENT)'/10X,I5,2X,'=OPTION TO INCREASE TIME STEP LENGTH THERMAB5
      3DURING RUN'/20X,'(1) INDICATES TIME STEP LENGTH REMAINED CONSTANT THERMAB5
      4DURING RUN'/20X,'(2) INDICATES TIME STEP WAS INCREASED DURING RUN'THERMAB5
      5/5X,'THE DISTANCE INTO THE WALL, FEET, IS GIVEN BELOW'/3X,19F6.0//THERMAB5
      65X,F10.1,2X,'=MAXIMUM TIME, HOURS'/10X,I5,2X,'=IMAX'/10X,I5,2X,'=JTHERMAB5
      7MAX'//5X,'THE VERTICAL DISTANCE PERPENDICULAR TO THE DIRECTION OF THERMAB5
      8THE FRACTURE IS GIVEN BELOW'/3X,23F5.1)
       KWHO=1
       DD=1.0
       PP=0.                                                            THERMAB5
       TIME=0.0
       ATMP=0.0
       PPP=0.0                                                          THERMAB5
       NN=1
       DO 76 I=1,LMAX
       DO 76 K=1,IMAX
   76 DIST(I,K)=0.0
       READ(5,200)(TP(I),I=1,NNTP)
  200 FORMAT(8F10.1)
C      TP=SOURCE TEMPERATURES, DEGREES FAHRENHEIT
       DO 81 J=1,JMAX                                                   THERMAB5
       DO 81 K=1,IMAX                                                   THERMAB5
       TEMP(J,K)=TTT                                                    THERMAB5
       VTMP(J,K)=TTT                                                    THERMAB5
    81 TMPS(J,K)=TEMP(J,K)                                              THERMAB5
       DO 172 K=1,6                                                     THERMAB5
       TEMP(1,K)=TP(NN)
       VTMP(1,K)=TP(NN)
  172 TMPS(1,K)=TP(NN)
       CALL CALCY                                                       THERMAB5
C      CALCY IS A SUBROUTINE THAT CALCULATES VARIABLE Y SPACING VALUES
       CALL CALCX                                                       THERMAB5
C      CALCX IS A SUBROUTINE THAT CALCULATES Z SPACING VALUES
       CALL CAPAC                                                       THERMAB5
C      CAPAC IS A SUBROUTINE THAT CALCULATES TOTAL HEAT REQUIREMENT
C           VALUES.
       DO 232 J=1,JMAX                                                  THERMAB5
       DO 232 K=1,IMAX                                                  THERMAB5
  232 CAP(J,K)=CAPT(J,K)                                                THERMAB5
       CALL COND4                                                       THERMAB5
C      COND4 IS A SUBROUTINE THAT CALCULATES CONDUCTANCES
   82 GO TO(995,994),KSKIP
  994  IF(ATMP.NE.TIME) GO TO 995
       ATMP=ATMP+1.
       IF(TIME.GT.5.)GO TO 318
       DELTT=1.
       GO TO 995
  318 IF(DELTT.GE.15.)GO TO 993
       UU=DD-5.
       TME=5.*((1.2)**UU)
       DELTT=TME-TIME
       IF(DELTT.LT.15.)GO TO 995
       DELTT=15.
  993 IF(TIME.GT.1000.)GO TO 996
       DELTT=15.0
       GO TO 995
  996 IF(TIME.GT.3000.)GO TO 997
       DELTT=30.0
       GO TO 995
  997 IF(TIME.GT.10000.)GO TO 998
       DELTT=50.0
       GO TO 995
  998 DELTT=100.
C      KWHO IS THE SWITCH TO DETERMINE DIRECTION OF ADIP SWEEP(HORIZONTAL
C           OR VERTICAL)
  995 GO TO(991,53),KWHO
  991 CALL COND1                                                        THERMAB5
C      COND1 IS A SUBROUTINE THAT SOLVES ADIP IN THE Z DIRECTION
       IIT=IIT+1
       KIT=KIT+1
       IF(IIT.EQ.5)GO TO 296
       IF(KIT.GT.15)GO TO 222
       IF(DENOM.EQ.0.0) GO TO 69                                        THERMAB5
       KK=1                                                             THERMAB5
       DO 84 N=1,JMAX                                                   THERMAB5
       DO 84 M=1,IMAX                                                   THERMAB5
       IF(ABS((VTMP(N,M)-TMPS(N,M))/TMPS(N,M)).GT.TOL)GO TO 86          THERMAB5
   84 CONTINUE                                                          THERMAB5
       GO TO 67                                                         THERMAB5
   86 DO 88 J=1,JMAX                                                    THERMAB5
       DO 88 K=1,IMAX                                                   THERMAB5
   88 VTMP(J,K)=TMPS(J,K)                                               THERMAB5
       KWHO=2
       GO TO 82
  296 DO 297 K=1,IMAX
       DO 297 J=1,JMAX
       TMPS(J,K)=TEMP(J,K)
  297 VTMP(J,K)=TEMP(J,K)
       IIT=0
       TOL=TOL+.01
       CM=CM+5000.
       GO TO 82
   53 CALL COND2                                                        THERMAB5
C      COND2 IS A SUBROUTINE THAT SOVLES ADIP IN THE Y DIRECTION
       IF(DENOM.EQ.0.0) GO TO 69                                        THERMAB5
       KK=2                                                             THERMAB5
       DO 64 N=1,JMAX                                                   THERMAB5
       DO 64 M=1,IMAX                                                   THERMAB5
       IF(ABS((VTMP(N,M)-TMPS(N,M))/TMPS(N,M)).GT.TOL) GO TO 65         THERMAB5
   64 CONTINUE                                                          THERMAB5
       GO TO 67                                                         THERMAB5
   65 DO 83 J=1,JMAX                                                    THERMAB5
       DO 83 K=1,IMAX                                                   THERMAB5
   83 VTMP(J,K)=TMPS(J,K)                                               THERMAB5
       KWHO=1
       GO TO 82                                                         THERMAB5
   67 DO 66 J=1,JMAX                                                    THERMAB5
       DO 66 K=1,IMAX                                                   THERMAB5
       CAP(J,K)=CAPT(J,K)
   66 TEMP(J,K)=TMPS(J,K)                                               THERMAB5
       PPP=PPP+1.0
       IF(PPP.NE.10.) GO TO 741                                         THERMAB5
       PPP=0.                                                           THERMAB5
```

```
      JBMAX=JMAX-2                                                    THERMAB5
      IBMAX=IMAX-2                                                    THERMAB5
      FIMAX=IMAX                                                      THERMAB5
C     THIS ACCOUNTS FOR THE HEAT STORED IN ROCK                         22241
      DO 40 J=1,JMAX                                                  22241
      SUM=0.                                                          22241
      DO 80 K=1,IBMAX,2                                               22241
      E(K)=X(K+1)-X(K)
      EA(K)=(X(K+2)-X(K+1))/E(K)
   80 SUM=SUM+(E(K)*(EA(K)+1.)/(6.*EA(K)))*((2.*EA(K)-1.)*CAP(J+2,K)+(EA
     1(K)+1.)**2*CAP(J+1,K)+EA(K)*(2.-EA(K))*CAP(J,K))
   40 AREA(J)=SUM
      BB2=0.                                                          22241
      DO 90 J=1,JBMAX,2                                               22241
      H(J)=Y(J+1)-Y(J)                                                22241
      HA(J)=(Y(J+2)-Y(J+1))/H(J)                                      22241
   90 BB2=BB2+(H(J)*(HA(J)+1.)/(6.*HA(J)))*((2.*HA(J)-1.)*AREA(J+2)+(HA( 22241
     1J)+1.)**2*AREA(J+1)+HA(J)*(2.-HA(J))*AREA(J))                   22241
      HTORG=Y(JMAX)*X(IMAX)*3149.                                     THERMAB5
C     HTORG=INITIAL HEAT CONTENT OF ROCK AT 100 DEGREES FAHRENHEIT
      HEAT=BB2-HTORG                                                  THERMAB5
  741 TIME=TIME+DELTT                                                 THERMAB5
      IF(TIME.GT.3000.)GO TO 56
      TOL=.05
      GO TO 55
   56 IF(TIME.GT.4000.)GO TO 57
      TOL=.04
      GO TO 55
   57 IF(TIME.GT.5000.)GO TO 58
      TOL=.03
      GO TO 55
   58 IF(TIME.GT.CM)GO TO 59
      TOL=.02
      GO TO 55
   59 TOL=.01
   55 ATMP=TIME
      DD=DD+1.
      PP=PP+1.0
      NN=NN+1
      IF(PP.NE.10.) GO TO 891                                         THERMAB5
      PP=0.0                                                          THERMAB5
      WRITE(6,33)                                                     THERMAB5
   33 FORMAT(1H1,2X2HJ=5X2H 15X2H 25X2H 35X2H 45X2H 55X2H 65X2H 75X2H 85  22241
     1X2H 95X2H105X2H115X2H125X2H135X2H145X2H155X2H165X2H17)
      DO 36 K=1,IMAX
   36 WRITE(6,37)K,(TEMP(J,K),J=1,17)
   37 FORMAT(I4,17F7.1)
      WRITE(6,210)
  210 FORMAT(1H1,2X,'J=   18',5X,'19')
      DO 211 K=1,IMAX
  211 WRITE(6,212)K,(TEMP(J,K),J=18,19)
  212 FORMAT(I4,2F7.1)
  887 WRITE(6,30)TIME,HEAT,HTORG,DELTT                                THERMAB5
   30 FORMAT(F10.1,2X,'HOURS',/E18.8,2X,'BTU OF HEAT STORED IN RESERVOIR'THERMAB5
     1/E18.8,2X,'HTORG'/F10.1,2X,'TIME INCREMENT, HOURS')             THERMAB5
      CALL CONTR
C     CONTR IS A SUBROUTINE THAT CALCULATES THE POSITION OF SELECTED
C           ISOTHERMS
  891 KIT=0
      IIT=0
      IF(NN.GT.NNTP)GO TO 100
C     BELOW A NEW SOURCE TEMPERATURE IS CALLED IN
      DO 173 K=1,6                                                    THERMAB5
      VTMP(1,K)=TP(NN)
  173 TMPS(1,K)=TP(NN)
      GO TO (86,65),KK
  222 WRITE(6,223)DD,KIT
  223 FORMAT(3X,F10.0,I5)
   69 STOP                                                            THERMAB5
      END                                                             THERMAB5
      SUBROUTINE CALCY                                                  22241
      DIMENSION TMPS(19,23),TEMP(19,23),W(19,23),G(19,23),Y(19),YPLUS(19THERMAB5
     1),YNEG(19),YMID(19),TTMPS(19,23),AREA(19),H(19),HA(19),VTMP(19,23)THERMAB5
     2,CAPT(19,23),CAP(19,23),X(23),XPLUS(23),XNEG(23),XMID(23),COND(19,THERMAB5
     323),TP(1350),E(23),EA(23),SLTMP(8),DIST(8,23)
      COMMON TMPS,TEMP,W,G,Y,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,      THERMAB5
     1DENOM,TTMPS,CAP,CAPT,X,XPLUS,XNEG,XMID,TP,NN,TIME,LMAX,SLTMP,DIST THERMAB5
      NMAX=JMAX-1                                                     22241
      DO 2 J=1,NMAX                                                   22241
    2 YPLUS(J)=Y(J+1)-Y(J)                                            22241
      DO 3 J=2,JMAX                                                   22241
    3 YNEG(J)=Y(J)-Y(J-1)                                             22241
      YNEG(1)=YPLUS(1)                                                22241
      YPLUS(JMAX)=YNEG(JMAX)                                          22241
      DO 4 J=1,JMAX                                                   22241
    4 YMID(J)=0.5*(YPLUS(J)+YNEG(J))                                  22241
      RETURN                                                          22241
      END                                                             22241
      SUBROUTINE CALCX
      DIMENSION TMPS(19,23),TEMP(19,23),W(19,23),G(19,23),Y(19),YPLUS(19THERMAB5
     1),YNEG(19),YMID(19),TTMPS(19,23),AREA(19),H(19),HA(19),VTMP(19,23)THERMAB5
     2,CAPT(19,23),CAP(19,23),X(23),XPLUS(23),XNEG(23),XMID(23),COND(19,THERMAB5
     323),TP(1350),E(23),EA(23),SLTMP(8),DIST(8,23)
      COMMON TMPS,TEMP,W,G,Y,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,      THERMAB5
     1DENOM,TTMPS,CAP,CAPT,X,XPLUS,XNEG,XMID,TP,NN,TIME,LMAX,SLTMP,DIST THERMAB5
      NMAX=IMAX-1                                                     THERMAB5
      DO 2 K=1,NMAX                                                   THERMAB5
    2 XPLUS(K)=X(K+1)-X(K)                                            THERMAB5
      DO 3 K=2,IMAX                                                   THERMAB5
    3 XNEG(K)=X(K)-X(K-1)                                             THERMAB5
      XNEG(1)=XPLUS(1)                                                THERMAB5
      XPLUS(IMAX)=XNEG(IMAX)                                          THERMAB5
      DO 4 K=1,IMAX                                                   THERMAB5
    4 XMID(K)=0.5*(XPLUS(K)+XNEG(K))                                  THERMAB5
      RETURN                                                          THERMAB5
      END                                                             THERMAB5
      SUBROUTINE COND1                                                  22241
      DIMENSION TMPS(19,23),TEMP(19,23),W(19,23),G(19,23),Y(19),YPLUS(19THERMAB5
     1),YNEG(19),YMID(19),TTMPS(19,23),AREA(19),H(19),HA(19),VTMP(19,23)THERMAB5
     2,CAPT(19,23),CAP(19,23),X(23),XPLUS(23),XNEG(23),XMID(23),COND(19,THERMAB5
     323),TP(1350),E(23),EA(23),SLTMP(8),DIST(8,23)
      COMMON TMPS,TEMP,W,G,Y,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,      THERMAB5
     1DENOM,TTMPS,CAP,CAPT,X,XPLUS,XNEG,XMID,TP,NN,TIME,LMAX,SLTMP,DIST THERMAB5
      NMAX=IMAX-1                                                     THERMAB5
      DO 1 J=1,JMAX                                                   THERMAB5
      DO 20 K=1,IMAX                                                  THERMAB5
      IF(J.GT.1)GO TO 9                                               THERMAB5
      IF(K.GT.6)GO TO 9                                               THERMAB5
      GO TO 20                                                        THERMAB5
    9 IF(J.EQ.1)GO TO 78                                              22242
   51 IF(J.EQ.JMAX)GO TO 70                                           22242
      IF(K.EQ.1)GO TO 79                                              22242
      IF(K.EQ.IMAX)GO TO 80                                           22242
      CXPUS=(COND(J,K)+COND(J,K+1))/2.                                22241
      CXMIN=(COND(J,K)+COND(J,K-1))/2.                                22241
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                22241
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                22241
      GO TO 77                                                        22241
   79 CXPUS=(COND(J,K)+COND(J,K+1))/2.                                22241
      CXMIN=CXPUS                                                     22241
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                22241
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                22241
      GO TO 77                                                        22241
   80 CXMIN=(COND(J,K)+COND(J,K-1))/2.                                22242
      CXPUS=CXMIN                                                     22241
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                22241
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                22241
      GO TO 77                                                        22241
   78 IF(K.EQ.1)GO TO 75                                              22242
      IF(K.EQ.IMAX)GO TO 76                                           22242
      CXPUS=(COND(J,K)+COND(J,K+1))/2.                                22242
      CXMIN=(COND(J,K)+COND(J,K-1))/2.                                22242
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                22242
      CYMIN=CYPUS                                                     22242
      GO TO 77                                                        22242
   70 IF(K.EQ.1)GO TO 72                                              22242
      IF(K.EQ.IMAX)GO TO 73                                           22242
      CXPUS=(COND(J,K)+COND(J,K+1))/2.                                22242
      CXMIN=(COND(J,K)+COND(J,K-1))/2.                                22242
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                22242
      CYPUS=CYMIN                                                     22242
      GO TO 77                                                        22242
   72 CXPUS=(COND(J,K)+COND(J,K+1))/2.                                22242
      CXMIN=CXPUS                                                     22242
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                22242
      CYPUS=CYMIN                                                     22242
      GO TO 77                                                        22242
   73 CXMIN=(COND(J,K)+COND(J,K-1))/2.                                22242
      CXPUS=CXMIN                                                     22242
```

```
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22242
      CYPUS=CYMIN                                                       22242
      GO TO 77                                                          22242
   75 CXPUS=(COND(J,K)+COND(J,K+1))/2.                                  22242
      CXMIN=CXPUS                                                       22242
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                  22242
      CYMIN=CYPUS                                                       22242
      GO TO 77                                                          22242
   76 CXMIN=(COND(J,K)+COND(J,K-1))/2.                                  22242
      CXPUS=CXMIN                                                       22242
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                  22242
      CYMIN=CYPUS                                                       22242
   77 CXMIN=CXMIN*.71428                                                
      CXPUS=CXPUS*.71428                                                
      B1=-(1.+(XNEG(K)*CXPUS)/(XPLUS(K)*CXMIN)+(XNEG(K)*XMID(K)*CAPT(J,K
     1))/(CXMIN*DELTT*TMPS(J,K)))                                       
      C1=(XNEG(K)*CXPUS)/(XPLUS(K)*CXMIN)                               THERMAB5
      A1=1.0                                                            
      D14=-(XMID(K)*XNEG(K)/(CXMIN*YMID(J)))                            THERMAB5
      IF(J.LT.JMAX)GO TO 22                                             22241
      D15=CYPUS*(TMPS(J-1,K)-TMPS(J,K))/YPLUS(J)                        THERMAB5
      GO TO 90                                                          22241
   22 D15=CYPUS*(TMPS(J+1,K)-TMPS(J,K))/YPLUS(J)                        THERMAB5
   90 IF(J.GT.1)GO TO 91                                                
      D16=CYMIN*(TMPS(J,K)-TMPS(J+1,K))/YNEG(J)                         THERMAB5
      GO TO 92                                                          
   91 D16=CYMIN*(TMPS(J,K)-TMPS(J-1,K))/YNEG(J)                         THERMAB5
   92 D17=(XNEG(K)*XMID(K)*CAP(J,K))/(DELTT*CXMIN)                      
      D1=D14*(D15-D16)-D17                                              22241
      IF(J.GT.1)GO TO 4                                                 THERMAB5
      IF(K.GT.7)GO TO 3                                                 THERMAB5
      W(1,7)=C1/B1                                                      THERMAB5
      G(1,7)=(D1-A1*TMPS(1,6))/B1                                       THERMAB5
      GO TO 20                                                          THERMAB5
    4 IF(K.NE.1)GO TO 3                                                 22241
      C1=C1+1.                                                          THERMAB5
      A1=0.0                                                            22241
      W(J,1)=C1/B1                                                      22241
      G(J,1)=D1/B1                                                      22241
      GO TO 20                                                          22241
    3 IF(K.NE.IMAX)GO TO 5                                              22241
      A1=1.+C1                                                          THERMAB5
      C1=0.                                                             THERMAB5
      GO TO 10                                                          22241
    5 A1=1.0                                                            22241
   10 DENOM=B1-A1*W(J,K-1)                                              22241
      IF(DENOM.NE.0.)GO TO 18                                           22241
      WRITE(6,50)                                                       22241
   50 FORMAT(18H ZERO DENOMINATOR.)                                     22241
      GO TO 800                                                         22241
   18 W(J,K)=C1/DENOM                                                   22241
      G(J,K)=(D1-A1*G(J,K-1))/DENOM                                     22241
   20 CONTINUE                                                          22241
      TTMPS(J,IMAX)=G(J,IMAX)                                           
      IF(J.NE.1)GO TO 31                                                
      DO 30 I=1,NMAX                                                    
      II=IMAX-I                                                         
   30 TTMPS(J,II)=G(J,II)-W(J,II)*TTMPS(J,II+1)                         
      GO TO 1                                                           
   31 DO 32 I=2,IMAX                                                    
      II=IMAX+1-I                                                       
   32 TTMPS(J,II)=G(J,II)-W(J,II)*TTMPS(J,II+1)                         
    1 CONTINUE                                                          22241
      DO 95 J=1,JMAX                                                    
      DO 95 K=I,IMAX                                                    
      TMPS(J,K)=TTMPS(J,K)                                              
   95 CONTINUE                                                          
      DO 172 K=1,6                                                      THERMAB5
  172 TMPS(1,K)=TP(NN)                                                  
      CALL CAPAC                                                        
      CALL COND4                                                        
  800 RETURN                                                            22241
      END                                                               22241
      SUBROUTINE COND2                                                  22241
      DIMENSION TMPS(19,23),TEMP(19,23),W(19,23),G(19,23),Y(19),YPLUS(19THERMAB5
     1),YNEG(19),YMID(19),TTMPS(19,23),AREA(19),H(19),HA(19),VTMP(19,23)THERMAB5
     2,CAPT(19,23),CAP(19,23),X(23),XPLUS(23),XNEG(23),XMID(23),COND(19,THERMAB5
     323),TP(1350),E(23),EA(23),SLTMP(8),DIST(8,23)                     
      COMMON TMPS,TEMP,W,G,Y,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,       THERMAB5
     1DENOM,TTMPS,CAP,CAPT,X,XPLUS,XNEG,XMID,TP,NN,TIME,LMAX,SLTMP,DIST THERMAB5
      NMAX=JMAX-2                                                       
      DO 1 K=1,IMAX                                                     
      DO 20 J=1,JMAX                                                    
      IF(J.GT.1)GO TO 9                                                 THERMAB5
      IF(K.GT.6)GO TO 9                                                 THERMAB5
      GO TO 20                                                          THERMAB5
    9 IF(J.EQ.1)GO TO 78                                                22242
   61 IF(J.EQ.JMAX)GO TO 70                                             22242
      IF(K.EQ.1)GO TO 79                                                22242
      IF(K.EQ.IMAX)GO TO 80                                             22242
      CXPUS=(COND(J,K)+COND(J,K+1))/2.                                  22241
      CXMIN=(COND(J,K)+COND(J,K-1))/2.                                  22241
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                  22241
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22241
      GO TO 77                                                          22241
   79 CXPUS=(COND(J,K)+COND(J,K+1))/2.                                  22242
      CXMIN=CXPUS                                                       22242
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                  22241
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22241
      GO TO 77                                                          22241
   80 CXMIN=(COND(J,K)+COND(J,K-1))/2.                                  22242
      CXPUS=CXMIN                                                       22242
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                  22241
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22242
      GO TO 77                                                          22241
   78 IF(K.EQ.1)GO TO 75                                                22242
      IF(K.EQ.IMAX)GO TO 76                                             22242
      CXPUS=(COND(J,K)+COND(J,K+1))/2.                                  22242
      CXMIN=(COND(J,K)+COND(J,K-1))/2.                                  22242
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                  22242
      CYMIN=CYPUS                                                       22242
      GO TO 77                                                          22242
   70 IF(K.EQ.1)GO TO 72                                                22242
      IF(K.EQ.IMAX)GO TO 73                                             22242
      CXPUS=(COND(J,K)+COND(J,K+1))/2.                                  22242
      CXMIN=(COND(J,K)+COND(J,K-1))/2.                                  22242
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22242
      CYPUS=CYMIN                                                       22242
      GO TO 77                                                          22242
   72 CXPUS=(COND(J,K)+COND(J,K+1))/2.                                  22242
      CXMIN=CXPUS                                                       22242
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22242
      CYPUS=CYMIN                                                       22242
      GO TO 77                                                          22242
   73 CXMIN=(COND(J,K)+COND(J,K-1))/2.                                  22242
      CXPUS=CXMIN                                                       22242
      CYMIN=(COND(J,K)+COND(J-1,K))/2.                                  22242
      CYPUS=CYMIN                                                       22242
      GO TO 77                                                          22242
   75 CXPUS=(COND(J,K)+COND(J,K+1))/2.                                  22242
      CXMIN=CXPUS                                                       22242
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                  22242
      CYMIN=CYPUS                                                       22242
      GO TO 77                                                          22242
   76 CXMIN=(COND(J,K)+COND(J,K-1))/2.                                  22242
      CXPUS=CXMIN                                                       22242
      CYPUS=(COND(J,K)+COND(J+1,K))/2.                                  22242
      CYMIN=CYPUS                                                       22242
   77 CXMIN=CXMIN*.71428                                                
      CXPUS=CXPUS*.71428                                                
      B1=-(1.+(CYPUS*YNEG(J))/(YPLUS(J)*CYMIN)+(YNEG(J)*YMID(J)*CAPT(J,KTHERMAB5
     1))/(DELTT*CYMIN*TMPS(J,K)))                                       
      A1=1.                                                             THERMAB5
      C1=(YNEG(J)*CYPUS)/(YPLUS(J)*CYMIN)                               THERMAB5
      D14= -(YMID(J)*YNEG(J)/(CYMIN*XMID(K)))                           
      IF(K.LT.IMAX) GO TO 24                                            
      D15= CXPUS*(TMPS(J,K-1)-TMPS(J,K))/XPLUS(K)                       THERMAB5
      GO TO 25                                                          
   24 D15= CXPUS*(TMPS(J,K+1)-TMPS(J,K))/XPLUS(K)                       THERMAB5
   25 IF(J.LT.2)GO TO 26                                                THERMAB5
      IF(K.GT.1)GO TO 26                                                THERMAB5
```

```
      D16=CXMIN*(TMPS(J,K)-TMPS(J,K+1))/XNEG(K)                          THERMAB5
      GO TO 27                                                           THERMAB5
   26 D16=CXMIN*(TMPS(J,K)-TMPS(J,K-1))/XNEG(K)                          THERMAB5
   27 D17=(CAP(J,K)*YMID(J)*YNEG(J))/(DELTT*CYMIN)                       THERMAB5
      D1=D14*(D15-D16)-D17                                                 22241
      IF(K.GT.6)GO TO 4                                                  THERMAB5
      IF(J.GT.2)GO TO 3                                                  THERMAB5
      W(2,K)=C1/B1                                                         22241
      G(2,K)=(D1-A1*TMPS(1,K))/B1                                          22241
      GO TO 20                                                             22241
    4 IF(J.GT.1)GO TO 3                                                  THERMAB5
      C1=C1+A1                                                           THERMAB5
      A1=0.0                                                             THERMAB5
      W(1,K)=C1/B1                                                       THERMAB5
      G(1,K)=D1/B1                                                       THERMAB5
      GO TO 20                                                           THERMAB5
    3 IF(J.NE.JMAX)GO TO 10                                                22241
      A1=A1+C1                                                             22241
      C1=0.0                                                               22241
   10 DENOM=B1-A1*W(J-1,K)                                                 22241
      IF(DENOM.NE.0.)GO TO 18                                              22241
      WRITE(6,50)                                                          22241
   50 FORMAT(18H ZERO DENOMINATOR.)                                        22241
   18 W(J,K)=C1/DENOM                                                      22241
      G(J,K)=(D1-A1*G(J-1,K))/DENOM                                        22241
   20 CONTINUE                                                             22241
      TTMPS(JMAX,K)=G(JMAX,K)
      IF(K.GT.6)GO TO 31
      DO 30 I=1,NMAX
      II=JMAX-I
   30 TTMPS(II,K)=G(II,K)-W(II,K)*TTMPS(II+1,K)
      GO TO 1
   31 DO 32 I=2,JMAX
      II=JMAX+1-I
   32 TTMPS(II,K)=G(II,K)-W(II,K)*TTMPS(II+1,K)
    1 CONTINUE                                                             22241
      DO 95 J=1,JMAX                                                     THERMAB5
      DO 95 K=1,IMAX                                                     THERMAB5
      TMPS(J,K)=TTMPS(J,K)
   95 CONTINUE
      DO 172 K=1,6                                                       THERMAB5
  172 TMPS(1,K)=TP(NN)
      CALL CAPAC
      CALL COND4
  800 RETURN                                                               22241
      END                                                                  22241
      SUBROUTINE CAPAC
      DIMENSION TMPS(19,23),TEMP(19,23),W(19,23),G(19,23),Y(19),YPLUS(19THERMAB5
     1),YNEG(19),YMID(19),TTMPS(19,23),AREA(19),H(19),HA(19),VTMP(19,23)THERMAB5
     2,CAPT(19,23),CAP(19,23),X(23),XPLUS(23),XNEG(23),XMID(23),COND(19,THERMAB5
     323),TP(1350),E(23),EA(23),SLTMP(8),DIST(8,23)
      COMMON TMPS,TEMP,W,G,Y,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,       THERMAB5
     1DENOM,TTMPS,CAP,CAPT,X,XPLUS,XNEG,XMID,TP,NN,TIME,LMAX,SLTMP,DIST THERMAB5
      DO 232 J=1,JMAX
      DO 232 K=1,IMAX
      IF(TIME.EQ.0.0)GO TO 9
      IF(TMPS(J,K).LT.100.01) GO TO 232
    9 IF(TMPS(J,K).GT.450.)GO TO 233
      CAPT(J,K)=(23.5+.307*(TMPS(J,K)-100.))*134.0
      GO TO 232
  233 IF(TMPS(J,K).GT.900.)GO TO 234
      CAPT(J,K)=(130.0+.344*(TMPS(J,K)-450.))*134.0
      GO TO 232
  234 IF(TMPS(J,K).GT.1100.)GO TO 235
      CAPT(J,K)=134.0*(285.0+.485*(TMPS(J,K)-900.))
      GO TO 232
  235 IF(TMPS(J,K).GT.1600.)GO TO 236
      CAPT(J,K)=134.0*(382.0+.6240*(TMPS(J,K)-1100.))
      GO TO 232
  236 CAPT(J,K)=134.0*(694.0+.247*(TMPS(J,K)-1600.))
  232 CONTINUE
      RETURN
      END
      SUBROUTINE COND4                                                     22242
      DIMENSION TMPS(19,23),TEMP(19,23),W(19,23),G(19,23),Y(19),YPLUS(19THERMAB5
     1),YNEG(19),YMID(19),TTMPS(19,23),AREA(19),H(19),HA(19),VTMP(19,23)THERMAB5
     2,CAPT(19,23),CAP(19,23),X(23),XPLUS(23),XNEG(23),XMID(23),COND(19,THERMAB5
     323),TP(1350),E(23),EA(23),SLTMP(8),DIST(8,23)
      COMMON TMPS,TEMP,W,G,Y,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,       THERMAB5
     1DENOM,TTMPS,CAP,CAPT,X,XPLUS,XNEG,XMID,TP,NN,TIME,LMAX,SLTMP,DIST THERMAB5
      DO 5 J=1,JMAX                                                        22242
      DO 5 K=1,IMAX                                                        22242
      IF(TMPS(J,K).GE.200.)GO TO 1
      COND(J,K)=0.9
      GO TO 5
    1 IF(TMPS(J,K).GT.1200.)GO TO 7
      TSQ=TMPS(J,K,*TMPS(J,K)
      TCUB=TMPS(J,K)*TSQ
      COND(J,K)=.12561573E+1-.18377799E-2*TMPS(J,K)+.10902421E-5*TSQ-.15
     1084869E-9*TCUB
      GO TO 5
    7 COND(J,K)=0.36
    5 CONTINUE
      RETURN
      END
      SUBROUTINE CONTR
      DIMENSION TMPS(19,23),TEMP(19,23),W(19,23),G(19,23),Y(19),YPLUS(19THERMAB5
     1),YNEG(19),YMID(19),TTMPS(19,23),AREA(19),H(19),HA(19),VTMP(19,23)THERMAB5
     2,CAPT(19,23),CAP(19,23),X(23),XPLUS(23),XNEG(23),XMID(23),COND(19,THERMAB5
     323),TP(1350),E(23),EA(23),SLTMP(8),DIST(8,23)
      COMMON TMPS,TEMP,W,G,Y,YPLUS,YNEG,YMID,JMAX,IMAX,COND,DELTT,       THERMAB5
     1DENOM,TTMPS,CAP,CAPT,X,XPLUS,XNEG,XMID,TP,NN,TIME,LMAX,SLTMP,DIST THERMAB5
      DO 263 I=1,LMAX
      DO 260 K=1,IMAX
      DO 261 J=1,JMAX
      IF(TEMP(J,K).LT.SLTMP(I))GO TO 262
  261 CONTINUE
      J=JMAX
  262 IF(J.EQ.1)GO TO 260
      DIST(I,K)=Y(J-1)+((TEMP(J-1,K)-SLTMP(I))/(TEMP(J-1,K)-TEMP(J,K)))*
     1(Y(J)-Y(J-1))
  260 CONTINUE
  263 CONTINUE
      WRITE(6,264)(SLTMP(I),I=1,LMAX)
  264 FORMAT(1H1,6X,'DISTANCE',7X,'ISOTHERM  ISOTHERM  ISOTHERM  ISOTHERTHERMAB6
     1M  ISOTHERM  ISOTHERM  ISOTHERM  ISOTHERM'/8X,'DOWN',9X,'DEGREES FTHERMAB6
     2 DEGREES F DEGREES F DEGREES F DEGREES F DEGREES F DEGREES F DEGRETHERMAB6
     3EES F'/7X,'FRACTURE',4X,8F10.1/9X,'FEET'/20X,'FEET INTO FEET INTO THERMAB6
     4FEET INTO FEET INTO FEET INTO FEET INTO FEET INTO FEET INTO'/22X,'THERMAB6
     5WALL',6X,'WALL',6X,'WALL',6X,'WALL',6X,'WALL',6X,'WALL',6X,'WALL',THERMAB6
     66X,'WALL')
      DO 266 K=1,IMAX
  266 WRITE(6,267)X(K),(DIST(I,K),I=1,LMAX)
  267 FORMAT(7X,F6.1,4X,8F10.2)
      RETURN
      END
```

VITA

Allen Lawrence Barnes

Candidate for the Degree of

Doctor of Philosophy

Thesis: A FEASIBILITY STUDY OF AN IN SITU RETORTING PROCESS FOR OIL SHALE

Major Field: Mechanical Engineering

Biographical:

Personal Data: Born in Wooster, Ohio, September 28, 1932, the son of Earl E. and Florence Belle Barnes.

Education: Graduated from Wooster Township grade school, Wayne County, Ohio, in 1946; attended Miami University, Oxford, Ohio, from 1950 to 1952; received Bachelor of Petroleum Engineering from The Ohio State University, in May, 1955; received Master of Science degree from The Pennsylvania State University in January, 1960; took graduate work at The University of Tulsa during academic years, 1960-1961, 1965 and 1966-1967; completed requirements for the Doctor of Philosophy degree in September, 1967.

Professional Experience: Teaching assistant in the Department of Mining and Petroleum Engineering at The Ohio State University during academic year of 1954-1955; employed as petroleum engineer with Magnolia Petroleum Company from 1955 to 1958; served in the United States Navy for six months during 1957 with The Office of Naval Petroleum Reserves, Washington, D. C.; employed as a research engineer with the Research Division, Sinclair Oil & Gas Company from 1959 to 1965 and 1966 to 1967; served as teaching assistant at Oklahoma State University during academic year of 1965-1966.

Professional Organizations: Member of AIME and API.