



Mathematical models for inorganic processes found in a zinc plant
by Rodney Allen James

A thesis submitted to the Graduate Faculty in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY in Chemical Engineering
Montana State University
© Copyright by Rodney Allen James (1973)

Abstract:

Man has modeled things since the beginning of civilization. The invention of the computer allowed him to solve mathematical models of many complex processes. Mathematical models of the processes found in the petroleum industry are quite common. Mathematical models of inorganic processes are less common. This research project was an attempt to develop mathematical models of some of the processes found in the inorganic processing industry. Using processes found in a zinc producing facility it was possible to develop mathematical models that provide a basis for future work. Mathematical models developed include: .

1. Particle size distribution equation.
2. Wedge roaster model.
3. Electrostatic precipitator model.
4. Ballmill model.
5. Mixer model.
6. Leach tank model.

Calculations were made using subroutines, written to fit a modified CHESS system, prepared from the mathematical models developed. The results of the calculations agreed with data supplied by the Anaconda Company zinc operations at Great Falls, Montana.

MATHEMATICAL MODELS FOR INORGANIC PROCESSES FOUND IN A ZINC PLANT

by

RODNEY ALLEN JAMES

A thesis submitted to the Graduate Faculty in partial
fulfillment of the requirements for the degree

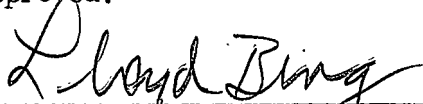
of

DOCTOR OF PHILOSOPHY

in

Chemical Engineering

Approved:



Head, Major Department



Chairman, Examining Committee



Graduate Dean

MONTANA STATE UNIVERSITY
Bozeman, Montana

June, 1973

Acknowledgement

The author would like to thank the people of the United States for the NDEA fellowship that made it possible to pursue this degree. He would like to thank the staff of the Chemical Engineering department for their support especially Dr. Michael Schaer, his adviser, and Dr. Robert L. Nickelson. He would like to thank his wife, Mary Lee Walton James, for her support and for typing this thesis.

Table of Contents

Vita	ii
Acknowledgement	iii
Table of Contents	iv
List of Tables	vi
List of Figures	vii
Abstract	x
Introduction	1
Objectives	5
Description of the Modified CHESS System	7
Executive Programs	19
Supporting Programs	23
Computing Programs	41
Computing Hardware and Software	42
Results and Discussion	46
Recommendations	62
Summary	63
Appendix	65
Detailed Description of Supporting Programs	66
ROASTER	67
SCREEN	76
GRINDER or BALLMILL	79

ELECTROSTATIC PRECIPITATOR	81
ACID and FERRIC SULFATE ADDITION	85
NEUTRAL and ACID LEACHS	87
CLASSIFIER	96
THICKENER	98
FILTER	100
MIXER	101
ELECTROLYTIC PLANT	103
DATA INPUT FOR PLANT SIMULATION	104
Executive Program Printouts	116
Supporting Subroutine Printouts	154
Computing Subroutine Printouts	190
Sample Problem Printouts	197
Bibliography	237

)

List of Tables

Table 1	Calculated Results vs Plant Data
---------	----------------------------------

53

List of Figures

Figure 1	Comparison of Particle Size Distribution Equation with Actual Data - Concentrates	13
Figure 2	Comparison of Particle Size Distribution Equation with Actual Data - Calcine	14
Figure 3	Comparison of Particle Size Distribution Equation with Actual Data - Crushed Screen Oversize	15
Figure 4	Specifying Recycle Loops	18
Figure 5	Particle Model for Roaster	27
Figure 6	Overlay Scheme	45
Figure 7	Roasting Plant Layout	47
Figure 8	Leach Plant Layout	48
Figure 9	Equipment Node Sequence for Roasting Plant Simulation	49
Figure 10	Equipment Node Sequence for Leach Plant Simulation	50
Figure 11	Zinc Production vs Plant Input	54
Figure 12	Zinc Production vs Roasting Constant	56
Figure 13	Zinc Production vs Number of Leach Tanks Removed from Neutral Leach Circuit	58
Figure 14	ZnO Leached vs Number of Leach Tanks Removed from Neutral Leach Circuit	59
Figure 15	Ferric Sulfate Concentration Leaving Neutral Leach vs Number of Leach Tanks Removed from Neutral Leach Circuit	61
Figure 16	MAIN CALLING PROGRAM	117

Figure 17	SUBROUTINE DREAD	120
Figure 18	SUBROUTINE CLEAN	128
Figure 19	SUBROUTINE ZERO	130
Figure 20	SUBROUTINE DCHECK	131
Figure 21	SUBROUTINE DPRINT	132
Figure 22	SUBROUTINE SUBSET	134
Figure 23	SUBROUTINE SCAN	137
Figure 24	SUBROUTINE RECYCLE	138
Figure 25	SUBROUTINE EQUIP	141
Figure 26	SUBROUTINE REQUIP	143
Figure 27	SUBROUTINE EQCALL	144
Figure 28	SUBROUTINE TEST	146
Figure 29	SUBROUTINE TRANSF	150
Figure 30	SUBROUTINE TPRINT	151
Figure 31	SUBROUTINE ROST	155
Figure 32	SUBROUTINE SCRIN	159
Figure 33	SUBROUTINE GRND	162
Figure 34	SUBROUTINE PREP	164
Figure 35	SUBROUTINE AMIX	167
Figure 36	SUBROUTINE LECH	170
Figure 37	SUBROUTINE CIAS	177
Figure 38	SUBROUTINE SETL	180

Figure 39	SUBROUTINE FLTR	182
Figure 40	SUBROUTINE MIXR	184
Figure 41	SUBROUTINE ELET	188
Figure 42	SUBROUTINE CONVT	191
Figure 43	SUBROUTINE LSC	192
Figure 44	SUBROUTINE SIME	195
Figure 45	SUBROUTINE TGS	196
Figure 46	Zinc Plant Simulation Roasting Section Data	198
Figure 47	Zinc Plant Simulation Leaching Section Data	200
Figure 48	Zinc Plant Simulation Roasting Section Results	204
Figure 49	Zinc Plant Simulation Leaching Section Results	212

Abstract

Man has modeled things since the beginning of civilization. The invention of the computer allowed him to solve mathematical models of many complex processes. Mathematical models of the processes found in the petroleum industry are quite common. Mathematical models of inorganic processes are less common. This research project was an attempt to develop mathematical models of some of the processes found in the inorganic processing industry. Using processes found in a zinc producing facility it was possible to develop mathematical models that provide a basis for future work. Mathematical models developed include:

1. Particle size distribution equation.
2. Wedge roaster model.
3. Electrostatic precipitator model.
4. Ballmill model.
5. Mixer model.
6. Leach tank model.

Calculations were made using subroutines, written to fit a modified CHESS system, prepared from the mathematical models developed. The results of the calculations agreed with data supplied by the Anaconda Company zinc operations at Great Falls, Montana.

Introduction

Simulation of actual events with models is not a new development. Models have been used since the early beginnings of mankind. Originally all models were descriptive in nature, but as mathematics developed, it became possible to formulate predictive as well as descriptive models. Man's need to know why something happened brought about a third type of model called an explanatory model.

The modern analysis of process problems usually involves some form of modeling. The model can be physical, like a pilot plant, or it can be analytical, like a mathematical model. Since a model is a partial representation of the real system, the quality of that representation is determined by comparing the behavior of the model to that of the real system, under various constraints. These constraints are called the initial and boundary conditions. The initial conditions describe the system at the starting time of the study, while the boundary conditions describe the environmental restrictions on the operation of the system.

Before the advent of the computer there were two bottlenecks in the procedure for obtaining quantitative solutions from an analytical approach. These were, analysis of the problem to get the mathematical model, and then solution of the equations resulting from the analysis. The bottleneck of computation was generally

impossible, so that efforts at mathematical modeling were of no practical value since the equations could not be solved. As a result, the effective way to solve technical problems was by laboratory or pilot-plant experimental procedures. The disciplines of the analytical approach were regarded as purely academic and consequently were not practiced as a way of life in industry.

In the last 15 to 20 years electronic digital computers have been used to obtain solutions to the equations developed from the analytical approach. At first the applications were restricted to small, simple models because of the limitations of early computer technology. As computers increased in flexibility of operation, speed, and memory capacity, it became feasible to apply them more frequently to models of larger scope. The development of software languages have made computers accessible to the average engineer if he can get past the first bottleneck of analysis. The bottleneck of computation does not exist today for most practical purposes.

There are various mathematical models for the same system, each one suited to solve a particular problem associated with the system. The two broad classifications constitute steady-state and dynamic models, and in either type the degree of detail depends on the problem to be solved and the amount of basic data available. A very precise description of a chemical process system will often lead to

a large set of unwieldy equations. Although they can be solved, it is advisable for the analyst to use engineering judgement to reduce the equations to a less complex set that for all practical purposes will yield an engineering solution within the accuracy of the available basic data.

The mathematical models used to simulate chemical processes use the laws governing the conservation of matter and energy to describe the physical or chemical transformations that occur in actual processing units. When the rate of transformation is important, the laws governing the transport of matter, energy, and momentum are applied.

The chemical engineer can use mathematical models of chemical processes to gain information that he would otherwise have to obtain by experimental methods. If for example, he desired to scale up a process from bench scale to a producing plant, he could simulate the scale up with his mathematical model, predict the results and thereby eliminate the need for an intermediate pilot plant to test the results of such a scale up. Also if he wished to make a change in an existing plant, he could simulate the change and determine the results of such a change without actually making that change.

Chemical Engineering Simulation System, CHESS, (13) is a digital computer program developed at the University of Houston. This

program is used to simulate the steady-state operation of chemical plants processing organic chemicals. CHESS can be used to simulate any plant processing petroleum derivatives if the user can specify the equipment sequence in the plant and the input stream data.

In CHESS each piece of processing equipment found in an actual plant is modeled mathematically in a subroutine simulating that piece of equipment. The subroutines are called by the CHESS executive programs at the proper position in the process matrix specified by the user. This allows the user to simulate similar pieces of equipment with the same subroutine. For example, if two mixers are used in a process, they can both be simulated by the same mixer subroutine with different input streams.

A simulation system similar to CHESS would be useful in the inorganic processing field. The pieces of equipment used in inorganic processing are generally large and expensive, and a great deal of money could be saved if proposed changes in plant design could be tested mathematically before the actual changes are made.

Objectives

In the petroleum products industry many of the process streams are homogeneous in nature. Most mathematical modeling done for the petroleum products industry can therefore be based on the assumption that all of the processes occur in a single homogeneous phase. The mathematical models in CHESS used to simulate petrochemical processes required that all streams be single phased and homogeneous.

In the inorganic process industry most reactions occur as the result of an interaction between two or more phases, and the phases are not usually homogeneous. In a multiphase system mass transfer between phases is often the rate controlling step. In the case of a solid phase with a distribution of sizes, the size of each particle may have an effect on the chemical and physical processes.

This study was undertaken to develop mathematical models that could be used to simulate inorganic processes. The development of techniques for simulating the physical processes involved in the handling of streams containing solids of various particle sizes was undertaken. Simulations of multiphase chemical reactions were developed. In particular, the chemical and physical processes found in a zinc producing facility were simulated, and computer subroutines were written to fit CHESS using the mathematical models developed. The results obtained were then checked against data supplied by the

Anaconda Company on their zinc plant at Great Falls, Montana.

Description of the Modified CHESS System

CHESS is a simulation program developed for the petrochemical industry. It was designed to allow the user to simulate any petrochemical plant by specifying the process matrix and the input stream data. The system allows the user to simulate 50 processing units with 100 process streams.

The CHESS system used NAMELIST input/output to allow the user the flexibility required to simulate several different systems with the same program. NAMELIST input/output allows the user to input and output numeric data without FORMAT statements. Each input record specifies exactly which variable is being input, rather than requiring the input data to interface with an internal input list. This allows the user to input information without knowing in advance which items are going to be processed. (26)

NAMELIST input statements have the form:

READ (u,x)

where

u is the device number that causes the proper computer input device to be called.

x is a NAMELIST block name

The NAMELIST block name is given to a specific list of variables or array names in a NAMELIST statement. (26) An example of a NAMELIST

statement is:

NAMELIST/PMLIST/KPM1, KPM2,..... where

PMLIST is the block name KPM1, KPM2 are variables.

The CHESS system uses NAMELIST to input data in vector notation. Instead of being single variables KPM1, and KPM2 are vector names and have the form:

KPM1 = x1, x2, x3, x4,.....

where x1, x2, x3, and x4 are elements in the vector. The same values could be expressed as KPM1(1) = x1, KPM1(2) = x2; but the vector manner gives a more compact data input and the computer can divide the vector into its separate elements.

All of the data used by the CHESS system is input as vectors in NAMELIST notation. Each separate vector is discussed as its use is developed and each vector is also discussed in detail in the appendix.

The CHESS system uses KPM vectors to specify the process layout of the plant being modeled. ⁽¹²⁾ There is room in the system for 50 pieces of processing units so there are 50 KPM vectors of the form

KPM1 = 1, 'ROST', 'R-1', 1, -2, -3, 4*0, ..

This vector says that the first piece of equipment in the process is a roaster that the user has designated R-1, the input stream is 1 and the output streams are 2 and 3. ROST is the four letter name given to the subroutine simulating a roaster and is the signal that is used by the CHESS system to call the roaster subroutine at the proper time.

R-1 is a user identification name for the roaster and can be any four character name because it doesn't affect the problem solution.

Positive numbers after the user identification name are the stream numbers of the input streams, and the negative numbers identify the output streams. The number of input and output streams together can not exceed 7 and when the number is less than 7 the remaining vector must be filled out with zeros (In this case four zeros are required).

One of the output streams from the first processing unit might be the input stream for the second processing unit. If for example stream 3 goes to a screen the KPM vector describing the screen would be:

KPM2 = 2, 'SCRN', 'S-1', 3, -6, -7, 4*0, .

This vector says that piece of equipment number 2 is a screen designated S-1 with stream number 3 as input and streams numbers 6 and 7 as outputs.

The user would continue to describe the process matrix with KPM vectors until the system was completely described. The remaining KPM vectors up to KPM 50 can be left out of the input data with the NAMELIST input system.

PMLIST is the NAMELIST block containing the KPM vectors.

PMLIST also contains COMPNT, and NOCOMP. NOCOMP is the total number of components and COMPNT is a vector listing of the component numbers

used. An example of each is:

COMPNT = 2, 6, 7, 9, 11, 15*0,

NOCOMP = 5

The zeros are used to fill out the COMPNT vector which can be used to specify the component numbers of up to 20 components.

The CHESS system can handle up to 100 process streams. The properties and compositions of these streams are contained in two stream vectors. In the original form of CHESS the intensive variables were specified in SIN vectors and the extensive variables in SEX vectors. In the modified version of CHESS presented in this paper, the stream properties were not divided into intensive and extensive variables.

The intensive variable vector has room for 10 variables and contains a stream flag and the roasting constant of the zinc concentrates entering the process. In terms of the NAMELIST convention:

SINJ = J, F, K, 7*0.,

where

J = Stream number

F = Stream flag; 0. = intermediate stream, 1. = feed stream,

2. = Product stream.

K = Roasting constant of zinc concentrates being processed

The seven zeros at the end of the vector provide room for additional variables that might be required as the system is developed and expanded further.

The SIN vector must be input for process input and product streams only. The input and product streams must be flagged by the user because the CHESS system automatically flags all streams not specified as input or product streams as intermediate streams in the process. All streams used in a simulation but not flagged as input or product streams are listed in vector SNAME.

The vectors SIN1 through SIN100 together with SNAME make up the NAMELIST block SINLST.

A SEX vector is 23 units long and contains constants describing the particle size distribution of the solids, an effective holding time, the average conversion, the total weight of the stream, the weight fraction of each component in the stream, and the standard cubic feet of gas in the stream. The form of the SEX vector is:

SEXJ = J, A, B, C, EHT, AVC, 5*0., TW, WF....., WFN, CFG, M*0

where

J = Stream number

A, B, C = Constants used to describe the particle size distribution of the solids; B and C are unitless while the units on A are inches.

EHT = Effective holding time required to get the average conversion given for the stream.

AVC = Average conversion of ZnS to ZnO for the stream.

The five zeros provide room for additional variables that are required as the system is expanded.

TW = Total weight of the stream - Tons/Day

WF.....WF_n = weight fraction of each component in the stream.

N = Total number of components, NOCOMP, minus 1; the last component is always the gas phase and is not figured in the total weight and therefore does not have a weight fraction.

CFG = Standard cubic feet of gas in the stream.

M*0. = The remaining vector up to 23 elements is filled out with zeros.

The constants A, B, and C are used in the equation

$$r = AB \frac{x}{(1-Cx)}$$

This equation gives the radius of a particle as a function of the weight fraction solids, x, with radii less than r. The equation gives the best fit to experimental data of several equations tested. Figures 1, 2, and 3 show the closeness of fit of this equation to experimental data.

The constants A, B, and C are fit to the experimental data using the method of "least squares". Assuming that C is known.

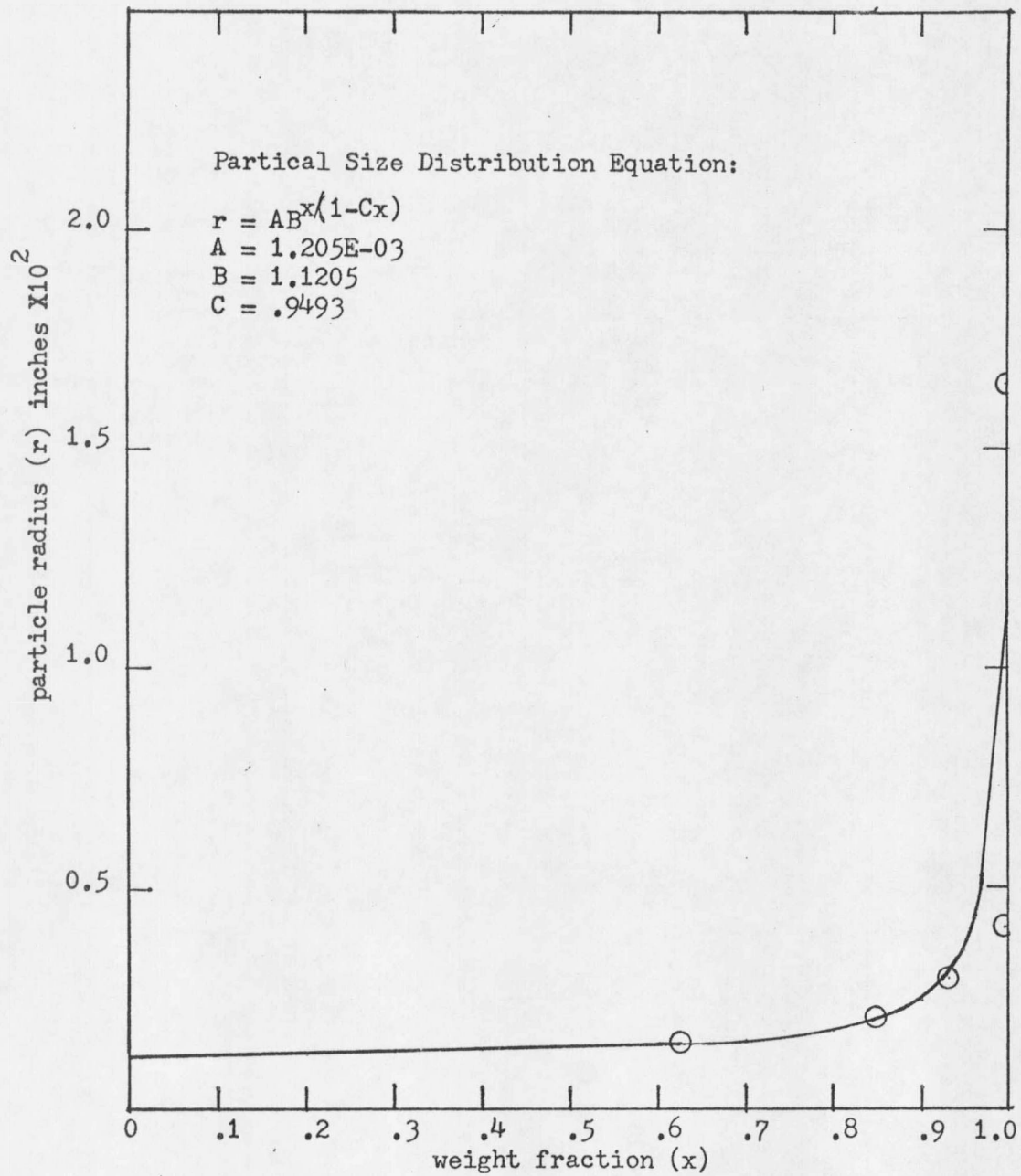


Figure 1
Comparison of Particle Size Distribution
Equation with Actual Data - Concentrates

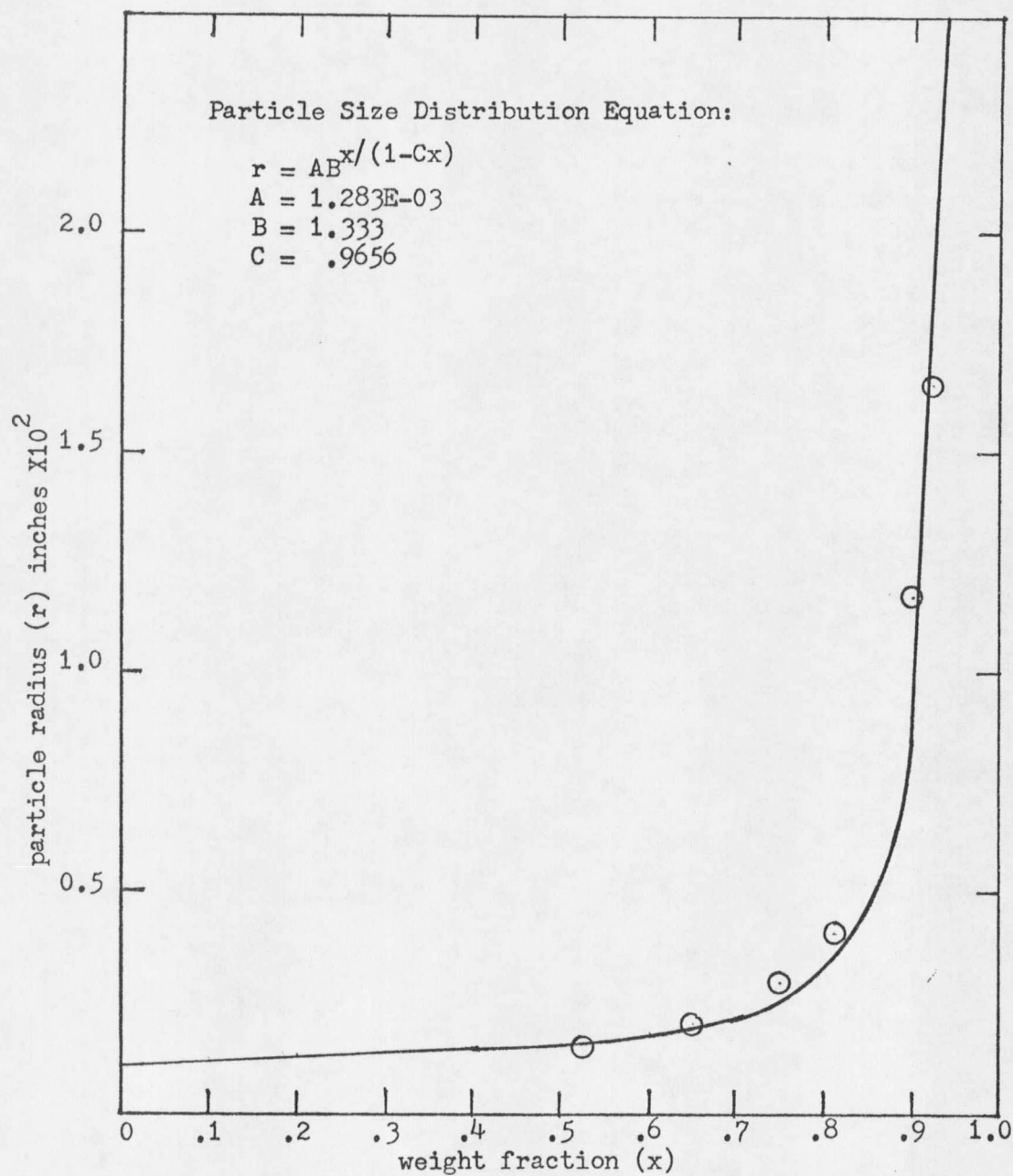
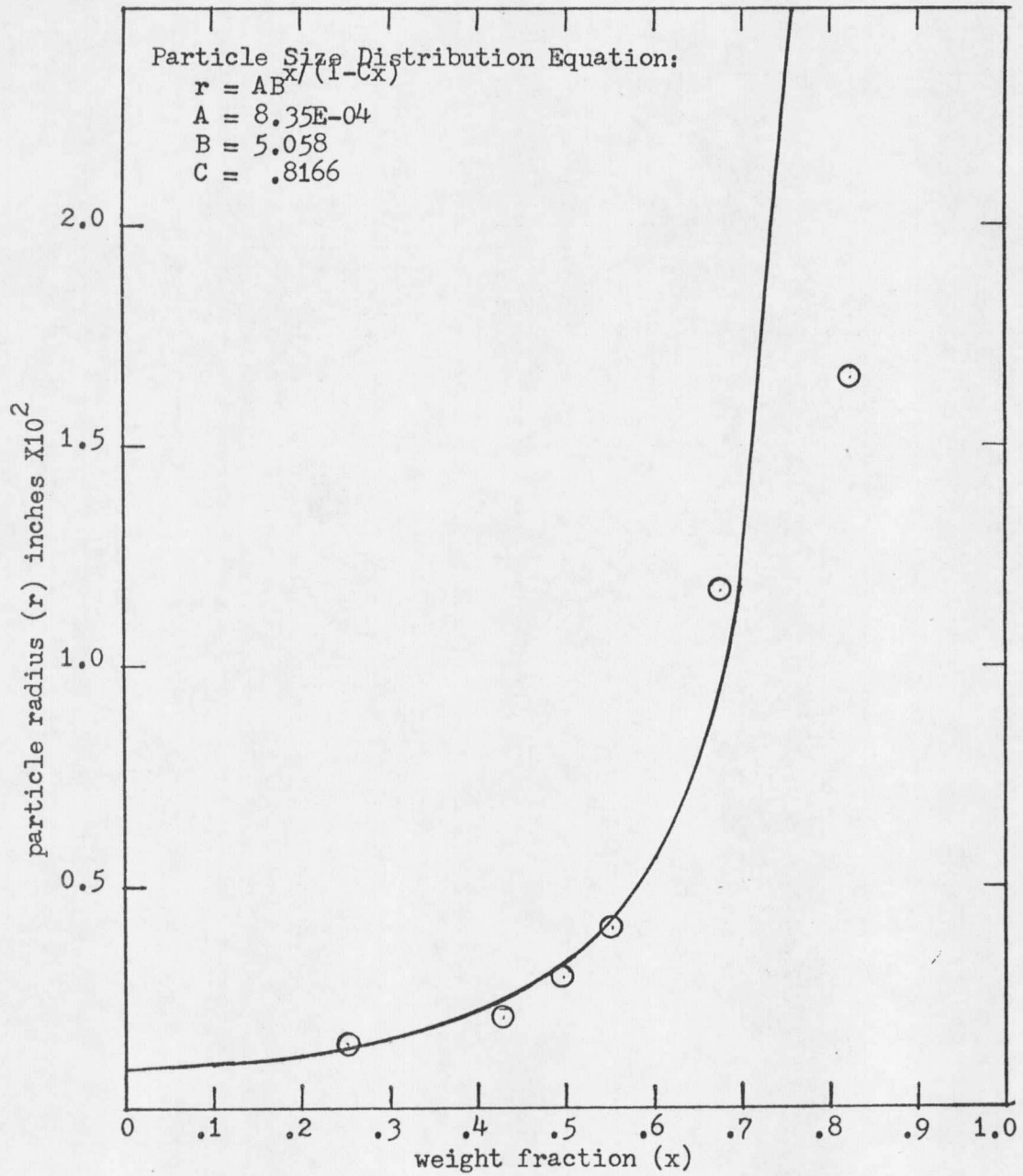


Figure 2
Comparison of Particle Size Distribution
Equation with Actual Data - Calcine



Fraction 3

Comparison of Particle Size Distribution
 Equation with Actual Data - Crushed Screen Oversize

and that $Q = x/(1-Cx)$, the equation becomes $r = AB^Q$. A and B can be obtained by taking the logarithm of both sides and fitting the resulting equation to the data using a least squares linear fit. ⁽¹³⁾

To find the best value for C a Golden Section search is made between $C = 0$ and $C = 1$ for the minimum sum of the squares for the difference between the calculated r values and the experimental r values. ⁽¹⁵⁾

The limits $C = 0$ and $C = 1$ were chosen because at $C = 0$ the curve becomes exponential and the distributions tested were at least exponential in nature, and at $C = 1$ the curve becomes discontinuous in the range from $x = 0$ to $x = 1$.

The effective holding time, EHT, is the holding time in the roaster required to obtain the average conversion of that stream. This parameter is carried along so that when a stream is divided the average conversions of the resulting streams can be obtained according to their particle size distributions.

The average conversion carried is the conversion of the component being roasted. In this case it is the fractional conversion of ZnS to ZnO.

Only the input streams must be specified by SEX vectors. The remaining streams are listed in a SNAME vector. The vectors SNAME and SEX1 through SEX100 make up the NAMELIST block SEXLIST.

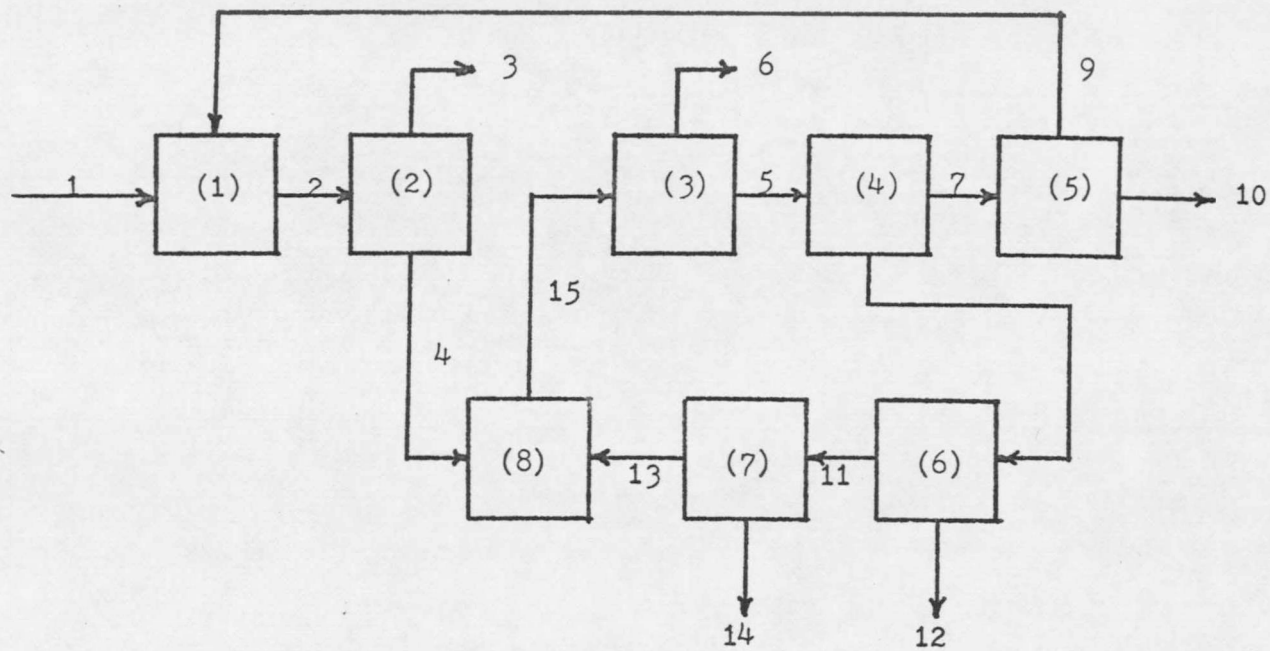
Although the vector name SNAME appears in both SINLIST and SEXLIST

it is not the same vector. The name SNAME is assigned to two different vectors that are handled separately by the CHESS system.

As a result of feedback or recycle streams in process networks the computer must either be programmed to solve a set of simultaneous equations or to carry out an iterative substitution calculation (12) around the loop or loops containing recycle streams. CHESS allows the user to use his insight into the processes being modeled to specify the best recycle sequence. The computer then makes iterative calculations around that loop until the results converge to a tolerance limit set by the user or the number of iterative calculations around the recycle loop reach a limit set by the user.

The sequence of the equipment node making up a recycle loop is specified in vectors KE2 and KE3 as input data. KE2 is used to describe the primary recycle loop and should always be used before KE3. It was felt by the designers of CHESS that the user would very rarely be required to use both vectors. Figure 4 shows how a recycle (12) loop might be specified using KE2.

The user can also specify certain streams on which he would like (12,25) to use Wegstein's method of convergence promotion. A list of these streams, up to ten in number, is input through the vector KE4. This allows the user to force convergence at several points in the



18

KE2 = 1,2,8,3,4,6,7,8,3,4,5,39*0,

Figure 4

Specifying Recycle Loops

recycle loop at the same time.

KE2, KE3 and KE4 are part of a NAMELIST block called KELIST. LOOPS, NPFREQ, DERROR and KTRACE make up the rest of the block. The value of LOOPS tells the computer the maximum number of loops in the recycle calculations. The allowable fraction of error in the recycle calculations is specified in DERROR. NPFREQ and KTRACE were used in the original CHESS system to print out intermediate data, but they are not used in the modified version.

The CHESS system can be divided into two major parts; the executive programs and the supporting programs. The executive programs include the main calling program and the subroutines that handle data input and manipulation. The supporting programs include the subroutines that simulate the actual chemical and physical processes that occur in the various pieces of equipment found in the plant being simulated. The supporting subroutines are generally built into the system and are called automatically as they are specified at a node in the process matrix; however the user can include his own equipment subroutines if required.

A Executive Programs (12)

1. MAIN - This program is the main calling program that controls the execution sequence in the system. Figure 16

2. DREAD - The principal function of this subroutine, called by MAIN, is data input to the system. This subroutine reads the input data and converts it to a form useable by the other subroutines in the system. Figure 17 shows a computer listing of DREAD.
3. CLEAN sets all of the data regions to zero. This subroutine is called by DREAD before data is read into the system for a new problem. Figure 18 is the computer listing for CLEAN.
4. ZERO is a matrix zeroing routine called by CLEAN to set the values for a large data array to zero. See Figure 19 for the computer listing of ZERO.
5. DCHECK, called by MAIN, takes the process matrix data read into the system by DREAD and checks it to make sure that all input, output, and intermediate streams are properly specified. If the data is inconsistent (If for example a stream labeled an intermediate stream does not go from one piece of process equipment to another) an error message is printed and the job terminated. A computer listing of DCHECK can be found in Figure 20.
6. DPRINT prints the stream and equipment sequences and the control constants read in as data. This subroutine,

called by MAIN, lists each piece of equipment by number, followed by the name of the pieces of equipment and the numbers of the streams entering and leaving it. The subroutines also list each process stream by number, and indicated between what pieces of equipment the stream exists. The control constants for the problem are also listed. Figure 21 is a program listing of DPRINT.

7. SUBSET is called by MAIN and is the subroutine that controls the computations made by the supporting programs. This subroutine is mainly a calling program that uses other subroutines to scan the equipment matrix and see that the necessary calculations are made. See Figure 22 for a program listing.
8. SCAN is a subroutine called by SUBSET that flags the process streams and pieces of equipment. Feed streams are given a flag of 1. All unused streams and pieces of equipment are given a flag of -1. And all other streams and pieces of equipment are given a flag of 0. Figure 23 is a computer listing of SCAN.
9. RCYCLE is called by SUBSET when recycle streams are present in the process being simulated. SUBSET systematically investigates each equipment node in the process matrix

and if the input streams to that node are defined the computations are made innediately. When the computations can no longer proceed due to the undefined status of some input stream or streams, RCYCLE uses an ordered list of equipment module numbers input by the user in an iterative recycle loop calculation. The system iterates on the loop calculation until the output stream data for each equipment node in the loop converges to a tolerance supplied by the user or until a specified iteration limit is exceeded. The system then sees if any further computations can be made with the newly defined stream data. If some equipment nodes remain uncalculated the system looks for a second list of equipment nodes and begins a second iterative recycle loop. A listing of RCYCLE can be found in Figure 24.

10. EQUIP transfers stream data from the system data storage to the input and output stream matrices of an equipment module. The stream data is then used by the equipment module to make the necessary calculations. EQUIP is called by either SUBSET or RCYCLE. The program listing is found in Figure 25.
11. REQUIP is the reverse of EQUIP in that it transfers data

from an equipment nodule back to the data storage block of the system after the equipment nodule calculations are completed. REQUIP is called by either SUBSET or RCYCLE. Figure 26 is a listing of this subroutine.

12. EQCALL is called by either SUBSET or RCYCLE and in turn calls the equipment subroutine to be used to make the calculations for a specific unit. If for example, a roaster is the piece of equipment specified at a particular equipment node, EQCALL will call the roaster subroutine. A listing of EQCALL can be found in Figure 27.
13. TEST called by RCYCLE controls the convergence of recycle loop iteration. This subroutine checks to see if output stream values are within convergence tolerances. It also incorporates Wegstein's method of convergence promotion^(12, 25). Wegstein's method allows convergence acceleration at every iteration step; the method is applied to each component as though it were the only variable in the loop. Convergence acceleration is applied to user - specified streams in the system. A listing of TEST is shown in Figure 28.

B Supporting Programs

The supporting subroutines are those subroutines developed

by the author during research to simulate a zinc plant. The input data, other than stream data, required for each subroutine to complete the necessary calculations is contained in a EQP vector. Each vector is developed specifically for the subroutine that it is used with because each equipment simulation subroutine requires a unique set of data. Complete descriptions of each subroutine and the models used are presented in the appendix along with the forms of the EQP vectors for those readers interested in using them. For those readers wanting an overall look at the system a brief description of each model follows.

ROASTER - ROST

The wedge roaster is presently in common use in the zinc smelting industry. The roaster consists of several hearths on which the zinc concentrates are roasted in beds approximately four inches deep. The material on each hearth is continually turned by large rotating rakes which move the concentrates to the inside or outside of the hearth where they fall to the next lower hearth. Some of the fine particles are carried out of the roaster by the roasting gases as overhead fines, the remaining solids are removed from the bottom of the roaster as roasted calcine.

For the development of this model it was assumed that:

1. All the concentrates going through the roaster have the

same reaction time. This is equivalent to saying the roaster is a plug flow reactor. Since the concentrates do not return to a hearth after falling to a lower one, this assumption is a valid one.

2. All the concentrates passing through the roaster experience the same effective environment. Since the rakes turn the concentrates on the hearths continually, it is assumed that each concentrate particle spends the same amount of time at each position in the bed.
3. All of the overhead fines have the same effective holding time in the roaster. The fines are carried out of the roaster by the overhead gases and are collected in the dust collecting system. It is assumed that all of these fines spend the same length of time in the reaction zone.
4. The diffusion of reaction gas through the roasted zone of a spherical particle is the rate controlling step. In a heterogeneous reaction the rate is determined by the kinetic rate, the rate of diffusion through a gas film surrounding the particle, or by diffusion through the reacted zone surrounding the unreacted core

(for reactions that have a solid product). The kinetic rate increases exponentially with temperature while the diffusion rate increases linearly with temperature. Therefore, at high temperatures the rate of diffusion is usually the controlling factor. Since diffusion through a solid is normally slower than diffusion through a gas film in a system where the gas in the bulk phase is moving, diffusion through the reacted zone was assumed to be the rate controlling step.

Figure 5 shows the particle model used for this simulation. Where R equals the overall particle radius, and r_c equals the radius of the unroasted core.

Setting up a material balance across a differential element yields:

$$-1/r^2 \left(d(r^2 N_{Ar})/dr \right) = dC_A/dt$$

where N_{Ar} is the mass transfer rate of oxygen at any r , and C_A is the concentration of oxygen.

By assuming a pseudo steady state, the time term can be neglected and the equation becomes:

$$d(r^2 N_{Ar})/dr = 0$$

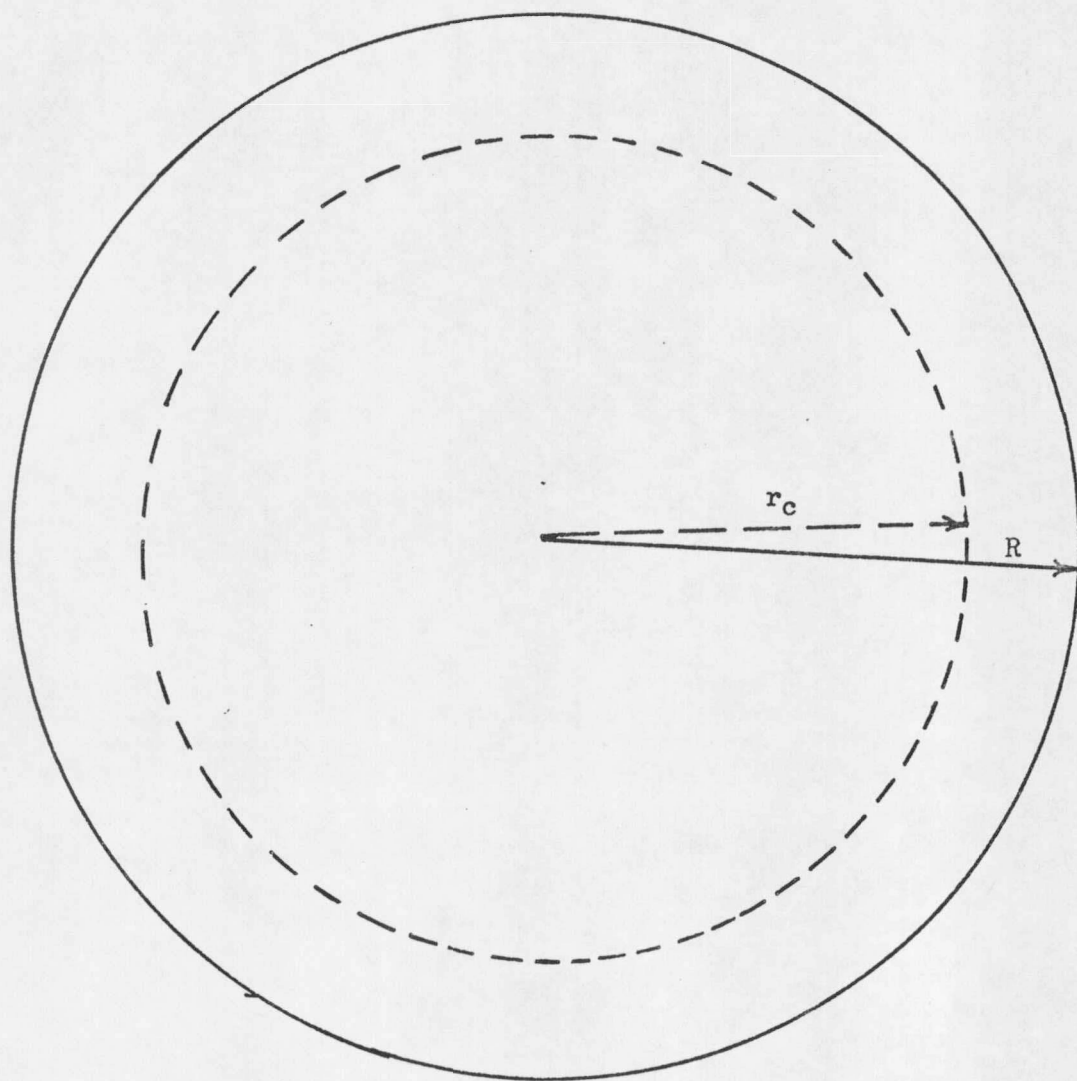


Figure 5
Particle Model for Roaster

This equation is then solved using the following boundary conditions:

a. $C_A = C_A^*$ at $r = R$, where C_A^* = the equilibrium concentration of oxygen absorbed on the surface of the particle.

b. $C_A = 0$ at $r = r_c$.

The time dependence was obtained by setting the number of moles of ZnS reacting per unit time equal to the number of moles ZnS per unit volume multiplied by the change in volume of the unreacted core per unit time.

When the equations are integrated and the appropriate substitutions made, the equation used in the simulation is obtained:

$$R^3 - r_c^3 - K(Rr_c/(R - r_c))t = 0$$

Where K is an empirical constant that depends on the ore being roasted and the roasting conditions, and t is the holding time in the roaster. The equation is solved for r_c using the Newton-Raphson technique.⁽⁷⁾

The amount of material left unreacted is proportional to the volume of the unreacted core. Therefore, the conversion can be obtained from the following equation:

$$\text{conv} = 1 - r_c^3/R^3$$

As the concentrates pass through the roaster they are divided

into two streams. The very small concentrate particles are carried along by the roasting gases leaving the top of the roaster. The remaining concentrates pass through the roaster and are removed from the bottom of the roaster as roasted calcine. The roaster model divides the incoming stream into two outgoing streams, an overhead fines stream, and the roasted calcine stream. The division is based on a critical particle radius. This radius is the radius of the particle whose buoyant forces, based on the upward velocity of the roasting gases, are equal to the forces of gravity. All particles with smaller radii are assigned to the overhead fines stream, and those with larger radii go to the calcine stream.

In the roaster model it is necessary to account for the degradation and agglomeration that occurs in the calcine stream. To do this the constants for the equation describing the particle size distribution are adjusted by multiplying them by "agglomeration factors". These factors are part of the equipment parameters and are empirical constants that depend on the ore being roasted and the roasting conditions.

After the incoming stream is divided into two streams, and the particle size distribution equation for each stream is specified, the average conversion for each outgoing stream is calculated. The average conversion is obtained by calculating the conversion at

several radii in the particle size distribution and taking the integral average. The average conversion is then used together with stoichiometric constants to obtain the composition and weights of the outgoing streams.

SCREEN - SCRN

Screens are used in industry whenever some solid material being processed must have a particle size less than or greater than a given size. The screen is a divider that separates material according to particle size. The material with particle sizes greater than the screen opening remains on top of the screen and goes to the oversize stream. The material with particle sizes smaller than the screen opening passes through the screen to become the undersize stream. The simulation model compares the size of the screen opening to the size of the particles. The fraction of material with particle sizes greater than the screen opening are assigned to the oversize stream, while the rest of the material is assigned to the undersize stream.

The conversion of the small particles in the incoming stream is greater than the conversion of the larger particles. When this stream is divided into two streams according to particle size, the average conversion of each outgoing stream will reflect the difference in their particle size distributions. The average conversion of the undersize stream will be higher than the average conversion of the

incoming stream, while the average conversion for the oversize stream will be lower. In the simulation model the average conversion of each stream is calculated from the effective roaster holding time and the particle size distribution of the corresponding stream. The average conversions are then used together with stoichiometric constants to obtain the compositions and the weights of the outgoing streams.

GRINDER or BALLMILL - GRND

For this model it was assumed that when a solid material is retained an infinite length of time in a size reduction unit, all of the material will reach a minimum size beyond which no further size reduction is possible. The difference between the radius of the particle being ground and the radius of the minimum sized particle was used as the driving force in the grinder model.

It was further assumed that the grinding process was first order, using the difference between the radius of the particle and the minimum particle radius as a driving force. The process is then analogous to a first order kinetic reaction with an equilibrium concentration; replacing the chemical concentration by the particle radius.⁽⁹⁾

The form of the equation used in the grinder model is:

$$r_n = (r - r_m) \text{EXP}(-k/w) + r_m$$

Where

r_n = the new particle radius - inches

r = the original particle radius - inches

r_m = the minimum particle radius - inches

w = material rate - tons/day

k = empirical constant that depends on the geometry of the grinder and the grindability of the solid material.

ELECTROSTATIC PRECIPITATION - PREP

In 1907 Frederick Cottrell developed the first electrostatic precipitator. This precipitator was used to remove acid mist from the gases leaving a sulfuric acid process.⁽²⁰⁾ Since that time the electrostatic precipitator has become a very important piece of equipment for the removal of particulate matter from gas streams.

The equation that has been used to describe the efficiency of an electrostatic precipitator was developed in 1922 by W. Deutsch.⁽¹⁹⁾

$$\text{Eff} = 100 (1 - e^{-Av/Q})$$

Where:

A = area of collecting surface.

v = the electrically induced drift velocity toward the collecting surface.

Q = volume rate of air flow through the precipitator.

FLUE - FLUE

Some of the dust particles leaving the roaster settle out the gas stream in the flue leading to the electrostatic precipitator. Also some of the dust particles not trapped in the electrostatic precipitator are collected in the flue between the precipitator and the plant stack. Since the fines collected from these flues are recycled back to the process, they must be considered in the plant simulation.

In a flue the dust settles out by gravity and is collected in bins at the bottom of the flue. The amount of dust that settles out is a function of the gas velocity in the flue, geometry of the flue, and solids loading. After trying several times to get an accurate model for a flue, it was decided to use the electrostatic precipitator model, with different constants, as the flue model.

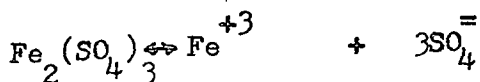
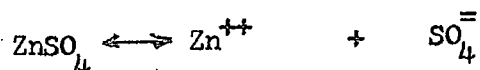
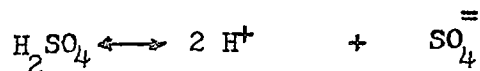
There was not enough core capacity in the computer for the extra electrostatic precipitator subroutine so the user must use PREP instead of FLUE whenever he wants to simulate a flue.

H_2SO_4 and $\text{Fe}_2(\text{SO}_4)_3$ Addition - AMIX

In actual plant operation some H_2SO_4 and $\text{Fe}_2(\text{SO}_4)_3$ must be added to the leach circuit. This subroutine simulates that addition along with adjusting the weight fraction solids in the leach tanks.

The subroutine calculates the total moles of ZnO entering the

The sulfate compounds all exist as ions in the leach solution in the following form:



In the leaching of the ZnO the amounts of H^+ , Fe^{+3} and Zn^{+2} change while the total amount of $\text{SO}_4^{=}$ is constant. Therefore in the model, the ions are used instead of the compounds from which they come.

The Fe^{+3} ion concentration can be related to the H^+ ion concentration by the equation:

$$\text{Fe}^{+3} = 2.56 \times 10^8 \left| \text{H}^+ \right|^3 \quad (\text{See Appendix})$$

where the concentrations are in lb mole/ft³. If the acid concentration is high enough, the calculated ferric ion concentration will be higher than the actual ferric ion concentration. When this condition exists, as it does in the acid leach, the solution is not buffered and the acid concentration changes rapidly with the amount of ZnO leached. When the calculated ferric ion concentration is less than the actual ferric ion concentration, ferric hydroxide will be precipitated until the actual ferric ion and acid concentrations are in equilibrium. This buffered condition exists in the neutral leach where the acid concentration is low. In the neutral leach the acid

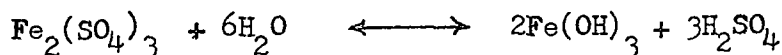
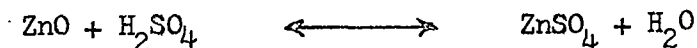
mixer. The total sulfate ($\text{H}_2\text{SO}_4 + \text{Fe}_2(\text{SO}_4)_3$) to ZnO ratio is then calculated. If the ratio is too low, H_2SO_4 or $\text{Fe}_2(\text{SO}_4)_3$, depending which is low, is added to the stream to make up the proper sulfate to ZnO ratio at the proper pH. In some cases both H_2SO_4 and $\text{Fe}_2(\text{SO}_4)_3$ are added. The user puts the desired $\text{H}_2\text{SO}_4/\text{ZnO}$ and $\text{Fe}_2(\text{SO}_4)_3/\text{ZnO}$ ratios into the computer as equipment parameter input data.

After adjusting the sulfate to ZnO ratio the subroutine calculates the weight fraction solids. If the calculated weight fraction solids is greater than a user specified value the subroutine adds water to the outgoing stream.

LEACH TANKS - LECH

The leaching of ZnO to form ZnSO_4 is a buffered leach with H_2SO_4 as the acid and $\text{Fe}_2(\text{SO}_4)_3$ as the buffer. This leach is divided into two sections a neutral leach and as acid leach. In the neutral leach the $\text{Fe}_2(\text{SO}_4)_3$ acts as a buffer, while the acid leach, the acid content is high enough that the $\text{Fe}_2(\text{SO}_4)_3$ does not act as a buffer.

The reactions involved in these leaches are:



ZnO and $\text{Fe}(\text{OH})_3$ are solids while the rest of the reactants are a part of the leach solution.

concentration changes very little as ZnO is leached, since ferric ion is being precipitated as ferric hydroxide to form acid.

The solution leaving the neutral leach goes to the electrolytic plant. Any iron in that stream will cause problems in the electrolysis of the zinc solution, so it is desirable to have the ferric ion concentration as low as possible in that solution. If the acid concentration is 10^{-4} lb mole/ft³ the ferric ion concentration will be 2.56×10^{-4} lb mole/ft³. If the acid concentration is lowered to 10^{-5} lb mole/ft³ the ferric ion concentration is lowered to 2.56×10^{-7} lb mole/ft³. It can be seen that maintaining the outlet stream from the neutral leach at a pH near 7 is necessary.

In the acid leach, the solids from the neutral leach are mixed with cell acid from the zinc electrolysis plant. This gives a high acid content and a very low ferric ion content so the ferric hydroxide present in the solids from the neutral leach reacts with some of the acid to form ferric ion. In the acid leach generally there is enough acid so that all of the ferric hydroxide goes to ferric ion leaving an excess of acid. The first thing the leach tank model does is to check the ferric ion - acid balance and to adjust it if there is ferric hydroxide present.

After the ferric ion - acid balance is checked, the subroutine calculates the amount of ZnO leached. This calculation is based on

the assumptions:

1. The leach tank is a well mixed reactor. The leach tanks are pachuca tanks (These are wooden cylindrical tanks with conical bottoms). In the center of each tank is an air lift that draws material from the bottom of the tank to a splash pan covering the top of the tank. The material then circulates from top to bottom with an overflow at the top.⁽⁶⁾ With this type of mixing the assumption of a well mixed reactor is valid.
2. The leaching rate is proportional to the acid concentration. Since acid is the leaching agent, it is reasonable to assume that the leaching rate is dependent on it.
3. The leaching rate is proportional to the surface area of the ZnO. Whenever a reaction involves a solid-liquid interface, the size of that interface usually affects the reaction rate.

When these assumptions are developed into equations; the result is three equations and three unknowns. These are:

Unknowns

- (1) r = radius of average ZnO particle in the leach tank at time t

(2) C_a = concentration of acid at time t

(3) C_f = concentration of ferric ion at time t

Equations

$$(1) \quad C_f = 2.56 \times 10^8 C_a^3$$

$$(2) \quad r = ((C_{AO} - C_A + 3(C_{FO} - C_F))r_o^3 / K T C_A (\text{wt } \% \text{ ZnO})_{in})^{\frac{1}{2}} = 0$$

$$(3) \quad r = r_o ((\text{ZnO}_{in} - (v_o/2)(C_{AO} - C_A + 3(C_{FO} - C_F)) / (\text{wt } \% \text{ ZnO})_{in})^{1/3}$$

where

K = empirical rate constant

T = holding time

C_{AO} = acid concentration at time = 0

C_{FO} = ferric ion concentration at time = 0

r_o = radius of average ZnO particle entering the leach tank

v_o = volumetric flow rate of slurry into the leach tank

ZnO_{in} = moles ZnO/ unit time entering leach tank

These equations were solved by a Golden Section search⁽²⁾ for the concentration of acid that gave the same value for r from equations 2 & 3. Once the proper value for C_A was known, C_F and r were fixed. The conversion of ZnO to ZnSO_4 was calculated using the equation:

$$\text{conv} = 1 - \frac{r_o^3}{r^3}$$

After the conversion of ZnO to ZnSO_4 was calculated, that conversion along with the necessary stoichiometric constants were used to calculate the amount of each component leaving the leach tank.

CLASSIFIER - CLAS

The classifier model is based on user input. The user inputs a critical radius and the computer puts all the solid material with particle radii greater than the critical radius in the oversize stream and the rest of the solid material in the undersize stream. The weight fraction solids in the oversize stream is also an equipment parameter entered as data by the user. The computer uses this piece of data together with the weight of solids in each stream to make a material balance on the liquid components, coming into the classifier, to obtain the composition of the outgoing streams.

THICKENER - SETL and FILTER - FLTR

The thickener and filter are simulated by the same subroutine. The subroutine makes a material balance around the piece of equipment based on the weight fraction solid in the two output streams⁽⁵⁾. As part of the equipment parameter data the user provides the computer with the weight fraction solids in each output stream.

Some theoretical equations have been developed for the thickening process, but the results of these developments required

more data than is generally available to the average user.^(18, 22, 23)

MIXER - MIXR

The mixer model will mix up to six incoming streams to form one output stream. The weight of each component in the output stream and the total weight of the output streams are calculated in the model by addition of the incoming values.

The particle size distribution equation of the output stream is obtained by ordering all of the incoming particle sizes from the smallest to the largest, and doing a least squares fit of the resulting distribution.

ZINC ELECTROLYSIS - ELET

In this subroutine, the computer uses a chemical efficiency supplied by the user to calculate the amount of zinc produced. The chemical efficiency is the fraction of the input zinc sulfate converted to zinc.

The electrical current required to produce the zinc is calculated using Faraday's Law.⁽¹⁰⁾ The cell current is calculated from the required current using a current efficiency supplied by the user.

The composition of the cell acid returning to the acid leach is calculated in the final section of this subroutine. This cell acid stream is the only output stream from the electrolysis process since

the amount of zinc produced is printed out in the subroutine itself.

C Computing Subroutines

The process module subroutines use four subroutines, CONVT, ISC, SIME, and TGS, to make calculations that are required in many places.

CONVT is a subroutine that solves the equations developed in the roaster model. This routine calculates a trial and error average conversion (ZnS to ZnO) from the particle size distribution, roasting constant, and the roaster holding time.

ISC does a least squares fit of the particle size data to get the particle size distribution equation ($r = AB^{x/(1-Cx)}$) for a process stream. The subroutine does a Golden Section search for the proper value for C, and for each value of C it does a least squares analysis of the data to compute A and B.

SIME solves two simultaneous equations for two unknowns.

TGS is the counter part of CONVT. It calculates the holding time required to get a certain average conversion. It uses the same equations as CONVT.

Computing Hardware and Software

The computer used in this study was an 80 K word Sigma 7 built by Xerox Data Systems. The programming was done in Fortran IV H computer language, and the program was executed under the BPM operating system.

In order to fit the program into the working core under the BPM system it was necessary to overlay⁽¹²⁾ the program. The subroutines were placed into seven element files and overlaid as shown in Figure 6.

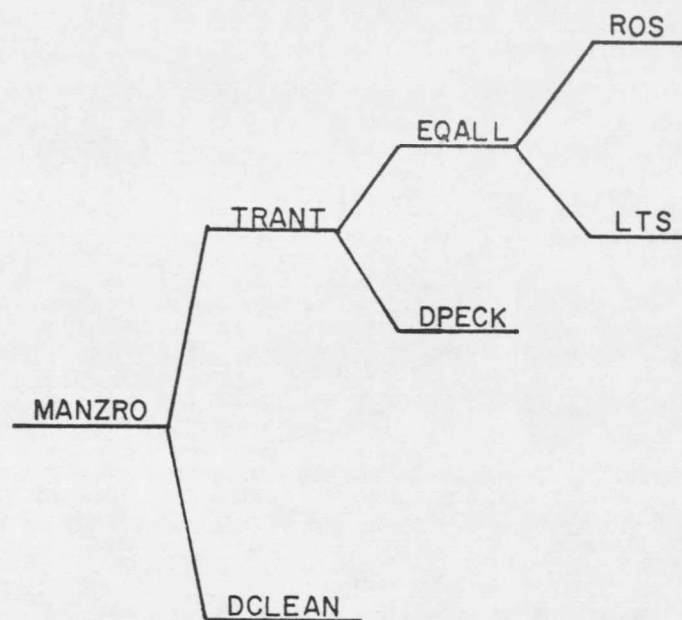
The subroutines placed in each element file were:

1. MANZRO
 - a. Main calling program
 - b. ZERO
2. DCLEAN
 - a. CLEAN
 - b. DREAD
3. DPECK
 - a. DCHECK
 - b. DPRINT
4. EQALL
 - a. EQCALL
 - b. EQUIP

- c. EQUIP
- d. REQUIP
- e. RCYCLE
- f. SCAN
- g. SUBSET
- h. TEST
- i. CONVT
- j. LSC
- k. SIME
- l. TGS
- m. GRND
- n. MIXR
- 5. TRANT
 - a. TRANSF
 - b. BIPRNT
 - c. PIPRNT
- 6. ROS
 - a. ROST
 - b. SCRN
 - c. PREP
- 7. LTS
 - a. AMIX

- b. LECH
- c. CIAS
- d. SETL .
- e. FLTR
- f. ELET

With the program overlayed in this manner, the data block required 41,136 bytes of core and the operational part of the program required 6,737 bytes of core.



45

Figure 6
Overlay Scheme

Results and Discussion

The primary objective of this project was to modify CHES to simulate a zinc plant. The plant was divided into two sections; a roasting section and a leaching section. Simplified block diagrams of each section are presented in Figures 7 and 8.

Simulation of the roasting section required ten equipment nodes interconnected as shown in Figure 9. The seven primary roasters used to roast fresh calcine are simulated by one roaster designated by "R-1", and the recycle solids from the flue system plus the crushed solids are roasted in roaster "R-2". The roaster nodes, flue nodes, precipitator node, screen node, and the crushing make up a total of seven nodes. The remaining three nodes are mixer nodes that must be placed at each point two or more streams come together.

The purification plant was not included in the simulation of the leaching section. The purification plant removes cadmium and other impurities from the solution going to electrolytic plant. For this problem the chemical and physical properties of only zinc and iron were considered. Compounds of other metals such as cadmium were included as gangue which was assumed to be a solid inert material and as such no purification plant was required to remove these compounds from solution. The 34 equipment nodes and their connecting elements used to simulate the leach section are shown in Figure 10.

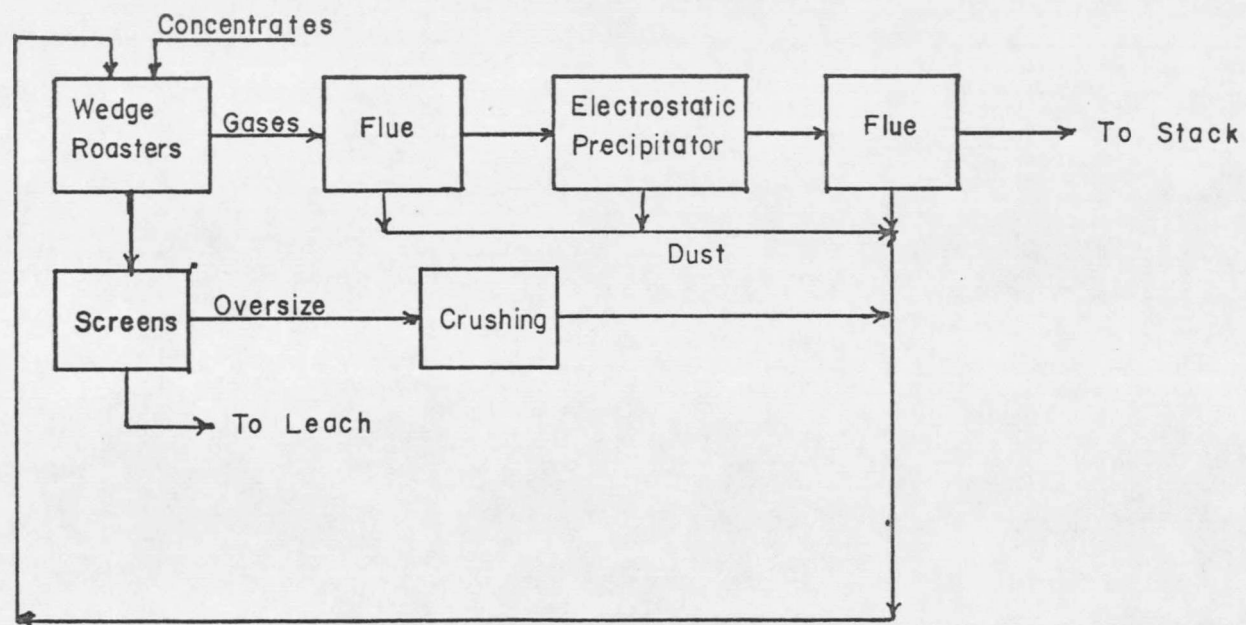


Figure 7
Roasting Plant Layout

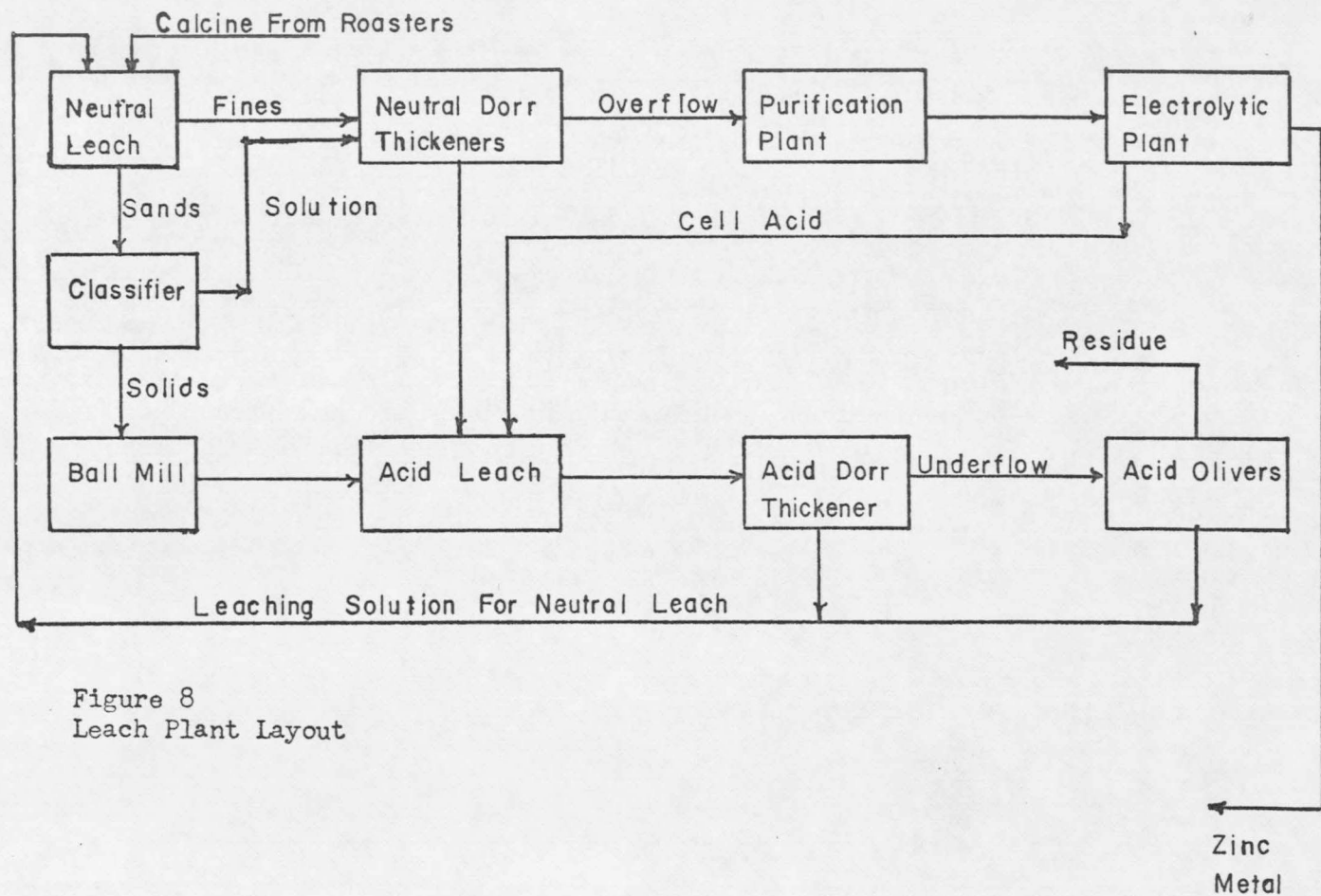


Figure 8
Leach Plant Layout

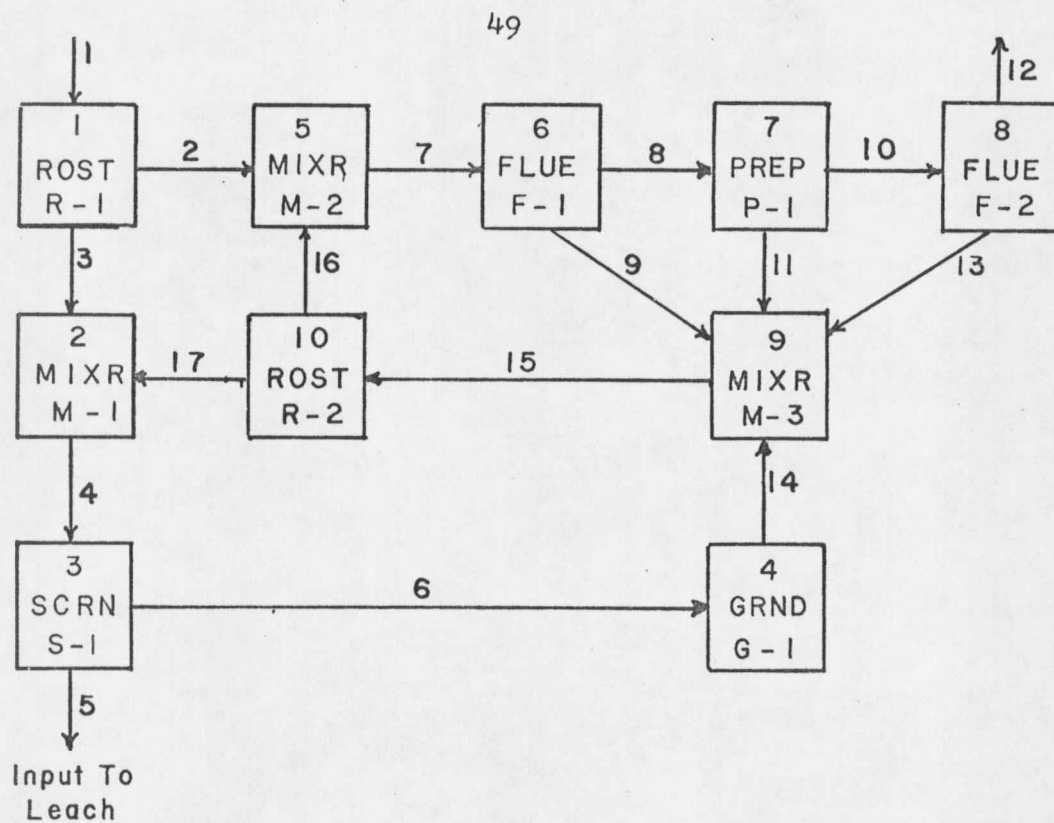


Figure 9
Equipment Node Sequence
For Roasting Plant Simulation

Figure 10
Equipment Node Sequence
for Leach Plant Simulation

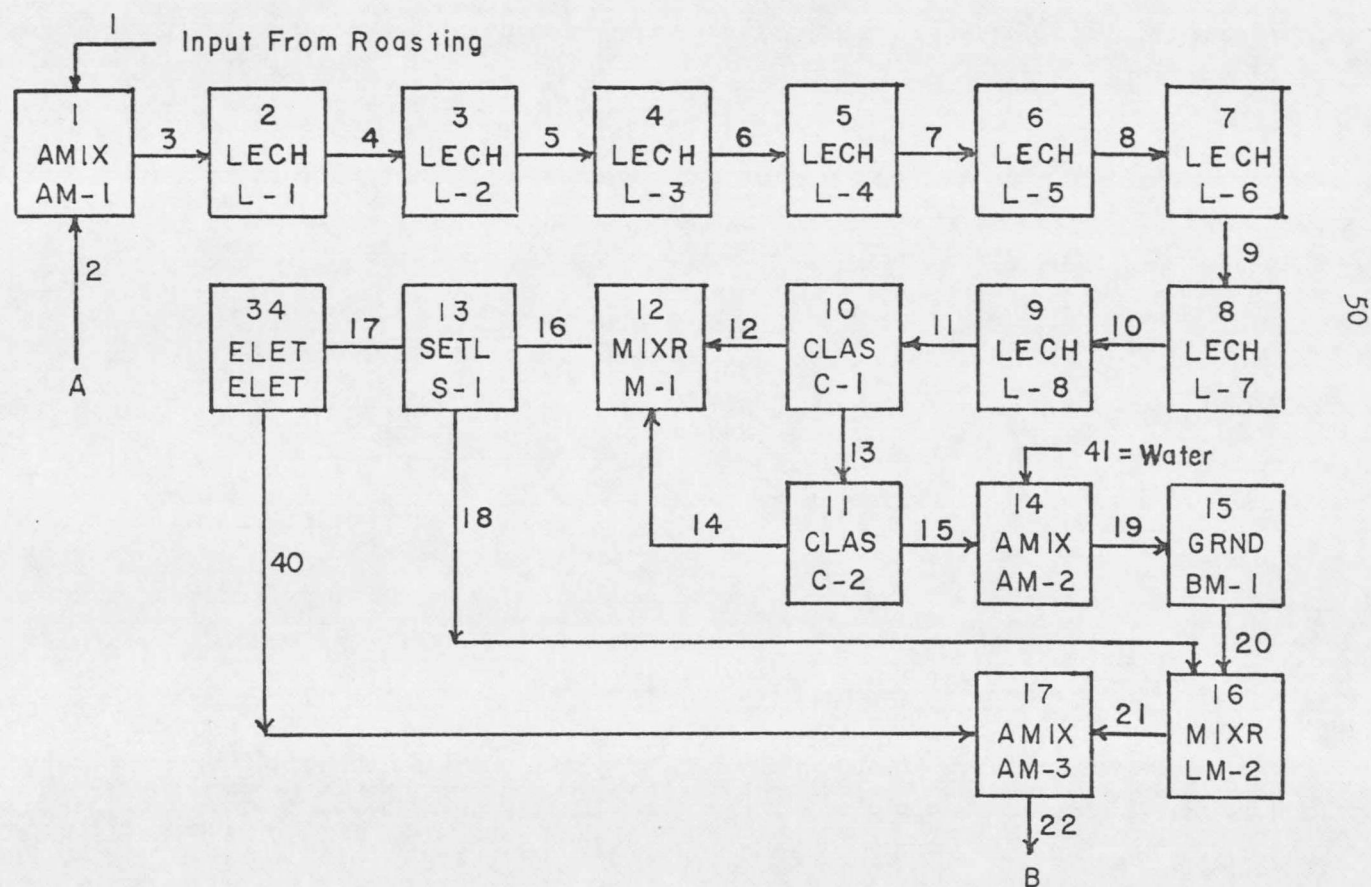
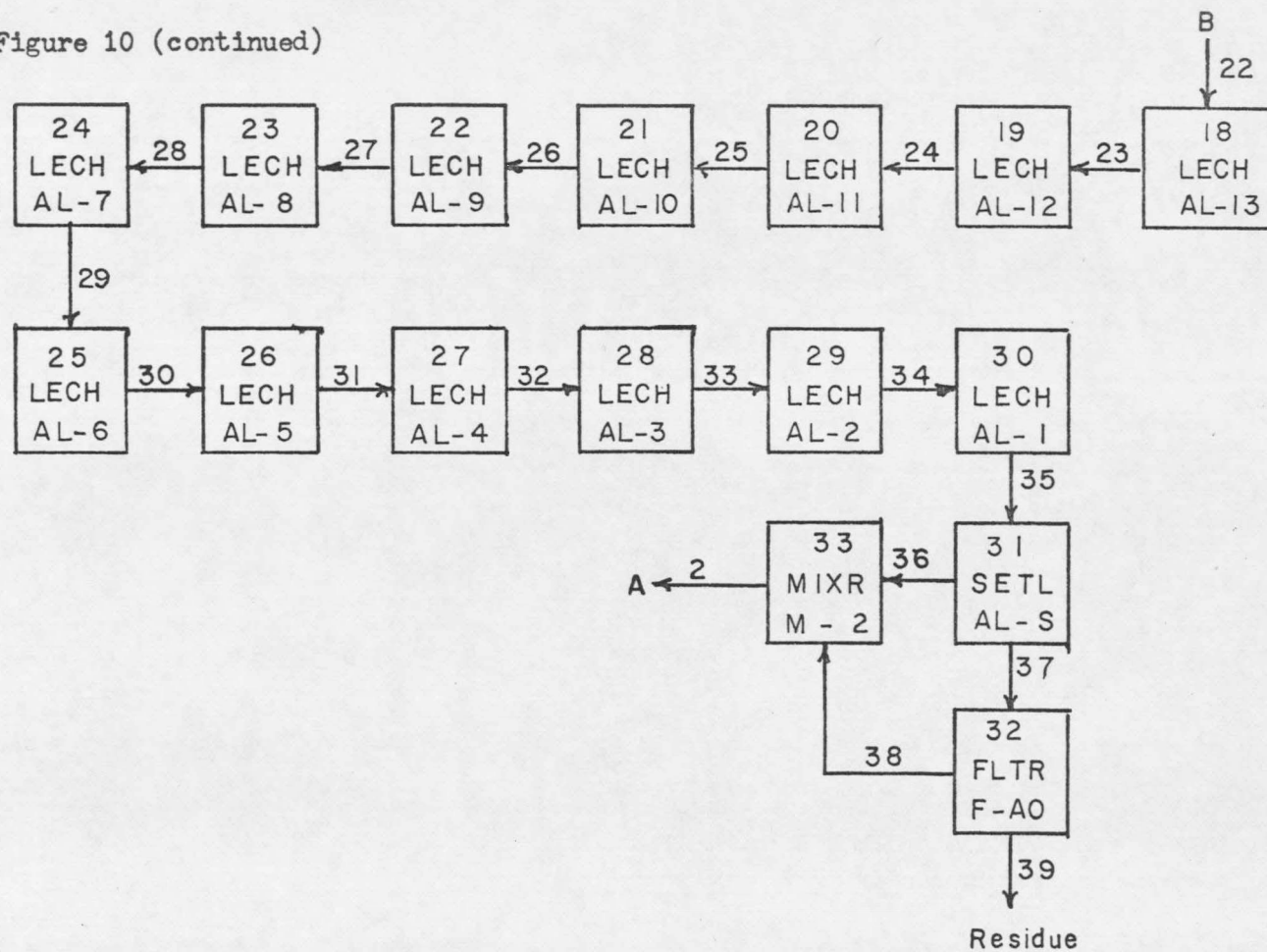


Figure 10 (continued)



An explanation of the data inputs for the two simulations, and the computer results are given in the Appendix. Table I compares the results of the computer calculations with plant data obtained from the Anaconda Company.

From Table I it can be seen that values calculated by the simulation program are close to the values supplied by Anaconda. This shows that through adjustment of input parameters, the models and sub-routines developed can be used to simulate the zinc plant. The results of this simulation were used as a basis for changes made in input parameters.

The first input parameter to be changed was the roasting capacity. The number of primary roasters was reduced from seven to six and then to five with corresponding decreases in concentrate input from 350 tons/day to 300 tons/day and 250 tons/day respectively. The results of this series of tests are shown in Figure 11.

Figure 11 shows that the change in zinc production was not directly proportional to the change in concentrate input. The reduction in zinc production for a reduction in plant input is less than would be experienced if zinc production was directly proportional to concentrate input. This should occur because as the roasting capacity and concentrate input are reduced the sizes of the other pieces of processing equipment do not change. The flue system has

Table I

Calculated Results vs Plant Data

Variable	Plant Data	Calculated Results
Plant Input	350 tons/day	350 tons/day (set)
Fines Lost overhead by primary roasters	70 tons/day	58 tons/day
Conversion of Calcine	93%	95.6%
Efficiency of Flue system to precipitator	39%	36%
Efficiency of electrostatic Precipitator	95%	94%
Efficiency of flue from precipitator to the stock	64%	56%
% incoming Zn leached in neutral leach	60%	60.17%
% Zn Remaining leached in acid leach	15%	14.4%
% Zn removed from solution in electrolytic plant	50%	50% (set)
Tons/Day Zn produced		116.3

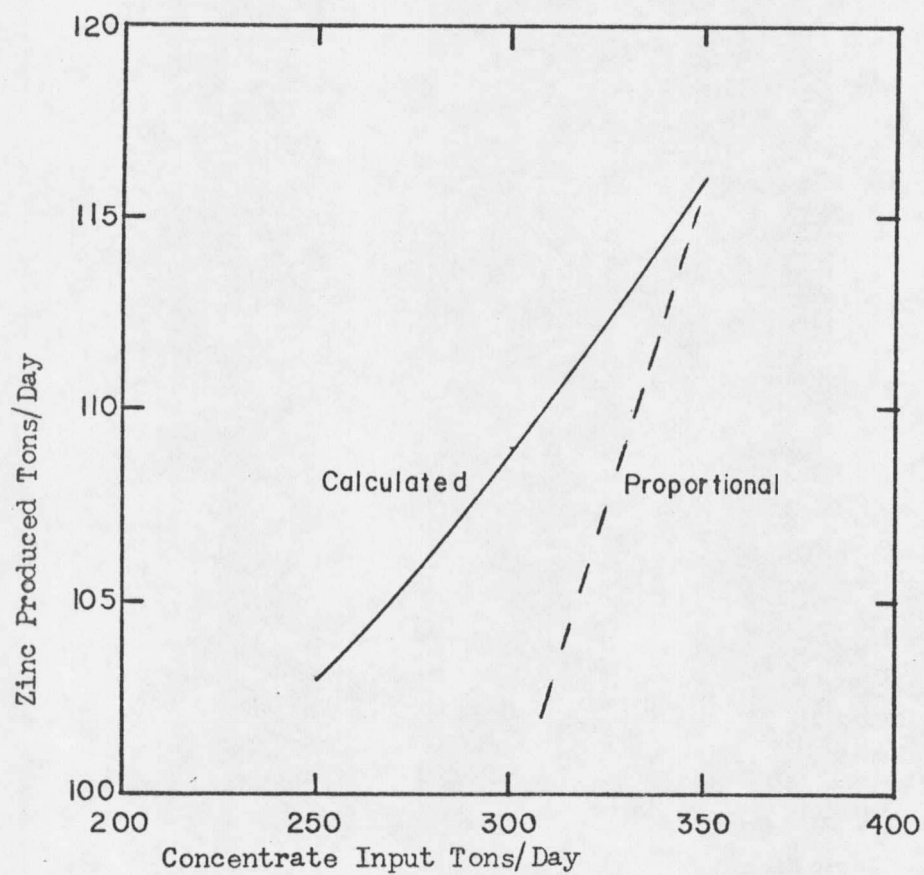


Figure 11
Zinc Production vs Plant Input

the same size so the velocities in the flue system will decrease, the collection efficiencies of the flues and precipitator will increase, and less will be lost up the stack. The leach capacity will not change causing the holding time in the leach system to increase. An increase in holding time will increase the amount of zinc leached from the calcine and will make more efficient use of the acid. There will be less zinc lost in the residue and more zinc recovered in the electrolytic plant. The net effect is that plant capacity damps changes in roasting capacity.

Figure 12 is a representation of the relationship between the roasting constant and the calculated zinc production. This relationship indicates that the effects of changing the constant are more pronounced as the value of the constant is reduced. When considering what this means in terms of actual plant operation it must be remembered that a concentrate with a low constant is hard to roast while a concentrate with a high constant is easy to roast. The highest possible fractional conversion of ZnS to ZnO is one, and as the roasting constant increases the conversion asymptotically approaches one. Increasing an already high roasting constant will increase the amount of ZnS converted to ZnO only slightly and will therefore not affect the zinc production significantly.

The calculated results of reducing the number of neutral leach

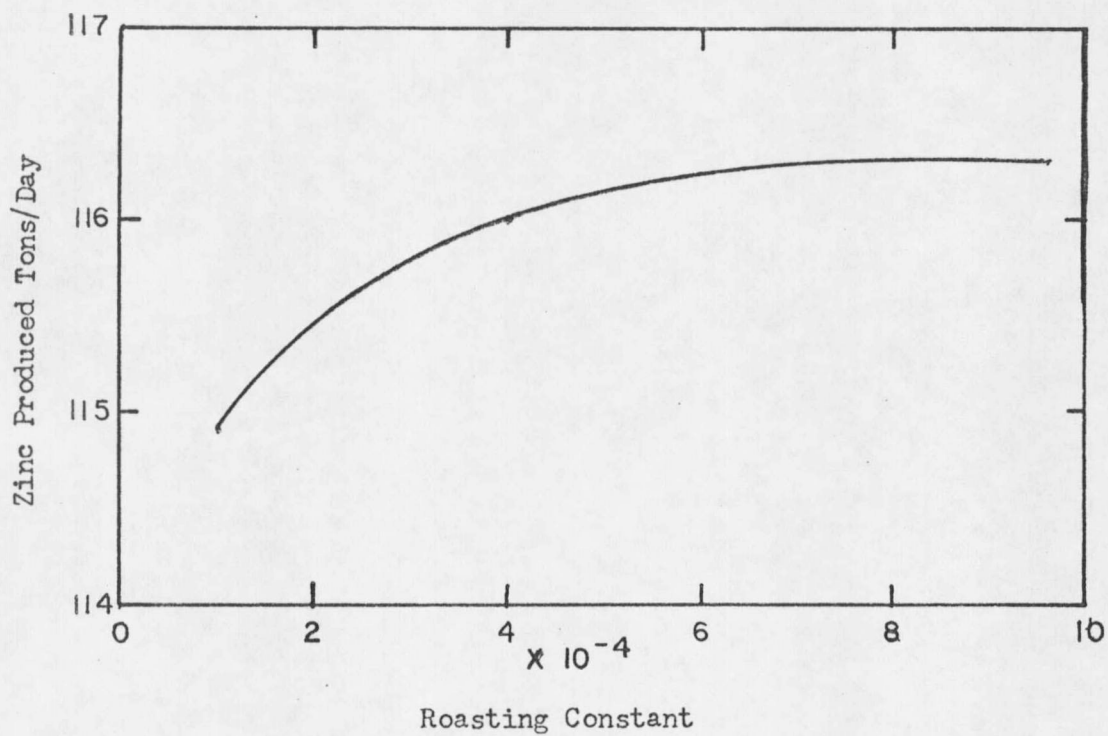


Figure 12
Zinc Production vs Roasting Constant

tanks is shown in Figure 13. It was not expected that the zinc production would first decrease and then increase as the number of neutral leach tanks removed from service increased. Examination of the neutral and acid leach systems provided the reason for this phenomenon. At the head end of the neutral and acid leaches, the acid and ferric sulfate to zinc oxide ratios were maintained at a given value. As tanks were removed from the neutral leach the fraction of incoming zinc leached in the neutral leach decreased, causing more zinc oxide to go to the acid leach. As more zinc oxide entered the acid leach, more acid was added and the acid leach became more acidic. Since the leaching rate is dependent on the acid concentration, the acid leach became more effective. The net result was that the increase in the amount of zinc leached from the calcine in the acid leach was greater than the decrease in the neutral leach (Figure 14).

Based on the results obtained by removing neutral leach tanks from service, it would appear that it would be more efficient to leach at a higher acid concentration in fewer leach tanks. It must be remembered that ferric sulfate is present. The purpose of ferric sulfate is twofold; it acts as a buffer by going to ferric hydroxide and releasing hydrogen ion, and as the ferric hydroxide precipitates some impurities such as arsenic and antimony are precipitated with

Figure 13
Zinc Production vs Number of Leach Tanks
Removed from Neutral Leach Circuit

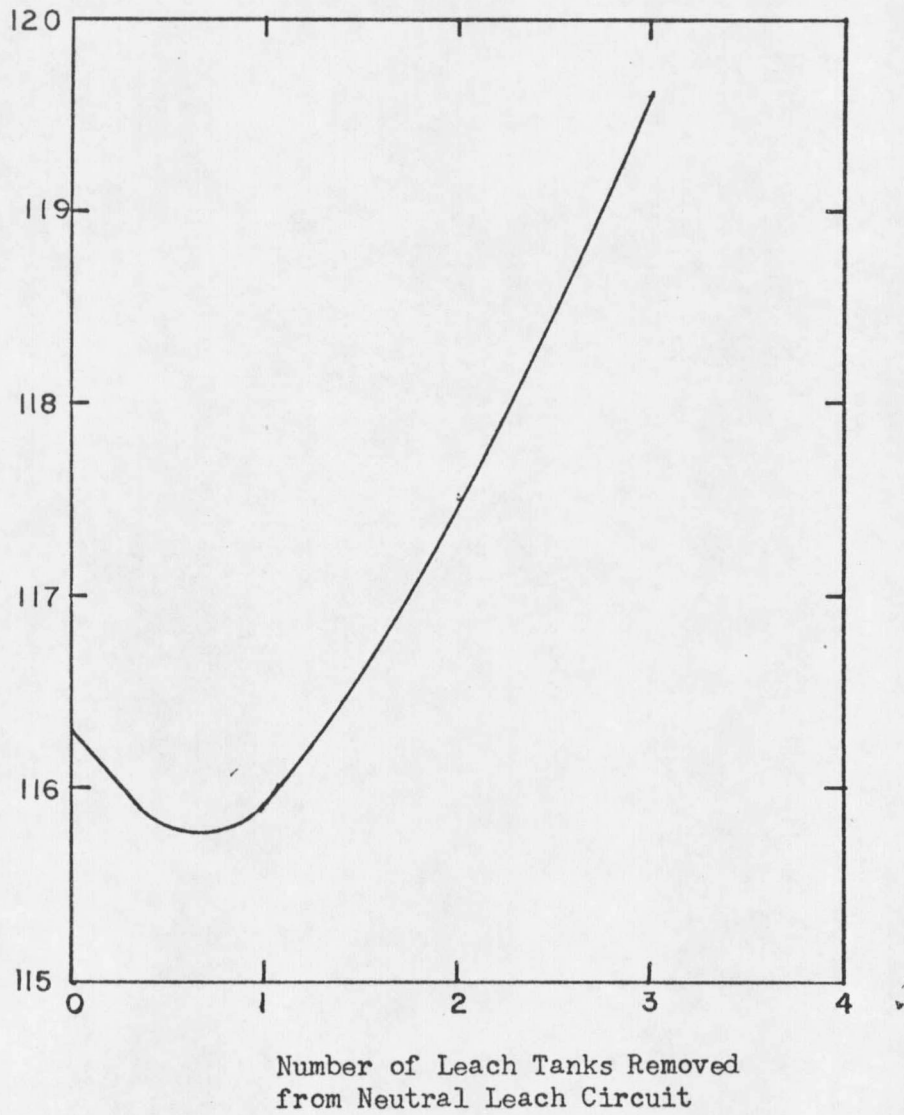
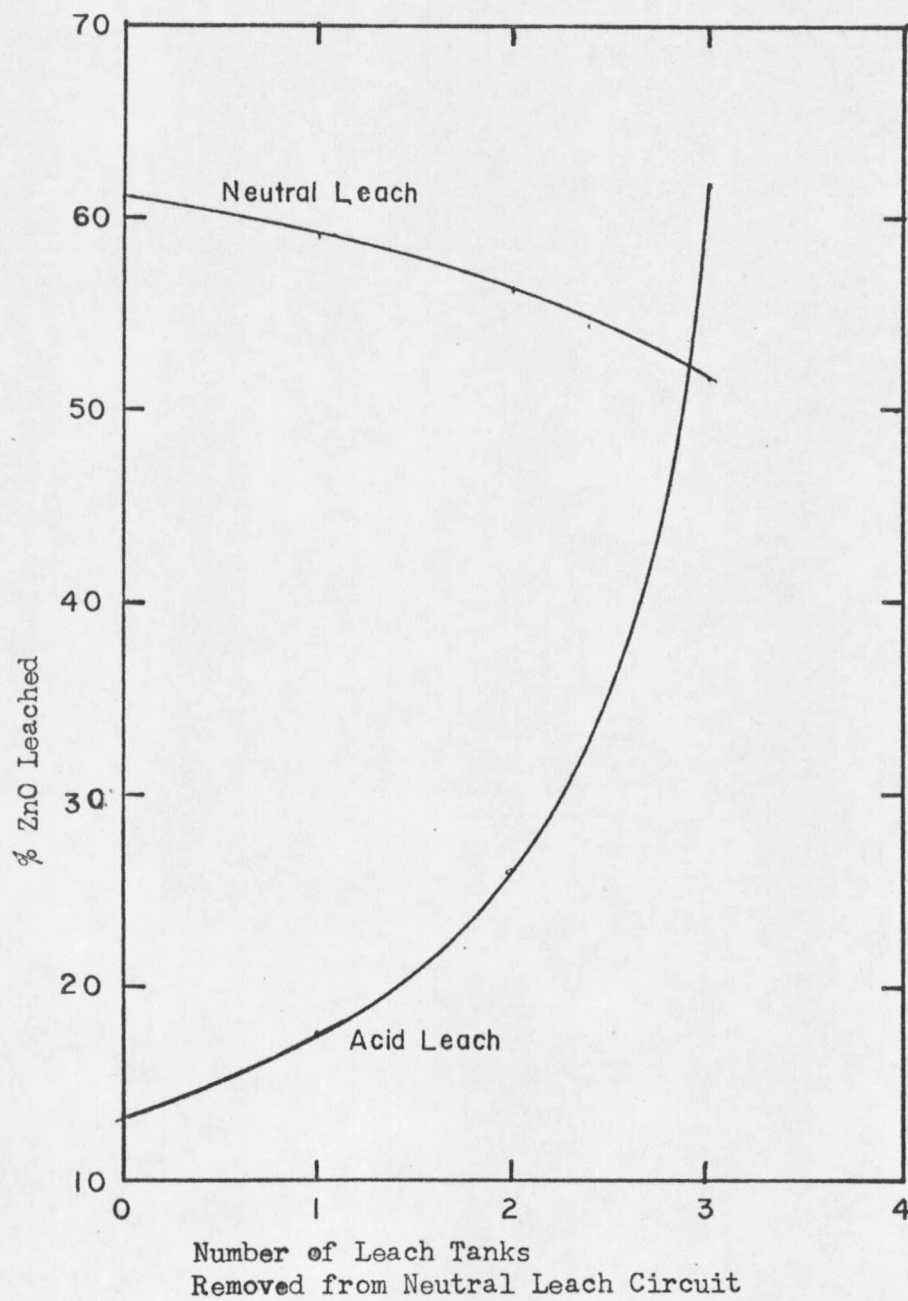


Figure 14
ZnO Leached vs Number of Leach Tanks
Removed from Neutral Leach Circuit



it. The ferric sulfate is unwanted in the electrolytic tanks, because it lowers the efficiency of the electrolytic plant if it remains in solution after the neutral leach, so it is necessary that the amount of ferric sulfate leaving the neutral leach be kept at a minimum. Removing three of the neutral leach tanks increased the weight percent ferric sulfate leaving the neutral leach by two orders of magnitude (Figure 15) showing the need for the neutral leach tanks.

The model for the electrolytic plant does not have the capability of predicting the effects of impurities on cell efficiency. For this reason it is not apparent what effect the increase in ferric sulfate, caused by removing neutral leach tanks, would have on the amount of power required to produce a ton of zinc.

The results of the tests show that the models developed can be used to simulate a zinc plant. Changes in the input parameters allow the user to determine the effect of changes in plant operation such as reducing roaster through put, changing the concentrate properties, and changing the leach configuration. Thus the primary objectives of the project have been met.

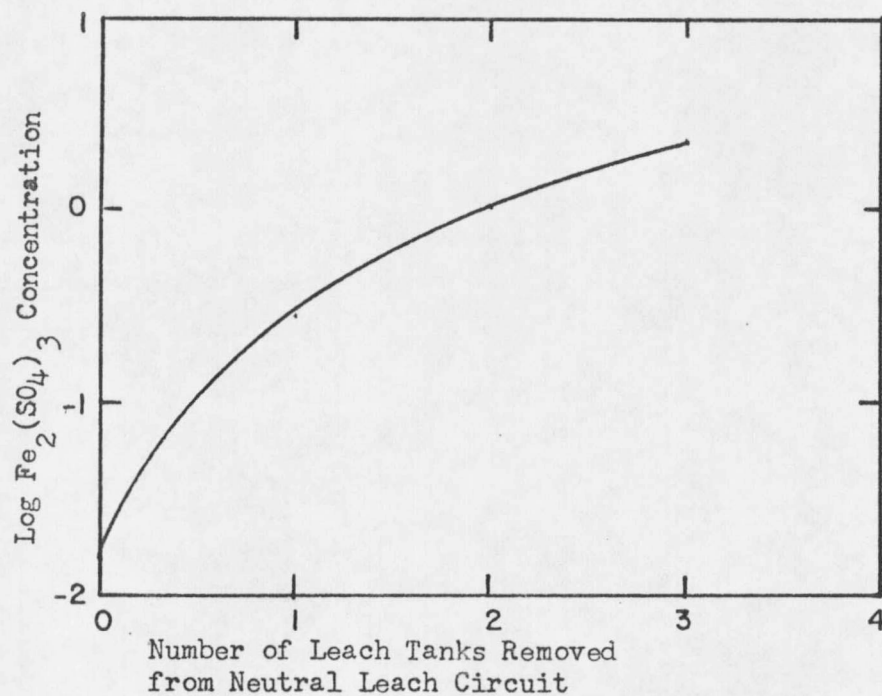


Figure 15

Ferric Sulfate Concentration Leaving Neutral
Leach vs Number of Neutral Leach Tanks
Removed from Neutral Leach Circuit

Recommendations

This project has shown that CHESS can be modified to simulate a zinc plant, and that the simulation can be a useful tool in the examination of the effects caused by changes in plant operation. To make the simulation a more useful tool it is recommended that:

1. Each piece of operating equipment be studied separately.

These studies would provide information that could be used to improve the individual equipment subroutines.

2. The effects of impurities on plant operation be studied, and the simulation modified to include these effects.

3. The simulation be expanded to simulate other types of inorganic plants. Thus ultimately achieving the goal of having a system that can be used to simulate most inorganic processing plants as CHESS does for organic processing plants.

Summary

The primary purpose of this research project was to develop mathematical models to simulate inorganic processes. The particular processes simulated were those in operation at the Anaconda zinc plant in Great Falls, Montana. The mathematical models developed were made into subroutines that fit the CHESS computer system. The calculated results obtained from the computer system were compared to data from the plant at Great Falls.

The models developed were:

1. An equation for the particle size distribution of the solid materials involved in the processes.
2. A model for a wedge roaster used to oxidize ZnS to ZnO . This model can be applied to any process involving a solid-gas reaction with a solid product.
3. A model for the division of a solids stream where some of each particle size goes to each stream. The example used in this paper is an electrolytic precipitation where some of the particles in each size range are collected, although the larger particles are collected more readily.
4. A model of a ballmill which can be used to simulate other particle size reducing processes where the final particle has a finite size.

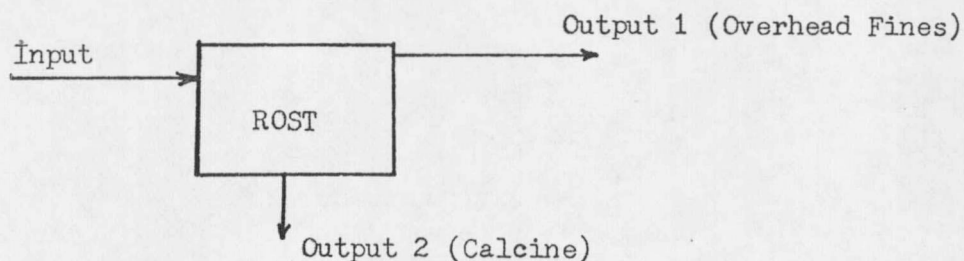
5. A mixer model which simulates the mixing of two or more solids streams of different size distributions.
6. A leach tank model which simulates a well mixed reactor where the reaction occurs at the interface between a solid and a liquid. This model also includes a simulation representing a buffered solution.

These models provide a basis for future simulation of inorganic processes. They can be modified and expanded to fit a particular application. Since this research provides a basis for future simulation of inorganic processes it has been successful.

Appendix

Detailed Description of Supporting Subroutines

ROASTER



The roaster model is based on the following assumptions:

1. All the concentrates passing through the roaster have the same reaction time
2. All the concentrates passing through the roaster experience the same effective environment
3. All of the overhead fines have the same effective holding time in the roaster
4. The rate controlling step is the diffusion of the reaction gases through the reacted zone of a spherical particle.

Figure 5 shows a particle with an unreacted core of ZnS covered by a layer of ZnO. Where r_c = the radius of the ZnS core and R is the total radius of the particle.

The equations used in the mathematical model were developed in the following manner:

1. Make a material balance across a differential element:

$$4\pi r^2 N_{Ar}|_r - 4\pi r^2 N_{Ar}|_{r+\Delta r} = 4\pi r^2 \Delta r \frac{dC_A}{dt}$$

Divide through by $4\pi r^2 \Delta r$

$$-(1/r^2)(d(r^2 N_{Ar})/dr) = dC_A/dt$$

Where N_{Ar} equals the mass transfer rate of oxygen at any r and C_A is the concentration of oxygen at any r .

2. Assume Pseudo Steady State;

$$-(1/r^2)(d(r^2 N_{Ar})/dr) = 0$$

Multiply both sides by $-r^2$

$$d(r^2 N_{Ar})/dr = 0$$

Integrate

$$r^2 N_{Ar} = Q = \text{Constant}$$

3. Apply Fick's First Law

$$N_A = -CD \frac{dx_A}{dr} + x_A (N_A + N_B)$$



$$N_B = 2/3 N_A$$

$$N_{Ar} = -CD \frac{dx_A}{dr} + x_A (N_{Ar} - 2/3 N_{Ar})$$

$$N_{Ar} = -CD \frac{dx_A}{(1 - 1/3 x_A)} dr = Q/r^2$$

$$- CD \int dx_A / (1 - 1/3 x_A) = Q \int dr / r^2$$

$$3 CD \ln(1 - 1/3 x_A) = -Q/r + S$$

4. Solve for Q and S using the boundary conditions;

a. At r equal to R , C_A equals C_A^* or assuming a constant concentration x_A equals x_A^* where x_A^* is the mole fraction oxygen absorbed on the surface of the particle

b. At r equal to r_c , x_A equals 0

$$\text{At } r = r_c$$

$$3 CD \ln(1 - 0) = -Q/r_c + S$$

$$Q/r_c - S = 0$$

$$S = Q/r_c$$

$$\text{At } r = R$$

$$3 CD \ln(1 - 1/3 x_A^*) = -Q/R + S$$

$$3 CD \ln(1 - 1/3 x_A^*) = -Q/R + Q/r_c$$

$$3 CD \ln(1 - 1/3 x_A^*) = (R - r_c)Q/R r_c$$

$$r_c^2 N_{Ar} = Q = 3 CD R r_c \ln(1 - 1/3 x_A^*) / (R - r_c)$$

5. Get the time dependence

moles of oxygen transferred = moles of oxygen reacted

2/3 of a mole ZnS reacts per mole of oxygen reacted

$$4\pi R^2 N_{Ar} = 3/2 4\pi r^2 P dr/M dt$$

Where P is the density of ZnS and M is the molecular weight of ZnS

$$R^2 N_{Ar} = Q = 3/2 r^2 (P/M)(dr/dt)$$

$$Q \int_0^t dt = (3P/2M) \int_R^{r_c} r^2 dr$$

$$Q t = (P/2M) r^3 \Big|_R^{r_c}$$

$$Q t = -(P/2M)(R^3 - r_c^3)$$

$$r_c^3 = R^3 + 6 M C R r_c \ln(1 - 1/3 x_A^*) t / P(R - r_c)$$

6. Combine constants to get the equation used in the roaster model

$$r_c^3 = R^3 + K(R r_c / (R - r_c)) t$$

Where K is a combination of the other constants and is determined empirically. The value for K depends on the ore being roasted and the roasting conditions. The easier an ore is to roast and the better the roasting

conditions, the higher the value for K.

The equation is solved for r_c using the "Newton Raphson" technique. (7)

To calculate the fractional conversion of a particle it is assumed that the unreacted fraction is proportional to the volume of the unreacted core:

$$\text{conv} = 1 - \frac{\text{volume of the unreacted core}}{\text{volume of the original particle}}$$

$$\text{conv} = 1 - (4/3\pi r_c^3) / (4/3\pi R^3)$$

$$\text{conv} = 1 - r_c^3 / R^3$$

The concentrates entering the roaster are divided into two output streams. The small particles are carried out the top of the roaster, as overhead fines, by the roasting gases. The remaining concentrates pass through the roaster and are removed from the bottom of the roaster as roasted calcine. The separation is based on a critical particle radius. This radius is the radius of a particle whose buoyant forces, in the upward flowing roasting gases, equal the force of gravity on that particle. This critical radius is proportional to the square root of the gas velocity, (5) and is calculated, in this model, by multiplying the square root of the

volumetric gas flow rate by an empirical constant. The constant is a function of the roaster geometry and the gas properties at the roasting conditions. The standard cubic feet of gas passing through the roaster is an input parameter. The subroutine corrects this value for pressure and temperature to get the volumetric gas flow in the roaster.

The fraction of material going to the overhead fines stream is equal to the fraction of material with particle sizes less than the critical particle radius. The particle size distribution equation for this stream is determined by fitting it to radii ranging from the minimum incoming radius up to the critical radius. These radii are calculated from the size distribution equation for the incoming stream.

The fraction of material going to the calcine stream is equal to the fraction of material with particle sizes greater than the critical radius. The particle size distribution equation for this stream cannot be obtained from the incoming particle size distribution directly due to agglomeration and degradation of this stream in the roaster. The constants for the calcine particle size distribution are determined by multiplying the corresponding constants of the incoming distribution by "agglomeration factors". These empirical factors are specified as equipment parameters by the user and are a function of the ore being roasted and the roasting conditions. The factor for

C must be chosen carefully because if C becomes equal to, or greater than one the size distribution equation becomes discontinuous.

After the particle size distribution of each output stream is specified, the average conversion of each stream is calculated. The conversions of representative particles, equally spaced along the particle size distribution, are calculated and the integral average of these conversions is used as the average conversion for the stream. The conversion of a particle is calculated from its radius and the holding time using the equations developed earlier in this section. For material entering the roaster already partially converted, the holding time used is the sum of the time previously spent in a roaster and the time spent in this pass through the roaster.

The equipment parameters for the roaster are specified in an equipment parameter vector in the following manner:

$$\text{EQPJ} = J, \text{KC}, \text{FG}, \text{P}, \text{T}, \text{RC}, \text{OC}, \text{CTC}, \text{AC}, \text{BC}, \text{CC}, \text{STC}_1, \dots$$

$$\dots \text{STC}_N, \text{M} * \text{O}, \dots$$

J = Equipment number

KC = Component number of the key component, in this case the component number of ZnS

FG = Standard cubic feet of gas/min. leaving the roaster

P = Pressure in atm.

T = Average temperature in the roaster ($^{\circ}\text{F}$)

RC = The constant used to calculate the radius of the particle whose buoyant forces equal the force of gravity; Critical radius = RC (Volumetric gas flow rate)

OC = The holding time constant for the overhead fines; This constant divided by the volumetric gas flow rate gives the holding time, (days) in the roaster for the overhead fines

CTC = The holding time constant for the calcine; This constant divided by the weight of the solids passing through the roaster gives the holding time, (days) in the roaster for the calcine

AC, BC, CC = "Agglomeration constants" for the constants (A, B, C) used in the particle size distribution equation for the calcine

$\text{STC}_1, \dots, \text{STC}_N$ = Stoichiometric constants; These constants give the pounds of each component produced per pound of the key component converted

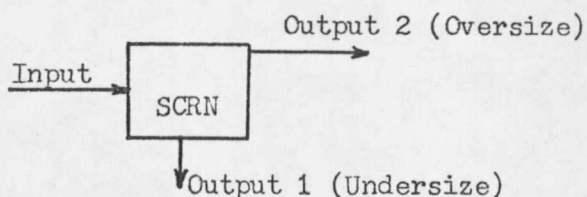
N = The total number of components (NOCOMP) minus 1; The last component is always the gas phase and figured in the total weight and therefore does not

75

have a weight fraction

M = Balance of vector, up to 25 elements, is filled
out with zeros.

SCREEN



A screen is a piece of equipment that divides a solids stream into two streams according to particle size. In the screen model, the particle size distribution of the incoming stream is compared to the size of the screen opening. The fraction of material with particle sizes less than the screen opening is assigned to the undersize stream, while the remaining solids are assigned to the oversize stream.

The particle size distribution for the output streams are determined by fitting the particle size distribution equation to the proper segment of the incoming size distribution. For the undersize stream the equation is fit to radii, calculated from the incoming size distribution, ranging from the minimum incoming radius up to the radius of the largest particle that will pass through the screen. The equation for the oversize stream is fit to radii ranging from the smallest particle remaining on the screen up to the largest particle in the incoming stream.

The conversion of a particle decreases as the particle radius increases for a constant roaster holding time, (as explained in the

roaster model).

Therefore, when a stream is divided according to particle size the oversize stream will have a lower conversion than the undersize stream. In the screen model, the average conversions of the undersize and oversize streams are calculated using the corresponding particle size distributions and the effective roaster holding time carried in the SEX vector of the input stream. The calculations made are the same calculations made to calculate conversion in the roaster model.

The equipment parameter vector for the screen is:

$$\text{EQPJ} = J, \text{ KC}, \text{ SS}, \text{ STC}_1 \dots \text{ STC}_N \text{ M} * 0 \dots$$

J = Equipment number

KC = Component number of the key component, in this case the component number of ZnS

SS = Screen size, the radius of the largest particle that will pass through the screen (Inches)

$\text{STC}_1 \dots \text{ STC}_N$ = Stoichiometric constants; These constants give the pounds of each component produced per pound of the key component reacted

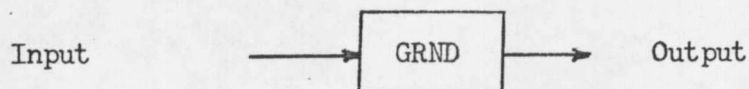
N = The total number of components (NOCOMP) minus 1; The last component is always the gas phase and it is handled separately from the other components

M = Balance of vector, up to 25 elements, is filled

78

out with zeros.

GRINDER or BALLMILL



The size reduction process was assumed to be first order, with an equilibrium size. It is reasonable to expect that if a given amount of material is processed in a size reduction unit for an infinite length of time, the particles will all reach an equilibrium size beyond which no further size reduction is possible.

The first order size reduction process with an equilibrium particle size is analogous to a first order chemical reaction with an equilibrium concentration. The equation for the chemical reaction⁽⁹⁾ was modified to the following form for the size reduction process.

$$r = (R - r^*)e^{-Kt} + r^*$$

where

r = new particle radius

R = original particle radius

r^* = equilibrium radius

K = grinding constant

t = holding time

The subroutine converts the incoming particle size distribution to the particle size distribution of the outgoing stream, using the

above equation and the equipment parameter vector:

$$\text{EQPJ} = J, \text{RE}, \text{Kt}, 22*0,$$

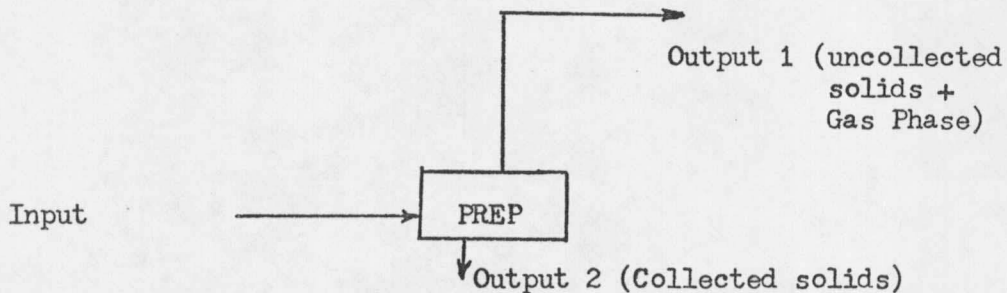
where

J = equipment node number

RE = equilibrium radius (inches)

Kt = combination grinding constant and holding time constant;
this number when divided by the mass flow rate through the
grinder gives the Kt term of the equation

ELECTROSTATIC PRECIPITATOR



In 1922, W. Deutsch⁽¹⁹⁾ developed a simple equation describing the efficiency of an electrostatic precipitator:

$$\text{Eff} = 100(1 - e^{-Av/Q})$$

where

A = area of collecting surface

v = the electrically induced drift velocity toward the collecting surface

Q = volume rate of flow through the precipitator

This equation is only good for a given particle size, since the drift velocity is proportional to the particle diameter, according to equations presented by Deutsch.

G.W. Penny in 1969⁽¹⁹⁾, expanded the Deutsch equation to describe the collection of dusts with particle size distributions.

For two particle sizes, the equation is:

$$\text{Eff} = 100(1 - F e^{-Av/Q} - (1-F)e^{-AKv/Q})$$

where

F = fraction of small particles

v = drift velocity of the small particles

K = a constant that when multiplied by v gives the drift velocity of the larger particles

Based on the examples given by Penny and the equations of Deutsch⁽¹⁹⁾, relating drift velocity to particle size, the equation can be modified to the form:

$$Eff = 100(1 - F e^{-AKr_1/Q} - (1-F)e^{-AKr_2/Q})$$

By expanding the equation to fit more particle sizes and making A part of the constant, (for a given precipitator A is constant) the equation used in the model is evolved:

$$Eff = 100(1 - F_1 e^{-Kr_1/Q} - F_2 e^{-Kr_2/Q} - \dots - F_n e^{-Kr_n/Q})$$

In the subroutine, the particle size distribution of the incoming stream is divided into 20 equal parts. The efficiency for the average particle in each segment is calculated. The above equation is then used to get the total efficiency and the efficiencies of 10 equal segments of the particle size distribution.

Since the smaller particle sizes have a higher efficiency than the larger particle sizes, the particle size distributions of the two output streams will be different from each other and from the input particle size distribution. The total efficiency; the efficiencies of the 10 particle size distribution segments, and the incoming particle size distribution are used to calculate new particle size distributions for the outgoing streams (See computer printout for method).

The last section of the subroutine calculates the fractional conversion of ZnS to ZnO for the outgoing streams based on the incoming conversion and effective roaster holding time (See roaster model write up). These conversions are then used to make the material balance necessary to calculate the composition of the outgoing streams.

The equipment parameter vector for this model is:

EQPJ = J, KC, T, K, STC, ... STC_N, M*O,

where

J = equipment node number

KC = component number of ZnS

T = temperature in precipitator °F

K = precipitator constant

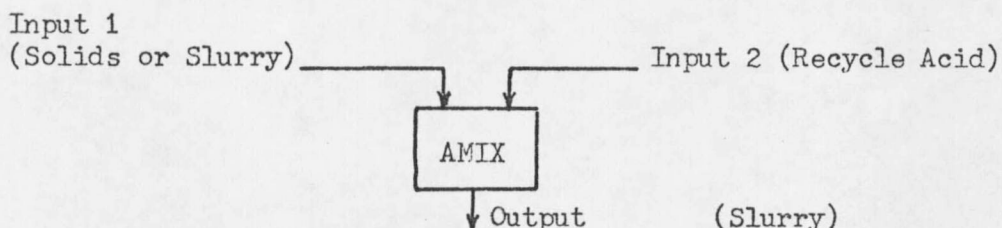
STC₁, ... STC_N = stoichiometric constants

(lb component produced/lb ZnS reacted)

M*O = balance of vector, up to 25 elements, filled out with
zeros.

ACID and FERRIC SULFATE ADDITION

Leach Slurry Tank



AMIX does not correspond to a single piece of equipment, instead it simulates the addition of the leaching agents to a solids stream. The ZnO , H_2SO_4 , and $\text{Fe}_2(\text{SO}_4)_3$ must be maintained at certain ratios to each other if the leaching process is to operate correctly. In the actual plant this is done by operator control, but in the CHES simulation scheme it must be done as though there is a piece of equipment to do it.

The user inputs, as part of the equipment parameters, a sulfuric acid to zinc oxide ratio and a ferric sulfate to zinc oxide ratio. The subroutine, after mixing streams 1 and 2, checks to see if enough acid and ferric sulfate are present, and if not one or both are added to the levels specified by the user. It is possible for the levels of one sulfate compound to be low while the level of the other sulfate compound is high, giving a total sulfate to zinc oxide ratio that is about right. The total sulfate to zinc oxide ratio is more important

than the acid and ferric sulfate to zinc oxide ratios, therefore the total sulfate ratio is checked by the subroutine and if it is low acid and/or ferric sulfate is added depending on which is needed.

After adjusting the acid and ferric sulfate levels, the subroutine checks the weight fraction solids in the outgoing stream. If the weight fraction solids in the output stream is higher than a user specified level water is added.

The subroutine prints out the amounts of acid, ferric sulfate, and water added.

The constants required by this subroutine are specified in the following equipment parameter vector:

EQPJ = J, KC1, KC2, KC3, KC4, NS, AR, FR, WFS, 16*0.,

where

J = equipment node number

KC1 = component number of ZnO

KC2 = component number of H_2SO_4

KC3 = component number of $\text{Fe}_2(\text{SO}_4)_3$

KC4 = component number of H_2O

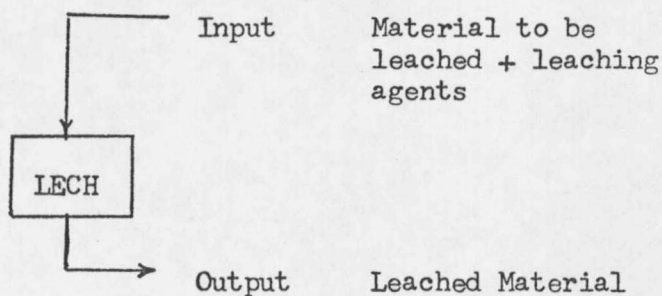
NS = number of solid components

AR = H_2SO_4 to ZnO ratio (moles H_2SO_4 /mole ZnO)

FR = $\text{Fe}_2(\text{SO}_4)_3$ to ZnO ratio (moles $\text{Fe}_2(\text{SO}_4)_3$ /mole ZnO)

WFS = weight fraction solids

NEUTRAL and ACID LEACHS



The neutral and acid leachs are parts of a ferric sulfate buffered leach. In the neutral leach the acid concentration is low enough that the ferric sulfate acts as a buffer while in the acid leach the acid concentration is high enough that the buffering capacity of the ferric sulfate is exceeded.

The ferric ion concentration can be related to the hydrogen ion, or acid concentration, through equilibrium considerations:

$$(\text{Fe}^{+3}) (\text{OH}^{-})^3 = 10^{-36}$$

$$(\text{H}^{+}) (\text{OH}^{-}) = 10^{-14}$$

$$(\text{OH}^{-}) = 10^{-14} / (\text{H}^{+})$$

$$(\text{Fe}^{+3}) / (\text{H}^{+})^3 = 10^6$$

$$\text{Fe}^{+3} = (\text{H}^+)^3 \times 10^6$$

Change units so that the concentrations are in lb mole/ft³:

$$\text{Fe}^{+3} = 2.56 \times 10^8 (\text{H}^+)^3 \quad \text{equation 1}$$

from this development it can be seen that the ferric ion will exist as ferric sulfate at high acid concentrations and as ferric hydroxide at low acid concentrations.

The ferric hydroxide is an insoluble precipitate. This is very important to the operation of the process, since iron is a contaminate in the electrolytic plant. The solution from the neutral leach goes to the electrolytic plant so the ferric ion concentration in that solution must be kept at a low value. Therefore, at the end of the neutral leach most of the iron must be in the ferric hydroxide form. This would be the case when the acid concentration is low, or the pH of the solution is above five.

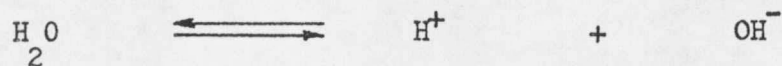
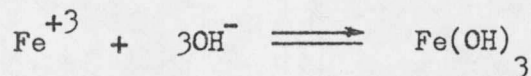
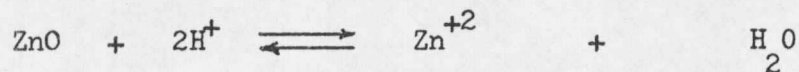
The solids from the neutral leach go to the acid leach where they are treated with cell acid from the electrolytic plant. The solids from the acid leach are washed and leave the plant, while the acid leach solution goes to the neutral leach. Since ferric sulfate is required as a buffer in the neutral leach the ferric ion must be in the sulfate form when it leaves the acid leach so that it can be recycled to the neutral leach. Any ferric ion present as ferric hydroxide is lost from the process and the hydrogen ion concentration must be high

enough at the end of the acid leach to keep the ferric ion in the sulfate form.

The development of the leach tank model was based on the assumptions:

1. The reaction rate depends on the acid concentration.
2. The reaction rate depends on the surface area of the ZnO.
3. The leach tanks are well mixed reactors.

The reactions used in the model development are:



The model development follows:

$$\text{moles ZnO reacted} = \frac{1}{2} \text{ moles H}^+ \text{ reacted} + \frac{3}{2} \text{ moles Fe}^{+3} \text{ reacted}$$

$$\text{moles ZnO reacted/unit time} = \frac{1}{2} \mathcal{V}_o (C_{A0} - C_A) + 3 (C_{F0} - C_F)$$

where

$\mathcal{V}_o$ = volumetric flow rate through the leach tank

C_{A0} = initial acid (H^+) concentration

C_A = acid (H^+) concentration leaving the reactor

C_{F0} = initial ferric ion concentration

C_F = ferric ion concentration leaving the reactor

also based on rate of reaction

$$\text{moles ZnO reacted/unit time} = K C_A V \frac{\text{surface area of ZnO}}{\text{ft}^3 \text{ of solution}}$$

$$= K C_A V \left(\frac{\text{number of ZnO particles}}{\text{ft}^3 \text{ of solution}} \right) \left(\frac{\text{area}}{\text{particle}} \right)$$

$$= K C_A V \left(\frac{\text{Wt. \% ZnO } \rho_s}{\rho \frac{4}{3} \pi r_o^3} \right) 4 \pi r^2$$

where

K = empirical constant

V = volume of leach tank

ρ_s = density of the slurry

ρ = density of ZnO

r_o = initial radius of particle

r = final radius of particle

Wt. \% ZnO = weight percent in incoming stream

Combining Constants

$$\text{moles ZnO reacted/unit time} = K C_A V \frac{\text{wt. \% ZnO } \rho_s r^2}{r_o^3}$$

Combining the two equations giving the rate of ZnO reaction

$$\frac{V_o}{2} (C_{Ao} - C_A + 3 (C_{Fo} - C_F)) r_o^3 = K C_A V \frac{\text{wt. \% ZnO } \rho_s r^2}{r_o^3}$$

$$\frac{(C_{ao} - C_A + 3 (C_{Fo} - C_F)) r_o^3}{2K C_A (\text{Wt. \% ZnO}) \rho_s} = \frac{V}{V_o} r^2$$

Assuming constant slurry density

$$\frac{(C_{Ao} - C_A + 3 (C_{Fo} - C_F)) r_o^3}{K C_A \text{ Wt. \% ZnO}} = \tau r^2$$

where τ is the holding time in the leach tank

$$r = \left(\frac{(C_{Ao} - C_A + 3 (C_{Fo} - C_F)) r_o^3}{K C_A \text{ Wt. \% ZnO } \tau} \right)^{\frac{1}{2}} \text{ equation 2}$$

The radius of a particle can also be calculated based on a material balance:

$$\frac{r^3}{r_o^3} = \frac{\text{ZnO in} - \text{ZnO reacted}}{\text{ZnO in}}$$

$$r = r_o \left(\frac{\text{ZnO in} - \text{ZnO reacted}}{\text{ZnO in}} \right)^{1/3}$$

$$r = r_o \left(\frac{N_o (\% \text{ ZnO in}) - (C_{Ao} - C_A + 3 (C_{Fo} - C_F))}{N_o (\% \text{ ZnO in})} \right)^{1/3} \text{ equation 3}$$

The development of the leach tank model left three equations (equations 1, 2, and 3) and three unknowns (C_A , C_F , r). The solution of these equations is a trial and error solution based on assuming a value for C_A , calculating C_F from equation 1, and then using equations 2 and 3 to get values for r . The proper value for C_A is obtained when

the two values for r obtained from equations 2 and 3 are equal to each other.

The first section of the subroutine calculates an average r_o to be used in the rest of the calculations. A straight arithmetic average did not give the proper results so another method of averaging was developed:

$$\begin{aligned} \text{surface area size 1} &= \frac{f_1 \text{ Wt. \% ZnO } \rho_s 4\pi r_1^2}{\rho 4/3\pi r_1^3} \\ &= \frac{3 f_1 \text{ Wt. \% ZnO } \rho_s}{\rho r_1} \end{aligned}$$

$$\text{surface area size 2} = \frac{3 f_2 \text{ Wt. \% ZnO } \rho_s}{\rho r_2}$$

$$\text{total surface area} = \frac{3 \text{ Wt. \% ZnO } \rho_s}{\rho r_o}$$

$$= \frac{3 f_1 \text{ Wt. \% ZnO } \rho_s}{\rho r_1} + \frac{3 f_2 \text{ Wt. \% ZnO } \rho_s}{\rho r_2}$$

$$\frac{1}{r_o} = \frac{f_1}{r_1} + \frac{f_2}{r_2}$$

$$r_o = \frac{1}{\frac{f_1}{r_1} + \frac{f_2}{r_2}}$$

This development can be extended to more size fractions and in the leach model the particle size distribution of the incoming solids is divided into ten equal parts.

The next section of the subroutine adjusts the values of the incoming acid and ferric ion concentrations so they fit equation 1. This is done using equation 1 and material balance calculations involving the acid and ferric ion concentrations and the amount of ferric hydroxide entering the leach tank. This section is required in case acid or ferric sulfate is added to the incoming stream. The addition of either of these components at any point ahead of a leach tank will disturb the acid - ferric ion balance. The leach tank is the only simulation model with the capability of adjusting that balance.

The next section searches for a minimum value for the acid concentration. In the acid leach there is more acid present than is required to leach all of the ZnO . If in the trial and error calculations of equations 1, 2, and 3 more acid is used in one of the trials, the model becomes unstable. Therefore a lower limit for the acid concentration must be established.

The next section in the model makes the actual trial and error calculations to obtain C_A , C_F , and r . This is done by using a golden section search for the value of C_A that yields the same value for r

from equations 2 and 3. Once this is done the remaining part of the subroutine calculates the amount of each component leaving the leach tank based on the material balance.

The equipment parameters for the leach tanks are specified in a vector of the form:

EQPJ = J, KC1, KC2, KC3, KC4, KC5, s, HTC, K, STC1, ...
STC_N, M*O, .

where

J = equipment node number

KC1 = component number of ZnO

KC2 = component number of H₂SO₄

KC3 = component number of Fe₂(SO₄)₃

KC4 = component number of Fe(OH)₃

KC5 = component number of H₂O

s = density of the slurry (lb/ft³)

HTC = holding time constant (the slurry capacity of the leach tanks in Tons)

K = leaching constant

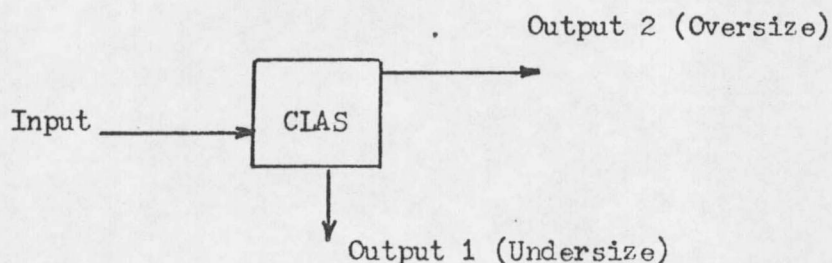
STC₁, ... STC_N = stoichiometric constants based on ZnO
(lb of each component produced per lb ZnO leached)

N = the total number of components (NOCOMP) minus 1; because the last component is always the gas phase and the gas phase is not

considered in the leach tank model

M*0 = balance of vector up to 25 elements, is filled out with
zeros

CLASSIFIER



This subroutine divides a solids stream based on a critical particle size input by the user. The subroutine searches the particle size distribution of the incoming stream until it finds the particle size corresponding to the critical size, it then assigns the material with smaller particle sizes to the undersize stream and the remaining material to the oversize stream.

If there is a liquid phase in the incoming stream, the subroutine assigns part of it to each of the outgoing streams according to the weight fraction solids, specified by the user, in the oversize stream.

The equipment parameter vector for this subroutine is:

$EQPJ = J, NS, CR, WFS, 21*0,$

where

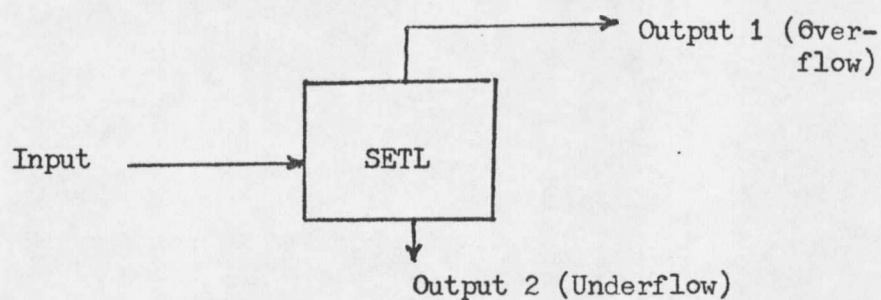
J = equipment node number

NS = number of solid components

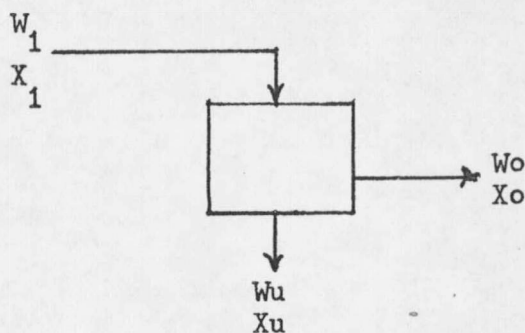
CR = critical radius (inches)

WFS = weight fraction solids in oversize stream

THICKENER



The equations used in this subroutine are based on a material balance around the thickener.



Total material balance

$$W_1 = W_o + W_u$$

Solids balance

$$W_1 X_1 = W_o X_o = W_u X_u$$

Solve for W_o

$$W_u = W_1 - W_o$$

$$W_1 X_1 = W_o X_o + W_1 X_u - W_o X_u$$

$$W_1 (X_1 - X_u) = W_o (X_o - X_u)$$

$$W_o = W_1 \frac{X_1 - X_u}{X_o - X_u}$$

where

W_1 = total weight input

X_1 = weight fraction solids input

W_o = total weight in overflow

X_o = weight fraction solids in overflow

W_u = total weight in underflow

X_u = weight fraction solids in underflow

The subroutine obtains values for W_1 and X_1 from the incoming stream data. The user must specify X_o and X_u . The subroutine then calculates values for W_o and W_u . Once W_o and W_u have been calculated, the weights of each component in each stream are calculated based on the fraction of incoming solid and liquid components going to each output stream.

The equipment parameter vector for a thickener has the form:

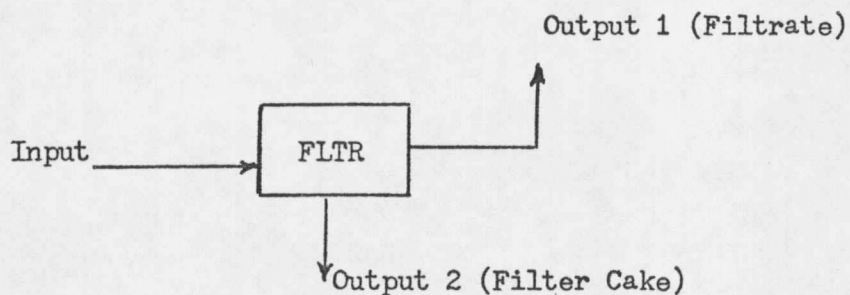
$$EQPJ = J, NS, X_o, X_u, 21 * 0$$

where

J = equipment node number

NS = number of solid components

FILTER



The filter subroutine uses the same equations as the thickener subroutine, with the overflow replaced by the filtrate and the underflow by the filtercake,

The equipment parameter vector is:

$$EQPJ = J, NS, X_F, X_C, 21 * 0,$$

where

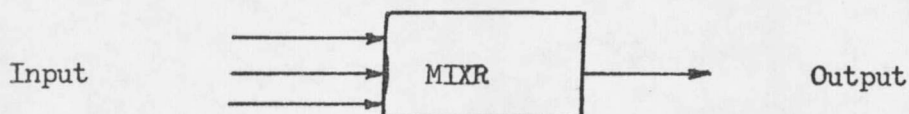
J = equipment node number

NS = number of solid components

X_F = weight fraction solids in the filtrate

X_C = weight fraction solids in the filter cake

MIXER



The mixer model will mix up to six incoming streams to form one output stream. The weight of each component in the output stream and the total weight of the output streams are calculated in the model by addition of the incoming values.

The particle size distribution equation of the output stream is obtained by ordering all of the incoming particle sizes from the smallest to the largest, and doing a least squares fit of the resulting distribution. The particle sizes are ordered by placing all of the incoming particle sizes in a matrix and then testing each particle size in the matrix to find the smallest one. Once the smallest particle size is found it is placed in the outgoing distribution, and the corresponding value matrix is arbitrarily set to a value of 10^6 , which is above all particle sizes. The process is repeated until all of the particle sizes are in the outgoing distribution.

The last section of the subroutine calculates the average conversion (ZnS to ZnO) for the outgoing stream by calculating the arithmetic average of the incoming conversions. The equations de-

veloped in the roaster model are then used to calculate the effective holding time in a roaster required to get the average conversion.

The equipment parameter vector for this subroutine is:

$EQU PJ = J, KC, NNN, 22*0,$

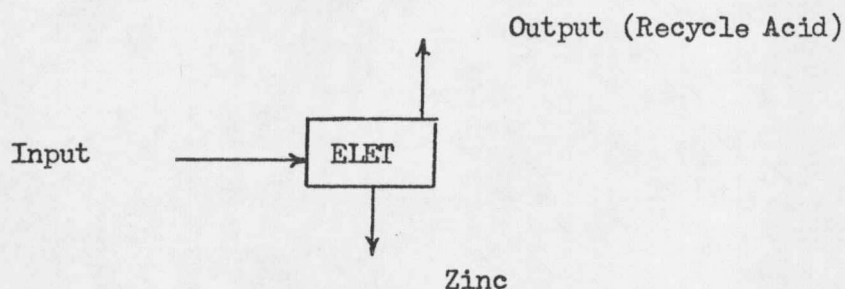
where

J = equipment node number

KC = component number of ZnS

NNN = number of solid components

ELECTROLYTIC PLANT



The electrolytic subroutine calculations are all user specified. The user specifies the fraction of incoming zinc sulfate converted to zinc, and the current efficiency. The subroutine uses the fraction of zinc sulfate converted to calculate the amount of zinc produced. From the amount of zinc produced, Faraday's constant, and the current efficiency the cell current is calculated. The final section in the subroutine makes stoichiometric calculations to determine the composition and amount of recycle acid produced.

The equipment parameter vector for this subroutine has the form:

EQPJ = J, KC1, EFF, CEFF, STC1, ... STC_N, M*O, .

where

J = equipment node number

KC1 = component number of ZnSO_4

EFF = fraction of incoming ZnSO_4 converted to Zn

CEFF = current efficiency

$STC_1 \dots STC_N$ = stoichiometric constants (lb of each component produced per lb of $ZnSO_4$ reacted)

N = number of components (NOCOMP) minus 1; because last component is always the gas phase

M*0 = balance of vector, up to 25 elements, is filled out with zeros

DATA INPUT FOR PLANT SIMULATION

For the purpose of this demonstration, the zinc plant was divided into two sections, the roasting section and the leaching section. Figure 7 shows the equipment sequence for the roasting section, and Figure 8 shows the equipment sequence for the leaching section of the zinc plant modeled. The first step in modeling each of the sections is to place the equipment subroutines used in a sequence that simulates the section modeled. This is done in Figures 9 and 10. The next step involves the preparation of the computer input and will be done step by step for the readers benefit.

The first card in the data deck is the "CLEAN" card. This card zeros all of the data storage blocks in the program. The word CLEAN, beginning in column one is the only thing on this card.

The title card follows the "CLEAN" card and can have anything on

it the programmer chooses that fits in the 80 columns. For the roasting section the title card used was:

ZINC PLANT SIMULATION ROASTING SECTION

The remaining cards of the input follow the NAMELIST format, with all input beginning in the second column of each card. The first section of input is the "PMLIST" input string. This block of data creates the equipment sequence and specifies the input and output streams of each piece of equipment. Each card has the following form:

KPM1 = 1, 'ROST', 'R-1', 1, -2, -3, 4*0,

This card specifies that a roaster signified by the user as R-1 is the first piece of equipment with stream number 1 as input and streams 2 and 3 as output. The remaining four data areas of KPM1 are filled with zeros. The second piece of equipment is a mixer specified as M-1 with input streams 3 and 17 and output stream 4. The card indicating this data has the form:

KPM 2 = 2, 'MIXR', 'M-1', 3, 17, -4, 4*0,

One data card of the above form is prepared for each piece of equipment simulated. The last data card of this block specifies the number of components and the component number of each component used. The roasting section used 6 components (component numbers 1 through 6) This data was specified:

NOCOMP = 6, COMPNT = 1,2,3,4,5,6,14*0,

The next section of input data (EQLIST) is the equipment parameters. This input depends on actual plant data and the data used in this simulation is shown in Table . Each piece of equipment has different input parameters so that each piece of input data in this section will be examined.

Equipment Node Number 1 - Roaster

EQPJ - J, KC, FG, P, T, RC, OC, CTC, A, BC, CC, STC_1 , ... STC_N ,

M*O

J = Equipment number = 1.

KC = Component number of ZnS = 1.

FG = Standard cubic feet/min gas leaving roaster = 65681.

P = Pressure = 1 atm.

T = Temperature = 1475° F

RC = Critical radius constant = $6.41E-08$

OC = Holding time constant for overhead fines = $3.E05$ cubic feet

CTC = Holding time constant for calcine = 86.0 tons

AC = agglomeration constant = .9825

BC = Agglomeration constant = 1.19

CC = Agglomeration constant = 1.017

STC_1 = Stoichiometric constant for component 1 (ZnS) = -1.

STC_2 = Stoichiometric constant for component 2 (ZnO) = .835

STC_3 = Stoichiometric constant for component 3 (FeS_2) = -.1148

STC_4 = Stoichiometric constant for component 4 (Fe_2O_3) = .0764

STC_5 = Stoichiometric constant for component 5 (inert material) = 0.

The remaining equipment parameters (9) are zero.

The component numbers were established by the programmer when he set up the problem. The gas rate, pressure, and temperature were obtained from data supplied by Anaconda, and other sources. (17, 21, 24)

The radius constant, holding time constants and agglomeration constants were determined empirically by trial and error to get the output desired. The stoichiometric constants give the pounds of the corresponding component produced/pound of ZnS reacted.

Equipment Node Number 2 - Mixer

$EQPJ = J, KC, NNN, 22*0.,$

$J =$ Equipment number = 2

$KC =$ Component number ZnS = 1.

$NNN =$ Number of solid components = 5.0

Equipment Node Number 3 - Screen

$EQPJ = J, KN, SS, STC_1, \dots, STC_N, M*0,$

$J =$ Equipment number = 3.

$KC =$ Component number of ZnS = 1.

$SS =$ Screen size (inches) = .0937

STC_1 through $STC_N =$ Stoichiometric constants same as in roaster.

Equipment Node Number 4 - Ballmill

EQPJ = J, RE, Kt, 22*0.,

J = Equipment number = 4.

RE = Minimum radius = 1.E-03

Kt = Grinding constant = 1.5E02

The minimum radius was determined by extending the lower end of the experimental particle size distribution curves until they asymptotically approached a minimum value. The grinding constant was empirically determined.

Equipment Node Number 5 - Mixer

Same as equipment node 2 with J changed to 5.

Equipment Node Number 6 - Flue

EQPJ = J, KC, T, K, STC₁, ... STC_N, M*O,

J = Equipment number = 6.

KC = Component number of ZnS = 1.

T = Temperature = 396° F

K = Precipitator constant = 5.E06

STC₁, ... STC_N = Stoichiometric constants = Same value as used in roaster.

Equipment Node Number 7 - Electrostatic Precipitator.

The flue and precipitator use the same model so the input has the same form. In this case only J was changed from 6 to 7 and the precipitator constant was raised from 8.E05 to 5.E06 to get a

higher efficiency.

Equipment Node Number 8 - Flue

This flue is the same as equipment node 6 but J has been changed from 6 to 8 and the precipitator constant has been raised from 8.E05 to 1.5E06 to get a higher efficiency.

Equipment Node Number 9 - Mixer

The equipment parameter vector for this mixer is identical to the other mixer vectors with J equal to 9.

Equipment Node Number 10 - Roaster

This roaster is a recycle roaster and is smaller than the first roaster, which is really a combination of several primary roasters. Some of the constants have been changed to account for the smaller size but the equipment parameter vector has the same form as equipment node 1.

The last data card of EQLIST is the ENAME card. This card is used to list the equipment nodes specified in PMLIST that have no equipment parameters. There is room for 50 numbers and in this case where all equipment nodes have equipment parameters it has the form:

ENAME = 50*0.,

SEXLIST is the next block of data, and is used to specify some of the stream properties for incoming, intermediate and outgoing streams. For this example only the properties of the roaster input

stream (Stream number 1) were known. The vector for this stream has the form:

SEXJ = J, A, B, C, 7*0, TW, WF, ... WF_N, M*0.,

J = Stream number = 1

A = Particle size distribution constant = 1.20496E-03

B = Particle size distribution constant = 1.12046

C = Particle size distribution constant = .949301

TW = Total weight stream = 350 Tons/day

WF₁ = Weight fraction ZnS = .784

WF₂ = Weight fraction ZnO = 0.

WF₃ = Weight fraction FeS₂ = .0899

WF₄ = Weight fraction Fe₂O₃ = 0.

WF₅ = Weight fraction inert material = .1261

The remaining 6 places are filled with zeros.

The final data card in the SEXLIST block is the SNAME card that contains a list of streams specified in FMLIST but not specified in SEXLIST. In this example there are 17 streams but only stream number 1 has its properties specified, so SNAME has the form:

SNAME = 2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,84*0,

The 84 zeros are required to fill out the 100 unit vector.

The remaining stream parameters are placed in the SINLIST data block. The stream parameter vector has the form:

SINJ = J, F, K, 7*0.,

J = Stream number

F = Stream flag; 0. = intermediate stream, 1. = feed stream,
2. = product stream

K = roasting constant of the ore being used.

In this example stream 1 is an input stream and the roasting constant for this ore, determined empirically, is .00096, so the vector has the form:

SIN1 = 1., 1., .00096, 7*0.,

There are two product streams, stream number 5 which is the calcine to the leach and stream number 12 which is the material lost up the stack.

These two streams are specified:

SIN5 = 5., 2., 8*0.,

SIN12 = 12., 2., 8*0.,

If no parameters need to be specified by the user for intermediate streams they can be listed in SNAME. In this case 1, 5 and 12 have been given parameters or are product streams, giving SNAME the form:

SNAME = 2,3,4,6,7,8,9,10,11,13,14,15,16,17,86*0,

The last section of input KELIST specifies recycle loops. For this problem the input can be put on one card in this manner:

KE2 = 2,3,4,5,6,7,8,9,10,41*0, LOOPS = 10, ERROR = 0.05,

KE4 = 5, 9*0,

This card tells the program that there is one recycle loop containing equipment nodes 2 through 10, the maximum number of calculations through the cycle is 10, with a maximum fractional error of .05 and that convergence promotion will be used on stream 5. If a secondary (25) recycle loop had been required it would have been specified in KE3.

The complete data input for the roasting section simulated is shown in Figure 46 and the computer output from this calculation is shown in Figure 47.

For the leaching section of the plant the data input follows the same format as that for the roasting section. The name of the problem was changed to: ZINC PLANT SIMULATION LEACHING SECTION, and the product stream from the roasting section became the feed stream for the leaching section. Only the equipment parameter vectors not discussed in the roasting section will be discussed here.

A complete data input is shown in Figure 48 and the results of this input are shown in Figure 49.

Equipment Node 1 Leach Slurry Tank

EQPJ = J, KC1, KC2, KC3, KC4, NS, AR, FR, WFS, 16*0,

J = Equipment Number = 1.

KC1 = Component number of ZnO = 2.

KC2 = Component number of H_2SO_4 = 7.

KC3 = Component number of $Fe_2(SO_4)_3$ = 8.

KC4 = Component number of $H_2O = 10$.

NS = Number of solid components = 6.

AR = Moles H_2SO_4 /mole ZnO = .0025

FR = Moles $Fe_2(SO_4)_3$ /moles ZnO = .2

WFS = Weight fraction solids = .2

The number given each component is established by the user at the beginning of the program. The acid to ZnO ratio, the $Fe_2(SO_4)_3$ to ZnO ratio and the weight fraction solids were adjusted to fit the data supplied by Anaconda for this problem.

There are several neutral and acid leach tanks in this section and only one will be shown. The one selected is Equipment node 2.

EQP2 = 2, KC1, KC2, KC3, KC4, KC5, HTC, K, $STC_1, \dots$

$STC_N, M^*O.,$

KC1 = Component number of ZnO = 2.

KC2 = Component number of $H_2SO_4 = 7$.

KC3 = Component number of $Fe_2(SO_4)_3 = 8$.

KC4 = Component number of $Fe(OH)_3 = 6$.

KC5 = Component number of $H_2O = 10$.

= Density of slurry = 112.3 lb/ft³

HTC = Holding time constant = 300. Tons

K = Leaching Constant = 6.

$STC_1, \dots, STC_N$ = Stoichiometric constants similar to

those used in the roasting section,

but in this case they are based on
ZnO in place of ZnS.

The leaching constant is empirically determined, and its value depends on the ore being processed.

Equipment node 10 Classifier

EQPJ = J, NS, CR, WFS, 21*0.,

J = Equipment number = 10

NS = Number of solid components = 6.

CR = Critical radius = 1.21E-03 inches

WFS = Weight fraction solids in oversize stream = .20

Equipment node 13 Thickener

EQPJ = J, NS, Xo, Xu, 21*0.,

J = Equipment number = 13.

NS = Number of solid components = 6.

Xo = Weight fraction solids in overflow = 0.0

Xu = Weight fraction solids in underflow = .60

Equipment Node 32 Filter

The same model was used for a filter as was used for a thickener.
The weight fraction solids in the filtrate replaces the weight fraction solids in the thickener overflow, and the weight fraction solids in the cake replaces the weight fraction solids in the

thickener underflow.

Equipment Node 34 Electrolytic Plant

The electrolytic plant is a part of the leach section since it recycles material back to the acid leach tanks.

$$EQPJ = J, KC1, EFF, CEFF, STC_1, \dots, STC_N, M*O.,$$

J = Equipment number = 34.

$KC1$ = Component number $ZnSO_4 = 9$.

EFF = Fraction of incoming $ZnSO_4$ converted to $Zn = .5$

$CEFF$ = Current efficiency = .9

$STC_1, \dots, STC_N$ = Stoichiometric constants based on $ZnSO_4$.

Executive Program Printouts

Figure 16

```

C
C   $$$ MAIN CALLING PROGRAM   $$$$$$$$$$$$$$$$$$$$$$$$
C
COMMON/SYSAA/ TITLE(20),COMPNT(20),KOMNAM(80)
COMMON/SYSB/  KE1(50),NE1MAX,KE2(50),NE2MAX,KE3(50),NE3MAX
COMMON/SYSB/  KE4(10),NE4MAX,KRET,KRET2,KRET3
COMMON/SYSC/  LIMIT,LIMIT2,LIMIT3,LOOP,LOOPS
COMMON/SYSD/  KEFLAG(50),KSFLAG(100),KTRACE,DERROR,NPFREQ,IPUNCH
COMMON/EQPB/  NECALL(50),NEXEQN(50),NAME(50)
COMMON/CONTL/NIN,NOUT,NOCOMP,NE,NEN
COMMON/PHD/  APC(20),ATC(20),AVC(20),AMW(20),AOMEG(20),ADEL(20),
X  AVW(20),APH(20),BET(20),GAM(20),DTA(20),EXFLAG
LOGICAL  EXFLAG
INTEGER  SNAME(100),ENAME(50),TITLE,COMPNT

```

```

C
C   FOLLOWING DUMMY ARRAYS ARE USED FOR 'NAMELIST' RIGID-INPUT OF
C   'KPM','EQPAR','SEXTSV',AND 'SINTSV' ARRAYS.
C

```

117

```

COMMON/SYSA/  KPM1(10),KPM2(10),KPM3(10),KPM4(10),KPM5(10),KPM6(10)
1,KPM7(10),KPM8(10),KPM9(10),KPM10(10),KPM11(10),KPM12(10),KPM13(10
2 ),KPM14(10),KPM15(10),KPM16(10),KPM17(10),KPM18(10),KPM19(10),
3 KPM20(10),KPM21(10),KPM22(10),KPM23(10),KPM24(10),KPM25(10),
4 KPM26(10),KPM27(10),KPM28(10),KPM29(10),KPM30(10),KPM31(10),
5 KPM32(10),KPM33(10),KPM34(10),KPM35(10),KPM36(10),KPM37(10),
6 KPM38(10),KPM39(10),KPM40(10),KPM41(10),KPM42(10),KPM43(10),
7 KPM44(10),KPM45(10),KPM46(10),KPM47(10),KPM48(10),KPM49(10),
8 KPM50(10),KSEM(3,100),N3MAX

```

```

C
COMMON/EQPA/  EQP1(25),EQP2(25),EQP3(25),EQP4(25),EQP5(25),EQP6(25)
1,EQP7(25),EQP8(25),EQP9(25),EQP10(25),EQP11(25),EQP12(25),EQP13(25
2 ),EQP14(25),EQP15(25),EQP16(25),EQP17(25),EQP18(25),EQP19(25),
3 EQP20(25),EQP21(25),EQP22(25),EQP23(25),EQP24(25),EQP25(25),
4 EQP26(25),EQP27(25),EQP28(25),EQP29(25),EQP30(25),EQP31(25),
5 EQP32(25),EQP33(25),EQP34(25),EQP35(25),EQP36(25),EQP37(25),

```

Figure 16 (contd.)

```
6 EQP38(25),EQP39(25),EQP40(25),EQP41(25),EQP42(25),EQP43(25),
7 EQP44(25),EQP45(25),EQP46(25),EQP47(25),EQP48(25),EQP49(25),
8 EQP50(25),NEMAX,MAXEQP
COMMON/STMA/ SEX1(23),SEX2(23),SEX3(23),SEX4(23),SEX5(23),SEX6(23)
1,SEX7(23),SEX8(23),SEX9(23),SEX10(23),SEX11(23),SEX12(23),
2 SEX13(23),SEX14(23),SEX15(23),SEX16(23),SEX17(23),SEX18(23),
3 SEX19(23),SEX20(23),SEX21(23),SEX22(23),SEX23(23),SEX24(23),
4 SEX25(23),SEX26(23),SEX27(23),SEX28(23),SEX29(23),SEX30(23),
5 SEX31(23),SEX32(23),SEX33(23),SEX34(23),SEX35(23),SEX36(23),
6 SEX37(23),SEX38(23),SEX39(23),SEX40(23),SEX41(23),SEX42(23),
7 SEX43(23),SEX44(23),SEX45(23),SEX46(23),SEX47(23),SEX48(23),
8 SEX49(23),SEX50(23),SEX51(23),SEX52(23),SEX53(23),SEX54(23),
9 SEX55(23),SEX56(23),SEX57(23),SEX58(23),SEX59(23),SEX60(23),
A SEX61(23),SEX62(23),SEX63(23),SEX64(23),SEX65(23),SEX66(23),
B SEX67(23),SEX68(23),SEX69(23),SEX70(23),SEX71(23),SEX72(23),
C SEX73(23),SEX74(23),SEX75(23),SEX76(23),SEX77(23),SEX78(23),
D SEX79(23),SEX80(23),SEX81(23),SEX82(23),SEX83(23),SEX84(23),
E SEX85(23),SEX86(23),SEX87(23),SEX88(23),SEX89(23),SEX90(23),
F SEX91(23),SEX92(23),SEX93(23),SEX94(23),SEX95(23),SEX96(23),
G SEX97(23),SEX98(23),SEX99(23),SEX100(23)
COMMON/STMA/ SIN1(10),SIN2(10),SIN3(10),SIN4(10),SIN5(10),SIN6(10)
1,SIN7(10),SIN8(10),SIN9(10),SIN10(10),SIN11(10),SIN12(10),
2 SIN13(10),SIN14(10),SIN15(10),SIN16(10),SIN17(10),SIN18(10),
3 SIN19(10),SIN20(10),SIN21(10),SIN22(10),SIN23(10),SIN24(10),
4 SIN25(10),SIN26(10),SIN27(10),SIN28(10),SIN29(10),SIN30(10),
5 SIN31(10),SIN32(10),SIN33(10),SIN34(10),SIN35(10),SIN36(10),
6 SIN37(10),SIN38(10),SIN39(10),SIN40(10),SIN41(10),SIN42(10),
7 SIN43(10),SIN44(10),SIN45(10),SIN46(10),SIN47(10),SIN48(10),
8 SIN49(10),SIN50(10),SIN51(10),SIN52(10),SIN53(10),SIN54(10),
9 SIN55(10),SIN56(10),SIN57(10),SIN58(10),SIN59(10),SIN60(10),
A SIN61(10),SIN62(10),SIN63(10),SIN64(10),SIN65(10),SIN66(10),
B SIN67(10),SIN68(10),SIN69(10),SIN70(10),SIN71(10),SIN72(10),
C SIN73(10),SIN74(10),SIN75(10),SIN76(10),SIN77(10),SIN78(10),
D SIN79(10),SIN80(10),SIN81(10),SIN82(10),SIN83(10),SIN84(10),
E SIN85(10),SIN86(10),SIN87(10),SIN88(10),SIN89(10),SIN90(10),
```

Figure 16 (contd.)

F SIN91(10),SIN92(10),SIN93(10),SIN94(10),SIN95(10),SIN96(10),
G SIN97(10),SIN98(10),SIN99(10),SIN100(10),NSMAX,MAXSEX,MAXSIN

C

DIMENSION KPM(10,50),EQPAR(25,50),SEXTSV(23,100),SINTSV(10,100)
EQUIVALENCE (KPM,KPM1),(EQPAR,EQP1),(SEXTSV,SEX1),(SINTSV,SIN1)

C

COMMON/STRMIN/SINUM(8,10),SICNST(10,8),SIMOLE(8),
1SICOMP(11,8),SIKV(10,8)
COMMON/STMOUT/SONUM(8,10),SOCNST(10,8),SOMOLE(8),
1SOCOMP(11,8),SOKV(10,8)

C

15 CALL DREAD
CALL DCHECK(&15)
CALL DPRINT
20 CALL SUBSET
30 CALL BIPRNT
GO TO 15
END

Figure 17

```

SUBROUTINE DREAD
C... TYPE 1 SHOULD CONTAIN WORD 'CLEAN' IN COL. 1-5.
C... ALL OTHERS ARE TREATED AS TYPE 2 AND DO NOT REQUIRE /PMLIST/
COMMON/SYSA/ TITLE(20),COMPNT(20),KOMNAM(80)
COMMON/SYSB/ KE1(50),NE1MAX,KE2(50),NE2MAX,KE3(50),NE3MAX
COMMON/SYSB/ KE4(10),NE4MAX,KRET,KRET2,KRET3
COMMON/SYSC/ LIMIT,LIMIT2,LIMIT3,LOOP,LOOPS
COMMON/SYSD/ KEFLAG(50),KSFLAG(100),KTRACE,DERROR,NPFREQ,IPUNCH
COMMON/EQPB/ NECALL(50),NEXEQN(50),NAME(50)
COMMON/CONTL/NIN,NOUT,NOCOMP,NE,NEN
COMMON/PHD/ APC(20),ATC(20),AVC(20),AMW(20),AOMEG(20),ADEL(20),
X AVW(20),APH(20),BET(20),GAM(20),DTA(20),EXFLAG
LOGICAL EXFLAG
INTEGER SNAME(100),ENAME(50),TITLE,COMPNT
C AVAILABLE '30' EQUIPMENT SUBROUTINE NAMES
INTEGER NMLIST(30)/'ROST','SCRN','MIXR','GRND','PREP','LECH','FLUE
1 ','FLTR','SETL','CLAS','RLTS','ELET','AMIX','ADD3','ADD4','ADD5',
2 'ADD6','ADD7','ADD8','ADD9','AD10','AD11','AD12','AD13','AD14',
3 'AD15','AD16','AD17','AD18','AD19'/
C
C FOLLOWING DUMMY ARRAYS ARE USED FOR 'NAMELIST' RIGID=INPUT OF
C 'KPM','EQPAR','SEXTSV',AND 'SINTSV' ARRAYS.
C
COMMON/SYSA/ KPM1(10),KPM2(10),KPM3(10),KPM4(10),KPM5(10),KPM6(10)
1,KPM7(10),KPM8(10),KPM9(10),KPM10(10),KPM11(10),KPM12(10),KPM13(10)
2 ),KPM14(10),KPM15(10),KPM16(10),KPM17(10),KPM18(10),KPM19(10),
3 KPM20(10),KPM21(10),KPM22(10),KPM23(10),KPM24(10),KPM25(10),
4 KPM26(10),KPM27(10),KPM28(10),KPM29(10),KPM30(10),KPM31(10),
5 KPM32(10),KPM33(10),KPM34(10),KPM35(10),KPM36(10),KPM37(10),
6 KPM38(10),KPM39(10),KPM40(10),KPM41(10),KPM42(10),KPM43(10),
7 KPM44(10),KPM45(10),KPM46(10),KPM47(10),KPM48(10),KPM49(10),
8 KPM50(10),KSEM(3,100),N3MAX
C
COMMON/EQPA/ EQP1(25),EQP2(25),EQP3(25),EQP4(25),EQP5(25),EQP6(25)
1,EQP7(25),EQP8(25),EQP9(25),EQP10(25),EQP11(25),EQP12(25),EQP13(25)

```

Figure 17 (contd.)

2),EQP14(25),EQP15(25),EQP16(25),EQP17(25),EQP18(25),EQP19(25),
 3 EQP20(25),EQP21(25),EQP22(25),EQP23(25),EQP24(25),EQP25(25),
 4 EQP26(25),EQP27(25),EQP28(25),EQP29(25),EQP30(25),EQP31(25),
 5 EQP32(25),EQP33(25),EQP34(25),EQP35(25),EQP36(25),EQP37(25),
 6 EQP38(25),EQP39(25),EQP40(25),EQP41(25),EQP42(25),EQP43(25),
 7 EQP44(25),EQP45(25),EQP46(25),EQP47(25),EQP48(25),EQP49(25),
 8 EQP50(25),NEMAX,MAXEQP

COMMON/STMA/ SEX1(23),SEX2(23),SEX3(23),SEX4(23),SEX5(23),SEX6(23)
 1,SEX7(23),SEX8(23),SEX9(23),SEX10(23),SEX11(23),SEX12(23),
 2 SEX13(23),SEX14(23),SEX15(23),SEX16(23),SEX17(23),SEX18(23),
 3 SEX19(23),SEX20(23),SEX21(23),SEX22(23),SEX23(23),SEX24(23),
 4 SEX25(23),SEX26(23),SEX27(23),SEX28(23),SEX29(23),SEX30(23),
 5 SEX31(23),SEX32(23),SEX33(23),SEX34(23),SEX35(23),SEX36(23),
 6 SEX37(23),SEX38(23),SEX39(23),SEX40(23),SEX41(23),SEX42(23),
 7 SEX43(23),SEX44(23),SEX45(23),SEX46(23),SEX47(23),SEX48(23),
 8 SEX49(23),SEX50(23),SEX51(23),SEX52(23),SEX53(23),SEX54(23),
 9 SEX55(23),SEX56(23),SEX57(23),SEX58(23),SEX59(23),SEX60(23),
 A SEX61(23),SEX62(23),SEX63(23),SEX64(23),SEX65(23),SEX66(23),
 B SEX67(23),SEX68(23),SEX69(23),SEX70(23),SEX71(23),SEX72(23),
 C SEX73(23),SEX74(23),SEX75(23),SEX76(23),SEX77(23),SEX78(23),
 D SEX79(23),SEX80(23),SEX81(23),SEX82(23),SEX83(23),SEX84(23),
 E SEX85(23),SEX86(23),SEX87(23),SEX88(23),SEX89(23),SEX90(23),
 F SEX91(23),SEX92(23),SEX93(23),SEX94(23),SEX95(23),SEX96(23),
 G SEX97(23),SEX98(23),SEX99(23),SEX100(23)

COMMON/STMA/ SIN1(10),SIN2(10),SIN3(10),SIN4(10),SIN5(10),SIN6(10)
 1,SIN7(10),SIN8(10),SIN9(10),SIN10(10),SIN11(10),SIN12(10),
 2 SIN13(10),SIN14(10),SIN15(10),SIN16(10),SIN17(10),SIN18(10),
 3 SIN19(10),SIN20(10),SIN21(10),SIN22(10),SIN23(10),SIN24(10),
 4 SIN25(10),SIN26(10),SIN27(10),SIN28(10),SIN29(10),SIN30(10),
 5 SIN31(10),SIN32(10),SIN33(10),SIN34(10),SIN35(10),SIN36(10),
 6 SIN37(10),SIN38(10),SIN39(10),SIN40(10),SIN41(10),SIN42(10),
 7 SIN43(10),SIN44(10),SIN45(10),SIN46(10),SIN47(10),SIN48(10),
 8 SIN49(10),SIN50(10),SIN51(10),SIN52(10),SIN53(10),SIN54(10),
 9 SIN55(10),SIN56(10),SIN57(10),SIN58(10),SIN59(10),SIN60(10),

Figure 17 (contd.)

```
A SIN61(10),SIN62(10),SIN63(10),SIN64(10),SIN65(10),SIN66(10),  
B SIN67(10),SIN68(10),SIN69(10),SIN70(10),SIN71(10),SIN72(10),  
C SIN73(10),SIN74(10),SIN75(10),SIN76(10),SIN77(10),SIN78(10),  
D SIN79(10),SIN80(10),SIN81(10),SIN82(10),SIN83(10),SIN84(10),  
E SIN85(10),SIN86(10),SIN87(10),SIN88(10),SIN89(10),SIN90(10),  
F SIN91(10),SIN92(10),SIN93(10),SIN94(10),SIN95(10),SIN96(10),  
G SIN97(10),SIN98(10),SIN99(10),SIN100(10),NSMAX,MAXSEX,MAXSIN
```

```
C  
DIMENSION KPM(10,50),EQPAR(25,50),SEXTSV(23,100),SINTSV(10,100)  
EQUIVALENCE (KPM,KPM1),(EQPAR,EQP1),(SEXTSV,SEX1),(SINTSV,SIN1)
```

```
C  
NAMELIST/PMLIST/ KPM1 ,KPM2 ,KPM3 ,KPM4 ,KPM5 ,KPM6 ,KPM7 ,KPM8 ,  
1KPM9 ,KPM10,KPM11,KPM12,KPM13,KPM14,KPM15,KPM16,KPM17,KPM18,KPM19,  
2KPM20,KPM21,KPM22,KPM23,KPM24,KPM25,KPM26,KPM27,KPM28,KPM29,KPM30,  
3KPM31,KPM32,KPM33,KPM34,KPM35,KPM36,KPM37,KPM38,KPM39,KPM40,KPM41,  
4KPM42,KPM43,KPM44,KPM45,KPM46,KPM47,KPM48,KPM49,KPM50  
5 NOCOMP,COMPNT,KOMNAM,IPUNCH
```

```
C  
NAMELIST/EQLIST/ ENAME,EQP1,EQP2,EQP3,EQP4,EQP5,EQP6,EQP7,EQP8,  
1 EQP9,EQP10,EQP11,EQP12,EQP13,EQP14,EQP15,EQP16,EQP17,EQP18,EQP19,  
2EQP20,EQP21,EQP22,EQP23,EQP24,EQP25,EQP26,EQP27,EQP28,EQP29,EQP30,  
3EQP31,EQP32,EQP33,EQP34,EQP35,EQP36,EQP37,EQP38,EQP39,EQP40,EQP41,  
4EQP42,EQP43,EQP44,EQP45,EQP46,EQP47,EQP48,EQP49,EQP50
```

```
C  
NAMELIST/SEXLIST/ SNAME,SEX1,SEX2,SEX3,SEX4,SEX5,SEX6,SEX7,SEX8,  
1 SEX9 ,SEX10,SEX11,SEX12,SEX13,SEX14,SEX15,SEX16,SEX17,SEX18,SEX19  
2,SEX20,SEX21,SEX22,SEX23,SEX24,SEX25,SEX26,SEX27,SEX28,SEX29,SEX30  
3,SEX31,SEX32,SEX33,SEX34,SEX35,SEX36,SEX37,SEX38,SEX39,SEX40,SEX41  
4,SEX42,SEX43,SEX44,SEX45,SEX46,SEX47,SEX48,SEX49,SEX50,SEX51,SEX52  
5,SEX53,SEX54,SEX55,SEX56,SEX57,SEX58,SEX59,SEX60,SEX61,SEX62,SEX63  
6,SEX64,SEX65,SEX66,SEX67,SEX68,SEX69,SEX70,SEX71,SEX72,SEX73,SEX74  
7,SEX75,SEX76,SEX77,SEX78,SEX79,SEX80,SEX81,SEX82,SEX83,SEX84,SEX85  
8,SEX86,SEX87,SEX88,SEX89,SEX90,SEX91,SEX92,SEX93,SEX94,SEX95,SEX96  
9,SEX97,SEX98,SEX99,SEX100
```

Figure 17 (contd.)

```

      NAMELIST/SINLIST/ SNAME,SIN1,SIN2,SIN3,SIN4,SIN5,SIN6,SIN7,SIN8,
1 SIN9 ,SIN10,SIN11,SIN12,SIN13,SIN14,SIN15,SIN16,SIN17,SIN18,SIN19
2,SIN20,SIN21,SIN22,SIN23,SIN24,SIN25,SIN26,SIN27,SIN28,SIN29,SIN30
3,SIN31,SIN32,SIN33,SIN34,SIN35,SIN36,SIN37,SIN38,SIN39,SIN40,SIN41
4,SIN42,SIN43,SIN44,SIN45,SIN46,SIN47,SIN48,SIN49,SIN50,SIN51,SIN52
5,SIN53,SIN54,SIN55,SIN56,SIN57,SIN58,SIN59,SIN60,SIN61,SIN62,SIN63
6,SIN64,SIN65,SIN66,SIN67,SIN68,SIN69,SIN70,SIN71,SIN72,SIN73,SIN74
7,SIN75,SIN76,SIN77,SIN78,SIN79,SIN80,SIN81,SIN82,SIN83,SIN84,SIN85
8,SIN86,SIN87,SIN88,SIN89,SIN90,SIN91,SIN92,SIN93,SIN94,SIN95,SIN96
9,SIN97,SIN98,SIN99,SIN100

C
      NAMELIST/KELIST/ KE2,KE3,KE4,LOOPS,NPFREQ,KTRACE,DERROR
C
C... READ ALPHANUMERIC HEADER CARD '20A4' FIELD ('TITLE(20)')
C
17 FORMAT(20A4)
3 ITYPE = 2
657 FORMAT (1H1)
      WRITE (6,657)
      READ(5,5,END=200)LCARD1,LCARD2
      WRITE(6,995)LCARD1,LCARD2
      INTEGER ICL(2)/'CLEAN'/
      IF(LCARD1.NE.ICL(1)) GO TO 23
      IF(LCARD2.NE.ICL(2)) GO TO 23
5 FORMAT(2A4)
995 FORMAT (2X,2A4)
      CALL CLEAN
      ITYPE =1
C... 'CLEAN'          = TYPE 1
C... OTHERS          = TYPE 2
23 READ( 5,17) TITLE
      WRITE( 6,17) TITLE
      IF(ITYPE.NE.1) GO TO 63

```

Figure 17 (contd.)

```
C
C**READ REMAINING CONTROL CONSTANTS, COMPONENT ID, AND PROCESS MATRIX
C** FROM NAMLIST/PMLIST/
C
      READ( 5,PMLIST)
      MAXSEX=NOCOMP+12
      MAXSIN=10
      MAXEQP=25
C
C SET UP EQUIPMENT SUBROUTINE CALLING NUMBER FROM EQUIPMENT NAME
C
      KVOID=0
51 DO 56 K2=1,50
      IF(KPM(1,K2).EQ.0) GO TO 56
      NEX=KPM(1,K2)
      NEMAX=K2
C
C NEX IS EQUIPMENT NUMBER OF PROCESS MATRIX ROW 1= KPM(1,K2)
C KPM(3,K2) IS THE EXTERNAL NAME GIVEN TO THIS UNIT.
C
      I=1
53 IF(KPM(2,K2).EQ.NMLIST(I)) GO TO 55
      IF(I.GT.30) GO TO 54
      I=I+1
      GO TO 53
54 WRITE( 6,108) NEX,KPM(2,K2),TITLE
108 FORMAT('1IN 'DREAD' EQUIPMENT NUMBER ',I3,' HAS BEEN GIVEN T
      1HE ILLEGAL NAME ',A4,' '//' ' THIS DATA IS NOT BEING CALCULATED
      2.'// ' PROBLEM ',I20A4, ' HAS BEEN VOIDED.')
      KVOID=1
55 NECALL(NEX)=I
      NAME(NEX)=KPM(2,K2)
      NEXEQN(NEX)=KPM(3,K2)
56 CONTINUE
```

Figure 17 (contd.)

```
C
C IF THE NUMBER OF EQUIPMENT SUBROUTINE NAMES OR AN EQUIPMENT SUBROU-
C TIME NAME ITSELF IS CHANGED, SUBROUTINE 'EQCALL' MUST BE MODIFIED.
C
59 CONTINUE
   DO 62 J=1,NEMAX
     IF(KPM(1,J).EQ.0) GO TO 62
     DO 61 I=2,8
61    KPM(I,J)=KPM(I+2,J)
62 CONTINUE
63 CALL ZEROX(ENAME,50)
C
C..READ 'EQPAR'(BY USING EQP1=EQP50) AND 'ENAME'.
   READ( 5,EQLIST)
   DO 65 I=1,NEMAX
     J=ENAME(I)
     IF(J.EQ.0) GO TO 65
     IF(ABS(EQPAR(1,J)).LT.1.E=20) EQPAR(1,J)=J
C   WRITE( 6,66) J,(EQPAR(K,J),K=1,25)
65 CONTINUE
C 66 FORMAT(' EQPAR(1,',I2,')=',10G10.2/(13X,10G10.2))
   CALL ZEROX(KEFLAG,50)
C
C..READ 'SEXTSV'(SEX1 =SEX100) AND 'SNAME'
C
   CALL ZEROX(SNAME,100)
C
   READ( 5,SEXLIST)
   DO 68 I=1,100
     J=SNAME(I)
     IF(J.EQ.0) GO TO 68
     IF(ABS(SEXTSV(1,J)).LT.1.E=20) SEXTSV(1,J)=J
C   WRITE( 6,69) J,(SEXTSV(K,J),K=1,23)
68 CONTINUE
```

Figure 17 (contd.)

```

C 69 FORMAT(' SEXTSV(1,'I2,')= ',2F3.0,11G10.3/30X,10G10.3 )
      DO 70 I=1,100
        IF (ABS(SEXTSV(1,I)) .GT. 1.E=20) NSMAX=I
      70 CONTINUE
C
C..READ 'SINTSV' ( SIN1=SIN100) AND 'SNAME'
C
      CALL ZEROX(SNAME,100)
      READ( 5,SINLIST)
      DO 72 I=1,NSMAX
        J=SNAME(I)
        IF (J.EQ.0) GO TO 72
        IF (ABS(SINTSV(1,J)) .LT. 1.E=20) SINTSV(1,J)=J
C      WRITE( 6,73) J,(SINTSV(K,J),K=1,10)
      72 CONTINUE
C 73 FORMAT(' SINTSV(1,'I2,')= ',F3.0,F4.0, 8G10.3)
      CALL ZEROX(KSFLAG,100)
C
C..READ EQUIPMENT LISTS FOR RECYCLE CALCULATIONS, KE2, KE3 AND KE4
C.. THEIR CONTROL CONSTANTS NE2MAX, NE3MAX AND NE4MAX. WILL BE CALC.
      75 READ( 5,KELIST)
C      WRITE( 6,KELIST)
      NE2MAX=0
      NE3MAX=0
      NE4MAX=0
      DO 76 I=1,50
        IF (KE2(I).EQ.0) GO TO 77
      76 NE2MAX=NE2MAX+1
      77 DO 78 I=1,50
        IF (KE3(I).EQ.0) GO TO 79
      78 NE3MAX=NE3MAX+1
      79 DO 80 I=1,10
        IF (KE4(I).EQ.0) GO TO 81
      80 NE4MAX=NE4MAX+1

```

Figure 17 (contd.)

```
C
  81 IF(KVOID.GT.0) GO TO 3
    IF(ITYPE.NE.1) RETURN
C
  N3MAX=8
  DO 83 K=2,8
    I=10-K
    DO 82 J=1,NEMAX
      IF(KPM(I,J).NE.0) GO TO 85
    82 CONTINUE
  83 N3MAX=N3MAX+1
  85 CONTINUE
C
C  BUILD UP THE STREAM CONNECTION MATRIX KSEM.
  DO 187 M1=1,NEMAX
    IF(KPM(1,M1).LE.0) GO TO 187
  188 M2=KPM(1,M1)
    DO 189 M3=2,N3MAX
      IF(KPM(M3,M1)) 191,187,190
  190 M4=(KPM(M3,M1))
      KSEM(3,M4)=M2
      GO TO 192
  191 M4=-(KPM(M3,M1))
      KSEM(2,M4)=M2
  192 KSEM(1,M4)=M4
  189 CONTINUE
  187 CONTINUE
    RETURN
  200 STOP
C  .....PUNCH OUT DATA BLOCK TO RESUBMIT THE CASE STUDY.....
  ENTRY DPUNCH
  RETURN
END
```

Figure 18

SUBROUTINE CLEAN

MAN

```

C
C.. THIS SUBROUTINE ZEROS FOLLOWING COMMON STORAGES BY CALLING SUBROUTINE
C.. 'ZEROX(IARRAY,ISIZE)'.
C.. COMMON /SYSA//SYSB//SYSD//EQPA//EQPB//STMA//STRMIN/ AND
C.. /STMOUT/ + /SYSAA/

```

```

C
COMMON/SYSA/ TITLE(20),COMPNT(20),KOMMAM(80)
COMMON/SYSA/ KPM(10,50),KSEM(3,100),N3MAX
COMMON/SYSB/ KE1(50),NE1MAX,KE2(50),NE2MAX,KE3(50),NE3MAX
COMMON/SYSB/ KE4(10),NE4MAX,KRET,KRET2,KRET3
COMMON/SYSD/ KEFLAG(50),KSFLAG(100),KTRACE,DERROR,NPFREQ,IPUNCH
COMMON/EQPA/ EQPAR(25,50),NEMAX,MAXEGP
COMMON/EQPB/ NECALL(50),NEXEQN(50),NAME(50)
COMMON/HXMS/ NESAVE(20)
COMMON/STMA/ SEXTSV(23,100),SINTSV(10,100),NSMAX,MAXSEX,MAXSIN
COMMON/STRMIN/SINUM(8,10),SICNST(10,8),SIMOLE(8),
1SICOMP(11,8),SIKV(10,8)
COMMON/STMOUT/SONUM(8,10),SOCNST(10,8),SOMOLE(8),
1SOCOMP(11,8),SOKV(10,8)
-----
DIMENSION ISYSA(801),ISYSB(167),ISYSD(154),IEQPA(1252),IEGPB(150)
DIMENSION ISTMA(3303),ISTIN(408),ISTOT(408),ISYSAA(120)
EQUIVALENCE (ISYSA,KPM),(ISYSB,KE1),(ISYSD,KEFLAG),(IEQPA,EQPAR),
1 (IEGPB,NECALL),(ISTMA,SEXTSV),(ISTIN,SINUM),(ISTOT,SONUM),
2 (ISYSAA,TITLE)

```

128

```

C
CALL ZEROX(ISYSAA,120)
CALL ZEROX(ISYSA,801)
CALL ZEROX(ISYSB,167)
CALL ZEROX(ISYSD,154)
CALL ZEROX(IEQPA,1252)
CALL ZEROX(IEGPB,150)
CALL ZEROX(ISTMA,3303)
CALL ZEROX(ISTIN,408)

```

Figure 18 (contd.)

```
CALL ZEROX(ISTOT,408)  
CALL ZEROX(NESAVE,20)
```

C

```
RETURN  
END
```

Figure 19

```
SUBROUTINE ZERO (A,N)
  DIMENSION A(N)
  DO 3 I=1,N
3  A(I)=0
  RETURN
  ENTRY ZEROX(K,N)
  DIMENSION K(N)
  DO 1 I=1,N
1  K (I)=0
  RETURN
END
```

Figure 20

```

SUBROUTINE DCHECK( *)
COMMON/SYSA/ KPM(10,50),KSEM(3,100),NSMAX
COMMON/STMA/ SEXTSV(23,100),SINTSV(10,100),NSMAX,MAXSEX,MAXSIN
C RETURN 1 HAS BEEN USED
C
LOGICAL FLAG
FLAG=.FALSE.
J=0
DO 10 I=1,NSMAX
IF(KSEM(1,I).EQ.0) GO TO 10
IF(KSEM(2,I).EQ.0.AND.KSEM(3,I).EQ.0) GO TO 2
IF(KSEM(2,I).EQ.0.AND..NOT.(ABS(SINTSV(2,I)=1.).LT.0.001.OR.
1 ABS(SINTSV(2,I)=3.).LT.0.001)) GO TO 4
IF(KSEM(3,I).EQ.0.AND.ABS(SINTSV(2,I)=2.).GT.0.001) GO TO 6
10 CONTINUE
GO TO 8
2 WRITE( 6,22) I
3 FLAG=.TRUE.
GO TO 10
4 WRITE( 6,24) I
GO TO 3
6 WRITE( 6,26) I
GO TO 3
8 IF(.NOT.FLAG) RETURN
WRITE( 6,28)
RETURN 1
22 FORMAT('O STREAM NO.',I4,' HAS NO CONNECTING EQUIPMENTS.')
```

MAN

131

```

24 FORMAT('O STREAM NO.',I4,' APPEARS AS A FEED STREAM IN THE PROCESS
1 MATRIX BUT IS NOT CODED AS SUCH IN THE STREAM VARIABLES MATRIX.')
```

26 FORMAT('O STREAM NO.',I4,' APPEARS IN THE PROCESS MATRIX AS A PROD

```

1UCT STREAM BUT IS NOT CODED AS SUCH IN THE STREAM VARIABLES MATRIX
2.')
```

28 FORMAT('O'///'***** COMPUTATION FOR THIS DATA SET WILL BE TERMINAT

```

1ED.')
```

END

Figure 21

```

SUBROUTINE DPRINT
COMMON/SYSAA/ TITLE(20),COMPNT(20),KOMNAM(80)
COMMON/SYSA/ KPM(10,50),KSEM(3,100),N3MAX
COMMON/SYSB/ KE1(50),NE1MAX,KE2(50),NE2MAX,KE3(50),NE3MAX
COMMON/SYSB/ KE4(10),NE4MAX,KRET,KRET2,KRET3
COMMON/SYSC/ LIMIT,LIMIT2,LIMIT3,LOOP,LOOPS
COMMON/SYSD/ KEFLAG(50),KSFLAG(100),KTRACE,DERROR,NPFREQ,IPUNCH
COMMON/EQPA/ EQPAR(25,50),NEMAX,MAXEQP
COMMON/EQPB/ NECALL(50),NEXEQN(50),NAME(50)
COMMON/STMA/ SEXTSV(23,100),SINTSV(10,100),NSMAX,MAXSEX,MAXSIN
COMMON/CONTL/NIN,NOUT,NOCOMP,NE,NEN
INTEGER COMPNT,TITLE

C
18 WRITE(6,19)
19 FORMAT('1'////2(5X,50('.'))//5X,18('.'),' C H E S S ',19('.'))
1 2(5X,50('.'))//5X,'...',T53,'...'/5X,'...',T20,'UNIVERSITY OF HOU
2STON',T53,'...'/5X,'...',T53,'...'/5X,'... CHEMICAL ENGINEERING
3SIMULATION SYSTEM ...'/5X,'...',T53,'...'/2(5X,50('.'))//5X,
4 18('.'),' C H E S S ',19('.')/2(5X,50('.'))

C
C**PRINT PROCESS MATRIX HEADING, THEN PROCESS MATRIX
WRITE( 6,22) TITLE
22 FORMAT('1',20A4//18X,'PROCESS VECTORS'//' ..... EQUIPMENT ...
1...',13X,'STREAM NUMBERS'//' NUMBER SUBROUTINE NAME' )
DO 24 I=1,NEMAX
IF(KPM(1, I).LE.0) GO TO 24
WRITE( 6,26) KPM(1,I),NAME(I),NEXEQN(I),(KPM(J,I),J=2,N3MAX)
24 CONTINUE
26 FORMAT(/15,7X,A4,5X,A4,6X,9I5)

C
C**WRITE STREAM AND EQUIPMENTS CONNECTION
WRITE( 6,27) TITLE
27 FORMAT('1',20A4//23X,'STREAM CONNECTIONS'//22X,'STREAM',4X,'EQ
1UIPMENT'/30X,'FROM TO' )
DO 28 I=1,NSMAX

```

132

Figure 21 (contd.)

```
      IF(KSEM(1,I).EQ.0) GO TO 28
      WRITE( 6,29) (KSEM(J,I),J=1,3)
28  CONTINUE
29  FORMAT(/20X,I6,2I7)
      WRITE(6,30) TITLE,NOCOMP,NE2MAX,NE3MAX,(COMPNT(I),I=1,NOCOMP)
30  FORMAT('1',2CA4//12X,'''OTHER SYSTEM VARIABLES'''//
      1' NUMBER OF COMPONENTS',T45,I4//1 NUMBER OF ITEMS IN RECYCLE LIST(
      2KE2)',T45,I4//1 NUMBER OF ITEMS IN RECYCLE LIST(KE3)',T45,I4//
      3' COMPONENT NUMBERS USED',T42,I2,9(1H,,I2)/(T41,10(1H,,I2)))
C
      I=2
      IF(NE2MAX.NE.0) WRITE( 6,40) I,(KE2(J),J=1,NE2MAX)
      I=3
      IF(NE3MAX.NE.0) WRITE( 6,40) I,(KE3(J),J=1,NE3MAX)
      IF(NE4MAX.NE.0) WRITE( 6,41) (KE4(J),J=1,NE4MAX)
41  FORMAT('0STREAMS USED IN CONV. ROUTINE(KE4)',T42,I2,9(1H,,I2)/(T41
      1,10(1H,,I2)))
40  FORMAT('0RECYCLE LIST KE',I1,T42,I2,9(1H,,I2)/(T41,10(1H,,I2)))
C
      WRITE( 6,42) DERROR, LOOPS
42  FORMAT('0TOLERANCE, ''DERROR''',18X,F10.4//1 MAX. LOOPS IN RECYCLE
      1 CALC.',T45,I4)
      RETURN
      END
```

Figure 22

```

SUBROUTINE SUBSET
COMMON/SYSAA/ TITLE(20),COMPNT(20),KOMNAM(80)
COMMON/SYSA/ KPM(10,50),KSEM(3,100),N3MAX
COMMON/SYSB/ KE1(50),NE1MAX,KE2(50),NE2MAX,KE3(50),NE3MAX
COMMON/SYSB/ KE4(10),NE4MAX,KRET,KRET2,KRET3
COMMON/SYSC/ LIMIT,LIMIT2,LIMIT3,LOOP,LOOPS
COMMON/SYSD/ KEFLAG(50),KSFLAG(100),KTRACE,DERROR,NPFREQ,IPUNCH
COMMON/EQPA/ EQPAR(25,50),NEMAX,MAXEQP
COMMON/EQPB/ NECALL(50),NEXEQN(50),NAME(50)
COMMON/STMA/ SEXTSV(23,100),SINTSV(10,100),NSMAX,MAXSEX,MAXSIN
COMMON/CONTL/NIN,NOUT,NOCOMP,NE,NEN

C
C   NAMELIST/FLLIST/ KEFLAG,KSFLAG
C   IF(IPUNCH.GE.1) GO TO 61.
C   KRET2=0
C   SAVE2 =NE2MAX
C   SAVE3 =NE3MAX
C
C   FEED STREAM FLAGS ARE (+1). UNUSED EQUIP. AND STREAM ARE (=1),
C   UNKNOWN STREAMS ARE (0) AS ARE EQUIP. NOT COMPUTED.
C   KES IS '+1' IF ANY EQUIPMENT IS CALCULATED IN KPM SCAN IN SUBSET.
C   CALL SCAN
C   21 KES=0
C   LOOP = 0
C   START PROCESS MATRIX SCAN COLUMN BY COLUMN
C
C   22 DO 24 I2=1,NEMAX
C       NE=KPM(1,I2)
C       IF(NE) 24,24,26
C   26 IF(KEFLAG(NE).NE.0) GO TO 24
C
C   START SCAN OF COLUMN OF UNCALC'D EQUIP. LOOK AT SIGN OF STREAM
C
C   27 DO 42 I=2,N3MAX

```

Figure 22 (contd.)

```
C
C CHECK SIGN OF STREAM IN PROCESS MATRIX
C
28 IF(KPM(I ,I2)) 44,42,29
C
C IF STREAM IS INPUT SEE IF IT IS KNOWN. IF NOT GO TO NEXT EQUIPMENT.
C
29 I6=KPM(I ,I2)
40 IF(KSFLAG(I6).LE.0) GO TO 24
42 CONTINUE
C OUTPUT STREAM REACHED WITH INPUTS KNOWN.
C CALL OUT SUBROUTINE OF EQUIPMENT 'NE'. CALC. OUTPUT STREAMS
44 CALL EQUIP
    CALL EQCALL
    CALL REQUIP
46 KES=1
47 KEFLAG(NE)=1
C SET OUTPUT STREAM FLAGS FROM 0 TO 1
C
48 DO 50 I7=2,N3MAX
    I8=KPM(I7,I2)
    IF(I8.GE.0) GO TO 50
    I8=-I8
    KSFLAG(I8)=1
50 CONTINUE
24 CONTINUE
C THIS COMPLETES A SCAN OF PROCESS MATRIX COLUMN BY COLUMN
C
C IF KES IS 1 RESCAN PROCESS MATRIX
C
C IF KES IS 0, SEE IF ALL EQUIP. HAS BEEN CALC. IF NOT GO TO RCYCLE
55 IF(KES.GT.0) GO TO 21
C
C ARE ALL EQUIP. KNOWN? IF NOT GO TO RCYCLE
```

Figure 22 (contd.)

```
      DO 58 I9=1,NEMAX
      IF(KEFLAG(I9).NE.0) GO TO 58
      IF(IPUNCH.LT.0 .AND. KRET2.EQ.0) WRITE(7,FLLIST)
C   START TRIAL & ERROR CALCULATION OF RECYCLE STREAMS.
C
      61 CALL RCYCLE
C
C   IF KRET2 IS 2, EQUIP. LIST FOR RECYCLE CALC. NOT AVAIL. IN 'RCYCLE'
      IF(KRET2.NE.2) GO TO 21
      GO TO 67
      58 CONTINUE
C
C   'SUBSET' LOOP COMPLETE.
      67 NE2MAX=SAVE2
      NE3MAX=SAVE3
      RETURN
      END
```

Figure 23

```

SUBROUTINE SCAN
COMMON/SYSA/ KPM(10,50),KSEM(3,100),NSMAX
COMMON/SYSD/ KEFLAG(50),KSFLAG(100),KTRACE,DERROR,NPFREQ,IPUNCH
COMMON/EGPA/ EQPAR(25,50),NEMAX,MAXEQP
COMMON/STMA/ SEXTSV(23,100),SINTSV(10,100),NSMAX,MAXSEX,MAXSIN
DO 8 I =1,NSMAX
  IF(SEXTSV(1,I )) 5,5,3
3 IF(SINTSV(2,I )=1.) 7,6,7
C
C FLAG UNUSED STREAM NUMBERS AS =1
  5 KSFLAG(I )=1
  GO TO 8
C
C FLAG FEED STREAMS EQUAL TO 1
  6 KSFLAG(I )= 1
  GO TO 8
C
C FLAG PRODUCT AND INTERMEDIATE STREAMS EQUAL TO 0
  7 KSFLAG(I )= 0
  8 CONTINUE
  DO 9 I =1,NEMAX
C
C FLAG UNUSED EQUIPMENT NUMBERS AS =1, OTHERS AS 0
    KEFLAG(I)=0
    IF(KPM(1,I) .GT. 0 ) GO TO 9
    KEFLAG(I ) = =1
  9 CONTINUE
  RETURN
  END

```

Figure 24

```

SUBROUTINE RCYCLE
COMMON/SYSA/ KPM(10,50),KSEM(3,100),N3MAX
COMMON/SYSB/ KE1(50),NE1MAX,KE2(50),NE2MAX,KE3(50),NE3MAX
COMMON/SYSB/ KE4(10),NE4MAX,KRET,KRET2,KRET3
COMMON/SYSC/ LIMIT,LIMIT2,LIMIT3,LOOP,LOOPS
COMMON/SYSD/ KEFLAG(50),KSFLAG(100),KTRACE,DERROR,NPFREQ,IPUNCH
COMMON/CCNTL/ NIN,NOUT,NOCOMP,NE,NEN
COMMON/EQPA/ EQPAR(25,50),NEMAX,MAXEQP
COMMON/STMA/ SEXTSV(23,100),SINTSV(10,100),NSMAX,MAXSEX,MAXSIN
INTEGER                                LKE2/'KE2.'/,LKE3/'KE3.'/
IF(NE2MAX .EQ. 0) GO TO 4
DO 2 I=1,NE2MAX
2 KE1(I)=KE2(I)
NE1MAX=NE2MAX
NE2MAX=0
WRITE( 6,3) LKE2
3 FORMAT('1*****BEGIN TRIAL AND ERROR RECYCLE CALCULATIONS WITH EQUI 138
1PMENT LIST ',A4)
GO TO 10
4 IF(NE3MAX .EQ. 0) GO TO 8
DO 5 I=1,NE3MAX
5 KE1(I)=KE3(I)
NE1MAX=NE3MAX
NE3MAX=0
WRITE( 6,3) LKE3
GO TO 10
8 WRITE( 6,9)
9 FORMAT('0*****RECYCLE COMPUTATION REQUIRED BUT EQUIP* LIST WAS NOT
1 SUPPLIED*****')
KRET2=2
RETURN
10 LOOP=1
C
35 LIMIT2=0

```

Figure 24 (contd.)

```

WRITE(6,36)LOOP
36 FORMAT('0... BEGIN ' 'LOOP',I4,' ' '...')
C      'I' IS COUNTING INDEX FOR KE1
C
DO 86 I=1,NE1MAX
37 NE=KE1( I)
CALL EQUIP
CALL EQCALL
CALL TEST
CALL REQUIP
C IF LIMIT IS ZERO, CONVERGENCE OBTAINED FOR EQUIP. KE1( I)
C NUMBER OF ITEMS EXCEEDING DERROR IS VALUE OF LIMIT
IF(LIMIT) 86,86,68
68 LIMIT2=LIMIT2+1
86 CONTINUE
87 IF(LIMIT2.GT.0) GO TO 113
C
C CONVERGENCE OBTAINED IF LIMIT2 IS ZERO
C
GO TO 13
C
C CHECK NUMBER OF LOOPS THRU KE1
C
113 IF(LOOPS=LOOP.LE.0) GO TO 116
IF(NPFREQ.EQ. 0 ) GO TO 115
IF(LOOP/NPFREQ*NPFREQ.EQ.LOOP) CALL PTPRNT
115 LOOP=LOOP+1
C GO THRU KE1 LIST AGAIN
C
GO TO 35
C MAXIMUM NUMBER OF LOOPS TRIED WITHOUT CONVERGING.
C
116 WRITE( 6,117)
117 FORMAT('0*****EQUIPMENT CALCULATION LOOP DID NOT CONVERGE, COMPUTA

```

Figure 24 (contd.)

```
      1TIONS WILL CONTINUE.')
```

C

```
13 DO 14 I=1,NE1MAX
    K=KE1(I)
    KEFLAG(K)=1
    DO 12 J=2,N3MAX
      L=KPM(J,K)
      IF(L) 11,14,12
11  L=L
    IF(KSFLAG(L).EQ.0) KSFLAG(L)=1
12  CONTINUE
14  CONTINUE
    WRITE( 6,16)
16  FORMAT('O***** END OF RECYCLE CALCULATIONS'////////)
    KRET2=1
    RETURN
    END
```

Figure 25

```

SUBROUTINE EQUIP
COMMON/SYSA/ KPM(10,50),KSEM(3,100),N3MAX
COMMON/STMA/ SEXTSV(23,100),SINTSV(10,100),NSMAX,MAXSEX,MAXSIN
COMMON/CONTL/NIN,NOU,NOCOMP,NE,NEN
COMMON/STRMIN/SINUM(8,10),SICNST(10,8),SIMOLE(8),
1SICOMP(11,8),SIKV(10,8)
COMMON/STMOUT/SONUM(8,10),SOCNST(10,8),SOMOLE(8),
1SOCOMP(11,8),SOKV(10,8)
C 12 ENTRY POINTS: EQUIP & REQUIP
    DIMENSION SIIDUM(8,10),SOIDUM(8,10)
    EQUIVALENCE (SINUM,SIIDUM),(SONUM,SOIDUM)
    K =1
    IU =1
    DO 20 L2=2,N3MAX
        L3 =KPM(L2,NE)
C 13 IS THE STREAM NUMBER OF KPM=COLUMN(NE)
C
    21 IF(L3) 24,23,22
C
C  SET INPUT STREAMS IN COMMON/STRMIN/
    22 DO 25 I =1,NOCOMP
    25 SICOMP(I,K)=SEXTSV(I+12,L3)
        SIMOLE(K)=SEXTSV(12,L3)
        DO 51 I=1,10
    51 SICNST(I,K)=SEXTSV(I+1,L3)
        DO 27 I =1,MAXSIN
    27 SIIDUM(K ,I )=SINTSV(I ,L3)
    28 K =K +1
        GO TO 20
    24 NIN=K =1
C  NIN = NUMBER OF INPUT STREAMS
C
C  SET OUTPUT STREAMS IN COMMON/STMOUT/
    30 L3==L3

```

Figure 25 (contd.)

```
      DO 53 I=1,10
53  SOCNST(I,J)=SEXTSV(I+1,L3)
      DO 32 I =1,NOCOMP
32  SOCOMP(I,J)=SEXTSV(I+12,L3)
      SOMOLE(J)=SEXTSV(12,L3)
      DO 34 I =1,MAXSIN
34  SOIDUM(J,I)=SINTSV(I,L3)
35  J =J +1
20  CONTINUE
23  NOUT=J +1
C   NOUT = NUMBER OF OUTPUT STREAMS
      RETURN
```

Figure 26

```
C *** REQUIP ***
C SET CALC: OUTPUT STREAM VALUES IN SEXTSV & SINTSV
C
  ENTRY REQUIP
  DO 39 I =1,NOUT
    J =SOIDUM(I ,1) + .01
    DO 55 K=1,10
      55 SEXTSV(K+1,J)=SOCNST(K,I)
      DO 41 K =1,NOCOMP
        SEXTSV(12,J)=SOMOLE(I)
      41 SEXTSV(K+12,J)=SOCOMP(K,I)
      DO 43 K =1,MAXSIN
        43 SINTSV( K ,J )=SOIDUM(I , K )
      39 CONTINUE
      DO 46 I=1,NIN
        K=SIIDUM(I,1) + .01
        DO 57 J=1,10
          57 SEXTSV(J+1,K)=SICNST(J,I)
          DO 44 J=1,NOCOMP
            44 SEXTSV(J+12,K)=SICOMP(J,I)
            SEXTSV(12,K)=SIMOLE(I)
          DO 45 J=1,MAXSIN
            45 SINTSV( J,K)=SIIDUM(I,J)
          46 CONTINUE
        RETURN
      END
```

Figure 27

```

SUBROUTINE EQCALL
COMMON/SYSD/ KEFLAG(50),KSFLAG(100),KTRACE,DERROR,NPFREQ,IPUNCH
COMMON/EQPB/ NECALL(50),NEXEQN(50),NAME(50)
COMMON/CCNTL/NIN,NOUT,NOCOMP,NE,NEN
COMMON/STRMIN/SINUM(8,10),SICNST(10,8),SIMOLE(8),
1SICOMP(11,8),SIKV(10,8)
NAMELIST/DATIN/ SINUM,SIVPFR,SITEMP,SIPRES,SIENTH,SIMCLE,SICOMP
IF(KTRACE.GT. 0 ) WRITE(6,60) NE,NAME(NE)
IF(KTRACE.EQ. 2 ) WRITE(6,DATIN)
60 FORMAT('ONOW CALLING EQUIPMENT',I4,' =',A4,'')
NEQUIP=NECALL(NE)
GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,
1 23,24,25,26,27,28,29,30),NEQUIP
1 CALL ROST
RETURN
2 CALL SCRN
RETURN
3 CALL MIXR
RETURN
4 CALL GRND
RETURN
5 CALL PREP
RETURN
6 CALL LECH
RETURN
7 CALL FLUE
RETURN
8 CALL FLTR
RETURN
9 CALL SETL
RETURN
10 CALL CLAS
RETURN
11 CALL RLTS

```

MAN

144

Figure 27 (contd.)

```
      RETURN
12  CALL ELET
      RETURN
13  CALL AMIX
      RETURN
14  CONTINUE
15  CONTINUE
16  CONTINUE
17  CONTINUE
18  CONTINUE
19  CONTINUE
20  CONTINUE
21  CONTINUE
22  CONTINUE
23  CONTINUE
24  CONTINUE
25  CONTINUE
26  CONTINUE
27  CONTINUE
28  CONTINUE
29  CONTINUE
30  CONTINUE
      RETURN
      END
```

Figure 28

```

SUBROUTINE TEST
COMMON/SYSB/ KE1(50),NE1MAX,KE2(50),NE2MAX,KE3(50),NE3MAX
COMMON/SYSB/ KE4(10),NE4MAX,KRET,KRET2,KRET3
COMMON/SYSC/ LIMIT,LIMIT2,LIMIT3,LOOP,LOOPS
COMMON/SYSD/ KEFLAG(50),KSFLAG(100),KTRACE,DERROR,NPFREQ,IPUNCH
COMMON/CONTL/NIN,NOUT,NOCOMP,NE,NEN
COMMON/STMA/ SEXTSV(23,100),SINTSV(10,100),NSMAX,MAXSEX,MAXSIN
COMMON/STMOUT/SONUM(8,10),SOCNST(10,8),SOMOLE(8),
1SOCOMP(11,8),SOKV(10,8)
DIMENSION SOIDUM(8,10)
EQUIVALENCE (SONUM,SOIDUM)
LOGICAL FLAG,FFLAG

C
C
REAL SAVEX(10,20)/200*0./,SAVEF(10,20)/200*0./
REAL SAVEQ(10,20)/200*1./
LIMIT=0
LIMIT3=0
I=1
20 J=SOIDUM(I,1) + 0.01

C
C J IS OUTPUT STREAM NUMBER OF EQUIP.(NE)
C USE 'DERROR' TO CHECK OUTPUT STREAM VALUES AGAINST 'SEXTSV','SINTSV'
C
TMOLE=SEXTSV(12,J)
IF(ABS(TMOLE).LT.1.E=20) GO TO 21
IF(ABS((TMOLE-SOMOLE(I))/TMOLE).GT.DERROR) LIMIT3=LIMIT3+1
GO TO 22
21 IF(ABS(SOMOLE(I)).GT.DERROR) LIMIT3=LIMIT3+1
GO TO 22
24 CONTINUE
K=3
SINK=SINTSV(K,J)
IF(ABS(SINK).LT.1.E=20) GO TO 25

```

Figure 28 (contd.)

```

        IF (ABS((SINK=SOIDUM(I,K))/SINK).GT.DERROR) LIMIT3=LIMIT3+1
        GO TO 26
25  IF (ABS(SCIDUM(I,K)).GT.DERROR) LIMIT3=LIMIT3+1
26  CONTINUE
        IF (LIMIT3.LE.0) GO TO 28
C    IF (KTRACE.GT. 0 ) WRITE(6,30) LOOP,J,NE,LIMIT3
        LIMIT3=0
        LIMIT=LIMIT+1
28  IF ( I .GE. NOUT ) RETURN
        I=I+1
        GO TO 20
30  FORMAT('OIN LOOP NUMBER',I4,' OF THE TRIAL & ERROR ITERATIVE CALCU
1  LATIONS','/ ' OUTPUT STREAM NUMBER',I5,' OF EQUIPMENT NUMBER',I4,
2  ' HAS',I4,' STREAM VALUES WHICH EXCEED THE CONVERGENCE LIMITS.')
C  WEGSTEIN'S ITERATIVE METHOD .....
22  LIMIT4=0
        FLAG=.FALSE.
        FFLAG=.FALSE.
C... TEST PERMISSIBLE STREAM NUMBERS ... 0 < NE4MAX .LE. 10
        IF (NE4MAX.LE.0 .OR. NE4MAX.GT.10) GO TO 42
        DO 40 IK=1,NE4MAX
        IF (IK.GT.10) GO TO 42
        IF (J.EQ.KE4(IK)) GO TO 41
40  CONTINUE
        GO TO 42
41  FLAG=.TRUE.
42  DO 55 K=1,NOCOMP
        SOCOM=SOCOMP(K,I)
        COMPK=SEXTSV(K+12,J)
        IF (ABS(COMPK).LT. 1.E=20) GO TO 27
        IF (ABS((COMPK=SOCOM )/COMPK).LT.DERROR) GO TO 43
        GO TO 47
27  IF (ABS(SOCOM ) .GT.DERROR) GO TO 47
43  IF (.NOT.FLAG) GO TO 55

```

Figure 28 (contd.)

```

C... NCW TEST LOOP COUNTER AND BRANCH FOR 1, 2 & GE 3
  IF(LOOP=2) 44,45,46
44 SAVEX(IK,K)=SOCOM
  GO TO 55
45 SAVEF(IK,K)=SOCOM
  GO TO 55
46 SAVEX(IK,K)=COMPK
  SAVEF(IK,K)=SOCOM
  GO TO 55
47 LIMIT4=LIMIT4+1
  IF(LOOP.LT.3) GO TO 43
  IF(.NOT.FLAG) GO TO 55
C... CALCULATE WEGSTEIN'S COEFFICIENTS = PHI & Q...
  GOLD=SAVEQ(IK,K)
  PHI=(COMPK=SAVEX(IK,K))/(SOCOM      =SAVEF(IK,K))
  IF(1.=PHI) 48,50,48
48 Q=1./(1.=PHI)
  IF(Q.LT.0.00)GO TO 52
  IF(KTRACE.GT.0) WRITE(6,49) J,I,Q
49 FORMAT('OWEGSTEIN/TEST2 STREAM',I4,'=',I3,5X,'Q=',F10.3)
  Q=1.E=10
  GO TO 53
50 IF(KTRACE.GT.0) WRITE(6,51) J,I,PHI
51 FORMAT('OWEGSTEIN/TEST1 STREAM',I4,'=',I3,5X,'PHI=',E15.5)
  GO TO 46
52 IF(Q.GT.=10.0) GO TO 53
  IF(KTRACE.GT.0) WRITE(6,49) J,I,Q
  Q.=10.0
53 X2=Q*COMPK+(1.=Q)*SOCOM
  IF(KTRACE.GT.0) WRITE(6,36) J,K,COMPK,SOCOM,X2,Q,SAVEX(IK,K),
    SAVEF(IK,K)
36 FORMAT('OSTREAM',I4,'=',I3,5X,'X=',E13.5,5X,'F=',E13.5,5X,'XNEW=',
  1 E13.5,5X,'Q=',F10.3/ 16X,'SAVEX=',E13.5,' SAVEF=',E13.5)
  IF(X2.LE.0.) X2=SOCOM/2.

```

Figure 28 (contd.)

```
      FFLAG=.TRUE.
      IF(G*QOLD.LT.0..OR.LOOP.EQ.3) GO TO 54
      IF((X2=COMP(K))*(SAVEF(IK,K)=COMP(K)).LE.0.) GO TO 54
      SOCOMP(K,I)=COMP(K)
      GO TO 37
54    SOCOMP(K,I)=X2
37    SAVEX(IK,K)=COMP(K)
      SAVEF(IK,K)=SOCOM
      SAVEQ(IK,K)=Q
55    CONTINUE
56    LIMIT3=LIMIT3+LIMIT4
      IF(FLAG.AND.FFLAG) GO TO 57
      GO TO 24
57    TMOLE=0.
      DO 58 K=1,NOCOMP
58    TMOLE=TMOLE+SOCOMP(K,I)
      IF(KTRACE.GT.0) WRITE(6,59) J,SOMOLE(I),TMOLE
59    FORMAT('OSTREAM',I4,' OLD AND NEW T=MOLES..',2F15.5)
      SOMOLE(I)=TMOLE
      GO TO 24
      END
```

Figure 29

```
      SUBROUTINE TRANSF(I,LTH,KFLAG,N1,SV,N2,N3,A,L)
C
C**THIS SUBROUTINE TRANSFERS DATA FROM TWO DIMENSIONAL ARRAY SV(N2,N3)
C** TO A ONE DIMENSIONAL A(1)....A(5) FOR PRINT-OUT.
      DIMENSION KFLAG(N1), SV(N2,N3), A(5)
      DO 5 K=1,LTH
      K1=K+1
      K2=KFLAG(K)
      5 A(K1)=SV(L,K2)
      RETURN
C**THIS PORTION FOR INTEGER AND LITERAL ARRAYS
      ENTRY TRANSX(I,LTH,KFLOG,N1,IV,IA)
      DIMENSION KFLOG(N1), IV(N1), IA(5)
      DO 7 K=1,LTH
      K1=K+1
      K2=KFLOG(K)
      7 IA(K1)=IV(K2)
      RETURN
      END
```

MAN

Figure 30

```

SUBROUTINE TPRINT
COMMON/SYSA/ KPM(10,50),KSEM(3,100),N3MAX
COMMON/SYSAA/ TITLE(20),COMPNT(20),KOMNAM(80)
COMMON/SYSB/ KE1(50),NE1MAX,KE2(50),NE2MAX,KE3(50),NE3MAX
COMMON/SYSB/ KE4(10),NE4MAX,KRET,KRET2,KRET3
COMMON/SYSC/ LIMIT,LIMIT2,LIMIT3,LOOP,LOOPS
COMMON/SYSD/ KEFLAG(50),KSFLAG(100),KTRACE,DERROR,NPFREQ,IPUNCH
COMMON/EQPA/ EQPAR(25,50),NEMAX,MAXEQP
COMMON/EQPB/ NECALL(50),NEXEQN(50),NAME(50)
COMMON/STMA/ SEXTSV(23,100),SINTSV(10,100),NSMAX,MAXSEX,MAXSIN
COMMON/CONTL/NIN,NOUT,NOCOMP,NE,NEN
INTEGER COMPNT,PFLAG(100),TITLE

C
C***** THIS PORTION WAS TO PRODUCE RELOAD DECK.....TEMPORARILY DELETED?
C
      RETURN

C
C***** BIPRNT *****
C
      ENTRY BIPRNT
C...COMMON AREA PUNCH DELETED...
      INTEGER FHDG(4)/'FINAL RESULTS'//'/
      INTEGER HDNG(16)/' (1H1,3X,15H'FINA', 'L RESULTS'//13X,' ', '20A4//32
1X,16H' 'ST', 'REAM SUMMARY' )'//
      INTEGER HDG(10)/' (1H1,3X,15H'FINA', 'L RESULTS'//13X,' ', '20A4//)'//

C
      DO 5 I=1,4
      HDNG(I+3)=FHDG(I)
      5 HDG (I+3)=FHDG(I)
      GO TO 10

C
C***** PTPRNT *****
C

```

Figure 30 (contd.)

```

ENTRY PTPRNT
DIMENSION IPNT(5),JPNT(5),RPNT(5)
INTEGER IHDG(4)/'INPUT DATA'/'/'
INTEGER LABEL(32)/'VAPOR FRACTION','TEMPERATURE, R','PRESSURE, PSI
1A','ENTHALPY, BTU','VISCOSITY, PFH','K, BTU=FT=HR=F','Z=FACTOR, LI
2G','Z=FACTOR, VAP.'/
DO 7 I=1,4
  HDNG(I+3)=IHDG(I)
7 HDG(I+3)=IHDG(I)
10 WRITE( 6,HDNG) TITLE
  J=0
  DO 12 I=1,NSMAX
    IF(SEXTSV(1,I).EQ.0.) GO TO 12
    J=J+1
    PFLAG(J)=I
12 CONTINUE
    DO 24 I=1,J,5
      LTH=I+4
      IF(LTH.GT.J) LTH=J
      WRITE( 6,14) ( PFLAG(K),K=I,LTH)
14 FORMAT('OSTREAM NUMBER ',5(I9,5X))

C
    DO 15 K=I,LTH
      KPNT=K+1
      MPNT= PFLAG(K)
      IPNT(KPNT)=KSEM(2,MPNT)
15 JPNT(KPNT)=KSEM(3,MPNT)
      LTHP=LTH+1
      WRITE( 6,16) (IPNT(K),JPNT(K),K=1,LTHP)
16 FORMAT('OCEQUIP. CONXION ',5(5H FR ,I2,4H TO ,I2,1X))
19 FORMAT(' ',4A4,T16,5F14.4)
      WRITE( 6,20)
20 FORMAT('O',40X,'COMPOSITION, LB=MOLES/HOUR' / )

```

Figure 30 (contd.)

```
      NOCOP2 = NOCOMP + 2
      DO 22 L = NOCOP2
      CALL TRANSF(I,LTH, PFLAG,100,SEXTSV,23,100,RPNT,L)
      K1 = 4*L-11
      K2=K1+3
22  WRITE( 6,19) (KOMNAM(K),K=K1,K2), (RPNT(K),K=1,LTHP)
      CALL TRANSF(I,LTH, PFLAG,100,SEXTSV,23,100,RPNT,3)
      WRITE( 6,23) (RPNT(K),K=1,LTHP)
23  FORMAT('0',9X,'TOTAL',5F14.4)
24  WRITE( 6,26)
26  FORMAT('////////')
      RETURN
      END
```

Supporting Subroutine Printouts

Figure 31

```

SUBROUTINE ROST
COMMON/SYSD/KEFLAG(50),KSFLAG(100),KTRACE,DERROR,NPFREQ,IPUNCH
COMMON/EQPA/ EQPAR(25,50), NEMAX,MAXEQP
COMMON/CONTL/NIN,NOUT,NOCOMP,NE,NEN
COMMON/STRMIN/SINUM(8,10),SICNST(10,8),SIMOLE(8),
1SICOMP(11,8),SIKV(10,8)
COMMON/STMOUT/SONUM(8,10),SOCNST(10,8),SOMOLE(8),
1SOCOMP(11,8),SOKV(10,8)
DIMENSION RM(5),AVC(5),SICPWT(11,8),SOCPWT(11,8),X(10),Y(10)
WRITE (6,4)
4 FORMAT('1 ROASTER')
M=NOCOMP-1
DO 13 J=1,NOUT
DO 11 K=1,M.
11 SOCPWT(K,J)=0.
13 SOMOLE(J)=0.
KC=EQPAR(2,NE)+.1
DO 15 K=1,M
SICPWT(K,1)=SICOMP(K,1)*SIMOLE(1)
15 WRITE(6,501) K,SICPWT(K,1)
501 FORMAT(' SICPWT(',I2,') = ',G14.7)
TWZNS=SICPWT(KC,1)/(1.-SICNST(5,1))
WRITE(6,502) TWZNS
502 FORMAT(' TWZNS = ',G14.7)
GVOL=EQPAR(3,NE)*(1./EQPAR(4,NE))*(EQPAR(5,NE)+460.)/492.*24.*60.
CR=EQPAR(6,NE)*SQRT(GVOL)
WRITE(6,600) CR
600 FORMAT('OCRITICAL RADIUS = ',G14.7)
SOCOMP(NOCOMP,1)=GVOL/((24.*60.)*492./((EQPAR(5,NE)+460.))
AR=SICNST(1,1)
BR=SICNST(2,1)
CC=SICNST(3,1)
XF=0.
19 XF=XF+.1

```

Figure 31 (contd.)

```
      Q=XF/(1.-CC*XF)
      ROP=AR*BR**Q
      IF(ROP.GE.CR) GO TO 21
      IF(XF.GE.1.) GO TO 20
      GO TO 19
20  XF=1.
      GO TO 25
21  XF=XF-.1
23  Q=XF/(1.-CC*XF)
      ROP=AR*BR**Q
      IF(ROP.GE.CR) GO TO 25
      XF=XF+.01
      GO TO 23
25  WRITE(6,338) XF
338  FORMAT('  XF      ',G14.7)
      DO 33 J=1,2
      IF(J.NE.1) GO TO 29
      HT=EQPAR(7,NE)/GVOL+SICNST(4,1)
      DO 27 I=1,10
      P=I
      X(I)=P*.1
      XX=P*XF/10.
      Q=XX/(1.-CC*XX)
27  Y(I)=AR*BR**Q
      CALL LSC(X,Y,10,A,B,C)
      GO TO 31
29  HT=EQPAR(8,NE)/(SIMOLE(1)*(1.-XF))+SICNST(4,1)
      A=EQPAR(9,NE)*AR
      B=EQPAR(10,NE)*BR
      C=EQPAR(11,NE)*CC
31  SOCNST(1,J)=A
      SOCNST(2,J)=B
      SOCNST(3,J)=C
      SOCNST(4,J)=HT
```

Figure 31 (contd.)

```

        WRITE(6,342) A,B,C
        WRITE(6,503) SOCNST(4,J)
503  FORMAT(' HT = ',G14.7)
342  FORMAT(' A = ',G14.7,' B = ',G14.7,' C = ',G14.7)
        CALL CONVTA(A,B,C,SINUM(1,3),HT,AVC(J))
        WRITE(6,344) AVC(J)
344  FORMAT(' AVC = ',G14.7)
        33 SOCNST(5,J)=AVC(J)
511  DO 45 J=1,2
        IF(J.NE.1) GO TO 37
        XJ=XF
        GO TO 39
37  XJ=1.-XF
39  DO 45 K=1,M
        L=K+11
        IF(K.NE.KC) GO TO 41
        SOCPWT(KC,J) = TWZNS*XJ*(1.-AVC(J))
        WRITE(6,504) J,TWZNS,XJ,SOCPWT(KC,J)
504  FORMAT(' J = ',I2,' TWZNS = ',G14.7,' XJ = ',G14.7,' SOCPWT(KC,J)
        1= ',G14.7)
        GO TO 45
41  SOCPWT(K,J)=(SICPWT(K,1)+EQPAR(L,NE)*TWZNS*(AVC(J)-SICNST(5,1)))*
        1XJ
        WRITE(6,505) SICPWT(K,1),EQPAR(L,NE),XJ
505  FORMAT(' SICPWT = ',G14.7,' EQPAR = ',G14.7,' XJ = ',G14.7)
        WRITE(6,506) AVC(J),SICNST(5,1),SOCPWT(K,J)
506  FORMAT(' AVC = ',G14.7,' SICNST(5,1) = ',G14.7,' SOCPWT = ',G14.7)
45  CONTINUE
        DO 51 J=1,2
        DO 47 K=1,M
        IF(SOCPWT(K,J).GT.0.) GO TO 47
        SOCPWT(K,J)=0.
47  SOMOLE(J)=SOMOLE(J)+SOCPWT(K,J)
        DO 49 K=1,M

```

Figure 31 (contd.)

```
49 SOCOMP(K,J)=SOCPWT(K,J)/SOMOLE(J)
   DO 51 I=3,10
51 SONUM(J,I)=SINUM(I,I)
   WRITE(6,6) SOCOMP(NOCOMP,1)
6  FORMAT('0  GAS FLOW OVERHEAD (CFM) = ',G14.7)
   DO 57 J=1,2
   WRITE(6,10) SONUM(J,1)
10 FORMAT('0  STREAM NUMBER ',G14.7)
   DO 53 K=1,M
53 WRITE (6,12) K,SOCOMP(K,J)
12 FORMAT('  WT FRACTION COMP ',I2,' = ',G14.7)
   WRITE (6,14) SOMOLE(J)
14 FORMAT('  TOTAL STREAM WEIGHT = ',G14.7)
57 WRITE (6,18) AVC(J)
18 FORMAT('  AVERAGE CONVERSION = ',G14.7)
   RETURN
   END
```

Figure 32

```

SUBROUTINE SCRN
COMMON/SYSD/KEFLAG(50),KSFLAG(100),KTRACE,DERROR,NPFREQ,IPUNCH
COMMON/EQPA/ EQPAR(25,50), NEMAX,MAXEQP
COMMON/CONTL/NIN,NOUT,NOCOMP,NE,NEN
COMMON/STRMIN/SINUM(8,10),SICNST(10,8),SIMOLE(8),
1SICOMP(11,8),SIKV(10,8)
COMMON/STMOUT/SO NUM(8,10),SOCNST(10,8),SOMOLE(8),
1SOCOMP(11,8),SOKV(10,8)
DIMENSION RM(5),AVC(5),SICPWT(11,8),SOCPWT(11,8),X(10),Y(10)
WRITE (6,4)
4 FORMAT('1 SCREEN')
M=NOCOMP-1
DO 13 J=1,NOUT
DO 11 K=1,M
11 SOCPWT(K,J)=0.
13 SOMOLE(J)=0.
KC=EQPAR(2,NE)+.1
DO 15 K=1,M
15 SICPWT(K,1)=SICOMP(K,1)*SIMOLE(1)
TWZNS=SICPWT(KC,1)/(1.-SICNST(5,1))
AR=SICNST(1,1)
BR=SICNST(2,1)
CC=SICNST(3,1)
XF=0.
19 XF=XF+.1
Q=XF/(1.-CC*XF)
ROP=AR*BR**Q
IF(ROP.GT.EQPAR(3,NE)) GO TO 23
IF(XF.GE.1.) GO TO 21
GO TO 19
21 XF=1.
DO 61 K=1,M
61 SOCOMP(K,1)=SICOMP(K,1)
SOMOLE(1)=SIMOLE(1)

```

Figure 32 (contd.)

```

DO 63 J=1,2
DO 65 I=3,10
65 SONUM(J,I)=SINUM(1,I)
DO 63 K=1,5
63 SOCNST(K,J)=SICNST(K,1)
GO TO 99
23 XF=XF-.1
25 Q=XF/(1.-CC*XF)
ROP=AR*BR**Q
IF(ROP.GE.EQPAR(3,NE)) GO TO 27
XF=XF+.01
GO TO 25
27 DO 35 J=1,2
SOCNST(4,J)=SICNST(4,1)
WRITE(6,503) SOCNST(4,J)
... 503 FORMAT(' HT = ',G14.7)
DO 33 I=1,10
X(I)=0.
Y(I)=0.
P=I
IF(J.NE.1) GO TO 29
XX=P*XF/10.
GO TO 31
29 XX=XF+(1.0-XF)/10.*P
31 X(I)=P*.1
Q=XX/(1.-CC*XX)
33 Y(I)=AR*BR**Q
CALL LSC(X,Y,10,A,B,C)
SOCNST(1,J)=A
SOCNST(2,J)=B
SOCNST(3,J)=C
WRITE(6,342) A,B,C
342 FORMAT(' A = ',G14.7,' B = ',G14.7,' C = ',G14.7)
HT=SICNST(4,J)
CALL CONVT(A,B,C,SINUM(1,3),HT,AVC(J))

```

Figure 32 (contd.)

```
      WRITE(6,344) AVC(J)
344  FORMAT('  AVC = ',G14.7)
      35  SOCNST(5,J)=AVC(J)
      DO 45 J=1,2
      IF(J.NE.1) GO TO 37
      XJ=XF
      GO TO 39
37  XJ=1.-XF
39  DO 45 K=1,M
      L=K+3
      IF(K.NE.KC) GO TO 41
      SOCPWT(KC,J) = TWZNS*XJ*(1.-AVC(J))
      GO TO 45
41  SOCPWT(K,J)=(SICPWT(K,1)+EQPAR(L,NE)*TWZNS*(AVC(J)-SICNST(5,1)))*
      IXJ
45  CONTINUE
      DO 51 J=1,2
      DO 47 K=1,M
47  SOMOLE(J)=SOMOLE(J)+SOCPWT(K,J)
      DO 49 K=1,M
49  SOCOMP(K,J)=SOCPWT(K,J)/SOMOLE(J)
      DO 51 I=3,10
51  SONUM(J,I)=SINUM(1,I)
99  DO 57 J=1,2
      WRITE(6,10) SONUM(J,1)
10  FORMAT('0  STREAM NUMBER ',G14.7)
      DO 53 K=1,M
53  WRITE (6,12) K,SOCOMP(K,J)
12  FORMAT('  WT FRACTION COMP ',I2,' = ',G14.7)
      WRITE (6,14) SOMOLE(J)
14  FORMAT('  TOTAL STREAM WEIGHT = ',G14.7)
57  WRITE (6,18) AVC(J)
18  FORMAT('  AVERAGE CONVERSION = ',G14.7)
      RETURN
      END
```

Figure 33

```

SUBROUTINE GRND
COMMON/SYSD/KEFLAG(50),KSFLAG(100),KTRACE,DERROR,NPFREQ,IPUNCH
COMMON/EQPA/ EQPAR(25,50), NEMAX,MAXEQP
COMMON/CONTL/NIN,NOUT,NOCOMP,NE,NEN
COMMON/STRMIN/SINUM(8,10),SICNST(10,8),SIMOLE(8),
1SICOMP(11,8),SIKV(10,8)
COMMON/STMOUT/SONUM(8,10),SOCNST(10,8),SOMOLE(8),
1SOCOMP(11,8),SOKV(10,8)
DIMENSION X(10),Y(10)
WRITE(6,4)
4 FORMAT('1 BALLMILL')
M=NOCOMP-1
AR=SICNST(1,1)
BR=SICNST(2,1)
CC=SICNST(3,1)
DO 7 I=1,10
X(I)=0.
Y(I)=0.
P=I
Z=P*.1
Q=Z/(1.-CC*Z)
RM=AR*BR**Q
RN=(RM-EQPAR(2,NE))*EXP(-EQPAR(3,NE)/SIMOLE(1))+EQPAR(2,NE)
X(I)=Z
7 Y(I)=RN
CALL LSC(X,Y,10,A,B,C)
SOCNST(1,1)=A
SOCNST(2,1)=B
SOCNST(3,1)=C
WRITE(6,342) A,B,C
342 FORMAT(' A = ',G14.7,' B = ',G14.7,' C = ',G14.7)
TLL=0.
UL=SICNST(4,1)
CALL TGS(TLL,UL,SOCNST(5,1),SONUM(1,3),A,B,C,AHT)

```

Figure 33 (contd.)

```
59 SOCNST(4,1)=AHT  
   WRITE(6,503) SOCNST(4,1)  
503 FORMAT(' HT = ',G14.7)  
   DO 13 K=1,M  
13  SUCOMP(K,1)=SICOMP(K,1)  
   SOMOLE(1)=SIMOLE(1)  
   DO 15 I=3,10  
15  SONUM(1,I)=SINUM(1,I)  
   WRITE(6,10) SONUM(1,1)  
10  FORMAT('0  STREAM NUMBER. ',G14.7)  
   DO 53 K=1,M  
53  WRITE (6,12) K,SUCOMP(K,1)  
12  FORMAT('    WT FRACTION COMP. ',I2,' = ',G14.7)  
   WRITE (6,14) SOMOLE(1)  
14  FORMAT('    TOTAL STREAM WEIGHT = ',G14.7)  
   RETURN  
   END
```

Figure 34

```

SUBROUTINE PREP
COMMON/SYSD/KEFLAG(50),KSFLAG(100),KTRACE,DERROR,NPFREQ,IPUNCH
COMMON/EQPA/ EQPAR(25,50), NEMAX,MAXEQP
COMMON/CONTL/NIN,NOUT,NOCOMP,NE,NEN
COMMON/STRMIN/SINUM(8,10),SICNST(10,8),SIMOLE(8),
1SICOMP(11,8),SIKV(10,8)
COMMON/STMOUT/SONUM(8,10),SOCNST(10,8),SOMOLE(8),
1SDCOMP(11,8),SOKV(10,8)
DIMENSION EFF(10),AVC(5),SICPWT(11,8),SOCPWT(11,8),X(10),Y(10)
WRITE (6,4)
4 FORMAT('1 PRECIPITATOR')
M=NOCOMP-1
DO 13 J=1,NOUT
DO 11 K=1,M
11 SOCPWT(K,J)=0.
13 SOMOLE(J)=0.
KC=EQPAR(2,NE)+.1
DO 15 K=1,M
15 SICPWT(K,1)=SICOMP(K,1)*SIMOLE(1)
TWZNS=SICPWT(KC,1)/(1.-SICNST(5,1))
AR=SICNST(1,1)
BR=SICNST(2,1)
CC=SICNST(3,1)
SOCOMP(NOCOMP,1)=SICOMP(NOCOMP,1)
GVOL=SICOMP(NOCOMP,1)/60.*(EQPAR(3,NE)+460.)/492.
DO 17 J=1,10
17 EFF(J)=0.
TEFF=0.
Z=-.025
DO 19 J=1,20
Z=Z+.05
Q=Z/(1.-CC*Z)
RM=AR*BR**Q
EFFZ=1.-EXP(-EQPAR(4,NE)*RM/GVOL)
TEFF=TEFF+EFFZ*.05

```

MAN

Figure 34 (contd.)

```
      I=(J-1)/2+1
19  EFF(I)=EFF(I)+EFFZ*.05
      DO 20 I=1,10
20  EFF(I)=EFF(I)/.1
      DO 31 J=1,2
      XF=0.
      DO 25 I=1,10
      X(I)=0.
      Y(I)=0.
      P=I
      Z=P*.1
      Q=Z/(1.-CC*Z)
      Y(I)=AR*BR**Q
      IF(J.NE.1) GO TO 21
      RX=.1*(1.-EFF(I))/(1.-TEFF)
      GO TO 23
21  RX=.1*EFF(I)/TEFF
23  XF=XF+RX
25  X(I)=XF
      CALL LSC(X,Y,10,A,B,C)
      SOCNST(1,J)=A
      SOCNST(2,J)=B
      SOCNST(3,J)=C
      SOCNST(4,J)=SOCNST(4,1)
      WRITE(6,503) SOCNST(4,J)
503  FORMAT(' HT = ',G14.7)
      HT=SOCNST(4,J)
      WRITE(6,342) A,B,C
342  FORMAT(' A = ',G14.7,' B = ',G14.7,' C = ',G14.7)
      CALL CONVT(A,B,C,SINUM(1,3),HT,AVC(J))
31  SOCNST(5,J)=AVC(J)
      DO 49 J=1,2
      IF(J.NE.1) GO TO 35
      XJ=1.-TEFF
      GO TO 37
```

Figure 34 (contd.)

```
35 XJ=TEFF
37 DO 43 K=1,M
    L=K+4
    IF(K.NE.KC) GO TO 39
    SOCPWT(KC,J)=TWZNS*XJ*(1.-AVC(J))
    GO TO 43
39 SOCPWT(K,J)=(SICPWT(K,1)+EQPAR(L,NE)*TWZNS*(AVC(J)-SICNST(5,1)))*
    IXJ
43 CONTINUE
    DO 45 K=1,M
45 SOMOLE(J)=SOMOLE(J)+SOCPWT(K,J)
    DO 47 K=1,M
47 SOCOMP(K,J)=SOCPWT(K,J)/SOMOLE(J)
    DO 49 I=3,10
49 SONUM(J,I)=SINUM(I,I)
    DO 57 J=1,2
    WRITE(6,10) SONUM(J,1)
10 FORMAT('0  STREAM NUMBER ',G14.7)
    DO 53 K=1,M
53 WRITE (6,12) K,SOCOMP(K,J)
12 FORMAT('  WT FRACTION COMP ',I2,' = ',G14.7)
    WRITE (6,14) SOMOLE(J)
14 FORMAT('  TOTAL STREAM WEIGHT = ',G14.7)
57 WRITE (6,18) AVC(J)
18 FORMAT('  AVERAGE CONVERSION = ',G14.7)
    WRITE (6,26) TEFF
26 FORMAT('  TOTAL EFF = ',G14.7)
    RETURN
END
```

Figure 35

```

SUBROUTINE AMIX
COMMON/SYSD/KEFLAG(50),KSFLAG(100),KTRACE,DERROR,NPFREQ,IPUNCH
COMMON/EQPA/ EQPAR(25,50), NEMAX,MAXEQP
COMMON/CONTL/NIN,NOUT,NOCOMP,NE,NEN
COMMON/STRMIN/SINUM(8,10),SICNST(10,8),SIMOLE(8),
1SICOMP(11,8),SIKV(10,8)
COMMON/STMOUT/SONUM(8,10),SOCNST(10,8),SOMOLE(8),
1SOCOMP(11,8),SOKV(10,8)
DIMENSION SICPWT(11,8),SOCPWT(11,8)
WRITE(6,4)
4 FORMAT('1 LEACH SLURRY TANK')
SOMOLE(1)=0.
M=NOCOMP-1
DO 11 J=1,2
DO 11 K=1,M
11 SICPWT(K,J)=SICOMP(K,J)*SIMOLE(J)
KC1=EQPAR(2,NE)+.1
KC2=EQPAR(3,NE)+.1
KC3=EQPAR(4,NE)+.1
KC4=EQPAR(5,NE)+.1
NS=EQPAR(6,NE)+.1
DO 13 K=1,M
13 SOCPWT(K,1)=SICPWT(K,1)+SICPWT(K,2)
ZNO=SOCPWT(KC1,1)/81.38
ACID=SOCPWT(KC2,1)/98.08
FESO=SOCPWT(KC3,1)/399.9
X1=ACID/ZNO
X2=FESO/ZNO
XS=X1+3.*X2
IF(XS.GE.(EQPAR(7,NE)+3.*EQPAR(8,NE))) GO TO 19
IF(X1.GE.EQPAR(7,NE)) GO TO 17
IF(X2.GE.EQPAR(8,NE)) GO TO 15
AREQ=ZNO*EQPAR(7,NE)*98.08
ANEED=AREQ-SOCPWT(KC2,1)

```

Figure 35 (contd.)

```

WRITE(6,400) ANEED
400 FORMAT('0 TONS/DAY MAKEUP ACID ADDED = ',G14.7)
SOCPWT(KC2,1)=SOCPWT(KC2,1)+ANEED
FEREQ=ZNO*EQPAR(8,NE)*399.9
FEMR=FEREQ-SOCPWT(KC3,1)
WRITE(6,402) FEMR
402 FORMAT('0 TONS/DAY MAKEUP FE2(SO4)3 ADDED = ',G14.7)
SOCPWT(KC3,1)=SOCPWT(KC3,1)+FEMR
GO TO 19
15 X1=EQPAR(7,NE)+3.*EQPAR(8,NE)-3.*X2
AREQ=ZNO*X1*98.08
ANEED=AREQ-SOCPWT(KC2,1)
WRITE(6,400) ANEED
SOCPWT(KC2,1)=SOCPWT(KC2,1)+ANEED
GO TO 19
17 X2=(EQPAR(7,NE)+3.*EQPAR(8,NE)-X1)/3.
FEREQ=ZNO*X2*399.9
FEMR=FEREQ-SOCPWT(KC3,1)
WRITE(6,402) FEMR
SOCPWT(KC3,1)=SOCPWT(KC3,1)+FEMR
19 WS=0.
DO 25 K=1,NS
25 WS=WS+SOCPWT(K,1)
TW=0.
DO 27 K=1,M
27 TW=TW + SOCPWT(K,1)
WFS=WS/TW
IF(WFS.LE.EQPAR(9,NE)) GO TO 29
TAW=WS/EQPAR(9,NE)
WADD=TAW-TW
WRITE(6,404) WADD
404 FORMAT('0 TONS/DAY MAKEUP WATER ADDED = ',G14.7)
SOCPWT(KC4,1)=SOCPWT(KC4,1)+WADD
29 DO 31 K=1,M

```

Figure 35 (contd.)

```
31 SOMOLE(1)=SOMOLE(1)+SOCPWT(K,1)
   DO 33 K=1,M
33 SOCOMP(K,1)=SOCPWT(K,1)/SOMOLE(1)
   DO 37 J=3,10
37 SONUM(1,J)=SINUM(1,J)
   DO 39 J=1,10
39 SOCNST(J,1)=SICNST(J,1)
   WRITE(6,10) SONUM(1,1)
10 FORMAT('O  STREAM NUMBER ',G14.7)
   DO 41 K=1,M
41 WRITE(6,12) K,SOCOMP(K,1)
12 FORMAT('  WT FRACTION COMP ',I2,' = ',G14.7)
59 WRITE(6,14)SOMOLE(1)
14 FORMAT('  TOTAL STREAM WEIGHT = ',G14.7)
   RETURN
   END
```

Figure 36

```

SUBROUTINE LECH
COMMON/SYSD/KEFLAG(50),KSFLAG(100),KTRACE,DERROR,NPFREQ,IPUNCH
COMMON/EQPA/ EQPAR(25,50), NEMAX,MAXEQP
COMMON/CONTL/NIN,NOUT,NOCOMP,NE,NEN
COMMON/STRMIN/SINUM(8,10),SICNST(10,8),SIMOLE(8),
1SICOMP(11,8),SIKV(10,8)
COMMON/STMOUT/SONUM(8,10),SOCNST(10,8),SOMOLE(8),
1SOCOMP(11,8),SOKV(10,8)
DIMENSION SICPWT(11,8),SOCPWT(11,8),Y(10),X(10),XF(2),RES(2),CA(2)
1,CF(2),R(2)
WRITE(6,4)
4 FORMAT('1 LEACH TANK')
M=NOCOMP-1
IF(EQPAR(8,NE).GT.1.E-06)GO TO 7
XA=1.
DO 3 K=1,M
3 SOCOMP(K,1)=SICOMP(K,1)
SOMOLE(1)=SIMOLE(1)
DO 5 J=1,10
5 SOCNST(J,1)=SICNST(J,1)
GO TO 99
7 DO 11 K=1,M
11 SICPWT(K,1)=SICOMP(K,1)*SIMOLE(1)
AR=SICNST(1,1)
BR=SICNST(2,1)
CC=SICNST(3,1)
XX=-.025
RAI=0.
DO 13 I=1,20
XX=XX+.05
Q=XX/(1.-CC*XX)
RM=AR*BR**Q
13 RAI=RAI+.05/RM
RA=1.0/RAI
KCI=EQPAR(2,NE)+.1

```

Figure 36 (contd.)

```
KC2=EQPAR(3,NE)+.1
KC3=EQPAR(4,NE)+.1
KC4=EQPAR(5,NE)+.1
ZNOIN=SICPWT(KC1,1)/81.38
ACIDIN=SICPWT(KC2,1)/98.08
FESDIN=SICPWT(KC3,1)/399.9
FEOHIN=SICPWT(KC4,1)/106.89
KC5=EQPAR(6,NE)+.1
VOL=SIMOLE(1)/EQPAR(7,NE)
CAO=2.*ACIDIN/VOL
CFO=2.*FESDIN/VOL
IF(FEOHIN.LE.0.) GO TO 30
EOC=CFO/(CAO**3)
IF(EOC.GE.2.535E08) GO TO 30
AA=0.
BB=1.
ERR=.001
XF(1)=AA+0.618034*(BB-AA)
XF(2)=AA+BB-XF(1)
DO 17 J=1,2
QF=CFO+1./3.*CAO*XF(J)
QFF=2.56E08*(CAO*(1.-XF(J)))*3
17 RES(J)=ABS(QFF-QF)
19 IF(RES(1).LE.RES(2)) GO TO 21
BB=XF(1)
XF(1)=XF(2)
RES(1)=RES(2)
XF(2)=AA+BB-XF(1)
QF=CFO+1./3.*CAO*XF(2)
QFF=2.56E08*(CAO*(1.-XF(2)))*3
RES(2)=ABS(QFF-QF)
GO TO 23
21 AA=XF(2)
XF(2)=XF(1)
RES(2)=RES(1)
```

Figure 36 (contd.)

```
XF(1)=AA+BB-XF(2)
QF=CFO+1./3.*CAO*XF(1)
QFF=2.56E08*(CAO*(1.-XF(1)))*3
RES(1)=ABS(QFF-QF)
23 IF((BB-AA).GT.ERR) GO TO 19
   IF(RES(2).LT.RES(1)) GO TO 25
   CAA=CAO*(1.-XF(1))
   CFF=CFO+1./3.*CAO*XF(1)
   FEOH=FEOHIN-(1./3.*CAO*XF(1))*VOL
   GO TO 27
25 CAA=CAO*(1.-XF(2))
   CFF=CFO+1./3.*CAO*XF(2)
   FEOH=FEOHIN-(1./3.*CAO*XF(2))*VOL
27 IF(FEOH.GE.0.) GO TO 29
   CFF=FEOHIN/VOL+CFO
   XFA=3.*(CFF-CFO)/CAO
   CAA=CAO*(1.-XFA)
   FEOH=0.
29 CAO=CAA
   CFO=CFF
   FEOHIN=FEOH
30 CONTINUE
   WRITE(6,701) CAO,CFO
701 FORMAT('0CAO = ',G14.7,' CFO = ',G14.7)
   ST=EQPAR(8,NE)/SIMOLE(1)
   ZNOR=1.0/2.*VOL*(CAO+3.*CFO)
   IF((ZNOIN-ZNOR).GE.0.) GO TO 91
   XFX=1.
71 XFX=XFX-.1
   CAX=CAO*XFX
   CFX=2.56E08*CAX**3
   IF(CFX.LE.CFO) GO TO 73
   CFX=CFO
73 ZNOR=1./2.*VOL*(CAO-CAX+3.*(CFO-CFX))
```

Figure 36 (contd.)

```
IF(ZNOR.LE.ZNOIN.AND.XFX.GT.0.) GO TO 71
XFX=XFX+.1
75 XFX=XFX-.01
   CAX=CA0*XFX
   CFX=2.56E08*CAX**3
   IF(CFX.LE.CF0) GO TO 77
   CFX=CF0
77 ZNOR=1./2.*VOL*(CA0-CAX+3.*(CF0-CFX))
   IF(ZNOR.LE.ZNOIN.AND.XFX.GT.0.) GO TO 75
   XFX=XFX+.01
79 XFX=XFX-.001
   CAX=CA0*XFX
   CFX=2.56E08*CAX**3
   IF(CFX.LE.CF0) GO TO 81
   CFX=CF0
81 ZNOR=1./2.*VOL*(CA0-CAX+3.*(CF0-CFX))
   IF(ZNOR.LE.ZNOIN.AND.XFX.GT.0.) GO TO 79
   XFX=XFX+.001
   AA=CA0*XFX
   GO TO 87
91 AA=0.
87 BB=CA0
   ERR=.01*(BB-AA)
   CA(1)=AA+0.618034*(BB-AA)
   CA(2)=AA+BB-CA(1)
   DO 33 J=1,2
   CF(J)=2.56E08*CA(J)**3
   IF(CF(J).LE.CF0) GO TO 31
   CF(J)=CF0
31 ZNOR=1./2.*VOL*(CA0-CA(J)+3.*(CF0-CF(J)))
   WRITE(6,702) CA(J),CF(J),ZNOIN,ZNOR
702 FORMAT('0CA = ',G14.7,' CF = ',G14.7,' ZNOIN = ',G14.7,
1' ZNOR = ',G14.7)
   R(J)=RA*((ZNOIN-ZNOR)/ZNOIN)**(1./3.)
```

Figure 36 (contd.)

```
RR=SQRT(RA**3*(CA0-CA(J)+3.*(CFO-CF(J)))/(EQPAR(9,NE)*ST*CA(J)
1*SICOMP(KC1,1)))
33 RES(J)=ABS(RR-R(J))
35 IF(RES(1).LE.RES(2)) GO TO 39
BB=CA(1)
CA(1)=CA(2)
CF(1)=CF(2)
R(1)=R(2)
RES(1)=RES(2)
CA(2)=AA+BB-CA(1)
CF(2)=2.56E08*CA(2)**3
IF(CF(2).LE.CFO) GO TO 37
CF(2)=CFO
37 ZNOR=1./2.*VOL*(CA0-CA(2)+3.*(CFO-CF(2)))
R(2)=RA*((ZNOIN-ZNOR)/ZNOIN)**(1./3.)
RR=SQRT(RA**3*(CA0-CA(2)+3.*(CFO-CF(2)))/(EQPAR(9,NE)*ST*CA(2)
1*SICOMP(KC1,1)))
RES(2)=ABS(RR-R(2))
GO TO 43
39 AA=CA(2)
CA(2)=CA(1)
CF(2)=CF(1)
R(2)=R(1)
RES(2)=RES(1)
CA(1)=AA+BB-CA(2)
CF(1)=2.56E08*CA(1)**3
IF(CF(1).LE.CFO) GO TO 41
CF(1)=CFO
41 ZNOR=1./2.*VOL*(CA0-CA(1)+3.*(CFO-CF(1)))
R(1)=RA*((ZNOIN-ZNOR)/ZNOIN)**(1./3.)
RR=SQRT(RA**3*(CA0-CA(1)+3.*(CFO-CF(1)))/(EQPAR(9,NE)*ST*CA(1)
1*SICOMP(KC1,1)))
RES(1)=ABS(RR-R(1))
43 IF((BB-AA).GT.ERR) GO TO 35
IF(RES(2).LT.RES(1)) GO TO 45
```

Figure 36 (contd.)

```
CAF=CA(1)
CFF=CF(1)
RF=R(1)
GO TO 47
45 CAF=CA(2)
CFF=CF(2)
RF=R(2)
47 XA=(RF**3)/(RA**3)
SOCPWT(KC1,1)=SICPWT(KC1,1)*XA
SOCPWT(KC2,1)=CAF*VOL*98.08/2.
SOCPWT(KC3,1)=CFF*VOL*399.9/2.
SOCPWT(KC4,1)=(FECHIN+(CFG-CFF)*VOL)*106.89
DO 49 K=1,M
L=K+9
IF(K.EQ.KC1.OR.K.EQ.KC2.OR.K.EQ.KC3.OR.K.EQ.KC4) GO TO 49
SOCPWT(K,1)=SICPWT(K,1)+EQPAR(L,NE)*(1.-XA)*SICPWT(KC1,1)
49 CONTINUE
DO 51 I=1,10
P=I
X(I)=P*.1
Q=X(I)/(1.-CC*X(I))
RM=AR*BR**Q
RN=RM*XA**(1./3.)
51 Y(I)=RN
CALL LSC(X,Y,10,A,B,C)
SOCNST(1,1)=A
SOCNST(2,1)=B
SOCNST(3,1)=C
WRITE(6,342) A,B,C
342 FORMAT(' A = ',G14.7,' B = ',G14.7,' C = ',G14.7)
SOMOLE(1)=0.
DO 53 K=1,M
53 SOMOLE(1)=SOMOLE(1)+SOCPWT(K,1)
SOCPWT(KC5,1)=SOCPWT(KC5,1)+SIMOLE(1)-SOMOLE(1)
```

Figure 36 (contd.)

```
SOMOLE(1)=SIMOLE(1)
DO 55 K=1,M
55 SOCOMP(K,1)=SOCPWT(K,1)/SOMOLE(1)
DO 59 J=4,10
59 SUCNST(J,1)=SICNST(J,1)
99 DO 57 J=3,10
57 SONUM(1,J)=SINUM(1,J)
WRITE(6,10) SONUM(1,1)
10 FORMAT('0 STREAM NUMBER ',G14.7)
DO 61 K=1,M
61 WRITE(6,12) K,SOCOMP(K,1)
12 FORMAT(' WT FRACTION COMP ',I2,' = ',G14.7)
WRITE(6,14)SOMOLE(1)
14 FORMAT(' TOTAL STREAM WEIGHT = ',G14.7)
XA=1.-XA
WRITE(6,18) XA
18 FORMAT('0 FRACTION OF ZNO LEACHED = ',G14.7)
RETURN
END
```

Figure 37

```

SUBROUTINE CLAS
COMMON/SYSD/KEFLAG(50),KSFLAG(100),KTRACE,DERROR,NPFREQ,IPUNCH
COMMON/EQPA/ EQPAR(25,50), NEMAX,MAXEQP
COMMON/CONTL/NIN,NOUT,NOCOMP,NE,NEN
COMMON/STRMIN/SINUM(8,10),SICNST(10,8),SIMOLE(8),
1SICOMP(11,8),SIKV(10,8)
COMMON/STMOU/SONUM(8,10),SOCNST(10,8),SOMOLE(8),
1SOCOMP(11,8),SOKV(10,8)
DIMENSION TWS(2),SICPWT(11,8),SOCPWT(11,8),X(10),Y(10)
WRITE(6,4)
4 FORMAT('1 CLASSIFIER')
M=NOCOMP-1
DO 143 J=1,NOUT
DO 141 K=1,M
141 SOCPWT(K,J)=0.
143 SUMOLE(J)=0.
DO 11 K=1,M
11 SICPWT(K,1)=SICOMP(K,1)*SIMOLE(1)
WIS=0.
NS=EQPAR(2,NE)+0.1
DO 15 I=1,NS
15 WIS=WIS+SICPWT(I,1)
AR=SICNST(1,1)
BR=SICNST(2,1)
CC=SICNST(3,1)
XF=0.
17 XF=XF+.1
Q=XF/(1.-CC*XF)
ROP=AR*BR**Q
IF(ROP.GE.EQPAR(3,NE)) GO TO 21
IF(XF.GE.1.) GO TO 19
GO TO 17
19 XF=1.
GO TO 25

```

Figure 37 (contd.)

```
21 XF=XF-.1
23 Q=XF/(1.-CC*XF)
   ROP=AR*BR**Q
   IF(ROP.GE.EQPAR(3,NE)) GO TO 25
   XF=XF+.01
   GO TO 23
25 WRITE(6,388) XF
388 FORMAT('  XF = ',G14.7)
   DO 37 J=1,2
   DO 27 I=1,10
   X(I)=0.
27 Y(I)=0.
   DO 35 I=1,10
   P=I
   IF(J.NE.1) GO TO 31
   XX=P*XF/10.
   GO TO 33
31 XX=XF+(1.-XF)/10.*P
33 Q=XX/(1.-CC*XX)
   RM=AR*BR**Q
   X(I)=P*.1
35 Y(I)=RM
   CALL LSC(X,Y,10,A,B,C)
   SOCNST(1,J)=A
   SOCNST(2,J)=B
   SOCNST(3,J)=C
37 WRITE(6,342) A,B,C
342 FORMAT('  A = ',G14.7,'  B = ',G14.7,'  C = ',G14.7)
   DO 45 J=1,2
   TWS(J)=0.
```

Figure 37 (contd.)

```
      IF(J.NE.1) GO TO 39
      XJ=XF
      GO TO 41
39  XJ=1.-XF
41  DO 43 I=1,NS
43  SOCPWT(I,J)=SICPWT(I,1)*XJ
45  TWS(J)=WIS*XJ
      SOMOLE(2)=TWS(2)/EQPAR(4,NE)
      SOMOLE(1)=SIMOLE(1)-SOMOLE(2)
      K=NS+1
      DO 47 J=1,2
      DO 47 I=K,M
47  SOCPWT(I,J) =SICPWT(I,1)*{SOMOLE(J)-TWS(J)}/{SIMOLE(1)-WIS}
      DO 51 J=1,2
      DO 211 I=3,10
211 SONUM(J,I)=SINUM(1,I)
      DO 213 I=4,10
213 SOCNST(I,J)=SICNST(I,1)
      WRITE(6,10) SONUM(J,1)
10  FORMAT('0  STREAM NUMBER ',G14.7)
      DO 49 K=1,M
      SOCOMP(K,J)=SOCPWT(K,J)/SOMOLE(J)
49  WRITE (6,12) K,SOCOMP(K,J)
12  FORMAT('  WT FRACTION COMP ',I2,' = ',G14.7)
51  WRITE (6,24) SOMOLE(J)
24  FORMAT('  TOTAL STREAM WEIGHT = ',G14.7)
      RETURN
      END
```

Figure 38

```

SUBROUTINE SETL
COMMON/SYSD/KEFLAG(50),KSFLAG(100),KTRACE,DERROR,NPFREQ,IPUNCH
COMMON/EQPA/ EQPAR(25,50), NEMAX,MAXEQP
COMMON/CONTL/NIN,NOOUT,NOCOMP,NE,NEN
COMMON/STRMIN/SINUM(8,10),SICNST(10,8),SIMOLE(8),
1SICOMP(11,8),SIKV(10,8)
COMMON/STMOUT/SONUM(8,10),SOCNST(10,8),SOMOLE(8),
1SUCOMP(11,8),SOKV(10,8)
DIMENSION SICPWT(11,8),SOCPWT(11,8)
WRITE(6,4)
4 FORMAT('1 THICKENER')
M=NOCOMP-1
DO 11 K=1,M
11 SICPWT(K,1)=SICOMP(K,1)*SIMOLE(1)
TWS=0.0
NS=EQPAR(2,NE)+0.1
DO 31 J=1,NS
31 TWS=TWS+SICPWT(J,1)
XI=TWS/SIMOLE(1)
SOMOLE(1)=SIMOLE(1)*(XI-EQPAR(4,NE))/(EQPAR(3,NE)-EQPAR(4,NE))
SOMOLE(2)=SIMOLE(1)-SOMOLE(1)
DO 39 I=1,2
DO 33 J=1,NS
33 SOCPWT(J,I)=SICPWT(J,1)*(SOMOLE(I)*EQPAR(I+2,NE))/(SIMOLE(1)*XI)
K=NS+1
DO 35 J=K,M
35 SOCPWT(J,1)=SICPWT(J,1)*(SOMOLE(1)*(1.0-EQPAR(I+2,NE))/(SIMOLE(1)*
1(1.0-XI)))
DO 37 K=1,10
37 SOCNST(K,I) =SICNST(K,1)
DO 39 J=3,10
39 SONUM(I,J)=SINUM(1,J)
DO 93 J=1,2
DO 65 K=1,M

```

Figure 38 (contd.)

```
65 SOCOMP(K,J)=SOCPWT(K,J)/SOMOLE(J)
   WRITE(6,10) SONUM(J,1)
10  FORMAT('0 STREAM NUMBER = ',G14.7)
   DO 41 K=1,M
   WRITE(6,32) K,SOCOMP(K,J)
32  FORMAT('   WT FRACTION COMP ',I2,' = ',G14.7)
41  CONTINUE
93  WRITE(6,49) SOMOLE(J)
49  FORMAT('   TOTAL WEIGHT STREAM = ',G14.7)
   RETURN
   END
```

Figure 39

```

SUBROUTINE FLTR
COMMON/SYSD/KEFLAG(50),KSFLAG(100),KTRACE,DERROR,NPFREQ,IPUNCH
COMMON/EQPA/ EQPAR(25,50), NEMAX,MAXEQP
COMMON/CONTL/NIN,NOUT,NOCOMP,NE,NEN
COMMON/STRMIN/SINUM(8,10),SICNST(10,8),SIMOLE(8),
1SICOMP(11,8),SIKV(10,8)
COMMON/STMOUT/SONUM(8,10),SOCNST(10,8),SOMOLE(8),
1SOCOMP(11,8),SOKV(10,8)
DIMENSION SICPWT(11,8),SOCPWT(11,8)
WRITE(6,4)
4 FORMAT('1 FILTER')
M=NOCOMP-1
DO 11 K=1,M
11 SICPWT(K,1)=SICOMP(K,1)*SIMOLE(1)
TWS=0.0
NS=EQPAR(2,NE)+0.1
DO 31 J=1,NS
31 TWS=TWS+SICPWT(J,1)
XI=TWS/SIMOLE(1)
SOMOLE(1)=SIMOLE(1)*(XI-EQPAR(4,NE))/(EQPAR(3,NE)-EQPAR(4,NE))
SOMOLE(2)=SIMOLE(1)-SOMOLE(1)
DO 39 I=1,2
DO 33 J=1,NS
33 SOCPWT(J,I)=SICPWT(J,1)*(SOMOLE(I)*EQPAR(I+2,NE))/(SIMOLE(1)*XI)
K=NS+1
DO 35 J=K,M
35 SOCPWT(J,1)=SICPWT(J,1)*(SOMOLE(1)*(1.0-EQPAR(I+2,NE))/(SIMOLE(1)*
1(1.0-XI)))
DO 37 K=1,10
37 SOCNST(K,1) =SICNST(K,1)
DO 39 J=3,10
39 SONUM(I,J)=SINUM(1,J)
DO 93 J=1,2
DO 65 K=1,M

```

Figure 39 (contd.)

```
65 SOCOMP(K,J)=SOCPWT(K,J)/SOMOLE(J)
   WRITE(6,10) SONUM(J,1)
10  FORMAT('0 STREAM NUMBER = ',G14.7)
   DO 41 K=1,NOCOMP
   WRITE(6,32) K,SOCOMP(K,J)
32  FORMAT(' WT FRACTION COMP ',I2,' = ',G14.7)
41  CONTINUE
93  WRITE(6,49) SOMOLE(J)
49  FORMAT(' TOTAL WEIGHT STREAM = ',G14.7)
   RETURN
   END
```

Figure 40

```

SUBROUTINE MIXR
COMMON/SYSD/KEFLAG(50),KSFLAG(100),KTRACE,DERROR,NPFREQ,IPUNCH
COMMON/EQPA/ EQPAR(25,50), NEMAX,MAXEQP
COMMON/CONTL/NIN,NOUT,NOCOMP,NE,NEN
COMMON/STRMIN/SINUM(8,10),SICNST(10,8),SIMOLE(8),
1SICOMP(11,8),SIKV(10,8)
COMMON/STMOUT/SONUM(8,10),SOCNST(10,8),SOMOLE(8),
1SOCOMP(11,8),SOKV(10,8)
DIMENSION SICPWT(11,8),SOCPWT(11,8),X(80),Y(80),R(10,8),WIS(8)
WRITE(6,4)
4 FORMAT('1 MIXER')
WF=0.
M=NOCOMP-1
DO 141 K=1,NOCOMP
141 SOCPWT(K,1)=0.
SOMOLE(1)=0.
SOCOMP(NOCOMP,1)=0.
KC=EQPAR(2,NE)+.1
NNN=EQPAR(3,NE)+.1
WOS=0.
DO 15 J=1,NIN
WIS(J)=0.
DO 11 K=1,M
11 SICPWT(K,J)=SICOMP(K,J)*SIMOLE(J)
DO 13 K=1,NNN
13 WIS(J)=WIS(J)+SICPWT(K,J)
SOCOMP(NOCOMP,1)=SOCOMP(NOCOMP,1)+SICOMP(NOCOMP,J)
15 WOS=WOS+WIS(J)
DO 19 K=1,M
DO 17 J=1,NIN
17 SOCPWT(K,1)=SOCPWT(K,1)+SICPWT(K,J)
19 SOMOLE(1)=SOMOLE(1)+SOCPWT(K,1)
DO 20 K=1,M
20 SOCOMP(K,1)=SOCPWT(K,1)/SOMOLE(1)

```

Figure 40 (contd.)

```
      IF(WDS.LE.1.E-06) GO TO 75
      DO 21 J=1,NIN
      AR=SICNST(1,J)
      BR=SICNST(2,J)
      CC=SICNST(3,J)
      DO 21 I=1,10
      P=I
      Z=P*.1
      Q=Z/(1.-CC*Z)
21  R(I,J)=AR*BR**Q
      K=0
23  MJ=1
      MI=1
      DO 25 J=1,NIN
      DO 25 I=1,10
      IF(R(I,J).GE.R(MI,MJ)) GO TO 25
      MI=I
      MJ=J
25  CONTINUE
      IF(R(MI,MJ).LE.0.) GO TO 35
      IF(R(MI,MJ).GE.1.E05) GO TO 27
      WFZ=WIS(MJ)*.10/WDS
      IF(WFZ.LT.1.E-04) GO TO 35
      WF=WF+WIS(MJ)*.10/WDS
      K=K+1
      X(K)=WF
      Y(K)=R(MI,MJ)
      WRITE(6,701) K,X(K),Y(K)
701  FORMAT(' K = ',I3,' X = ',G14.7,' Y = ',G14.7)
      35  R(MI,MJ)=1.E06
      GO TO 23
27  CALL LSC(X,Y,K,A,B,C)
      SOCNST(1,1)=A
      SOCNST(2,1)=B
```

Figure 40 (contd.)

```
      SOCNST(3,1)=C
      WRITE(6,342) A,B,C
342  FORMAT('  A = ',G14.7,'  B = ',G14.7,'  C = ',G14.7)
      TKS=0.
      CONST=0.
      DO 29 J=1,NIN
      CONST=CONST+SICNST(5,J)*WIS(J)
29   TKS=TKS+SINUM(J,3)*WIS(J)
      SOCNST(5,1)=CONST/WOS
      SONUM(1,3)=TKS/WOS
      TLL=0.
      UL=0.
      DO 63 J=1,NIN
      IF(SIMOLE(J).LE.1.E-04) GO TO 63
      IF(TLL.GT.1.E-05) GO TO 61
      TLL=SICNST(4,J)
      GO TO 63
61   IF(SICNST(4,J).GT.TLL) GO TO 63
      TLL=SICNST(4,J)
63   CONTINUE
      DO 67 J=1,NIN
      IF(SIMOLE(J).LE.1.E-04) GO TO 67
      IF(UL.GT.1.E-05) GO TO 65
      UL=SICNST(4,J)
      GO TO 67
65   IF(SICNST(4,J).LT.UL) GO TO 67
      UL=SICNST(4,J)
67   CONTINUE
      IF(UL.NE.TLL) GO TO 69
      AHT=UL
      GO TO 71
69   CALL TGS(TLL,UL,SOCNST(5,1),SONUM(1,3),A,B,C,AHT)
71   SOCNST(4,1)=AHT
      WRITE(6,503) SOCNST(4,1)
```

Figure 40 (contd.)

```
503 FORMAT(' HT = ',G14.7)
75 CONTINUE
DO 31 I=4,10
31 SUNUM(1,I)=SINUM(1,I)
WRITE(6,10) SONUM(1:1)
10 FORMAT('0 STREAM NUMBER ',G14.7)
DO 53 K=1,M
53 WRITE (6,12) K,SOCOMP(K,1)
12 FORMAT(' WT FRACTION COMP ',I2,' = ',G14.7)
WRITE (6,24) SOMOLE(1)
24 FORMAT(' TOTAL STREAM WEIGHT = ',G14.7)
WRITE(6,26) SOCNST(5,1)
26 FORMAT(' AVERAGE CONVERSION = ',G14.7)
RETURN
END
```

Figure 41

```

SUBROUTINE ELET
COMMON/SYSD/KEFLAG(50),KSFLAG(100),KTRACE,DError,NPFREQ,IPUNCH
COMMON/EQPA/ EQPAR(25,50), NEMAX,MAXEQP
COMMON/CONTL/NIN,NOUT,NOCOMP,NE,NEN
COMMON/STRMIN/SINUM(8,10),SICNST(10,8),SIMOLE(8),
1SICOMP(11,8),SIKV(10,8)
COMMON/STMOUT/SONUM(8,10),SOCNST(10,8),SOMOLE(8),
1SOCOMP(11,8),SOKV(10,8)
DIMENSION SICPWT(11,8),SOCPWT(11,8)
M=NOCOMP-1
DO 11 K=1,M
11 SICPWT(K,1)=SICOMP(K,1)*SIMOLE(1)
WRITE(6,4)
4 FORMAT('1 ELECTROLYTIC PLANT')
KC1=EQPAR(2,NE)+.1
EFF=EQPAR(3,NE)
WZNP=SICPWT(KC1,1)*EFF*65.38/161.44
WRITE(6,6) WZNP
6 FORMAT('0 TONS/DAY ZINC PRODUCED = ',G14.7)
SOCPWT(KC1,1)=SICPWT(KC1,1)*(1.-EFF)
AMP=WZNP*68.3
CAMP=AMP/EQPAR(4,NE)
WRITE(6,8) CAMP
8 FORMAT('0 CELL CURRENT (AMPS) = ',G14.7)
SOMOLE(1)=0.
DO 13 K=1,M
L=K+4
IF(K.EQ.KC1) GO TO 13
SOCPWT(K,1)=SICPWT(K,1)+EQPAR(L,NE)*SICPWT(KC1,1)*EFF
13 SOMOLE(1)=SOMOLE(1)+SOCPWT(K,1)
DO 15 K=1,M
15 SOCOMP(K,1)=SOCPWT(K,1)/SOMOLE(1)
WRITE(6,10) SONUM(1,1)
10 FORMAT('0 STREAM NUMBER = ',G14.7)

```

Figure 41 (contd.)

```
      DO 17 K=1,M
17 WRITE(6,12) K,SOCOMP(K,1)
12 FORMAT('    WT FRACTION COMP ',I2,' = ',G14.7)
      WRITE(6,14) SOMOLE(1)
14 FORMAT('    TOTAL WEIGHT STREAM = ',G14.7)
      RETURN
      END
```

Computing Subroutine Printouts

Figure 42

```
SUBROUTINE CONV(AR,B,CC,CK,TH,AVC)
X=-.025
AVC=0.
DO 99 J=1,20
X=X+.05
Q=X/(1.-CC*X)
RM=AR*B**Q
DO 11 I=1,10
P=I
RC=P*RM*.09999
VRC=RM**3-RC**3-CK*TH*(RM*RC/(RM-RC))
IF(VRC.LE.0.) GO TO 13
11 CONTINUE
13 A=RC-.09999*RM
DO 15 I=1,10
P=I
RC=A+P*.009999*RM
VRC=RM**3-RC**3-CK*TH*(RM*RC/(RM-RC))
IF(VRC.LE.0.) GO TO 17
15 CONTINUE
17 A=RC-.009999*RM
DO 19 I=1,10
P=I
RC=A+P*.0009999*RM
VRC=RM**3-RC**3-CK*TH*(RM*RC/(RM-RC))
IF(VRC.LE.0.) GO TO 21
19 CONTINUE
21 CONV=1.-RC**3/RM**3
AVC = AVC+CONV*.05
99 CONTINUE
RETURN
END
```

Figure 43

```

SUBROUTINE LSC(Z,R,N,A,B,C)
DIMENSION Z(1),R(1),RES(2),CC(2),TA(2),TB(2),X(3),Y(3)
AA=0.
BB=1.
ERR=.01
CC(1)=AA+0.618034*(BB-AA)
CC(2)=AA+BB-CC(1)
DO 19 J=1,2
DO 11 I=1,3
X(I)=0.
11 Y(I)=0.
DO 15 I=1,N
Q=Z(I)/(1.-CC(J)*Z(I))
X(1)=X(1)+Q
X(2)=X(2)+Q**2
X(3)=X(3)+1.
Y(1)=Y(1)+ALOG(R(I))
15 Y(2)=Y(2)+ALOG(R(I))*Q
CALL SIME(X,Y)
TA(J)=EXP(Y(1))
TB(J)=EXP(Y(2))
RES(J)=0.
DO 17 I=1,N
Q=Z(I)/(1.-CC(J)*Z(I))
RC=TA(J)*TB(J)**Q
17 RES(J)=RES(J)+(R(I)-RC)**2
19 CONTINUE
21 IF(RES(1).LE.RES(2)) GO TO 29
BB=CC(1)
CC(1)=CC(2)
RES(1)=RES(2)
RES(2)=0.
CC(2)=AA+BB-CC(1)

```

Figure 43 (contd.)

```
      DO 23 I=1,3
      X(I)=0.
23  Y(I)=0.
      DO 25 I=1,N
      Q=Z(I)/(1.-CC(2)*Z(I))
      X(1)=X(1)+Q
      X(2)=X(2)+Q**2
      X(3)=X(3)+1.
      Y(1)=Y(1)+ALOG(R(I))
25  Y(2)=Y(2)+ALOG(R(I))*Q
      CALL SIME (X,Y)
      TA(2)=EXP(Y(1))
      TB(2)=EXP(Y(2))
      DO 27 I=1,N
      Q=Z(I)/(1.-CC(2)*Z(I))
      RC=TA(2)*TB(2)**Q
27  RES(2)=RES(2)+(R(I)-RC)**2
      GO TO 37
29  AA=CC(2)
      CC(2)=CC(1)
      RES(2)=RES(1)
      RES(1)=0.
      CC(1)=AA+BB-CC(2)
      DO 31 I=1,3
      X(I)=0.
31  Y(I)=0.
      DO 33 I=1,N
      Q=Z(I)/(1.-CC(1)*Z(I))
      X(1)=X(1)+Q
      X(2)=X(2)+Q**2
      X(3)=X(3)+1.
      Y(1)=Y(1)+ALOG(R(I))
33  Y(2)=Y(2)+ALOG(R(I))*Q
```

Figure 43 (contd.)

```
      CALL SIME(X,Y)
      TA(1)=EXP(Y(1))
      TB(1)=EXP(Y(2))
      DO 35 I=1,N
      Q=Z(I)/(1.-CC(1)*Z(I))
      RC=TA(1)*TB(1)**Q
35  RES(1)=RES(1)+(R(I)-RC)**2
37  IF((BB-AA).GT.ERR) GO TO 21
      IF(RES(2).LT.RES(1)) GO TO 39
      A=TA(1)
      B=TB(1)
      C=CC(1)
      RETURN
39  A=TA(2)
      B=TB(2)
      C=CC(2)
      RETURN
      END
```

Figure 44

```
SUBROUTINE SIME(X,Y)
  DIMENSION X(1),Y(1)
  A=(X(2)*Y(1)-X(1)*Y(2))/(X(3)*X(2)-X(1)**2)
  B=(X(3)*Y(2)-X(1)*Y(1))/(X(3)*X(2)-X(1)**2)
  Y(1)=A
  Y(2)=B
  RETURN
END
```

Figure 45

```
SUBROUTINE TGS(TLL,UL,AVC,CK,A,B,C,AHT)
DIMENSION RES(2),CVC(2),X(2)
ERR=(UL-TLL)*.01
X(1)=TLL+0.618034*(UL-TLL)
X(2)=TLL+UL-X(1)
DO 7 I=1,2
CALL CONVT(A,B,C,CK,X(I),CVC(I))
7 RES(I)=ABS((CVC(I)-AVC)/AVC)
21 IF(RES(1).LE.RES(2)) GO TO 29
UL=X(1)
X(1)=X(2)
RES(1)=RES(2)
X(2)=TLL+UL-X(1)
CALL CONVT(A,B,C,CK,X(2),CVC(2))
RES(2)=ABS((CVC(2)-AVC)/AVC)
GO TO 37
29 TLL=X(2)
X(2)=X(1)
RES(2)=RES(1)
X(1)=TLL+UL-X(2)
CALL CONVT(A,B,C,CK,X(1),CVC(1))
RES(1)=ABS((CVC(1)-AVC)/AVC)
37 IF((UL-TLL).GT.ERR) GO TO 21
IF(RES(2).LT.RES(1)) GO TO 39
AHT=X(1)
RETURN
39 AHT=X(2)
RETURN
END
```

Sample Problem Printouts

Figure 46

CLEAN

ZINC PLANT SIMULATION ROASTING SECTION DATA

&PMLIST

KPM1=1,'ROST','R-1 ',1,-2,-3,4*0,
KPM2=2,'MIXR','M-1 ',3,17,-4,4*0,
KPM3=3,'SCRN','S-1 ',4,-5,-6,4*0,
KPM4=4,'GRND','G-1 ',6,-14,5*0,
KPM5=5,'MIXR','M-2 ',2,16,-7,4*0,
KPM6=6,'PREP','F-1 ',7,-8,-9,4*0,
KPM7=7,'PREP','P-1 ',8,-10,-11,4*0,
KPM8=8,'PREP','F-2 ',10,-12,-13,4*0,
KPM9=9,'MIXR','M-3 ',9,11,13,14,-15,2*0,
KPM10=10,'ROST','R-2 ',15,-16,-17,4*0,
NOCOMP=6,COMPNT=1,2,3,4,5,6,14*0,

&END

&EQLIST

EQP1=1.,1.,65681.,1.,1475.,6.41E-08,3.E05,86.,.9825,1.19,1.017,-1.,
.835,-.1148,.0764,10*0.,
EQP2=2.,1.,5.,22*0.,
EQP3=3.,1.,.0937,22*0.,
EQP4=4.,1.E-03,1.5E02,22*0.,
EQP5=5.,1.,5.,22*0.,
EQP6=6.,1.,396.,8.E05,-1.,.835,-.1148,.0764,17*0.,
EQP7=7.,1.,396.,5.E06,-1.,.835,-.1148,.0764,17*0.,
EQP8=8.,1.,396.,1.5E06,-1.,.835,-.1148,.0764,17*0.,
EQP9=9.,1.,5.,22*0.,
EQP10=10.,1.,4266.,1.0,1475.,2.17E-07,.80E05,24.6,.9825,1.19,1.000,-1.,
.835,-.1148,.0764,10*0.,
ENAME=50*0,

&END

&SEXLIST

SEX1=1.,1.20496E-03,1.12046,.949301,7*0.,350.,.784,-0,.0899,.0,.1261,
6*0.,

Figure 46 (contd.)

```
SNAME=2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,84*0,  
&END  
&SINLST  
SIN1=1.,1.,.00096,7*0,  
SIN5=5.,2.,8*0.,  
SIN12=12.,2.,8*0.,  
SNAME=2,3,4,6,7,8,9,10,11,13,14,15,16,17,86*0,  
&END  
&KELIST  
KE2=2,3,4,5,6,7,8,9,10,41*0,LOOPS=10,DERROR=0.05,KE4=5,9*0,  
&END
```

Figure 47

CLEAN

ZINC PLANT SIMULATION LEACHING SECTION DATA

&PMLIST

KPM1=1,'AMIX','AM-1',1,2,-3,4*0,
KPM2=2,'LECH','L-1 ',3,-4,5*0,
KPM3=3,'LECH','L-2 ',4,-5,5*0,
KPM4=4,'LECH','L-3 ',5,-6,5*0,
KPM5=5,'LECH','L-4 ',6,-7,5*0,
KPM6=6,'LECH','L-5 ',7,-8,5*0,
KPM7=7,'LECH','L-6 ',8,-9,5*0,
KPM8=8,'LECH','L-7 ',9,-10,5*0,
KPM9=9,'LECH','L-8 ',10,-11,5*0,
KPM10=10,'CLAS','C-1 ',11,-12,-13,4*0,
KPM11=11,'CLAS','C-2 ',13,-14,-15,4*0,
KPM12=12,'MIXR','M-1 ',12,14,-16,4*0,
KPM13=13,'SETL','S-1 ',16,-17,-18,4*0,
KPM14=14,'AMIX','AM-2',15,41,-19,4*0,
KPM15=15,'GRND','BM-1',19,-20,5*0,
KPM16=16,'MIXR','LM-2',18,20,-21,4*0,
KPM17=17,'AMIX','AM-3',21,40,-22,4*0,
KPM18=18,'LECH','AL13',22,-23,5*0,
KPM19=19,'LECH','AL12',23,-24,5*0,
KPM20=20,'LECH','AL11',24,-25,5*0,
KPM21=21,'LECH','AL10',25,-26,5*0,
KPM22=22,'LECH','AL-9',26,-27,5*0,
KPM23=23,'LECH','AL-8',27,-28,5*0,
KPM24=24,'LECH','AL-7',28,-29,5*0,
KPM25=25,'LECH','AL-6',29,-30,5*0,
KPM26=26,'LECH','AL-5',30,-31,5*0,
KPM27=27,'LECH','AL-4',31,-32,5*0,
KPM28=28,'LECH','AL-3',32,-33,5*0,

Figure 47 (contd.)

```
KPM29=29,'LECH','AL-2',33,-34,5*0,
KPM30=30,'LECH','AL-1',34,-35,5*0,
KPM31=31,'SETL','AL-S',35,-36,-37,4*0,
KPM32=32,'FLTR','F-AO',37,-38,-39,4*0,
KPM33=33,'MIXR','M-2 ',36,38,-2,4*0,
KPM34=34,'ELET','ELET',17,-40,5*0,
NOCOMP=11,COMPNT=1,2,3,4,5,6,7,8,9,10,11,9*0,
&END
&EQLIST
EQP1=1.,2.,7.,8.,10.,6.,.0025,.200,.2,16*0.,
EQP2=2.,2.,7.,8.,6.,10.,112.3,300.,0.6E01,0.,-1.,3*0.,3*0.,1.973,7*0.,
EQP3=3.,2.,7.,8.,6.,10.,112.3,300.,0.6E01,0.,-1.,3*0.,3*0.,1.973,7*0.,
EQP4=4.,2.,7.,8.,6.,10.,112.3,300.,0.6E01,0.,-1.,3*0.,3*0.,1.973,7*0.,
EQP5=5.,2.,7.,8.,6.,10.,112.3,300.,0.6E01,0.,-1.,3*0.,3*0.,1.973,7*0.,
EQP6=6.,2.,7.,8.,6.,10.,112.3,300.,0.6E01,0.,-1.,3*0.,3*0.,1.973,7*0.,
EQP7=7.,2.,7.,8.,6.,10.,112.3,300.,0.6E01,0.,-1.,3*0.,3*0.,1.973,7*0.,
EQP8=8.,2.,7.,8.,6.,10.,112.3,300.,0.6E01,0.,-1.,3*0.,3*0.,1.973,7*0.,
EQP10=10.,6.,1.21E-03,.20,21*0.,
EQP11=11.,6.,1.370E-03,.9,21*0.,
EQP12=12.,6.,23*0.,
EQP13=13.,6.,.00,.60,21*0.,
EQP14=14.,2.,7.,8.,10.,6.,0.00,0.,1.,16*0.,
EQP15=15.,1.E-03,7.5E02,22*0.,
EQP16=16.,6.,23*0.,
EQP17=17.,2.,7.,8.,10.,6.,1.60,0.,.3,16*0.,
EQP18=18.,2.,7.,8.,6.,10.,112.3,300.,0.2E-01,0.,-1.,3*0.,3*0.,1.973,
7*0.,
EQP19=19.,2.,7.,8.,6.,10.,112.3,300.,0.2E-01,0.,-1.,3*0.,3*0.,1.973,
7*0.,
EQP20=20.,2.,7.,8.,6.,10.,112.3,300.,0.2E-01,0.,-1.,3*0.,3*0.,1.973,
7*0.,
EQP21=21.,2.,7.,8.,6.,10.,112.3,300.,0.2E-01,0.,-1.,3*0.,3*0.,1.973,
7*0.,
```

Figure 47 (contd.)

```
EQP22=22.,2.,7.,8.,6.,10.,112.3,300.,0.2E-01,0.,-1.,3*0.,3*0.,1.973,
7*0.,
EQP23=23.,2.,7.,8.,6.,10.,112.3,300.,0.2E-01,0.,-1.,3*0.,3*0.,1.973,
7*0.,
EQP24=24.,2.,7.,8.,6.,10.,112.3,300.,0.2E-01,0.,-1.,3*0.,3*0.,1.973,
7*0.,
EQP25=25.,2.,7.,8.,6.,10.,112.3,300.,0.2E-01,0.,-1.,3*0.,3*0.,1.973,
7*0.,
EQP26=26.,2.,7.,8.,6.,10.,112.3,300.,0.2E-01,0.,-1.,3*0.,3*0.,1.973,
7*0.,
EQP27=27.,2.,7.,8.,6.,10.,112.3,300.,0.2E-01,0.,-1.,3*0.,3*0.,1.973,
7*0.,
EQP28=28.,2.,7.,8.,6.,10.,112.3,300.,0.2E-01,0.,-1.,3*0.,3*0.,1.973,
7*0.,
EQP29=29.,2.,7.,8.,6.,10.,112.3,300.,0.2E-01,0.,-1.,3*0.,3*0.,1.973,
7*0.,
EQP30=30.,2.,7.,8.,6.,10.,112.3,300.,0.2E-01,0.,-1.,3*0.,3*0.,1.973,
7*0.,
EQP31=31.,6.,.00,.50,21*0.,
EQP32=32.,6.,.00,1.0,21*0.,
EQP33=33.,6.,23*0.,
EQP34=34.,9.,.5,.9,6*0.,.607,0.,-1.,-.1117,11*0.,
ENAME=50*0,
&END
&SEXLIST
SEX1=1.,1.117E-03,1.271379,.9543234,7*0.,292.4,.1427344E-01,.7628663,
.3622894E-02,.6979978E-01,.1494384,6*0.,
SEX2=2.,10*0.,386.,8*0.,1.,2*0.,
SEX40=40.,10*0.,304.,8*0.,1.,2*0.,
SNAME=3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,
26,27,28,29,30,31,32,33,34,35,36,37,38,39,41,62*0,
&END
```

Figure 47 (contd.)

&SINLST

SIN1=1.,1.,8*0.,

SIN39=39.,2.,8*0.,

SIN41=41.,1.,8*0.,

SNAME=2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,
26,27,28,29,30,31,32,33,34,35,36,37,38,40,62*0,

&END

&KELIST

KE2=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,
26,27,28,29,30,31,32,33,34,16*0,LOOPS=05,DERROR=.05,

&END

Figure 48

ZINC PLANT SIMULATION ROASTING SECTION RESULTS

'PROCESS VECTORS'

NUMBER	EQUIPMENT SUBROUTINE	NAME	STREAM NUMBERS				
1	ROST	R-1	1	-2	-3	0	0
2	MIXR	M-1	3	17	--	0	0
3	SCRN	S-1	4	-5	-6	0	0
4	GRND	G-1	6	-14	0	0	0
5	MIXR	M-2	2	16	-7	0	0
6	PREP	F-1	7	-8	-9	0	0
7	PREP	P-1	8	-10	-11	0	0
8	PREP	F-2	10	-12	-13	0	0
9	MIXR	M-3	9	11	13	14	-15
10	ROST	R-2	15	-16	-17	0	0

Figure 48 (contd.)

ZINC PLANT SIMULATION ROASTING SECTION

'STREAM CONNECTIONS'

STREAM	EQUIPMENT	
	FROM	TO
1	0	1
2	1	5
3	1	2
4	2	3
5	3	0
6	3	4
7	5	6
8	6	7
9	6	9
10	7	8
11	7	9
12	8	0
13	8	9
14	4	9
15	9	10
16	10	5
17	10	2

Figure 48 (contd.)

ZINC PLANT SIMULATION ROASTING SECTION

'OTHER SYSTEM VARIABLES'

NUMBER OF COMPONENTS	6	
NUMBER OF ITEMS IN RECYCLE LIST (KE2)	9	
NUMBER OF ITEMS IN RECYCLE LIST (KE3)	0	
COMPONENT NUMBERS USED	1, 2, 3, 4, 5, 6,	
RECYCLE LIST KE2	2, 3, 4, 5, 6, 7, 8, 9, 10,	
STREAMS USED IN CONV. ROUTINE (KE4)	5,	
TOLERANCE, 'DERROR'	0.0500	206
MAX. LOOPS IN RECYCLE CALC.	10	

Figure 48 (contd.)

ROASTER

GAS FLOW OVERHEAD (CFM) = 65680.94

STREAM NUMBER 2.000000

WT FRACTION COMP 1 = 0.1947530

WT FRACTION COMP 2 = 0.5853676

WT FRACTION COMP 3 = 0.2223969E-01

WT FRACTION COMP 4 = 0.5355933E-01

WT FRACTION COMP 5 = 0.1440808

TOTAL STREAM WEIGHT = 58.20093

AVERAGE CONVERSION = 0.7825913

STREAM NUMBER 3.000000

WT FRACTION COMP 1 = 0.4025484E-01

WT FRACTION COMP 2 = 0.7388481

WT FRACTION COMP 3 = 0.4499458E-02

WT FRACTION COMP 4 = 0.6760228E-01

WT FRACTION COMP 5 = 0.1487952

TOTAL STREAM WEIGHT = 240.2586

AVERAGE CONVERSION = 0.9564861

MIXER

STREAM NUMBER 4.000000

WT FRACTION COMP 1 = 0.3183531E-01

WT FRACTION COMP 2 = 0.7492753

WT FRACTION COMP 3 = 0.3558355E-02

WT FRACTION COMP 4 = 0.6855631E-01

WT FRACTION COMP 5 = 0.1467761

TOTAL STREAM WEIGHT = 303.8013

AVERAGE CONVERSION = 0.9655874

Figure 48 (contd.)

SCREEN

STREAM NUMBER 5.000000
WT FRACTION COMP 1 = 0.1427344E-01
WT FRACTION COMP 2 = 0.7628663
WT FRACTION COMP 3 = 0.3622894E-02
WT FRACTION COMP 4 = 0.6979978E-01
WT FRACTION COMP 5 = 0.1494384
TOTAL STREAM WEIGHT = 292.4209
AVERAGE CONVERSION = 0.9848459

STREAM NUMBER 6.000000

WT FRACTION COMP 1 = 0.4767834
WT FRACTION COMP 2 = 0.4049237
WT FRACTION COMP 3 = 0.1923006E-02
WT FRACTION COMP 4 = 0.3704922E-01
WT FRACTION COMP 5 = 0.7932079E-01
TOTAL STREAM WEIGHT = 11.24337
AVERAGE CONVERSION = 0.4632967E-01

BALLMILL

STREAM NUMBER 14.00000
WT FRACTION COMP 1 = 0.4767834
WT FRACTION COMP 2 = 0.4049237
WT FRACTION COMP 3 = 0.1923006E-02
WT FRACTION COMP 4 = 0.3704922E-01
WT FRACTION COMP 5 = 0.7932079E-01
TOTAL STREAM WEIGHT = 11.24337

MIXER

STREAM NUMBER 7.000000
WT FRACTION COMP 1 = 0.1337095
WT FRACTION COMP 2 = 0.6492445
WT FRACTION COMP 3 = 0.1514918E-01
WT FRACTION COMP 4 = 0.5940391E-01
WT FRACTION COMP 5 = 0.1424931
TOTAL STREAM WEIGHT = 85.44159
AVERAGE CONVERSION = 0.8504454

Figure 48 (contd.)

PRECIPITATOR

STREAM NUMBER 8.000000
WT FRACTION COMP 1 = 0.1326493
WT FRACTION COMP 2 = 0.6502986
WT FRACTION COMP 3 = 0.1502741E-01
WT FRACTION COMP 4 = 0.5950038E-01
TOTAL STREAM WEIGHT = 54.41214
AVERAGE CONVERSION = 0.8516642

STREAM NUMBER 9.000000
WT FRACTION COMP 1 = 0.1348186
WT FRACTION COMP 2 = 0.6481422
WT FRACTION COMP 3 = 0.1527654E-01
WT FRACTION COMP 4 = 0.5930304E-01
WT FRACTION COMP 5 = 0.1424600
TOTAL STREAM WEIGHT = 31.02451
AVERAGE CONVERSION = 0.8491700
TOTAL EFF = 0.3630238

PRECIPITATOR

STREAM NUMBER 10.00000
WT FRACTION COMP 1 = 0.1281393
WT FRACTION COMP 2 = 0.6547818
WT FRACTION COMP 3 = 0.1450947E-01
WT FRACTION COMP 4 = 0.5991061E-01
WT FRACTION COMP 5 = 0.1426592
TOTAL STREAM WEIGHT = 3.331538
AVERAGE CONVERSION = 0.8568426

STREAM NUMBER 11.00000
WT FRACTION COMP 1 = 0.1327724
WT FRACTION COMP 2 = 0.6501761
WT FRACTION COMP 3 = 0.1504156E-01
WT FRACTION COMP 4 = 0.5948918E-01
WT FRACTION COMP 5 = 0.1425211
TOTAL STREAM WEIGHT = 51.07874
AVERAGE CONVERSION = 0.8515227
TOTAL EFF = 0.9387144

Figure 48 (contd.)

PRECIPITATOR

STREAM NUMBER 12.00000

WT FRACTION COMP 1 = 0.1267599
WT FRACTION COMP 2 = 0.6561542
WT FRACTION COMP 3 = 0.1435106E-01
WT FRACTION COMP 4 = 0.6003617E-01
WT FRACTION COMP 5 = 0.1427006
TOTAL STREAM WEIGHT = 1.453330
AVERAGE CONVERSION = 0.8584247

STREAM NUMBER 13.00000

WT FRACTION COMP 1 = 0.1291199
WT FRACTION COMP 2 = 0.6538072
WT FRACTION COMP 3 = 0.1462210E-01
WT FRACTION COMP 4 = 0.5982143E-01
WT FRACTION COMP 5 = 0.1426300
TOTAL STREAM WEIGHT = 1.878169
AVERAGE CONVERSION = 0.8557175
TOTAL EFF = 0.5636395

MIXER

STREAM NUMBER 15.00000

WT FRACTION COMP 1 = 0.1739850
WT FRACTION COMP 2 = 0.6206276
WT FRACTION COMP 3 = 0.1356090E-01
WT FRACTION COMP 4 = 0.5678556E-01
WT FRACTION COMP 5 = 0.1350412
TOTAL STREAM WEIGHT = 95.22472
AVERAGE CONVERSION = 0.7502980

Figure 48 (contd.)

ROASTER

GAS FLOW OVERHEAD (CFM) = 4265.996

STREAM NUMBER 16.00000

WT FRACTION COMP 1 = 0.3279337E-02

WT FRACTION COMP 2 = 0.7857931

WT FRACTION COMP 3 = 0.0000000

WT FRACTION COMP 4 = 0.7189769E-01

WT FRACTION COMP 5 = 0.1390306

TOTAL STREAM WEIGHT = 27.74767

AVERAGE CONVERSION = 0.9954286

STREAM NUMBER 17.00000

WT FRACTION COMP 1 = 0.6415503E-06

WT FRACTION COMP 2 = 0.7887595

WT FRACTION COMP 3 = 0.0000000

WT FRACTION COMP 4 = 0.7216913E-01

WT FRACTION COMP 5 = 0.1390710

TOTAL STREAM WEIGHT = 64.72583

AVERAGE CONVERSION = 0.9999991

Figure 49

ZINC PLANT SIMULATION LEACHING SECTION RESULTS

'PROCESS VECTORS'

NUMBER	EQUIPMENT SUBROUTINE	NAME	STREAM NUMBERS		
1	AMIX	AM-1	1	2	-3
2	LECH	L-1	3	4	0
3	LECH	L-2	4	5	0
4	LECH	L-3	5	6	0
5	LECH	L-4	6	7	0
6	LECH	L-5	7	8	0
7	LECH	L-6	8	9	0
8	LECH	L-7	9	10	0
9	LECH	L-8	10	11	0
10	CLAS	C-1	11	12	-13
11	CLAS	C-2	13	14	-15
12	MIXR	M-1	12	14	-16
13	SETL	S-1	16	17	-18
14	AMIX	AM-2	15	41	-19
15	GRND	BM-1	19	20	0
16	MIXR	LM-2	18	20	-21
17	AMIX	AM-3	21	40	-22

Figure 49 (contd.)

18	LECH	AL13	22 -23	0
19	LECH	AL12	23 -24	0
20	LECH	AL11	24 -25	0
21	LECH	AL10	25 -26	0
22	LECH	AL-9	26 -27	0
23	LECH	AL-8	27 -28	0
24	LECH	AL-7	28 -29	0
25	LECH	AL-6	29 -30	0
26	LECH	AL-5	30 -31	0
27	LECH	AL-4	31 -32	0
28	LECH	AL-3	32 -33	0
29	LECH	AL-2	33 -34	0
30	LECH	AL-1	34 -35	0
31	SETL	AL-S	35 -36	-37
32	FLTR	F-AO	37 -38	-39
33	MIXR	M-2	36 38	-2
34	ELET	ELET	17 -40	0

Figure 49 (contd.)

ZINC PLANT SIMULATION LEACHING SECTION

'STREAM CONNECTIONS'

STREAM	EQUIPMENT	
	FROM	TO
1	0	1
2	33	1
3	1	2
4	2	3
5	3	4
6	4	5
7	5	6
8	6	7
9	7	8
10	8	9
11	9	10
12	10	12
13	10	11
14	11	12
15	11	14
16	12	13
17	13	34

Figure 49 (contd.)

18	13	16
19	14	15
20	15	16
21	16	17
22	17	18
23	18	19
24	19	20
25	20	21
26	21	22
27	22	23
28	23	24
29	24	25
30	25	26
31	26	27
32	27	28
33	28	29
34	29	30
35	30	31
36	31	33
37	31	32
38	32	33

216

Figure 49 (contd.)

39	32	0
40	34	17
41	0	14

Figure 49 (contd.)

ZINC PLANT SIMULATION LEACHING SECTION

'OTHER SYSTEM VARIABLES'

NUMBER OF COMPONENTS	11
NUMBER OF ITEMS IN RECYCLE LIST (KE2)	34
NUMBER OF ITEMS IN RECYCLE LIST (KE3)	0
COMPONENT NUMBERS USED	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,
RECYCLE LIST KE2	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, 26, 27, 28, 29, 30, 31, 32, 33, 34,
TOLERANCE, 'DERROR'	0.0500
MAX. LOOPS IN RECYCLE CALC.	5

Figure 49 (contd.)

LEACH SLURRY TANK

TONS/DAY MAKEUP ACID ACCED = 0.1872117

TONS/DAY MAKEUP $\text{Fe}_2(\text{SO}_4)_3$ ADDED = 0.7049561E-02

STREAM NUMBER 3.0000000

WT FRACTION COMP 1 = 0.2854519E-02

WT FRACTION COMP 2 = 0.1525642

WT FRACTION COMP 3 = 0.7245359E-03

WT FRACTION COMP 4 = 0.1395911E-01

WT FRACTION COMP 5 = 0.2988590E-01

WT FRACTION COMP 6 = 0.0000000

WT FRACTION COMP 7 = 0.4596794E-03

WT FRACTION COMP 8 = 0.1499395

WT FRACTION COMP 9 = 0.2638180

WT FRACTION COMP 10 = 0.3857947

TOTAL STREAM WEIGHT = 1462.086

Figure 49 (contd.)

LEACH TANK

STREAM NUMBER 4.000000
 WT FRACTION COMP 1 = 0.2854518E-02
 WT FRACTION COMP 2 = 0.1287601
 WT FRACTION COMP 3 = 0.7245352E-03
 WT FRACTION COMP 4 = 0.1395910E-01
 WT FRACTION COMP 5 = 0.2988589E-01
 WT FRACTION COMP 6 = 0.2070048E-01
 WT FRACTION COMP 7 = 0.2728770E-03
 WT FRACTION COMP 8 = 0.1112167
 WT FRACTION COMP 9 = 0.3107656
 WT FRACTION COMP 10 = 0.3808512
 TOTAL STREAM WEIGHT = 1462.086

FRACTION OF ZNO LEACHED = 0.1559678

LEACH TANK

STREAM NUMBER 5.000000
 WT FRACTION COMP 1 = 0.2854518E-02
 WT FRACTION COMP 2 = 0.1098006
 WT FRACTION COMP 3 = 0.7245347E-03
 WT FRACTION COMP 4 = 0.1395909E-01
 WT FRACTION COMP 5 = 0.2988588E-01
 WT FRACTION COMP 6 = 0.3728958E-01
 WT FRACTION COMP 7 = 0.2446850E-03
 WT FRACTION COMP 8 = 0.8018482E-01
 WT FRACTION COMP 9 = 0.3481901
 WT FRACTION COMP 10 = 0.3768662
 TOTAL STREAM WEIGHT = 1462.086

FRACTION OF ZNO LEACHED = 0.1473058

Figure 49 (contd.)

LEACH TANK

STREAM NUMBER 6.000000
 WT FRACTION COMP 1 = 0.2854517E-02
 WT FRACTION COMP 2 = 0.9458804E-01
 WT FRACTION COMP 3 = 0.7245340E-03
 WT FRACTION COMP 4 = 0.1395908E-01
 WT FRACTION COMP 5 = 0.2988587E-01
 WT FRACTION COMP 6 = 0.5058973E-01
 WT FRACTION COMP 7 = 0.2161883E-03
 WT FRACTION COMP 8 = 0.5530532E-01
 WT FRACTION COMP 9 = 0.3782042
 WT FRACTION COMP 10 = 0.3736727
 TOTAL STREAM WEIGHT = 1462.086

FRACTION OF ZNO LEACHED = 0.1385473

LEACH TANK

STREAM NUMBER 7.000000
 WT FRACTION COMP 1 = 0.2854516E-02
 WT FRACTION COMP 2 = 0.8258682E-01
 WT FRACTION COMP 3 = 0.7245333E-03
 WT FRACTION COMP 4 = 0.1395907E-01
 WT FRACTION COMP 5 = 0.2988586E-01
 WT FRACTION COMP 6 = 0.6107715E-01
 WT FRACTION COMP 7 = 0.1868163E-03
 WT FRACTION COMP 8 = 0.3568734E-01
 WT FRACTION COMP 9 = 0.4018823
 WT FRACTION COMP 10 = 0.3711558
 TOTAL STREAM WEIGHT = 1462.086

FRACTION OF ZNO LEACHED = 0.1268785

Figure 49 (contd.).

LEACH TANK

STREAM NUMBER 8.000000
WT FRACTION COMP 1 = 0.2854516E-02
WT FRACTION COMP 2 = 0.7335812E-01
WT FRACTION COMP 3 = 0.7245326E-03
WT FRACTION COMP 4 = 0.1395906E-01
WT FRACTION COMP 5 = 0.2988585E-01
WT FRACTION COMP 6 = 0.6913549E-01
WT FRACTION COMP 7 = 0.1555817E-03
WT FRACTION COMP 8 = 0.2061323E-01
WT FRACTION COMP 9 = 0.4200903
WT FRACTION COMP 10 = 0.3692237
TOTAL STREAM WEIGHT = 1462.086

FRACTION OF ZNO LEACHED = 0.1117453

LEACH TANK

STREAM NUMBER 9.000000
WT FRACTION COMP 1 = 0.2854515E-02
WT FRACTION COMP 2 = 0.6660622E-01
WT FRACTION COMP 3 = 0.7245319E-03
WT FRACTION COMP 4 = 0.1395905E-01
WT FRACTION COMP 5 = 0.2988584E-01
WT FRACTION COMP 6 = 0.7502216E-01
WT FRACTION COMP 7 = 0.1206017E-03
WT FRACTION COMP 8 = 0.9601269E-02
WT FRACTION COMP 9 = 0.4334114
WT FRACTION COMP 10 = 0.3678145
TOTAL STREAM WEIGHT = 1462.086

FRACTION OF ZNO LEACHED = 0.9203970E-01

Figure 49 (contd.)

LEACH TANK

STREAM NUMBER 10.00000

WT FRACTION COMP 1 = 0.2854514E-02
WT FRACTION COMP 2 = 0.6237651E-01
WT FRACTION COMP 3 = 0.7245315E-03
WT FRACTION COMP 4 = 0.1395904E-01
WT FRACTION COMP 5 = 0.2988583E-01
WT FRACTION COMP 6 = 0.7869571E-01
WT FRACTION COMP 7 = 0.7929774E-04
WT FRACTION COMP 8 = 0.2729307E-02
WT FRACTION COMP 9 = 0.4417562
WT FRACTION COMP 10 = 0.3669392
TOTAL STREAM WEIGHT = 1462.086

FRACTION OF ZNO LEACHED = 0.6350321E-01

LEACH TANK

STREAM NUMBER 11.00000

WT FRACTION COMP 1 = 0.2854514E-02
WT FRACTION COMP 2 = 0.6076643E-01
WT FRACTION COMP 3 = 0.7245308E-03
WT FRACTION COMP 4 = 0.1395903E-01
WT FRACTION COMP 5 = 0.2988582E-01
WT FRACTION COMP 6 = 0.8007014E-01
WT FRACTION COMP 7 = 0.3068696E-04
WT FRACTION COMP 8 = 0.1581728E-03
WT FRACTION COMP 9 = 0.4449325
WT FRACTION COMP 10 = 0.3666183
TOTAL STREAM WEIGHT = 1462.086

FRACTION OF ZNO LEACHED = 0.2581191E-01

Figure 49 (contd.)

CLASSIFIER

STREAM NUMBER 12.00000

WT FRACTION COMP 1 = 0.2763275E-02
WT FRACTION COMP 2 = 0.5882416E-01
WT FRACTION COMP 3 = 0.7013720E-03
WT FRACTION COMP 4 = 0.1351285E-01
WT FRACTION COMP 5 = 0.2893057E-01
WT FRACTION COMP 6 = 0.7751083E-01
WT FRACTION COMP 7 = 0.3091441E-04
WT FRACTION COMP 8 = 0.1593452E-03
WT FRACTION COMP 9 = 0.4482303
WT FRACTION COMP 10 = 0.3693355
TOTAL STREAM WEIGHT = 966.6306

STREAM NUMBER 13.00000

WT FRACTION COMP 1 = 0.3032516E-02
WT FRACTION COMP 2 = 0.6455570E-01
WT FRACTION COMP 3 = 0.7697111E-03
WT FRACTION COMP 4 = 0.1482949E-01
WT FRACTION COMP 5 = 0.3174946E-01
WT FRACTION COMP 6 = 0.8506322E-01
WT FRACTION COMP 7 = 0.3024313E-04
WT FRACTION COMP 8 = 0.1558850E-03
WT FRACTION COMP 9 = 0.4384975
WT FRACTION COMP 10 = 0.3613158
TOTAL STREAM WEIGHT = 495.4551

Figure 49 (contd.)

CLASSIFIER

STREAM NUMBER 14.00000

WT FRACTION COMP 1 = 0.7026556E-03
 WT FRACTION COMP 2 = 0.1495802E-01
 WT FRACTION COMP 3 = 0.1783476E-03
 WT FRACTION COMP 4 = 0.3436099E-02
 WT FRACTION COMP 5 = 0.7356580E-02
 WT FRACTION COMP 6 = 0.1970976E-01
 WT FRACTION COMP 7 = 0.3605199E-04
 WT FRACTION COMP 8 = 0.1858260E-03
 WT FRACTION COMP 9 = 0.5227207
 WT FRACTION COMP 10 = 0.4307145
 TOTAL STREAM WEIGHT = 406.2729

STREAM NUMBER 15.00000

WT FRACTION COMP 1 = 0.1364630E-01
 WT FRACTION COMP 2 = 0.2905005
 WT FRACTION COMP 3 = 0.3463700E-02
 WT FRACTION COMP 4 = 0.6673270E-01
 WT FRACTION COMP 5 = 0.1428726
 WT FRACTION COMP 6 = 0.3827842
 WT FRACTION COMP 7 = 0.3780385E-05
 WT FRACTION COMP 8 = 0.1948558E-04
 WT FRACTION COMP 9 = 0.5481212E-01
 WT FRACTION COMP 10 = 0.4516442E-01
 TOTAL STREAM WEIGHT = 89.18192

MIXER

STREAM NUMBER 16.00000

WT FRACTION COMP 1 = 0.2153493E-02
 WT FRACTION COMP 2 = 0.4584324E-01
 WT FRACTION COMP 3 = 0.5465981E-03
 WT FRACTION COMP 4 = 0.1053093E-01
 WT FRACTION COMP 5 = 0.2254636E-01
 WT FRACTION COMP 6 = 0.6040628E-01
 WT FRACTION COMP 7 = 0.3243478E-04
 WT FRACTION COMP 8 = 0.1671816E-03
 WT FRACTION COMP 9 = 0.4702742
 WT FRACTION COMP 10 = 0.3874993
 TOTAL STREAM WEIGHT = 1372.901
 AVERAGE CONVERSION = 0.0000000

Figure 49 (contd.)

THICKENER

STREAM NUMBER 17.00000

WT FRACTION COMP 1 = 0.0000000
WT FRACTION COMP 2 = 0.0000000
WT FRACTION COMP 3 = 0.0000000
WT FRACTION COMP 4 = 0.0000000
WT FRACTION COMP 5 = 0.0000000
WT FRACTION COMP 6 = 0.0000000
WT FRACTION COMP 7 = 0.3780395E-04
WT FRACTION COMP 8 = 0.1948564E-03
WT FRACTION COMP 9 = 0.5481220
WT FRACTION COMP 10 = 0.4516448
TOTAL WEIGHT STREAM = 1047.919

STREAM NUMBER 18.00000

WT FRACTION COMP 1 = 0.9097539E-02
WT FRACTION COMP 2 = 0.1936671
WT FRACTION COMP 3 = 0.2309132E-02
WT FRACTION COMP 4 = 0.4448845E-01
WT FRACTION COMP 5 = 0.9524816E-01
WT FRACTION COMP 6 = 0.2551894
WT FRACTION COMP 7 = 0.1512158E-04
WT FRACTION COMP 8 = 0.7794256E-04
WT FRACTION COMP 9 = 0.2192488
WT FRACTION COMP 10 = 0.1806579
TOTAL WEIGHT STREAM = 324.9819

LEACH SLURRY

STREAM NUMBER 19.00000

WT FRACTION COMP 1 = 0.1364631E-01
WT FRACTION COMP 2 = 0.2905006
WT FRACTION COMP 3 = 0.3463703E-02
WT FRACTION COMP 4 = 0.6673276E-01
WT FRACTION COMP 5 = 0.1428727
WT FRACTION COMP 6 = 0.3827845
WT FRACTION COMP 7 = 0.3780387E-05
WT FRACTION COMP 8 = 0.1948560E-04
WT FRACTION COMP 9 = 0.5481217E-01
WT FRACTION COMP 10 = 0.4516446E-01
TOTAL STREAM WEIGHT = 89.18182

Figure 49 (contd.)

BALLMILL

STREAM NUMBER 20.00000

WT FRACTION COMP 1 = 0.1364631E-01
 WT FRACTION COMP 2 = 0.2905006
 WT FRACTION COMP 3 = 0.3463703E-02
 WT FRACTION COMP 4 = 0.6673276E-01
 WT FRACTION COMP 5 = 0.1428727
 WT FRACTION COMP 6 = 0.3827845
 WT FRACTION COMP 7 = 0.3780387E-05
 WT FRACTION COMP 8 = 0.1948560E-04
 WT FRACTION COMP 9 = 0.5481217E-01
 WT FRACTION COMP 10 = 0.4516446E-01
 TOTAL STREAM WEIGHT = 89.18182

MIXER

STREAM NUMBER 21.00000

WT FRACTION COMP 1 = 0.1007704E-01
 WT FRACTION COMP 2 = 0.2145186
 WT FRACTION COMP 3 = 0.2557750E-02
 WT FRACTION COMP 4 = 0.4927841E-01
 WT FRACTION COMP 5 = 0.1055033
 WT FRACTION COMP 6 = 0.2826650
 WT FRACTION COMP 7 = 0.1267949E-04
 WT FRACTION COMP 8 = 0.6535515E-04
 WT FRACTION COMP 9 = 0.1838410
 WT FRACTION COMP 10 = 0.1514824
 TOTAL STREAM WEIGHT = 414.1628
 AVERAGE CONVERSION = 0.0000000

LEACH SLURRY TANK

STREAM NUMBER 22.00000

WT FRACTION COMP 1 = 0.3174416E-02
 WT FRACTION COMP 2 = 0.6757647E-01
 WT FRACTION COMP 3 = 0.8057281E-03
 WT FRACTION COMP 4 = 0.1552341E-01
 WT FRACTION COMP 5 = 0.3323509E-01
 WT FRACTION COMP 6 = 0.8904362E-01
 WT FRACTION COMP 7 = 0.1347375
 WT FRACTION COMP 8 = 0.1759013E-03
 WT FRACTION COMP 9 = 0.2798288
 WT FRACTION COMP 10 = 0.3758992
 TOTAL STREAM WEIGHT = 1314.741

Figure 49 (contd.)

LEACH TANK

STREAM NUMBER 23.00000

WT FRACTION COMP 1 = 0.3174415E-02

WT FRACTION COMP 2 = 0.6520712E-01

WT FRACTION COMP 3 = 0.8057274E-03

WT FRACTION COMP 4 = 0.1552340E-01

WT FRACTION COMP 5 = 0.3323508E-01

WT FRACTION COMP 6 = 0.0000000

WT FRACTION COMP 7 = 0.9325441E-02

WT FRACTION COMP 8 = 0.1667417

WT FRACTION COMP 9 = 0.2845032

WT FRACTION COMP 10 = 0.4214839

TOTAL STREAM WEIGHT = 1314.741

FRACTION OF ZNO LEACHED = 0.3506088E-01

LEACH TANK

STREAM NUMBER 24.00000

WT FRACTION COMP 1 = 0.3174414E-02

WT FRACTION COMP 2 = 0.6342798E-01

WT FRACTION COMP 3 = 0.8057267E-03

WT FRACTION COMP 4 = 0.1552339E-01

WT FRACTION COMP 5 = 0.3323507E-01

WT FRACTION COMP 6 = 0.0000000

WT FRACTION COMP 7 = 0.7181235E-02

WT FRACTION COMP 8 = 0.1667415

WT FRACTION COMP 9 = 0.2880132

WT FRACTION COMP 10 = 0.4218975

TOTAL STREAM WEIGHT = 1314.741

FRACTION OF ZNO LEACHED = 0.2728420E-01

Figure 49 (contd.)

LEACH TANK

STREAM NUMBER 25.00000
WT FRACTION COMP 1 = 0.3174413E-02
WT FRACTION COMP 2 = 0.6206777E-01
WT FRACTION COMP 3 = 0.8057260E-03
WT FRACTION COMP 4 = 0.1552338E-01
WT FRACTION COMP 5 = 0.3323505E-01
WT FRACTION COMP 6 = 0.0000000
WT FRACTION COMP 7 = 0.5541917E-02
WT FRACTION COMP 8 = 0.1667411
WT FRACTION COMP 9 = 0.2906967
WT FRACTION COMP 10 = 0.4222141
TOTAL STREAM WEIGHT = 1314.741

FRACTION OF ZNO LEACHED = 0.2144474E-01

LEACH TANK

STREAM NUMBER 26.00000
WT FRACTION COMP 1 = 0.3174413E-02
WT FRACTION COMP 2 = 0.6102891E-01
WT FRACTION COMP 3 = 0.8057253E-03
WT FRACTION COMP 4 = 0.1552337E-01
WT FRACTION COMP 5 = 0.3323504E-01
WT FRACTION COMP 6 = 0.0000000
WT FRACTION COMP 7 = 0.4289903E-02
WT FRACTION COMP 8 = 0.1667407
WT FRACTION COMP 9 = 0.2927460
WT FRACTION COMP 10 = 0.4224558
TOTAL STREAM WEIGHT = 1314.741

FRACTION OF ZNO LEACHED = 0.1673728E-01

Figure 49 (contd.)

LEACH TANK

STREAM NUMBER 27.00000

WT FRACTION COMP 1 = 0.3174412E-02
WT FRACTION COMP 2 = 0.6023568E-01
WT FRACTION COMP 3 = 0.8057246E-03
WT FRACTION COMP 4 = 0.1552335E-01
WT FRACTION COMP 5 = 0.3323503E-01
WT FRACTION COMP 6 = 0.0000000
WT FRACTION COMP 7 = 0.3333907E-02
WT FRACTION COMP 8 = 0.1667405
WT FRACTION COMP 9 = 0.2943106
WT FRACTION COMP 10 = 0.4226408
TOTAL STREAM WEIGHT = 1314.741

FRACTION OF ZNO LEACHED = 0.1299733E-01

LEACH TANK

STREAM NUMBER 28.00000

WT FRACTION COMP 1 = 0.3174411E-02
WT FRACTION COMP 2 = 0.5962082E-01
WT FRACTION COMP 3 = 0.8057239E-03
WT FRACTION COMP 4 = 0.1552334E-01
WT FRACTION COMP 5 = 0.3323502E-01
WT FRACTION COMP 6 = 0.0000000
WT FRACTION COMP 7 = 0.2592938E-02
WT FRACTION COMP 8 = 0.1667402
WT FRACTION COMP 9 = 0.2955234
WT FRACTION COMP 10 = 0.4227841
TOTAL STREAM WEIGHT = 1314.741

FRACTION OF ZNO LEACHED = 0.1020736E-01

Figure 49 (contd.)

LEACH TANK

STREAM NUMBER 29.00000

WT FRACTION COMP 1 = 0.3174411E-02
WT FRACTION COMP 2 = 0.5914369E-01
WT FRACTION COMP 3 = 0.8057232E-03
WT FRACTION COMP 4 = 0.1552333E-01
WT FRACTION COMP 5 = 0.3323501E-01
WT FRACTION COMP 6 = 0.0000000
WT FRACTION COMP 7 = 0.2017967E-02
WT FRACTION COMP 8 = 0.1667398
WT FRACTION COMP 9 = 0.2964645
WT FRACTION COMP 10 = 0.4228956
TOTAL STREAM WEIGHT = 1314.741

FRACTION OF ZNO LEACHED = 0.8002460E-02

LEACH TANK

STREAM NUMBER 30.00000

WT FRACTION COMP 1 = 0.3174410E-02
WT FRACTION COMP 2 = 0.5877560E-01
WT FRACTION COMP 3 = 0.8057223E-03
WT FRACTION COMP 4 = 0.1552332E-01
WT FRACTION COMP 5 = 0.3323499E-01
WT FRACTION COMP 6 = 0.0000000
WT FRACTION COMP 7 = 0.1574388E-02
WT FRACTION COMP 8 = 0.1667395
WT FRACTION COMP 9 = 0.2971904
WT FRACTION COMP 10 = 0.4229816
TOTAL STREAM WEIGHT = 1314.741

FRACTION OF ZNO LEACHED = 0.6223381E-02

Figure 49 (contd.)

LEACH TANK

STREAM NUMBER 31.00000
WT FRACTION COMP 1 = 0.3174409E-02
WT FRACTION COMP 2 = 0.5848790E-01
WT FRACTION COMP 3 = 0.8057216E-03
WT FRACTION COMP 4 = 0.1552331E-01
WT FRACTION COMP 5 = 0.3323498E-01
WT FRACTION COMP 6 = 0.0000000
WT FRACTION COMP 7 = 0.1227715E-02
WT FRACTION COMP 8 = 0.1667392
WT FRACTION COMP 9 = 0.2977577
WT FRACTION COMP 10 = 0.4230490
TOTAL STREAM WEIGHT = 1314.741

FRACTION OF ZNO LEACHED = 0.4894674E-02

LEACH TANK

STREAM NUMBER 32.00000
WT FRACTION COMP 1 = 0.3174409E-02
WT FRACTION COMP 2 = 0.5826546E-01
WT FRACTION COMP 3 = 0.8057209E-03
WT FRACTION COMP 4 = 0.1552330E-01
WT FRACTION COMP 5 = 0.3323497E-01
WT FRACTION COMP 6 = 0.0000000
WT FRACTION COMP 7 = 0.9597018E-03
WT FRACTION COMP 8 = 0.1667390
WT FRACTION COMP 9 = 0.2981963
WT FRACTION COMP 10 = 0.4231011
TOTAL STREAM WEIGHT = 1314.741

FRACTION OF ZNO LEACHED = 0.3802836E-02

Figure 49 (contd.)

LEACH TANK

STREAM NUMBER 32.00000

WT FRACTION COMP 1 = 0.3174408E-02

WT FRACTION COMP 2 = 0.5809046E-01

WT FRACTION COMP 3 = 0.8057202E-03

WT FRACTION COMP 4 = 0.1552328E-01

WT FRACTION COMP 5 = 0.3323496E-01

WT FRACTION COMP 6 = 0.0000000

WT FRACTION COMP 7 = 0.7488327E-03

WT FRACTION COMP 8 = 0.1667387

WT FRACTION COMP 9 = 0.2985414

WT FRACTION COMP 10 = 0.4231424

TOTAL STREAM WEIGHT = 1314.741

FRACTION OF ZNO LEACHED = 0.3003240E-02

LEACH TANK

STREAM NUMBER 34.00000

WT FRACTION COMP 1 = 0.3174407E-02

WT FRACTION COMP 2 = 0.5795519E-01

WT FRACTION COMP 3 = 0.8057195E-03

WT FRACTION COMP 4 = 0.1552327E-01

WT FRACTION COMP 5 = 0.3323495E-01

WT FRACTION COMP 6 = 0.0000000

WT FRACTION COMP 7 = 0.5859009E-03

WT FRACTION COMP 8 = 0.1667384

WT FRACTION COMP 9 = 0.2988078

WT FRACTION COMP 10 = 0.4231743

TOTAL STREAM WEIGHT = 1314.741

FRACTION OF ZNO LEACHED = 0.2328217E-02

Figure 49 (contd.)

LEACH TANK

STREAM NUMBER 35.00000

WT FRACTION COMP	1 =	0.3174406E-02
WT FRACTION COMP	2 =	0.5784915E-01
WT FRACTION COMP	3 =	0.8057188E-03
WT FRACTION COMP	4 =	0.1552326E-01
WT FRACTION COMP	5 =	0.3323494E-01
WT FRACTION COMP	6 =	0.0000000
WT FRACTION COMP	7 =	0.4582019E-03
WT FRACTION COMP	8 =	0.1667382
WT FRACTION COMP	9 =	0.2990167
WT FRACTION COMP	10 =	0.4231994
TOTAL STREAM WEIGHT	=	1314.741

FRACTION OF ZNO LEACHED = 0.1829207E-02

Figure 49 (contd.)

THICKENER

STREAM NUMBER 36.00000

WT FRACTION COMP	1 =	0.0000000
WT FRACTION COMP	2 =	0.0000000
WT FRACTION COMP	3 =	0.0000000
WT FRACTION COMP	4 =	0.0000000
WT FRACTION COMP	5 =	0.0000000
WT FRACTION COMP	6 =	0.0000000
WT FRACTION COMP	7 =	0.5151734E-03
WT FRACTION COMP	8 =	0.1874700
WT FRACTION COMP	9 =	0.3361956
WT FRACTION COMP	10 =	0.4758187
TOTAL STREAM WEIGHT	=	1023.953

STREAM NUMBER 37.00000

WT FRACTION COMP	1 =	0.1435247E-01
WT FRACTION COMP	2 =	0.2615538
WT FRACTION COMP	3 =	0.3642900E-02
WT FRACTION COMP	4 =	0.7018536E-01
WT FRACTION COMP	5 =	0.1502653
WT FRACTION COMP	6 =	0.0000000
WT FRACTION COMP	7 =	0.2575866E-03
WT FRACTION COMP	8 =	0.9373492E-01
WT FRACTION COMP	9 =	0.1680977
WT FRACTION COMP	10 =	0.2379093
TOTAL STREAM WEIGHT	=	290.7881

Figure 49 (contd.)

FILTER

STREAM NUMBER 38.00000

WT FRACTION COMP 1 = 0.0000000
WT FRACTION COMP 2 = 0.0000000
WT FRACTION COMP 3 = 0.0000000
WT FRACTION COMP 4 = 0.0000000
WT FRACTION COMP 5 = 0.0000000
WT FRACTION COMP 6 = 0.0000000
WT FRACTION COMP 7 = 0.5151725E-03
WT FRACTION COMP 8 = 0.1874697
WT FRACTION COMP 9 = 0.3361952
WT FRACTION COMP 10 = 0.4758183
WT FRACTION COMP 11 = 0.0000000
TOTAL WEIGHT STREAM = 145.3941

STREAM NUMBER 39.00000

WT FRACTION COMP 1 = 0.2870493E-01
WT FRACTION COMP 2 = 0.5231076
WT FRACTION COMP 3 = 0.7285796E-02
WT FRACTION COMP 4 = 0.1403707
WT FRACTION COMP 5 = 0.3005305
WT FRACTION COMP 6 = 0.0000000
WT FRACTION COMP 7 = 0.0000000
WT FRACTION COMP 8 = 0.0000000
WT FRACTION COMP 9 = 0.0000000
WT FRACTION COMP 10 = 0.0000000
WT FRACTION COMP 11 = 0.0000000
TOTAL WEIGHT STREAM = 145.3940

Figure 49 (contd.)

MIXER

STREAM NUMBER 2.000000
WT FRACTION COMP 1 = 0.0000000
WT FRACTION COMP 2 = 0.0000000
WT FRACTION COMP 3 = 0.0000000
WT FRACTION COMP 4 = 0.0000000
WT FRACTION COMP 5 = 0.0000000
WT FRACTION COMP 6 = 0.0000000
WT FRACTION COMP 7 = 0.5151739E-03
WT FRACTION COMP 8 = 0.1874701
WT FRACTION COMP 9 = 0.3361956
WT FRACTION COMP 10 = 0.4758191
TOTAL STREAM WEIGHT = 1169.346
AVERAGE CONVERSION = 0.0000000

ELECTROLYTIC PLANT

TONS/DAY ZINC PRODUCED = 116.3077

CELL CURRENT (AMPS) = 8826.457

STREAM NUMBER 40.00000
WT FRACTION COMP 1 = 0.0000000
WT FRACTION COMP 2 = 0.0000000
WT FRACTION COMP 3 = 0.0000000
WT FRACTION COMP 4 = 0.0000000
WT FRACTION COMP 5 = 0.0000000
WT FRACTION COMP 6 = 0.0000000
WT FRACTION COMP 7 = 0.1931026
WT FRACTION COMP 8 = 0.2261354E-03
WT FRACTION COMP 9 = 0.3180539
WT FRACTION COMP 10 = 0.4886174
TOTAL WEIGHT STREAM = 902.9714

Bibliography

Books

1. Bird, R. Bryon; Stewart, Warren E.; Lightfoot, Edwin N.; Transport Phenomena; John Wiley & Sons, Inc.; 1960; New York
2. Crowe, C.M.; Hamielec, A.E.; Hoffman, T.W.; Johnson, A.I.; Shannon, P.T.; Woods, D.R.; Chemical Plant Simulation An Introduction to Computer - Aided Steady-State Process Analysis; Prentice - Hall, Inc.; 1971; Englewood Cliffs, N.J.
3. Evans, George W.; Wallace, Graham F.; Sutherland, Georgia L.; Simulation Using Digital Computers; Prentice - Hall, Inc.; 1967; Englewood Cliffs, N.J.
4. Franks, Roger G.E.; Mathematical Modeling in Chemical Engineering; John Wiley & Sons, Inc.; 1967; New York
5. Foust, Alan S.; Wenzel, Leonard A.; Clump, Curtis W.; Mans, Louis; Andersen L. Bryce; Principles of Unit Operations; John Wiley & Sons, Inc.; 1962; New York
6. Hayward, Carle R.; An Outline of Metallurgical Practice; Third Edition Second Printing; D. Van Norstrand Company, Inc.; 1955; New York
7. Henley, Ernest F.; Rosen, Edward M.; Material and Energy Balance Computations; John Wiley & Sons, Inc.; 1969; New York
8. Lange, Norbert Adolph, Ph.D.; Handbook of Chemistry, Tenth Edition; McGraw - Hill; 1961; New York
9. Levenspiel, Octave; Chemical Reaction Engineering; John Wiley & Sons, Inc.; 1962; New York
10. Lingane, James J.; Electroanalytical Chemistry, Second Edition; Interscience Publishers, Inc.; 1966; New York
11. Martin, Francis F. Computer Modeling and Simulation; John Wiley & Sons, Inc.; 1968; New York
12. Motard, R.L.; Lee, H.M.; Barkley, R.W.; Ingels, D.M.; CHESS -

Chemical Engineering Simulation System - System Guide; Tech Publishing Co.; 1968; Houston

13. Moyle, Morton P.; Introduction to Computers for Engineers; John Wiley & Sons, Inc.; 1967; New York
14. Pritsker, A. Alan B.; Kiviat, Philip J.; Simulation With GASP II A Fortran Based Simulation Language; Prentice-Hall, Inc.; 1969; Englewood Cliffs, N.J.
15. Wilde, Douglas J.; Beightler, Charles S.; Foundations of Optimization; Prentice-Hall, Inc.; 1967; Englewood Cliffs, N.J.

Periodicals

16. Beveridge, G.S.G.; "The Prediction of Reaction Zone Propagation Rates in Beds of Reacting Solids - The Effect of the Moisture Content in The Solids"; Symposium on Chem. Engineering in The Metallurgical Industries (1963; Instn Chem. Engrs); pp. 87 - 104
17. Eyre, E.; "Suspension Roasting of Zinc Concentrate at Trail, British Columbia"; The Canadian Mining and Metallurgical Bulletin, August, 1961; pp 589 - 594
18. Fitch, Bryant, "Current Theory and Thickener Design"; Industrial and Engineering Chemistry; Vol 58, No. 10; October 1966; pp 18 - 28
19. Penny, G.W.; "Some Problems in Application of the Deutsch Equation to Industrial Electrostatic Precipitation"; Journal of the Air Pollution Control Association; Vol 19, No. 8; August, 1969; pp 596 - 600
20. Robinson, Myron; "The Early History of the Electrodeposition and Separation of Particles"; Electrochemical Society News & Reviews; May, 1968; pp 131C - 137C
21. Schneider, Ben; "Multiple Hearth Roasting of Sulphides in the Zinc Roasting Plant of the Hudson Bay Mining and Smelting Co., Limited Fin Flon, Manitoba"; The Canadian Mining and Metallurgical Bulletin; August, 1961; pp 585 - 588
22. Scott, Keith J.; "Experimental Study of Continuous Thickening of a Flocculated Silica Slurry"; I & EC Fundamentals; Vol. 7, No. 4;

November, 1968; pp 582 - 595

23. Shannon, Paul T.; Shoupe, Elwood; Troy, Elmer M.; "Batch and Continuous Thickening"; I & EC Fundamentals; Vol. 2, No. 3; August 1963; pp 203 - 211
24. Treilhard, D.G.; "Flash Roasting of Zinc Concentrates"; The Canadian Mining and Metallurgical Bulletin; November, 1961; pp 847 - 853
25. Wegstein, J.H.; "Accelerated Convergence of Iterative Processes" Comm. ACM; Vol. 1, No. 6, 9; 1958

Handbooks & Reference Manuals

26. Fortran IV - H Reference Manual for SDS Sigma 5/7 Computers; Scientific Data Systems; El Segundo, California; 1968
27. Handbook of Chemistry and Physics; 44th Edition; Chemical Rubber Publishing Co.; 1962 - 1963
28. Perry's Chemical Engineers Handbook; Fourth Edition; Perry, Robert H.; Chilton, Cecil H.; Kirkpatrick, Sidney D.; McGraw-Hill; New York

MONTANA STATE UNIVERSITY LIBRARIES



3 1762 10005658 7

D378
J236 James, Rodney A
con.2 Mathematical models
for inorganic processes
found in a zinc plant

NAME AND ADDRESS

~~MAY 25 1995~~

~~RECEIVED OCT 1 1995~~

~~MSU~~
D378
J236
con.2