220
19/70
AEC RESEARCH AND DEVELOPMENT REPORT

1252
Y-1699

Mathematics and
Computers

MASTER

**HYPFIL: A FORTRAN COMPUTER PROGRAM FOR THE
INVESTIGATION OF PROCESS PARAMETERS IN
SINGLE-STAGE HYPERFILTRATION PROCESSES**

W. L. Griffith (Y-12)
R. M. Keller (Y-12)
R. A. Ebel (ORGDP)
R. S. Dillon (ORNL)

**UNION CARBIDE CORPORATION
NUCLEAR DIVISION
OAK RIDGE Y-12 PLANT**

operated for the **ATOMIC ENERGY COMMISSION** *under* **U. S. GOVERNMENT** **Contract W-7405 eng 26**

**UNION
CARBIDE**

OAK RIDGE Y-12 PLANT
P. O. Box Y
OAK RIDGE, TENNESSEE 37830

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED.

DISCLAIMER

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

DISCLAIMER

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

Printed in the United States of America. Available from Clearinghouse for Federal
Scientific and Technical Information, National Bureau of Standards,
U.S. Department of Commerce, Springfield, Virginia 22151
Price: Printed Copy \$3.00; Microfiche \$0.65

— LEGAL NOTICE —

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

- A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or
- B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

Date Issued: February 10, 1970

Document Y-1699

TID-4500

UNION CARBIDE CORPORATION
Nuclear Division

OAK RIDGE Y-12 PLANT

Contract W-7405-eng-26
With the US Atomic Energy Commission

HYPFIL: A FORTRAN COMPUTER PROGRAM FOR THE INVESTIGATION
OF PROCESS PARAMETERS IN SINGLE-STAGE
HYPERFILTRATION PROCESSES.

W. L. Griffith (Y-12)
R. M. Keller (Y-12)
R. A. Ebel (ORGDP)
R. S. Dillon (ORNL)

This Research was jointly sponsored by the US Atomic
Energy Commission and The Office of Saline Water,
US Department of the Interior, under contract with
the Union Carbide Corporation.

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission, makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe upon privately owned rights; or represents any liability with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above statement, the expression "employees of the Commission" includes any employee or contractor of the Commission, or any person acting on behalf of the Commission, who is employed or contracted by the Commission, or provides access to, any information, apparatus, method, or process disclosed in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe upon privately owned rights; or represents any liability with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

Oak Ridge, Tennessee

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISTRIBUTION

Aerojet-General Corporation-Environmental Systems Division

Longwell, P. A.

Atomic Energy Commission-ORO

Keller, C. A.

Roth, H. M.

Atomic Energy Commission-Washington

McDaniel, P. W. (3)

Battelle Memorial Institute

Research Library

Bonner and Moore Associates, Inc

Kephart, W. L.

Brookhaven National Laboratory

Uphoff, R. L.

Coca Cola Company-Atlanta

Copper, J. R.

Dow Chemical Company-Midland

Wheaton, R. M.

Eastman Chemical Products, Inc

Mahoney, J. G.

E I duPont de Nemours and Company-Wilmington

Rosenblatt, N. W.

Federal Water Pollution Control Administration-Cincinnati

Cohen, J. M.

Masse, A. N.

Middleton, F. M.

Schwartz, W. A.

Federal Water Pollution Control Administration-Washington

Shackelford, J. M.

Stephan, D. G.

Gulf General Atomic, Inc

Merten, U.

Hydronautics, Inc

Matz, R.

Kaiser Engineering

Harris, F. L.

Kansas State University-Department of Chemical Engineering

Fan, L. T.

Massachusetts Institute of Technology-Department of Chemical Engineering

Fleming, S. M.

Sherwood, T. K.

National Research Council-Division of Applied Chemistry

Sourirajan, S.

Oak Ridge Gaseous Diffusion Plant

Cross, K. E. (5)
Ebel, R. A. (5)
Hammerling, F. D.
Jordan, R. G.
Parsons, J. A.
Pasquier, L. P.
Wilcox, W. J., Jr.

Oak Ridge National Laboratory

Beall, S. E.
Boegly, W. J., Jr.
Bohlmann, E. G.
Central Research Library (2)
Culler, F. L.
Dahlheimer, J. A.
Dillon, R. S. (5)
Dresner, L.
Ferguson, D. E.
Griess, J. C.
Grimes, W. R.
Hammond, R. P. (2)
Hoffman, H. W.
Jacobs, D. G.
Johnson, J. S.
Johnsson, K. O.
Kraus, K. A. (50)
Laboratory Records Department (5)
Laboratory Records Department (ORNL RC)
Lietzke, M. H.
Liverman, J. L.
Lyon, R. N.
MacPherson, H. G.
Mayer, R. H. (MIT Practice School)
Meyer, R. E.
Perona, J. J.
Posey, F. A.
Raridon, R. J.
Reed, S. A.
Rupp, A. F.
Rush, R. M.
Shor, A. J.
Skinner, M. J.

Spiewak, I.
Taylor, E. H.
Thomas, D. G.
Weinberg, A. M.
Young, G.
Y-12 Technical Library (2)

Office of Saline Water

Channabasappa, K. C.
Coley, F. H.
Gillam, W. S.
Heintz, J. W.
Horowitz, R.
Johnson, S.
Kindley, L.
McCoy, W. H.
Mintz, M. S.
Podall, H.
Savage, W. F.
Sieder, E. N.
Sieveka, E. H.
Strobel, J. J.
Wong, C. M.

Office of Water Resources Research

Mangan, G. F.

Paducah Gaseous Diffusion Plant

Winkel, R. A.

Rensselaer Polytechnic Institute

Hardt, F. W.

The University of Tennessee-Department of Civil Engineering

Larson, F. C.

Union Carbide Corporation-New York

Swartout, J. A.

Union Carbide Research Institute

Litz, L. M.
Ullmann, J. W.

University of California-Department of Engineering

McCutchan, J. W.

University of California-Sea Water Conversion Laboratory

Howe, E. D.
Laird, A. D. K., Director
Spiegler, K. S.

University of Florida-College of Engineering

Keppel, R. A.

University of Texas

Hackerman, N., Chancellor

Vanderbilt University-Department of Sanitary Engineering

Parker, F.

Yale University-Sterling Chemistry Laboratory

Dodge, B. F.

Oak Ridge Y-12 Plant

Ballenger, H. F. (2)
Beeson, H. C.
Bernander, N. K.
Burkhart, L. E.
Evans, G. W.
Foulk, D. L.
Griffith, W. L. (50)
Haeusler, K. R.
Hemphill, L. F.
Hibbs, R. F.
Kahl, K. G.
Keith, Alvin

Keller, R. M. (5)
Kite, H. T.
Lang, D. M.
Marrow, G. B.
McLendon, J. D.
Mitchel, G. W. (2)
Phillips, L. R.
Smith, R. D.
Tench, F. M.
Trotter, T. C.
Warner, J. F.
Waters, J. L.
Wesley, R. L.
Yaggi, W. J.
Y-12 Central Files (15)
Y-12 Central Files (route)
Y-12 Central Files (Y-12RC)

In addition, this report is distributed in accordance with the category Mathematics and Computers, as given in the "USAEC Standard Distribution Lists for Unclassified Scientific and Technical Reports", TID-4500.

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

ABSTRACT

A FORTRAN IV computer program was developed for the IBM 360 computer to solve a system of equations to simulate the hydrodynamic flow conditions in tubular systems (rectangular or circular) and to calculate relative water costs. The mathematical model includes the effect of concentration polarization and the dependence of membrane salt rejection on flux and salt concentration. Provisions have been made to simulate the hydrodynamic effect of turbulence promoters and to include allowance for brine disposal costs in the economic model.

The inlet velocity and pressure, water recovery, and flow geometry can be optimized to calculate minimum water costs subject to inequality constraints on the independent variables. The optimized solution conforms to an upper limit placed on the product salt concentration and a lower limit on the Reynolds number.

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

CONTENTS

1. IDENTIFICATION	13
2. PURPOSE.	13
3. RESTRICTIONS	13
3.1 Hardware	13
3.2 System.	13
3.3 Inherent Limitations	14
4. METHOD.	15
4.1 Flow Arrangement	15
4.2 Mathematical Model.	16
4.2.1 Membrane Theory and Hydrodynamics	16
4.2.2 Economics	19
4.3 Description of Program	21
4.4 Numerical Procedure	23
4.4.1 Program Control and Optimization.	23
4.4.2 Material Balance Calculations	49
4.4.3 Calculation of Economic Results (Subroutine ECON)	69
4.5 Nomenclature	72
4.5.1 Variables Used in Equations in Sections 4.1 - 4.4	72
4.5.2 Variables Used in Subroutine TUBE in Subsection 4.4.2	75
5. USAGE	82
5.1 Input	83
5.2 Output	83
5.2.1 Initial Conditions for Namelist Input Variables	83
5.2.2 Physical Conditions and Economic Results at Optimum	84
5.2.3 Conditions Encountered During Optimization	86
5.2.4 Detailed Analysis of Optimization Procedure	87
5.3 Operating Instructions	87
5.3.1 Use of Namelist for Data Input	87
5.3.2 Optimization Variables and Scaling	88
5.3.3 Additional Information Concerning Namelist Input Variables	89
5.3.4 Errors Occurring During Program Execution	91
5.3.5 Validity of the Optimum	92
5.4 Example	92

6.	CODING INFORMATION	92
7.	REFERENCES	93
8.	ACKNOWLEDGEMENTS	94
9.	APPENDIX A	95
	9.1 Salt Concentration in a Turbulent Core	95
10.	APPENDIX B	97
	List B-1 HYPFIL Source Program.	98
	List B-2 Namelist Input Variables	124
	List B-3 Results at Optimum	125
	List B-4 Progress of Optimization (List-6 option).	126
	List B-5 Progress of Optimization (List-1 and List-6 options).	127

1. IDENTIFICATION

HYPFIL: Hyperfiltration Plant Design

Machine: IBM 360

Language: FORTRAN IV for the IBM 360 Operating System, March 20, 1969

Union Carbide Corporation
Oak Ridge, Tennessee

2. PURPOSE

HYPFIL is a computer program to solve a system of equations to simulate the hydrodynamic flow conditions in tubular hyperfiltration systems and to calculate the optimum relative water costs.

3. RESTRICTIONS

3.1 Hardware

The program is written entirely in FORTRAN IV language, is intended for batch processing, and uses neither programmed pauses nor switches. All input is by data cards using the "Namelist" feature. The output requires a 132-character printer. The program uses 130,000 bytes of storage, but this could be reduced to 50,000 bytes by reducing the size of some large arrays (see Division 6, Page 92).

3.2 System

Unless otherwise stated, use of the IBM 360/75 with the standard IBM FORTRAN IV Level H compiler at the Oak Ridge National Laboratory with their cataloged procedures, et cetera, is presumed. However, since the source language is FORTRAN IV, it would be easy to compile and execute this program on a wider class of machines. Changes in setup cards or syntax may be required at other installations.

A test case has been built into the program to facilitate trial testing.

3.3 Inherent Limitations

Several restrictions are inherent in the present version of the program which cannot be changed without additional programming effort, namely:

1. Flow in cylindrical tubes with the membrane at the inside wall or rectangular tubes with the membrane at the wide-wall boundary (two membranes on opposite sides of rectangular tubes), as shown in Figure 1, was considered. No provision for other tube geometries or flow outside tubes (with the product being collected on the inside) has been made.

Figure 1. FLOW ARRANGEMENT OF HYPERFILTRATION.

2. Only single-stage, straight-through flows are considered. No provision for re-circulation pumps has been made.
3. One to five variables: inlet pressure, inlet velocity, concentration ratio, plant taper, and tube diameter, may be optimized to calculate the minimum water. Some programming changes are required to include other variables.

4. A simple cost model based on three cost parameters (see Section 4.2, Page 16) is included; however, it would be easy to modify this to include more definitive engineering relationships when these can be determined.
5. Constant permeability of the membranes is presumed. No relationship to vary permeability with pressure, velocity, time, or temperature is currently included.
6. The maximum number of concentration increments that can be used in the numerical procedure is 500.
7. Physical property functions for aqueous sodium chloride solutions are provided. Program changes are required to include other solutes (or solvents).

4. METHOD

4.1 Flow Arrangement

A single-stage process with turbulent-flow inside cylindrical or rectangular tubes, with the membrane located at the inside wall boundary (on two opposite sides only for rectangular tubes), was considered. This arrangement is shown in Figure 1. Also considered was a "tapered" single-stage arrangement with the flows as shown in Figure 1 except that the number of parallel tubes at any point through the stage was varied (continuously) to increase velocity and decrease concentration polarization. Except for this taper, all flow systems considered in this report are straight-through systems, and no recirculation pumps are employed.

The variable called "taper" was defined as the fraction of the flow velocity lost by transfer of fluid through the membrane that is preserved by stepping the plant "width" in a continuous manner. For example, Taper = 1 is a constant velocity system, and Taper = 0 is a system in which the velocity decreases in a natural way through a system of parallel tubes (ie, no steps). Taper is closely approximated by:

$$\text{Taper} \approx \left(u_r - u_{ro} \right) / \left(u_f - u_{ro} \right), \quad (1)$$

where:

u_f represents the inlet velocity, and

u_r and u_{ro} are the terminal (reject stream) velocities in corresponding tapered and untapered systems, respectively.

As defined here, taper may be any positive or negative value.

4.2 Mathematical Model

The mathematical model is based largely on developments given by Johnson, Dresner, and Kraus.⁽¹⁾ The model includes the effect of concentration polarization and the dependence of membrane flux and salt concentration on rejection. The latter is of particular importance when considering ion-exchange membranes where rejection may be a rapidly varying function of concentration at the entrance interface of the membrane. Also included are parameters which permit consideration of a turbulence promoter as a means to reduce concentration polarization.

4.2.1 Membrane Theory and Hydrodynamics

A good approximation to the volume flux of water, $J_{1(v)}$, through the membrane is given by:

$$J_{1(v)} = P_{\text{eff}} \mathcal{P}, \quad (2)$$

where:

$\mathcal{P}$ represents the permeability of the membrane, and the effective pressure, P_{eff} , is defined by:

$$P_{\text{eff}} = \left(P_{\alpha} - P_{\omega} \right) - \left(\pi_{\alpha} - \pi_{\omega} \right). \quad (3)$$

Subscripts α and ω designate the entrance and exit interfaces of the membrane, and π the osmotic pressure of the solution.

The volume flux, J , is given by:

$$J = J_{1(v)} \rho_1 / \left[\rho_{\omega} (1 - x_{\omega}) \right], \quad (4)$$

where:

ρ_1 represents the density of water,

ρ_{ω} the density of the product, and

x_{ω} the weight fraction of solute in the product.

Assuming the effective pressure to be the same at any point on the circumference at any given distance, y , along a tube with tube cross-sectional area, A_t , and membrane perimeter, P_m (ft), the change in average axial velocity, u (ft/sec), of the bulk stream through a tube is given by:

$$\frac{du}{dy} = -1.55 \times 10^{-6} \int P_m / A_t. \quad (5)$$

Similarly, the change in the bulk flowrate, B (lbs/day), in a tube may be expressed as:

$$\frac{dB}{dy} = -P_m \int \rho_\omega, \quad (6)$$

when P_m is in feet, ρ_ω is lbs/gal, and $\int$ is in gpd ft⁻². The pressure drop in a tube is related to friction losses and momentum changes by the relation:

$$\frac{dP}{dy} = -\frac{\rho_t}{283} \left(\frac{2u^2 f}{g_c d} + \frac{u du}{g_c dy} \right), \quad (7)$$

when P is in atm, ρ is in lbs/gal, u is in ft/sec, d is the "equivalent tube diameter" in ft, and g_c is in lb_m ft lb⁻¹ f sec⁻².

The steady-state salt rejection of the membrane is expressed in terms of solution concentrations by:

$$R = (m_\alpha - m_\omega) / m_\alpha = 1 - (m_\omega / m_\alpha) \approx 1 - (x_\omega / x_\alpha). \quad (8)$$

Consideration of salt and mass balances gives:

$$\frac{d\bar{x}}{dy} = P_m \int \rho_\omega (\bar{x} - x_\omega) / B. \quad (9)$$

Here, $\bar{x}$ is the average "cup mixing" weight fraction of salt at point y .

For turbulent flow conditions, concentration polarization for perfectly rejecting membranes ($R = 1$) as expressed by Sherwood, et al(2) is:

$$\psi_t [R = 1] = \exp \left[2v/(uf) (N_{Sc})^{2/3} \right] - 1, \text{ or} \quad (10a)$$

$$\psi_t [R = 1] = \exp \left[25 (v/u) (N_{Re})^{1/4} (N_{Sc})^{2/3} \right] - 1, \quad (10b)$$

if the Blasius equation is used to calculate the Fanning friction factor, f . Equation 10b has been extended⁽¹⁾ to the case $R < 1$, where concentration polarization is given by:

$$\psi_t [R < 1] = \frac{R \psi_t [R = 1]}{1 + \psi_t [R = 1] (1-R)} \quad (11)$$

The validity of this relation for turbulent systems has been experimentally verified by Shor, et al⁽³⁾ for tubular, dynamically formed, high-flux, hydrous zirconium oxide membranes. By substituting concentrations for R , Equation 3, the relationship between the concentration at the entrance membrane surface and the turbulent core may be conveniently expressed as a "mixing efficiency", E_M , by:

$$E_M \equiv \frac{m_t - m_\omega}{m_\alpha - m_\omega} = \exp - \left[25 (v/u) \left(N_{Re} \right)^{1/4} \left(N_{Sc} \right)^{2/3} \right] \quad (12)$$

The salt concentration in the turbulent core, x_t , has been assumed equal to the average "cup mixing" concentration at Point y . The difference is small, as shown in Appendix A. Also, the mixing efficiency is calculated using weight fractions instead of molalities, which gives a very close approximation for sodium chloride in the concentration range of interest.

Turbulence promoters may be included in the model by separately modifying the friction factors predicted by the Blasius equation which appear in the friction loss and concentration polarization calculations. The exact form of the factors used is presented in detail in Subsection 4.4.2 (Page 49), and usage is explained in Section 5.3 (Page 87).

The dependence of salt rejection on salt concentration and flux may be approximated by:⁽¹⁾

$$\left(\frac{1}{R} - 1 \right) = \left(\frac{e^{\beta\sigma}}{e^{\beta\sigma} - 1} \right) \left(\frac{\beta D^*}{1 - \beta D^*} \right); \beta \neq 0, \text{ and} \quad (13)$$

$$\left(\frac{1}{R} - 1 \right) = D^*/\sigma; \beta = 0,$$

where:

β represents a coupling coefficient for salt and water fluxes, and

$$\beta\sigma = \beta J_1 z / D_2^* \approx K J_{1(v)} \quad (13a)$$

Here:

D^* represents the distribution coefficient of the salt,

D_2^* a special diffusion coefficient, and

z the thickness of the membrane.

At high flux ($\beta\sigma \gg 1$), a limiting value of rejection, R_∞ , is reached and Equation 13 reduces to:

$$1 - R_\infty = \beta D^* . \quad (14)$$

For further details see Johnson, et al. (1)

Although the distribution coefficient, D^* , could be expressed as a function of salt concentration in any way desired in the computer program, two methods are provided in this program: (1) a concentration-dependent polynomial fit (including a constant distribution coefficient), and (2) a standard Donnan equilibrium concentration dependence with a membrane capacity, C^* moles/kg H_2O . In the latter case, solutes are provided with the formula MX_b : (4)

$$D^* = \left(\frac{Q}{1 + QD^*} \right)^b \left(\frac{1}{\Gamma^*} \right)^{b+1} , \quad (15)$$

where:

$$Q = bm / \alpha C^* . \quad (15a)$$

Unit activity coefficient ratios, Γ^* , have been assumed in the computer program.

4.2.2 Economics

Since the ultimate factor in determining the process parameters in any separations process is usually product cost subject to certain technological restraints, water cost was chosen as the variable to be minimized in determining the optimum flow conditions. Although quite detailed and sophisticated cost relationships are used to estimate costs in the more highly developed desalination processes, such as distillation, a simplified cost equation was used here because much of the equipment for hyperfiltration is in the early stage of development. Only three cost parameters: (1) a daily charge, C_A , per square foot of membrane area, A ; (2) a unit cost for pumping power, C_E ; and (3) a unit cost for brine disposal, C_D , are used. Expressed in this form, the unit cost, C_P , of product water is:

$$C_P = (C_A^A + C_E^P + C_D^W R) / W_P, \quad (16)$$

where:

W_P represents the product volume (kgals/day),

W_R the volume of waste brine (kgals/day), and

P_E the energy (kwh) consumed per day.

Equation 16 can be rewritten as:

$$C_P = C_E (R_{AE}^A + P_E + R_{DE}^W R) / W_P, \quad (17)$$

if R_{AE} is defined as:

$$C_A / C_E, \text{ and}$$

$$R_{DE} \text{ as } C_D / C_E. \quad (18)$$

Thus, for the case of zero brine disposal cost, C_D , product water costs, C_P , may be expressed in terms of the energy costs, C_E , and the ratio: C_A / C_E (R_{AE}). Since in this form, C_E is a proportionality factor, the minimum values of C_P occur at the same optimum flow conditions for R_{AE} equal to a constant. Costs at a fixed value of R_{AE} , but different values of C_E , are proportional to C_E .

For brine disposal cost that is not zero, C_E is likewise a proportionality factor where the minimum values of C_P occur at the same optimal flow conditions for fixed values of both cost ratios R_{AE} and R_{DE} . Here costs for fixed values of R_{AE} and R_{DE} but different values of C_E are proportional to C_E .

A simplified equation of the type of Equation 16 is most meaningful if the cost parameters are interpreted as incremental costs. The area cost, C_A , would then be the additional capital charge and operating expense (expressed as $\phi/\text{ft}^2 \text{ day}$) associated with increasing the size of the plant. This includes, for example, capital charges arising from additional membranes, backing materials, collection systems, and space requirements, as well as additional operating expense to maintain the increased area. The capital charges arising from the cost of pumps might be included in C_A , though it seems more appropriate to include pump costs in C_E , the cost of energy (per kwh), since additional pump costs are, as a first approximation, proportional to the energy consumption.

In summary, the costs discussed in this report are relative and incremental. To obtain total water costs, terms not proportional to area or energy consumption would have to be included. For rough estimates of total water costs for a given set of conditions (or designs), however, it is probably sufficient to evaluate C_A from the total daily plant cost and the area. C_E may then be the unit energy cost (with or without pumps).

4.3 Description of Program

A computer program called HYPFIL, written in FORTRAN IV, was developed to solve this system of equations and to optimize a given set of flow parameters to obtain the minimum water cost. The independent variables which may be specified include:

a. Feed Concentration and Concentration Ratio (CR)

Defined by:

$$CR = x_{re} / x_f. \quad (19)$$

Fractional water recovery, F_p , is related to concentration ratio by:

$$F_p = (CR - 1) / \left[CR - \left(1 - \bar{R}_{obs} \right) \right]. \quad (20)$$

b. Inlet Flow Conditions

- i. Inlet pressure.
- ii. Inlet velocity.

c. Flow Geometry

- i. Tube diameter.
- ii. Plant taper.
- iii. Turbulence promoter characteristics (If used).

d. Physical Properties of Solution (supplied as built-in functions of concentration and temperature)

- i. Osmotic pressure.

- ii. Viscosity.
- iii. Density.
- iv. Diffusivity of solute (or Schmidt number directly).
- e. Temperature
- f. Membrane Properties
 - i. Permeability.
 - ii. Salt rejection as a function of concentration.
 - iii. Salt rejection as a function of pressure.
- g. Economic Parameters
 - i. Daily plant cost per square foot of area (exclusive of energy costs).
 - ii. Unit cost of power.
 - iii. Brine disposal cost.
 - iv. Pressurizing pump efficiency.
 - v. Recovery turbine efficiency (zero if no energy recovery).
 - vi. Blending option.
- h. Optimization Parameters
 - i. Variables to be optimized (usually inlet pressure, inlet velocity, and concentration ratio).
 - ii. Maximum permissible average salt concentration in combined product.
 - iii. Inequality constraints on variables to be optimized and the minimum Reynolds number allowed.

The numerical procedure developed to approximate the differential process is based on subdividing the tube into a number of subsections such that the concentration change in each section is equal and arbitrarily small. For a given feed concentration, concentration ratio, tube diameter, inlet flow conditions, and membrane properties; the concentration polarization, effective pressure, and salt concentration

in the product are calculated at the inlet to the first subsection. An iterative procedure is used to calculate concentration polarization, effective pressure, and salt concentration in the product at the end of the first subsection, as well as the length of the subsection, the incremental water recovery, and the associated pressure and velocity drop over the subsection. This procedure is continued recursively down the tube, and the pertinent variables are summed to obtain the overall water recovery, tube length, and pressure drop. From these results, the area and pumping requirements are calculated. Then a cost is calculated for given economic parameters.

A general nonlinear optimization procedure developed by W. L. Kephart, et al,⁽⁵⁾ based on the gradient search,⁽⁶⁾ is provided to estimate flow conditions where optimum performance can be expected. Inequality constraints such as maximum velocity and pressure or minimum water recovery are imposed to restrict the parameters being optimized to specified domains. Additional constraints using Lagrange multipliers are employed, as described by Cross,⁽⁷⁾ to keep the flow conditions in the turbulent range and to limit the product salt concentration. It is also possible to obtain results at nonoptimum conditions; eg, with fixed pressure rather than optimum pressure, with the computer program developed.

4.4 Numerical Procedure

4.4.1 Program Control and Optimization

4.4.1.1 Main Program HYPFIL - The first portion of the main program handles initialization and the input of data required for material balance calculations, economic calculations, and control of the optimization. The second portion of the main program is a nonlinear optimization procedure based on gradient search. The optimization portion of HYPFIL evolved from an early version of Program OPTM, developed by W. L. Kephart, et al.⁽⁵⁾ The HYPFIL optimization is basically equivalent to a subset of the most recent version of OPTM (SHARE Release SDA 3541) and operates in the same manner. One of the modifications made by the authors was to remove all the unused sections and variables from the program.

As described in Subsection 4.4.1.2 (Page 39), Subroutine LINCON handles three types of constraints: variable boundaries, linear and nonlinear equality constraints, and linear and nonlinear inequality constraints. The current version of the program has no equality constraints and two inequality constraints (maximum product salt concentration and minimum Reynolds number).

1. Initialization for Loop 1 (data input) - The following initialization is done before starting Loop 1, the data input loop:

- a. Initialize ITIME0 (the initial value of the total elapsed time), ITIME1 (the initial time for each case), and DATE (the date for printing) by:

```
ITIME0 = ICLOCK(0),
ITIME1 = ITIME0, and
CALL IDAY (DATE).
```

ICLOCK is a local (Oak Ridge) function giving an integer which is the time expressed in 0.01 second. IDAY is a local subroutine which supplies an eight-character date for printing on the list. (See List B-1, Appendix B, and Subsections 4.4.1.5 and 4.4.1.6, Pages 48 and 49, for dummy routines that have been supplied for ICLOCK and IDAY.)

- b. Set all 56 Namelist variables to their initial values. (See Table 1 for an explanation of the variables and their initial values.) These variables will remain the same until changed by the input statement at the beginning of Loop 1. See Step 2.a, this subsection, for an explanation of the input card formats.

The initial values supplied serve as a test case. The output should be as shown in Lists B-2 through B-5, Appendix B.

2. Begin Loop 1 - This section reads in one set of data on each pass and handles the initialization required before beginning Loop 2.

- a. Read in one set of data on each pass according to the standard Namelist convention. Detailed operating instructions are given in Sections 5.1 (Page 83) and 5.3 (Page 87).

Once a Namelist variable has been changed, it will keep that value for the rest of the computer run unless it is reset or optimized. Therefore, it is not necessary to keep resetting variables to the same value in each data set.

The program is terminated by reading an end-of-file code after processing all of the data.

- b. List the values of all 56 Namelist variables before starting the optimization.
- c. Skip this case and go back to Step 2.a if any of the following variables are zero or negative: XB(1), P; XB(2), V; XB(3), CR; XB(5), D; PERM; or CFEED.
- d. Call Entry SETUP in Subroutine TUBE to initialize certain variables (see Subsection 4.4.2.2, Page 64).

Table 1
SUMMARY OF NAMELIST INPUT VARIABLES

Variable Name	Initial Value	Description
P	50	XB(1), Inlet pressure, POF (atm)
V	5	XB(2), Inlet velocity, VIN (ft/sec)
CR	2	XB(3), Concentration ratio (rej conc/feed conc)
TAPER	0	XB(4), Taper
D	0.4545	XB(5), Inside tube diameter, 4 hydraulic radii (inches)
PL	0	XL(1), Lower variable boundary for P (atm)
VL	0.001	XL(2), Lower variable boundary for V (ft/sec)
CRL	1.001	XL(3), Lower variable boundary for CR (rej conc/feed conc)
TAPERL	-30	XL(4), Lower variable boundary for TAPER
DL	0	XL(5), Lower variable boundary for D (inches)
PU	750	XH(1), Upper variable boundary for P (atm)
VU	100	XH(2), Upper variable boundary for V (ft/sec)
CRU	15	XH(3), Upper variable boundary for CR (rej conc/feed conc)
TAPERU	30	XH(4), Upper variable boundary for TAPER
DU	3	XH(5), Upper variable boundary for D (inches)
PWT	1	WT(1), Weighting factor for P
VWT	1	WT(2), Weighting factor for V
CRWT	1	WT(3), Weighting factor for CR
TWT	1	WT(4), Weighting factor for TAPER
DWT	1	WT(5), Weighting factor for D
NVARL	3	Minimum number of variables to be optimized
NVARU	3	Maximum number of variables to be optimized
INCR	50	Number of tube increments
WP	1×10^6	Water production rate (gals/day)
TEMP	70	Temperature (F)
FRATIO	1	"Roughness factor" used to modify Fanning friction factor
XRATIO	1	Factor to modify exponent for turbulence promoters
REJECT	0.95	Salt rejection, initial guess (fraction)
ALPH	0.001	Initial step size factor for gradient evaluation
BETA	0.005	Initial step size factor for searching
LOOPS	1	Number of loops per case, see Step 3.d (1 or 2)
MAXTIM	1080000	Maximum time allowed per case (0.01 sec)
LOGCR	0	Indicator for adjusting CR (0 = no)(1 = yes)
CRADJ	0.9	CR adjustment factor, see Step 3.c (fraction)
LIST1	1	List indicator (0 = list on Tape 1)(1 = skip)
LIST6	0	List indicator (0 = list on Tape 6)(1 = skip)
PERM	0.0005	Membrane permeability (kgal ft ⁻² atm ⁻¹ day ⁻¹)
CFEED	0.005	Feed concentration (wt fraction NaCl)
PLIM	0.0005	Maximum allowable salt concentration of product (wt fraction)
RELIM	2100	Minimum allowable Reynolds number
MTYPE	2	Membrane type (1 = ion exchange)(2 = poly fit)
CSTAR	0.35	Ion-exchange membrane capacity (moles/kg H ₂ O)
CONST	165	Salt rejection versus flux, see Section 4.2, Equation 13a (ft ² day/kgal)
A3	0.05	Polynomial fit membrane, constant (dimensionless)
B3	0	Polynomial fit membrane coefficient
C3	0	Polynomial fit membrane coefficient
D3	0	Polynomial fit membrane coefficient
XP	0.25	Reynolds number exponent used in Fanning friction factor
ITUBE	2	Tube geometry (1 = round)(2 = rectangular)
RA	10	Aspect ratio, rectangular tubes (membrane side/solid side)
IBLEND	2	Blend feed with product water (1 = no)(2 = yes)
CA	1	Unit cost of membrane area (cents ft ⁻² day ⁻¹)
CE	1	Unit cost of electric power (cents/kwh)
CD	0	Unit cost of waste disposal (cents/kgal)
EP	0.7	Pump efficiency (fraction)
ET	0	Recovery turbine efficiency (fraction)

3. Begin Loop 2 - Loop 2, within Loop 1, performs one or more optimizations for each pass through Loop 1.

- a. The five independent variables available for optimization are in the XB array:

XB(1), P	Inlet pressure
XB(2), V	Inlet velocity
XB(3), CR	Concentration ratio
XB(4), TAPER	Taper
XB(5), D	Tube diameter

- b. On the first pass through Loop 2, N, the number of variables to be optimized, is equal to NVARL, the lower limit. If NVARU, the upper limit on the number of variables to be optimized, is greater than NVARL, then NVARU - NVARL + 1 optimizations are done. N has the value NVARU on the final pass. Any variables that are not being optimized will remain constant; any variable that would normally be optimized can be forced to remain constant by causing the corresponding weighted range to be equal to zero, as explained in Step e (15), Page 28, and Subsection 5.3.2, Page 88.

For example, if NVARL = 3 and NVARU = 4, the first three variables, starting with P, will be optimized on the first pass. On the second and final pass, the fourth variable will be optimized as well. The optimum conditions for the first pass become the starting conditions for the second pass. If desired, inlet pressure could be removed from the optimization by setting PWT = 0 or by setting PL and PU equal.

- c. If this is the second or greater pass through Loop 2 and the flag LOGCR is 1, then reduce CR to a "safer" value by:

$$XB(3) = (XB(3) - 1) * CRADJ + 1.$$

- d. When LOOPS = 1, one pass through Loop 2 is made for each value of N. A value of 50 for INCR, the number of tube increments, has been found to be a good compromise between accuracy and speed on a computer such as the IBM 360/75. In some cases, time can be saved by setting flag LOOPS = 2. On the first pass (LOOPS = 2, LOOP = 1) the normal value of INCR is saved, INCR is set to five increments, and some of the printing is skipped by:

INCRSV = INCR,
INCR = 5, and
LIST6 = 1.

The optimization proceeds at about five times the rate it does when INCR is 50. The first pass ends when an optimum is found or after one minute of

elapsed time (if a clock is available). A second pass is made (LOOPS = 2, LOOP = 2) to find a more accurate optimum. The normal value of INCR is restored and the printing is turned on by:

INCR = INCRSV and
LIST6 = 0.

e. Initialize the following variables:

- (1) Perturbation factor used when approximating the gradient

ALPHA = ALPH (= 0.001).

- (2) Step size factor for searching

BET = BETA (= 0.005).

- (3) Type of optimization (- 1 = minimization, + 1 = maximization)

CMM = - 1.

- (4) Scale factors 1 and 2 for changing step size

SF1 = 1.5 and
SF2 = 0.9.

- (5) Reduction factor used after unsuccessful step

RF = 0.25.

- (6) STEP prints one of four characters for each function evaluation:

- (a) "I" for initial call.
- (b) "G" for gradient evaluation.
- (c) "L" for vector following gradient, handled by Subroutine LINCON.
- (d) "A" for acceleration vector handled by Subroutine LINCON.

- (7) Counter for number of function evaluations:

K = 0.

- (8) Flag used in determining convergence:

$$ISTOP = 1.$$

- (9) Counter for number of "linear optimums" (best point along a successful vector) available for use in acceleration steps (parallel tangent technique):

$$KNTVEC = -1.$$

- (10) Counter for number of consecutive gradient failures:

KT2 = 3. (A gradient failure is a gradient that yields no successful steps. Note that KT2 counts backwards.)

- (11) Flag showing type of vector (1 or 2 when following gradient, 3 during acceleration step):

$$KT5 = 1.$$

- (12) Vector flag, counts how many times each variable has been held on a boundary:

$$KNT(I) = 0.$$

- (13) Array for storing most recent vector direction for use in ridge analysis:

$$FP1(I) = 0.$$

- (14) Ridge analysis scale factor array used to retard "oscillation" of one or more variables:

$$SLEFAC(I) = 1.$$

- (15) Calculate the weighted ranges by:

$$R(I) = ABS((XH(I)-XL(I)) * WT(I))$$

where XH is the upper limit or bound (= PU, VU, CRU, TAPERU, DU), XL is the lower limit (= PL, VL, CRL, TAPERL, DL), and WT is a weighting factor (= PWT, VWT, CRWT, TWT, DWT). The range is used to scale or "normalize" the variables in an attempt to make the optimization run more efficiently by making long elliptical contours appear more circular. The term (XH-XL) is a good first approximation. Further adjustment may be accomplished by adjusting the weighting factors. Increase

the factor to make a variable change more rapidly; decrease the factor to retard a component.

This is an arbitrary modification made by the authors simply because it showed merit. If one R is off by an order of magnitude relative to another R, it may take 10 times as long to find the optimum.

If the range, R, for any variable is zero, that variable will be held constant and not optimized. This is accomplished by setting the corresponding weighting factor to zero or by making the upper and lower bounds equal. If all the R's are zero, the results are printed out for the given set of conditions.

(16) Let RMAX be the largest value in the R array.

(17) If a variable has a finite range, R, then set the flag

$$\text{LIM}(I) = 0.$$

If a variable has zero range, R, set the flag LIM to remove it from the optimization by:

$$\text{LIM}(I) = 1.$$

f. Call Subroutine EVAL (see Subsection 4.4.1.4, Page 47) which uses Subroutines TUBE and ECON to get a function evaluation for the starting conditions.

g. The two inequality constraints are:

- (1) product salt concentration CPM (= CPMAX) $\leq$ product limit PLIM, and
- (2) minimum Reynolds number REM (= REMIN) $\geq$ Reynolds number limit RELIM.

If either constraint is violated at the starting point, attempt to find an acceptable starting point by calling entry ADJUST in Subroutine TUBE which reduces the concentration ratio by:

$$\text{CR} = (\text{CR}-1) * ((\text{IGOOD})/\text{INCR}) + 1,$$

where IGOOD is the number of the last tube increment before the constraint was first violated. Go back to the previous step (Step f) and call Subroutine EVAL until either an acceptable starting point is found or until CR becomes less than its lower limit, CRL. In the latter situation, abandon this case and go back to Item 4.4.1.1.2 (Page 24) for the next case.

- h. Initialize TCPM, TREM, and TCPE which hold initial values, or recent successful step values of the constraint variables CPM (= CPMAX), REM (= REMIN), and CPE (= CPEND). At this time, only the first two inequality constraints are used in Subroutine LINCON:

$$\begin{aligned} \text{TCPM} &= \text{CPM}, \\ \text{TREM} &= \text{REM}, \text{ and} \\ \text{TCPE} &= \text{CPE}. \end{aligned}$$

- i. At this point, initialization (done only once per optimization) is finished.
4. Begin New Vector Direction - This section does the initialization when starting either a gradient vector direction or an acceleration (parallel tangent) vector direction.

- a. Initialize L, a counter for successful steps along one vector, by:

$$L = 0.$$

- b. Initialize KT, which counts (backwards from 4) the number of unsuccessful steps encountered for this vector direction, by:

$$KT = 4.$$

- c. Advance KNTVEC, which counts the number of "linear optima", by:

$$\text{KNTVEC} = \text{KNTVEC} + 1.$$

- d. Set SF, the actual scale factor, equal to scale factor 1 (= 1.5) by:

$$\text{SF} = \text{SF1}.$$

- e. Set SCALE, the cumulative scale factor used in determining the step size, equal to CMM (= -1.0) by:

$$\text{SCALE} = \text{CMM}.$$

- f. Reset CPM, REM, and CPE for use in calculating the partial derivatives of the constraints by:

$$\begin{aligned} \text{CPM} &= \text{TCPM}, \\ \text{REM} &= \text{TREM}, \text{ and} \\ \text{CPE} &= \text{TCPE}. \end{aligned}$$

5. Acceleration Step - HYPFIL uses the parallel tangent method for accelerating a steepest descent algorithm. It makes use of the fact that a gradient calculated at a linear optimum located along a given gradient is perpendicular to the given gradient. Thus, the third gradient would be parallel to the first. Buehler, Shah, and Kempthorne⁽⁸⁾ have shown that if the response surface contours are elliptical, the global optimum lies on a vector passing through the linear optimums of two parallel lines. However, as few real response surfaces are elliptical, the arguments for including this method are empirical. It costs only two function evaluations and has shown merit in many cases.

The first acceleration (parallel tangent) vector direction is calculated after three successful (position improved) gradient vector directions. Thereafter, an acceleration step is attempted after each successful vector direction that follows a gradient calculation. The "acceleration" (parallel tangent) vector direction should not be confused with the "acceleration" of step sizes that occurs for both parallel tangent and gradient vector directions when the cumulative scale factor, SCALE, is multiplied by SF (=SF1 = 1.5) after each successful step along the direction of the vector.

The following example will help explain the use of the acceleration vector direction and conditions at this point in the program. The example is based on a case in which each vector direction is successful (results in an improved position).

<u>KNTVEC</u>	<u>Last Improved Point</u>	<u>KT5</u>	<u>Reset KT5</u>	<u>New Vector</u>	<u>Store Improved Point</u>	<u>Acceleration Vector Points</u>
0		1	1	grad		
1	1	1	1	grad	1	
2	2	1	2	grad	2	
3	3	2	3	accel		1, 3
4	4	3	2	grad	4	
5	5	2	3	accel		2, 5
6	6	3	2	grad	6	
7	7	2	3	accel		4, 7
8	8	3	2	grad	8	

- If the new vector direction is to be a gradient, then store the last point in array ACC2, adjust KT5 if necessary, and transfer to Item 4.4.1.1.6 (Page 32) for calculation of the gradient.
- If the new vector direction is an acceleration step, then calculate the vector direction ANUM by:

$$ANUM(I) = (XB(I) - ACC1(I)) * CMM * RMAX/R(I).$$

Set ANUM(I) to zero if R(I) is zero.

Reset KT5 to 3.

Call LINCON to adjust this vector direction in the same manner as the gradient is adjusted.

Transfer to Step 7.b (Page 34) in order to bypass adjustment of the vector direction by SLEFAC, the ridge analysis retarding factor.

6. Approximation of Partial - A simple, one-sided, forward difference approximation method is employed to estimate the partial derivatives of the objective function and the constraint functions. In order to scale or "normalize" the variables to make the optimization run faster by making long elliptical contours appear more circular, the gradient is calculated by:

$$[P] = (F(XB + ALPHA * R) - F(XB))/ALPHA.$$

- a. Store the constraint variables for the last good point by:

SCPM = CPM,
SREM = REM, and
SCPE = CPE.

- b. Store the coordinates of the last good point XB in the array X by:

$$X(I) = XB(I).$$

- c. If LIM(J) is 1 (range R(J) is zero) then set FP, the partial derivative of the function, and FPC, the partial derivatives of the constraints, equal to zero by:

FP(J) = 0,
FPC(J,1) = 0,
FPC(J,2) = 0, and
FPC(J,3) = 0,

and transfer to Step g after all the independent variables have been examined.

- d. If an independent variable is on a boundary (XL or XH), then use counter KNT(J) to hold the variable on the boundary for five gradient vectors. This step is accomplished by setting FP and FPC to zero as in Step c. After KNT(J) reaches five, the variable is released, KNT(J) is reset to zero, and the partial derivatives are calculated in the normal manner.

e. Set:

$$X(J) = X(J) + \text{ALPHA} * R(J),$$

where ALPHA is an arbitrary perturbation increment (initially 0.001). Call Subroutine EVAL with this point to get a function evaluation FH.

f. Approximate the partial derivative of the function and the partial derivatives of the constraints by:

$$\begin{aligned} \text{FP}(J) &= (\text{FH} - \text{SL}) / \text{ALPHA}, \\ \text{FPC}(J, 1) &= -(\text{CPM} - \text{SCPM}) / \text{ALPHA}, \\ \text{FPC}(J, 2) &= (\text{REM} - \text{SREM}) / \text{ALPHA}, \\ \text{FPC}(J, 3) &= -(\text{CPE} - \text{SCPE}) / \text{ALPHA}, \text{ and} \\ \text{ANUM}(J) &= \text{FP}(J). \end{aligned}$$

At this time, Subroutine LINCON utilizes only the first two constraints.

g. Initialize the KON array to zero by:

$$\text{KON}(I) = 0.$$

KON(1) and KON(2) apply to the first and second constraints, respectively. Each KON will be set to 1 in Subroutine LINCON when that constraint is first violated. See Subsection 4.4.1.2, Subroutine LINCON (Page 39), for further information concerning KON and the adjustment of the partials. The KON array is reinitialized only for gradient vector directions, not acceleration vector directions. Again, this is an arbitrary decision that has shown merit.

h. Call Subroutine LINCON to adjust the partials. No adjustments are made at this time, however, because this version of the program has no equality constraints.

7. Ridge Analysis - This section of the program was inserted to detect variables that lie along a ridge (or, in this case, a valley). Gradient-search techniques are often inefficient when dealing with ridges, and a "sawtooth" search pattern results. This ridge-analysis technique will help overcome problems caused by poor scaling or the nature of the problem. An "oscillating" variable can be detected by a change in sign of the partial derivative. The gradient elements (vector components) for such variables can be reduced to give a skewed vector direction that tends to point more nearly along the ridge (down the valley).

If the program detects a change in sign between ANUM, the current vector direction, and FPI, the last vector direction, then the ridge analysis scale factor,

SLEFAC(I) (originally 1.0), is multiplied by 0.64. If there is no change in sign, then SLEFAC(I) is multiplied by 1.25. SLEFAC is not allowed to exceed 1.0.

- a. Use SLEFAC to adjust the gradient (not acceleration vectors) by:

$$\text{ANUM}(J) = \text{ANUM}(J) * \text{SLEFAC}(J).$$

- b. Call Subroutine LINCON to adjust the partials.

8. Step Size Determination - This section calculates DX, a term to be used in the proposed step along the vector direction.

- a. Compute DX for the gradient vector by:

$$\text{DX}(J) = \text{ANUM}(J) * R(J) * \text{BET}/\text{AMAX},$$

where AMAX is the largest value of R(J) and BET is an arbitrary increment size (initially 0.005). If R(J) is zero, set DX(J) equal to zero.

- b. Compute DX for the acceleration vector by:

$$\text{DX}(J) = \text{ANUM}(J) * R(J) * 0.05/\text{RMAX},$$

where 0.05 is an arbitrary factor that limits the first step to 5 percent of the distance between the two points used to project the acceleration vector direction.

9. Boundary Check - If necessary, the step sizes are adjusted so that no variable lies outside its domain or boundaries. If no boundaries are violated, the coordinates of the new point are calculated as:

$$X(I) = \text{XB}(I) + \text{DX}(I) * \text{SCALE} * \text{BOUND},$$

where XB is the previous point and BOUND is 1.

If an upper limit, XH, is violated, then the fraction of the vector inside the domain, BOUNDT ($0 < \text{BOUNDT} < 1$), is calculated by:

$$\text{BOUNDT} = (\text{XH}(I) - \text{XB}(I))/(\text{X}(I) - \text{XB}(I)).$$

Similarly, if a lower limit, XL, is violated, the fraction of the vector inside the domain is given by:

$$\text{BOUNDT} = (\text{XB}(I) - \text{XL}(I))/(\text{XB}(I) - \text{X}(I)).$$

The smallest value of BOUNDT is applied to all the elements of the vector (step size $DX * SCALE * BOUNDT$) to give a new point at which one or more variables are on the boundaries and the remainder of the variables are inside the boundaries.

If a variable is being held on a boundary ($KNT(I) > 0$), no effort is made to scale this variable unless an acceleration step is being taken ($KT5 = 3$).

10. Evaluation of a New Point - HYPFIL calls Subroutine LINCON which, in turn, calls Subroutine EVAL for an evaluation of the objective function at the new point. If constraints are violated at this new point, Subroutine LINCON will adjust the variables until the constraints are no longer violated.

If the new cost value, F , is not smaller than the previous best cost, FB (note $CMM = -1$), then transfer to Item 4.4.1.1.12 (Page 37). If the new point is an improvement, then proceed to the next item, Item 11.

11. Successful Step - This section describes the action taken when the step just taken resulted in an improved position.

- a. Advance L , which counts successful steps along a vector by:

$$L = L + 1.$$

- b. Set ISTOP for use in determining convergence by:

$$ISTOP = 1.$$

- c. The new point is considered the best point thus far. Save its location, cost value, and constraint variables by:

$$\begin{aligned} XB(I) &= X(I), \\ FB &= F, \\ TCPM &= CPM, \\ TREM &= REM, \text{ and} \\ TCPE &= CPE. \end{aligned}$$

- d. If this is an acceleration (parallel tangent) vector, then transfer to Step i (Page 36).
- e. If L is 30 (30 good points along the vector direction) then the search is moving too slowly. Increase step sizes ALPHA and BET and set KT2 to 3 to indicate a successful step by:

$$ALPHA = ALPHA * 5,$$

$$\text{BET} = \text{BET} * 5, \text{ and} \\ \text{KT2} = 3,$$

and transfer back to Item 4.4.1.1.4 (Page 30) to start over with a new gradient or acceleration vector direction.

- f. If L is 10 (10 successful steps) and SF, the scale factor, is SF2 (= 0.9), due to a failure on the first step, then set SF equal to SF1 (= 1.5) to cause further steps to increase in size rather than decrease by:

$$\text{SF} = \text{SF1}.$$

Then transfer to Step i.

- g. If L is 10 and SF is already SF1 (= 1.5), then double the step sizes ALPHA and BET by:

$$\text{ALPHA} = \text{ALPHA} * 2 \text{ and} \\ \text{BET} = \text{BET} * 2,$$

and transfer to Step i.

- h. Similarly, when L reaches 20 successful steps, again double ALPHA and BET by:

$$\text{ALPHA} = \text{ALPHA} * 2 \text{ and} \\ \text{BET} = \text{BET} * 2,$$

and proceed to Step i.

- i. The cumulative scale factor (SCALE) is multiplied by scale factor SF by:

$$\text{SCALE} = \text{SCALE} * \text{SF}.$$

SF is equal to SF1 (= 1.5) if no unsuccessful steps have been taken or if it was reset from SF2 to SF1 after 10 successful steps following one unsuccessful step. SF is equal to SF2 (= 0.9) for 10 successful steps following one unsuccessful step.

Once, again, the "acceleration" or parallel tangent vector should not be confused with the "acceleration" of step size that occurs when SF is 1.5 for both gradient and acceleration vectors.

- j. KT2 is set to 3 to indicate a successful step by:

$$\text{KT2} = 3.$$

- k. If one variable was placed on a boundary for the first time ($IBOUND > 0$), then that element of the vector is set to zero and counter KNT is advanced by:

$$\begin{aligned} ANUM (IBOUND) &= 0 \text{ and} \\ KNT (IBOUND) &= KNT (IBOUND) + 1. \end{aligned}$$

- l. Transfer back to Step b of Item 4.4.1.1.7 (Page 34) to try another step along the same vector direction. End of section to handle successful steps.

12. Unsuccessful Step - This section describes the action taken following an unsuccessful step.

- a. KT, initially 4, counts (backwards) the number of unsuccessful steps along one vector direction. KT is reduced by:

$$KT = KT - 1.$$

At this point, KT will have one of two values (never 1). If KT is 3, proceed to the next step (Step b) because the first unsuccessful step has occurred. If KT is 2, then transfer to Step c to handle the second unsuccessful step.

- b. If there were any successful steps along this vector direction ($L \geq 1$) before the first failure ($KT = 3$), then revert to Item 4.4.1.1.4 (Page 30) to begin a new vector. This means that the last successful point becomes the terminal point along that vector direction. The parabolic fit used in other versions of the optimization program to refine the "linear optimum" along one vector direction is not employed because the possible improvement in position does not justify the function evaluation, feasibility check, possible adjustment, and checks to avoid using points that are not collinear.

If there were no successful steps along this vector direction preceding the first failure ($L = 0$), then the scale factor, SF, is set to SF1 (= 0.9) and the cumulative scale factor, SCALE, is reduced to one-fourth by reduction factor RF (= 0.25) by:

$$\begin{aligned} SF &= SF1 \text{ and} \\ SCALE &= SCALE * RF. \end{aligned}$$

Then transfer back to Item 4.4.1.1.9 (Page 34) to try the reduced step in the same direction. Contrary to expectations, in the constrained situations, a large number of successful steps may follow the first unsuccessful step.

- c. A second failure has occurred along this vector direction ($KT = 2$). If this has been an acceleration (parallel tangent) vector, or if there were any successful steps along this vector direction, then return to Item 4.4.1.1.4 (Page 30) for an evaluation of a new gradient at the last successful point. All the remaining sections of HYPFIL are reached only after a second failure in the direction of the gradient.
- d. Step sizes ALPHA and BET are reduced by a factor of 20 and ACC1 is placed in ACC2 to keep the acceleration steps in order by:

$$\begin{aligned} \text{ALPHA} &= \text{ALPHA}/20, \\ \text{BET} &= \text{BET}/20, \text{ and} \\ \text{ACC2(I)} &= \text{ACC1(I)}. \end{aligned}$$

- e. If the number of successful vector directions, KNTVEC, is less than 3, then KNTVEC is reduced and KT5 is set to 1 to request another gradient calculation by:

$$\begin{aligned} \text{KNTVEC} &= \text{KNTVEC} - 1 \text{ and} \\ \text{KT5} &= 1. \end{aligned}$$

If KNTVEC is 3 or greater, then a gradient calculation is requested by:

$$\text{KT5} = 3.$$

- f. KT2 is reduced by 1 by:

$$\text{KT2} = \text{KT2} - 1.$$

- g. At this point, KT2 is either 2 or 1. KT2 was initialized at 3 and reset to 3 after any successful step. One convergence criterion is that KT2 be 1. If KT2 is 2, then one set of two successive failures along one gradient vector direction has occurred. If KT2 is 2, return to Item 4.4.1.1.4 (Page 30) to try a new gradient or acceleration vector direction.
- h. At this point, KT2 is 1 and two gradient evaluations in a row have failed because their first two steps failed. If this is the first time this point has been reached ($\text{ISTOP} = 1$), then check to see if any variables are on a boundary ($\text{KNT} > 0$). If not ($\text{KNT} = 0$), then the case is finished and a transfer is made to Item 4.4.1.1.13 (Page 39).

If ISTOP is 1 and one or more variables were on a boundary ($\text{KNT} > 0$), then the variable is released, ISTOP is set to 2, KT2 is set to 2 to allow at least one more gradient evaluation, and KT5 is set to 3 to request a gradient next, and the KNT array is set to 5 by:

ISTOP = 2,
 KT2 = 2,
 KT5 = 3, and
 KNT(I) = 5,

and a transfer is made back to Item 4.4.1.1.4 (Page 30) for a gradient evaluation at the last successful point.

If this point is reached with KT2 = 1 and ISTOP = 2, then proceed to Item 13 to terminate the case because a variable on a boundary has already been "released" once at this point.

Termination of the optimization (usually the failure of the first and reduced step along the direction of the gradient, followed by reduction of ALPHA and BET by 20, followed by failure of the first and reduced step along approximately the same gradient direction) is based on failures, not the magnitude of change of the cost function. This method seems to work acceptably. The procedure would be expected to quit sooner or later due to "noise" in the cost function.

13. Optimization Completed - HYPFIL has found an optimum (minimum) cost for this case.
 - a. Call Subroutine EVAL one last time so that Subroutine TUBE and Subroutine ECON can print out the detailed conditions for the optimum point, XB.
 - b. Calculate and print in seconds and minutes the elapsed time for this case (IT1, T1) and for all cases (IT0, T0).
 - c. If LOOPS is 2 and LOOP is 1, then go back to Item 4.4.1.1.3, Step d (Page 26) for a second pass through Loop 2 to refine the optimum using the original value of INCR, the number of tube increments.
 - d. End of Loop 2 within Loop 1. If N has reached the value NVARU, then proceed to the next step since Loop 2 has been completed. If N is less than NVARU, then increase N by 1 and transfer back to Item 4.4.1.1.3, Step d for another pass through Loop 2.
 - e. End of Loop 1. Transfer back to Item 4.4.1.1.2 (Page 24) for another pass through Loop 2. The program normally terminates by exhausting the input data and reading an end of file at Step a of Item 4.4.1.1.2.

4.4.1.2 Subroutine LINCON - The purpose of Subroutine LINCON, as described by Cross, (7) is to allow the gradient-search optimization code to handle constrained problems as well as unconstrained ones. LINCON deals with three classes of

constraints: (1) variable boundaries; (2) equality, linear and nonlinear; (3) inequality ($\geq$), linear and nonlinear.

The basic idea is to project the steepest descent gradient onto the constrained space. The variable boundaries are easily projected. Merely set the appropriate partial derivatives to zero, both of the gradient and the constraint partial derivative matrix. This action effectively removes the variable from the search.

The equality and inequality constraints are accounted for by projecting the gradient onto the set of all equality constraints plus those inequality constraints that have been violated. However, before the gradient can be projected onto an inequality, it must be considered an equality constraint, and the offending variables placed on the "equality". Note that the equality mentioned is merely the edge of the constraint.

Hence, LINCON has two distinct functions: (1) adjust the variables to satisfy the constraints (described in Items 4.4.1.2.1, Page 41, through 4.4.1.2.6, Page 43); and (2) project the partials (gradient vector direction) or acceleration vector direction onto the constrained space (described in Items 4.4.1.2.7, Page 45, through 4.4.1.2.10, Page 47).

During the optimization, Subroutine LINCON is called to adjust the variables to the constraints before each step is attempted. LINCON checks the inequality constraints (and boundaries) and adjusts the variables, if necessary, to lie in the feasible region. No projection of the vector direction is undertaken at this time, but each inequality violated is added to the set of constraints to be used the next time the vector direction is projected.

Each time a new set of partials is calculated and after each successful step, LINCON is called to adjust the partials by projecting them onto the constraints.

For case of the one-stage hyperfiltration code, there are:

0 equality constants (NLINEQ = 0), and
 2 inequality constraints (NNOTEQ = 2).
 Product purity: $CPM \leq PLIM$ (= 0.000500).
 Reynolds number: $REM \geq RELIM$ (= 2100).

A third inequality constraint, product water concentration less than feed concentration ($CPE \leq CFEED$), is not used in this version of the program, but the capability for using it was left in the program. Care should be taken to avoid a situation in which the total number of constraints violated exceeds the number of optimization variables.

The two uses of Subroutine LINCON will be described separately, even though they overlap somewhat in the FORTRAN program.

1. If a vector direction is to be adjusted to satisfy the constraints ($IFLAG = 3$), then transfer to Item 4.4.1.2.7 (Page 45) and proceed from there. If the variables are to be adjusted to satisfy the constraints ($IFLAG = 1$), then proceed to the next item, Item 2.

2. Initialization for Loop 1 - The following initialization is done before starting Loop 1:

- a. The iteration counter, ITER, is initialized by:

$$ITER = 0.$$

- b. Since there are no equality constraints, NLINEQ is initialized by:

$$NLINEQ = 0.$$

- c. NNOTEQ, the number of inequality constraints, is set to two by:

$$NNOTEQ = 2.$$

3. Begin Loop 1 - This loop adjusts variables to satisfy constraints. Counter is ITER.

- a. NLIN, the total number of constraints that enter into the adjustments, is initialized by:

$$NLIN = NLINEQ (= 0).$$

- b. Call Subroutine EVAL with point X to get cost FH ($FL = FH$) and constraint variables CPM ($= CPMAX$), REM ($= REMIN$), and CPE ($= CPEND$).

- c. Evaluation counter KFLG (K in HYPFIL) is advanced by $KFLG = KFLG + 1$.

- d. If an error occurred in Subroutine TUBE ($NFLAG \neq 1$) ($IER \neq 1$), then Subroutine LINCON returns to HYPFIL at this point without making any effort to adjust the variables.

4. Calculation of Inequality Constraints.

- a. BSUMNT is the $[\Delta g]$ array, an m-dimensional column vector representing the change in constraints $[g]$ necessary for their satisfaction. $[\Delta g]$ is calculated by:

$$\begin{aligned} \text{BSUMNT}(1) &= \text{PLIM} - \text{CPM}, \\ \text{BSUMNT}(2) &= \text{REM} - \text{RELIM}, \text{ and} \\ \text{BSUMNT}(3) &= \text{CFEED} - \text{CPE}. \end{aligned}$$

The equations are arranged so that a positive value ($\text{BSUMNT} \geq 0$) is acceptable and a negative value ($\text{BSUMNT} < 0$) is a violation. In the first constraint, product water, of concentration CPM, is acceptable if $\text{CPM} \leq \text{PLIM}$ ($= 0.000500$). In the second constraint, turbulence is assured if REM (minimum Reynolds number) $\geq \text{REMIN}$ ($= 2100$). The third constraint is not used at this time.

- b. If there is a constraint violation ($\text{BSUMNT}(1) < 0$), then set flag $\text{KON}(1) = 1$. Once KON is set to 1, it is reset to 0 only after a new gradient vector direction is calculated in HYPFIL. This means that an acceleration (parallel tangent) vector direction that originally pointed toward an unconstrained region would still go through an adjustment of direction and variables for each KON that is not zero.
- c. The NLINLD array contains the indices of all the constraints to be projected onto. Since all of the equality constraints are projected onto, the first NLINEQ positions of NLINLD contain the indices of all the equalities. In this case, there are no equality constraints ($\text{NLINEQ} = 0$).
- d. Next in NLINLD are stored the indices of the inequality constraints that have been violated ($\text{KON} = 1$). Counter NLIN is advanced by one for each violation.
- e. At this point, NLIN, the total number of entries in array NLINLD, ranges from $\text{NLINEQ} (= 0)$ to $\text{NLINEQ} + \text{NNOTEQ} (= 2)$. If NLIN is zero, no adjustment of variables is necessary (unconstrained condition) and Subroutine LINCON returns control to HYPFIL. Proceed if NLIN is 1 or 2.
- f. If there are no equality constraints ($\text{NLINEQ} = 0$) (such is the case) and if there are no violations ($\text{BSUMNT} \geq 0$) and if IFLAG is 2 (second or greater pass), then return to HYPFIL with correctly adjusted variables. Once a constraint has been violated ($\text{NLIN} > 0$), all further steps along the same (adjusted) vector direction will undergo a minimum of one adjustment even if it is unnecessary.

5. Boundary Check

- a. If this is the fourth iteration ($\text{ITER} = 4$), then skip the step in which variables outside the boundaries are placed on the boundaries by moving ahead to Step c. This arbitrary choice was made because it appeared to help.

- b. If a variable $X(I)$ is outside the upper boundary $XH(I)$, the variable is placed on the boundary by:

$$X(I) = XH(I),$$

or if a variable is outside its lower boundary, it is placed on the boundary by:

$$X(I) = XL(I).$$

In either case, the partial derivative (vector direction) is set to zero by:

$$FPC(I, JT) = 0,$$

where JT is 1 or 2 (the number of the constraint).

- c. The iteration counter is advanced by:

$$ITER = ITER + 1.$$

- d. If ten iterations ($ITER > 10$) are not sufficient to adjust the variables, then the procedure in LINCON is terminated. The cost function is set to a large number by:

$$FL = 101000$$

in order to cause an unsuccessful step, a note is printed, and Subroutine LINCON returns control to HYPFIL. Failure to converge is usually due to the fact that the partial derivatives of the constraints (FPC) are no longer a good approximation at relatively large distances away from the point where they were calculated.

6. Equation Solving

- a. The $[\Delta g]$ or BSUM vector is loaded with the values of the equality constraints (none in this case) and those inequality constraints that have been violated (a total of either 1 or 2) by:

$$BSUM(I) = -BSUMNT(K).$$

- b. At this point, a convergence check is made. If the BSUMNT values are within an arbitrary amount of perfect adjustment ($BSUMNT = 0$), then the adjustments are terminated. If

$$BSUMNT(1) \geq -1 * 10^{-10}, \text{ and if}$$

$$\text{BSUMNT}(2) \geq -0.02,$$

and if IFLAG is 2 (at least one adjustment has been made), then adjustment is successful and Subroutine LINCON returns control to HYPFIL.

- c. The AX matrix generated by:

$$\text{AX}(J, L) = \text{AX}(J, L) + (\text{FPC}(I, LT) * \text{FPC}(I, JT))$$

corresponds to the $[GG^T]$ matrix described by Cross.⁽⁷⁾ This matrix also includes all equality constraints (none in this case) and those inequality constraints that have been violated (1 or 2). $[G^T]$ is the transpose of $[G]$.

- d. If any of the principal diagonal elements of $[GG^T]$ are zero, they are set to 1 to avoid singularity problems by:

$$\text{AX}(I, I) = 1.$$

- e. The equation:

$$[\Delta g] = [GG^T][\gamma],$$

is then solved for $[\gamma]$,

where:

$[\Delta g]$ represents the change in constraints necessary for their satisfaction,

$[GG^T]$ the square constraint partial derivative matrix, and

$[\gamma]$ the column vector which when applied to $[G^T]$ yields the changes in the variables $[\Delta X]$ necessary to arrive at a feasible point.

If there is only one violation ($\text{NLIN} = 1$), then the equation is solved directly by:

$$\text{BSUM}(1) = \text{BSUM}(1)/\text{AX}(1, 1).$$

If NLIN is greater than 1, a linear simultaneous equation solving routine AMATQ4 (see Subsection 4.4.1.3, Page 47) is used. The solution vector, $[\gamma]$, is now in BSUM.

- f. Finally, the desired changes in the variables

$$[\Delta X] = [G^T][\gamma] \text{ are computed by:}$$

$$X(I) = X(I) + BXUM(J) * FPC(I, JT) * R(I),$$

where R is the weighted range used in HYPFIL to scale the variables.

- g. IFLAG is set to 2, indicating that at least one pass has been made at adjusting the variables.
- h. Any variables outside the boundaries are placed on the boundaries by:

$$\begin{aligned} X(I) &= DMAX1(XL(I), X(I)) \text{ and} \\ X(I) &= DMIN1(XH(I), X(I)). \end{aligned}$$

- i. End of Loop 1. Transfer back up to Item 3 (Page 41) for another iteration. Control of the program will eventually be returned to HYPFIL from one of the five possible return points previously described.
7. Initialization for Loop 2 - The initialization for Loop 2, the adjustment of a gradient or parallel tangent vector direction, is done first.
- a. The following variables are initialized by:

$$\begin{aligned} \text{ITER} &= 0, \\ \text{NLINEQ} &= 0, \\ \text{NNOTEQ} &= 2, \text{ and} \\ \text{NLIN} &= \text{NLINEQ} (= 0), \end{aligned}$$

where these variables (iteration counter, number of equality constraints, number of inequality constraints, and total number of effective constraints) are the same as described in Items 2 and 3 (Page 41).

- b. As in Step 4.c (Page 42), there are no equality constraints ($\text{NLINEQ} = 0$) to be stored in the NLINLD array.
- c. As in Step 4.d (Page 42), the indices of the inequality constraints that have been violated ($\text{KON} = 1$) are stored in NLINLD by:

$$\text{NLINLD}(\text{NLIN}) = 1 + \text{NLINEQ},$$

where counter NLIN was advanced by one for each violation by:

$$\text{NLIN} = \text{NLIN} + 1.$$

- d. If NLIN, the total number of entries in array NLINLD, is zero, then no adjustment of the vector direction is necessary (unconstrained condition) and Subroutine LINCON returns control to HYPFIL. Proceed if NLIN is 1 or 2.

8. Begin Loop 2 - Loop 2 is used to adjust either the gradient (partial derivative) or acceleration (parallel tangent) vector direction.

If the partial derivative of a variable is zero ($FP(I) = 0$) ($KNT(I) > 0$), then the partial derivatives of all (1 or 2) the constraints are set to zero by:

$$FPC(I, JT) = 0,$$

where JT is 1 through NLIN.

9. Equation Solving

- a. The FP array is $[P]$, the column vector representing the gradient, as described by Cross.⁽⁷⁾ Array FPC is $[G]$, the matrix of partial derivatives of the constraints. The $[GP]$ vector is stored in BSUM by:

$$BSUM(J) = BSUM(J) + (FPC(I, JT) * FP(I)).$$

- b. The sign is changed to give $[-GP]$ by:

$$BSUM(I) = -BSUM(I).$$

- c. As in Step 6.c (Page 44), the AX or $[GG^T]$ matrix is generated by:

$$AX(J, L) = AX(J, L) + (FPC(I, LT) * FPC(I, JT)).$$

- d. If any of the principal diagonal elements of $[GG^T]$ are zero, they are set to 1 to avoid singularity problems by:

$$AX(I, I) = 1.$$

- e. The Lagrange multipliers are the elements of a column vector $[\lambda]$:

$$\begin{aligned} [\lambda] &= [GG^T]^{-1} [GP] \text{ or} \\ [-\lambda] &= [GG^T]^{-1} [-GP]. \end{aligned}$$

If there is only one violation ($NLIN = 1$), then the equation is solved directly by:

$$BSUM(1) = BSUM(1)/AX(1, 1).$$

If NLIN is greater than 1, the linear simultaneous equation solving routine AMATQ4 (see Subsection 4.4.1.3, Page 47) is used. The solution vector $[-\lambda]$ is now in BSUM.

- f. Using this vector, the new direction, $[PP]$, can finally be completed by adjusting $[P]$:

$$\begin{aligned} [PP] &= [P] - [G^T \lambda] \text{ or} \\ [PP] &= [P] + [G^T] [-\lambda]. \end{aligned}$$

$[PP]$, the adjusted FP, is calculated by:

$$FP(I) = FP(I) + (BSUM(J) * FPC(I, JT)).$$

10. Continue Loop 2.

- a. If any variable, $X(I)$, is on the lower boundary, $XL(I)$, or within 10^{-9} of the upper boundary, $XH(I)$, and the direction, $FP(I)$, is such as to drive the variable out, then proceed to the next step, Step (b). Otherwise, transfer control back to HYPFIL.
- b. The direction components for the one or more variables in danger of moving outside the boundaries are set to zero by:

$$FP(I) = 0.$$

- c. The (endangered) variable which has the direction component, FP , with the largest magnitude is held on the boundary for five gradient vector direction changes. This action is accomplished by:

$$KNT(ISET) = KNT(ISET) + 1.$$

- d. End of Loop 2. Revert to Item 4.4.1.2.8 (Page 46) for another pass through Loop 2. LINCON will eventually return control from Step a to HYPFIL after FP and FPC are set to zero and KNT is increased for a sufficient number of variables.

4.4.1.3 Subroutine AMATQ4 - AMATQ4 is a routine written by Funderlic and Edwards to solve simultaneous linear equations.⁽⁹⁾ The authors of HYPFIL converted it to a REAL * 8 precision subroutine and removed the calculation of the determinant.

4.4.1.4 Subroutine EVAL - Subroutine EVAL, which is called by either HYPFIL or Subroutine LINCON, initializes and translates variables and calls Subroutine TUBE for an evaluation of physical conditions and economic parameters.

1. The three constraint variables (see Subsection 4.4.1.2, Page 39) are initialized by:

$$\begin{aligned} CPMAX &= 0, \\ REMIN &= 0, \text{ and} \\ CPEND &= 0. \end{aligned}$$

2. The list flag (0 = searching, 1 = optimum found) is transferred by:

$$\text{IOUT} = \text{IOU}.$$

3. The inlet pressure, inlet velocity, concentration ratio, taper, and tube diameter are initialized using the corresponding X variable by:

$$\begin{aligned}\text{POF} &= \text{X}(1), \\ \text{VIN} &= \text{X}(2), \\ \text{CR} &= \text{X}(3), \\ \text{TAPER} &= \text{X}(4), \text{ and} \\ \text{D} &= \text{X}(5).\end{aligned}$$

4. Subroutine EVAL calls Subroutine TUBE (see Subsection 4.4.2.1, Page 49) which, in turn, calls Subroutine ECON (see Subsection 4.4.3, Page 69) for an evaluation of the physical conditions and economics. It has been observed that, for 50 tube increments, the program spends over 99 percent of its time in Subroutines EVAL, TUBE, and ECON; mostly in TUBE.
5. The total water cost just calculated (blended or unblended basis, as desired) is transferred to the subroutine argument by:

$$\text{F} = \text{COST}.$$

6. The error flag value is transferred to argument IBJ by:

$$\text{IBJ} = \text{NFLAG}.$$

7. The values of the newly calculated inequality constraint variables are transferred to the variables used by LINCON by:

$$\begin{aligned}\text{CPM} &= \text{CPMAX}, \\ \text{REM} &= \text{REMIN}, \text{ and} \\ \text{CPE} &= \text{CPEND}.\end{aligned}$$

8. Return control back to either HYPFIL or Subroutine LINCON. End of Subroutine EVAL.

4.4.1.5 Function ICLOCK - ICLOCK is a function that is used by HYPFIL to get the time by:

$$\text{ITIME1} = \text{ICLOCK}(0).$$

The standard Oak Ridge National Laboratory (ORNL) version returns the time as an integer in the form of 0.01 second. If the time is not available, the program will

run correctly using the dummy function ICLOCK, shown in List B-1, Appendix B, which returns a value of zero. The only capability lost when using the dummy routine is the ability to terminate each case after a specified length of time through the use of Namelist variable, MAXTIM.

4.4.1.6 Subroutine IDAY - IDAY is a subroutine called by HYPFIL to get the date by:

CALL IDAY(DATE).

The ORNL version returns an eight-character date which is printed at the top of the lists. The dummy version of IDAY, shown in List B-1, supplies eight blanks. The dummy version of IDAY is actually combined with Function ICLOCK to make one deck by using ENTRY IDAY(DATE). A call made to an entrypoint in a function will work correctly when using IBM 360 FORTRAN H.

4.4.2 Material Balance Calculations

4.4.2.1 Subroutine TUBE - This routine solves the system of equations that simulate the hydrodynamic flow conditions. An alphabetical listing and description of all FORTRAN variables is found in Subsection 4.5.2 (Page 75).

1. Initialization for Loop 1 - This section consists primarily of initialization of conditions at the upstream end of the first tube increment. The main program, HYPFIL, optimizes up to five independent variables: POF (inlet gage pressure, atm), VIN (inlet axial velocity, ft/sec), CR (concentration ratio, reject conc/feed conc), TAPER (see Section 4.1), and D (inside tube diameter, in). It must be remembered, however, that these parameters are fixed for each call to Subroutine TUBE.
 - a. The error indicator, NFLAG, is set to 1 before any error checks are made. The indicator is changed if any error is detected during the calculation in Subroutine TUBE. The types of errors detected can be found by examining the program listing and are described in Subsection 5.3.4 (Page 91).
 - b. Set C(1)(wt fraction), the concentration of the turbulent stream at the upstream end of the first tube increment, equal to the feed concentration by:

$$C(1) = CFEED.$$

- c. Loop 1 uses N equal concentration increments from x_f to x_{rej} (where $x_f = CFEED$ and $x_{rej} = CFEED * CR$). Calculate the concentration increment, DC (wt fraction), by:

$$DC = C(1) * (CR - 1)/N,$$

where CR is the concentration ratio (reject conc/feed conc).

- d. Set the upstream gage pressure, PONE (atm), equal to the inlet pressure, POF, for the first increment by:

$$PONE = POF.$$

- e. Set the upstream axial velocity, VELONE (ft/sec), equal to the inlet velocity, VIN, for the first tube increment by:

$$VELONE = VIN.$$

- f. Set the salt rejection of the membrane at this point, R (fraction), equal to REJECT, an initial guess for the tube entrance, by:

$$R = REJECT.$$

- g. The fraction of salt transmitted, FRTRAN (fraction), is then given by:

$$FRTRAN = 1 - R.$$

- h. An initial guess if made for x_{α} , CWONE (wt fraction), the salt concentration at the membrane entrance interface, at the tube entrance by:

$$CWONE = CFEED/(EONE * R + FRTRAN),$$

where EONE, the mixing efficiency, is given an initial value of 0.99.

- i. The corresponding x_{ω} , CPONE (wt fraction), the salt concentration at the membrane/product interface, is given by:

$$CPONE = FRTRAN * CWONE.$$

- j. For round tubes, the cross-sectional area, AX (ft²), is calculated by:

$$AX = \pi * DFT ** 2/4,$$

and the tube perimeter, DFTPI (ft), is calculated by:

$$DFTPI = DFT * \pi,$$

where DFT is the inside "equivalent" tube diameter (ft), and equals D/12, in which D is the inside "equivalent" tube diameter in inches.

For rectangular tubes, the cross-sectional area is given by:

$$AX = RA * A * A,$$

where $A = DFT * (RA + 1)/(2 * RA)$,

in which DFT (ft) is four hydraulic radii, RA is the aspect ratio (membrane wall/nonpermeable wall), and A (ft) is the length of the nonpermeable side.

The perimeter with a useful membrane surface for a rectangular tube, DFTPI (ft), is given by:

$$DFTPI = 2 * RA * A.$$

- k. An initial value for the length, DL (ft), of the first tube increment is approximated by:

$$DL(1) = 10.358 * DC * VIN * AX * RHOF(CFEED)/PERM/DFTPI/CFEED/(POF-BETAF(CFEED)),$$

where the factor is $10.358 \text{ kgal sec lb}^{-1} \text{ day}^{-1}$ (based on 62.4 lbs/ft^3), RHOF (CFEED) (lbs/ft^3) is the density of the feed stream, PERM ($\text{kgal ft}^{-2} \text{ atm}^{-1} \text{ day}^{-1}$) is the membrane permeability, and BETAF (CFEED) is the osmotic pressure at the membrane entrance interface at the tube entrance.

- l. Set CPROD(1) (wt fraction), the cumulative product concentration at the tube entrance; and CP(N+1) (wt fraction), the product interface salt concentration at the end of the tube (wt fraction), equal to zero by:

$$\begin{aligned} \text{CPROD}(1) &= 0 \text{ and} \\ \text{CP}(N+1) &= 0. \end{aligned}$$

- m. Initialize CPEND, the largest product salt concentration for any increment, equal to zero by:

$$\text{CPEND} = 0.$$

- n. Initialize the cumulative length of the tube, L (ft), and the effective length, EFFLEN (ft), at the beginning of the tube by:

$$\begin{aligned} L(1) &= 0 \text{ and} \\ \text{EFFLEN} &= 0. \end{aligned}$$

- o. Set the cumulative velocity ratio, VRCUM, to its initial value of 1 by:

$$\text{VRCUM} = 1.$$

- p. Initialize the cumulative total salt flow rate in the product, S (lbs salt/sec), and the cumulative total water flow rate in the product water, W (lbs water/sec) by:

$$S = 0 \text{ and}$$

$$W(1) = 0.$$

- q. V ($\text{ft sec}^{-1} \text{ atm}^{-1}$), the product of several constant terms used to calculate ZZ (see Equation 12, Page 18), is calculated by:

$$V = -0.038683 * \text{PERM} * \text{XRATIO},$$

where the constant is $-0.038683 (\text{day ft}^3 \text{ kgal}^{-1} \text{ sec}^{-1})$ and XRATIO is used to modify the exponent in Equation 12 when turbulence promoters are to be considered. (This V is not the inlet velocity, V , used in the main program, HYPFIL .)

2. Begin Loop 1 - 1 Loop for N concentration increments linear from x_f to x_{re} . Initialize variables for Loop 2, upstream end of the $(i-1)$ th tube increment.

- a. Calculate brine concentration x_i (wt fraction) at the downstream end of $(i-1)$ th increment by:

$$C(i) = C(i-1) + DC.$$

The salt concentration, $C(1)$, $C(2)$, ... $C(N+1)$, are at the endpoints of N equal-concentration-change increments into which the tube is divided.

- b. When the taper is zero, there is no change in velocity between the tube increments. The conditions at the upstream end of the $(i-1)$ th increment are the same as the conditions at the downstream end of the $(i-2)$ th increment. Therefore, if the taper is zero and i is greater than 2, skip Loop 2 and go to Item 4, Page 57.
- c. Store concentration x_i at the upstream end of the $(i-1)$ th increment; CONE (wt fraction) as a nonsubscripted variable by:

$$\text{CONE} = C(i-1).$$

- d. Calculate the osmotic pressure differential across the membrane, $\pi_\alpha - \pi_\omega$ (atm), by:

$$\text{BETA} = \text{BETAF} (\text{CWONE}) - \text{BETAF} (\text{CPONE}),$$

where the statement function, BETAF, calculates the osmotic pressure for CWONE (x_w , salt concentration at the membrane entrance interface) and for CPONE (x_w , salt concentration at the product interface).

- e. Calculate effective pressure, P_{eff} , at this point by:

$$PWONE = PONE - BETA,$$

where PONE (P_Q) is the gage pressure (atm) at the entrance interface of the membrane at that point. Gage pressure, P_w , is assumed to be zero.

- f. Calculate the fluid density, ρ_t , at the upstream end of the (i-1)th increment, RHOONE (lb/ft³), by:

$$RHOONE = RHOF (CONE),$$

where RHOF is a density function statement.

- g. Calculate the viscosity at this point, MUONE (lb sec⁻¹ ft⁻¹), by:

$$MUONE = MUF (CONE),$$

where MUF is a viscosity function statement.

- h. Calculate the bulk velocity at the upstream end of the (i-1)th increment, BONE (lb/sec), by:

$$BONE = VELONE * RHOONE * AX,$$

where VELONE is the axial velocity (ft/sec) and AX the cross-sectional tube area (ft²).

- i. Calculate the mass velocity at this point, GONE (lb ft⁻² sec⁻¹), by:

$$GONE = VELONE * RHOONE.$$

- j. Calculate the Reynolds number at the upstream end of the (i-1)th increment by:

$$REONE = DFT * GONE/MUONE.$$

- k. Calculate $(N_{Re})^{1/4}$ as RE25 by:

$$RE25 = REONE ** 0.25.$$

- l. If the Reynolds number exponent in Equation 10b (Page 17) is to be other than 0.25, as has been used for turbulence promoter calculations, calculate it by:

$$REXP = REONE ** XP,$$

where XP is the desired exponent.

- m. Calculate the Fanning friction factor, FONE, at the upstream end of the (i-1)th section by using the Blasius equation:

$$FONE = FF/RE25,$$

where $FF = 0.08 * FRATIO$,

and FRATIO is a "roughness" factor which may be supplied as input data.

3. Begin Loop 2 - Loop 2, within Loop 1, calculates x_α and x_ω for the upstream end of the (i-1)th increment by iteration.

- a. Let ITIMES be an iteration counter.

- b. Let $F(x_\alpha, x_\omega) = \exp(-25b(v/u) N_{Re}^\alpha N_{Sc}^{2/3}) - (x_t - x_\omega)/(x_\alpha - x_\omega)$ by modifying Equation 12 where $\alpha = XP = 1/4$ and $b = XRATIO = 1$ unless turbulence promoters are used. For this case, α and b are empirically fitted to the turbulence promoter data (see Thomas, et al).⁽¹⁰⁾ Expressing radial velocity, v , in terms of P_{eff} and P , calculate F as:

$$RES1 = EXPZZZ - EONE,$$

where: $EXPZZZ = \exp(-0.038683 * XRATIO * PERM * PWONE * REXP * SC667(CONE)/VELONE)$,

and $EONE = (CONE - CPONE)/(CWONE - CPONE)$.

$$c. \text{ Let } G(x_\alpha, x_\omega) = \left(\frac{1}{R} - 1\right) - \left(\frac{e^{\beta\sigma}}{e^{\beta\sigma} - 1}\right) \left(\frac{\beta D^*}{1 - \beta D^*}\right),$$

by arranging Equation 13 (Page 18). Expressing $\beta\sigma$ ($\beta\sigma \neq 0$), as in Equation 13, by $BETSIG = CONST * PWONE * PERM$, G is calculated by:

$$RES2 = BDSTAR/(1 - BDSTAR) * EXPFBS - FRTRAN/R,$$

where: $EXPFBS = EXPBS/(EXPBS - 1)$, $EXPBS = \exp(\beta\sigma)$, and $FRTRAN = 1 - R$.

- d. For univalent ion exchange membranes, for βD^* :

$$BDSTAR = 0.5 * (-CSTAR + (CSTAR^2 + 4)^{1/2}),$$

where: $CSTAR = CSTAR/CCW$.

- e. For membranes where a polynomial fit is used to relate the distribution coefficient to concentration:

$$BDSTAR = ((D3 * CWONE + C3) * CWONE + B3) * CWONE + A3.$$

- f. By the Newton-Raphson method, solve for a new estimate of x_α and x_ω to satisfy functions F and G using Gaussian elimination to solve the simultaneous equations:

$$\begin{aligned} C11 * \Delta x_\alpha + C12 * \Delta x_\omega &= RES1 \text{ and} \\ C21 * \Delta x_\alpha + C22 * \Delta x_\omega &= RES2, \end{aligned}$$

where, in terms of partial derivatives:

$$(1) \quad C11 = \frac{\partial F}{\partial x_\alpha}, \text{ and is calculated by:}$$

$$C11 = DEDCW + DZDPF * DPFDCW,$$

where:

DEDCW = EONE/CWMCP, in which
 CWMCP = CWONE - CPONE, and where
 DZDPF = - ZZZ * EXP (ZZZ), in which
 ZZZ = ZZ * PWONE, in which
 ZZ = V * REXP/VELONE * SC667(CONE), and where
 DPFDCW = - DBETA (CWONE).

$$(2) \quad C12 = \frac{\partial F}{\partial x_\omega}, \text{ and is calculated by:}$$

$$C12 = DEDCP + DZDPF * DPFDCP,$$

where:

DEDCP = (CONE - CWONE)/CWMCP2, in which
 CWMCP2 = CWMCP * CWMCP, and where
 DPFDCP = DBETA (CPONE).

$$(3) \quad C21 = \frac{\partial G}{\partial x_\alpha}, \text{ and is calculated by:}$$

$$C21 = DFRDCW + DYDBD * DBDDCW + DYDBS * DBSDPE * DPFDCW,$$

where:

$$\begin{aligned} DFRDCW &= -CPONE/CWMCP2, \text{ and where} \\ DYDBD &= -EXPFB/(1 - BDSTAR)^2, \text{ in which} \\ EXPFB &= EXPBS/(EXPBS - 1), \text{ in which} \\ EXPBS &= \exp(BETSIG), \text{ and where} \\ DBDDCW &= BDSTAR * A3 * DCWCM/(CCW * (2 * CCW * BDSTAR + A3)), \text{ in which} \\ DCWCM &= CMWF/(1 - CWONE)/(1 - CWONE), \text{ in which} \\ CMWF &= 1000/58.45, \text{ and} \\ CCW &= CMOLAL(CWONE), \text{ and where} \\ DYDBS &= EXPFB/(1 - BDSTAR) * BDSTAR * BETSIG/(EXPBS - 1), \text{ and} \\ \text{where} \\ DBSDPE &= BETSIG/PWONE. \end{aligned}$$

(4) $C22 = \frac{\partial G}{\partial x_\omega}$, and is calculated by:

$$C22 = DFRDCP + DYDBS * DBSDPE * DPFDCP.$$

- g. New estimates for x_α and x_ω are obtained from the solutions to these equations from:

$$x_\alpha = x_\alpha + \Delta x_\alpha \text{ and}$$

$$x_\omega = x_\omega + \Delta x_\omega \text{ by the program statements:}$$

$$\begin{aligned} CWONE &= CWONE + DCW \text{ and} \\ CPONE &= CPONE + DCP. \end{aligned}$$

- h. Calculate the corresponding mixing efficiency, EONE, by:

$$EONE = (CONE - CPONE)/(CWONE - CPONE).$$

- i. Calculate the corresponding membrane rejection, RONE, by:

$$RONE = R,$$

where:

$$\begin{aligned} FRTRAN &= CPONE/CWONE \text{ and} \\ R &= 1 - FRTRAN. \end{aligned}$$

- j. Calculate the osmotic pressure differential across the membrane, $\pi_\alpha - \pi_\omega$ (atm), by:

$$\text{BET A} = \text{BETAF}(\text{CWONE}) - \text{BETAF}(\text{CPONE}).$$

- k. Calculate the effective pressure, P_{eff} (atm), at this point by:

$$\text{PWONE} = \text{PONE} - \text{BETA}.$$

- l. End of Loop 2. If a sufficiently good solution is obtained by current estimates of x_{α} (CWONE) and x_{ω} (CPONE) to satisfy functions G and F so that:

$$\text{ABS}(\text{RES1}) < 10^{-6} \text{ and } \text{ABS}(\text{RES2}) < 10^{-6} * \text{FRTRAN},$$

then transfer to the next item, Item 4. If not, then transfer back to Item 3 (Page 54) and repeat Loop 2 until the convergence criteria are met.

4. Initialization for Loop 3 and Loop 4 - Loop 2 is completed, giving x_{α} and x_{ω} for the upstream end of the (i-1)th increment.

- a. Initialize the following variables for Loop 3 and Loop 4:

CTWO = C(I),
 CWTWO = CWONE,
 CPTWO = CPONE,
 CPBAR(I-1) = CPONE,
 PWBAR(I-1) = PWONE,
 ETWO = EONE,
 RHOTWO = RHOF(CTWO), and
 MUTWO = MUF(CTWO).

For example, CTWO (x_i , wt fraction) is the unsubscripted variable for the downstream end of the (i-1)th increment which corresponds to CONE at the upstream end of the (i-1)th increment. Unsubscripted variables are used extensively in Loops 2 and 4 to save computing time.

- b. The following variables are used to check for convergence of P_{α} (pressure at entrance membrane interface), N_{Re} , and increment length in Loop 3:

PLAST = PONE,
 RLAST = REONE, and
 DLAST = DL(I-1).

- c. NTIMES is the iteration counter for Loop 3.

5. Begin Loop 3 - Loop 3, within Loop 1, is used for iteration of the following variables at the downstream end of the (i-1)th increment: P_{α} (pressure at the membrane entrance interface), N_{Re} (Reynolds number), length of the increment, x_{α} , and x_{ω} .

- a. Calculate the bulk velocity at the downstream end of the (i-1)th increment BTWO (lbs/sec) by:

$$BTWO = BONE - DB(I-1),$$

where DB(I-1) is estimated by:

$$DB(I-1) = 0.09652 * DFTPI * PERM * PWBAR(I-1) * DL(I-1) / (1 - CPBAR(I-1)).$$

- b. The corresponding downstream mass velocity, GTWO (lb ft⁻² sec⁻¹), is then GTWO = BTWO/AX.

- c. Calculate the salt passing through the membrane in the (i-1)th increment, DS (lb salt/sec), by:

$$DS = CPBAR(I-1) * DB(I-1).$$

- d. Estimate the velocity change in the (i-1)th increment, DV(I-1) (ft/sec), by:

$$DV(I-1) = VELTWO - VELONE,$$

where:

$$VELTWO = GTWO / RHOTWO.$$

- e. The following downstream variables are initialized for Loop 4 in the same manner as they were for Loop 2 (see Item 3, Page 54):

N_{Re} (RETWO), $(N_{Re})^{1/4}$ (RE25), Reynolds number to the exponent XP (REXP), and Fanning friction factor (FTWO).

- f. The average velocity, VELBAR (ft/sec); average Fanning friction factor, FBAR; and the average density, RHOBAR (lb/ft³), for the (i-1)th increment are estimated by:

$$\begin{aligned} VELBAR &= 0.5 * (VELONE + VELTWO), \\ FBAR &= 0.5 * (FONE + FTWO), \text{ and} \\ RHOBAR &= 0.5 * (RHOONE + RHOTWO). \end{aligned}$$

- g. Calculate the pressure drop in the (i-1)th increment, DP(I-1)(atm), by:

$$DP(I-1) = RHOBAR * (-VELBAR * DV(I-1) + 2 * FBAR * DL(I-1) * VELBAR * VELBAR / DFT) / GC / 2116.8,$$

where:

$$GC = 32.2 \text{ lb}_m \text{ ft lb}_f^{-1} \text{ sec}^{-2} \text{ and the factor is } 2116.8 \text{ lb ft}^{-2} \text{ atm}^{-1}.$$

- h. The downstream pressure at the membrane entrance interface, P_α (atm), is calculated as:

$$PTWO = PONE - DP(I-1).$$

- i. Calculate effective pressure, P_{eff} (atm), at this point by:

$$PWTWO = PTWO - BETA.$$

6. Begin Loop 4 - Loop 4, within Loop 3 (within Loop 1), is used to calculate x_α and x_ω for the downstream end of the (i-1)th increment in exactly the same manner as Loop 2 is used to calculate x_α and x_ω for the upstream end of the increment. (See Loop 2, Item 3, Page 54, for details.)

To save computing time, the convergence criteria in Loop 4 are less stringent than in Loop 2. This is because the convergence is checked again at the end of Loop 3. Transfer out of Loop 4 to Item 7, when the following criteria are met:

$$ABS(RES1) < 10^{-3} \text{ and } ABS(RES2) < 0.5 * 10^{-3}.$$

End of Loop 4. Repeat this item until convergence.

7. Continuation of Loop 3 (within Loop 1) - Loop 4 (within Loop 3, within Loop 1) has been completed.

- a. Recalculate the average product concentration, x_ω , for the (i-1)th increment, $CPBAR(I-1)$ (wt fraction), by:

$$CPBAR(I-1) = 0.5 * (CPONE + CPTWO),$$

where CPTWO is the downstream x_ω calculated in Loop 4.

- b. Recalculate the average effective pressure in the (i-1)th increment, $PWBAR(I-1)$ (atm), by:

$$PWBAR(I-1) = 0.5 * (PWONE + PWTWO),$$

where PWTWO is the downstream effective pressure calculated in Loop 4.

- c. Calculate $DL(I-1)(ft)$, the length of the $(i-1)$ th increment, by:

$$DL(I-1) = 10.358 * DC * BONE / (C(I) - CPBAR(I-1)) / PERM / PWBAR(I-1) / DFTPI,$$

where the factor is $10.358 \text{ kgal sec lb}^{-1} \text{ day}^{-1}$ (based on 62.4 lbs/ft^3).

- d. Check the following five variables for convergence in Loop 3:

- (1) Check x_α for

$$ABS(RES1) < 10^{-6},$$

- (2) Check x_ω for

$$ABS(RES2) < 10^{-6} * FRTRAN,$$

- (3) Check the downstream membrane entrance interface pressure, P_α , for

$$ABS ((PTWO - PLAST)/DP(I-1)) < 0.01,$$

- (4) Check the Reynolds number, N_{Re} , for

$$ABS(RETWO - RLAST)/RETWO < 0.01, \text{ and}$$

- (5) Check the length of the $(i-1)$ th increment for

$$ABS ((DL(I-1) - DLAST)/DL(I-1)) < 10^{-4}.$$

If all five of these variables have converged, then transfer out of Loop 3 to the next item, Item 8.

- e. If convergence for Loop 3 is not achieved, then the latest estimates are saved by:

$$\begin{aligned} PLAST &= PTWO, \\ RLAST &= RETWO, \text{ and} \\ DLAST &= DL(I-1); \end{aligned}$$

and a transfer back to Item 5, Page 57, is made to repeat Loop 3. End of section for Loop 3.

8. Continuation of Loop - Loop 3 (within Loop 1) has been completed.

- a. Recalculate, using the latest values found in Loop 3, the change in bulk velocity, $DB(I-1)$; the bulk velocity, $BTWO$ (lb/sec); and the mass velocity,

GTWO ($\text{lb ft}^{-2} \text{sec}^{-1}$), for the downstream end of the (i-1)th increment in exactly the same manner as in Item 5. The velocity and other related variables are not recalculated because their most recent values are close enough.

- b. Calculate the product rate for the (i-1)th increment, RATE(I-1) ($\text{gal day}^{-1} \text{ft}^{-2}$), by:

$$\text{RATE}(I-1) = \text{DB}(I-1) * 10358 / \text{DL}(I-1) / \text{DFTPI},$$

where the factor is $10358 \text{ gals sec lb}^{-1} \text{ day}^{-1}$.

- c. Calculate the velocity ratio for the (i-1)th increment by:

$$\text{VRATIO} = (\text{VELONE} - \text{VELTWO}) * \text{TAPER} / \text{VELTWO} + 1.$$

For TAPER = 0 (decreasing velocity, constant total cross-sectional area), the velocity ratio is 1. For TAPER $\neq$ 0 (changing total cross-sectional area), the velocity ratio is VELONE/VELTWO.

- d. Calculate the "effective length", EFFLEN (ft), by:

$$\text{EFFLEN} = \text{EFFLEN} * \text{VRATIO} + \text{DL}(I-1).$$

- e. Calculate the salt flow rate in the product in the (i-1)th increment, DS (lb salt/sec), by:

$$\text{DS} = \text{CPBAR}(I-1) * \text{DB}(I-1) / \text{VRCUM}.$$

- f. The cumulative salt flow in all i-1 increments, S (lb salt/sec), is then given by:

$$S = S + \text{DS}.$$

- g. The water produced in the (i-1)th increment, DW (lb water/sec), is calculated by:

$$\text{DW} = \text{DB}(I-1) / \text{VRCUM} - \text{DS}.$$

- h. The total amount of water produced by i increments is calculated by:

$$\text{W}(I) = \text{W}(I-1) + \text{DW}.$$

- i. The average salt concentration is the product from i tube increments, CPROD(I)(wt fraction), and is calculated by:

$$CPROD(I) = S/(S + W(I)).$$

- j. Calculate $L(I)$ (ft), the length of i tube increments, by:

$$L(I) = L(I-1) + DL(I-1).$$

- k. The cumulative velocity ratio (VRCUM) is multiplied by the velocity ratio (VRATIO) for this tube increment by:

$$VRCUM = VRCUM * VRATIO.$$

- l. The final values for the upstream end of the $(i-1)$ th increment are saved by storing the unsubscripted variables explained previously in the corresponding subscripted variable arrays by:

$CW(I-1) = CWONE,$
 $CP(I-1) = CPONE,$
 $P(I-1) = PONE,$
 $PW(I-1) = PWONE,$
 $VEL(I-1) = VELONE,$
 $REJ(I-1) = RONE,$
 $E(I-1) = EONE,$
 $RE(I-1) = REONE,$
 $RHO(I-1) = RHOONE,$
 $MU(I-1) = MUONE,$
 $B(I-1) = BONE,$ and
 $G(I-1) = GONE.$

- m. Save the lowest Reynolds number and the largest product salt concentration for any increment for use in the second and third inequality constraints in subroutine LINCON:

$REMIN = \min(REMIN, RETWO)$ and
 $CPEND = \max(CPEND, CPTWO).$

- n. Initialize $VELONE$, the upstream axial velocity, for the next increment (next pass through Loop 1) by:

$$VELONE = VELTWO * VRATIO.$$

- o. Initialize the length of the next increment, $DL(I)$ (ft), by the estimate:

$$DL(I) = DL(I-1),$$

when $i = 2$. For $i > 2$, use differences for a second-order estimate of the length by:

$$DL(I) = DL(I-1) + (DL(I-1) - DL(I-2)).$$

- p. Estimate the mixing efficiency, E_M , for the next (ith) increment by:

$$EONE = E(I-1),$$

when $i = 2$. When $i > 2$, differences are used for a second-order estimate by:

$$EONE = E(I-1) = (E(I-1) - E(I-2)).$$

- q. Initialize PONE, CWONE, and CPONE for the upstream end of the next increment by:

$$\begin{aligned} PONE &= PTWO, \\ CWONE &= CWTWO, \text{ and} \\ CPONE &= CPTWO. \end{aligned}$$

- r. When the taper is not zero, a step change in velocity occurs between tube increments. If such is the case, transfer back to Item 2 (Page 52) and repeat Loop 1 until all N tube increments have been processed; then move to Item 9 (Page 64).

- s. If the taper is zero, the conditions at the upstream end of the ith increment will be the same as the conditions at the downstream end of the (i-1)th increment. Variables PONE, CWONE, and CPONE have been set; EONE must be changed. The following variables for the upstream end of the next tube increment are set by:

$$\begin{aligned} CONE &= CTWO, \\ PWONE &= PWTWO, \\ RHOONE &= RHOTWO, \\ MUONE &= MUTWO, \\ REONE &= RETWO, \\ EONE &= ETWO, \\ RONE &= RTWO, \\ BONE &= BTWO, \\ GONE &= GTWO, \text{ and} \\ FONE &= FTWO. \end{aligned}$$

- t. End of Loop 1. Go back to Item 2 (Page 52) until all N tube increments have been processed, then proceed to Item 9.

9. Calculations following Loop 1 - Loop 1 has been completed for all N tube increments.

- a. Store the values for the downstream end of the nth (last) tube increment in the (n+1)th location of the following variable arrays by:

$CP(N+1) = CPTWO,$
 $CW(N+1) = CWTWO,$
 $P(N+1) = PTWO,$
 $PW(N+1) = PWTWO,$
 $VEL(N+1) = VELTWO,$
 $RE(N+1) = RETWO,$
 $E(N+1) = ETWO,$
 $REJ(N+1) = RTWO,$
 $RHO(N+1) = RHOTWO,$
 $MU(N+1) = MUTWO,$
 $B(N+1) = BTWO,$ and
 $G(N+1) = GTWO.$

- b. The last velocity ratio term (VRATIO) is removed from the cumulative velocity ratio (VRCUM) since the plant does not have to be retapered after the last tube increment by:

$VRCUM = VRCUM/VRATIO.$

- c. Set CPMAX (wt fraction) equal to the final product salt concentration for use in the first inequality constraint in Subroutine LINCON by:

$CPMAX = CPROD(N+1).$

- d. Check the material balance and print a warning note if the criterion $ABS((C(N+1) - EXITS)/C(N+1)) < 0.01$ is not satisfied, where (wt fraction):

$EXITS = (B(1) * CFEED - S) * VRCUM/B(N+1).$

- e. Call Subroutine ECON (see Subsection 4.4.3, Page 69) which calculates the economic results.

- f. Return with the calculated water cost through Subroutine EVAL to the calling program, HYPFIL or Subroutine LINCON.

4.4.2.2 Entry SETUP in Subroutine TUBE - This section of Subroutine TUBE is called only once per optimization run to initialize the necessary variable storage arrays and to set certain constants and coefficients.

1. Set all of the 23 subscripted variable storage arrays to zero.
2. The constants π (PI) and g_c (GC , $\text{lb}_m \text{ ft lb}_f^{-1} \text{ sec}^{-2}$) are set by:

$$\text{PI} = 3.14159, \text{ and}$$

$$\text{GC} = 32.2.$$
3. Set CMWF (gmole NaCl/kg NaCl) for use in converting weight fraction salt to moles salt/kg solvent by:

$$\text{CMWF} = 1000/58.45.$$
4. FF, the constant for the Blasius equation, is set by:

$$\text{FF} = 0.08 * \text{FRATIO},$$

where FRATIO is a "roughness factor" which may be supplied as input data.
5. For round tubes, the fraction of the tube perimeter that consists of the useful membrane, FW, is given by:

$$\text{FW} = 1.$$

For rectangular tubes, it is calculated by:

$$\text{FW} = \text{RA}/(\text{RA} + 1),$$

where RA is the aspect ratio (membrane side/nonpermeable side).
6. If the Reynolds number exponent used to calculate the Fanning friction factor, XP, is 0.25, then set JTYPE = 2; otherwise set JTYPE = 1.
7. Set the solubility limit, SOLIM (wt fraction), equal to the concentration at which NaCl crystallizes by:

$$\text{SOLIM} = 0.26.$$
8. The temperatures T (F) and TEMP (F) are equivalent. Calculate several temperature-related terms by:

$$\text{T2} = \text{T} * \text{T},$$

$$\text{LOG10T} = \log_{10} \text{T}, \text{ and}$$

$$\text{LOGT2} = \text{LOG10T} ** 2.$$

9. Calculate AA and BB, the temperature-dependent coefficients used in arithmetic statement function SC667, a fit for Schmidt number to the 0.667 power. See Item 4.4.2.3.6 (Page 68) for the coefficients.
10. Calculate A1, B1, and C1; the temperature-dependent coefficients used in arithmetic statement function RHOF (lb/ft^3), the density function. See Item 4.4.2.3.2 (Page 67) for the coefficients.
11. Set TWOC1 for use in a partial derivative in TUBE by:

$$\text{TWOC1} = 2 * \text{C1}.$$

12. DEN77 (lb/ft^3) is set equal to the value of the density at 77° F (25° C) and zero NaCl concentration by:

$$\text{DEN77} = 62.18.$$

13. Calculate A2, B2, and C2; the temperature-dependent coefficients used in arithmetic statement function MUF ($\text{lb sec}^{-1} \text{ft}^{-1}$), the viscosity function. See Item 4.4.2.3.3 (Page 67) for the coefficients.
14. Set CTAB (atm kg solvent/gmole NaCl) for use in arithmetic statement function BETAF (atm), the osmotic pressure fit for low NaCl concentrations, by:

$$\text{CTAB} = 48.8 * ((\text{TEMP} + 460)/537) * (\text{RHOF}(0)/\text{DEN77}).$$

The constant $2 * 24.4$ (at 25° C or 77° F) is thus adjusted for absolute temperature and molal volume.

4.4.2.3 Calculation of Physical Properties in Subroutine TUBE - Outlined in this subsection are the relationships used to calculate the physical properties of aqueous sodium chloride solutions. All of the equations appear as arithmetic function statements at the beginning of Subroutine TUBE. All equations are a function of Z, weight fraction NaCl. The temperature-dependent terms need be initialized only once since temperature is a constant for each optimization case.

1. Conversion of Weight Fraction (g NaCl/g solution) to Molality (gmole NaCl/kg solvent H_2O)

Calculate the molality by:

$$\text{CMOLAL}(Z) = \text{CMWF} * Z/(1-Z),$$

where $\text{CMWF} = 1000/58.45$ gmole NaCl/kg solvent and $Z = \text{wt fraction NaCl}$.

2. Density of Salt Solution

A density fit (lb/ft^3) for aqueous sodium chloride solutions is given by:

$$\text{RHO}(Z) = (C1 * Z + B1) * Z + A1,$$

where:

Z = weight fraction NaCl,

$$A1 = 62.717753 - 0.32152986 * 10^{-2} * T - 0.48932777 * 10^{-4} * T^2,$$

$$B1 = 44.315006 - 0.11647394 * 10^{-1} * T + 0.34231326 * 10^{-5} * T^2,$$

$$C1 = 16.449945 + 0.13761984 * 10^{-1} * T - 0.92275301 * 10^{-4} * T^2, \text{ and}$$

T = temperature, $^{\circ}\text{F}$.

In comparison with reference density values,⁽¹¹⁾ the largest absolute error for 36 points selected over a temperature range of 32 to 300 $^{\circ}\text{F}$ and for concentrations up to 20 weight percent NaCl was less than 0.4 percent.

3. Viscosity of Salt Solution

A viscosity fit ($\text{lb sec}^{-1} \text{ft}^{-1}$) for aqueous sodium chloride solutions is given by:

$$\text{MUF}(Z) = (C2 * Z + B2) * Z + A2,$$

where:

Z = wt fraction NaCl,

$$A2 = (22.216306 - 16.460588 * \text{LOG}10T + 3.0949585 * \text{LOG}T2)/3600,$$

$$B2 = (-14.161033 + 16.430566 * \text{LOG}10T - 4.205263 * \text{LOG}T2)/3600,$$

$$C2 = (284.84684 - 219.40291 * \text{LOG}10T + 42.287729 * \text{LOG}T2)/3600,$$

$\text{LOG}10T = \log_{10} T$, and

$$\text{LOG}T2 = (\text{LOG}10T)^2.$$

In comparison with reference viscosities,⁽¹²⁾ the largest absolute error for 24 points selected over a temperature range of 32 to 100 $^{\circ}\text{F}$ and for concentrations up to 25 percent NaCl was less than 5 percent. The largest absolute error for 24 points selected over a temperature range of 100 to 200 $^{\circ}\text{F}$ and for the same concentration range was less than 10 percent. When this viscosity is used to calculate the Reynolds number to the 0.25 power, the resulting relative error is only about one-fourth as large.

4. Osmotic Pressure

Calculation of the osmotic pressure at the interface of the membrane for an aqueous sodium chloride solution is based on the simplified Equation 8.10 given

by Johnson, Dresner, and Kraus;⁽¹⁾ however, since, ideally, the osmotic pressure is proportional to the absolute temperature and inversely proportional to the molal volume, the equation was modified to include these proportionalities. In this form, the equation becomes:

$$\text{BETAF}(Z) = \text{CTAB} * \text{CMWF} * Z/(1-Z),$$

where:

Z = wt fraction NaCl,

$\text{CMWF} = 1000/58.45$ gmole NaCl/kg solvent, and

$\text{CTAB} = 48.8 * ((\text{TEMP} + 460)/537) * (\text{RHOF}(0)/\text{DEN77})$.

This simplified equation fits the data given by Stoughton and Lietzke⁽¹³⁾ to within 10 percent for temperatures ranging from 77 to 212° F and for concentrations ranging from 0.01 molal (~0.058 wt %) to 3 molal (~14.9 wt %) NaCl. The absolute error does not exceed one atmosphere for concentrations of less than one weight percent NaCl. The fit could be made more accurate if a variable osmotic coefficient was included.

5. First Derivative of Osmotic Pressure

The first derivative of osmotic pressure with respect to Z , wt fraction, is calculated by:

$$\text{DBETA}(Z) = \text{CTAB} * \text{CMWF}/(1-Z)^2.$$

6. Schmidt Number to the Two-Thirds Power

A least-squares fit for the Schmidt number to the exponent 2/3 was made using data for aqueous sodium chloride solutions and is calculated by:

$$\text{SC667}(Z) = \exp(\text{AA} + \text{BB} * Z),$$

where:

Z = wt fraction NaCl,

$\text{AA} = 5.637387 - 2.045509 * 10^{-2} * T + 0.29927225 * 10^{-4} * T^2$, and

$\text{BB} = 0.8673596 - 0.7184158 * 10^{-3} * T + 0.8589322 * 10^{-5} * T^2$.

The Schmidt number is defined as the ratio of the kinematic viscosity to the diffusion coefficient. In comparison with Schmidt numbers calculated using reference kinematic viscosities⁽¹⁴⁾ and diffusion coefficients,⁽¹⁵⁾ the largest absolute error for twelve points selected over a temperature range of 75 to 200° F and for concentrations up to 12 weight percent NaCl was less than 5

percent. The largest absolute error for 18 points selected over the same temperature range and for concentrations up to 20 weight percent NaCl was less than 7 percent. The relative error in the Schmidt number to the two-thirds power is, of course, only about two-thirds the error in the Schmidt number.

4.4.3 Calculation of Economic Results (Subroutine ECON)

Subroutine ECON, by use of "common" statements, has access to variables (of the same name) that were read in or set by other routines.

1. Calculate product flow, WPM (lb/sec), by:

$$WPM = WP/10358,$$

where the factor is 10358 (gals sec lb⁻¹ day⁻¹) which is based on a reference density of 62.4 lbs/ft³.

2. Calculate the number of tubes, NT, by:

$$NT = WPM/AX/(G(1) * VRCUM - G(N+1)).$$

3. Calculate the feed flow rates.

- a. On a weight basis, WFM (lb/sec) is calculated by:

$$WFM = G(1) * NT * VRCUM * AX.$$

- b. On a volume basis, WF (gal/day) is calculated by:

$$WF = WFM * 646317/RHO(1),$$

where the factor is 646317 (gals sec day⁻¹ ft⁻³).

4. Calculate the reject flow rates.

- a. On a weight basis, WRM (lb/sec) is calculated by:

$$WRM = G(N+1) * NT * AX.$$

- b. On a volume basis, WR (gal/day) is calculated by:

$$WR = WRM * 646317/RHO(N+1).$$

5. Calculate the pumping power, PP (kw), by:

$$PP = WFM * (2.8711 * P(1)/RHO(1) + 2.11 * 10^{-5} * (VEL(1))^2)/EP,$$

where EP is the pump efficiency, and the factors are $2.8711 \text{ (kw sec atm}^{-1} \text{ ft}^{-3})$ and $2.11 * 10^{-5} \text{ (kw sec}^3 \text{ ft}^{-2} \text{ lb}^{-1})$.

6. Calculate the recovery turbine power, TP (kw), by:

$$TP = WRM * (2.8711 * P(N+1)/RHO(N+1) + 2.211 * 10^{-5} * (VEL(N+1))^2)/ET,$$

where ET is the recovery turbine efficiency.

7. Calculate the unit power requirement, UNP (kwh/kgal), by:

$$UNP = 24000 * NP/WP,$$

where the factor is $24000 \text{ (gal hr kgal}^{-1} \text{ day}^{-1})$ and

NP, the net power, is calculated by:

$$NP = PP - TP.$$

8. Calculate the average flux, QOPT (kgal day⁻¹ ft⁻²), by:

$$QOPT = WP/A/1000,$$

where the factor is 1000 (gals/kgal) and

the area, A (ft²), is calculated by:

$$A = DFTPI * EFFLEN * NT.$$

9. Calculate the percent water recovery, WREC (percent), by:

$$WREC = 100 * WPM/WFM.$$

10. Calculate the unblended water costs.

- a. Area cost, C1 (cents/kgal), is calculated by:

$$C1 = CA * A/WP * 1000.$$

where CA is the unit area cost (cents day⁻¹ ft⁻²).

- b. Power cost, C_2 (cents/kgal), is calculated by:

$$C_2 = CE * UNP,$$

where CE is the unit energy cost (cents/kwh).

- c. Waste disposal cost, C_3 (cents/kgal), is calculated by:

$$C_3 = WR * CD/WP,$$

where CD is the unit brine removal cost (cents/kgal brine).

- d. The total unblended product water cost, CT (cents/kgal), is summed as:

$$CT = C_1 + C_2 + C_3.$$

11. When the product water concentration (CP_{MAX}) is less than the product limit ($PLIM$), blending with raw feed may be accomplished by blending feed of concentration x_f (C_{FEED}) with the product (concentration CP_{MAX}) to give a blended product with its concentration equal to the product limit ($PLIM$). Several additional parameters are calculated when blending takes place.

- a. Percent feed in blended product, PC_{FEED} (percent), is calculated by:

$$PC_{FEED} = 100 * (PLIM - CP_{MAX}) / (C_{FEED} - CP_{MAX}).$$

- b. Blended percent water recovery, BW_{REC} (percent), is calculated by:

$$BW_{REC} = 100 * (C_{FEED} * CR - C_{FEED}) / (C_{FEED} * CR - PLIM).$$

- c. Total blended cost, BC (cents/kgal), is calculated by:

$$BC = CT * (1 - (PLIM - CP_{MAX}) / (C_{FEED} - CP_{MAX})).$$

- d. Blended area cost, BC_1 , blended power cost, BC_2 , and blended waste disposal cost, BC_3 (cents/kgal), are calculated by:

$$\begin{aligned} BC_1 &= C_1 * BC/CT, \\ BC_2 &= C_2 * BC/CT, \text{ and} \\ BC_3 &= C_3 * BC/CT. \end{aligned}$$

4.5 Nomenclature

4.5.1 Variables Used in Equations in Sections 4.1 - 4.4

A	Membrane area (ft^2)
A_t	Tube cross-sectional area (ft^2)
B	Bulk flow in tube (lbs day^{-1})
C^*	Ion-exchange capacity (moles/kg water)
C_A	Unit cost of area (cents $\text{day}^{-1} \text{ft}^{-2}$)
C_E	Unit cost of power (cents kwh^{-1})
C_D	Unit cost of brine disposal (cents kgal^{-1} on brine basis)
C_P	Product water cost - incremental portion (cents kgal^{-1})
CR	Concentration ratio (see text, Equation 19)
d	Equivalent tube diameter (ft) (4x "mean hydraulic radius" for rectangular tubes)
D^*	Distribution coefficient (moles per kg water in membrane phase/moles per kg water in aqueous phase)
D_2^*	Special salt diffusion coefficient (see Johnson, et al) ⁽¹⁾
E_t	Turbine efficiency (overall)
E_M	Mixing efficiency (see Equation 12)
f	Fanning friction factor
F_p	Fractional water recovery
g_c	Gravitational constant ($32.2 \text{ lbs}_m \text{ ft lb}_f^{-1} \text{ sec}^{-2}$)
J_1	Water mass flux ($\text{kgs cm}^{-2} \text{ sec}^{-1}$)
g	Total volume flux ($\text{gals day}^{-1} \text{ft}^{-2}$)

$J_{1(v)}$	Volume water flux (gals day ⁻¹ ft ⁻²)
K	Salt rejection versus flux parameters (see Equation 13a)
l	Thickness of membrane
m_α	Salt concentration at inner (high-pressure) membrane interface (moles/kg solvent)
m_ω	Salt concentration at product interface (moles/kg solvent)
N_{Re}	Reynolds number
N_{Sc}	Schmidt number
P	Pressure (atm)
P_E	Power (kwh/day)
P_{eff}	Effective pressure (atm)
P_m	Membrane perimeter (ft)
P_α	Pressure at entrance membrane interface (atm)
P_ω	Pressure at product membrane interface (atm)
$\mathcal{P}$	Permeability (gals day ⁻¹ ft ⁻² atm ⁻¹)
R	Salt rejection of membrane (intrinsic)
$\bar{R}_{obs}$	Mean salt rejection relative to feed concentration
R_∞	Asymptotic salt rejection at large values of the parameter, σ
R_{AE}	Cost ratio, c_A/c_E
R_{DE}	Cost ratio, c_D/c_E
u	Axial velocity (ft sec ⁻¹)
u_f	Axial velocity at feed end of tube (ft sec ⁻¹)
u_r	Axial velocity at terminal end of tube in untapered plant (ft sec ⁻¹)

u_{ro}	Axial velocity at terminal end of tube in tapered plant (ft sec^{-1})
v	Radial velocity at membrane surface, product rate (ft sec^{-1})
W_R	Brine rejected (gals day^{-1})
W_P	Product rate (kgals day^{-1})
x_f	Feed salt concentration (wt fraction)
x_{rej}	Waste (reject) salt concentration (wt fraction)
x_α	Salt concentration at entrance membrane interface (wt fraction)
x_ω	Salt concentration at product interface (wt fraction)
x_t	Salt concentration in turbulent core (wt fraction)
$\bar{x}$	Cup mixing salt concentration (wt fraction)
y	Distance down tube (ft)
β	Coupling coefficient
Γ^*	Activity-coefficient quotient
π_α	Osmotic pressure at entrance interface (atm)
π_ω	Osmotic pressure at product interface (atm)
ρ_1	Density of water (lbs gal^{-1})
ρ_t	Density of turbulent stream (lbs gal^{-1})
ρ_ω	Density at product interface (lbs gal^{-1})
σ	Membrane flow parameter ($J_1 z / D_2^*$, dimensionless)
ψ_t	Concentration polarization

4.5.2 Variables Used in Subroutine TUBE in Subsection 4.4.2

A	Width of solid (no membrane) side of rectangular tube (ft)
AX	Cross-sectional area of tube or channel (ft ²)
A3	Constant term in polynomial fit for BDSTAR (dimensionless)
B (501)	Bulk velocity entering tube increment (lbs/sec)
BDSTAR	βD^* (see Section 4.2, Equations 14, 15, and 15a)
BETA	Net osmotic pressure at the membrane interface (atm)
BETAF	Osmotic pressure function statement (atm) (see Subsection 4.4.2.3)
BETSIG	$\beta \sigma$ (see Section 4.2, Equation 13a)
BONE	Bulk velocity at upstream end of tube increment (lbs/sec)
BTWO	Bulk velocity at downstream end of tube increment (lbs/sec)
C (501)	Average salt concentration entering tube increment (wt fraction)
CCW	Molal salt concentration at entrance interface (gmoles NaCl/kg solvent)
CFEED	Feed concentration, C(1) (wt fraction NaCl)
CMOLAL	Function statement for calculation of molality from concentration expressed as weight fraction (gmoles NaCl/kg solvent)
CMWF	Factor, 1000/58.45 (gmoles NaCl/kg NaCl)
CONE	Average salt concentration at upstream end of tube increment (wt fraction)
CONST	Constant for salt rejection versus flux (see K, Section 4.2, Equation 13a, ft ² day/kgal)
COST	Cost of product water, incremental portion (cents/kgal)
CP (501)	Salt concentration at product interface entering tube increment (wt fraction)

CPBAR (501)	Average product salt concentration for tube increment (wt fraction)
CPEND	Largest value of CPTWO for any tube increment (wt fraction)
CPMAX	Average product concentration for whole tube, CPROD(N+1) (wt fraction)
CPONE	Salt concentration at product interface at upstream end of tube increment (wt fraction)
CPROD (501)	Cumulative average salt concentration of product water at each increment (wt fraction)
CPTWO	Salt concentration at product interface at downstream end of tube increment (wt fraction)
CR	Concentration ratio (reject stream conc/feed conc)
CSTAR	Ion-exchange membrane capacity, C^* (moles/kg H ₂ O)
CSTARM	Ion-exchange membrane parameter, C^*/m (see Section 4.2, Equation 15a)
CTAB	Constant used to calculate osmotic pressure (atm kgs solvent/gmole NaCl (see Item 4.4.2.2.14)
CTWO	Average salt concentration at downstream end of tube increment (wt fraction)
CW (501)	Salt concentration at membrane entrance interface entering tube increment (wt fraction)
CWMCP	Salt concentration difference across membrane, CW-CP (wt fraction)
CWMCP2	Square of CWMCP (wt fraction ²)
CWONE	Salt concentration at membrane entrance interface at upstream end of tube increment (wt fraction)
CWTWO	Salt concentration at membrane entrance interface at downstream end of tube increment (wt fraction)
C11	Partial derivative $\partial \text{RES1} / \partial \text{CW}$ (see Step 4.4.2.1.3.f)
C12	Partial derivative $\partial \text{RES1} / \partial \text{CP}$ (see Step 4.4.2.1.3.f)

C21	Partial derivative $\partial \text{RES2} / \partial \text{CW}$ (see Step 4.4.2.1.3.f)
C22	Partial derivative $\partial \text{RES2} / \partial \text{CP}$ (see Step 4.4.2.1.3.f)
D	Diameter of tube, 4 X hydraulic radius (in)
DB (501)	Change in bulk velocity in tube increment (lbs/sec)
DBDDCW	Partial derivative $\partial \text{BDSTAR} / \partial \text{CW}$ (see Step 4.4.2.1.3.f)
DBETA	Function statement for first derivative of osmotic pressure with respect to concentration (see Subsection 4.4.2.3)
DBSDPE	Partial derivative $\partial \text{BETSIG} / \partial \text{PW}$ (see Step 4.4.2.1.3.f)
DC	Change in average salt concentration in tube increment (wt fraction)
DCP	Adjustment for finding CP, concentration at product interface (wt fraction)
DCW	Adjustment for finding CW, concentration at entrance interface (wt fraction)
DCWCM	Used in $\partial \text{BDSTAR} / \partial \text{CW}$ for ion-exchange membrane
DEDCP	Partial derivative $\partial E / \partial \text{CP}$
DEDCW	Partial derivative $\partial E / \partial \text{CW}$
DELTA1	Convergence criterion for Loop 2, 10^{-6}
DELTA2	Convergence criterion for Loop 2, $R * 10^{-6}$
DEN77	Density of water at 77° F (lbs/ft ³)
DFRDCP	Partial derivative $\partial ((1 - R)/R) / \partial \text{CP}$ (see Step 4.4.2.1.3.f)
DFRDCW	Partial derivative $\partial ((1 - R)/R) / \partial \text{CW}$ (see Step 4.4.2.1.3.f)
DFT	Diameter of tube, 4 X hydraulic radius (ft)
DFTPI	Effective membrane perimeter (ft)
DIFCP	Convergence of first inequality constraint, PLIM - CPMAX (wt fraction)

DIFCPE	Convergence of third inequality constraint, CFEED - CPEND (wt fraction)
DIFRE	Convergence of second inequality constraint, REMIN - RELIM
DL (501)	Length of tube increment (ft)
DLAST	Length of tube increment (ft)
DP (501)	Drop in gage pressure at membrane entrance interface for tube increment (atm)
DPFDCP	Partial derivative $\partial ZZZ / \partial CP$
DPFDCW	Partial derivative $\partial ZZZ / \partial CW$
DRHDCW	Partial derivative $\partial RHO / \partial CW$
DS	Salt flow rate in the product from tube increment (lbs salt/sec)
DV (501)	Change in axial velocity in tube increment (ft/sec)
DW	Water flow rate in product from tube increment (lbs water/sec)
DYDBD	Partial derivative $\partial (BDSTAR / (1 - BDSTAR) * EXPFBS) / \partial BDSTAR$
DYDBS	Partial derivative $\partial (BDSTAR / (1 - BDSTAR) * EXPFBS) / \partial BETSIG$
DZDPF	Partial derivative $\partial EXPZZZ / \partial ZZZ$
E (501)	Mixing efficiency at beginning of tube increment (wt fraction basis)
EFFLEN	Cumulative length of tube (ft)
EONE	Mixing efficiency at upstream end of tube increment
ERRDC	Material balance discrepancy (wt fraction salt)
ETWO	Mixing efficiency at downstream end of tube increment
EXITS	Alternate calculation of reject stream concentration (wt fraction salt)
EXPBS	Variable, $\exp (BETSIG)$
EXPFB	Variable, $\exp (BETSIG) / (\exp (BETSIG) - 1)$

EXPZZZ	Variable, exp (ZZZ), equal to mixing efficiency (see Section 4.2, Equation 12)
FBAR	Average Fanning friction factor in tube increment
FF	Term used to calculate Fanning friction factor, Blasius equation
FONE	Fanning friction factor at upstream end of tube increment
FRATIO	A "roughness factor" which may be used to modify the Fanning friction factor
FRTRAN	Fraction of salt transmitted (not rejected) by membrane (fraction)
FTWO	Fanning friction factor at downstream end of tube increment
FW	Ratio: membrane/wetted perimeter
G (501)	Mass velocity entering tube increment ($\text{lbs ft}^{-2} \text{sec}^{-1}$)
GC	Gravitational constant ($32.2 \text{ lbs}_m \text{ ft lb}_f^{-1} \text{sec}^{-2}$)
GONE	Mass velocity at upstream end of tube increment ($\text{lbs ft}^{-2} \text{sec}^{-1}$)
GTWO	Mass velocity at downstream end of tube increment ($\text{lbs ft}^{-2} \text{sec}^{-1}$)
IOUT	Lists flat (0 = searching)(1 = optimum answer found)
ITUBE	Tube geometry (1 = round)(2 = rectangular)
JTYPE	Reynolds number exponent (1 = XP not 0.25)(2 = XP is 0.25)
L (501)	Cumulative tube length (ft)
MTYPE	Membrane type (1 = ion exchange)(2 = polynomial fit)
MU (501)	Viscosity of solution entering tube increment ($\text{lbs sec}^{-1} \text{ft}^{-1}$)
MUF	Viscosity function statement ($\text{lbs sec}^{-1} \text{ft}^{-1}$)
MUONE	Viscosity at upstream end of tube increment ($\text{lbs sec}^{-1} \text{ft}^{-1}$)
MUTWO	Viscosity at downstream end of tube increment ($\text{lbs sec}^{-1} \text{ft}^{-1}$)
N	Number of tube Increments, INCR

NFLAG	Indicator showing type of error that occurred
P (501)	Gage pressure at membrane entrance interface (atm)
PERM	Membrane permeability ($\text{kgals ft}^{-2} \text{ atm}^{-1} \text{ day}^{-1}$)
PI	Constant, 3.1416
PLAST	Gage pressure at membrane entrance interface at downstream end of tube increment (atm)
PLIM	Maximum allowable salt concentration of product (wt fraction)
POF	Gage pressure at feed end of tube, P(1)(atm)
PONE	Gage pressure at membrane entrance interface, upstream end (atm)
PTWO	Gage pressure at membrane entrance interface, downstream end (atm)
PW (501)	Effective pressure at beginning of tube increment (atm)
PWBAR (501)	Average effective pressure for tube increment (atm)
PWONE	Effective pressure at upstream end of tube increment (atm)
PWTWO	Effective pressure at downstream end of tube increment (atm)
R	Fraction of salt rejected by membrane (fraction)
RA	Aspect ratio for rectangular tubes (membrane side/nonpermeable side)
RATE (501)	Product rate for tube increment ($\text{gals day}^{-1} \text{ ft}^{-2}$)
RE (501)	Reynolds number entering tube increment
REJ (501)	Salt rejection at beginning of tube increment (fraction)
REJECT	Salt rejection of membrane, initial guess for R (fraction)
RELIM	Minimum allowable Reynolds number
REMIN	Lowest Reynolds number for any tube increment
REONE	Reynolds number at upstream end of tube increment

RES1	Function "F" (see Step 4.4.2.1.3.b)
RES2	Function "G" (see Step 4.4.2.1.3.c)
RETWO	Reynolds number at downstream end of tube increment
REXP	Reynolds number to exponent XP
RE25	Reynolds number to exponent 0.25
RHO (501)	Density of turbulent stream entering tube increment (lbs/ft^3)
RHOBAR	Average density in tube increment (lbs/ft^3)
RHOF	Density function statement (lbs/ft^3)
RHOONE	Density of stream at upstream end of tube increment (lbs/ft^3)
RHOTWO	Density of stream at downstream end of tube increment (lbs/ft^3)
RLAST	Reynolds number at downstream end of tube increment
RONE	Salt rejection at upstream end of tube increment (fraction)
RTWO	Salt rejection at downstream end of tube increment (fraction)
S	Cumulative total salt flow in the product water (lbs salt/sec)
SC667	Function statement for Schmidt number to exponent 2/3
T	Temperature, TEMP (F)
TAPER	Taper (see Section 4.1, Equation 1)
TEMP	Temperature, T (F)
TWOC1	From density fit, $2 * C1$
V	Collection of terms in exponent of Equation 12 ($\text{ft sec}^{-1} \text{ atm}^{-1}$)
VEL (501)	Axial velocity entering tube increment (ft/sec)
VELBAR	Average axial velocity in tube increment (ft/sec)
VELONE	Axial velocity at upstream end of tube increment (ft/sec)

VELTWO	Axial velocity at downstream end of tube increment (ft/sec)
VIN	Axial velocity at feed end of tube, VEL(1) (ft/sec)
VRATIO	Velocity ratio (axial velocity) (see Step 4.4.2.1.8.c)
VRCUM	Velocity ratio (axial velocity) (see Step 4.4.2.1.8.k)
W (501)	Total water flow rate in product water (lbs water/sec)
XP	Reynolds number exponent used to calculate Fanning friction factor
XRATIO	Factor to modify exponent in Equation 12 for turbulence promoters
ZZ	Collection of terms in exponent of Equation 12 (1/atm)
ZZZ	Exponent in Equation 12; exp (ZZZ) is equal to mixing efficiency

5. USAGE

The program is written entirely in FORTRAN IV language. The version described in this report was used on an IBM 360/75 at the ORNL which had an IBM FORTRAN IV Level H compiler, Version 16. With the necessary modifications, the program could be used on a wide variety of computers.

One difficult aspect of using the program is choosing an initial set of values for the parameters to be optimized. If one or more constraints are violated at the initial conditions, an attempt is made to adjust the concentration ratio to give a feasible starting point. If this fails, the case is terminated and a new set of starting values must be selected.

Another problem is "scaling" the optimization variables. The speed with which the procedure converges on the optimum is highly dependent on the scaling of the variables being optimized. Scaling and efficiency are discussed in the operating instructions in Subsection 5.3.2 (Page 88).

A third problem is deciding whether the program has located a true optimum for the domain, a local optimum, or has quit because of numerical difficulties such as poor scaling. This problem is discussed in Subsection 5.3.5 (Page 92).

5.1 Input

Input variables are used to control optimization, calculate material balances, and calculate economic results. All necessary input data are read from cards according to the "Namelist" convention. In order to facilitate testing the code, all 56 Namelist variables have been initialized to create a "test case". Running the program with a blank card or no card in the first data set will cause the test case to be run for comparison with the sample lists (see Lists B-2 through B-5, Appendix).

Any number of cases may be run in one computer run. Once the value of a Namelist variable has been changed, the value will remain until it is changed by more input data or optimized. The 56 Namelist variables and their initial values are listed in Table 1, Page 25. Detailed operating instructions are found in Section 5.3 (Page 87).

5.2 Output

The output was designed for listing on a 132-character printer. Logical Units 6 and 1 are used with options available for suppressing some of the listing. The four basic types of output are:

1. Initial conditions for Namelist input variables.
2. Physical conditions and economic results at optimum.
3. Conditions encountered during optimization (List-6 option).
4. Detailed analysis of optimization procedure (List-1 option).

5.2.1 Initial Conditions for Namelist Input Variables

The initial values of all Namelist variables are listed first. This list, a sample of which appears in List B-2, uses Logical Unit 6 and always appears. All Namelist variables and their initial values for the test case are shown in Table 1. The only Namelist variables that change in value during optimization are the variables being optimized (zero to five of P, V, CR, TAPER, and D).

5.2.2 Physical Conditions and Economic Results at Optimum

A sample of the list printed upon termination of the optimization procedure is shown in List B-3. It is always printed at the end of the listing (Logical Unit 6) and may or may not be the true optimum (lowest relative cost) for the domain.

The optimum values of the variables being optimized (zero to five of $POF = P$, $VIN = V$, CR , $TAPER$, D) are shown at the top of this list. Most of the remaining variables at the top of the list are Namelist input variables.

In order to show conditions at different positions along the tube, the second part of this list shows the following parameters for every concentration increment:

1. Increment number
2. Gage pressure at upstream end of increment (atm)
3. Effective pressure (atm)
4. Velocity at upstream end of increment (ft/sec)
5. Reynolds number at upstream end of increment
6. Cumulative length at upstream end of increment (ft)
7. Mixing efficiency at upstream end of increment (fraction)
8. Membrane rejection at upstream end of increment (fraction)
9. Salt concentration (cup mixing) at upstream end of increment (wt fraction)
10. Salt concentration at entrance interface at upstream end of increment (wt fraction)
11. Mean salt concentration in product for increment (wt fraction)
12. Average flux for tube increment ($\text{gals day}^{-1} \text{ ft}^{-2}$)
13. Cumulative water production at upstream end of increment per tube at the feed end (lbs/sec)
14. Cumulative mean salt concentration at upstream end of increment (wt fraction)

The third part of this list shows economic results calculated in Subroutine ECON:

1. Number of tubes required (feed end of plant)
2. Water recovery (percent)
3. Feed stream flow rate (gals/day)
4. Reject stream flow rate (gals/day)
5. Feed-pump power (kw)
6. Recovery-turbine power (kw)
7. Net power requirement (kw, unblended basis)
8. Unit power (kwh/kgal unblended product)
9. Membrane area (ft^2)
10. Unit area cost (cents/kgal unblended product)
11. Unit power cost (cents/kgal unblended product)
12. Unit brine disposal cost (cents/kgal unblended product)
13. Average optimum flux ($\text{kgals day}^{-1} \text{ft}^{-2}$)
14. Plant length (ft)
15. Total unit cost (cents/kgal unblended product)
16. Percent feed in blended product (percent)
17. Blended water recovery (percent)
18. Unit area cost (cents/kgal blended product)
19. Unit power cost (cents/kgal blended product)
20. Unit disposal cost (cents/kgal blended product)
21. Total unit cost (cents/kgal blended product)
22. Execution time for case and accumulated execution time

5.2.3 Conditions Encountered During Optimization

A list that follows the progress of the optimization is available. Often, more than 100 separate evaluations of the cost function are required for gradient evaluation and steps while searching. As can be seen in List B-4, one line is printed for each evaluation. This list, using Logical Unit 6, can be printed or suppressed through use of the List-6 option (LIST6 = 0, print. LIST6 = 1, skip). It is recommended that this list be obtained in order to spot cases that terminated due to unexpected difficulties. The results printed on each line are:

1. One character showing the purpose of the evaluation:
 - a. I: Initial point, called by HYPFIL
 - b. G: gradient evaluation, called by HYPFIL
 - c. L: searching along gradient, called by LINCON
 - d. A: searching along acceleration (parallel tangent) vector direction, called by LINCON
2. Inlet pressure (atm gage)
3. Inlet velocity (ft/sec)
4. Concentration ratio
5. Plant taper
6. Tube diameter (in)
7. Plant length (ft)
8. Average optimum flux ($\text{kgals day}^{-1} \text{ ft}^{-2}$)
9. Water recover (percent)
10. Difference between salt concentration in product water and product limit (wt fraction)
11. Difference between maximum salt concentration in product at any point in the plant and feed concentration (wt fraction)

12. Difference between minimum Reynolds number at any point in the plant and minimum allowable Reynolds number.
13. Unit area cost (cents/kgal, unblended basis)
14. Unit power cost (cents/kgal, unblended basis)
15. Total unit water cost (cents/kgal, blended or unblended, as specified)

5.2.4 Detailed Analysis of Optimization Procedure

This list, like the preceding one, follows the progress of the optimization. This one, however, shows details of calculations made by HYPFIL and LINCON rather than physical conditions and costs calculated by TUBE and ECON. This list uses Logical Unit 1 and can be listed or suppressed by using the List-1 option (LIST1 = 0, print. List 1 = 1, skip). This list may be written separately, but the control cards supplied define Logical Unit 1 as standard output on the same unit as Logical Unit 6. In this case, if both this list and the previous list (Subsection 5.2.3) are requested, the lines of output will be interspersed. It is recommended that this list be used only for debugging purposes or for learning how the optimization routine functions. A sample of this list is presented in List B-5.

5.3 Operating Instructions

HYPFIL is the main routine; all other routines are either subroutines or functions.

5.3.1 Use of Namelist for Data Input

The only method used for input of data during execution of the program is reading input cards according to the "Namelist" convention. The 56 Namelist variables and their initial values are shown in Table 1 and List B-2.

Each set of data cards is read by HYPFIL at Step 4.4.1.1.2.a (Page 24). Any number of cases may be run during one computer run. Once the value of a Namelist variable has been changed, that value will remain for the rest of the computer run unless it is reset. Therefore, it is not necessary to keep resetting variables to the same value in each data set.

The first card of a data set must have a blank in Column 1 and the characters & RESET in Columns 2 through 7. The last card of a set must have a blank in Column 1 and the characters & END in Columns 2 through 5. Any number of data cards containing Namelist variables and assigned values may be placed between the & RESET card and the & END card. (A FORTRAN manual should be consulted because the Name-list format was changed to allow a minimum of one card per data set.) The card format is a blank in Column 1 and a list of variables and values, separated by commas, in Columns 2 through 80. Each variable name must be identical in spelling to one of the 56 Namelist variables shown in Table 1. The variable name is followed by an equal sign, =, and a value. For example:

TAPER = -1.0, WP = 2.0D6, MTYPE = 1, P = 100.

Blanks are ignored. Commas at the end of each card are optional. If a variable is changed more than once in one data set, the latest value is kept. A blank data card or no data cards will result in the optimization being run with no change in the starting conditions. The test case is run in this manner by using a blank data set with the program as supplied.

The program terminates normally by reading the end of file after using all the data sets.

5.3.2 Optimization Variables and Scaling

The five independent variables available for optimization are also elements of the XB array. The Namelist variable names, in order, are:

1. P XB(1), Inlet pressure
2. V XB(2), Inlet velocity
3. CR XB(3), Concentration ratio
4. TAPER XB(4), Taper
5. D XB(5), Tube diameter

From zero through five variables, in any combination, may be optimized.

In HYPFIL, Step 4.4.1.1.3.e (15), Page 28, the weighted range is calculated by:

$$R(I) = \text{ABS}((XH(I) - XL(I)) * WT(I)),$$

where:

XH represents the upper limit or bound (= PU, VU, CRU, TAPERU, DU),

XL the lower limit (= PL, VL, CRL, TAPERL, DL), and

WT a weighting factor (= PWT, VWT, CRWT, TWT, DWT).

The range is used to scale or "normalize" the variables in an attempt to make the optimization run more efficiently by making long elliptical contours appear more circular. The term $(X_H - X_L)$ is a good first approximation. Further adjustment may be accomplished by adjusting the weighting factors. Increase the factor to make a variable change more rapidly; decrease the factor to retard a component. This is an arbitrary modification made by the authors simply because it showed merit.

If the range for any variable is zero, that variable will be held constant and not optimized. This is accomplished by setting the corresponding weighting factor to zero or by making the upper and lower bounds equal. If all the ranges are zero, the results are printed out for the given set of conditions.

On the first pass through Loop 2 in HYPFIL (Item 4.4.1.1.3, Page 26), N, the number of variables the program attempts to optimize is equal to NVARL, the lower limit. If NVARU, the upper limit on the number of variables the program attempts to optimize, is greater than NVARL, then $NVARU - NVARL + 1$ optimizations are done for one set. N has the value NVARU on the final pass. Any variables that are not being optimized will remain constant. Any variable that would normally be optimized can be forced to remain constant by causing the corresponding weighted range, $R(I)$, to be calculated as zero in the manner just described.

As an example, if $NVARL = 3$ and $NVARU = 4$, the first three variables, starting with P, will be optimized on the first pass. On the second and final pass, the fourth variable (TAPER) will be optimized as well. The optimum conditions for the first pass become the starting conditions for the second pass. If desired, inlet pressure could be removed from the optimization by setting $PWT = 0$, or by setting PL and PU equal to the same value ($PL = 40.$, $PU = 40.$).

5.3.3 Additional Information Concerning Namelist Input Variables

A brief description of each of the 56 Namelist input variables is given in Table 1. This section contains additional helpful information for several of these variables.

1. Information concerning the optimization variables (P, etc), their limits (PL, PU, etc), and their weighting factors (PWT, etc) can be found in Subsection 5.3.2. NVARL and NVARU are also described.
2. INCR - The upper limit is 500 tube increments. A value of 50 increments has been found to be a good compromise between speed and accuracy.

3. WP - The optimum relative water cost is actually independent of the water production rate, WP (gal/day), since only variable costs (not fixed costs) are included.
4. TEMP - Temperature-dependent physical properties are calculated once for each case at ENTRY SETUP in TUBE. If temperature were ever made an optimized variable, these physical property equations would have to be moved to the top of Subroutine TUBE.
5. FRATIO - See TUBE, Step 4.4.2.1.2.m (Page 54). This "roughness" factor has been used in turbulence promoter studies.
6. XRATIO - See TUBE, Step 4.4.2.1.1.q (Page 52). This number is used to modify the exponent in Equation 12 when studying turbulence promoters.
7. ALPH - Initial value of ALPHA, used for approximating gradient. Value 0.001 found by experience.
8. BETA - Initial value of BET which is used in determining step size while searching. Value of 0.005 found by experience. A ratio of BETA/ALPH of 5 is recommended. Both ALPH and BETA may be changed as experience is gained or due to the nature of a problem.
9. LOOPS - See HYPFIL, Step 4.4.1.1.3.d (Page 26).
10. MAXTIM - If the time is available from Function ICLOCK, this limit can be used to terminate a case after a reasonable amount of time since the program will occasionally get stuck in a loop. The time is an integer expressed in 0.01 second. The test case will run in 2 or 3 minutes on an IBM 360/75. The initial value (MAXTIM = 1080000) represents 3 hours.
11. LOGCR, CRADJ - See HYPFIL, Step 4.4.1.1.3.c (Page 26). Flag LOGCR indicates use of factor CRADJ to adjust CR when NVARU is greater than NVARL.
12. LIST1, LIST6 (see Subsections 5.2.3, Page 86, and 5.2.4, Page 87) - It is recommended that LIST1 = 1 and LIST6 = 0 be used.
13. PERM - Membrane permeability is assumed to be constant (not a function of pressure, velocity, or time).
14. PLIM - A value of 0.000500 weight fraction NaCl has been used as the maximum allowable salt concentration of the product in recent studies.
15. RELIM - A minimum allowable Reynolds number of 2100 assures turbulence. Lower values have been used in turbulence promoter studies.

16. MTYPE - See TUBE, Step 4.4.2.1.3.d (Page 55). Either a univalent ion exchange membrane (MTYPE = 1) or a polynomial fit (MTYPE = 2) can be specified.
17. CSTAR - This is the membrane capacity C^* (moles/kg H₂O) shown in Equation 15a for ion exchange membranes.
18. A3, B3, C3, D3 - These coefficients are used in a polynomial fit of the type: $\beta D^* = A3 + (B3)(X) + (C3)(X^2) + (D3)(X^3)$. See Step 4.4.2.1.3.e (Page 55).
19. CONST - This variable is used to calculate $\beta\sigma$ for both ion exchange and polynomial fit membranes. It is the K in Equation 13a (see Section 4.2, Page 16, and Step 4.4.2.1.3.c, Page 54).
20. XP - See TUBE, Step 4.4.2.1.2.m (Page 54). The Reynolds number exponent in Equation 10 has been changed from 0.25 to other values in turbulence promoter studies.
21. ITUBE, RA - Both round and rectangular channels are allowed. D must be four hydraulic radii for rectangular tubes. In the test case, RA, the "aspect ratio", is 10 (length of membrane side/length of nonpermeable side).
22. IBLEND - See Subroutine ECON, Subsection 4.4.3.11 (Page 71). The optimization proceeds on an unblended basis (IBLEND = 1) or on a blended basis (IBLEND = 2). In blending, raw feed water is added to the product water until concentration PLIM is reached.

5.3.4 Errors Occurring During Program Execution

Most of the conditions that cause a case to be skipped in main program HYPFIL are described in Subsection 4.4.1.1 (Page 23). A warning note is printed if Namelist variable MAXTIM causes a case to be terminated after a specified length of time. The operating system will terminate the program in HYPFIL upon reading an end of file (not an error) or an unrecognizable Namelist input data set, and will print an appropriate note.

Subroutine LINCON immediately returns control to HYPFIL if IER (= NFLAG) is not 1 after a call to Subroutine EVAL, since this indicates that an error (described below) occurred in Subroutine TUBE. Subroutine LINCON also returns control to HYPFIL if the variables cannot be acceptably adjusted in 10 iterations.

Several different types of error conditions cause termination of processing in Subroutine TUBE. The values of indicator NFLAG and the associated error conditions are:

1. No error
2. Gage pressure, P , zero or negative
3. Effective pressure, PW , zero or negative
4. Velocity, VEL , negative
5. Avoid exponential overflow due to low velocity
6. Failure of iterative loop to converge
7. Mixing efficiency negative or greater than one
8. Salt concentration, CW , exceeds solubility limit

If an error is detected, the cost is set to a large number to cause an unsuccessful step and Subroutine TUBE returns control to Subroutine EVAL.

5.3.5 Validity of the Optimum

It must be decided for each case whether the program has converged on the true optimum (minimum cost) for the domain, a local optimum, or has quit due to problems such as poor scaling. Experience and the List-6 option printout that follows the progress of the optimization (Subsection 5.2.3, Page 86) are helpful. The use of several different starting points is a helpful technique for verifying an optimum. Weighting factors and step-size parameters may be adjusted as experience is gained. By plotting data obtained in parametric studies, it is usually easy to determine which cases should be rerun.

5.4 Example

A base case was established using a typical case from recent studies. The values of all Namelist input variables are shown in Table 1 and List B-2. The resulting output appears as sample lists in Lists B-2 through B-5; and is described in Section 5.2, Page 83. This test case can be used to test the program by using a blank card in the first set of data. The resulting output should match the sample lists in this report.

6. CODING INFORMATION

A complete listing of the FORTRAN statements is presented in List B-1. The control cards used when running on the IBM 360/75 at the Oak Ridge National Laboratory have been supplied with the deck, but some may have to be changed depending on the installation.

The program makes extensive use of "commons", as well as arguments, to transfer information between routines. With very few exceptions, the names of common variables are the same in different routines.

The core space required, currently 130,000 bytes, can be reduced to 50,000 bytes by reducing the size of the large arrays (23 arrays dimensioned 501) in Subroutine TUBE.

The initial values of Namelist variables may be changed in the FORTRAN deck, but it is probably better to read temporary changes from input data cards. Variables may be added to the Namelist or common statements. Equality constraints and additional inequality constraints can also be added. Additional variables could also be optimized.

The running time for each case depends on the number of variables optimized, the starting values, and optimization parameters (such as ALPH, BETA, and weighting factors) selected. A typical case for three variables requires two or three minutes on the IBM 360/75. Since the program spends over 99 percent of the time in Subroutine TUBE, apparent inefficiencies in HYPFIL and Subroutine LINCON (for example, flexibility such as allowing for variable numbers of equality and inequality constraints) are negligible.

7. REFERENCES

- (1) Johnson, J. S., Dresner, L., and Kraus, K. A.; Hyperfiltration (Reverse Osmosis), Chapter 8 in Principles of Desalination, K. W. Spiegler, Editor; Academic Press, New York, New York (1966).
- (2) Sherwood, T. K., Brian, P. L. T., Fisher, R. E., and Dresner, L.; Ind and Eng Chem Fundamentals, 4, p 113 (1965).
- (3) Shor, A. J., Kraus, K. A., Johnson, J. S., and Smith, W. T., Jr.; Ind and Eng Chem Fundamentals, 7, p 44 (1968).
- (4) Kraus, K. A., Shor, A. J., and Johnson, J. S.; Desalination, 2, p 243 (1967).
- (5) Copper, J. R. and Kephart, W. L.; Some Improvements in the Gradient Search Method of Optimization, K-DP-1063, Union Carbide Corporation-Nuclear Division, Oak Ridge, Tennessee (1965).
- (6) Zellnik, H. E., Sondak, N. E., and Davis, R. S.; Chem Eng Prog, 58, p 35 (1962).

- (7) Cross, K. E.; A Gradient Projection Method for Constrained Optimization, K-1746, Union Carbide Corporation-Nuclear Division, Oak Ridge, Tennessee (1968).
- (8) Beuhler, R. J., Shah, B. V., and Kempthorne, O.; Some Properties of Steepest Ascent and Related Procedures for Finding Optimum Conditions, Iowa State University Statistical Laboratory (1961).
- (9) Funderlic, R. E.; The Programmers Notebook - A Compendium of Numerical Analysis Utility Programs, p 131, K-1729, Union Carbide Corporation-Nuclear Division, Oak Ridge, Tennessee (1968).
- (10) Thomas, D. G., Oak Ridge National Laboratory, Union Carbide Corporation-Nuclear Division, Oak Ridge, Tennessee; to be published.
- (11) Saline Water Conversion Engineering Data Book, Curve OSW 12.60, M. W. Kellogg Co (1966).
- (12) Saline Water Conversion Engineering Data Book, Curve OSW 12.90, M. W. Kellogg Co (1960).
- (13) Stoughton, R. W. and Lietzke, M. H.; Journal of Chem and Eng Data, 10, 3, p 254 (1965).
- (14) Saline Water Conversion Engineering Data Book, Curve OSW 12.910, M. W. Kellogg Co (1966).
- (15) Saline Water Conversion Engineering Data Book, Curve OSW 12.921, M. W. Kellogg Co (1966).
- (16) Vieth, W. R., Porter, J. H., and Sherwood, T. K.; Ind and Eng Chem Fundamentals, 2, p 1 (1963).

8. ACKNOWLEDGEMENTS

The authors wish to acknowledge, with thanks, the continuing interest and many helpful discussions with J. S. Johnson, K. A. Kraus, and D. G. Thomas of the Oak Ridge National Laboratory, and the assistance of W. L. Kephart (now at Bonner and Moore Associates, Inc, Houston, Texas), J. R. Copper (now at the Coca Cola Company, Atlanta, Georgia), and K. E. Cross, of the Oak Ridge Computing Technology Center in testing, developing, and modifying their optimization procedure for use with the present computer program. The work and suggestions by L. P. Pasquier during the development of the computer program are also greatly appreciated.

9. APPENDIX A

9.1 Salt Concentration in a Turbulent Core

As noted in Section 4.2, Equation 9, Page 17, the present version of the program assumes the average "cup mixing" weight fraction of salt at the point y ; that is, $\bar{x}$ is equal to x , the weight fraction of salt in the turbulent core. In order to justify this assumption, differences have been estimated which would occur using the equations for fully developed turbulent flow by Vieth.⁽¹¹⁾ Their development employs normalized distance and concentrations, $Y = r/r_t$ and $X = \frac{x - x_t}{x_\alpha - x_t}$, so the boundary conditions for the cross-sectional concentration gradient are $X = 1$ at $Y = 0$ and $X = 0$ at $Y = 1$. At any point Y , the concentration X is given by:

$$X = \frac{b_1}{3\alpha^{1/3}} \left\{ \frac{1}{2} \ln \left[\frac{(\alpha^{-1/3} + Y)^2}{\alpha^{-2/3} - \alpha^{-1/3} Y + Y^2} \right] + \sqrt{3} \tan^{-1} \left(\frac{2Y - \alpha^{-1/3}}{\alpha^{-1/3} \sqrt{3}} \right) \right\} + b_2 \quad (\text{A-1})$$

for $X = \frac{x - x_t}{x_\alpha - x_t}$ and $Y = \frac{2r}{d}$,

where: $\alpha = \frac{\beta N_{Sc}}{8} \left(N_{Re} \frac{f}{2} \right)$ and $\beta = 1.77$.

For $\alpha^{-1/3} \ll 1$, the integration constants b_1 and b_2 can be closely approximated by:

$$b_1 = \left(\frac{dX}{dY} \right)_{Y=0} \approx \frac{-3\sqrt{3}}{2\pi} \alpha^{1/3}, \text{ and}$$

$$b_2 \approx 3/4.$$

This concentration gradient was integrated with a computer program written in PL-1 language developed by Frank Hammerling of the Oak Ridge Computer Technology Center to obtain values for $\bar{x}$ for parametric values of N_{Re} and N_{Sc} using a modified Simpson's rule with an adjustable step size. Estimates of error for this procedure were developed to determine the fineness of subdivisions required.

The results obtained are presented in Table A-1. It is evident that the maximum overstatement of x_o by assuming it is equal to $\bar{x}$ would be about 5 percent in the transition region for $N_{Sc} = 100$. For greater values of N_{Re} or N_{Sc} , the relative error is less. Since concentrations and distances have been normalized, these relative results are independent of the degree of concentration polarization and tube diameter; however, occurrence of low Reynolds numbers would obviously be associated with small tubes. It is noteworthy that the calculated differences (before rounding) can be fitted to a cubic polynomial (Equation A-2) in $\alpha^{-1/3}$ with a maximum deviation from the calculated value of 0.17 percent as:

$$\left[\left(23.5437\alpha^{-1/3} - 4.59907 \right) \alpha^{-1/3} + 1.992539 \right] \alpha^{-1/3} + 8.8 \times 10^{-6}. \quad (A-2)$$

Table A-1
CALCULATED RATIOS OF NORMALIZED CUP MIXING SALT
CONCENTRATION, $\bar{x}$, AND TURBULENT CORE
SALT CONCENTRATION, x_t
(Weight Fraction Minus One)

N_{Re}	N_{Sc}				
	100	200	300	400	500
2,100	0.0540	0.0433	0.0380	0.0347	0.0323
5,000	0.0289	0.0231	0.0202	0.0184	0.0171
10,000	0.0174	0.0139	0.0121	0.0110	0.0103
20,000	0.0104	0.0083	0.0073	0.0066	0.0061
30,000	0.0077	0.0061	0.0054	0.0049	0.0045
40,000	0.0062	0.0050	0.0043	0.0039	0.0037
50,000	0.0053	0.0042	0.0037	0.0035	0.0031
100,000	0.0032	0.0025	0.0022	0.0020	0.0018

Therefore, it would be unnecessary to employ the complicated integration procedure just described in the program if this refinement were to be incorporated.

Fair approximations are also given for a quadratic and even linear fits by:

$$\left(-3.63575\alpha^{-1/3} + 1.982692 \right) \alpha^{-1/3} + 2.78 \times 10^{-5}, \text{ and} \quad (A-3)$$

$$1.895194\alpha^{-1/3} + 0.0002797. \quad (A-4)$$

10. APPENDIX B

List B-1. HYPFIL Source Program.

List B-2. Namelist Input Variables.

List B-3. Results at Optimum.

List B-4. Progress of Optimization (List-6 option)

List B-5. Progress of Optimization (List-1 and List-6 options).

List B-1. HYPFIL Source Program

```

C
C HYPFIL - MAIN PROGRAM TO HANDLE INPUT, OPTIMIZATION FOR OPTM0010
C HYPERFILTRATION (REVERSE OSMOSIS) STUDIES. OPTM0020
C OPTM0030
C OPTM0040
C OPTIMIZATION PROG AUTHORS - W.L.KEPHART, K.E.CROSS, J.R.COPPER OPTM0050
C COMPUTING TECHNOLOGY CENTER, OAK RIDGE, TENN. 37830. OPTM0060
C MODIFIED THROUGH 07/69 BY R.M.KELLER, R.S.DILLON, W.L.GRIFFITH OPTM0070
C UNION CARBIDE NUCLEAR DIV, 9704-2, Y-12 PLANT, OPTM0080
C P O BOX Y, OAK RIDGE, TENN. 37830. OPTM0090
C OPTM0100
C FORTRAN IV, IBM 360, 029 KP. OPTM0110
C OPTM0120
C ACCELERATION SCHEME WITH RIDGE RESTRAINTS. OPTM0130
C SF1 FIRST VECTOR STEP SIZE FACTOR. OPTM0140
C SF2 SUB INTERVAL STEP SIZE FACTOR. OPTM0150
C RF REDUCTION FACTOR AFTER UNSUCCESSFUL STEP. OPTM0160
C ALPHA VARIABLE RANGE OF DERIVATIVE EVALUATION. OPTM0170
C BET RANGE FACTOR OF VARIABLE CHANGING MOST RAPIDLY. OPTM0180
C KNT(J) NUMBER OF TIMES VARIABLE IS HELD ON BOUNDARY. OPTM0190
C INEQUALITY CONSTRAINTS (NOW USING FIRST 2 ONLY). OPTM0200
C CPM, PLIM MAX PRODUCT SALT CONC. OPTM0210
C REM, RELIM MIN REYNOLDS NUM ALLOWED. OPTM0220
C CPE, CFEEED PRODUCT CONC SHOULD NOT EXCEED FEED CONC. OPTM0230
C OPTM0240
C * * * * * OPTM0250
C OPTM0260
C IMPLICIT REAL*8(A-H,O-Z) OPTM0270
C OPTM0280
C NAME OF COMMON TELLS WHICH ROUTINES USE IT.. O=OPT=HYPFIL, OPTM0290
C L=LINCON, E=EVAL, T=TUBE, C=ECON. OPTM0300
C COMMON / COMOT / PERM,FRATIO,XRATIO, REJECT,TEMP,CONST, OPTM0310
C 1 A3,B3,C3,D3,XP,RA, CSTAR, CRA, MTYPE,ITUBE,INCR OPTM0320
C COMMON / COMOTC / WP,CA,CE,CD,EP,ET, IBLEND OPTM0330
C COMMON / CMOLTC / STEP,DATE,CFEED,PLIM,RELIM, LIST1,LIST6,LINES OPTM0340
C COMMON / COMOL / R(20), KON(10),K OPTM0350
C OPTM0360
C REAL*8 X(20),XB(20),XL(20),XH(20),DX(20), ACC1(20), OPTM0370
C 1 ACC2(20), ANUM(20),FP(20),FP1(20),FPC(20,10), SLEFAC(20) OPTM0380
C DIMENSION KNT(20),LIM(20),WT(20) OPTM0390
C DATA STEPA,STEPG,STEPI,STEPL / 1HA,1HG,1HI,1HL / OPTM0400
C OPTM0410
C NAMELIST STARTING POINTS, LOWER LIMITS, UPPER LIMITS EQUIVALENCED OPTM0420
C TO XB, XL, XH ARRAYS. OPTM0430
C EQUIVALENCE (XB(1),PI),(XL(1),PL),(XH(1),PU),(XB(2),V),(XL(2),VL), OPTM0440
C 1 (XH(2),VU),(XB(3),CR),(XL(3),CRL),(XH(3),CRU),(XB(4),TAPER), OPTM0450
C 2 (XL(4),TAPERL),(XH(4),TAPERU),(XB(5),D),(XL(5),DL),(XH(5),DU), OPTM0460
C 3 (WT(1),PWT),(WT(2),VWT),(WT(3),CRWT),(WT(4),TWT),(WT(5),DWT) OPTM0470
C OPTM0480
C NAMELIST RESET USED IN HYPFIL TO CHANGE INPUT VARIABLES. OPTM0490
C NAMELIST / RESET / P,PL,PU, V,VL,VU, CR,CRL,CRU, TAPER,TAPERL, OPTM0500
C 1 TAPERU, D,DL,DU, NVARL,NVARU,INCR, WP,TEMP,CFEED,PLIM,RELIM, OPTM0510
C 2 FRATIO,XRATIO,PERM,REJECT, MTYPE,A3,B3,C3,D3,CONST, XP, OPTM0520
C 3 ITUBE,RA, IBLEND,CA,CE,CD,EP,ET, LIST1,LIST6, MAXTIM,LOGCR, OPTM0530
C 4 CRADJ, ALPH,BETA,PWT,VWT,CRWT,TWT,DWT, LOOPS, CSTAR OPTM0540
C OPTM0550
C * * * * * OPTM0560
C OPTM0570
C GET CLOCK TIME, DATE (LOCAL FUNCTIONS, ARG 0) OPTM0580
C ITIME0 = ICLOCK(0) OPTM0590
C ITIME1 = ITIME0 OPTM0600
C CALL ICAY(CATE) OPTM0610
C * * * * * OPTM0620
C OPTM0630
C * * SECTION TO INITIALIZE NAMELIST VARIABLES TO BASE CASE CONDITIONS OPTM0640
C WHICH ARE USED UNLESS CHANGED BY READING INPUT CARDS. SEE OPTM0650

```

```

C      FORMAT 6C01 FOR UNITS OF VARIABLES.
C      *****
C      SET XB STARTING VALUES FOR P, V, CR, TAPER, D.
C      ONE OR MORE VAR ARE OPTIMIZED STARTING WITH FIRST, P.
C      XB(1) = 50.0 DO
C      XB(2) = 5.0 DO
C      XB(3) = 2.0 DO
C      XB(4) = 0.0 DO
C      XB(5) = 0.454500
C      SET XL LOWER LIMITS. EQUIV TO PL, VL, CRL, TAPERL, DL.
C      XL(1) = 0.0 DO
C      XL(2) = 0.0100
C      XL(3) = 1.00100
C      XL(4) = -30.000
C      XL(5) = 0.0 DO
C      SET XH UPPER LIMITS. PU, VU, CRU, TAPERU, DU.
C      XH(1) = 750.0 DO
C      XH(2) = 100.0 DO
C      XH(3) = 15.0 DO
C      XH(4) = 30.0 DO
C      XH(5) = 3.0 DO
C      LOWEST, HIGHEST NUM OF VAR TO BE OPTIM. NUM OF CONC INCREMENTS.
C      NVARL = 3
C      NVARU = 3
C      INCR = 50
C      WP, PRODUCT (GPD). TEMP, F. CFEEED, FEED CONC (WT FR). PLIM, PROD
C      CONC LIM (WT FR). RELIM, MIN RE NUM.
C      WP = 1.006
C      TEMP = 70.0 DO
C      CFEEED = 0.005000 DO
C      PLIM = 0.000500 DO
C      RELIM = 2100.0 DO
C      FRATIO, FRIC FACT RATIO. XRATIO, LOG MIX EFFIC.
C      FRATIO = 1.0 DO
C      XRATIO = 1.0 DO
C      PERM, MEMB PERM. REJECT, SALT REJ (FR). MTYPE, MEMB TYPE (1= ION
C      EXCH) (2= POLY FIT). CONSTS FOR BDSTAR, A3,B3,C3,D3,CONST.
C      PERM = 0.000500 DO
C      REJECT = 0.95 DO
C      MTYPE = 2
C      A3 = 0.0500
C      B3 = 0.0 DO
C      C3 = 0.0 DO
C      D3 = 0.0 DO
C      CONST = 165.0 DO
C      ION EXCHANGE.
C      CSTAR = 0.35 DO
C      XP, RE NUM EXPON. ITUBE, (1=ROUND) (2=RECT). RA, RECT RATIO(GT 1).
C      XP = 0.25 DO
C      ITUBE = 2
C      RA = 10.0 DO
C      IBLEND, (1=NO) (2=YES). CA, AREA COST (C/SQFT D). CE, ELEC COST
C      (C/KWH). CD, WASTE DISP (C/KGAL). EP, PUMP EFFIC. ET, TURB EFF.
C      IBLEND = 2
C      CA = 1.0 DO
C      CE = 1.0 DO
C      CD = 0.0 DO
C      EP = 0.7 DO
C      ET = 0.0 DO
C      LIST1, (0= PRINT DETAILED PROGRESS OF OPTIMIZATION ON TAPE 1 )
C      (1=SKIP). LIST6, (0= PRINT 1 LINE/CALL TO TUBE, 6) (1=SKIP).
C      LIST1 = 1
C      LIST6 = 0
C      TIME LIMIT, 180 MIN IN 0.01 SEC.
C      MAXTIM=1080000
C      LOGCR MEANS (1=ADJUST CR) (0=NO). CRADJ = FACTOR.
C      LOGCR=0
C      CRADJ = 0.9 DO
C      STEP SIZES AND WEIGHTING FACTORS.

```

```

OPTMC660
OPTMC670
OPTMC680
OPTMC690
OPTMC700
OPTMC710
OPTMC720
OPTMC730
OPTMC740
OPTMC750
OPTMC760
OPTMC770
OPTMC780
OPTMC790
OPTMC800
OPTMC810
OPTMC820
OPTMC830
OPTMC840
OPTMC850
OPTMC860
OPTMC870
OPTMC880
OPTMC890
OPTMC900
OPTMC910
OPTMC920
OPTMC930
OPTMC940
OPTMC950
OPTMC960
OPTMC970
OPTMC980
OPTMC990
OPTM1000
OPTM1010
OPTM1020
OPTM1030
OPTM1040
OPTM1050
OPTM1060
OPTM1070
OPTM1080
OPTM1090
OPTM1100
OPTM1110
OPTM1120
OPTM1130
OPTM1140
OPTM1150
OPTM1160
OPTM1170
OPTM1180
OPTM1190
OPTM1200
OPTM1210
OPTM1220
OPTM1230
OPTM1240
OPTM1250
OPTM1260
OPTM1270
OPTM1280
OPTM1290
OPTM1300
OPTM1310
OPTM1320
OPTM1330
OPTM1340
OPTM1350

```

```

ALPH = 0.001 DO                                OPTM1360
BETA = 0.005 DO                                OPTM1370
PWT = 1. DO                                    OPTM1380
VWT = 1. DO                                    OPTM1390
CRWT = 1. DO                                   OPTM1400
TWT = 1. DO                                    OPTM1410
DWT = 1. DO                                    OPTM1420
C LOOPS (1, DO 8010 LOOP ONCE PER N) (2, TWICE / N, SAVE TIME) OPTM1430
  LOOPS = 1                                     OPTM1440
C END OF NAMELIST INIT. VALUES USED UNTIL CHANGED. OPTM1450
C * * * * * OPTM1460
C BEGIN LOOP 1.                                OPTM1470
C * * 1492 IS RETURN POINT FOR LOOP WHICH READS 1 SET OF INPUT DATA. OPTM1480
C ONE OR MORE OPTIMIZATIONS MAY BE DONE IN DO 8010 LOOP. OPTM1490
C OPTM1500
C OPTM1510
C * * * * * OPTM1520
1492 CONTINUE OPTM1530
C OPTM1540
C * READ ONE SET OF NAMELIST VAR TO RESET VALUES. NEW VALUES ARE KEPT OPTM1550
C UNTIL CHANGED BY ANOTHER DATA SET. NOT RESET TO BASE CASE. OPTM1560
C OPTM1570
C READ (5,RESET) OPTM1580
C OPTM1590
C NOW PRINT OUT NAMELIST VAR VALUES. OPTM1600
6001 FORMAT(1H1/ 'O HYPERFILTRATION DESIGN OPTIMIZATION - INPUT VARI OPTM1610
1ABLES (NAMELIST RESET)', 45X,A8 / OPTM1620
A 'O INITIAL VALUE',9X,'LOWER LIMIT',11X, OPTM1630
2 'UPPER LIMIT',10X,'WEIGHTING FACTOR',5X,'VARIABLE' / 1H / OPTM1640
3 4X,6HP ,F12.6, 4X,6HPL ,F12.6, 4X,6HPU ,F12.6, OPTM1650
4 3X,6HPWT ,F12.6,3X,'INLET PRESSURE (ATM) ' / OPTM1660
5 4X,6HV ,F12.6, 4X,6HVL ,F12.6, 4X,6HVV ,F12.6, OPTM1670
6 3X,6HVWT ,F12.6,3X,'INLET VELOCITY (FT/SEC) ' / OPTM1680
7 4X,6HCR ,F12.6, 4X,6HCRL ,F12.6, 4X,6HCRU ,F12.6, OPTM1690
8 3X,6HCRWT ,F12.6,3X,'CONCENTRATION RATIO, WASTE/FEED ' / OPTM1700
9 4X,6HTAPER ,F12.6, 4X,6HTAPERL,F12.6, 4X,6HTAPERU,F12.6, OPTM1710
X 3X,6HTWT ,F12.6,3X,'TAPER ' / OPTM1720
1 4X,6HD ,F12.6, 4X,6HDL ,F12.6, 4X,6HDD ,F12.6, OPTM1730
2 3X,6HDWT ,F12.6,3X,'INSIDE TUBE DIAMETER (INCHES) ' ) OPTM1740
WRITE(6,6001) DATE, P,PL,PU,PWT, V,VL,VU,VWT, CR,CRL,CRU,CRWT, OPTM1750
1 TAPER,TAPERL,TAPERU,TWT, D,DL,DU,DWT OPTM1760
C OPTM1770
6003 FORMAT(1H / 1H / 2(4X,'VARIABLE VALUE DESCRIPTION',29X)/1H /OPTM1780
14X,6HNVARL ,I12 ,3X,40HMIN NUM OF VARIABLES TO BE OPTIMIZED , OPTM1790
24X,6HPERM ,F12.6,3X,40HMEMBRANE PERMEABILITY(KGAL/SQFT/ATM/DAY)/ OPTM1800
34X,6HNVARU ,I12 ,3X,40HMAX NUM OF VARIABLES TO BE OPTIMIZED , OPTM1810
44X,6HCFEED ,F12.6,3X,40HFEED CONCENTRATION (WT FRACTION) / OPTM1820
54X,6HINCR ,I12 ,3X,40HNUMBER OF CONC INCREMENTS , OPTM1830
64X,6HPLIM ,F12.6,3X,40HMAX SALT CONC OF PRODUCT (WT FRACTION) / OPTM1840
74X,2HWP ,1PD16.2,3X,35HPRDUCT FLOW RATE (GAL/DAY) ,9X, OPTM1850
8 6HRELIM ,OPF12.6,3X,40HMINIMUM REYNOLDS NUM ALLOWABLE / OPTM1860
94X,6HTEMP ,F12.6,3X,40HTEMPERATURE (DEG F) , OPTM1870
X4X,6HMTYPE ,I12 ,3X,40HMEMB TYPE (1= ION EXCH) (2= POLY FIT) / OPTM1880
14X,6HFRATIO,F12.6,3X,40H'ROUGHNESS FACTOR' , OPTM1890
24X,6HCSTAR ,F12.6,3X,40HION-EXCH MEMB CAPACITY (MOLES/KG H2O) / OPTM1900
34X,6HXRATIO,F12.6,3X,40HMODIFY EXPONENT, TURB PROMOTERS , OPTM1910
44X,6HCONST ,F12.6,3X,40HSALT REJ VS FLUX (SQFT DAY/KGAL) / OPTM1920
54X,6HREJECT,F12.6,3X,40HSALT REJECTION, INIT GUESS (FRACTION) , OPTM1930
64X,6HA3 ,F12.6,3X,40HPOLY FIT CONSTANT TERM , OPTM1940
WRITE (6,6003) NVARL,PERM,NVARU,CFEED,INCR,PLIM,WP,RELIM, OPTM1950
1 TEMP,MTYPE,FRATIO,CSTAR,XRATIO,CONST,REJECT,A3 OPTM1960
C OPTM1970
6004 FORMAT( OPTM1980
74X,6HALPH ,F12.6,3X,40HSTEP SIZE FOR GRADIENT , OPTM1990
84X,6HB3 ,F12.6,3X,40HPOLY FIT / OPTM2000
94X,6HBETA ,F12.6,3X,40HSTEP SIZE FOR SEARCHING , OPTM2010
X4X,6HC3 ,F12.6,3X,40HPOLY FIT / OPTM2020
14X,6HLCOPS ,I12 ,3X,40HNUM LOOPS PER CASE (1= GNE,REG) (2= TWO), OPTM2030
24X,6HD3 ,F12.6,3X,40HPOLY FIT / OPTM2040
34X,6HMAXTIM,I12 ,3X,40HMAX TIME PER CASE (.01 SEC) , OPTM2050

```



```

44X,6HXP      F12.6,3X,40HREYNOLDS NUM EXPONENT      / OPTM2060
54X,6HLOGCR ,I12 ,3X,40HADJUST CR (0= NO) (1= YES)    , OPTM2070
64X,6HITUBE ,I12 ,3X,40HTUBE SHAPE (1= ROUND) (2= RECTANGULAR) / OPTM2080
74X,6HCRADJ ,F12.6,3X,40HCR ADJUSTMENT FACTOR        , OPTM2090
84X,6HRA      ,F12.6,3X,40HASPECT RATIO,RECT (MEMB SIDE/SOLID SIDE)/ OPTM2100
94X,6HLIST1 ,I12 ,3X,40H(0= LIST ON TAPE 1) (1= SKIP) , OPTM2110
X4X,6HIBLEND,I12 ,3X,40HBLEND FEED WITH PRODUCT (1= NO)(2= YES) / OPTM2120
14X,6HLIST6 ,I12 ,3X,40H(0= LIST ON TAPE 6) (1= SKIP) , OPTM2130
24X,6HCA      F12.6,3X,40HUNIT COST OF MEMB AREA (CENTS/SQFT/DAY) ) OPTM2140
WRITE (6,6004) ALPH,B3,BETA,C3,LOOPS,D3,MAXTIM,XP,LOGCR,ITUBE, OPTM2150
1 CRADJ,RA,LIST1,IBLEND,LIST6,CA                      OPTM2160
C                                                        OPTM2170
6007 FORMAT(                                           OPTM2180
34X,6HEP      ,F12.6,3X,40HPUMP EFFICIENCY (FRACTION) , OPTM2190
44X,6HCE      F12.6,3X,40HUNIT COST OF ELEC POWER (CENTS/KWH) / OPTM2200
54X,6HET      ,F12.6,3X,40HRECOV TURBINE EFFICIENCY (FRACTION) , OPTM2210
64X,6HCD      ,F12.6,3X,40HUNIT COST OF WASTE DISPOSAL (CENTS/KGAL) ) OPTM2220
WRITE (6,6007) EP,CE,ET,CD                          OPTM2230
C                                                        OPTM2240
C CHECK STARTING VALUES.                             OPTM2250
C IF ZERO SKIP TO NEXT CASE.                          OPTM2260
C IF(XB(1).LE.0.D0) GO TO 1492                        OPTM2270
IF(XB(2).LE.0.D0) GO TO 1492                        OPTM2280
IF(XB(3).LE.0.D0) GO TO 1492                        OPTM2290
IF(XB(5).LE.0.D0) GO TO 1492                        OPTM2300
IF(PERM .LE.0.D0) GO TO 1492                       OPTM2310
IF(CFEED.LE.0.D0) GO TO 1492                       OPTM2320
C                                                        OPTM2330
C CALL SETUP IN TUBE TO INITIALIZE VARIABLES FOR THIS CASE. OPTM2340
CALL SETUP                                           OPTM2350
C                                                        OPTM2360
C * * * * *                                           OPTM2370
C * * * * *                                           OPTM2380
C BEGIN LOOP 2.                                       OPTM2390
C * * DO 8010 N LOOP. ONE OR MORE OPTIMIZATIONS MAY BE DONE. FIRST TIME OPTM2400
C OPTIMIZE NVARL VARIABLES STARTING WITH P. ORDER OF VAR = P, V, OPTM2410
C CR, TAPER, D. LOOP IS DONE ((NVARU-NVARL)+1) TIMES. OPTM2420
C * * * * *                                           OPTM2430
C * * * * *                                           OPTM2440
C * * * * *                                           OPTM2450
C DO 8010 N=NVARL,NVARU                             OPTM2460
C                                                        OPTM2470
C LOOP = 0                                           OPTM2480
90 LINES=100                                         OPTM2490
LOOP = LOOP + 1                                     OPTM2500
GO TO (91,92), LOOP                                OPTM2510
C LOWER CR IF LOOP IS DONE MORE THAN ONCE.          OPTM2520
91 IF(N.NE.NVARL.OR.LOGCR.EQ.1) XB(3)=(XB(3)-1.0D0)*CRADJ + 1.0D0 OPTM2530
IF(N.NE.NVARL.OR.LOGCR.EQ.1) WRITE(6,6006) CR      OPTM2540
6006 FORMAT( 1H1/ ' * * CR HAS BEEN ADJUSTED TO',F9.4,' * */1H ) OPTM2550
IF(LOOPS.EQ.1) GO TO 93                            OPTM2560
INCRSV = INCR                                       OPTM2570
INCR = 5                                           OPTM2580
LIST6 = 1                                          OPTM2590
GO TO 93                                           OPTM2600
92 INCR = INCRSV                                    OPTM2610
LIST6 = 0                                          OPTM2620
93 ALPHA = ALPH                                     OPTM2630
BET = BETA                                         OPTM2640
CMM = -1.0 DO                                     OPTM2650
SF1 = 1.5 DO                                       OPTM2660
SF2 = 0.9 DO                                       OPTM2670
RF = 0.25DO                                       OPTM2680
RMAX = 0.0 DO                                     OPTM2690
STEP = STEPI                                       OPTM2700
K = 0                                              OPTM2710
ISTOP = 1                                          OPTM2720
KNTVEC = -1                                        OPTM2730
KT2=3                                              OPTM2740
KT5 = 1                                            OPTM2750

```

```

DO 73 I=1,N
KNT(I) = 0
FP1(I) = 0.0 DO
SLEFAC(I) = 1.0 DO
R(I) = DABS((XH(I)-XL(I))*WT(I))
RMAX = DMAX1(RMAX, R(I))
LIM(I) = 0
IF(R(I).EQ.0.00) LIM(I) = 1
73 CONTINUE
IF(RMAX.LE.0.00) RMAX = 1. DO
DO 29 I=1,20
29 X(I) = XB(I)
K=K+1
C CALL EVAL WHICH CALLS TUBE, ECON TO GET INITIAL COST.
337 CALL EVAL(XB,FB,0,IER,CPM,REM,CPE)
C
IF(IER.NE.1) GO TO 340
IF(CPM.LE.PLIM+1.0D-10 .AND. REM.GE.RELIM-0.02D0) GO TO 341
IF(N.LT.3 .OR. R(3).LE.0.00) GO TO 338
C
C CALL ADJUST TO SET CR TO LOWER ACCEPTABLE VALUE.
340 CALL ADJUST
C
IF(CRA.LT.CRL) GO TO 338
XB(3) = CRA
GO TO 337
C
338 IS ERROR. CALL EVAL, GO TO NEXT CASE.
338 CALL EVAL(XB,FB,1,IER,CPM,REM,CPE)
IF(LOOPS.EQ.1) GO TO 1492
INCR = INCRSV
LIST6 = 0
GO TO 1492
C
C
341 TCPE=CPE
TCPM=CPM
TREM=REM
C
INITIALIZATION FOR THIS CASE IS DONE.
C
C
C * * * * *
C
C BEGIN A NEW VECTOR DIRECTION STARTING AT POINT XB.
C EITHER GRADIENT OR ACCELERATION (PARALLEL TANGENT) DIRECTION.
C
C * * * * *
C
3 L=0
KT=4
KNTVEC = KNTVEC + 1
SF=SF1
SCALE=CMM
CPM=TCPM
REM=TREM
CPE=TCPE
IF(LIST1.EQ.0) WRITE(1,100) K,L,FB,(XB(I),I=1,N)
C
C * * * * *
C
C STORE ACCELERATION VECTOR COORDINATES OR CALC ITS DIRECTION.
C
C * * * * *
C
GO TO (200,204,201),KT5
200 IF(KNTVEC -1)5,201,201
201 DO 202 I = 1,N
ACC1(I) = ACC2(I)
202 ACC2(I) = XB(I)
IF(KNTVEC.EQ.1) GO TO 5
KT5 = 2
GO TO 5
C
CALC ACCEL VECTOR DIRECTION.
204 DO 205 I = 1,N
IF(R(I).NE.0.00) ANUM(I) = (XB(I)-ACC1(I))*CMM*RMAX / R(I)

```

OPTM2760
OPTM2770
OPTM2780
OPTM2790
OPTM2800
OPTM2810
OPTM2820
OPTM2830
OPTM2840
OPTM2850
OPTM2860
OPTM2870
OPTM2880
OPTM2890
OPTM2900
OPTM2910
OPTM2920
OPTM2930
OPTM2940
OPTM2950
OPTM2960
OPTM2970
OPTM2980
OPTM2990
OPTM3000
OPTM3010
OPTM3020
OPTM3030
OPTM3040
OPTM3050
OPTM3060
OPTM3070
OPTM3080
OPTM3090
OPTM3100
OPTM3110
OPTM3120
OPTM3130
OPTM3140
OPTM3150
OPTM3160
OPTM3170
OPTM3180
OPTM3190
OPTM3200
OPTM3210
OPTM3220
OPTM3230
OPTM3240
OPTM3250
OPTM3260
OPTM3270
OPTM3280
OPTM3290
OPTM3300
OPTM3310
OPTM3320
OPTM3330
OPTM3340
OPTM3350
OPTM3360
OPTM3370
OPTM3380
OPTM3390
OPTM3400
OPTM3410
OPTM3420
OPTM3430
OPTM3440
OPTM3450

```

      IF(R(I).EQ.0.DO) ANUM(I) = 0.DO
205 CONTINUE
      KT5 = 3
      STEP = STEPA
      IFLAG=3
      IF(ICLOCK(0)-ITIME1.GT.MAXTIM) GO TO 7111
C     CALL LINCON TO ADJUST THE ACCELERATION VECTOR DIRECTION.
      CALL LINCON(ANUM,XB,XH,XL,KNT,N,IFLAG,CMM,FPC,FL,CPM,REM,CPE)
C
      GO TO 89
C     ACCEL VECTOR SKIPS RESTRAINING FACTOR SLEFAC. GO TO STEP SIZE.
C     * * * * *
C     APPROXIMATING PARTIALS BY ONE SIDED DIFFERENCES.
C     * * * * *
C     5 SCPE=CPE
      SCPM=CPM
      SREM=REM
      STEP = STEPG
      IF(LIST6.EQ.0) WRITE(6,6008)
6008 FORMAT(10X,1H )
      LINES = LINES+1
      IF(LIST1.EQ.0) WRITE(1,1409) REM,CPM,CPE
1409 FORMAT(5HOREM=,E15.8,5X,4HCPM=,E15.8,5X,4HCPE=,E15.8)
      DO 4 I=1,N
4     X(I)=XB(I)
C
C     IF VARIABLE IS ON BOUNDARY, IT IS HELD THERE FOR 5 GRAD VECTOR
C     DIRECTION CHANGES BEFORE IT IS USED AGAIN. THE NUMBER OF VARIABLES
C     IS REDUCED BY ONE FOR THE NEXT 5 GRAD VECT DIRECTION CHANGES.
C
      DO 984 J=1,N
        SL = FB
        IF(LIM(J)-1)969,72,969
969 IF((X(J)-XH(J))*(X(J)-XL(J))168 ,67,67
67 KNT(J) = KNT(J)+1
        IF(KNT(J)-5)72,68,68
68 KNT(J) = 0
C     ARBITRARY STEP ALPHA*R FOR EACH VAR.
        X(J) = X(J) + ALPHA*R(J)
        K=K+1
        IF(ICLOCK(0)-ITIME1.GT.MAXTIM) GO TO 7111
        IF(LOOP.EQ.1 .AND. (ICLOCK(0)-ITIME1).GT.6000 .AND.
1     LOOPS.EQ.2) GO TO 7111
C     CALL EVAL WHICH CALLS TUBE, ECON TO GET COST.
      CALL EVAL(X ,FH,0,IER,CPM,REM,CPE)
C
      IF(LIST1.EQ.0) WRITE(1,1409) REM,CPM,CPE
      IF(LIST1.EQ.0) WRITE(1,106) K,L,FH,(X(I),I=1,N)
      GO TO 6
72 FH = 0. DO
      SL = 0. DO
      6 IF(R(J).EQ.0.DO) GO TO 7
C     PARTIAL DERIV OF COST FCT, 3 CONSTRAINTS.
      FP(J) = (FH-SL)/ALPHA
      FPC(J,1) = -(CPM-SCPM)/ALPHA
      FPC(J,2) = (RFM-SREM)/ALPHA
      FPC(J,3) = -(CPE-SCPE)/ALPHA
      GO TO 35
7     FP(J) = 0. DO
      FPC(J,1) = 0. DO
      FPC(J,2) = 0. DO
      FPC(J,3) = 0. DO
35 ANUM(J) = FP(J)
      X(J) = XB(J)
984 CONTINUE
      STEP = STEPL
      IF(LIST1.EQ.0) WRITE(1,107) (FP(I),I=1,N)
107 FORMAT(1H0,42HPARTIAL DERIVATIVES OF GRADIENT EVALUATION/

```

OPTM3460
 OPTM3470
 OPTM3480
 OPTM3490
 OPTM3500
 OPTM3510
 OPTM3520
 OPTM3530
 OPTM3540
 OPTM3550
 OPTM3560
 OPTM3570
 OPTM3580
 OPTM3590
 OPTM3600
 OPTM3610
 OPTM3620
 OPTM3630
 OPTM3640
 OPTM3650
 OPTM3660
 OPTM3670
 OPTM3680
 OPTM3690
 OPTM3700
 OPTM3710
 OPTM3720
 OPTM3730
 OPTM3740
 OPTM3750
 OPTM3760
 OPTM3770
 OPTM3780
 OPTM3790
 OPTM3800
 OPTM3810
 OPTM3820
 OPTM3830
 OPTM3840
 OPTM3850
 OPTM3860
 OPTM3870
 OPTM3880
 OPTM3890
 OPTM3900
 OPTM3910
 OPTM3920
 OPTM3930
 OPTM3940
 OPTM3950
 OPTM3960
 OPTM3970
 OPTM3980
 OPTM3990
 OPTM4000
 OPTM4010
 OPTM4020
 OPTM4030
 OPTM4040
 OPTM4050
 OPTM4060
 OPTM4070
 OPTM4080
 OPTM4090
 OPTM4100
 OPTM4110
 OPTM4120
 OPTM4130
 OPTM4140
 OPTM4150

```

11H ,7E16.8/1H ,7E16.8/1H ,7E16.8) OPTM4160
IFLAG=3 OPTM4170
DO 69 I=1,10 OPTM4180
69 KON(I) = 0 OPTM4190
IF(ICLOCK(0)-ITIME1.GT.MAXTIM) GO TO 7111 OPTM4200
C CALL LINCON TO ADJUST THE PARTIALS (GRAD VECT DIRECTION). OPTM4210
CALL LINCON(ANUM,XB,XH,XL,KNT,N,IFLAG,CMM,FPC,FL,CPM,REM,CPE) OPTM4220
C ***** OPTM4230
C RIDGE ANALYSIS. SLEFAC RESTRAINS OSCILLATING VARIABLES. OPTM4240
C ***** OPTM4250
C ***** OPTM4260
C ***** OPTM4270
DO 14 I = 1,N OPTM4280
IF(FP1(I)*ANUM(I)) 12,14,13 OPTM4290
12 SLEFAC(I) = SLEFAC(I)*0.64 DO OPTM4300
GO TO 14 OPTM4310
13 SLEFAC(I) = SLEFAC(I)*1.25 DO OPTM4320
SLEFAC(I)=DMIN1(SLEFAC(I),1.000) OPTM4330
14 FP1(I) = ANUM(I) OPTM4340
IF(LIST1.EQ.0) WRITE(1,1429) (SLEFAC(J),J=1,N) OPTM4350
1429 FORMAT(8H SLEFAC=,7E15.6) OPTM4360
209 DO 80 J=1,N OPTM4370
80 ANUM(J)=ANUM(J)*SLEFAC(J) OPTM4380
C ***** OPTM4390
89 CONTINUE OPTM4400
IFLAG=3 OPTM4410
C CALL LINCON TO ADJUST THE DIRECTION (GRAD OR ACCEL). OPTM4420
CALL LINCON(ANUM,XB,XH,XL,KNT,N,IFLAG,CMM,FPC,FL,CPM,REM,CPE) OPTM4430
C ***** OPTM4440
C ***** OPTM4450
C ***** OPTM4460
C CALC DX, TERM TO BE USED IN STEP SIZE (DX(I)*SCALE*BOUND). OPTM4470
C ***** OPTM4480
C ***** OPTM4490
IF(KT5.EQ.3) GO TO 11 OPTM4500
C GRADIENT VECTOR. OPTM4510
AMAX = 0. DO OPTM4520
DO 8 J=1,N OPTM4530
AMAX = DMAX1(AMAX, DABS(ANUM(J) )) OPTM4540
8 CONTINUE OPTM4550
DO 9 J=1,N OPTM4560
IF(AMAX.GT.0.00) DX(J) = ANUM(J)*R(J)*BET / AMAX OPTM4570
IF(AMAX.LE.0.00) DX(J) = 0.00 OPTM4580
9 CONTINUE OPTM4590
GO TO 20 OPTM4600
C ACCELERATION (PARALLEL TANGENT) VECTOR. 0.05 DIST BETWEEN PTS. OPTM4610
11 DO 10 J=1,N OPTM4620
10 DX(J) = ANUM(J)*R(J)*0.0500 / RMAX OPTM4630
C ***** OPTM4640
C ***** OPTM4650
C CALCULATES NEW VECTOR AND CHECKS PREDICTED VECTOR IS EXTERIOR OPTM4660
C TO BOUNDED DOMAIN. MULT STEP BY BOUND (SMALLEST VALUE 0 - 1) OPTM4670
C TO PUT ALL VAR IN OR ON BOUNDARY. OPTM4680
C ***** OPTM4690
C ***** OPTM4700
20 IBOUND = 0 OPTM4710
BOUNDT = 1.000 OPTM4720
BOUND = 1.000 OPTM4730
85 IBOUND=0 OPTM4740
21 DO 22 I=1,N OPTM4750
IF(I-IBOUND)83,22,83 OPTM4760
STEP FROM XB TO X. OPTM4770
83 X(I)=XB(I)+DX(I)*SCALE*BOUND OPTM4780
IF(KT5.EQ.3) GO TO 56. OPTM4790
IF(KNT(I))22,56,22 OPTM4800
56 IF(X(I)-XH(I))43,43,44 OPTM4810
44 IF(X(I)-XB(I) .NE. 0.00) BOUNDT = (XH(I)-XB(I)) / (X(I)-XB(I)) OPTM4820
IF(X(I)-XB(I) .EQ. 0.00) BOUNDT = 0. DO OPTM4830
X(I)=XH(I) OPTM4840
GO TO 42 OPTM4850

```

```

43 IF(X(I)-XL(I))45,42,42                                OPTM4860
45 IF(XB(I)-X(I) .NE. 0.00) BOUNDT = (XB(I)-XL(I)) / (XB(I)-X(I)) OPTM4870
   IF(XB(I)-X(I) .EQ. 0.00) BOUNDT = 0. 00                OPTM4880
   X(I)=XL(I)                                              OPTM4890
42 IF(BOUND-BOUNDT)22,22,82                                OPTM4900
82 BOUND=BCUNDT                                           OPTM4910
   IBOUND=I                                              OPTM4920
   IBOUNDT=I                                             OPTM4930
22 CONTINUE                                              OPTM4940
   IF(IBOUNT)85,86,85                                    OPTM4950
86 CONTINUE                                              OPTM4960
C   * * * * *                                           OPTM4970
C   EVALUATION OF POINT ON PREDICTED VECTOR.              OPTM4980
C   * * * * *                                           OPTM4990
C   * * * * *                                           OPTM5000
C   * * * * *                                           OPTM5010
C   IFLAG=1                                              OPTM5020
C   IF(ICLOCK(0)-ITIME1.GT.MAXTIM) GO TO 7111            OPTM5030
C   CALL LINCON TO ADJUST THE VARIABLES TO SATISFY THE CONSTRAINTS. OPTM5040
C   CALL LINCON(ANUM,X ,XH,XL,KNT,N,IFLAG,CMM,FPC,FL,CPM,REM,CPE) OPTM5050
C   * * * * *                                           OPTM5060
C   IF(LIST1.EQ.0) WRITE(1,1409) REM,CPM,CPE             OPTM5070
C   IF(LIST1.EQ.0) WRITE(1,106) K,L,F,(X(I),I=1,N)       OPTM5080
106 FORMAT(3H0 ,14,2X,14,10X,E16.8/3H ,7E16.8/3H ,7E16.8) OPTM5090
C   F=FL                                                 OPTM5100
C   CHECK VALUE F FOR IMPROVEMENT. 23 IF BETTER, 30 IF WORSE. OPTM5110
C   IF(CMM*(F-FB))30,30,23                                OPTM5120
C   * * * * *                                           OPTM5130
C   * * * * *                                           OPTM5140
C   GOOD MOVE ALONG VECTOR. STEP IMPROVED POSITION.       OPTM5150
C   * * * * *                                           OPTM5160
C   * * * * *                                           OPTM5170
23 L=L+1                                                 OPTM5180
   ISTOP = 1                                             OPTM5190
   FB=F                                                 OPTM5200
   TCPM=CPM                                             OPTM5210
   TREM=REM                                             OPTM5220
   TCPE=CPE                                             OPTM5230
   DO 24 I=1,N                                           OPTM5240
24  XB(I)=X(I)                                           OPTM5250
   IF(KT5.EQ.3) GO TO 26                                OPTM5260
   IF(L.NE.30) GO TO 71                                  OPTM5270
C   * * * * *                                           OPTM5280
C   VARIABLES ARE RELATIVELY INSENSITIVE,WE ARE MOVING TOO SLOWLY, OPTM5290
C   LETS SPEED THINGS ALONG.                            OPTM5300
C   * * * * *                                           OPTM5310
C   AFTER 30 GOOD STEPS INCREASE ALPHA; BET, START NEW VECT DIR. OPTM5320
C   ALPHA = ALPHA*5.000                                  OPTM5330
C   BET = BET*5.000                                       OPTM5340
C   IF(LIST1.EQ.0) WRITE(1,65)                            OPTM5350
65  FORMAT(1H0,47HPROCEEDING TOO SLOWLY,INCREASING ALPHA AND BETA, OPTM5360
      127H,BEGIN ANEW FROM THIS POINT)                   OPTM5370
      KT2=3                                                OPTM5380
      GO TO 3                                              OPTM5390
C   * * * * *                                           OPTM5400
71 IF(L.NE.10 .AND. L.NE.20) GO TO 26                    OPTM5410
C   AFTER 10 GOOD STEPS INCREASE ALPHA, BET OR RESET SF=SF1. OPTM5420
C   AFTER 20 GOOD STEPS INCREASE ALPHA, BET.             OPTM5430
C   IF(SF.NE.SF2) GO TO 74                                OPTM5440
C   SF = SF1                                              OPTM5450
C   GO TO 26                                              OPTM5460
74 ALPHA = ALPHA*2.000                                  OPTM5470
   BET = BET*2.000                                       OPTM5480
C   MULT SCALE BY 1.5 TO INCREASE STEP SIZE (OR BY 0.9 TO DECREASE). OPTM5490
26 SCALE=SCALE*SF                                         OPTM5500
   KT2=3                                                 OPTM5510
C   IF(LIST1.EQ.0) WRITE(1,64) L                         OPTM5520
64  FORMAT(1H0,19HSUCCESSFUL STEP NO.,12)                OPTM5530
   IF(IBOUND.LE.0) GO TO 89                              OPTM5540
88 ANUM(IBOUND) = 0.00                                    OPTM5550

```

```

      KNT(IBOUND)=KNT(IBOUND)+1
      GO TO 89
C      * * * * *
C      THIS STEP WAS A SEARCH FAILURE. DETERMINE COURSE OF ACTION.
C      KT5 = 3 WHEN ACCELERATION STEP IS BEING TAKEN.
C      * * * * *
C      30 KT = KT-1
      GO TO (34,34,31),KT
C      KT=3, FIRST FAILURE. 20 IF NO GOOD STEPS, 3 IF GOOD STEPS.
      31 CONTINUE
      IF(LIST6.EQ.0 .AND. L.LE.0) WRITE (6,6009)
      IF(LIST6.EQ.0 .AND. L.GT.0) WRITE (6,6010) ALPHA
6009 FORMAT('0FIRST STEP FAILED. REDUCE STEP, TRY AGAIN.')
```

```

6010 FORMAT('0THIS STEP FAILED. ALPHA=', F12.9)
      LINES = LINES + 2
      IF(L.GE.1) GO TO 3
      SF = SF2
      SCALE=SCALE*RF
      IF(LIST1.EQ.0) WRITE(1,59)
59  FORMAT(1H0, 'FIRST SEARCH FAILURE ALONG VECT DIRECTION,' ,
      1  ' NOW SCANNING LAST INTERVAL.' )
      GO TO 20
C      KT=2 (NEVER 1) SECOND FAILURE. GO BACK TO 3, RESTART UNLESS DONE.
      34 IF(LIST6.EQ.0) WRITE(6,6010) ALPHA
      LINES = LINES+1
      IF(KT5.EQ.3) GO TO 3
      IF(L.GE.1) GO TO 3
C      BACK TO 3 IF ACCEL. OR ANY GOOD GRAD STEPS.
      ALPHA = ALPHA/20.0DO
      BET = BET/20.0DO
217 DO 213 I = 1,N
213 ACC2 (I) = ACC1 (I)
      IF(KNTVEC-3)214,215,215
214 KT5 = 1
      KNTVEC = KNTVEC-1
      GO TO 216
215 KT5 = 3
216 CONTINUE
      IF(LIST1.EQ.0) WRITE(1,62)
      KT2 = KT2-1
      62 FORMAT(1H0,39HUNSUCCESSFUL FIRST STEP WHILE SCANNING ,
      152HWITH REDUCED SF AFTER UNSUCCESSFUL FIRST STEP WHILE ,
      225HSCANNING WITH ENLARGED SF/ 1H ,
      354HREDUCE ALPHA,BETA,RE EVALUATE ELSE OPTIMUM IS OBTAINED)
      GO TO (46,3),KT2
C      PARABOLIC FIT CANNOT BE USED BECAUSE OF INEQUALITY CGNSTRAINTS.
C
C      CASE HAS BEEN COMPLETED, UNLESS INDEPENDENT VARIABLE WAS
C      HELD ON BOUNDARY FOR LAST VECTOR EVALUATION.
C      RELEASE VAR 1 TIME FOR ONE MORE GRAD CALC.
C
      46 GO TO (15,18),ISTOP
C      QUIT WHEN KT=2, KT2=1, ISTOP=1(KNT=0) OR ISTOP=2.
      15 DO 17 I = 1,N
      IF(KNT(I))17,17,16
      16 ISTOP=2
      KT2=2
      KT5 = 3
      17 KNT(I)=5
      GO TO (18,3),ISTOP
      18 GO TO 7112
C
C      WRITE NOTE WHEN QUITTING ON TIME LIMIT, CALL EVAL FOR LIST.
7111 WRITE(6,6005)
6005 FORMAT(1H1/ ' * * THIS CASE PULLED ON TIME. * * ANSWER IS NOT OPTI
      1MUM. * *')
```

```

C      * * * * *
C
OPTM5560
OPTM5570
OPTM5580
OPTM5590
OPTM5600
OPTM5610
OPTM5620
OPTM5630
OPTM5640
OPTM5650
OPTM5660
OPTM5670
OPTM5680
OPTM5690
OPTM5700
OPTM5710
OPTM5720
OPTM5730
OPTM5740
OPTM5750
OPTM5760
OPTM5770
OPTM5780
OPTM5790
OPTM5800
OPTM5810
OPTM5820
OPTM5830
OPTM5840
OPTM5850
OPTM5860
OPTM5870
OPTM5880
OPTM5890
OPTM5900
OPTM5910
OPTM5920
OPTM5930
OPTM5940
OPTM5950
OPTM5960
OPTM5970
OPTM5980
OPTM5990
OPTM6000
OPTM6010
OPTM6020
OPTM6030
OPTM6040
OPTM6050
OPTM6060
OPTM6070
OPTM6080
OPTM6090
OPTM6100
OPTM6110
OPTM6120
OPTM6130
OPTM6140
OPTM6150
OPTM6160
OPTM6170
OPTM6180
OPTM6190
OPTM6200
OPTM6210
OPTM6220
OPTM6230
OPTM6240
OPTM6250

```

```

C      CASE IS TERMINATED, OPTIMUM FOUND. POINT XB, FB. OPTM6260
C      * * * * * OPTM6270
C      * * * * * OPTM6280
C      CALL EVAL LAST TIME TO PRINT OUT OPTIM RESULT IN TUBE, ECON. OPTM6290
7112 CALL EVAL(XB,FB,1,IER,CPM,REM,CPE) OPTM6300
C      IF(LIST1.EQ.0) WRITE(1,1409) REM,CPM,CPE OPTM6310
C      IF(LIST1.EQ.0) WRITE(1,100) K,L,FB,(XB(I),I=1,N) OPTM6320
100 FORMAT(3H0 *,I4,2X,I4,10X,E16.8/3H *,7E16.8/3H *, OPTM6330
17E16.8/3H *,7E16.8) OPTM6340
C      IF(LIST1.EQ.0) WRITE(1,47) OPTM6350
47 FORMAT(1H1,9HEXCELSIOR) OPTM6360
C      WRITE DATE, TIME FOR THIS CASE. OPTM6370
C      ITIME = ICLOCK(0) OPTM6380
C      ITO = (ITIME-ITIME0)/100 OPTM6390
C      IT1 = (ITIME-ITIME1)/100 OPTM6400
C      TO = DFLOAT(ITIME-ITIME0)/6000. DO OPTM6410
C      T1 = DFLOAT(ITIME-ITIME1)/6000. DO OPTM6420
C      WRITE (6,6C02) DATE,IT1,T1,ITO,TO OPTM6430
6002 FORMAT(7H1 DATE ,A8,11H. CASE TOOK,I5,' SEC. (=',F7.2,' MIN.) AL OPTM6440
1L CASES',I6,' SEC. (=',F7.2,' MIN.)' ) OPTM6450
C      ITIME1 = ITIME OPTM6460
C      IF(LOOPS.EQ.2 .AND. LOOP.EQ.1) GO TO 90 OPTM6470
C      * * * * * OPTM6480
C      * * * * * OPTM6490
C      * * * * * OPTM6500
C      8010 CONTINUE OPTM6510
C**** 8010 END N LOOP. NVARL,NVARU. **** GO TO 1492 FOR NEW CASE. **** OPTM6520
C      END OF LOOP 2 WITHIN LOOP 1. OPTM6530
C      GO TO 1492 OPTM6540
C      END OF LOOP 1. PROG QUITs BY READING END-OF-FILE. OPTM6550
C      END OPTM6560
C      END OPTM6570
C      END OPTM6580
C      END OPTM6590

```

```

SUBROUTINE LINCON(FP,X,XH,XL,KNT,N,IFLAG,CMM,FPC,FL,CPM,REM,CPE) LINC0010
C      IF IFLAG=1 ADJUST THE VARIABLES TO SATISFY THE CONSTRAINTS. LINC0020
C      IF IFLAG=2 SECOND PASS AT ADJUSTING VARIABLES LINC0030
C      IF IFLAG=3 ADJUST THE PARTIALS (VECTOR DIRECTION). LINC0040
C      FPC(I,J)=PARIAL OF CONSTRAINT J WITH RESPECT TO VARIABLE I LINC0050
C      N=NUMBER OF VARIABLES LINC0060
C      NLINEQ = 0 EQUALITY CONSTRAINTS. LINC0070
C      NNOTEQ = 2 INEQUALITY CONSTRAINTS (3 POSSIBLE). LINC0080
C      INEQUALITY CONSTRAINTS (NOW USING ONLY FIRST 2). LINC0090
C      CPM, PLIM MAX PRODUCT SALT CONC. LINC0100
C      REM, RELIM MIN REYNOLDS NUM ALLOWED. LINC0110
C      CPE, CFEEED PRODUCT CONC SHOULD NOT EXCEED FEED CONC. LINC0120
C      LINC0130
C      LINC0140
C      IMPLICIT REAL*8(A-H,O-Z) LINC0150
C      LINC0160
C      NAME OF COMMON TELLS WHICH ROUTINES USE IT.. 0=OPT=HYPPFIL, LINC0170
C      L=LINCON, E=EVAL, T=TUBE, C=ECON. LINC0180
C      COMMGN / CMOLTC / STEP,DATE,CFEEED,PLIM,RELIM, LIST1,LIST6,LINES LINC0190
C      COMMON / COMOL / R(20), KON(10),KFLG LINC0200
C      LINC0210
C      DIMENSION NLINLD(10), BSUMNT(10) LINC0220
C      DIMENSION FP(20),FPC(20,10),BSUM(10),AX(10,10) LINC0230
C      DIMENSION X(20),XH(20),XL(20),KNT(20) LINC0240
C      LINC0250
C      * * * * * LINC0260
C      * * * * * LINC0270
C      IF(LIST6.EQ.0 .AND. IFLAG.LT.3) WRITE(6,6001) LINC0280
6001 FORMAT(30X,1H ) LINC0290
C      IF(IFLAG.LT.3) LINES=LINES+1 LINC0300

```

```

ITER=0
NLINEQ=0
NNOTEQ=2
C 2 IS TOP OF LOOP FOR ADJUSTMENT OF VARIABLES.
2 CONTINUE
NLIN=NLINEQ
IF(IFLAG.EQ.3) GO TO 111
C CALL EVAL WHICH CALLS TUBE, ECON TO GET COST.
C ADJUSTMENT OF VARIABLES ONLY (IFLAG=1,2).
CALL EVAL(X ,FH,O,IER,CPM,REM,CPE)
C
KFLG=KFLG+1
FL=FH
IF(IER .NE. 1) GO TO 50
C QUIT IF IER NOT 1, ERROR IN TUBE.
9 CONTINUE
C
C LIST FPC(I,J) WITH EQUALITY CONSTRAINTS FOR LOW VALUES OF J AND
C CONTINUE J TO INCLUDE INEQUALITY CONST.
C ORDER BSUMNT FOR INEQ. CONST IN SAME ORDER
C THIS VERSION OF PROGRAM USES ONLY 2 CONSTRAINTS.
BSUMNT(1)=PLIM-CPM
BSUMNT(2)=REM-RELM
BSUMNT(3)=CFEED-CPE
DO 15 I=1,NNOTEQ
IF(BSUMNT(I))16,16,15
16 KON(I)=1
15 CONTINUE
111 CONTINUE
C ADJUSTMENT OF EITHER VARIABLES OR VECTOR DIRECTION.
IF(NLINEQ.EQ.0) GO TO 112
DO 12 I=1,NLINEQ
12 NLINLD(I)=1
112 CONTINUE
DO 13 I=1,NNOTEQ
IF(KON(I))13,13,14
14 NLIN=NLIN+1
NLINLD(NLIN)=I+NLINEQ
13 CONTINUE
IF(NLIN)50,50,3
3 CONTINUE
GO TO (21,21,4),IFLAG
4 CONTINUE
C 4 IS TOP OF LOOP FOR ADJUSTMENT OF PARTIALS, VECT DIRECTION.
C ADJUSTMENT OF VECTOR DIRECTION ONLY (IFLAG=3).
C
C IF PARTIAL OF I VARIABLE IS ZERO THEN PARTIAL OF ALL CONSTRAINTS WITH
C RESPECT TO THE I TH VARIABLE ARE SET TO ZERO
C
DO 10 I=1,N
IF(KNT(I))5,10,5
5 DO 6 J=1,NLIN
JT=NLINLD(J)
6 FPC(I,JT) = 0.000
10 CONTINUE
C
C COMPUTES THE VECTOR OF CONSTANTS
C
DO 400 J = 1,NLIN
BSUM(J) = 0.000
DO 400 I = 1,N
JT=NLINLD(J)
400 BSUM(J)=BSUM(J)+(FPC(I,JT)*FP(I))
DO 399 I = 1,NLIN
399 BSUM(I) = -BSUM(I)
GO TO 385
C ADJUSTMENT OF VARIABLES ONLY (IFLAG=1,2).
21 IF(NLINEQ)125,123,125

```

```

LINC0310
LINC0320
LINC0330
LINC0340
LINC0350
LINC0360
LINC0370
LINC0380
LINC0390
LINC0400
LINC0410
LINC0420
LINC0430
LINC0440
LINC0450
LINC0460
LINC0470
LINC0480
LINC0490
LINC0500
LINC0510
LINC0520
LINC0530
LINC0540
LINC0550
LINC0560
LINC0570
LINC0580
LINC0590
LINC0600
LINC0610
LINC0620
LINC0630
LINC0640
LINC0650
LINC0660
LINC0670
LINC0680
LINC0690
LINC0700
LINC0710
LINC0720
LINC0730
LINC0740
LINC0750
LINC0760
LINC0770
LINC0780
LINC0790
LINC0800
LINC0810
LINC0820
LINC0830
LINC0840
LINC0850
LINC0860
LINC0870
LINC0880
LINC0890
LINC0900
LINC0910
LINC0920
LINC0930
LINC0940
LINC0950
LINC0960
LINC0970
LINC0980
LINC0990

```



```

123 DO 124 I=1,NNOTEQ                                LINC1000
    IF(BSUMNT(I))125,124,124                          LINC1010
124 CONTINUE                                          LINC1020
    GO TO (125,50,50),IFLAG                          LINC1030
125 IF(ITER-4)20,28,20                              LINC1040
    20 CONTINUE                                       LINC1050
C                                                     LINC1060
C PUT VARIABLES ON THE BOUNDRIES IF OUT, SETS THOSE PARTIALS=C.0 LINC1070
C                                                     LINC1080
    DO 29 I=1,N                                       LINC1090
    IF(X(I)-XH(I))24,23,23                            LINC1100
    23 X(I)=XH(I)                                     LINC1110
    GO TO 26                                           LINC1120
    24 IF(XL(I)-X(I))29,25,25                        LINC1130
    25 X(I)=XL(I)                                     LINC1140
    26 DO 27 J=1,NLIN                                LINC1150
    JT=NLINLD(J)                                       LINC1160
    27 FPC(I,JT) = 0.0D0                             LINC1170
    29 CONTINUE                                       LINC1180
    28 CONTINUE                                       LINC1190
    ITER=ITER+1                                       LINC1200
C QUIT IF NO CONVERGENCE IN 10 PASSES. SET COST LARGE. LINC1210
    IF(ITER-10)330,330,384                            LINC1220
    384 FL = 101000. DO                               LINC1230
    IF(LIST6.EQ.0) WRITE(6,410)                       LINC1240
    IF(LIST1.EQ.0) WRITE(1,410)                       LINC1250
    410 FORMAT(50HODID NOT CONVERGE AFTER 10 ITERATIONS IN LINCON ) LINC1260
    LINES = LINES + 2                                LINC1270
    GO TO 50                                           LINC1280
    330 CONTINUE                                       LINC1290
    IF(NLIN-NLINEQ)333,333,331                        LINC1300
    331 ITEMP=NLINEQ+1                                LINC1310
    DO 332 I=ITEMP ,NLIN                             LINC1320
    K=NLINLD(I)-NLINEQ                               LINC1330
    332 BSUM(I)=-BSUMNT(K)                            LINC1340
    333 CONTINUE                                       LINC1350
C THIS VERSION OF PROGRAM USES ONLY 2 CONSTRAINTS.   LINC1360
C MUST BE WITHIN TOLERANCE BAND OF CONSTRAINT.      LINC1370
    IF(BSUMNT(1) + 1.0D-10) 385,501,501             LINC1380
    501 IF(BSUMNT(2) + 0.02D0 ) 385,502,502          LINC1390
    CC502 IF(BSUMNT(3) + 1.0D-9 ) 385,503,503         LINC1400
    502 CONTINUE                                       LINC1410
    503 CONTINUE                                       LINC1420
    GO TO (385,50,50),IFLAG                          LINC1430
    385 CONTINUE                                       LINC1440
C ADJUSTMENT OF EITHER VARIABLES OR VECTOR DIRECTION. LINC1450
C                                                     LINC1460
C COMPUTES THE A MATRIX                             LINC1470
C                                                     LINC1480
    DO 401 J = 1,NLIN                                LINC1490
    DO 401 L = 1,NLIN                                LINC1500
    AX(J,L) = 0.0D0                                   LINC1510
    DO 401 I = 1,N                                   LINC1520
    JT=NLINLD(J)                                       LINC1530
    LT=NLINLD(L)                                       LINC1540
    401 AX(J,L)=AX(J,L)+(FPC(I,LT)*FPC(I,JT))        LINC1550
C                                                     LINC1560
C IF A(I,I) IS ZERO THEN THE I TH COLUMN AND ROW ARE ZERO , SET A(I,I)=1 LINC1570
C                                                     LINC1580
    DO 390 I=1,NLIN                                  LINC1590
    IF(AX(I,I))390,389,390                            LINC1600
    389 AX(I,I) = 1.0D0                                LINC1610
    390 CONTINUE                                       LINC1620
C CALL AMATQ4, LINEAR SIMULT EQN SOLVING ROUTINE, IF NLIN GT 1. LINC1630
    IF(NLIN.LE.1) BSUM(1) = BSUM(1)/AX(1,1)          LINC1640
    IF(NLIN.GT.1)                                     LINC1650
    1 CALL AMATQ4(AX,BSUM,NLIN,1,10)                 LINC1660
C                                                     LINC1670
C THE SOLUTION VECTOR IS IN BSUM(I)                  LINC1680

```

```

C          GO TO (30,30,395),IFLAG
C          ADJUSTMENT OF VARIABLES ONLY (IFLAG=1,2).
C ADJUST THE X(I) TO SATISFY CONSTRAINTS
  30 DO 35 I=1,N
      DO 35 J=1,NLIN
      JT=NLINLD(J)
CC 35 X(I)=X(I)+BSUM(J)*FPC(I,JT)
  35 X(I)=X(I)+BSUM(J)*FPC(I,JT)*R(I)
      IFLAG=2
      DO 36 I=1,N
      X(I)=CMAX1(XL(I),X(I))
  36 X(I)=DMIN1(XH(I),X(I))
      GO TO 2
C          TRANSFER TO TOP OF VARIABLE ADJUSTMENT LOOP, 2.
395 CONTINUE
C          ADJUSTMENT OF VECTOR DIRECTION ONLY (IFLAG=3).
C
C THIS AREA CHECKS PARTIALS GOING OUT OF BOUNDS AND SETS=0 , HOLDS
C LARGEST ON BOUND
C
      FPMAX = 0.D0
      ISET=0
      DO 404 I = 1,N
      DO 405 J=1,NLIN
      JT=NLINLD(J)
      FP(I)=FP(I)+BSUM(J)*FPC(I,JT)
405 CONTINUE
      IF(FP(I)*CMM)41,404,45
  41 IF(X(I)-XL(I))49,49,404
  45 IF(XH(I)-X(I)-1.0D-9) 49,49,404
  49 IF(DABS(FP(I)) - FPMAX)48,48,46
  46 FPMAX = DABS(FP(I))
      ISET=I
  48 FP(I) = 0.D0
404 CONTINUE
      IF(LIST1.EQ.0) WRITE(1,100) (FP(I),I=1,N)
100 FORMAT(17HOPART. IN LINCON ,7E15.8/7E15.8/7E15.8)
      IF(ISET)50,50,47
  47 KNT(ISET)=KNT(ISET)+1
      GO TO 4
C          TRANSFER TO TOP OF VECTOR DIRECTION ADJUSTMENT LOOP, 4.
C
50 RETURN
END

```

LINC1690
 LINC1700
 LINC1710
 LINC1720
 LINC1730
 LINC1740
 LINC1750
 LINC1760
 LINC1770
 LINC1780
 LINC1790
 LINC1800
 LINC1810
 LINC1820
 LINC1830
 LINC1840
 LINC1850
 LINC1860
 LINC1870
 LINC1880
 LINC1890
 LINC1900
 LINC1910
 LINC1920
 LINC1930
 LINC1940
 LINC1950
 LINC1960
 LINC1970
 LINC1980
 LINC1990
 LINC2000
 LINC2010
 LINC2020
 LINC2030
 LINC2040
 LINC2050
 LINC2060
 LINC2070
 LINC2080
 LINC2090
 LINC2100
 LINC2110
 LINC2120
 LINC2130

```

C          SUBROUTINE AMATQ4(A,B,III,JJJ,ID)
C          LINEAR SIMULTANEOUS EQN SOLVING ROUTINE.
C          CALLED BY LINCON. REAL*8 PRECISION.
C          DETERMINANT D REMOVED, CHANGED TO SUBROUTINE. RMK 5/68.
C          AUTHORS R.G.EDWARDS AND R.E.FUNDERLIC.
C          COMPUTING TECHNOLOGY CENTER, K-25.
C          UNION CARBIDE NUCLEAR, OAK RIDGE, TENNESSEE.
C
C          IMPLICIT REAL*8(A-H,O-Z)
C          DIMENSION A(ID,1),B(ID,1)
C
      KK=III
      NV=IABS(JJJ)
      KKM=KK-1
      DO20I=1,KKM
      S=0.0D0
      DO2J=1,KK
      R=DABS(A(J,I))

```

MATQ0010
 MATQ0020
 MATQ0030
 MATQ0040
 MATQ0050
 MATQ0060
 MATQ0070
 MATQ0080
 MATQ0090
 MATQ0100
 MATQ0110
 MATQ0120
 MATQ0130
 MATQ0140
 MATQ0150
 MATQ0160
 MATQ0170
 MATQ0180
 MATQ0190

IF(R-S)2,1,1	MATQ0200
1 S=R	MATQC210
L=J	MATQ0220
2 CONTINUE	MATQ0230
IF(L-I)3,8,3	MATQ0240
3 DO4J=1, KK	MATQ0250
S=A(I, J)	MATQC260
A(I, J)=A(L, J)	MATQC270
4 A(L, J)=S	MATQ0280
IF(NV)8,8,5	MATQ0290
5 DO6J=1, NV	MATQ0300
S=8(I, J)	MATQ0310
B(I, J)=B(L, J)	MATQ0320
6 B(L, J)=S	MATQC330
8 IF(A(I, I))9,20,9	MATQ0340
9 IPO=I+1	MATQ0350
DO14J=IPO, KK	MATQ0360
IF(A(J, I))10,14,10	MATQ0370
10 S=A(J, I)/A(I, I)	MATQC380
A(J, I)=0.0D0	MATQ0390
DO 11 K=IPO, KK	MATQC400
11 A(J, K)=A(J, K)-A(I, K)*S	MATQC410
IF(NV)14,14,12	MATQ0420
12 DO13K=1, NV	MATQC430
13 B(J, K)=B(J, K)-B(I, K)*S	MATQ0440
14 CONTINUE	MATQ0450
20 CONTINUE	MATQC460
IF(NV)19,19,16	MATQ0470
16 KMO=KK-1	MATQ0480
DO18K=1, NV	MATQC490
B(KK, K)=B(KK, K)/A(KK, KK)	MATQ0500
DO18I=1, KMO	MATQ0510
N=KK-I	MATQC520
DO17J=N, KMO	MATQ0530
17 B(N, K)=B(N, K)-A(N, J+1)*B(J+1, K)	MATQ0540
18 B(N, K)=B(N, K)/A(N, N)	MATQ0550
19 RETURN	MATQ0560
END	MATQC570

SUBROUTINE EVAL(X, F, IOU, IBJ, CPM, REM, CPE)	EVAL0010
C	EVAL0020
C SUBROUTINE EVAL CALLED FROM MAIN PROG HYPFIL OR LINCON. SET VAR.	EVAL0030
C CALL TUBE WHICH CALLS ECON TO GET COST.	EVAL0040
C	EVAL0050
IMPLICIT REAL*8(A-H, O-Z)	EVAL0060
C NAME OF COMMON TELLS WHICH ROUTINES USE IT.. O=OPT=HYPFIL,	EVAL0070
C L=LINCON, E=EVAL, T=TUBE, C=ECON.	EVAL0080
COMMON / COMETC / POF, VIN, CR, TAPER, D, COST, REMIN, CPMAX, CPEND,	EVAL0090
1 NFLAG, IOUT	EVAL0100
DIMENSION X(20)	EVAL0110
C	EVAL0120
CPMAX = 0. DO	EVAL0130
REMIN = 0. DO	EVAL0140
CPEND = 0. DO	EVAL0150
IOUT=IOU	EVAL0160
POF=X(1)	EVAL0170
VIN=X(2)	EVAL0180
CR=X(3)	EVAL0190
TAPER=X(4)	EVAL0200
D=X(5)	EVAL0210
C	EVAL0220
CALL TUBE	EVAL0230
C	EVAL0240
F = COST	EVAL0250
IBJ=NFLAG	EVAL0260

```

CPM=CPMAX
REM=REMIN
CPE=CPEND
C
RETURN
END

```

```

EVAL0270
EVAL0280
EVAL0290
EVAL0300
EVAL0310
EVAL0320

```

SUBROUTINE TUBE

```

SUBROUTINE TUBE CALCS PHYSICAL CONDITIONS FOR GIVEN INLET
PRESSURE, VEL, CONC RATIO, TAPER, DIAM. CALLS ECON FOR COSTS.

```

```

SUBROUTINE TUBE AUTHORS - W.L.GRIFFITH, R.M.KELLER
UNION CARBIDE NUCLEAR DIV, 9704-2, Y-12 PLANT,
P.O.BOX Y, OAK RIDGE, TENNESSEE, 37830
LATEST VERSION OF PROGRAM - JULY 1969. FORTRAN IV, IBM-360.

```

```

IMPLICIT REAL*8(A-H,O-Z)

```

```

NAME OF COMMON TELLS WHICH ROUTINES USE IT.. O=OPT=HYPPFIL,
L=LINCON, E=EVAL, T=TUBE, C=ECON.
COMMON / COMOT / PERM,FRATIO,XRATIO, REJECT,TEMP,CONST,
1 A3,B3,C3,D3,XP,RA, CSTAR, CRA, MTYPE,ITUBE,N
COMMON / COMOTC / WP,CA,CE,CD,EP,ET, IBLEND
COMMON / COMETC / POF,VIN,CR,TAPER,D,COST,REMIN,CPMAX,CPEND,
1 NFLAG,IOUT
COMMON / COMTC / P(501),VEL(501),G(501),RHO(501),L(501), VRCUM,
1 EFFLEN,A,AX,DFTPI,DIFCP,DIFCPE,DIFRE, NPLUS1
COMMON / CMOLTC / STEP,DATE,CFEED,PLIM,RELIM, LIST1,LIST6,LINES

```

```

DOUBLE PRECISION MU,MUF,L,LOG10T,LOGT2,MUONE,MUTWO
DIMENSIONRE(501),E(501),CW(501),C(501),DL(501),W(501),REJ(501),
1 MU(501),PW(501),DP(501),DV(501),CPBAR(501),
2 B(501),CP(501),DB(501),PWBAR(501),CPROD(501)
3,RATE(501)
EQUIVALENCE (TEMP,T)

```

```

STATEMENTS FOR PROPERTIES.
CONVERT WT FRACTION TO GMOLES NACL/ KG SOLVENT H2O.
CMOLAL(Z)=CMWF*Z/(1.00-Z)
KELLOGG FIT DENSITY (LB/CU FT) Z= WT FRACTION.
RHO(Z)=(C1*Z+B1)*Z+A1
KELLOGG FIT VISCOSITY (LB/SEC FT) Z= WT FRACTION.
MUF(Z)=((C2*Z+B2)*Z+A2)
OSMOTIC PRESSURE (ATMOS). LOW NACL SALT CONC.
BETAF(Z)=CTAB*CMWF*Z/(1.00-Z)
FIRST DERIVATIVE OF OSMOTIC PRESSURE.
DBETA(Z) = CTAB*CMWF/(1.00-Z)/(1.00-Z)
SCHMIDT NUMBER **0.667
SC667(Z)=DEXP(AA+BB*Z)

```

```

*****
INITIALIZATION FOR LOOP 1 (I LOOP) AND LOOPS WITHIN LOOP 1 (SEE
DOCUMENTATION REPORT ON HYPPIL.) SET CONDITIONS FOR UPSTREAM
END OF FIRST TUBE INCREMENT (= TUBE INLET).
NFLAG=1
IF(CR.LE.1.000) GO TO 920
NPLUS1=N+1
C(1)=CFEED
DC = (C(1)*(CR-1.000)) /DFLOAT(N)
PONE=POF
VELONE=VIN
R = REJECT
FRTRAN = 1.000 - R

```

```

TUBE0010
TUBE0020
TUBE0030
TUBE0040
TUBE0050
TUBE0060
TUBE0070
TUBE0080
TUBE0090
TUBE0100
TUBE0110
TUBE0120
TUBE0130
TUBE0140
TUBE0150
TUBE0160
TUBE0170
TUBE0180
TUBE0190
TUBE0200
TUBE0210
TUBE0220
TUBE0230
TUBE0240
TUBE0250
TUBE0260
TUBE0270
TUBE0280
TUBE0290
TUBE0300
TUBE0310
TUBE0320
TUBE0330
TUBE0340
TUBE0350
TUBE0360
TUBE0370
TUBE0380
TUBE0390
TUBE0400
TUBE0410
TUBE0420
TUBE0430
TUBE0440
TUBE0450
TUBE0460
TUBE0470
TUBE0480
TUBE0490
TUBE0500
TUBE0510
TUBE0520
TUBE0530
TUBE0540
TUBE0550
TUBE0560
TUBE0570
TUBE0580

```

```

EONE=0.99 DO
CWONE = CFEED/(EONE*R + FRTRAN)
CPONE = FRTRAN*CWONE
C
DFT = D/12.00
C
AX, CROSS-SECTIONAL AREA (FT**2)
C
DFTPI, EFFECTIVE MEMBRANE PERIMETER (FT)
C
ITUBE 1=ROUND, 2=RECT.
C
GO TO (301,302),ITUBE
C
ROUND TUBES
301 AX = PI*DFT**2 /4.00
DFTPI=DFT*PI
GO TO 303
C
RECTANGULAR TUBES
C
ASSUME NO FLOW THRU NARROW WALL OF RECT TUBE (ITUBE=2).
302 A = DFT*(RA+1.00) / (2.00*RA)
DFTPI = 2.00*RA*A
AX=RA*A*A
303 IF(10UT.EQ. 0) GO TO 8
C
PRINT HEADING FOR FINAL OPTIMUM CONDIT. SOME INPUT VAR.
C
WRITE(6,1) DATE
1 FORMAT(72H1HYPERFILTRATION DESIGN OPTIMIZATION TUBULAR CHANNEL-
1TURBULENT FLOW ,43X,A8/ 16HINPUT VARIABLES )
WRITE (6,2) POF,PERM,VIN,CFEED,CR,PLIM,TAPER,RELIM,D,MTYPE
2 FORMAT(45HINLET PRESSURE, POF (ATM) ,F15.4,
1 48H MEMBRANE PERMEABILITY (KGAL/SQFT/ATM/DAY) ,F15.6/
2 45H INLET VELOCITY, VIN (FT/SEC) ,F15.4,
3 48H FEED CONCENTRATION, CFEED (WT FRACTION NACL),F15.6/
4 45H CONCENTRATION RATIO, CR (REJ CONC/FEED CONC),F15.4,
5 48H MAX SALT CONC OF PRODUCT, PLIM (WT FRACTION),F15.6/
6 45H TAPER ,F15.4,
7 48H MINIMUM REYNOLDS NUM ALLOWABLE, RELIM ,F15.0/
8 45H INSIDE TUBE DIAMETER, D, (4HR) (INCHES) ,F15.4,
9 48H MEMBRANE TYPE (1# ION EXCH) (2# POLY FIT) ,I15 )
WRITE (6,3) WP,CSTAR,T,CONST,IBLEND,A3,ITUBE,XP,RA,CA
3 FORMAT(45H PRODUCTION RATE (MILLIONS OF GAL/DAY) ,-6PF15.6,
1 48H ION-EXCH MEMB CAPACITY, CSTAR (MOLES/KG H2O),0PF15.6/
2 45H TEMPERATURE (F) ,F15.3,
3 48H CONST, REJ VS FLUX (SQFT DAY/KGAL) ,F15.6/
4 45H BLEND FEED WITH PRODUCT (1# NO) (2# YES) ,I15 ,
5 48H POLY FIT CONSTANT TERM, A3 ,F15.6/
6 45H TUBE SHAPE (1# ROUND) (2# RECTANGULAR) ,I15 ,
7 48H REYNOLDS NUM EXPONENT, XP ,F15.6/
8 45H ASPECT RATIO (RECT) (MEMB SIDE/SOLID SIDE) ,F15.6,
9 48H MEMB AREA UNIT COST, CA (CENTS/SQFT/DAY) ,F15.3)
WRITE (6,5) AX,CE,DFTPI,CD,FW,EP,FRATIO,ET,XRATIO
5 FORMAT(45H CROSS-SECTIONAL AREA (SQFT) ,F15.6,
1 48H ELEC POWER UNIT COST, CE (CENTS/KWH) ,F15.3/
2 45H EFFECTIVE MEMBRANE PERIMETER (FT) ,F15.6,
3 48H WASTE DISPOSAL UNIT COST, CD (CENTS/KGAL) ,F15.3/
4 45H FRACTION (MEMBRANE/WETTED PERIM) ,F15.6,
5 48H PUMP EFFICIENCY, EP ,F15.3/
6 45H FRATIO, 'ROUGHNESS FACTOR' ,F15.3,
7 48H RECOV TURBINE EFFICIENCY, ET ,F15.3/
8 45H XRATIO, MODIFY EXPONENT, TURB PROMOTERS ,F15.3/1H )TUBE1140
WRITE(6,4)
4 FORMAT( ' INCR PRES P-WALL VEL RE NUM LENGTH MIX-EFTTUBE1160
1F REJECT C-CORE C-ALPHA CPBAR RATE PROD CUM-CP' TUBE1170
2 / 8X,'(ATM) (ATM) (FT/SEC)',12X,4H(FT),22X,3(9H(WT-FR) ), TUBE1180
3 8X,'(LB/SEC) (WT-FR)' / 1H ) TUBE1190
C
TUBE1200
C
8 DL(1)= 10.35800*DC*VIN*AX*RHOF(CFEED) / PERM / DFTPI / CFEED /
1 (POF - BETAF(CFEED) ) TUBE1210
IF(DL(1).LT.1.D-10) DL(1) = 1.00 TUBE1220
CPROD(1)=0.0 DO TUBE1230
CP(NPLUS1) = 0.00 TUBE1240
CPEND = 0.000 TUBE1250
TUBE1260
TUBE1270

```



```

      DBDDCW = BDSTAR*CCSTAR*DCWCM / (CCW*(2.00*CCW*BDSTAR + CCSTAR))
      GO TO 65
64  BDSTAR=((D3*CWONE+C3)*CWONE+B3)*CWONE+A3
      DBDDCW = (3.00*D3*CWONE + 2.00*C3)*CWONE + B3
65  BETSIG=CONST*PWONE*PERM
      C      AVOID EXPONENTIAL OVERFLOW DUE TO BETSIG.
      IF(DABS(BETSIG) .GT. 174.600) GO TO 902
      EXPBS=DEXP(BETSIG)
      EXPFBS = EXPBS/(EXPBS-1.00)
      DYCBD = -EXPFBS/(1.00-BDSTAR)/(1.00-BDSTAR)
      DYDBS = EXPFBS*BDSTAR*BETSIG / (1.00-BDSTAR) / (EXPBS-1.00)
      DBSDPE= BETSIG/PWONE
      RES1=EXPZZZ-EONE
      RES2=BDSTAR/(1.00-BDSTAR)*EXPFBS - FRTRAN/R
      C11=DEDCW*DZDPF*DPFDCW
      C12=DEDCP*DZDPF*DPFDCP
      C21=DFRDCW+DYDBD*DBDDCW+DYDBS*DBSDPE*DPFDCW
      C22=DFRDCP+DYDBS*DBSDPE*DPFDCP
      IF(DABS(C11).GT.DABS(C21)) GO TO 50
      DCP = (RES1-C11*RES2/C21)/(C12-C22*C11/C21)
      DCW =(RES2-C22*DCP)/C21
      GO TO 51
50  DCP = (RES2-C21*RES1/C11)/(C22-C12*C21/C11)
      DCW =(RES1-C12*DCP)/C11
51  CWONE=CWONE+DCW
      CPONE=CPONE+DCP
      EONE=(CONE-CPONE)/(CWONE-CPONE)
      FRTRAN = CPONE/CWONE
      R = 1.000 - FRTRAN
      RONE = R
      BETA=BETAF(CWONE)-BETAF(CPONE)
      PWONE=PONE-BETA
      C      CHECK RES1, RES2 FOR CONVERGENCE OF CW, CP.
      C      IF(DABS(RES1).LT.DELTA1 .AND. DABS(RES2).LT.DELTA2) GO TO 12
      C
      C      14 IF(ITIMES.LT.5000) GO TO 10
      GO TO 903
      C      END OF LOOP 2. CWONE, CPONE CONVERGED. INIT FOR LOOP 3, LOOP 4.
      C
      C      WRITE NOTE, QUIT IF PW NEGATIVE
      C      12 IF(PWONE .LE. 0.000) GO TO 901
      C      QUIT IF E IS NEGATIVE OR GREATER THAN ONE
      C      IF(EONE.LE.0.000 .OR. EONE.GT.1.000) GO TO 904
      C
      16 CTWO=C(I)
      CWTWO=CWONE
      CPTWO=CPONE
      CPBAR(I-1)=CPONE
      PWBAR(I-1)=PWONE
      ETWO=EONE
      RHOTWO= RHOF(CTWO)
      MUTWO= MUF(CTWO)
      PLAST=PONE
      RLAST=REONE
      DLAST=DL(I-1)
      NTIMES=0
      C
      C      BEGIN LOOP 3 (NTIMES LOOP) WITHIN LOOP 1 (I LOOP). ITERATE UNTIL
      C      CONVERGENCE OF DOWNSTREAM END PTWO, RETWO (REYNOLDS NUM),
      C      DL (TUBE INCR LENGTH), CWTWO, AND CPTWO. INIT FOR LOOP 4
      C      (CWTWO, CPTWO) WITHIN LOOP 3.
      100 NTIMES=NTIMES+1
      DB(I-1) = 0.0965200*DFTPI*PERM*PWBAR(I-1)*DL(I-1) /
      1  (1.00-CPBAR(I-1))
      BTWO=BONE-DB(I-1)
      GTWO=BTWO/AX
      DS=CPBAR(I-1)*DB(I-1)

```

TUBE1980
 TUBE1990
 TUBE2000
 TUBE2010
 TUBE2020
 TUBE2030
 TUBE2040
 TUBE2050
 TUBE2060
 TUBE2070
 TUBE2080
 TUBE2090
 TUBE2100
 TUBE2110
 TUBE2120
 TUBE2130
 TUBE2140
 TUBE2150
 TUBE2160
 TUBE2170
 TUBE2180
 TUBE2190
 TUBE2200
 TUBE2210
 TUBE2220
 TUBE2230
 TUBE2240
 TUBE2250
 TUBE2260
 TUBE2270
 TUBE2280
 TUBE2290
 TUBE2300
 TUBE2310
 TUBE2320
 TUBE2330
 TUBE2340
 TUBE2350
 TUBE2360
 TUBE2370
 TUBE2380
 TUBE2390
 TUBE2400
 TUBE2410
 TUBE2420
 TUBE2430
 TUBE2440
 TUBE2450
 TUBE2460
 TUBE2470
 TUBE2480
 TUBE2490
 TUBE2500
 TUBE2510
 TUBE2520
 TUBE2530
 TUBE2540
 TUBE2550
 TUBE2560
 TUBE2570
 TUBE2580
 TUBE2590
 TUBE2600
 TUBE2610
 TUBE2620
 TUBE2630
 TUBE2640
 TUBE2650
 TUBE2660
 TUBE2670

```

VELTWO=GTWO/RHOTWO
IF(VELTWO.LE.0.0D0) GO TO 905
DV(I-1)=VELTWO-VELONE
RETWO=DFT/MUTWO*GTWO
RE25=RETWO**0.25
GO TO (321,322),JTYPE
321 REXP=RETWO**XP
GO TO 323
322 REXP=RE25
323 FTWO=FF/RE25
VELBAR = 0.5D0*(VELONE+VELTWO)
FBAR = 0.5D0*(FONE+FTWO)
RHOBAR = 0.5D0*(RHOONE+RHOTWO)
DP(I-1) = RHOBAR * (-VELBAR*DV(I-1) + 2.D0*FBAR*DL(I-1)*VELBAR*
1 VELBAR/DFT) / GC / 2116.8D0
C PRESSURE IN ATM 144*14.7=2116.8
PTWO=PONE-DP(I-1)
PWTWO=PTWO-BETA
IF(PTWO .LE. 0.0D0) GO TO 906
ZZ=V*REXP/VELTWO*SC667(CTWO)
ITIMES=0
C
C
C BEGIN LOOP 4 (ITIMES LOOP) WITHIN LOOP 3 WITHIN LOOP 1. ITERATE
C UNTIL DOWNSTREAM CW, CP (CWTWO, CPTWO) (RES1, RES2) CONVERGE.
C MOST TIME SPENT IN LOOP 4 (AND LOOP 2 IF TAPER NOT ZERO). USE
C UNSUBSCRIPTED VAR TO SAVE TIME.
120 ITIMES=ITIMES+1
ZZZ=ZZ*PWTWO
C AVOID EXPONENTIAL OVERFLOW DUE TO LOW VEL.
IF(DABS(ZZZ).GT. 174.6D0) GO TO 902
EXPZZZ=DEXP(ZZZ)
C CALC PARTIAL DERIVATIVES.
DZDPF = -ZZZ*EXPZZZ
DPFDCP=DBETA(CPTWO)
DRHDCW=TWOC1*CWTWO+B1
DPFDCW=-DBETA(CWTWO)
CWMCP=(CWTWO-CPTWO)
CWMCP2=CWMCP*CWMCP
DEDCW=-ETWO/CWMCP
DEDCP=(CTWO-CWTWO)/CWMCP2
DFRDCW=-CPTWO/CWMCP2
DFRDCP=CWTWO/CWMCP2
GO TO (72,74),MTYPE
72 CCW=CMOLAL(CWTWO)
CSTAR=CSTAR/CCW
BDSTAR = 0.5D0*(-CSTAR + DSQRT(CSTAR*CSTAR + 4.D0) )
DCWCM = CMWF/(1.D0-CWTWO)/(1.D0-CWTWO)
DBDDCW = BDSTAR*CCW*DCWCM / (CCW*(2.D0*CCW*BDSTAR + CSTAR))
GO TO 75
74 BDSTAR=((D3*CWTWO+C3)*CWTWO+B3)*CWTWO+A3
DBDDCW = (3.D0*D3*CWTWO + 2.D0*C3)*CWTWO + B3
75 BETSIG=CONST*PWTWO*PERM
C AVOID EXPONENTIAL OVERFLOW DUE TO BETSIG.
IF(DABS(BETSIG) .GT. 174.6D0) GO TO 902
EXPBS=DEXP(BETSIG)
EXPFB = EXPBS/(EXPBS-1.D0)
DYDBD = -EXPFB/(1.D0-BDSTAR)/(1.D0-BDSTAR)
DYDBS = EXPFB*BDSTAR*BETSIG / (1.D0-BDSTAR) / (EXPBS-1.D0)
DBSDPE= BETSIG/PWTWO
RES1=EXPZZZ-ETWO
RES2=BDSTAR/(1.D0-BDSTAR)*EXPFB - FRTRAN/R
C11=DEDCW*DZDPF*DPFDCW
C12=DEDCP*DZDPF*DPFDCP
C21=DFRDCW*DYDBD*DBDDCW*DYDBS*DBSDPE*DPFDCW
C22=DFRDCP*DYDBS*DBSDPE*DPFDCP
IF(DABS(C11).GT.DABS(C21)) GO TO 9119
DCP = (RES1-C11*RES2/C21)/(C12-C22*C11/C21)
DCW =(RES2-C22*DCP)/C21
GO TO 111
TUBE2680
TUBE2690
TUBE2700
TUBE2710
TUBE2720
TUBE2730
TUBE2740
TUBE2750
TUBE2760
TUBE2770
TUBE2780
TUBE2790
TUBE2800
TUBE2810
TUBE2820
TUBE2830
TUBE2840
TUBE2850
TUBE2860
TUBE2870
TUBE2880
TUBE2890
TUBE2900
TUBE2910
TUBE2920
TUBE2930
TUBE2940
TUBE2950
TUBE2960
TUBE2970
TUBE2980
TUBE2990
TUBE3000
TUBE3010
TUBE3020
TUBE3030
TUBE3040
TUBE3050
TUBE3060
TUBE3070
TUBE3080
TUBE3090
TUBE3100
TUBE3110
TUBE3120
TUBE3130
TUBE3140
TUBE3150
TUBE3160
TUBE3170
TUBE3180
TUBE3190
TUBE3200
TUBE3210
TUBE3220
TUBE3230
TUBE3240
TUBE3250
TUBE3260
TUBE3270
TUBE3280
TUBE3290
TUBE3300
TUBE3310
TUBE3320
TUBE3330
TUBE3340
TUBE3350
TUBE3360
TUBE3370

```



```

9119 DCP = (RES2-C21*RES1/C11)/(C22-C12*C21/C11)
      DCW =(RES1-C12*DCP)/C11
111 CWTWO=CWTWO+DCW
      CPTWO=CPTWO+DCP
      ETWO=(CTWO-CPTWO)/(CWTWO-CPTWO)
      FRTRAN = CPTWO/CWTWO
      R = 1.000 - FRTRAN
      RTWO = R
      BETA=BETAF(CWTWO)-BETAF(CPTWO)
      PWTWO=PTWO-BETA
C     CHECK RES1, RES2 FOR CONVERGENCE OF CW, CP.
C     CONVERGENCE CRITERIA FOR CWTWO, CPTWO (RES1, RES2) LOOSER IN
C       LOOP 4 (WITHIN LOOP 3) THAN IN LOOP 3.
C
      IF(DABS(RES1).LT.1.0D-3 .AND. DABS(RES2).LT.0.5D-3) GO TO 125
C
      IF(ITIMES.LE.5000) GO TO 120
      GO TO 903
C     END LOOP 4 WITHIN LOOP 3. DOWNSTREAM CW, CP (CWTWO, CPTWO) CONV.
C
C     QUIT IF PW LT 0.
C 125 IF(PWTWO.LT.0.0D0) GO TO 901
C     QUIT IF E LT 0 OR GT 1
      IF(ETWO.LE.0.0D0 .OR. ETWO.GT.1.0D0) GO TO 904
      CPBAR(I-1) = 0.5D0*(CPONE+CPTWO)
      PWBAR(I-1) = 0.5D0*(PWONE+PWTWO)
C     24*3600*7.48/(PI*62.4*1000)=3.2967
      DL(I-1) = DC*BONE*10.358 D0 / (C(I)-CPBAR(I-1)) / PERM /
1     PWBAR(I-1) / DFTPI
      IF(NTIMES.GT.5000) GO TO 903
C
C     CONVERGENCE CRITERIA FOR CWTWO, CPTWO (RES1, RES2) TIGHTER IN
C     LOOP 3 THAN IN LOOP 4 (WITHIN LOOP 3).
C     CHECK P, RE, AND DL FOR CONVERGENCE, BACK TO 100 IF NOT
      IF(DELTA1.LT.DABS(RES1) .OR. DELTA2.LT.DABS(RES2)) GO TO 119
      IF(DABS((PTWO-PLAST)/DP(I-1)) .GT. 0.01D0) GO TO 119
      IF(DABS((DL(I-1)-DLAST)/DL(I-1)) .GE. 1.0D-4) GO TO 119
      IF(DABS((RETWO-RLAST)/RETWO) .LE. 0.01D0) GO TO 130
119 PLAST=PTWO
      RLAST=RETWO
      DLAST=DL(I-1)
      GO TO 100
C     END OF LOOP 3 WITHIN LOOP 1. ALL DOWNSTREAM CONDITIONS CONVERGED.
C
C 130 J=I-1
C     CALC DB, D, G, OTHER VAR (VEL) CLOSE ENOUGH.
      DB(I-1) = 0.09652D0*DFTPI*PERM*PWBAR(I-1)*DL(I-1) /
1     (1.0D0-CPBAR(I-1))
      BTWO=BONE-DB(I-1)
      GTWO=BTWO/AX
      RATE(I-1) = DB(I-1)*10357.00 / DL(I-1) / DFTPI
      VRATIO = (VELONE-VELTWO)/VELTWO*TAPER + 1.0D0
      EFFLEN=EFFLEN*VRATIO + DL(I-1)
      DS=CPBAR(I-1)*DB(I-1)/VRCUM
      S=S+DS
      DW=DB(I-1)/VRCUM-DS
      W(I)=W(I-1)+DW
      CPROD(I)=S/(S+W(I))
      L(I)=L(I-1)+DL(I-1)
      VRCUM=VRCUM*VRATIO
C     STORE VARIABLES FOR UPSTREAM END OF TUBE INCR.
      CW(I-1)=CWONE
      CP(I-1)=CPONE
      P(I-1)=PONE
      PW(I-1)=PWONE
      VEL(I-1)=VELONE
      REJ(I-1)=RONE
      E(I-1)=EONE

```

TUBE3380
TUBE3390
TUBE3400
TUBE3410
TUBE3420
TUBE3430
TUBE3440
TUBE3450
TUBE3460
TUBE3470
TUBE3480
TUBE3490
TUBE3500
TUBE3510
TUBE3520
TUBE3530
TUBE3540
TUBE3550
TUBE3560
TUBE3570
TUBE3580
TUBE3590
TUBE3600
TUBE3610
TUBE3620
TUBE3630
TUBE3640
TUBE3650
TUBE3660
TUBE3670
TUBE3680
TUBE3690
TUBE3700
TUBE3710
TUBE3720
TUBE3730
TUBE3740
TUBE3750
TUBE3760
TUBE3770
TUBE3780
TUBE3790
TUBE3800
TUBE3810
TUBE3820
TUBE3830
TUBE3840
TUBE3850
TUBE3860
TUBE3870
TUBE3880
TUBE3890
TUBE3900
TUBE3910
TUBE3920
TUBE3930
TUBE3940
TUBE3950
TUBE3960
TUBE3970
TUBE3980
TUBE3990
TUBE4000
TUBE4010
TUBE4020
TUBE4030
TUBE4040
TUBE4050
TUBE4060
TUBE4070

```

RE(I-1)=REONE
RHO(I-1)=RHONE
MU(I-1)=MUONE
B(I-1)=BONE
G(I-1)=GONE
REMIN = DMIN1(REMIN,RFTWO)
CPEND = DMAX1(CPEND,CPTWO)
C
C
C      INITIALIZATION FOR NEXT INCREMENT, TAPER=0 OR NOT 0.
      VELONE=VELTWO*VRATIO
      IF(I.LE.2) DL(I) = DL(I-1)
      IF(I.GT.2) DL(I) = DL(I-1) + (DL(I-1)-DL(I-2))
      IF(I.LE.2) EONE = E(I-1)
      IF(I.GT.2) EONE = E(I-1) + E(I-1)-E(I-2)
      PONE=PTWO
      CWGNE=CWTWO
      CPONE=CPTWO
C
C      IF(TAPER.NE. 0.DO) GO TO 170
C      INITIALIZATION FOR TAPER=0. SET UPSTREAM VAR FOR NEXT INCR (PONE)
C      = DOWNSTREAM VALUES (PTWO) FOR THIS INCR. PONE, CPONE, CWONE,
C      VELONE OK. FIX EONE. CONE, MUONE, RONE, GONE ARE RESET.
      CONE = CTWO
      PWGNE = PWTWO
      RHOONE = RHOTWO
      MUONE = MUTWO
      REONE = RETWO
      EONE = ETWO
      RONE = RTWO
      BONE = BTWO
      GONE = GTWO
      FONE = FTWO
C
C      170 IF(IOUT.EQ. 0) GO TO 180
      WRITE (6,6C03) J,P(J),PW(J),VEL(J),RE(J),L(J),E(J),REJ(J),C(J),
      1CW(J),CPBAR(J),RATE(J),W(J),CPROD(J)
6C03  FORMAT(1H ,I4,3F9.3,F9.0,F9.2,5F9.6,F9.3,F9.5,F9.6)
C      CHECK FOR SOLUBILITY AT WALL AND PRODUCT SALT CONCENTRATION.
C      180 IF(CWTWO.GT. SOLIM ) GO TO 200
C      190 CONTINUE
C
C      * * * * *
C      * * 190 END I LOOP. N CONC INCREMENTS. * * *
C      END LOOP 1.
C
C      * * * * *
C      STORE VALUES FOR END OF N-TH INCREMENT (= TUBE EXIT).
      CP(NPLUS1)=CPTWO
      CW(NPLUS1)=CWTWO
      P(NPLUS1)=PTWO
      PW(NPLUS1)=PWTWO
      VEL(NPLUS1)=VELTWO
      RE(NPLUS1)=RETWO
      E(NPLUS1)=ETWO
      REJ(NPLUS1)=RTWO
      RHO(NPLUS1)=RHOTWO
      MU(NPLUS1)=MUTWO
      B(NPLUS1)=BTWO
      G(NPLUS1)=GTWO
      VRCUM=VRCUM/VRATIO
      CPMAX=CPROD(NPLUS1)
      DIFCP = PLIM - CPMAX
      DIFRE = REMIN - RELIM
      DIFCPE = CFEEED - CPEND
      J=NPLUS1
      EXITS=(B(1)*CFEEED-S)*VRCUM/B(NPLUS1)
      ERRDC=C(NPLUS1)-EXITS

```

```

        IF(DABS(ERRDC)/C(NPLUS1) .LE. 0.01D0) GO TO 2G9
6006 FORMAT(33H0* MATERIAL BALANCE NOT SATISFIED ,5E15.7)
        WRITE(6,6006) EXITS,ERRDC,VRCUM,EFFLEN
        LINES=LINES+2
209 IF(IOUT.EQ. 0) GO TO 211
        WRITE (6,6003) J,P(J),PW(J),VEL(J),RE(J),L(J),E(J),REJ(J),C(J),
        1CW(J),CPBAR(J),RATE(J),W(J),CPROD(J)
C
C      CALL ECON TO GET TOTAL COST AND LIST.
211 CALL ECON
C
C      RETURN
C
C      * * * * *
C * * SECTION TO HANDLE ERRORS.      * * *
C PROGRAM NOT TERMINATED * * LINCON HANDLES SITUATION.
C
C      * * * * *
200 NFLAG=8
        IF(LIST6.EQ.0) WRITE(6,6009)
6009 FORMAT( 'OSOLUBILITY LIMIT EXCEEDED.' )
        LINES=LINES+2
        NPLUS1=1
        CPBAR(NPLUS1) = 0.D0
        COST = 1.0D5 * CR
        GO TO 925
C
901 IF(LIST6.EQ.0) WRITE(6,1901) PWONE,PWTWO
1901 FORMAT( 7H0PW(I)=,2E15.7,20H IS ZERO OR NEGATIVE )
        NFLAG=3
        GO TO 920
C
902 IF(LIST6.EQ.0) WRITE(6,1902) VELONE,VELTWO,BETSIG
1902 FORMAT( 'OPREVENTED EXPONENTIAL OVERFLOW. VELONE=', D14.6,
        1 ' ', VELTWO=' ', D14.6, ' ', BETSIG=' ', D14.6 )
        NFLAG=5
        GO TO 920
C
903 IF(LIST6.EQ.0) WRITE(6,1903) I,ITIMES,NTIMES
1903 FORMAT(22H0DID NOT CONVERGE I=,I4,9H ITIMES=,I4,9H NTIMES=,I4)
        WRITE(6,6005) CWTWO,CPTWO, BETA,PWTWO,ZZZ,EXPZZZ,
        1 DZDPF,DPFDCP,DRHDCW,DPFDCW,CWMCP,DEDCW,DEDCP,DFRDCW,DFRDCP,
        2 CCW,BDSTAR,DBDDCW,BETSIG,EXPBS,DYDBD,DYDBS,DBSDPE,C11,C12,C21,C22
        3,RES1,RES2, DCW,DCP, R ,EONE
6005 FORMAT(1H0,8E15.7)
        LINES = LINES + 10
        NFLAG=6
        GO TO 920
C
904 IF(LIST6.EQ.0) WRITE(6,1904) EONE,ETWO,I
1904 FORMAT( 6H0E(I)=,2E15.7,33H IS NEGATIVE OR GREATER THAN ONE ,I6)
        NFLAG=7
        GO TO 920
C
905 IF(LIST6.EQ.0) WRITE(6,1905) VELTWO
1905 FORMAT( 8H0VEL(I)=,E14.7,12H IS NEGATIVE )
        NFLAG=4
        GO TO 920
C
906 IF(LIST6.EQ.0) WRITE(6,1906) PTWO
1906 FORMAT( 6H0P(I)=,E14.7,20H IS ZERO OR NEGATIVE )
        NFLAG=2
C
920 IF(NFLAG.EQ. 1) NFLAG= 9
        LINES=LINES+2
C      IN CASE OF ERROR SET COST= LARGENUMBER FOR OPTIM ROUTINE.
C
C      COST = 1.01 D5
925 IF(LIST6.EQ.0 .AND. IOUT.EQ.0) WRITE(6,6008) STEP,POF,VIN,

```

```

1 CR,TAPER,D, COST
6008 FORMAT( 1H0,A1,F9.3,F8.3,F7.3,F7.4,F8.4, 69X,F20.4)
      LINES = LINES + 2
      RETURN
C
C      * * * * *
C
C * * ENTRY SETUP * * SEPARATE SECTION CALLED ONLY BY MAIN PROG HYPFIL
C      INITIALIZE VARIABLES USED BY TUBE BEFORE EACH CASE.
C
C      * * * * *
C
C      ENTRY SETUP
C
C      ZERO ALL 23 SUBSCRIPTED VAR STORAGE ARRAYS,MAX 500 INCR.
C
      DO 35 I=1,501
      C(I) = 0.000
      CW(I) = 0.000
      CP(I) = 0.000
      CPBAR(I) = 0.000
      CPROD(I) = 0.000
      P(I) = 0.000
      DP(I) = 0.000
      PW(I) = 0.000
      PWBAR(I) = 0.000
      VEL(I) = 0.000
      DV(I) = 0.000
      L(I) = 0.000
      DL(I) = 0.000
      RHO(I) = 0.000
      MU(I) = 0.000
      RE(I) = 0.000
      E(I) = 0.000
      REJ(I) = 0.000
      B(I) = 0.000
      DB(I) = 0.000
      G(I) = 0.000
      W(I) = 0.000
35 RATE(I) = 0.000
C
      PI = 3.14159265358979 DO
      GC=32.2 DO
C      CMWF (GMOLE NACL/KG NACL). FF FOR BLASIVUS EQN.
      CMWF = 1000.DO/58.45DO
      FF = 0.08DO*FRATIO
C
C      ITUBE 1=ROUND, 2=RECT. CALC FRACT WETTED PERIM, FW. RATIO RA.
C      ASSUME NO FLOW THRU NARROW WALL OF RECT TUBE (ITUBE=2).
      FW = RA/(RA+1.0DO)
      IF(ITUBE.EQ.1) FW = 1.0DO
C
C      SET JTYPE=1, OR 2 IF XP=0.25 (RE NUM EXPON).
      JTYPE=1
      IF(DABS(XP-.25DO).LT. 1.D-7) JTYPE=2
C      SALT SOLUBILITY LIMIT AT MEMBRANE (WT FRACTION).
      SOLIM = 0.26 DO
C
C      T = TEMP EQUIV.
      T2=T*T
      LOG10T=DLOG10(T)
      LOGT2=LOG10T**2
C
C      TEMP DEPENDENT COEFFICIENTS FOR FIT OF SCHMIDT NUM**0.667, SC667.
      AA = 5.637387 DO - 2.045509 D-2*T + 0.29927225D-4*T2
      BB = 0.8673596DO - 0.7184158D-3*T + 0.8589322 D-5*T2
C
C      SET TEMP DEP COEFFS FOR KELLOGG DENSITY FIT, RHOF.
      A1 = 62.717753DO - 0.32152986D-2*T - 0.48932777D-4*T2
      B1 = 44.315006DO - 0.11647394D-1*T + 0.34231326D-5*T2

```

TUBE5480
TUBE5490
TUBE5500
TUBE5510
TUBE5520
TUBE5530
TUBE5540
TUBE5550
TUBE5560
TUBE5570
TUBE5580
TUBE5590
TUBE5600
TUBE5610
TUBE5620
TUBE5630
TUBE5640
TUBE5650
TUBE5660
TUBE5670
TUBE5680
TUBE5690
TUBE5700
TUBE5710
TUBE5720
TUBE5730
TUBE5740
TUBE5750
TUBE5760
TUBE5770
TUBE5780
TUBE5790
TUBE5800
TUBE5810
TUBE5820
TUBE5830
TUBE5840
TUBE5850
TUBE5860
TUBE5870
TUBE5880
TUBE5890
TUBE5900
TUBE5910
TUBE5920
TUBE5930
TUBE5940
TUBE5950
TUBE5960
TUBE5970
TUBE5980
TUBE5990
TUBE6000
TUBE6010
TUBE6020
TUBE6030
TUBE6040
TUBE6050
TUBE6060
TUBE6070
TUBE6080
TUBE6090
TUBE6100
TUBE6110
TUBE6120
TUBE6130
TUBE6140
TUBE6150
TUBE6160
TUBE6170

```

C1 = 16.449945D0 + 0.13761984D-1*T - 0.92275301D-4*T2
TWOC1 = 2.D0*C1
C DEN77 IS RHOF DENSITY FIT FOR H2O AT 77F AND ZERO NACL CONC.
C FOR USE IN CTAB, BETAF.
C DEN77 = 62.1800527 D0
C
C CTAB (ATM KG SOLVENT/GMOLE NACL) FOR USE IN BETAF, OSMOTIC
C PRESSURE FIT FOR LOW NACL CONC. 2*24.4 AT 77F, H2O.
C CTAB = 48.8D0*(TEMP+460.D0)/537.D0 *RHOF(0.D0)/DEN77
C
C TEMP DEP COEFFS FOR KELLOGG VISCOSITY FIT, MUF.
C A2=( 22.216306D0- 16.460588D0*LOG10T+ 3.0949585D0*LOGT2)/3600.D0
C B2=(-14.161033D0+ 16.430566D0*LOG10T- 4.205263 D0*LOGT2)/3600.D0
C C2=(284.84684 D0-219.40291 D0*LOG10T+42.287729 D0*LOGT2)/3600.D0
C
C UNUSED COEFFS FOR OSMOTIC PRESSURE FIT, BETAF, FOR SEA WATER.
CC A0=.602699E+3+.135850E+1*T-.145757E-2*T2
CC B0=.272902E+3+.714610E+1*T-.196794E-1*T2
CC C0=.186395E+4+.319417E+2*T-.460908E-1*T2
CC D0=.176486E+5-.133310E+3*T+.249592E+0*T2
C UNUSED ARITH STATEMENT FCT FOR SEA WATER FOR BETAF, DBETA.
CC BETAF(Z)=((D0*Z+C0)*Z+B0)*Z+A0)*Z
CC DBETA(Z)=(((W4*Z+W3)*Z+W2)*Z+W1)
C
C RETURN
C
C * * * * *
C ENTRY ADJUST REDUCES CR TO ACCEPTABLE VALUE.
C
C * * * * *
C ENTRY ADJUST
C
C IGOOD = 0
C NN = MINO(NPLUS1,I)
C CHECK PRODUCT LIMIT FIRST. USE HIGHEST GOOD INCR.
C DO 610 I=1,NN
C J = NN-I+1
C IF(CPROD(J).LE.PLIM) GO TO 615
610 CONTINUE
C IGOOD = 0
C GO TO 630
615 IGOOD = J-1
C NN = J
C II = 0
C CHECK REYNOLDS NUM. ALLOW NO VIOLATIONS.
C DO 620 I=1,NN
C IF(II.LT.(I-1)) GO TO 620
C IF(RE(I).GE.RELIM) II = I
620 CONTINUE
C IGOOD = MAX0(0,II-1)
630 CRA = DFLOAT(IGOOD)/DFLOAT(N)*(CR-1.0D0) + 1.0D0
C WRITE (6,6C07) CR,CRA
6007 FORMAT( 1H0/1H0/1H0/1H0/1H0/1H0/1H0/ '0 * * * CR HAS BEEN ADJUSTED
1 FROM',F9.4,' TO',F9.4, ' * * * * * /
2 1H0/1H0/1H0/1H0/1H0/1H0/1H0/1H0 )
C LINES = LINES + 32
C
C RETURN
C END

```

TUBE6180
TUBE6190
TUBE6200
TUBE6210
TUBE6220
TUBE6230
TUBE6240
TUBE6250
TUBE6260
TUBE6270
TUBE6280
TUBE6290
TUBE6300
TUBE6310
TUBE6320
TUBE6330
TUBE6340
TUBE6350
TUBE6360
TUBE6370
TUBE6380
TUBE6390
TUBE6400
TUBE6410
TUBE6420
TUBE6430
TUBE6440
TUBE6450
TUBE6460
TUBE6470
TUBE6480
TUBE6490
TUBE6500
TUBE6510
TUBE6520
TUBE6530
TUBE6540
TUBE6550
TUBE6560
TUBE6570
TUBE6580
TUBE6590
TUBE6600
TUBE6610
TUBE6620
TUBE6630
TUBE6640
TUBE6650
TUBE6660
TUBE6670
TUBE6680
TUBE6690
TUBE6700
TUBE6710
TUBE6720
TUBE6730
TUBE6740
TUBE6750
TUBE6760
TUBE6770
TUBE6780

ECON0010
ECON0020
ECON0030
ECON0040

```

lc      IMPLICIT REAL*8(A-H,O-Z)
C
C      NAME OF COMMON TELLS WHICH ROUTINES USE IT.. Q=OPT=HYPFIL,
C      L=LINCON, E=EVAL, T=TUBE, C=ECON.
C      COMMON / COMOTC / WP,CA,CE,CD,EP,ET, IBLEND
COMMON / COMETC / POF,VIN,CR,TAPER,D,COST,REMIN,CPMAX,CPEND,
1 NFLAG,IOUT
COMMON / COMTC / P(501),VEL(501),G(501),RHO(501),L(501), VRCUM,
1 EFFLEN,A,AX,DFTPI,DIFCP,DIFCPE,DIFRE, NPLUS1
COMMON / CMOLTC / STEP,DATE,CFEED,PLIM,RELIM, LIST1,LIST6,LINES
C
C      REAL*8          NP,NT,L
C
C      * * * * *
C
C      CALC PRODUCT FLOW (LB/SEC)
C      24*3600*7.48/62.4=10358.
C      WPM=WP/10358. DO
C      CALC NUMBER OF TUBES
C      4*144/PI=183.3465
C      D2=D*D
C      NT=WPM/AX/(G(1)*VRCUM-G(NPLUS1))
C      CALC FEED FLOW
C      WFM=G(1)*NT*VRCUM*D2/183.3465
C      WFM=G(1)*NT*VRCUM*AX
C      24*3600*7.48=646317
C      WF=WFM*646317.DO/RHO(1)
C      CALC REJECT FLOW
C      WRM=G(NPLUS1)*NT*D2/183.3465
C      WRM=G(NPLUS1)*NT*AX
C      WR=WRM*646317.DO/RHO(NPLUS1)
C
C      CALC PUMP POWER (KW)
C      .746*14.7*144/550=2.8711
C      0.746/(2*32.2*550)=2.11E-5
C      PP=WFM*(P(1)*2.8711/RHO(1)+VEL(1)**2*2.11E-5)/EP
C      CALC TURBINE POWER (KW)
C      TP=WRM*(P(NPLUS1)/RHO(NPLUS1)*2.8711+VEL(NPLUS1)**2*2.11E-5)*ET
C      CALC NET POWER (KW)
C      NP=PP-TP
C      CALC POWER/KGAL (KWH/KGAL)
C      UNP=NP/WP*24000.
C      CALC AREA (FT**2)
C      PI/12=0.2617993
C      A=0.2617993*D*EFFLEN*NT
C      A=DFTPI*EFFLEN*NT
C      CALC QOPT
C      QOPT=WP/A/1000.
C      CALC PERCENT WATER RECOVERY. MASS BASIS.
C      WREC = WPM/WFM *100.DO
C
C      CALC AREA COST (CENTS/KGAL)
C      C1=CA*A/WP*1000.
C      CALC POWER COST (CENTS/KGAL)
C      C2=CE*UNP
C      CALC WASTE DISPOSAL COST (CENTS/KGAL).
C      C3 = WR*CD/WP
C
C      CALC TOT UNIT COST, UNBLENDED (CENTS/KGAL).
C      CT = C1+C2+C3
C      BLEND FEED WITH PRODUCT WATER UP TO PLIM IF IBLEND=2.
C      BC = CT
C      COST = CT
C      IF(CPMAX.LT.PLIM)BC=CT*(1.-(PLIM-CPMAX)/(CFEED-CPMAX))
C      IF(IBLEND.EQ.2) COST = BC
C
C      CALC PERCENT FEED IN BLENDED PRODUCT.
C      PCFEED = 100.DO*(PLIM-CPMAX)/(CFEED-CPMAX)
C      IF(PCFEED.LT.1.D-4) PCFEED=0.DO

```

ECON0050
 ECON0060
 ECON0070
 ECON0080
 ECON0090
 ECON0100
 ECONC110
 ECONC120
 ECON0130
 ECONC140
 ECON0150
 ECONC160
 ECON0170
 ECON0180
 ECON0190
 ECONC200
 ECONC210
 ECONC220
 ECONC230
 ECONC240
 ECONC250
 ECONC260
 ECONC270
 ECONC280
 ECONC290
 ECONC300
 ECONC310
 ECONC320
 ECONC330
 ECONC340
 ECONC350
 ECONC360
 ECONC370
 ECONC380
 ECONC390
 ECONC400
 ECONC410
 ECONC420
 ECONC430
 ECONC440
 ECONC450
 ECONC460
 ECONC470
 ECONC480
 ECONC490
 ECONC500
 ECONC510
 ECONC520
 ECONC530
 ECONC540
 ECONC550
 ECONC560
 ECONC570
 ECONC580
 ECONC590
 ECONC600
 ECONC610
 ECONC620
 ECONC630
 ECONC640
 ECONC650
 ECONC660
 ECONC670
 ECONC680
 ECONC690
 ECONC700
 ECONC710
 ECONC720
 ECONC730
 ECONC740

```

C      CALC BLENDED WATER RECOVERY, PERCENT, WT BASIS.
      BWREC = WREC
      IF(CPMAX.LT.PLIM) BWREC= 100.DO*(CFEED*CR-CFEED)/(CFEED*CR-PLIM)
C      CALC BLENDED AREA, POWER, DISPOSAL COSTS.
      BC1 = C1*BC/CT
      BC2 = C2*BC/CT
      BC3 = C3*BC/CT
C
      IF(IQOUT.EQ.0) GO TO 5
C      WRITE ECONOMIC RESULTS ONCE FOR OPTIM. (IQOUT=1)
      WRITE(6,1) NT,WREC,WF,WR,PP,TP,NP,UNP,A,C1,C2,C3,
1      QOPT,L(NPLUS1),CT
C
1      FORMAT( 1H0/ 'OECONOMIC RESULTS/' ' UNBLENDED BASIS'/
A      20HONUMBER OF TUBES, NT ,16X,1PE14.7 /
      133H WATER RECOVERY, PERCENT , 3X,E14.7/
      129H FEED FLOW, WF (GPD) ,7X, E14.7/
      229H REJECT FLOW, WR (GPD) ,7X, E14.7/
      329H PUMP POWER, PP (KW) ,7X, E14.7/
      429H TURBINE POWER, TP (KW) ,7X, E14.7/
      529H NET POWER, NP (KW) ,7X, E14.7/
      629H UNIT POWER, UNP (KWH/KGAL) ,7X, E14.7/
      729H AREA, A (SQ FT) ,7X, E14.7/
      829H AREA COST, C1 (CENTS/KGAL) ,7X, E14.7/
      929H POWER COST, C2 (CENTS/KGAL) ,7X, E14.7/
      131H DISPOSAL COST, C3 (CENTS/KGAL),5X,E14.7/
      134H AVG OPT FLUX, QOPT (KGAL/SQFTDAY),2X,E14.7/
      132H LENGTH OF TUBE, L (FT) ,4X,E14.7/
      233H TOTAL UNIT COST,CT (CENTS/KGAL) ,3X,E14.7 )
C
      WRITE (6,6003) PCFEED,BWREC,BC1,BC2,BC3,BC
6003 FORMAT( 'OBLENDED BASIS' / 1H /
      136H FEED IN BLENDED PRODUCT, PERCENT ,1PE14.7/
      236H BLENDED WATER RECOVERY, PERCENT ,E14.7/
      236H BLENDED AREA COST (CENTS/KGAL) ,E14.7/
      336H BLENDED POWER COST (CENTS/KGAL) ,E14.7/
      436H BLENDED DISPOSAL COST (CENTS/KGAL) ,E14.7/
      233H BLENDED WATER COST (CENTS/KGAL) ,3X,E14.7 )
C
      GO TO 3
C
5      IF(LIST6.NE.0) GO TO 3
      IF(LINES.LT.50) GO TO 10
C      PRINT 1 LINE FOR 1 CALL FROM HYPFIL OR LINCON. (SEARCHING,IQOUT=0)
      WRITE(6,6001)
6001 FORMAT(1H1/1H ,41X, 'RESULTS CALCULATED WHILE SEARCHING FOR OPTIMECON1190
1UM' / 1H0,73X, 'PLIM CFEED REM- C1 C2 ECON1200
2 WATER' / ' POF VIN CR TAPEECON1210
3R D L QOPT WREC -CPM -CPE RELIECON1220
4M UNBLENDED UNBLENDED COST' /1H )
      LINES = 0
      10 WRITE(6,6002) STEP,POF,VIN,CR,TAPER,D,L(NPLUS1),QOPT,WREC,DIFCP,
1      DIFCPE,DIFRE,C1,C2,COST
6002 FORMAT(1H ,A1,F9.3,F8.3,F7.3,F7.4,F8.4,F9.1,F9.5,1PE11.2,3E10.2,
1      OP3F10.4)
      LINES = LINES+1
C
3      RETURN
      END

```

List B-2. Namelist Input Variables

HYPERFILTRATION DESIGN OPTIMIZATION - INPUT VARIABLES (NAMELIST RESET)

09-09-69

INITIAL VALUE	LOWER LIMIT	UPPER LIMIT	WEIGHTING FACTOR	VARIABLE
P 50.000000	PL 0.0	PU 750.000000	PWT 1.000000	INLET PRESSURE (ATM)
V 5.000000	VL 0.010000	VU 100.000000	VWT 1.000000	INLET VELOCITY (FT/SEC)
CR 2.000000	CRL 1.001000	CRU 15.000000	CRWT 1.000000	CONCENTRATION RATIO, WASTE/FEED
TAPER 0.0	TAPERL -30.000000	TAPERU 30.000000	TWT 1.000000	TAPER
D 0.454500	DL 0.0	DU 3.000000	DWT 1.000000	INSIDE TUBE DIAMETER (INCHES)

VARIABLE	VALUE	DESCRIPTION	VARIABLE	VALUE	DESCRIPTION
NVARL	3	MIN NUM OF VARIABLES TO BE OPTIMIZED	PERM	0.000500	MEMBRANE PERMEABILITY (KGAL/SQFT/ATM/DAY)
NVARU	3	MAX NUM OF VARIABLES TO BE OPTIMIZED	CFEED	0.005000	FEED CONCENTRATION (WT FRACTION)
INCR	50	NUMBER OF CONC INCREMENTS	PLIM	0.000500	MAX SALT CONC OF PRODUCT (WT FRACTION)
WP	1.000 06	PRODUCT FLOW RATE (GAL/DAY)	RELIM	2100.000000	MINIMUM REYNOLDS NUM ALLOWABLE
TEMP	70.000000	TEMPERATURE (DEG F)	MTYPE	2	MEMB TYPE (1= ION EXCH) (2= POLY FIT)
FRATIO	1.000000	'ROUGHNESS FACTOR'	CSTAR	0.350000	ION-EXCH MEMB CAPACITY (MOLES/KG H2O)
XRATIO	1.000000	MODIFY EXPONENT, TURB PROMOTERS	CONST	165.000000	SALT REJ VS FLUX (SQFT DAY/KGAL)
REJECT	0.950000	SALT REJECTION, INIT GUESS (FRACTION)	A3	0.050000	POLY FIT CONSTANT TERM
ALPH	0.001000	STEP SIZE FOR GRADIENT	B3	0.0	POLY FIT
BETA	0.005000	STEP SIZE FOR SEARCHING	C3	0.0	POLY FIT
LOOPS	1	NUM LOOPS PER CASE (1= ONE, REG) (2= TWO)	D3	0.0	POLY FIT
MAXTIM	1080000	MAX TIME PER CASE (.01 SEC)	XP	0.250000	REYNOLDS NUM EXPONENT
LOGCR	0	ADJUST CR (0= NO) (1= YES)	ITUBE	2	TUBE SHAPE (1= ROUND) (2= RECTANGULAR)
CRADJ	0.900000	CR ADJUSTMENT FACTOR	RA	10.000000	ASPECT RATIO, RECT (MEMB SIDE/SOLID SIDE)
LIST1	1	(0= LIST ON TAPE 1) (1= SKIP)	IBLEND	2	BLEND FEED WITH PRODUCT (1= NO) (2= YES)
LIST6	0	(0= LIST ON TAPE 6) (1= SKIP)	CA	1.000000	UNIT COST OF MEMB AREA (CENTS/SQFT/DAY)
EP	0.700000	PUMP EFFICIENCY (FRACTION)	CE	1.000000	UNIT COST OF ELEC POWER (CENTS/KWH)
ET	0.0	RECV TURBINE EFFICIENCY (FRACTION)	CD	0.0	UNIT COST OF WASTE DISPOSAL (CENTS/KGAL)

List B-3. Results At Optimum

HYPERFILTRATION DESIGN OPTIMIZATION TUBULAR CHANNEL-TURBULENT FLOW												09-09-69	
INPUT VARIABLES													
INLET PRESSURE, POF (ATM)	98.3201	MEMBRANE PERMEABILITY (KGAL/SQFT/ATM/DAY)	0.000500										
INLET VELOCITY, VIN (FT/SEC)	9.2286	FEED CONCENTRATION, CFED (WT FRACTION NAEL)	0.005000										
CONCENTRATION RATIO, CR (REJ CONC/FEED CONC)	2.6299	MAX SALT CONC OF PRODUCT, PLIM (WT FRACTION)	0.000500										
TAPER	0.0	MINIMUM REYNOLDS NUM ALLOWABLE, RELIM	2100.										
INSIDE TUBE DIAMETER, D, (4HR) (INCHES)	0.4545	MEMBRANE TYPE (1= ION EXCH) (2= POLY FIT)	2										
PRODUCTION RATE (MILLIONS OF GAL/DAY)	1.000000	ION-EXCH MEMB CAPACITY, CSTAR (MOLES/KG H2O)	0.350000										
TEMPERATURE (F)	70.0000	CONST, REJ VS FLUX (SQFT DAY/KGAL)	165.000000										
BLEND FEED WITH PRODUCT (1= NO) (2= YES)	2	POLY FIT CONSTANT TERM, A3	0.050000										
TUBE SHAPE (1=ROUND) (2=RECTANGULAR)	2	REYNOLDS NUM EXPONENT, XP	0.250000										
ASPECT RATIO (RECT) (MEMB SIDE/SOLID SIDE)	10.000000	MEMB AREA UNIT COST, CA (CENTS/SQFT/DAY)	1.0000										
CROSS-SECTIONAL AREA (SQFT)	0.004339	ELEC POWER UNIT COST, CE (CENTS/KWH)	1.0000										
EFFECTIVE MEMBRANE PERIMETER (FT)	0.416625	WASTE DISPOSAL UNIT COST, CD (CENTS/KGAL)	0.0										
FRACTION (MEMBRANE/WETTED PERIM)	0.909091	PUMP EFFICIENCY, EP	0.7000										
FRATIO, 'ROUGHNESS FACTOR'	1.000	RECVD TURBINE EFFICIENCY, ET	0.0										
XRATIO, MODIFY EXPONENT, TURB PROMOTERS	1.000												
INCR	PREM (ATM)	P-WALL (ATM)	VEL (FT/SEC)	RE NUM	LENGTH (FT)	MIX-EFF	REJECT	C-CORE (WT-FR)	C-ALPHA (WT-FR)	CPBAR (WT-FR)	RATE	PROD (LLB/SEC)	CUM-CR (WT-FR)
1	98.320	93.530	9.229	32875.	0.0	0.813971	0.949979	0.005000	0.006073	0.000309	46.453	0.0	0.0
2	97.249	92.289	8.918	31766.	44.96	0.811869	0.949977	0.005163	0.006287	0.000320	45.861	0.08399	0.000309
3	96.290	91.160	8.626	30728.	87.63	0.809704	0.949974	0.005326	0.006501	0.000331	45.321	0.16268	0.000314
4	95.430	90.128	8.353	29752.	128.17	0.807485	0.949972	0.005489	0.006717	0.000342	44.827	0.23657	0.000319
5	94.655	89.180	8.095	28835.	166.73	0.805223	0.949970	0.005652	0.006935	0.000352	44.373	0.30608	0.000324
6	93.957	88.308	7.853	27971.	203.44	0.802925	0.949967	0.005815	0.007154	0.000363	43.953	0.37157	0.000329
7	93.326	87.501	7.624	27155.	238.41	0.800598	0.949965	0.005978	0.007375	0.000375	43.565	0.43339	0.000334
8	92.753	86.752	7.408	26383.	271.78	0.798249	0.949963	0.006141	0.007597	0.000386	43.203	0.49184	0.000339
9	92.233	86.054	7.203	25653.	303.63	0.795883	0.949961	0.006304	0.007820	0.000397	42.866	0.54717	0.000344
10	91.760	85.401	7.009	24960.	334.07	0.793504	0.949959	0.006467	0.008045	0.000408	42.550	0.59964	0.000348
11	91.328	84.789	6.824	24303.	363.18	0.791119	0.949956	0.006630	0.008271	0.000420	42.254	0.64945	0.000353
12	90.933	84.212	6.649	23677.	391.05	0.788729	0.949954	0.006793	0.008499	0.000431	41.974	0.69680	0.000358
13	90.571	83.668	6.482	23082.	417.76	0.786339	0.949952	0.006956	0.008727	0.000443	41.709	0.74187	0.000362
14	90.239	83.152	6.323	22515.	443.37	0.783951	0.949950	0.007119	0.008957	0.000454	41.458	0.78482	0.000366
15	89.933	82.662	6.171	21975.	467.94	0.781568	0.949948	0.007282	0.009188	0.000466	41.219	0.82579	0.000371
16	89.652	82.195	6.027	21458.	491.55	0.779192	0.949946	0.007445	0.009421	0.000477	40.991	0.86491	0.000375
17	89.392	81.748	5.888	20964.	514.24	0.776826	0.949944	0.007608	0.009655	0.000489	40.773	0.90231	0.000379
18	89.152	81.321	5.756	20492.	536.07	0.774471	0.949942	0.007771	0.009890	0.000501	40.564	0.93810	0.000383
19	88.930	80.910	5.629	20039.	557.08	0.772128	0.949940	0.007934	0.010126	0.000513	40.363	0.97327	0.000388
20	88.724	80.514	5.507	19606.	577.33	0.769800	0.949938	0.008097	0.010363	0.000525	40.169	1.00522	0.000392
21	88.533	80.132	5.390	19190.	596.84	0.767487	0.949936	0.008260	0.010601	0.000537	39.981	1.03674	0.000396
22	88.356	79.762	5.278	18790.	615.67	0.765190	0.949934	0.008423	0.010841	0.000549	39.800	1.06700	0.000400
23	88.190	79.404	5.171	18406.	633.84	0.762911	0.949932	0.008586	0.011082	0.000561	39.623	1.09608	0.000404
24	88.036	79.056	5.067	18037.	651.40	0.760650	0.949930	0.008749	0.011323	0.000573	39.452	1.12405	0.000408
25	87.892	78.717	4.968	17681.	668.37	0.758409	0.949928	0.008912	0.011566	0.000585	39.285	1.15096	0.000411
26	87.758	78.386	4.872	17339.	684.78	0.756187	0.949926	0.009075	0.011810	0.000598	39.122	1.17688	0.000415
27	87.632	78.064	4.779	17009.	700.66	0.753985	0.949924	0.009238	0.012055	0.000610	38.963	1.20186	0.000419
28	87.514	77.748	4.690	16691.	716.04	0.751805	0.949922	0.009401	0.012301	0.000622	38.808	1.22594	0.000423
29	87.403	77.439	4.604	16384.	730.93	0.749645	0.949920	0.009564	0.012548	0.000635	38.655	1.24918	0.000427
30	87.300	77.136	4.521	16088.	745.37	0.747507	0.949918	0.009727	0.012796	0.000647	38.505	1.27162	0.000430
31	87.202	76.838	4.441	15802.	759.38	0.745391	0.949916	0.009890	0.013045	0.000660	38.358	1.29330	0.000434
32	87.110	76.545	4.363	15525.	772.97	0.743297	0.949914	0.010053	0.013294	0.000672	38.213	1.31426	0.000437
33	87.024	76.256	4.288	15257.	786.16	0.741226	0.949912	0.010216	0.013545	0.000685	38.070	1.33452	0.000441
34	86.942	75.972	4.215	14998.	798.97	0.739177	0.949910	0.010379	0.013797	0.000697	37.929	1.35413	0.000445
35	86.866	75.691	4.145	14747.	811.43	0.737152	0.949908	0.010542	0.014050	0.000710	37.790	1.37312	0.000448
36	86.793	75.414	4.077	14504.	823.53	0.735149	0.949905	0.010705	0.014303	0.000723	37.653	1.39151	0.000452
37	86.724	75.140	4.011	14269.	835.31	0.733169	0.949903	0.010868	0.014557	0.000736	37.517	1.40933	0.000455
38	86.660	74.868	3.947	14041.	846.77	0.731212	0.949901	0.011031	0.014813	0.000749	37.382	1.42661	0.000458
39	86.598	74.600	3.885	13819.	857.92	0.729278	0.949899	0.011194	0.015069	0.000761	37.249	1.44338	0.000462
40	86.540	74.334	3.825	13604.	868.79	0.727368	0.949897	0.011357	0.015325	0.000774	37.117	1.45964	0.000465
41	86.485	74.070	3.766	13356.	879.37	0.725481	0.949894	0.011520	0.015583	0.000787	36.986	1.47544	0.000468
42	86.432	73.808	3.709	13193.	889.69	0.723617	0.949892	0.011683	0.015841	0.000800	36.856	1.49078	0.000472
43	86.382	73.548	3.654	12996.	899.75	0.721776	0.949890	0.011846	0.016101	0.000813	36.727	1.50568	0.000475
44	86.335	73.290	3.601	12805.	909.57	0.719958	0.949887	0.012009	0.016361	0.000826	36.599	1.52017	0.000478
45	86.290	73.034	3.548	12619.	919.14	0.718164	0.949885	0.012172	0.016621	0.000840	36.471	1.53426	0.000481
46	86.247	72.779	3.498	12438.	928.49	0.716392	0.949882	0.012335	0.016883	0.000853	36.345	1.54796	0.000485
47	86.206	72.525	3.448	12261.	937.62	0.714644	0.949880	0.012498	0.017145	0.000866	36.218	1.56130	0.000488
48	86.168	72.273	3.400	12090.	946.54	0.712919	0.949877	0.012660	0.017407	0.000879	36.093	1.57428	0.000491
49	86.130	72.022	3.353	11923.	955.26	0.711216	0.949875	0.012823	0.017671	0.000892	35.968	1.58692	0.000494
50	86.095	71.771	3.308	11760.	963.77	0.709536	0.949872	0.012986	0.017935	0.000906	35.844	1.59924	0.000497
51	86.061	71.522	3.263	11602.	972.11	0.707879	0.949870	0.013149	0.018199	0.0	0.0	1.61124	0.000500

ECONOMIC RESULTS
UNBLENDED BASIS

NUMBER OF TUBES, NT	5.90090320	01
WATER RECOVERY, PERCENT	6.44365740	01
FEED FLOW, WF (GPD)	1.55009870	06
REJECT FLOW, WR (GPD)	5.48134540	05
PUMP POWER, PP (KW)	9.67561930	02
TURBINE POWER, TP (KW)	0.0	
NET POWER, NP (KW)	9.67561930	02
UNIT POWER, UNP (KWH/KGAL)	2.32214860	01
AREA, A (SQ FT)	2.42552650	04
AREA COST, C1 (CENTS/KGAL)	2.42552650	01
POWER COST, C2 (CENTS/KGAL)	2.32214860	01
DISPOSAL COST, C3 (CENTS/KGAL)	0.0	
AVG CPT FLUX, QOPT (KGAL/SQFT/DAY)	4.12281610	-02
LENGTH OF TUBE, L (FT)	9.72105520	02
TOTAL UNIT COST, CT (CENTS/KGAL)	4.74767520	01

BLENDED BASIS

FEED IN BLENDED PRODUCT, PERCENT	0.0	
BLENDED WATER RECOVERY, PERCENT	6.44365740	01
BLENDED AREA COST (CENTS/KGAL)	2.42552650	01
BLENDED POWER COST (CENTS/KGAL)	2.32214860	01
BLENDED DISPOSAL COST (CENTS/KGAL)	0.0	
BLENDED WATER COST (CENTS/KGAL)	4.74767520	01

List B-4. Progress of Optimization (List 6 option)

RESULTS CALCULATED WHILE SEARCHING FOR OPTIMUM

	POF	VIN	CR	TAPER	D	L	QOPT	WREC	PLIM -CPM	CFEED -CPE	REM- RELIM	C1 UNBLENDED	C2 UNBLENDED	WATER COST
I	50.000	5.000	2.000	0.0	0.4545	862.3	0.02041	5.220 01	7.270-05	4.350-03	6.370 03	48.9870	14.5641	62.5402
G	50.750	5.000	2.000	0.0	0.4545	846.3	0.02080	5.220 01	7.220-05	4.350-03	6.370 03	48.0744	14.7817	61.8632
G	50.000	5.100	2.000	0.0	0.4545	882.3	0.02035	5.220 01	7.390-05	4.360-03	6.540 03	49.1454	14.5661	62.6818
G	50.000	5.000	2.014	0.0	0.4545	869.0	0.02039	5.260 01	7.120-05	4.350-03	6.300 03	49.0318	14.4658	62.5080
L	53.750	4.895	2.003	0.0	0.4545	771.2	0.02240	5.230 01	6.780-05	4.340-03	6.170 03	44.6519	15.6232	59.3804
L	59.375	4.739	2.008	0.0	0.4545	662.9	0.02531	5.250 01	5.760-05	4.320-03	5.880 03	39.5169	17.1986	55.9983
L	67.812	4.503	2.016	0.0	0.4545	542.2	0.02957	5.280 01	3.780-05	4.280-03	5.430 03	33.8183	19.5327	52.9063
L	80.469	4.150	2.027	0.0	0.4545	416.4	0.03580	5.330 01	-1.970-07	4.210-03	4.770 03	27.9293	22.9666	50.8959
L	80.449	4.156	2.026	0.0	0.4545	416.9	0.03580	5.330 01	1.890-07	4.210-03	4.780 03	27.9357	22.9731	50.9067
L	99.434	4.278	1.918	0.0	0.4545	323.4	0.04517	5.070 01	-2.490-05	4.180-03	5.390 03	22.1410	29.8649	52.0058
L	96.992	5.033	1.785	0.0	0.4545	356.8	0.04417	4.650 01	2.570-05	4.310-03	7.470 03	22.6421	31.7668	54.1002
THIS STEP FAILED. ALPHA= 0.001000000														
G	81.199	4.156	2.026	0.0	0.4545	412.8	0.03616	5.330 01	-1.580-06	4.200-03	4.780 03	27.6539	23.1829	50.8368
G	80.449	4.256	2.026	0.0	0.4545	426.9	0.03579	5.330 01	3.380-06	4.210-03	4.950 03	27.9395	22.9809	50.8821
G	80.449	4.156	2.040	0.0	0.4545	419.9	0.03578	5.360 01	-1.730-06	4.200-03	4.730 03	27.9500	22.8240	50.7741
L	82.424	4.215	2.096	0.0	0.4545	426.1	0.03666	5.500 01	-1.210-05	4.170-03	4.630 03	27.2751	22.8062	50.0813
L	81.476	4.443	2.077	0.0	0.4545	450.6	0.03621	5.450 01	-6.640-08	4.200-03	5.070 03	27.6152	22.7534	50.3685
L	81.471	4.445	2.077	0.0	0.4545	450.7	0.03621	5.450 01	-3.280-09	4.200-03	5.070 03	27.6171	22.7531	50.3702
L	81.471	4.445	2.077	0.0	0.4545	450.7	0.03621	5.450 01	-1.630-10	4.200-03	5.070 03	27.6172	22.7531	50.3702
L	81.471	4.445	2.077	0.0	0.4545	450.7	0.03621	5.450 01	-8.070-12	4.200-03	5.070 03	27.6172	22.7531	50.3702
L	83.573	5.052	2.182	0.0	0.4545	522.3	0.03701	5.680 01	-1.280-06	4.170-03	5.630 03	27.0196	22.3937	49.4134
L	83.473	5.076	2.180	0.0	0.4545	525.1	0.03696	5.670 01	-2.000-07	4.170-03	5.680 03	27.0576	22.3860	49.4435
L	83.458	5.079	2.179	0.0	0.4545	525.5	0.03695	5.670 01	-3.200-08	4.170-03	5.690 03	27.0635	22.3848	49.4483
L	83.455	5.080	2.179	0.0	0.4545	525.6	0.03695	5.670 01	-5.150-09	4.170-03	5.690 03	27.0645	22.3846	49.4490
L	83.455	5.080	2.179	0.0	0.4545	525.6	0.03695	5.670 01	-8.300-10	4.170-03	5.690 03	27.0646	22.3845	49.4491
L	83.455	5.080	2.179	0.0	0.4545	525.6	0.03695	5.670 01	-1.340-10	4.170-03	5.690 03	27.0646	22.3845	49.4492
L	83.455	5.080	2.179	0.0	0.4545	525.6	0.03695	5.670 01	-2.150-11	4.170-03	5.690 03	27.0646	22.3845	49.4492
L	86.608	5.990	2.337	0.0	0.4545	634.8	0.03803	5.980 01	-3.990-06	4.130-03	6.430 03	26.2980	22.0387	48.3367
L	86.296	6.065	2.330	0.0	0.4545	644.4	0.03785	5.970 01	-1.090-06	4.140-03	6.560 03	26.4229	22.0073	48.4302
L	86.211	6.086	2.329	0.0	0.4545	647.1	0.03780	5.960 01	-3.060-07	4.140-03	6.600 03	26.4573	21.9987	48.4560
L	86.187	6.091	2.328	0.0	0.4545	647.8	0.03778	5.960 01	-8.700-08	4.140-03	6.610 03	26.4671	21.9963	48.4633
L	86.180	6.093	2.328	0.0	0.4545	648.0	0.03778	5.960 01	-2.480-08	4.140-03	6.610 03	26.4698	21.9956	48.4654
L	86.178	6.094	2.328	0.0	0.4545	648.1	0.03778	5.960 01	-7.060-09	4.140-03	6.610 03	26.4706	21.9954	48.4660
L	86.178	6.094	2.328	0.0	0.4545	648.1	0.03778	5.960 01	-2.010-09	4.140-03	6.610 03	26.4709	21.9953	48.4662
L	86.178	6.094	2.328	0.0	0.4545	648.1	0.03778	5.960 01	-5.730-10	4.140-03	6.610 03	26.4709	21.9953	48.4662
L	86.178	6.094	2.328	0.0	0.4545	648.1	0.03778	5.960 01	-1.630-10	4.140-03	6.610 03	26.4709	21.9953	48.4662
L	86.178	6.094	2.328	0.0	0.4545	648.1	0.03778	5.960 01	-4.650-11	4.140-03	6.610 03	26.4709	21.9953	48.4662
L	90.907	7.459	2.564	0.0	0.4545	817.3	0.03908	6.350 01	-8.380-06	4.080-03	7.520 03	25.5881	21.7739	47.3620

List B-5. Progress of Optimization (List-1 and List-6 options)

RESULTS CALCULATED WHILE SEARCHING FOR OPTIMUM

	PDF	VIN	CR	TAPER	D	L	QUPT	WREC	PLIM -CPM	CFEED -CPE	REM- RELIM	C1 UNBLENDED	C2 UNBLENDED	WATER COST
I	50.000	5.000	2.000	0.0	0.4545	862.3	0.02041	5.220 01	7.270-05	4.350-03	6.370 03	48.9870	14.5641	62.5402
* 1 0 0.625401690 D2 * 0.500000000 02 0.500000000 01 0.200000000 01														
REM= 0.846801940 C4 CPM= 0.427261540-03 CPE= 0.646470670-03 G 50.750 5.000 2.000 0.0 0.4545 846.3 0.02080 5.220 01 7.220-05 4.350-03 6.370 03 48.0744 14.7817 61.6632														
REM= 0.846752090 C4 CPM= 0.427773260-03 CPE= 0.647320510-03														
2 0 0.618631970 D2 0.507500000 02 0.500000000 01 0.200000000 01 G 50.000 5.100 2.000 0.0 0.4545 882.3 0.02035 5.220 01 7.390-05 4.360-03 6.540 03 49.1454 14.5661 62.6818														
REM= 0.863854200 C4 CPM= 0.426074820-03 CPE= 0.644312040-03														
3 0 0.626617870 D2 0.500000000 02 0.509999000 01 0.200000000 01 G 50.000 5.000 2.014 0.0 0.4545 869.0 0.02039 5.260 01 7.120-05 4.350-03 6.300 03 49.0318 14.4658 62.5080														
REM= 0.840451000 C4 CPM= 0.428759290-03 CPE= 0.651667820-03														
4 0 0.625080250 D2 0.500000000 02 0.500000000 01 0.201399900 01														
PARTIAL DERIVATIVES OF GRADIENT EVALUATION -0.676971420 03 0.141618760 03 -0.321434570 02 SLEFAC= 0.1000000 01 0.1000000 01 0.1000000 01														
L	53.750	4.895	2.003	0.0	0.4545	771.2	0.02240	5.230 01	6.780-05	4.340-03	6.170 03	44.6519	15.6232	59.3804
REM= 0.827167420 C4 CPM= 0.432201100-03 CPE= 0.655931230-03														
5 0 0.0 0.537500000 02 0.489541320 01 0.200332350 01														
SUCCESSFUL STEP NO. 1														
L	59.375	4.735	2.008	0.0	0.4545	662.9	0.02531	5.250 01	5.760-05	4.320-03	5.880 03	39.5169	17.1986	55.5983
REM= 0.797623030 C4 CPM= 0.442365850-03 CPE= 0.676057890-03														
6 1 0.593804310 D2 0.593750000 02 0.473853290 01 0.200830860 01														
SUCCESSFUL STEP NO. 2														
L	67.812	4.503	2.016	0.0	0.4545	542.2	0.02957	5.280 01	3.780-05	4.280-03	5.430 03	33.8183	19.5327	52.9063