A Transient Numerical Lumped-Parameter Isotopic Evolution and Water Balance Model for the Paleo-Owens River System, California

By Randall M. Roberts

An Independent Study Report Submitted In Partial
Fulfillment of the Requirements of the Master of Science
Degree In Hydrology

New Mexico Institute of Mining and Technology

March, 1990

1. Introduction

In recent years, analysis of continuous marine sediment cores has allowed great advances in the understanding of marine and polar climate (Shackleton and Opdyke, 1973; Hays et al., 1977). Similar progress has not been made in the understanding of continental climate due to a lack of old, continuous sediment records. Using a stable isotope record developed from detailed analysis of the Kerr-McGee KM-3 core from Searles Lake, California, this study addresses this problem by quantitatively reconstructing lake surface-area histories for the paleo-Owens River system. The paleo-Owens River closed-basin lake system was comprised of Owens Lake, China Lake, Searles Lake, Panamint Lake, and Lake Manly (Death Valley). Water levels in closed-basin lakes are a direct measure of the basin water balance, and thus comparison of the water-level reconstruction with independent climatic records will enable determination of the response of the hydrologic system to climatic perturbations.

The main tool for the lake-level reconstructions is a transient numerical model. The equations used were initially presented by Gonfiantini (1965). The model is based on the one described in Phillips and others(1986), but differs from that model in being formulated on a time-derivative basis rather than a volume-derivative basis. The model consists of linked water mass balance and isotope mass balance equations. In order to compute the mass balances the model requires histories of temperature, humidity, precipitation rate, composition of the lake water. For cases (such as the paleo-Owens River system) in which a series of lakes are involved, the model sequentially calculates the lake history for each lake individually, then uses the outflow from that lake as the inflow to the next lake in the chain.

The basic approach used in implementing the transient numerical model was to vary the inflow history so as to match the isotopic history. Based on the linked water/isotope mass balance, the model then calculates the lake surface area as a function of time. In order to avoid a long and tedious matching procedure, the initial "guess" for the inflow history of the lake was taken from the output of a steady-state lake simulation model. The steady-state model back-calculated the lake inflow at each ^{18}O data point, based on the much simpler steady-state equations. This steady-state approximation was then entered into the transient model and used as the input for the initial simulations. The inflow was then varied in order to better match the ^{18}O data for intervals where transient effects were important. This approach proved to be much more efficient than intuitive guessing of an initial inflow history.

In addition to calculating an isotopic history of the lake water, the model also produces a history of chloride accumulation in the lake sediments. It performs this using a constant chloride influx and mass balance equations, assuming that the chloride is conservative in the lake water unless its concentration exceeds that for solubility of halite. Although variations in accumulation of soluble salts at Searles Lake have sometimes previously been explained by variations in the influx, we have found that the chloride accumulation history can largely be explained simply by climatically controlled variations in the inflow of water, and we believe that accumulation of many other solutes can be explained on a similar basis. There is some evidence for relative constancy of the chloride influx over long time periods (Jannik, 1989), and we feel that explaining the chloride deposition variations using only the climatically-controlled water inflow changes is more parsimonious than arbitrary "deus ex machina" variations in chloride influx.

2. Model Formulation

The model is a transient numerical lumped-parameter model which simulates surface area and isotopic evolution of five lakes in the pales-Owens Rover system. This includes Owens Lake, China Lake, Searles Lake, Panamint Lake and Lake Manly (Death Valley). Water mass balance for each lake is given by:

$$\frac{d(V_L)}{dt} = Q_I + Q_P + Q_C - Q_O - Q_E \qquad (1)$$

Where V_L is the lake volume, Q_I is the inflow flux (from the Owens River or the preceding lake), Q_P is the flux of precipitation onto the lake, Q_C is the back-condensation flux, Q_O is the overflow flux, Q_E is the gross evaporative flux, and t is time. The isotopic mass balance is described by:

$$\frac{d(V_L \delta_L)}{dt} = \delta_I Q_I + \delta_P Q_P + \delta_C Q_C - \delta_O Q_O - \delta_E Q_E \qquad (2)$$

Where δ is the relative isotopic enrichment of the reservoirs and fluxes indicated by the subscripts. Applying the chain rule to (2) we have:

$$V_L \frac{d\delta_L}{dt} + \delta_L \frac{dV_L}{dt} = \delta_I Q_I + \delta_P Q_P + \delta_C Q_C - \delta_O Q_O - \delta_E Q_E \qquad (3)$$

Letting $B = \delta_I Q_I + \delta_P Q_P + \delta_C Q_C - \delta_O Q_O - \delta_E Q_E$ and solving (3) for $d\delta_L/dt$ gives:

$$\frac{d\delta_L}{dt} = \frac{\left(B - \delta_L \frac{dV_L}{dt}\right)}{V_L} \qquad (4)$$

(1) and (4) are solved for each time step of the numerical simulation using the Runge-Kutta-Fehlberg difference-equation method (Fehlberg, 1970). Because calculations in the modeling involved numbers which differed by many orders of magnitude, all variables in the FORTRAN code were declared as double precision to minimize rounding errors. The accuracy of the numerical solution was verified by comparison with an analytical solution.

Bathymetric data for each of the lake basins were used to convert the volumes calculated by the model to surface area. Data for Owens and Searles Lakes are from G.I. Smith (1979); for China Lake and Lake Manly from N.O. Jannik (1989); and for Panamint Lake from R.S.U. Smith (1976).

Salinity strongly affects both evaporation and isotopic fractionation and must therefore be taken into account by the model. The model calculates changes in salinity with time for Owens, China, Searles, and Panamint Lake using a simple mass-balance approach. Specifically calculated is the concentration of chloride ions. The starting point for the mass balance is a constant input of chloride. Several studies have concluded that the chloride load of the Owens River has been at least roughly constant with time (Jannik, 1989; Smith, 1976) and approximately equal to 5.9 x 10^6 kg yr^{-1} (Smith, 1976). For a given time step of the model the mass (kg) of chloride entering Owens Lake is simply:

$$Cl = \Delta t \frac{5.9\times10^{6}\,kg}{yr} \qquad (5)$$

Where Δt is the is the size (yr) of the current time step. To determine whether chloride will be precipitated, the model then calculates the chloride concentration (moles ℓ^{-1}) at the end of the time step as:

$$Conc_{Cl} = Cl \frac{10^3 g}{kg} \frac{mole}{35.453 g} VOL^{-1} \frac{m^3}{10^3 \ell} \qquad (6)$$

where VOL is the volume (m^3) of the lake at the end of the current time step. If $Conc_{Cl}$ is greater than 6.1 moles ℓ^{-1}, the solubility of halite, the mass (kg) of chloride precipitated in the lake is given by:

$$Cl_{dep} = (Conc_{Cl} - 6.1) \frac{35.453\times10^{-3} kg}{mole} VOL \frac{10^3 \ell}{m^3} \qquad (7)$$

For a lake at overflow, $dV_L/dt = 0$ and from (1) we see that the flux out of the lake is:

$$Q_O = Q_I + Q_P + Q_C - Q_E \qquad (8)$$

The chloride flux (moles yr^{-1}) out of one lake and into the next is then:

$$Q_{Cl} = Q_0 \frac{10^3 \ell}{m^3} Conc_{Cl} \qquad (9)$$

The gross evaporative flux is given by:

$$Q_E = q_E \, a_w \, A \qquad (10)$$

where q_E is the evaporation rate for pure water, a_w is the chemical activity of the lake water, and A is the lake surface area. The back-condensation flux, Q_C, can be related to the gross evaporative flux, the relative humidity, h, and the activity of water by:

$$Q_C = \frac{h \, Q_E}{a_w} \qquad (11)$$

The activity of water is given by (Graf, 1982):

$$a_w = \exp(-M_w \, \phi \, \Sigma m_i) \, 10^{-3} \qquad (12)$$

Where M_w is the molecular weight of water (g $mole^{-1}$), Φ is the osmotic coefficient for water in the solution, and $\Sigma \, m_i$ is the sum of the molal electrolyte concentrations in the solution. These concentrations were obtained for each time step from the chloride mass balance subprogram described above. The osmotic coefficient was calculated using a simplified form of equation (11) from Pitzer and Kim (1974).

The isotopic composition of the back-condensation flux can be determined from the equilibrium isotopic enrichment factor (ϵ_V) for the liquid/vapor phase change and the isotopic composition of the atmospheric water vapor (δ_A):

$$\delta_C = \epsilon_V\left(1 + \frac{\delta_A}{10^3}\right) + \delta_A \tag{13}$$

The equilibrium enrichment factor is a function of temperature alone and was calculated according to Friedman and O'Neil (1977):

$$\epsilon_V = \left\{\exp\left[\frac{1.534\,(10^6 T^{-2}) - 3.206\,(10^3 T^{-1}) + 2.644}{10^3}\right] - 1\right\}10^3 \tag{14}$$

Because the degree of fractionation during evaporation is determined by the kinetics of the vapor diffusion away from the liquid surface, the $\delta^{18}O$ of the gross vapor flux (δ_E) is a function of humidity, wind speed, and temperature (Craig and Gordon, 1965). Modification of an expression from Merlivat and Jouzel (1979) was used in the model to take these factors into account:

$$\delta_E = \left[\frac{(1 + 10^{-3}\delta_L)\,(1 - \kappa)}{(1 + 10^{-3}\epsilon_V)\,(1 - \kappa h)} - 1\right]10^3 \tag{15}$$

κ is a variable which accounts for both diffusive and turbulent transport of the isotopic species away from the water surface. For the range of wind speeds expected in most continental settings, κ may be treated as a constant, having the value of 6.8×10^{-3} for $H_2{}^{18}O$ (Merlivat and Jouzel, 1979).

The model solves (1) and then uses the necessary values along with the other calculated parameters to solve (4). It should be noted that $\delta_L = \delta_0$ in the model, based on the assumption that each lake can be treated as a well-mixed system over each time step. The δ value calculated by the model from (4) is the relative isotopic enrichment of the water. Because the isotopic history the model is attempting to match is given as $\delta_{dolomite}$, the water values are converted by first calculating the equilibrium isotopic enrichment factor for water/dolomite:

$$\epsilon_{H_2O,dol} = \left[\exp\left\{\left(\left(3.2\left(\frac{10^6}{T^2}\right) - 4.3\right)10^{-3}\right) - 1\right\}\right]10^3 \qquad (16)$$

and then calculating the relative isotopic enrichment of the dolomite:

$$\delta_{dol} = \epsilon_{H_2O,dol} + \delta_{H_2O}\left(\frac{\epsilon_{H_2O,dol}}{10^3} + 1\right) \qquad (17)$$

3. Model Parameterization

In order to calculate lake surface areas, the numerical model requires histories of $\delta^{18}O$ of the inflow, $\delta^{18}O$ of the atmospheric humidity, temperature, evaporation rate, precipitation rate on the lake surfaces, and relative humidity. Obviously, there are no independent histories covering the past 1.4 Myr for each of these parameters, for the study area. Consequently, histories for the independent parameters must be constructed by correlation to existing paleoclimatic parameter histories that cover the time period. We know of only two such histories. One is the $\delta^{18}O$

measurements at Searles lake, themselves, and the other is the marine $\delta^{18}O$ record.

Fortunately, there is some independent basis for choosing the appropriate history with which to link the various independent parameters. Winograd and others (1988) have published a U/Th dated chronology of $\delta^{18}O$ variations in vein calcite from Devils Hole, Nevada. The $\delta^{18}O$ of the calcite reflects ^{18}O variations in the groundwater from which the calcite was precipitated,and thus ultimately the $\delta^{18}O$ of precipitation, probably mostly on the Spring Mountains of Nevada. The most notable feature of this ^{18}O record is the remarkable fidelity with which it mimics the marine ^{18}O record. (We note that Winograd and others (198) observed dating discrepancies between their record and the marine one. We assume the discrepancies are due to errors in one or both of the chronologies, but since the differences are not within the uncertainties of our chronology, we have not attempted to resolve them.) The primary control on $\delta^{18}O$ of precipitation is temperature (Dansgaard, 1964) and we take the strong similarity of the Devil's Hole and Marine $\delta^{18}O$ records as evidence that the marine $\delta^{18}O$ is a good proxy for temperature in the study area. The similarity is certainly evidence that the $\delta^{18}O$ of precipitation (and thus also the $\delta^{18}O$ of atmospheric moisture) can be reconstructed on the basis of the marine $\delta^{18}O$ record.

In this model, evaporation is parameterized as the difference of the gross evaporation flux and the atmospheric back-condensation flux. The gross evaporation is thus independent of atmospheric humidity and is presumably largely a function of temperature (although the results of Benson (1986) indicate that factors we have not tried to incorporate into the model, such as the amount of cloudiness, also play a significant role). We have therefore linked the (gross) evaporation to the temperature record.

The remaining independent parameter of major significance is the relative humidity. Sensitivity analyses showed that wide variations in humidity are necessary to explain the lacustrine $\delta^{18}O$ data. High humidities were required for the model to produce the observed light values, even if all other parameters were favorable. Conversely, low humidities were required to match the observed heavy $\delta^{18}O$ values. We therefore considered that the Searles $\delta^{18}O$ record itself was the best guide to the relative humidity history. This approach is consistent with the basic assumption that lighter $\delta^{18}O$ periods represent times of more favorable water balance (and thus presumably higher humidity) while periods of heavy $\delta^{18}O$ represent unfavorable water balances, and thus lower humidity.

The histories were constructed for all of the independent parameters (except humidity) by first establishing a correlation between temperature and the marine $\delta^{18}O$ record, then correlating the rest of the parameters to the temperature history. A linear relation between temperature and $\delta^{18}O$ was assumed and was calibrated by matching the modern mean annual temperature of 19.1 °C at Searles to the modern $\delta^{18}O$ of 3.51

per mil, and the late Wisconsin minimum assumed temperature of 12.1 °C at 13.6 ka to the corresponding $\delta^{18}O$. The 12.1 °C temperature at 13.6 ka is based on a temperature reduction estimate of about 7 °C at the late Wisconsin glacial maximum Dohrenwend ( ), Spaulding and others (1983), and Phillips and others (1986).

The correlations of climatic parameters to temperature were obtained by assuming that the ancient temperature correlations of these parameters with changing climate were similar to their modern temperature correlations with changing elevation. Climatic data for stations at different elevations in the region were assembled from Smith (1979), Smith and Street-Perrott (1983), Ruffner (1985), NOAA (1982) University of California (1988) and Meyers (1962). Temperature and the other climatic parameters were regressed against elevation. The regressions showed good correlations with elevation. The regression "r^2" values ranged from 0.94 to 0.99. The regression plots and equations can be found in Appendix . The temperature versus elevation and other climatic parameters versus elevation regressions were then combined to regress the other climatic parameters against temperature.

The assumption that the ancient relationship of temperature with the other climatic parameters, as a function of time, mimics the modern relationship with elevation is obviously at best an approximation. However, it does provide an internally consistent basis for reconstructing the covariation of the various parameters. The effect of inadequacies in the reconstruction was assessed by means of a sensitivity analysis, described below.

The calibration of relative humidity to Searles $\delta^{18}O$ was performed using the other parameters obtained as described above. The calibration was accomplished by matching lake surface areas from the model with estimates from times for which there is independent evidence of lake level. The most important calibration points were 12 to 11 ka, when the lake volume had declined until the lake water was near saturation with halite and the isotopic composition showed evidence of equilibrium, a similar period during the interval 810 to 800 ka, and the period 15 to 13 ka, when varnish radiocarbon dates show that Searles was overflowing (Dorn and others, in press) but geological evidence from Panamint Valley indicates only a small lake resulting from that overflow (Smith, 1976). The calibration exercise yielded the following relation of humidity to $\delta^{18}O$:

$$h = 135 - 2.2\ \delta^{18}O_{dol} \qquad (18)$$

A sensitivity analysis was performed by varying the slope and intercept of the humidity/$\delta^{18}O$ equation and substituting them into the steady-state lake model. In general, the resultant steady-state lake levels were quite insensitive to the slope and intercept parameters used, so long as the high and low humidities were within certain ranges. Humidities above 50% were required to keep lake levels within reasonable bounds during light isotopic episodes. If humidities fell below 30%, the model was unable to yield low levels, even during times when geological evidence showed that desiccation was imminent. However, given that the maximum and minimum humidities were above 50% and 30%, respectively, additional changes in the humidity/$\delta^{18}O$ relationship produced only small variations in the simulated lake surface areas. The humidity parameterization thus appears to be quite robust.

4. Model Implementation

The initial inflow history used to "drive" the transient model was derived using a FORTRAN code, SS.FOR (appendix ?). In the main routine this steady-state model sequentially calculated the surface area and $\delta^{18}O$ for each lake in the system. Using nested DO loops to iterate selected input variables, specifically inflow (Q_I), temperature (T), and humidity (h), $\delta^{18}O$ and surface area were calculated as functions of these variables. The steady-state model then stored these "possible solutions" as a 3-D array and a search through the array was implemented to back-calculate Q_I. Simulations were performed using variable temperatures and also constant high and low temperatures to provide sensitivity analysis.

Three parameters had to be entered at the start of the program, the chemical activity of the water (a_W), a starting value for Q_I, and an ending value for Q_I. a_W was assigned a value of 1.0, the activity of pure water, to simulate non-saline conditions, and a value of .761, the activity of halite saturated water, to simulate saline conditions. It should be noted that the variable a_w was used only in the Searles Lake calculations. The implicit assumption was therefore that a_w was a constant with a value of 1.0 for all of the other lakes in the system. For the variable temperature steady-state simulations, the starting and ending values for Q_I were 4.0×10^7 and 5.0×10^9 m^3 yr^{-1} respectively. These values were increased by approximately half an order of magnitude for the constant high temperature simulations and decrease by the same amount for the constant low temperature simulations. Diagnostics written to files along with the output aided in calibration of the inflow ranges. If the range of input values used for Q_I was such that the steady-state model could not match the simulated "possible solutions" to the actual Searles Lake histories, error messages were displayed in the output.

The outermost FORTRAN DO loop was the one which iterated temperature. For the variable temperature simulations, the temperature at Owens Lake was varied from 6.5 °C to 15.5 °C in steps of 0.25 °C. The corresponding temperature for both China and Searles Lake was 3.6 °C higher. The temperature at Panamint Lake was calculated by adding 2.35 °C to the Searles Lake temperature and the temperature at Lake Manly was 4.08 °C higher than that at Panamint Lake. For the constant low and high temperature simulations, the temperatures at Owens Lake were 7.7 and 15.2 °C respectively. Temperature variations between lakes were the same as those given above. The next DO loop nested within the temperature loop was the one which incremented Q_I. For the range of Q_I entered at the beginning of the program, this loop simply divided that range into 100 equal increments. The inner-most DO loop was used to increment humidity from 30% to 90% in steps of 1%. For a given iteration, the humidity values for all of the lakes were the same.

The first parameters calculated in the steady-state model were those which were assumed to be a function of temperature. These included the isotopic compositions of the inflow (δ_I), the atmospheric water vapor (δ_A), and the precipitation (δ_P), as well as the precipitation (q_P) and evaporation (q_E) rates. q_E for any of the lakes was not allowed to be less than 1.35 m yr^{-1}. All of these parameters were calculated as linear functions of temperature using regressions described in the **Model Parameterization** section above.

The isotopic composition of the back-condensation flux (δ_C) was then determined from (13), and the equilibrium enrichment factor was calculated using (14). The kinetic isotopic enrichment factor was calculated as:

$$\epsilon_j = \left[\left(\frac{\alpha_V (1-\kappa h)}{1-\kappa}\right) - 1\right] 10^3 \qquad (19)$$

where α_V, the isotopic enrichment factor, is related to ϵ_V by:

$$\alpha_V = 1 + \frac{\epsilon_V}{10^3} \qquad (20)$$

Surface area was given by:

$$A = Q_I \left[q_E \left(\frac{1-h}{a_W} \right) - q_P \right]^{-1} \qquad (21)$$

The steady-state model then determined fluxes for the different parameters. Gross evaporation flux was given by:

$$Q_E = q_E \times A \times a_W \qquad (22)$$

the precipitation flux was given by:

$$Q_P = q_P \times A \qquad (23)$$

and the back-condensation flux was:

$$Q_C = Q_E \times \frac{h}{a_W} \qquad (24)$$

When the steady-state model was run, the range of inflow values were adjusted so that the first two lakes in the system, Owens Lake and China Lake, were always at overflow. The lakes were determined to be at overflow when the calculated surface area was greater than the maximum surface area as determined by bathymetric data. The relative isotopic enrichment of the lakes at overflow was calculated as:

$$\delta_{H_2O} = \frac{[\, Q_I \delta_I + Q_P \delta_P + Q_E \ (h \delta_C + a_W + \epsilon_K) \,]}{Q_I + h Q_E + Q_P} \qquad (25)$$

Searles Lake however was not always overflowing. For non-overflow conditions the $\delta^{18}O$ of the lake was given by:

$$\delta_{H_2O} = \frac{Q_I \ \delta_I}{Q_E} + \epsilon_j + h \ \frac{\delta_C}{a_W} + \frac{Q_P \ \delta_P}{Q_E} \qquad (26)$$

These $\delta^{18}O$ values were the relative isotopic enrichment of the water. Because the isotopic values used in the matching process were given as $\delta_{dolomite}$, the water values were converted by first calculating the equilibrium isotopic enrichment factor for water/dolomite using (16) and then calculating the relative isotopic enrichment of the dolomite from (17). The flux out of one lake into the next was calculated using (8). At various times during the modeled history, Searles Lake would exceed a critical volume (65.87×10^9 m^3) and couple with China Lake, forming one lake. During these periods the outflow from Owens Lake became the inflow for Searles Lake and there essentially was no China Lake.

The above equations were used to calculate $\delta^{18}O$ and surface area for each lake in the system for each iteration, I, of the temperature DO loop, J, of the Q_I DO loop, and K, of the humidity DO loop. At the end of each iteration the $\delta^{18}O$ of Searles Lake and total surface area of the system were saved in 3-D FORTRAN arrays as functions of these variables, i.e., $A(Q_I(I),T(J),h(K))$, and $\delta(Q_I(I),T(J),h(K))$. After this process was completed the steady-state model read the data files containing the temperature ($T_{Searles}$), humidity ($h_{Searles}$), and $\delta^{18}O$ ($\delta_{Searles}$) histories for Searles Lake. For a specific point in time, $h_{Searles}$ was compared with each $h(K)$ until a match was found and likewise with $T_{Searles}$ and $T(I)$. The values of $Q_I(J)$ were then searched sequentially until $\delta_{Searles}$ matched $\delta(T(I),Q_I(J),h(K))$. $Q_I(J)$ was then saved as the inflow value for that point in time. The corresponding surface area for the system was given by $A(T(I),Q_I(J),h(K))$.

This process was used to generate three inflow histories to be used as input for the transient model; a variable temperature history, a constant high temperature history, and a constant low temperature history. The variable temperature inflow history was used as the initial "guess" for the actual transient modeling. The constant temperature histories were used for sensitivity analysis. When the steady-state model was run with a constant temperature, all the parameters assumed to be a function of temperature also became constants, i.e., δ_I, δ_A, δ_P, q_P, and q_E. This enabled testing of the assumption utilized to develop the histories for the above parameters, i.e., that the ancient relationship of temperature with the other climatic parameters, as a function of time, mimicked the modern relationship with elevation.

Each of the three histories was actually generated in two parts. By varying a_W, the steady-state model could be used to simulate both low lake levels (saline conditions) and high lake levels (non-saline conditions). For each of the temperature histories the model was run once with $a_W = 1$, the activity of pure water, and once with $a_W = .761$, the activity of water at halite saturation. The final history was constructed by substituting Q_I values from the saline history into the non-saline history when geologic evidence showed that Searles Lake was at or near chloride saturation.

The transient modeling was performed using a numerical lumped-parameter code, TRANSISO ~~ISO.FOR~~, written in FORTRAN (appendix ?). The code solves (1) and (4) for each time step of the simulation using the Runge-Kutta-Fehlberg (RKF) difference-equation method (Fehlberg, 1970). Results of the simulation are displayed along with the actual histories in graph form. This allows for visual evaluation of the simulations success.

The main program first reads three files, OWENS.INP, C_S.INP, and P_D.INP. These files contain initial values for parameters necessary to begin and control the simulation. The file OWENS.INP contains the Owens Lake parameters, China and Searles Lake parameters are contained in C_S.INP, and Panamint and Lake Manly (Death Valley) parameters are found in P_D.INP. Contained in these files are values for maximum iterations, lake volume (m^3), $\delta^{18}O_{Dolomite}$, maximum time step, minimum time step, lake volume tolerance, $\delta^{18}O$ tolerance, and chloride concentration (moles ℓ^{-1}).

Maximum iterations refers to the number iterations the RKF subroutine is allowed to use while solving (1) and (4) over the designated simulation period. The RKF subroutine calculates and constantly changes the step size used by the program while solving the differential equations. The number of iterations necessary to simulate a designated block of time is therefore not known in advance. To prevent overflow of the solution arrays, the maximum number of iterations must be limited. Values used have been 6000 for Owens Lake and 25000 for China and Searles Lake. The logic used in assigning these values involves an "optimization process", i.e., allowing enough iterations for the RKF algorithm to simulate the entire designated time period but not allowing so many as to allow overflow of the solution arrays. Both the array sizes and number of iterations must be limited to maintain a reasonably sized FORTRAN executable file.

Lake volume is entered in units of m^3. Particular care should be taken when entering and interpreting volumes for China and Searles Lake. A zero volume for China Lake may mean that, A) China Lake is dry, or, B) the volume for Searles Lake is greater than or equal to 65.87 m^3 and the two lakes have coupled. If the Searles volume at the start of the simulation is greater than or equal to 65.87 m^3, the model will automatically assign a volume of zero to China Lake regardless of the number that is entered.

$\delta^{18}O$ values should be entered as $\delta_{Dolomite}$. All of the modeling calculations however are actually done using δ_{H2O}. The final values are then converted and output as $\delta_{Dolomite}$ to allow comparison to the Searles $\delta^{18}O$ history.

Maximum and minimum time step are the limits imposed on the RKF step-size calculations. If the maximum step size is too large, the program may use a large number of iterations reducing the step size to control the error in the calculated derivative. The minimum step size need not be unreasonably small or again iterations may be wasted. It must, however, be small enough to allow the RKF algorithm to iterate to a solution when the derivative is changing rapidly. The maximum and minimum time steps used when modeling Owens Lake were 250 and 10 years respectively. The corresponding values for the China and Searles Lake system were 100 and 0.1 years. These smaller values were used because, given its "pancake" bathymetry, China Lake tended to experience relatively rapid volume and $\delta^{18}O$ fluctuations.

Lake volume tolerance and $\delta^{18}O$ tolerance are the maximum error size allowed in the calculation of the corresponding derivative. If these limits are too restrictive, the simulation will be solved in unreasonably small time steps. The RKF algorithm will not iterate to a solution, however, if these limits are not restrictive enough. When the calculated error exceeds the tolerance, the RKF routine decreases the step size to reduce the error. Values for these parameters were determined by observing the errors calculated by the model during actual simulations. The lake volume tolerance was set at $5x10^3$ and the $\delta^{18}O$ tolerance was set at $1x10^{-3}$. Chloride concentration is the concentration entered as moles per liter at the beginning of the simulation.

At the start of each simulation the default value for each of these eight parameters is the value assigned in the previous simulation. The model then prompts the user, asking if changes are to be made. Final values for each of the parameters are displayed at the end of the simulation. These values should then be input at the start of the next simulation if the modeling is to continue from that point in time.

To match the chloride deposition history at Searles Lake, a beginning value for the mass of chloride deposited per square meter at KM-3 must be entered at the start of each simulation. This value is usually taken from the output of the previous simulation or from the actual depositional history, depending on the particular simulation. All of the chloride deposited over the course of the simulation is then added to this initial value. The default value for this parameter is zero.

The model allows the user to choose from a variety of inflow functions. Possible choices include the following simple functions; linear, exponential, logarithmic, power, sinusoidal, step, and zero inflow. The user may also choose from among three steady-state derived histories. These include a variable temperature history, constant high temperature history, and a constant low temperature history. Each inflow history was derived using a steady-state model (SS.FOR) and has been previously described in this section. The constant temperature histories are to be used in conjunction with the constant parameter option. If one of the constant temperature histories are chosen, the constant parameter option should be used. Three constant parameter options are available and are prompted for in the model; UPPER LIMIT, to be used with the constant high temperature history; LOWER LIMIT, to be used with constant low temperature history; and CUSTOM, which may be used with either. The UPPER LIMIT and LOWER LIMIT options read values for temperature, precipitation, gross evaporation,

δ_A, δ_I, and δ_P from existing data files. Derivation of the values in the LOWER LIMIT and UPPER LIMIT files was described in the steady-state modeling section above. The CUSTOM option allows the user to input the values for these parameters from the terminal. If the variable temperature inflow history is chosen, the values for the corresponding parameters are read from data files. Unique data files exist for Owens and Panamint Lake, and Lake Manly. Searles and China Lake, however, share history files. Sensitivity analysis showed that wide variations in relative humidity are necessary for the model to match the Searles $\delta^{18}O$ data. For this reason humidity is not treated as a constant during the constant temperature simulations. A single humidity history is used for all of the lakes.

Simulations may be run with or without graphical output to the screen. All of the graphics routines in the code are written to run on the VAX computer in conjunction with the DISPLAY software graphics package. Modifications would be necessary for any variations in hardware or software.

Values for parameters necessary to restart the modeling process may be saved at any point in time during a simulation. The model prompts for this option, asking for the number of restart points desired and the corresponding times. This feature is useful for "fine tuning" the end of a simulation without having to rerun the entire simulation.

The remaining parameters necessary to begin a simulation are a beginning time and an ending time. These should define a continuous block of time over which the simulation is to proceed. Present day is designated as time 0 and any positive number represents a corresponding number of years in the past. For example, to model from ten thousand in the past to the present, 1 x 10^4 would be entered for the beginning time and 0 would be entered for the ending time.

To proceed with the modeling, the main program first calls the RKF subroutine. This subroutine solves (1) and then (4) for Owens Lake for each time step of the numerical simulation using the Runge-Kutta-Fehlberg difference equation method (Fehlberg, 1970). Intermediate values for (1) and (4) are actually calculated at six different points within a given time step in a function subprogram, F(t). Each derivative is calculated first at the beginning of the time step, then at points 1/4, 3/8, 12/13, at the end, and 1/2 of the way through the time step. Using these intermediate values, the RKF subroutine determines whether calculating dV/dt and $d\delta/dt$ over the current step size will result in an unacceptably large error. If the calculated error is greater than the designated tolerance, the RKF subroutine incrementally decreases the step size within the given limits. If after the step size is decreased the error becomes less than the tolerance, the results are written to an array and the process moves forward. At the end of a successful iteration, the subroutine will increase the step size, attempting to speed the simulation process.

The RKF algorithm assumes the function for which the derivative is calculated is continuous. Unfortunately, volume calculated as a function of time, VOL(t), contains two singularities. These occur at the points where the lake desiccates or exceeds maximum volume and begins to overflow. To avoid simulating across these singularities, a logical variable, ZEROCHK, is used to detect their approach and allow the model to "step across" the singularity. If the calculated volume at any of the six intermediate points within the time step is zero and the step size is the minimum allowable step size, ZEROCHK is set to TRUE. When ZEROCHK is TRUE, the model sets the volume to zero without calculating an error and comparing it to the tolerance as would normally be done. In this way the solving routine comes as close to the singularity as the minimum step size will allow while preventing the model from "crashing" at the singularity.

When the calculated lake volume exceeds maximum volume as determined by bathymetric data, dV/dt is zero and the lake begins to overflow. At this point, however, portions of the model are "lied to",i.e., dV/dt is not allowed to equal zero. The RKF subroutine solves (1) and from that calculates a new volume. Because dV/dt is not set to zero, this volume grows beyond the known maximum volume as long as the derivative is positive. The volume written to the solution array, however, is the known maximum volume. When the calculated derivative becomes negative, the lake volume is reduced. By doing this, the part of the model that solves (1) "sees" the function as continuous. The excess volume at the end of each time step is the amount of water that flows into the next lake over that time step. In reality dV/dt would be zero when the lake begins to overflow. Because the term dV/dt is found in (4), the portion of the model that calculates (4) is "told" that dV/dt is zero as long as the calculated volume indicates that the lake is at overflow. This continues until the calculated volume is less than the maximum volume.

The parameters necessary to solve (1) and (4) at the intermediate points within the time step are calculated in the function subprogram F(t). This function first determines values for δ_I, δ_P, δ_A, temperature, evaporation, and precipitation at a designated point in time, t. If the simulation is being run with constant parameters, these values have already been assigned in the main program. For variable temperature simulations, these values are calculated by linearly interpolating between data points in the histories read in the main program. These histories are read as 1-D arrays, and each value of a particular parameter has a corresponding time array element, e.g., $\delta_P(40)$ corresponds to time(40). The subroutine FINDT is used to determine where the current time, t, lies in relation to the time array elements. Using a binary search routine, the subroutine may find, for example, that t is between time(40) and time(41). This in turn means that the current value of δ_P is between $\delta_P(40)$ and $\delta_P(41)$ and likewise for the other parameters. The values for these parameters are then calculated by linear interpolation in the subroutine OINTERP. Because the humidity history and inflow history have unique chronologies, this process is repeated using two subroutines, FINDHT and HUMTERP, for humidity, and two subroutines, FINDQT and QINTERP, for inflow. The current surface area of Owens Lake is calculated in the subroutine OAREA, which uses bathymetric data to convert a given volume to surface area. As explained above, this volume (VOL) may exceed the known maximum volume (VOLMAX), indicating that the lake is overflowing. The volume of water (m^3) that flows out of Owens Lake during the time step is: V_{OUT} = VOL-VOLMAX. The precipitation flux, Q_P, is given by (23).

The next parameters calculated are those related to the chloride mass balance. The total moles of chloride (CL) in Owens Lake at the start of a time step is either given at the beginning of the simulation (for the first time step) or known from the last point solved in the RKF subroutine. The model must then calculate the total moles of chloride (TCL) at points 1/4, 3/8, 12/13, at the end, and 1/2 of the way through the time step. The total moles of chloride at each of these points is given by:

$$TCL = (OTIME - t)\, QCL + CL - V_{OUT} \frac{10^3\, \ell}{m^3}\, CONC_{CL} \qquad (27)$$

Where OTIME is the time at the beginning of the time step, t is the time at each intermediate point, QCL is the chloride flux (1.67×10^8 moles yr^{-1}) into Owens Lake, and $CONC_{CL}$ is the chloride concentration (moles ℓ^{-1}) at the beginning of the time step. If the chloride concentration at any time t within the time step exceeds 6.1 moles ℓ^{-1}, the model sets the concentration at 6.1 moles ℓ^{-1}. This would indicate that halite precipitated at that time. Geologic evidence indicates that no halite was precipitated in Owens Lake during the time period covered by the modeling. Chloride saturation in Owens Lake should therefore not exceed 6.1 moles ℓ^{-1} during a successful simulation.

The chemical activity of water is given by (12). The function subprogram FPHI, which calculates Φ, the osmotic coefficient for water, was adapted from Pitzer and Kim (1974). It should be noted that this routine has been modified to work with halite only. Q_E is given by (10) and Q_C is given by (11). The isotopic enrichment factor, ϵ_V, is calculated in the function subprogram FEPS using (14). The del of the back-condensation is given by (13). The del of the evaporation is calculated in the function subprogram DELE using (15). (2) and (4) are then solved and these values are returned to the RKF subroutine.

Once the Owens Lake simulation is complete, the process is repeated for China and Searles Lake using the subroutine RKF_CS and the function subprogram F_CS. Because these lakes couple and decouple over time, they must be simulated simultaneously. If China and Searles Lake are coupled, the flux out of Owens Lake is the flux to Searles Lake. Inflow into China/Searles is calculated by linearly interpolating between outflow data points saved as an array, OQO, during the Owens simulation. This linear interpolation process is the same as that previously described above. The chloride flux out of Owens is saved in the array TIMECL_IN. Chloride flux into China/Searles is also calculated using linear interpolation.

The transient model was not used to simulate periods of chloride deposition at Searles Lake, but is does contain a correction factor which aids in this process. The chloride deposition history for Searles Lake was derived from analysis of the KM-3 core. After developing a chronology for the core, chloride analysis was used to produce a table of cumulative chloride versus age/depth (Jannik, 1989). A small part of this chloride can be attributed to brine which soaked the core during drilling. This chloride was therefore not actually precipitated as halite from Searles Lake, but it does show up in the history. To account for this added chloride in the history, the model deposits a small amount of chloride over each time step regardless of the chloride saturation in Searles Lake. The calculated correction factor, SLTCONST, is 1.27×10^{-2} kg m^{-2} yr^{-1}. The actual chloride history-matching process is described later in this section.

Equations (2) and (4) are not used in the Panamint and Lake Manly simulations. $\delta^{18}O$ histories are not simulated for Panamint Lake or Lake Manly. The areas for these two lakes are calculated in the subroutine RKF_CS in the following manner. PDINTERP is a linear interpolation subroutine used to calculate evaporation rates from histories read in the main program. When the evaporation rate is determined, the surface area for each lake is given by:

$$A = \frac{Q_I}{q_E} \tag{28}$$

The final calculation in the transient model is the calculation of total surface area for the paleo-Owens River system. This is done by finding the change in surface area over the current time step for each lake. This change is added it to the previous surface area and the surface areas are then summed.

The transient modeling using the variable temperature history was begun at 1.3487 Mya. Because the model is not sensitive to initial conditions, the starting parameters were chosen arbitrarily. Owens and China Lake were both at overflow and the δ values were 25 ‰ and 32 ‰ respectively. The starting volume for Searles Lake was 65 x $10^9 m^3$, and the δ value was 32 ‰. Chloride concentration for each of the three lakes above was set at 2.0 x 10^{-4}moles/liter. Both Panamint Lake and Lake Manly were dry at the start of the simulation.

To guide the modeling process, a table which detailed the salt deposition history at Searles Lake was produced (Table ?). The transient modeling was performed in discrete blocks, simulating intervals between these tabulated periods of salt deposition, that is, intervals during which the model could match the Searles isotopic history. Starting with the "initial guess" inflow values, adjustments were made until the transient model output matched Searles O^{18} values and the modeled lake was at or very near halite saturation (6.1 moles/liter) immediately preceding a known period of salt deposition. If Searles Lake was not quite at saturation at the end of the simulation period, the volume at saturation was calculated by:

where VOL_{END} and $CONC_{END}$ were the volume and chloride concentration output by the transient model at the end of a simulation period.

Periods of salt deposition were not modeled with the transient code. Simple mass balance calculations were done by hand and the volume of Searles Lake was adjusted accordingly, giving the starting parameters for the next non-depositional period. The tabulated salt deposition history gave the observed salt deposition ($SALT_{OBS}$) at the site of the KM-3 core in Searles Lake and also the length of the depositional episode (Δt). The mass of salt per square meter that flowed in during the depositional period was calculated by:

$$SALT_{INFLOW} = \Delta t \times 0.19 \frac{kg}{m^2\,yr} \tag{30}$$

where 0.19 kg $m^{-2}yr^{-1}$ was the average chloride accumulation rate at KM-3 (Jannik, 1989). The additional mass of salt per square meter necessary to match the observed salt deposition at KM-3 was given by:

$$SALT_{ADD} = SALT_{OBS} - SALT_{INFLOW} \tag{31}$$

If $SALT_{ADD}$ had a positive value, VOL_{SAT} was decreased by the amount necessary to precipitate the mass of salt per square meter equal to $SALT_{ADD}$. If $SALT_{ADD}$ had a negative value, VOL_{SAT} was increased by the amount necessary to accommodate the mass of salt equal to $SALT_{ADD}$ and remain at halite saturation. Because salt was not deposited evenly over the Searles playa, an additional correction was necessary to determine the mass of salt that would be deposited per square meter at KM-3 for a given volume of Searles Lake at halite saturation. This correction was calculated by:

$$SALT_{CORR} = \frac{\dfrac{0.19\,kg}{m^2\,yr}}{\dfrac{6.7\times10^6\,kg}{yr}} = 3.22\times10^{-8}\,m^{-2} \tag{32}$$

where $5.9\text{x}10^6$ kg yr^{-1} was the approximate chloride load of the Owens River over the past 20 kyr (Smith, 1976). The new volume (m^3) was given by:

$$VOL_{NEW} = \frac{SALT_{ADD}}{\left[\dfrac{1000\,\ell}{m^3}\;\dfrac{6.1\,moles}{\ell}\;\dfrac{35.435\,g}{mole}\;\dfrac{1\,kg}{1000\,g}\;SALT_{CORR}\right]} \tag{33}$$

VOL_{SAT} was changed linearly over Δt such that VOL_{SAT} equaled VOL_{NEW} at the end of the depositional period.

To start the transient model at the beginning of the next simulation period it was necessary to input values for volume, chloride concentration, and δ_{WATER} for Owens, China, and Searles Lake. These values for Owens and China Lake were taken from the final output for the previous simulation period. The starting volume for Searles Lake was VOL_{NEW}, chloride concentration was set at 6.1 moles/liter, and δ_{WATER} was the final value from the previous simulation. Surface area for Searles Lake was calculated from VOL_{NEW} using bathymetric data for Searles. A simplified version of the steady-state model was used to calculate a value for Q_I that was in equilibrium with VOL_{NEW}, allowing the simulation to begin under stable conditions.

In addition to the modeling described above, the transient model was also used to investigate the system response to variations in the inflow frequency. To ensure that the observed responses were due only to variations in the inflow, the model was run with constant parameters until equilibrium was achieved. Starting with a constant inflow (B) of 9.434×10^7 m^3 yr^{-1}, the change in inflow with time was given by:

$$Q_I = B + A \, SIN(C\,t) \tag{34}$$

Where A, the amplitude, was 2.82×10^7 m^3 yr^{-1}, C was the designated frequency, and t was time. Analysis of the signal response showed that the lag time for the $\delta^{18}O$ response increased from 24 to 70 years as the inflow period ($2\pi/C$) increased from 90 to 675 years. As the period increased from 675 to 9000 years, the $\delta^{18}O$ lag time decreased linearly from

70 to -565 years. The negative lag times indicated that the $\delta^{18}O$ equilibrated prior to the lake volume equilibration.

The amplitude of the $\delta^{18}O$ response increased from 0.125 to 0.85 ‰ as the inflow period increased from 90 to 1575 years. As the inflow period increased from 1575 to 9000 years, the amplitude of the $\delta^{18}O$ response decreased from 0.85 to 0.5 ‰.

The system response to a step change in inflow is shown in figure ?. The calculated response time was approximately 900 years, which is greater than the average spacing of the data points in the Searles $\delta^{18}O$ history. This would indicate that all major changes in the state of the system would be represented in the present $\delta^{18}O$ history.

```
********************************************************************************
********************************************************************************
**                              PROGRAM SS.FOR                                **
********************************************************************************
********************************************************************************

      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      LOGICAL OVER,DELCHK,TEMPCHK,HUMCHK

      DOUBLE PRECISION OQI(100),TEMPSL(50),SUMAREA(36,100,60),
     +     DELDOL_S(36,100,60),HUM(60),ISO(500),STIME(500),
     +     STEMP(2000),WTIME(2000)

      PARAMETER(AMAX_O=0.694D9,AMAX_C=0.155D9,AMAX_S=0.994D9,
     +     CONST_K=6.8D-3,AMAX_P=0.727D9,AMAX_D=0.583D9)

      WRITE(6,*)
      WRITE(6,*)'ENTER THE ACTIVITY OF WATER IN SEARLES LAKE'


      READ(5,*)AW

      WRITE(6,*)
      WRITE(6,*)'ENTER STARTING VALUE FOR QI'
      READ(5,*)STARTQI

      WRITE(6,*)
      WRITE(6,*)'ENTER ENDING VALUE FOR QI'
      READ(5,*)ENDQI

      HS=-2.2D0

      HI=135.D0

      QIINC=(ENDQI-STARTQI)/30.D0

      TEMP_I=6.25D0

      DO 100 I=1,36
         TEMP_I=TEMP_I+0.25D0
**         TEMP_I=15.2D0
         TEMPO=TEMP_I
         OQI_I=STARTQI
         TEMPSL(I)=TEMPO+3.6D0

         DO 200 J=1,100
            OQI_I=OQI_I+QIINC
            OQI(J)=OQI_I

            DO 300 K=1,60
               HUM(K)=DBLE(30.D0+K)/100.D0

*****************************
** OWENS LAKE CALCULATIONS **
*****************************
*****************************************************************
** CALCULATE ALL PARAMETERS THAT ARE A FUNCTION OF TEMPERATURE **
*****************************************************************

** DEL OF THE INFLOW **

               ODELI=TEMPO*0.289874D0 - 20.74367D0
```

```
** DEL OF THE ATMOSPHERE (SAME FOR ALL THE LAKES) **

              DELA=TEMPO*2.898861D-1 - 35.79326D0

**  DEL OF THE PRECIPITATION **

              ODELP=TEMPO*2.915851D-1 - 1.60108D1

**  PRECIPITATION **

              OPRECIP= -1.6353711D-2*TEMPO + 4.07238D-1
              IF(OPRECIP .LT. 0.D0)OPRECIP=0.D0

**  EVAPORATION **

              OEVAP=1.46896D-1*TEMPO - 6.37164D-1
              IF(OEVAP .LT. 1.35D0)OEVAP=1.35D0

**  EPSILON , THE ISOTOPIC ENRICHMENT FACTOR **

              EPS=FEPS(TEMPO)
              ALPHA=1.D0 + EPS/1.D3

**  DEL OF THE BACK-CONDENSATION **

              ODELC=EPS*(1.D0+(DELA/1.D3))+DELA

*****************************************
** NOW CALCULATE THE REST OF THE STUFF **
*****************************************

** CALCULATE AREA OF OWENS **

              AREA_O=OQI(j)/(OEVAP*(1.D0-HUM(K))-OPRECIP)
              IF (AREA_O .GT. AMAX_O)THEN
                 AREA_O=AMAX_O
              ENDIF

** GROSS EVAPORATION FLUX, QE (M^3/YR) **

              OQE=OEVAP*AREA_O

** PRECIP FLUX, QP (M^3/YR) **

              OQP=OPRECIP*AREA_O

** BACK-CONDENSATION FLUX, QC **

              OQC=OQE*HUM(K)

** OVERFLOW FLUX **

              OQO=OQI(j)+OQP+OQC-OQE

              IF(OQO .LE. 0.D0)OQO=0.D0

** KINETIC ISOTOPIC ENRICHMENT FACTOR, EPS_K **

              EPS_K=(ALPHA*(1.D0-CONST_K*HUM(K))/(1.D0-CONST_K)-
     +             1.D0)*1.D3

** DEL OF THE LAKE **

              ODELL=(OQI(J)*ODELI+OQP*ODELP+OQE*(HUM(K)*ODELC+EPS_K))/
     +             (OQI(J)+HUM(K)*OQE+OQP)
```

```
** CONVERT DEL WATER TO DEL DOLOMITE **

              DELDOL_O=FDDOL(TEMPO,ODELL)

** FLUX OUT OF OWENS EQUALS FLUX INTO CHINA **

              CQI=OQO

** SET DEL OF OWENS EQUAL TO THE DEL OF THE INFLOW TO CHINA LAKE **

              CDELI=ODELL

******************************
** CHINA LAKE CALCULATIONS **
******************************
*******************************************************************
** CALCULATE ALL PARAMETERS THAT ARE A FUNCTION OF TEMPERATURE **
*******************************************************************
              DELA=DELA+5.D0
              TEMPC = TEMPO+3.6D0

** DEL OF THE PRECIPITATION **

              CDELP=TEMPC*2.89886D-1 - 1.47368D1

** PRECIPITATION **

              CPRECIP=TEMPC*(-1.62597D-2)+4.05687D-1
              IF(CPRECIP .LT. 0.D0)CPRECIP=0.D0

**  EVAPORATION **

              CEVAP=1.46896D-1*TEMPC - 6.37164D-1
              IF(CEVAP .LT. 1.35D0)CEVAP=1.35D0

**  EPSILON , THE ISOTOPIC ENRICHMENT FACTOR **

              EPS=FEPS(TEMPC)
              ALPHA=1.D0 + EPS/1.D3

**  DEL OF THE BACK-CONDENSATION **

              CDELC=EPS*(1.D0+(DELA/1.D3))+DELA

******************************************
** NOW CALCULATE THE REST OF THE STUFF **
******************************************

** CALCULATE AREA OF CHINA (HA,HA,HA) **

              AREA_C=CQI/(CEVAP*(1.D0-HUM(K))-CPRECIP)

              IF (AREA_C .GT. AMAX_C)THEN
                 AREA_C=AMAX_C
              ENDIF

** GROSS EVAPORATION FLUX, QE (M^3/YR) **

              CQE=CEVAP*AREA_C

** PRECIP FLUX, QP (M^3/YR) **

              CQP=CPRECIP*AREA_C

** BACK-CONDENSATION FLUX **
```

```
            CQC=CQE*HUM(K)

** OVERFLOW FLUX **

            CQO=CQI+CQP+CQC-CQE

            IF(CQO .LE. 0.D0)CQO=0.D0

** KINETIC ISOTOPIC ENRICHMENT FACTOR, EPS_K **

            EPS_K=(ALPHA*(1.D0-CONST_K*HUM(K))/(1.D0-CONST_K)-
     +           1.D0)*1.D3

** DEL OF THE LAKE **

            IF(CQI .GT. 0.D0)THEN

               CDELL=(CQI*CDELI+CQP*CDELP+CQE*(HUM(K)*CDELC+EPS_K))/
     +              (CQI+HUM(K)*CQE+CQP)

            ENDIF

** CONVERT DEL WATER TO DEL DOLOMITE **

            DELDOL_C=FDDOL(TEMPC,CDELL)

** FLUX OUT OF CHINA EQUALS FLUX INTO SEARLES **

            SQI=CQO

** SET DEL OF CHINA EQUAL TO THE DEL OF THE INFLOW TO SEARLES LAKE **

            SDELI=CDELL

********************************
** SEARLES LAKE CALCULATIONS **
********************************
*******************************************************************
** CALCULATE ALL PARAMETERS THAT ARE A FUNCTION OF TEMPERATURE **
*******************************************************************

            TEMPS=TEMPC

** DEL OF THE PRECIPITATION **

            SDELP=TEMPS*2.89886D-1 - 1.47368D1

** PRECIPITATION **

            SPRECIP=TEMPS*(-1.62597D-2)+4.05687D-1
            IF(SPRECIP .LT. 0.D0)SPRECIP=0.D0

**  EVAPORATION **

            SEVAP=1.46896D-1*TEMPS - 6.37164D-1
            IF(SEVAP .LT. 1.35D0)SEVAP=1.35D0

**  EPSILON , THE ISOTOPIC ENRICHMENT FACTOR **

            EPS=FEPS(TEMPS)
            ALPHA=1.D0 + EPS/1.D3

**  DEL OF THE BACK-CONDENSATION **

            SDELC=EPS*(1.D0+(DELA/1.D3))+DELA
```

```
** KINETIC ISOTOPIC ENRICHMENT FACTOR, EPS_K **

               EPS_K=(ALPHA*(1.D0-CONST_K*HUM(K))/(1.D0-CONST_K)-1.D0)
     +              *1.D3

*******************************
** CALCULATE AREA OF SEARLES **
*******************************

               AREA_S=SQI/(SEVAP*(1.D0-HUM(K)/AW)-SPRECIP)
               IF(AREA_S .LT. 10.D0)AREA_S=0.D0

               OVER=.FALSE.

               IF(AREA_S .GT. 0.994D9)THEN
                  AREA_S=0.994D9
                  OVER=.TRUE.
               ENDIF

** GROSS EVAPORATION FLUX **

               SQE=SEVAP*AREA_S*AW

** PRECIP FLUX **

               SQP=SPRECIP*AREA_S

** BACK-CONDENSATION FLUX **

               SQC=SQE*HUM(K)/AW

** OH MY, WHICH EQUATION DO I USE **

** SEARLES OVERFLOWING **

               IF(OVER)THEN
                  SQI=OQO
                  SDELI=ODELL

                  SDELL=(SQI*SDELI+SQP*SDELP+SQE*(HUM(K)*SDELC+
     +                 AW*EPS_K))/(SQI+HUM(K)*SQE+SQP)
                  SQO=SQI+SQP+SQC-SQE
                  AREA_C=0.D0

** SEARLES NOT OVERFLOWING, SEARLES AND CHINA COALESCED **

               ELSE IF(AREA_S.LE.0.994D9.AND.AREA_S.GE.0.715D9)THEN
                  SQI=OQO
                  SDELI=ODELL
                  SDELL=(SQI/SQE)*SDELI+EPS_K+HUM(K)*SDELC/AW+(SQP/SQE)
     +                 *SDELP

                  SQO=0.D0
                  AREA_C=0.D0

** SEARLES NOT OVERFLOWING, SEARLES AND CHINA NOT COALESCED **

               ELSE IF(AREA_S.LT.0.715D9.AND.AREA_S.GT.0.D0)THEN
                  SQI=CQO
                  SDELI=CDELL
                  SDELL=(SQI/SQE)*SDELI+EPS_K+HUM(K)*SDELC/AW+(SQP/SQE)
     +                 *SDELP
                  SQO=0.D0
               ELSE
                  SQO=0.D0
               ENDIF
```

```
** CONVERT DEL WATER TO DEL DOLOMITE **

               IF(AREA_S .GT. 0.D0)THEN
                  DELDOL_S(I,J,K)=FDDOL(TEMPS,SDELL)
               ELSE
                     DELDOL_S(I,J,K)=99.D0
               ENDIF

***************************
** PANAMINT CALCULATIONS **
***************************

               IF(SQO .GT. 0.D0)THEN

                  TEMPP=TEMPS+2.35D0


** EVAPORATION **

                  PEVAP=1.46896D-1*TEMPP - 6.37164D-1
                  IF(PEVAP .LT. 1.35D0)PEVAP=1.35D0

** PRECIPITATION **

                  PPRECIP=TEMPP*(-1.62787D-2)+4.05912D-1
                  IF(PPRECIP .LT. 0.D0)PPRECIP=0.D0

** CALCULATE AREA OF PANAMINT **

                  PQI=SQO

                  AREA_P=PQI/(PEVAP*(1.D0-HUM(K))-PPRECIP)

                  IF(AREA_P .GT. AMAX_P)THEN
                     AREA_P=AMAX_P
                     PQE=PEVAP*AREA_P
                     PQC=PQE*HUM(K)
                     PQP=PPRECIP*AREA_P
                     PQO=PQI+PQC+PQP-PQE
                  ELSE
                     PQO=0.D0
                  ENDIF

*******************************
** DEATH VALLEY CALCULATIONS **
*******************************

                  IF(PQO .GT. 0.D0)THEN

                     TEMPD = TEMPP+4.08D0

** EVAPORTAION **

                     DEVAP=1.46896D-1*TEMPD - 6.37164D-1
                     IF(DEVAP .LT. 1.35D0)DEVAP=1.35D0

** PRECIPITATION **

                     DPRECIP=TEMPD*(-1.62890-2)+4.06423D-1
                     IF(DPRECIP .LT. 0.D0)DPRECIP=0.D0

** CALCULATE AREA OF LAKE MANLY **

                     DQI=PQO
```

```
                  AREA_D=DQI/(DEVAP*(1.D0-HUM(K))-DPRECIP)

                  IF(AREA_D .GT. AMAX_D)THEN
                     AREA_D=AMAX_D
                     DQE=DEVAP*AREA_D
                     DQC=DQE*HUM(K)
                     DQP=DPRECIP*AREA_D
                     DQO=DQAI+DQC+DQP-DQE
                  ELSE
                     DQO=0.D0
                  ENDIF
               ELSE
                  AREA_D=0.D0
               ENDIF
            ELSE
               AREA_D=0.D0
               AREA_P=0.D0
            ENDIF

            SUMAREA(i,j,k)=(AREA_O+AREA_C+AREA_S+AREA_P+AREA_D)/1.D9
 300     CONTINUE
 200   CONTINUE
 100 CONTINUE

** READ CYNDY'S ISOTOPE DATA AND TIMES **

     OPEN(UNIT=99,FILE='REALISO.DAT',STATUS='OLD')

     DO 400 I=1,500
        READ(99,*,END=401)STIME(I),ISO(I)
 400 CONTINUE

 401 NPTS=I-1

     CLOSE(UNIT=99)

     DO 500 I=1,NPTS
        STIME(I)=STIME(I)*1.D6
 500 CONTINUE

** FIND TEMPERATURES CORRESPONDING TO TIMES **

     OPEN(UNIT=98,FILE='SEARLES.UF',STATUS='OLD',FORM=
    +      'UNFORMATTED')

     DO 600 I=1,2000
        READ(98,END=601)STEMP(I),GBG1,GBG2,GBG3
 600 CONTINUE

 601 NSPTS=I-1

     CLOSE(UNIT=98)

     OPEN(UNIT=97,FILE='OWENS.UF',STATUS='OLD',FORM=
    +      'UNFORMATTED')

     DO 700 I=1,NSPTS
        READ(97)WTIME(I),GBG1,GBG2,GBG3,GBG4
 700 CONTINUE

     DO 725 I=1,NSPTS
        WTIME(I)=WTIME(I)*1.D6
 725 CONTINUE

     OPEN(UNIT=96,FILE='SSQI.DAT',STATUS='UNKNOWN')
```

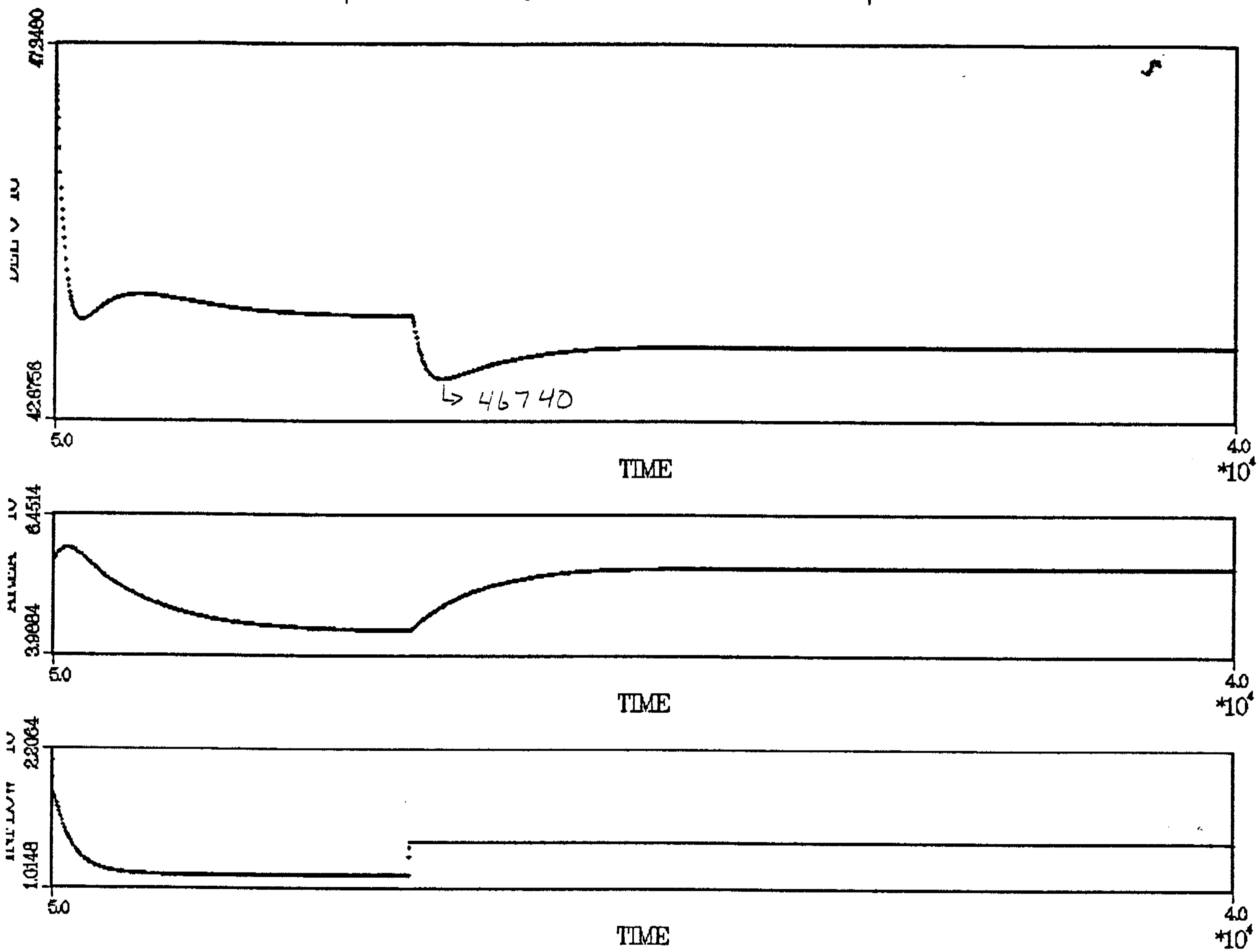
```
      OPEN(UNIT=92,FILE='STEMP.DAT',STATUS='UNKNOWN')
      OPEN(UNIT=50,FILE='OOPS.DAT',STATUS='UNKNOWN')

      DO 800 I=1,NPTS
         TEMPCHK=.FALSE.
         DELCHK=.FALSE.
         HUMCHK=.FALSE.

         CALL FINDT(NSPTS,STIME(I),INDEX,WTIME)
         CALL SINTERP(STIME(I),INDEX,TEMP,STEMP,WTIME)
         WRITE(50,*)
         WRITE(50,*)'TIME',STIME(I)

** WRITE SEARLES TEMP TO FILE **

         WRITE(92,*)STIME(I)/1.D6,TEMP

** CALCULATE HUMIDITY FROM DEL **

         SHUM=DBLE(ANINT(HI+HS*ISO(I)))

** FIND CORRESPONDING HUMIDITY IN ARRAY **

         DO 900 II=1,60
            IF(DABS(SHUM-HUM(II)*100.D0) .LT. 0.01D0)THEN
               WRITE(50,*)'HUMIDITY MATCH',SHUM,HUM(II)
               WRITE(50,*)
               HUMCHK=.TRUE.
               GOTO 901
            ENDIF
 900     CONTINUE

 901     NK=II

** DON'T ENTER THIS LOOP UNLESS THE PROGRAM FOUND A HUMIDITY MATCH **

         IF(HUMCHK)THEN

            DO 1000 II=1,36
               IF(TEMP .GT. TEMPSL(II) .AND. TEMP.LT.TEMPSL(II+1))THEN
                  TEMPCHK=.TRUE.
                  WRITE(50,*)'TEMP MATCH',TEMP,TEMPSL(II),TEMPSL(II+1)
                  WRITE(50,*)
                  TEST1=DABS(TEMP-TEMPSL(II))
                  TEST2=DABS(TEMP-TEMPSL(II+1))
                  GOTO 1001
               ENDIF
 1000       CONTINUE

 1001       IF(TEST1 .GT. TEST2)THEN
               NI=II+1
            ELSE
               NI=II
            ENDIF

** LOCATE DELS IN ARRAY THAT CORRESPOND TO GIVEN DEL **

            CHCK=ISO(I)-DELDOL_S(NI,1,NK)
            WRITE(50,*)'ISO(I)',ISO(I)
            IF(CHCK .LT. 0.D0 .AND. TEMPCHK)THEN
               DO 1100 IJ=1,100
                  CHCK=ISO(I)-DELDOL_S(NI,IJ,NK)
                  IF(CHCK .GT. 0.D0)THEN
                     WRITE(50,*)'DEL MATCH',ISO(I),DELDOL_S(NI,IJ,NK)
                     NJ=IJ
                     DELCHK=.TRUE.
```

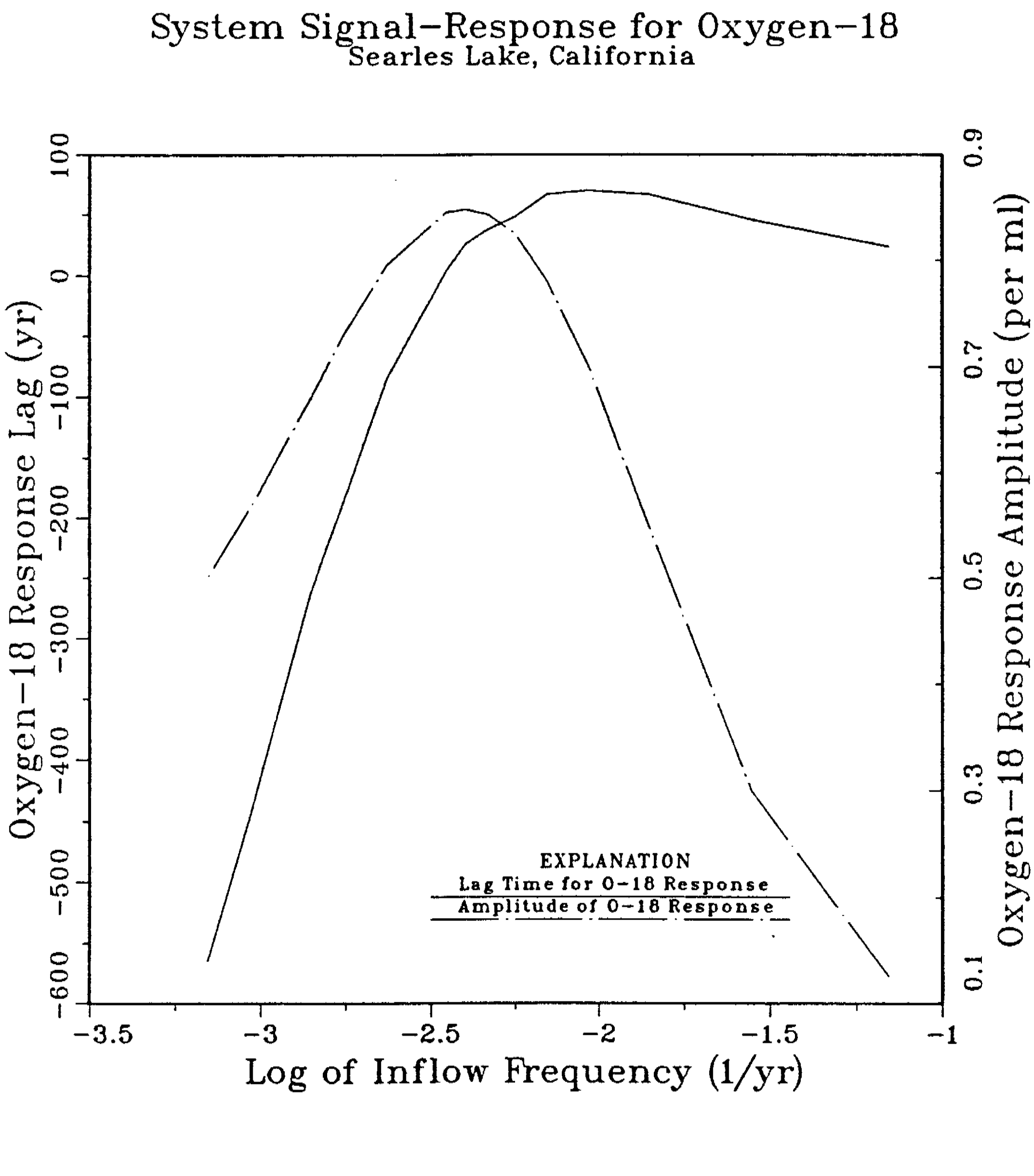
```
                  GOTO 1101
               ENDIF
 1100        CONTINUE
          ENDIF

 1101     IF(DELCHK)THEN
             FQI=(OQI(NJ)-OQI(NJ-1))/(DELDOL_S(NI,NJ-1,NK)-
     +            DELDOL_S(NI,NJ,NK))*(DELDOL_S(NI,NJ-1,NK)
     +            -ISO(I))+OQI(NJ-1)

             FAREA=(SUMAREA(NI,NJ,NK)-SUMAREA(NI,NJ-1,NK))/
     +            (DELDOL_S(NI,NJ-1,NK)-DELDOL_S(NI,NJ,NK))*
     +            (DELDOL_S(NI,NJ-1,NK)-ISO(I))+SUMAREA(NI,NJ-1,NK)
             WRITE(96,*)STIME(I)/1.D6,FAREA,FQI
          ELSEIF(TEMPCHK)THEN
             WRITE(96,*)STIME(I)/1.D6,99,REAL(DELDOL_S(NI,1,NK)),
     +           REAL(DELDOL_S(NI,30,NK)),REAL(ISO(I))
          ELSE
             WRITE(96,*)STIME(I)/1.D6,98,TEMP
          ENDIF
       ENDIF

       IF(.NOT. HUMCHK)THEN
          WRITE(96,*)STIME(I)/1.D6,97,REAL(HUM(1)),REAL(HUM(50)),SHUM
       ENDIF

 800  CONTINUE

      CLOSE(UNIT=96)
      CLOSE(UNIT=92)

      END

*****************************************************************************
*****************************************************************************
*  This is a function subprogram to calculate the isotopic enrichment factor  *
*****************************************************************************
*****************************************************************************

        double precision function feps(temp)
        implicit double precision (a-h, o-z)

**************************
**CONVERT TEMP TO KELVIN**
**************************

        tempk=temp+273.15d0
        feps =(dexp((1.534d0*(1.0d6/(tempk)**2)-3.206d0*(1.0d3/tempk)+
     +      2.644d0)/1.d3)-1.d0)*1.d3
        return
        end



*********************************************************
*********************************************************
**    A function subprogram to  calculate del dolomite **
*********************************************************
*********************************************************

        DOUBLE PRECISION FUNCTION FDDOL(TEMP,DELH2O)
        implicit double precision (a-h,o-z)
```

```
************************************
** CONVERT TEMP TO DEGREES KELVIN **
************************************

        TEMPK = TEMP+273.15D0

**********************************************
** CALCULATE EPSILON FOR DOLOMITE AND WATER **
**********************************************

        EPSDOL=(DEXP((3.2D0*(1.D6/TEMPK**2)-1.5D0)/1.D3)-1.D0)*1.D3

        FDDOL=EPSDOL+DELH2O*(EPSDOL/1.0D3+1.D0)

        RETURN
        END


*******************************************************************************
*******************************************************************************
*          A SUBROUTINE TO FIND THE TIME INDEX WITH A BINARY SEARCH          *
*******************************************************************************
*******************************************************************************

        SUBROUTINE FINDT(NPTS,TM,INDEX,TIME)
        IMPLICIT DOUBLE PRECISION(A-H,O-Z)
        LOGICAL FOUND
        PARAMETER(NUMA=2000)
        DIMENSION TIME(NUMA)
        TFIRST=1
        TLAST=NPTS
        FOUND = .FALSE.
        DO 200 I = 1,100
           IF(TLAST .LT. TFIRST) THEN
              INDEX=TLAST
              GOTO 210
           ENDIF
           IF( TFIRST .LE. TLAST .AND. .NOT. FOUND)THEN
              MIDDLE = (TFIRST+TLAST)/2
              TEST = ABS(TM-TIME(MIDDLE))
              IF( TEST .LT. 1.0D-1) THEN
                 FOUND = .TRUE.
                 INDEX=MIDDLE
                 GOTO 210
              ELSE IF(TM .LT. TIME(MIDDLE))THEN
                 TLAST = MIDDLE-1
              ELSE
                 TFIRST = MIDDLE+1
              END IF
           END IF
200     CONTINUE

210     RETURN
        END


*******************************************************************************
*******************************************************************************
*      A subroutine to linearly interpolate between points in data sets      *
*                    containing Searles Lake parameters                      *
*******************************************************************************
*******************************************************************************

        subroutine sinterp(time,ndx,tstemp,STEMP,WTIME)
```

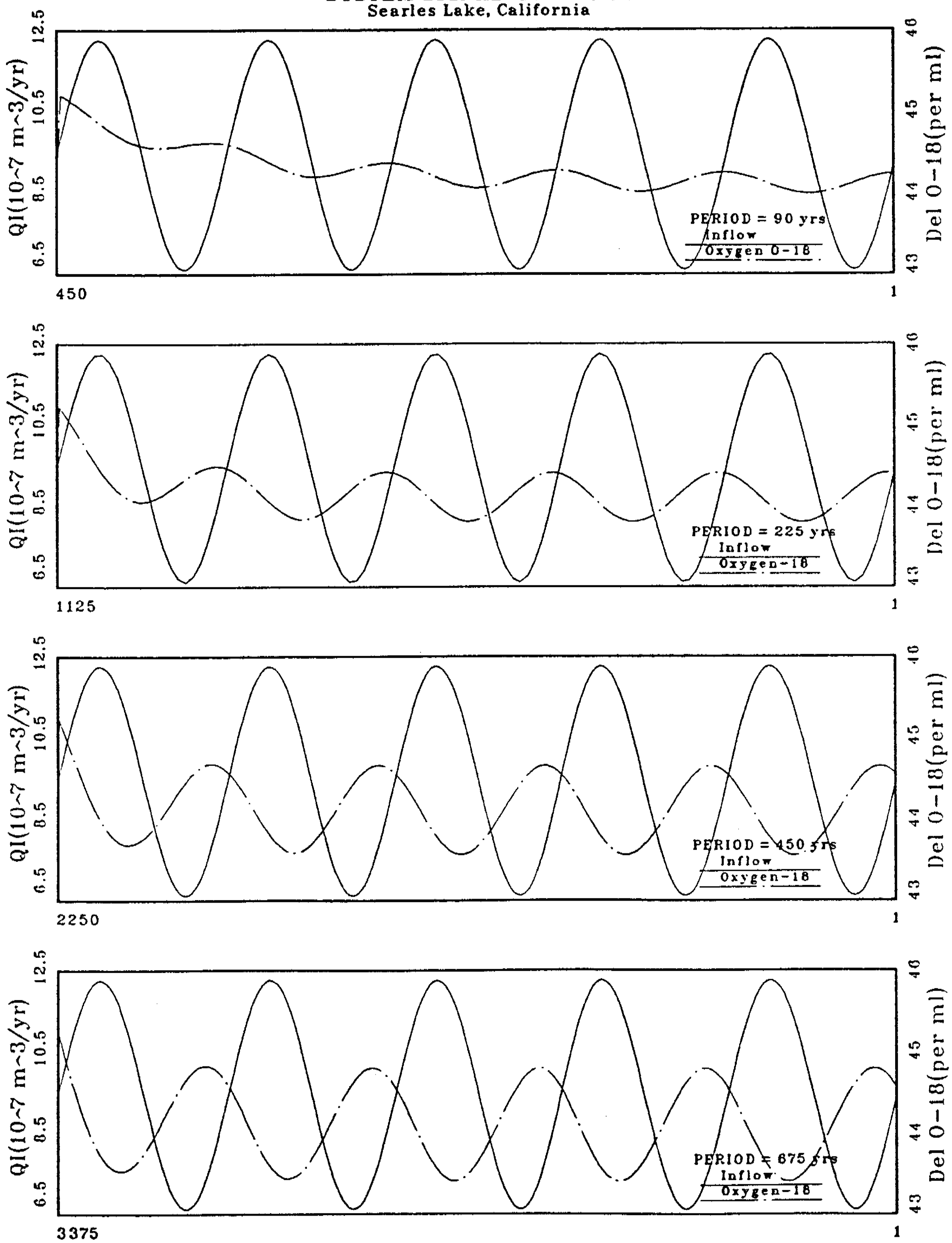
```
      implicit double precision (a-h,o-z)

      DIMENSION WTIME(2000),STEMP(2000)

      tstemp=(stemp(ndx+1)-stemp(ndx))/(wtime(ndx+1)-wtime(ndx))*
     +        (time-wtime(ndx))+stemp(ndx)

      return
      end
```

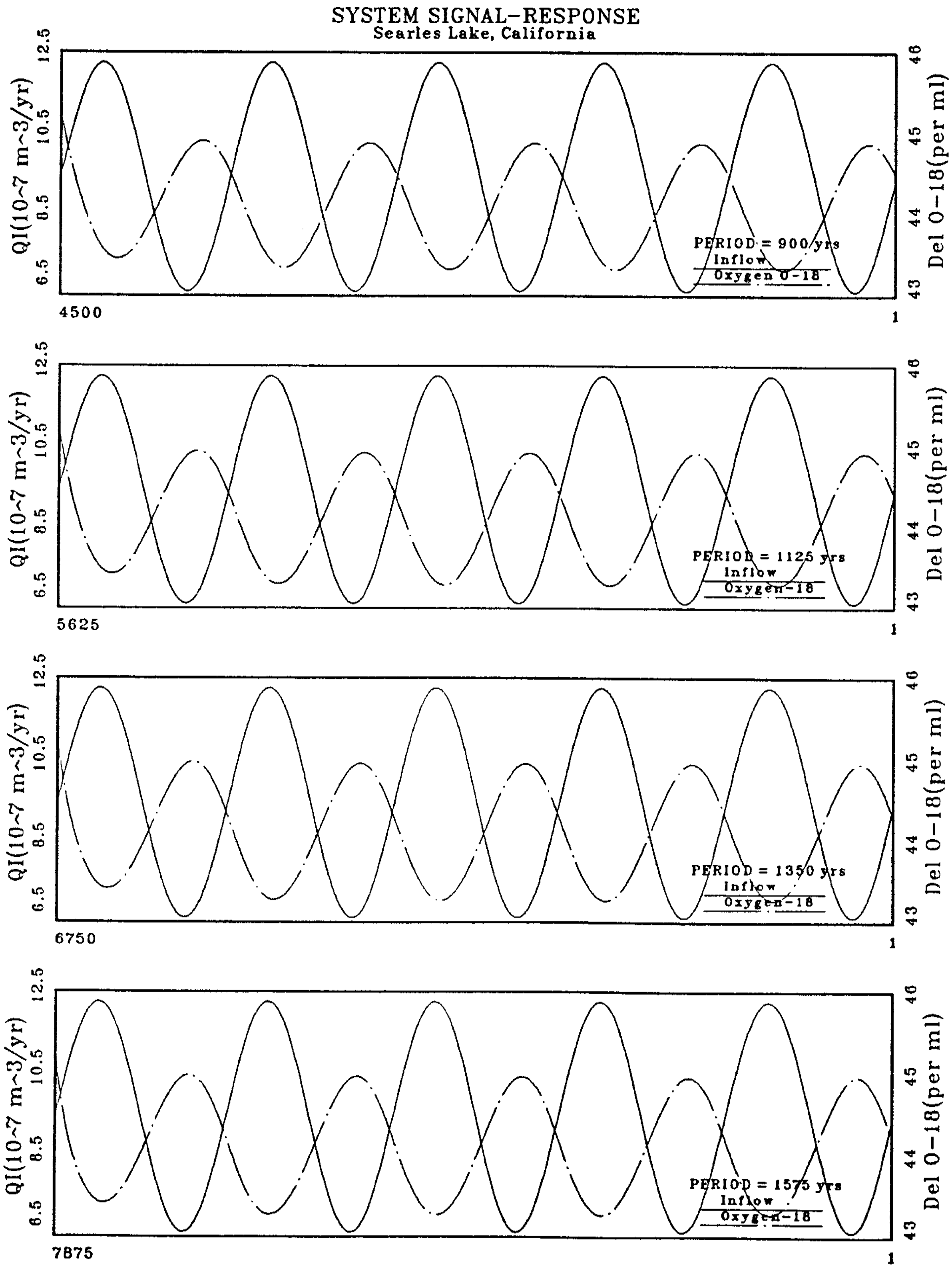
Step change to calc. response time




System Signal−Response for Oxygen−18
Searles Lake, California
Oxygen−18 Response Lag (yr)
100
0
−100
−200
−300
−400
−500
−600
Oxygen−18 Response Amplitude (per ml)
0.9
0.7
0.5
0.3
0.1
−3.5
−3
−2.5
−2
−1.5
−1
Log of Inflow Frequency (1/yr)
EXPLANATION
Lag Time for O−18 Response
Amplitude of O−18 Response

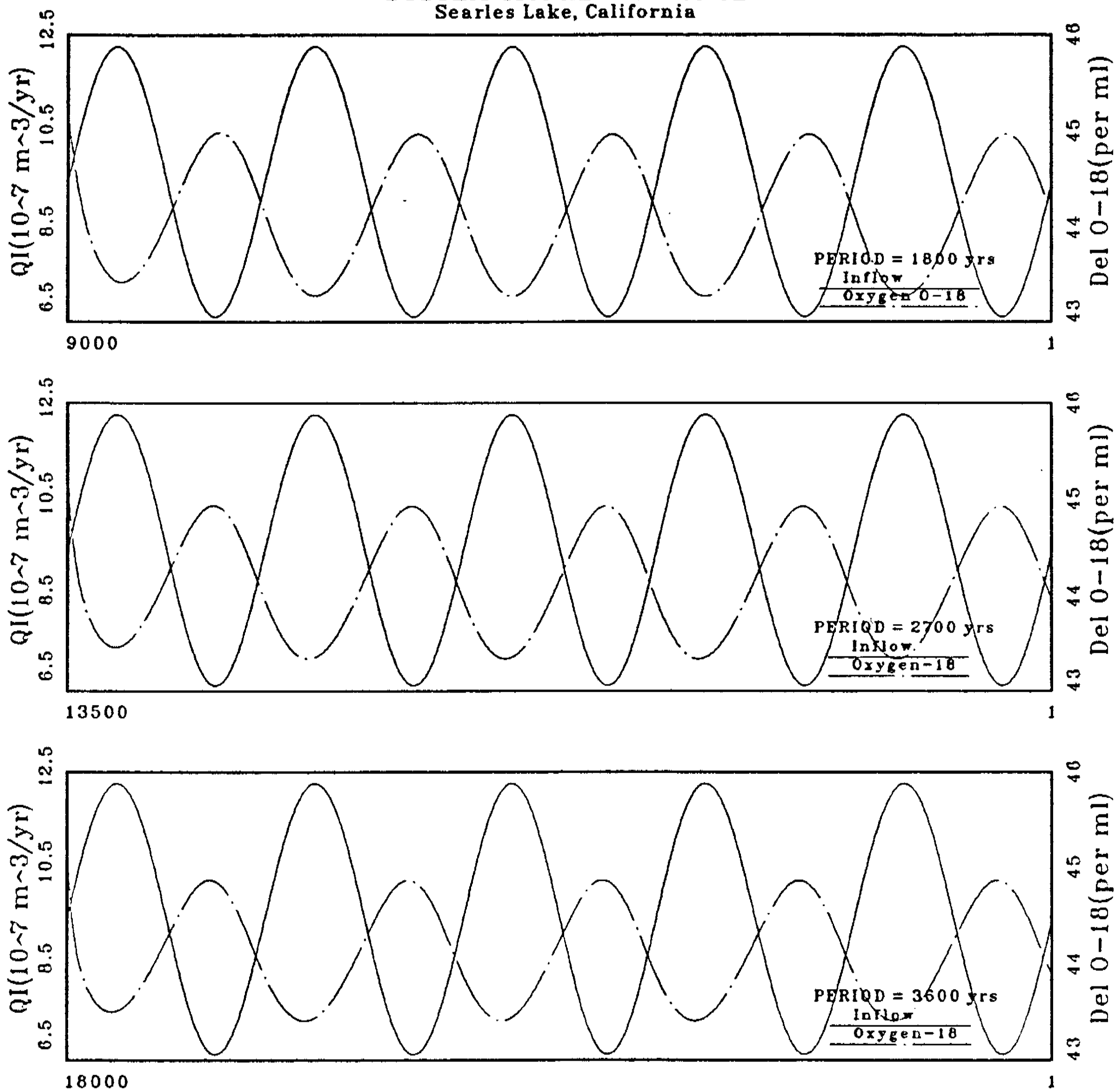
SYSTEM SIGNAL–RESPONSE
Searles Lake, California



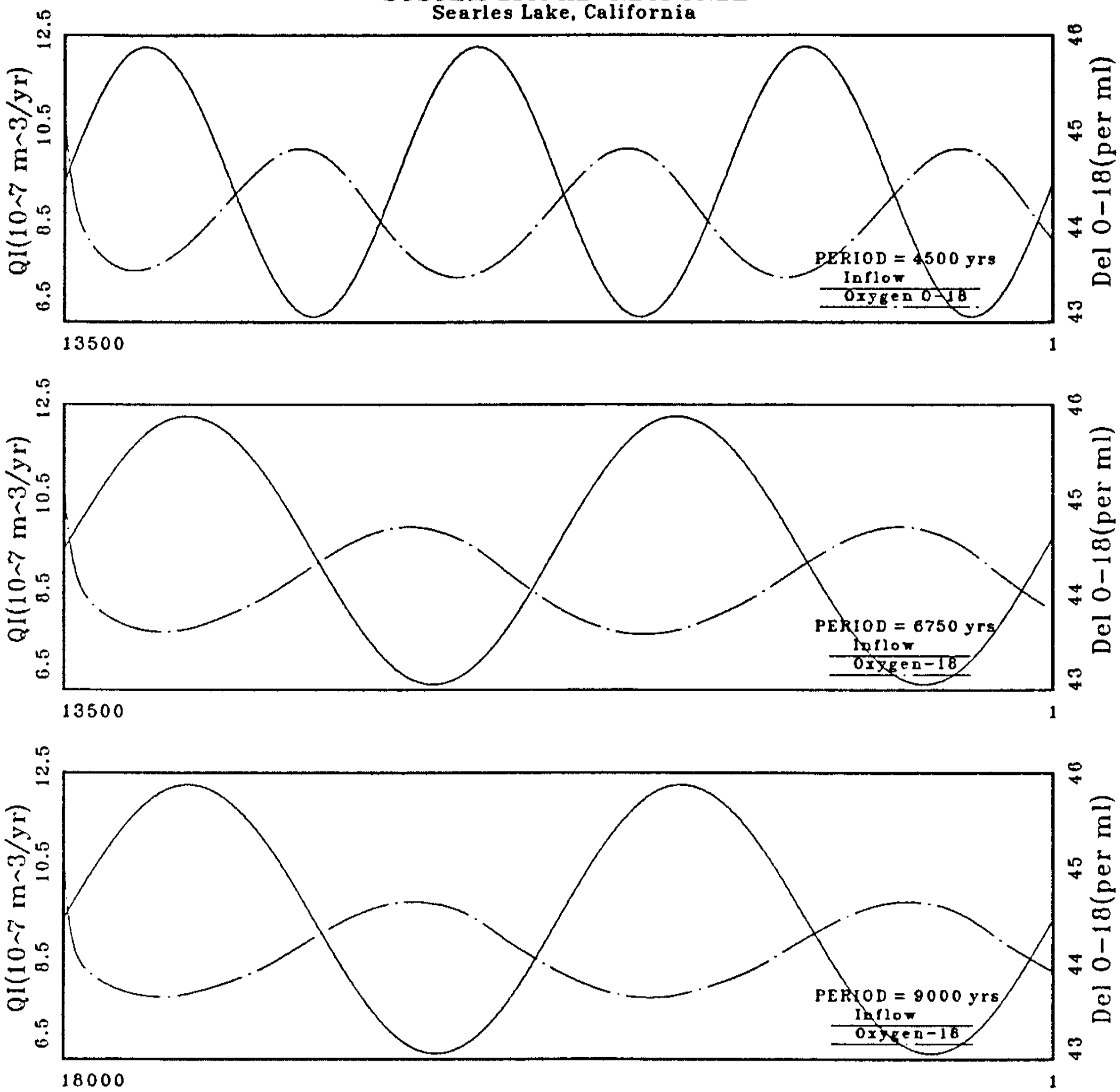


SYSTEM SIGNAL–RESPONSE
Searles Lake, California
QI(10~7 m~3/yr)
Del O–18(per ml)
PERIOD = 900 yrs
Inflow
Oxygen O-18
4500
PERIOD = 1125 yrs
Inflow
Oxygen-18
5625
PERIOD = 1350 yrs
Inflow
Oxygen-18
6750
PERIOD = 1575 yrs
Inflow
Oxygen-18
7875

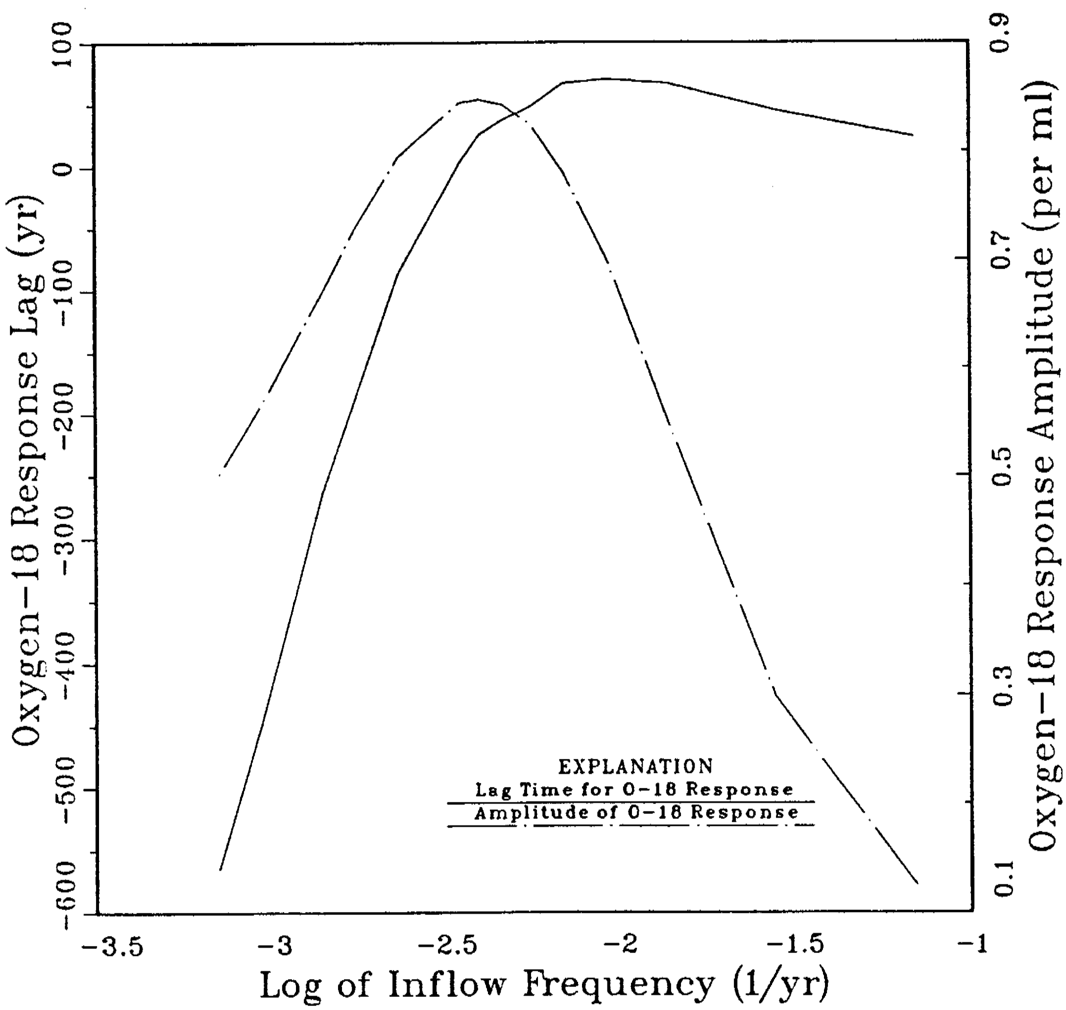
SYSTEM SIGNAL-RESPONSE
Searles Lake, California



SYSTEM SIGNAL–RESPONSE
Searles Lake, California



System Signal–Response for Oxygen–18
Searles Lake, California



```
      PROGRAM           TRANSISO

*******************************************************************************
*******************************************************************************
* THIS PROGRAM CALCULATES CHANGES IN LAKE VOLUME AND ISOTOPIC COMPOSITION    *
* WITH RESPECT TO TIME.                                                      *
*******************************************************************************
*******************************************************************************

*******************************************************************************
*     A PROGRAM TO CALL THE RUNGE-KUTTA-FEHLBERG ORDER 4 ROUTINE TO SOLVE    *
*     A SYSTEM OF PARTIAL DIFFERENTIAL EQUATION OF THE FORM:  F(T,X)= X'     *
*******************************************************************************


      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION X(3),TOL(3),X_CS(4),TOL_CS(4)

      LOGICAL GUESS,GRAF,CPARAM,GUESS2,FOUND,SU

      PARAMETER(NUMA=25000,NUMB=2000)

      COMMON/FIRST/WTIME(NUMB),OTEMP(NUMB),OEVAP(NUMB),
     + OHUM(500),OPRECIP(NUMB),ODELP(NUMB),ODELA(NUMB),ODELI(NUMB),
     + GUESS,TEMPC,EVAPC,PRECIPC,DELAC,DELIC,CPARAM,DELPC,
     + EVAPC_P,EVAPC_D

      COMMON/SECOND/CTEMP(NUMB),CEVAP(NUMB),CHUM(500),CPRECIP(NUMB),
     + CDELP(NUMB),CSDELA(NUMB),GUESS2,TEMPC_C,EVAPC_C,
     + PRECIPC_C,DELAC_C,DELPC_C,STEMP(NUMB),SEVAP(NUMB),SHUM(500),
     + SPRECIP(NUMB),SDELP(NUMB),TEMPC_S,EVAPC_S,
     + PRECIPC_S,DELAC_S,DELPC_S

      COMMON/BOTH/CL(NUMA),TSOD(10),TCL(10),OTIME(NUMA),TCO3(10),
     + OQO(NUMA),QO(10),CONC_CL(NUMA),DOLTEMP,DELDOL,TODELI,TOTEMP,
     + ODEL_OUT(NUMA),OCL_OUT(NUMA),PEVAP(NUMB),DEVAP(NUMB),
     + SUM_PCL_DEP,AREA_P,AREA_D,
     + AREA(NUMA),SOAREA,ALLAREA,CONC_CL_P,
     + CL_P

      COMMON/BOTH2/CL_C,TSOD_C(10),TCL_C(10),TIME,TCO3_C(10),
     + CQO(10),GRAF,CONC_CL_C,DOLTEMP_C,DELDOL_C,
     + TCDELI,TCTEMP,CDEL_OUT,CCL_OUT,CL_S,NOPTS,
     + TSOD_S(10),TCL_S(10),TCO3_S(10),SQO(10),CONC_CL_S,
     + DOLTEMP_S,DELDOL_S,TSDELI,TSTEMP,SCL_DEP,TTLCL_IN,NPTS,
     + SUM_SCL_DEP,QI_C,QI_S,PQI

      COMMON/PREV/PCL_P,PCONC_CL_P,PSUM_PCL_DEP,PCL_C,PCONC_CL_C,
     + PCDEL_OUT,PCL_S,PCONC_CL_S,PSUM_SCL_DEP,PALLAREA,DATSAV,
     + SDC,SDP1,SDP2

      COMMON/HIST/QIHIST(1000),HUMTIME(500),NHPTS,SU,SUT(15),
     + SAVETIME,NUMST,QITIME(1000),NQPTS

      COMMON/SEARCH/IFIRST,ILAST,HIFIRST,HILAST

      COMMON/INFLOW/INCHOICE,A,B,C

      REAL START,FINISH,HALF

      INCHOICE=0
      A=0.D0
      B=0.D0
      C=0.D0
```

```
      OPEN(UNIT=98,FILE='OWENS.INP',STATUS='OLD')
      WRITE(6,*)
      WRITE(6,*)'READING OWENS LAKE STARTING PARAMETERS'
      READ(98,*)ITMAX,N,(X(I),I=1,N),DTMAX,DTMIN,(TOL(I),I=1,N),
     +     CONC_CL(1)

      OPEN(UNIT=89,FILE='C_S.INP',STATUS='OLD')
      WRITE(6,*)
      WRITE(6,*)'READING CHINA & SEARLES LAKE STARTING PARAMETERS'
      READ(89,*)ITMAX_CS,NCS,(X_CS(I),I=1,NCS),DTMAX_CS,DTMIN_CS,
     +     (TOL_CS(I),I=1,NCS),PCONC_CL_C,PCONC_CL_S

      OPEN(UNIT=69,FILE='P_D.INP',STATUS='OLD')
      WRITE(6,*)
      WRITE(6,*)'READING PANAMINT AND DEATH VALLEY STUFF'
      READ(69,*)AREA_P,PCONC_CL_P,PSUM_PCL_DEP,AREA_D

******************************************
** A CHANCE TO ADJUST INPUT PARAMETERS **
******************************************

      WRITE(6,*)
      WRITE(6,*)'THE CURRENT OWENS LAKE PARAMETERS ARE:'
      WRITE(6,*)'   1 : MAX ITERATIONS =',ITMAX
      WRITE(6,*)'   2 : LAKE VOL =',X(1)
      WRITE(6,*)'   3 : DEL O-18 DOLOMITE =',X(2)
      WRITE(6,*)'   4 : MAX TIME STEP =',DTMAX
      WRITE(6,*)'   5 : MIN TIME STEP =',DTMIN
      WRITE(6,*)'   6 : LAKE VOL TOLERANCE=',TOL(1)
      WRITE(6,*)'   7 : O-18 TOLERANCE=',TOL(2)
      WRITE(6,*)'   8 : CHLORIDE CONC=',CONC_CL(1)

      WRITE(6,*)
      WRITE(6,*)'CHANGE ANY OF THE STARTING PARAMETERS ?'
      WRITE(6,*)'1=YES    0=NO'
      READ(5,*)ISEE
      WRITE(6,*)

      IF(ISEE .EQ. 1)THEN

         WRITE(6,*)
         WRITE(6,*)'HOW MANY PARAMETERS WOULD YOU LIKE TO CHANGE ?'
         READ(5,*)K
         WRITE(6,*)

         DO 250 I = 1,K
            WRITE(6,*)
300         WRITE(6,*)'ENTER THE PARAMETER NUMBER'
            WRITE(6,*)
            WRITE(6,*)'   1 : MAX ITERATIONS ='
            WRITE(6,*)'   2 : LAKE VOL ='
            WRITE(6,*)'   3 : DEL O-18 DOLOMITE ='
            WRITE(6,*)'   4 : MAX TIME STEP ='
            WRITE(6,*)'   5 : MIN TIME STEP ='
            WRITE(6,*)'   6 : LAKE VOL TOLERANCE='
            WRITE(6,*)'   7 : O-18 TOLERANCE='
            WRITE(6,*)'   8 : CHLORIDE CONC='
            READ(5,*)NUMP
            IF(NUMP .EQ. 1)THEN
               WRITE(6,*)
               WRITE(6,*)'MAX ITERATIONS :',ITMAX
               WRITE(6,*)'ENTER NEW VALUE'
               READ(5,*)ITMAX
            ELSE IF(NUMP .EQ. 2)THEN
               WRITE(6,*)
               WRITE(6,'(A,D15.7)')' LAKE VOLUME :',X(1)
```

```
            WRITE(6,*)'NOTE: MAX LAKE VOL IS 30.02D9 (M^3)'
            WRITE(6,*)'ENTER NEW VALUE'
            READ(5,*)X(1)
         ELSE IF(NUMP .EQ. 3)THEN
            WRITE(6,*)
            WRITE(6,*)'DEL O-18 DOLOMITE :',X(2)
            WRITE(6,*)'ENTER NEW VALUE'
            READ(5,*)X(2)
         ELSE IF(NUMP .EQ. 4)THEN
            WRITE(6,*)
            WRITE(6,*)'MAX TIME STEP :',DTMAX
            WRITE(6,*)'ENTER NEW VALUE'
            READ(5,*)DTMAX
         ELSE IF(NUMP .EQ. 5)THEN
            WRITE(6,*)
            WRITE(6,*)'MIN TIME STEP :',DTMIN
            WRITE(6,*)'ENTER NEW VALUE'
            READ(5,*)DTMIN
         ELSE IF(NUMP .EQ. 6)THEN
            WRITE(6,*)
            WRITE(6,*)'LAKE VOL TOLERANCE :',TOL(1)
            WRITE(6,*)'ENTER NEW VALUE'
            READ(5,*)TOL(1)
         ELSE IF(NUMP .EQ. 7)THEN
            WRITE(6,*)
            WRITE(6,*)'DEL O-18 TOLERANCE :',TOL(2)
            WRITE(6,*)'ENTER NEW VALUE'
            READ(5,*)TOL(2)
         ELSE IF(NUMP .EQ. 8)THEN
            WRITE(6,*)
            WRITE(6,*)'CHLORIDE CONC :',CONC_CL(1)
            WRITE(6,*)'NOTE: 6.1 IS SATURATION'
            WRITE(6,*)'ENTER NEW VALUE'
            READ(5,*)CONC_CL(1)
         ELSE
            WRITE(6,*)
            WRITE(6,*)'YOU SUFFER FROM CALCULATOR DEPENDENCY'
            WRITE(6,*)'PICK A NUMBER BETWEEN 1 AND 8'
            WRITE(6,*)
            GO TO 300
         ENDIF
250     CONTINUE

*************************************
** WRITE NEW VALUES TO INPUT FILE **
*************************************

        REWIND 98
        WRITE(98,*)ITMAX,N,(X(I),I=1,N),DTMAX,DTMIN,(TOL(I),
     +       I=1,N),CONC_CL(1)
        CLOSE(UNIT=98)
      ENDIF

******************************************************************
** A        VARIABLE USED TO WRITE STARTING VALUES TO FILES **
******************************************************************

      GUESS=.TRUE.

*****************************************************
** CALCULATE MOLES OF CHLORIDE FROM CONC AND VOL **
*****************************************************

      IF(X(1) .LE. 0.D0)THEN
         CL(1)=0.D0
         CONC_CL(1)=0.D0
```

```
      ELSE
         CL(1)=CONC_CL(1)*(X(1)*1.D3)
      ENDIF


*********************************************************
** A CHANCE TO ADJUST INPUT PARAMETERS FOR CHINA LAKE **
*********************************************************

      WRITE(6,*)
      WRITE(6,*)'THE CURRENT CHINA LAKE PARAMETERS ARE:'
      WRITE(6,*)'   1 : MAX ITERATIONS =',ITMAX_CS
      WRITE(6,*)'   2 : LAKE VOL =',X_CS(1)
      WRITE(6,*)'   3 : DEL O-18 DOLOMITE =',X_CS(2)
      WRITE(6,*)'   4 : MAX TIME STEP =',DTMAX_CS
      WRITE(6,*)'   5 : MIN TIME STEP =',DTMIN_CS
      WRITE(6,*)'   6 : LAKE VOL TOLERANCE=',TOL_CS(1)
      WRITE(6,*)'   7 : O-18 TOLERANCE=',TOL_CS(2)
      WRITE(6,*)'   8 : CHLORIDE CONC=',PCONC_CL_C

      WRITE(6,*)
      WRITE(6,*)'CHANGE ANY OF THE STARTING PARAMETERS ?'
      WRITE(6,*)'1=YES    0=NO'
      READ(5,*)ISEE
      WRITE(6,*)

      IF(ISEE .EQ. 1)THEN

         WRITE(6,*)
         WRITE(6,*)'HOW MANY PARAMETERS WOULD YOU LIKE TO CHANGE ?'
         READ(5,*)K
         WRITE(6,*)

         DO 350 I = 1,K
            WRITE(6,*)
360         WRITE(6,*)'ENTER THE PARAMETER NUMBER'
            WRITE(6,*)
            WRITE(6,*)'   1 : MAX ITERATIONS ='
            WRITE(6,*)'   2 : LAKE VOL ='
            WRITE(6,*)'   3 : DEL O-18 DOLOMITE ='
            WRITE(6,*)'   4 : MAX TIME STEP ='
            WRITE(6,*)'   5 : MIN TIME STEP ='
            WRITE(6,*)'   6 : LAKE VOL TOLERANCE='
            WRITE(6,*)'   7 : O-18 TOLERANCE='
            WRITE(6,*)'   8 : CHLORIDE CONC='
            READ(5,*)NUMP
            IF(NUMP .EQ. 1)THEN
               WRITE(6,*)
               WRITE(6,*)'MAX ITERATIONS :',ITMAX_CS
               WRITE(6,*)'ENTER NEW VALUE'
               READ(5,*)ITMAX_CS
            ELSE IF(NUMP .EQ. 2)THEN
               WRITE(6,*)
               WRITE(6,'(A,D15.7)')' LAKE VOLUME :',X_CS(1)
               WRITE(6,*)'NOTE: MAX LAKE VOL IS 0.696D9 (M^3)'
               WRITE(6,*)'ENTER NEW VALUE'
               READ(5,*)X_CS(1)
            ELSE IF(NUMP .EQ. 3)THEN
               WRITE(6,*)
               WRITE(6,*)'DEL O-18 DOLOMITE :',X_CS(2)
               WRITE(6,*)'ENTER NEW VALUE'
               READ(5,*)X_CS(2)
            ELSE IF(NUMP .EQ. 4)THEN
               WRITE(6,*)
               WRITE(6,*)'MAX TIME STEP :',DTMAX_CS
               WRITE(6,*)'ENTER NEW VALUE'
```

```
            READ(5,*)DTMAX_CS
         ELSE IF(NUMP .EQ. 5)THEN
            WRITE(6,*)
            WRITE(6,*)'MIN TIME STEP :',DTMIN_CS
            WRITE(6,*)'ENTER NEW VALUE'
            READ(5,*)DTMIN_CS
         ELSE IF(NUMP .EQ. 6)THEN
            WRITE(6,*)
            WRITE(6,*)'LAKE VOL TOLERANCE :',TOL_CS(1)
            WRITE(6,*)'ENTER NEW VALUE'
            READ(5,*)TOL_CS(1)
         ELSE IF(NUMP .EQ. 7)THEN
            WRITE(6,*)
            WRITE(6,*)'DEL O-18 TOLERANCE :',TOL_CS(2)
            WRITE(6,*)'ENTER NEW VALUE'
            READ(5,*)TOL_CS(2)
         ELSE IF(NUMP .EQ. 8)THEN
            WRITE(6,*)
            WRITE(6,*)'CHLORIDE CONC :',PCONC_CL_C
            WRITE(6,*)'NOTE: 6.1 IS SATURATION'
            WRITE(6,*)'ENTER NEW VALUE'
            READ(5,*)PCONC_CL_C
         ELSE
            WRITE(6,*)
            WRITE(6,*)'OBVIOUSLY MATH IS NOT YOUR FORTE'
            WRITE(6,*)'PICK A NUMBER BETWEEN 1 AND 8'
            WRITE(6,*)
            GO TO 360
         ENDIF
350     CONTINUE
      ENDIF

***************************************************
** CALCULATE MOLES OF CHLORIDE FROM CONC AND VOL **
***************************************************

      IF(X_CS(1) .LE. 0.D0)THEN
        PCL_C=0.D0
        PCONC_CL_C=0.D0
      ELSE
        PCL_C=PCONC_CL_C*(X_CS(1)*1.D3)
      ENDIF


**********************************************************
** A CHANCE TO ADJUST INPUT PARAMETERS FOR SEARLES LAKE **
**********************************************************

      WRITE(6,*)
      WRITE(6,*)'THE CURRENT SEARLES LAKE PARAMETERS ARE:'
      WRITE(6,*)'   1 : MAX ITERATIONS =',ITMAX_CS
      WRITE(6,*)'   2 : LAKE VOL =',X_CS(3)
      WRITE(6,*)'   3 : DEL O-18 DOLOMITE =',X_CS(4)
      WRITE(6,*)'   4 : MAX TIME STEP =',DTMAX_CS
      WRITE(6,*)'   5 : MIN TIME STEP =',DTMIN_CS
      WRITE(6,*)'   6 : LAKE VOL TOLERANCE=',TOL_CS(3)
      WRITE(6,*)'   7 : O-18 TOLERANCE=',TOL_CS(4)
      WRITE(6,*)'   8 : CHLORIDE CONC=',PCONC_CL_S

      WRITE(6,*)
      WRITE(6,*)'CHANGE ANY OF THE STARTING PARAMETERS ?'
      WRITE(6,*)'1=YES    0=NO'
      READ(5,*)ISEE
      WRITE(6,*)

      IF(ISEE .EQ. 1)THEN
```

```
      WRITE(6,*)
      WRITE(6,*)'HOW MANY PARAMETERS WOULD YOU LIKE TO CHANGE ?'
      READ(5,*)K
      WRITE(6,*)

      DO 370 I = 1,K
         WRITE(6,*)
380      WRITE(6,*)'ENTER THE PARAMETER NUMBER'
         WRITE(6,*)
         WRITE(6,*)'   1 : MAX ITERATIONS ='
         WRITE(6,*)'   2 : LAKE VOL ='
         WRITE(6,*)'   3 : DEL O-18 DOLOMITE='
         WRITE(6,*)'   4 : MAX TIME STEP ='
         WRITE(6,*)'   5 : MIN TIME STEP ='
         WRITE(6,*)'   6 : LAKE VOL TOLERANCE='
         WRITE(6,*)'   7 : O-18 TOLERANCE='
         WRITE(6,*)'   8 : CHLORIDE CONC='
         READ(5,*)NUMP
         IF(NUMP .EQ. 1)THEN
            WRITE(6,*)
            WRITE(6,*)'MAX ITERATIONS :',ITMAX_CS
            WRITE(6,*)'ENTER NEW VALUE'
            READ(5,*)ITMAX_CS
         ELSE IF(NUMP .EQ. 2)THEN
            WRITE(6,*)
            WRITE(6,'(A,D15.7)')' LAKE VOLUME :',X_CS(3)
            WRITE(6,*)'NOTE: MAX LAKE VOL IS 85.28D9 (M^3)'
            WRITE(6,*)'ENTER NEW VALUE'
            READ(5,*)X_CS(3)
         ELSE IF(NUMP .EQ. 3)THEN
            WRITE(6,*)
            WRITE(6,*)'DEL O-18 DOLOMITE :',X_CS(4)
            WRITE(6,*)'ENTER NEW VALUE'
            READ(5,*)X_CS(4)
         ELSE IF(NUMP .EQ. 4)THEN
            WRITE(6,*)
            WRITE(6,*)'MAX TIME STEP :',DTMAX_CS
            WRITE(6,*)'ENTER NEW VALUE'
            READ(5,*)DTMAX_CS
         ELSE IF(NUMP .EQ. 5)THEN
            WRITE(6,*)
            WRITE(6,*)'MIN TIME STEP :',DTMIN_CS
            WRITE(6,*)'ENTER NEW VALUE'
            READ(5,*)DTMIN_CS
         ELSE IF(NUMP .EQ. 6)THEN
            WRITE(6,*)
            WRITE(6,*)'LAKE VOL TOLERANCE :',TOL_CS(3)
            WRITE(6,*)'ENTER NEW VALUE'
            READ(5,*)TOL_CS(3)
         ELSE IF(NUMP .EQ. 7)THEN
            WRITE(6,*)
            WRITE(6,*)'DEL O-18 TOLERANCE :',TOL_CS(4)
            WRITE(6,*)'ENTER NEW VALUE'
            READ(5,*)TOL_CS(4)
         ELSE IF(NUMP .EQ. 8)THEN
            WRITE(6,*)
            WRITE(6,*)'CHLORIDE CONC :',PCONC_CL_S
            WRITE(6,*)'NOTE: 6.1 IS SATURATION'
            WRITE(6,*)'ENTER NEW VALUE'
            READ(5,*)PCONC_CL_S
         ELSE
            WRITE(6,*)
            WRITE(6,*)'GET A CLUE KELP BREATH'
            WRITE(6,*)'PICK A NUMBER BETWEEN 1 AND 8'
            WRITE(6,*)
```

```
              GO TO 380
            ENDIF
370       CONTINUE
        ENDIF

************************************
** WRITE NEW VALUES TO INPUT FILE **
************************************

        REWIND 89
        WRITE(89,*)ITMAX_CS,NCS,(X_CS(I),I=1,NCS),DTMAX_CS,
     +       DTMIN_CS,(TOL_CS(I),I=1,NCS),PCONC_CL_C,
     +      PCONC_CL_S
        CLOSE(UNIT=89)

****************************************************
** CALCULATE MOLES OF CHLORIDE FROM CONC AND VOL **
****************************************************

        IF(X_CS(3) .LE. 0.D0)THEN
           PCL_S=0.D0
           PCONC_CL_S=0.D0
        ELSE
           PCL_S=PCONC_CL_S*(X_CS(3)*1.D3)
        ENDIF

*****************************************************************
** TOTAL MASS OF CHLORIDE DEPOSITED IN SEARLES AT START OF RUN **
*****************************************************************

        WRITE(6,*)
        WRITE(6,*)'ENTER THE MASS OF CL DEPOSITED IN SEARLES LAKE'
        WRITE(6,*)'AT THE BEGINNING OF THIS RUN (KG/M^2)'
        WRITE(6,*)
        READ(5,*)PSUM_SCL_DEP

***********************************************************************
** A CHANCE TO ADJUST INPUT PARAMETERS FOR PANAMINT AND DEATH VALLEY **
***********************************************************************

        WRITE(6,*)
        WRITE(6,*)'THE CURRENT PANAMINT LAKE PARAMETERS ARE:'
        WRITE(6,*)'   1 : SURFACE AREA =',AREA_P
        WRITE(6,*)'   2 : CHLORIDE CONC=',PCONC_CL_P
        WRITE(6,*)'   3 : CHLORIDE DEPOSITED (KG/M^2)',PSUM_PCL_DEP

        WRITE(6,*)
        WRITE(6,*)'CHANGE ANY OF THE STARTING PARAMETERS ?'
        WRITE(6,*)'1=YES    0=NO'
        READ(5,*)ISEE
        WRITE(6,*)

        IF(ISEE .EQ. 1)THEN

           WRITE(6,*)
           WRITE(6,*)'HOW MANY PARAMETERS WOULD YOU LIKE TO CHANGE ?'
           READ(5,*)K
           WRITE(6,*)

           DO 400 I = 1,K
              WRITE(6,*)
 410          WRITE(6,*)'ENTER THE PARAMETER NUMBER'
              WRITE(6,*)
              WRITE(6,*)'   1 : SURFACE AREA'
              WRITE(6,*)'   2 : CHLORIDE CONC'
              WRITE(6,*)'   3 : CHLORIDE DEPOSITED (KG/M^2)'
```

```
            READ(5,*)NUMP
            IF(NUMP .EQ. 1)THEN
               WRITE(6,*)
               WRITE(6,'(A,D15.7)')' SURFACE AREA =',AREA_P
               WRITE(6,*)'NOTE: MAX SURFACE AREA IS 0.727D9'
               WRITE(6,*)'ENTER NEW VALUE'
               READ(5,*)AREA_P
            ELSE IF(NUMP .EQ. 2)THEN
               WRITE(6,*)
               WRITE(6,*)'CHLORIDE CONC :',PCONC_CL_P
               WRITE(6,*)'NOTE: 6.1 IS SATURATION'
               WRITE(6,*)'ENTER NEW VALUE'
               READ(5,*)PCONC_CL_P
            ELSE IF(NUMP .EQ. 3)THEN
               WRITE(6,*)
               WRITE(6,*)'SUM OF CHLORIDE DEPOSITED',PSUM_PCL_DEP
               WRITE(6,*)'ENTER NEW VALUE'
               READ(5,*)PSUM_PCL_DEP
            ELSE
               WRITE(6,*)
               WRITE(6,*)'EVERY DAY MUST BE MONDAY FOR SOMEONE LIKE
     +YOU'
               WRITE(6,*)'PICK A NUMBER BETWEEN 1 AND 8'
               WRITE(6,*)
               GO TO 410
            ENDIF
400      CONTINUE
        ENDIF

*****************************************************
** CALCULATE MOLES OF CHLORIDE FROM CONC AND VOL **
*****************************************************

        IF(PCONC_CL_P .LE. 0.D0)THEN
           PCL_P=0.D0
        ELSE
           VOL_P=PVOL(AREA_P)
           PCL_P=PCONC_CL_P/(VOL_P*1.D3)
        ENDIF

************************
** DEATH VALLEY STUFF **
************************

        WRITE(6,*)
        WRITE(6,*)'THE CURRENT DEATH VALLEY SURFACE AREA IS:'
        WRITE(6,*)'    1 : SURFACE AREA =',AREA_D

        WRITE(6,*)
        WRITE(6,*)'WOULD YOU LIKE TO CHANGE THIS ?'
        WRITE(6,*)'1=YES    0=NO'
        READ(5,*)ISEE
        WRITE(6,*)

        IF(ISEE .EQ. 1)THEN

           WRITE(6,*)
           WRITE(6,'(A,D15.7)')' SURFACE AREA =',AREA_D
           WRITE(6,*)'NOTE: MAX SURFACE AREA IS 0.583D9'
           WRITE(6,*)'ENTER NEW VALUE'
           READ(5,*)AREA_D

        ENDIF

************************************
** WRITE NEW VALUES TO INPUT FILE **
```

```
************************************

      REWIND 69
      WRITE(69,*)AREA_P,PCONC_CL_P,PSUM_PCL_DEP,AREA_D
      CLOSE(UNIT=69)

******************************************************************************
*            LET'S CHOOSE AN INFLOW FUNCTION AND TIME INTERVAL              *
******************************************************************************


      WRITE(6,*)
      WRITE(6,*)
115   WRITE(6,*)'WHICH INFLOW FUNCTION WOULD YOU LIKE FOR OWENS LAKE?'
      WRITE(6,*)'1=LINEAR'
      WRITE(6,*)'2=EXPONENTIAL'
      WRITE(6,*)'3=LOGARITHMIC'
      WRITE(6,*)'4=POWER'
      WRITE(6,*)'5=SINUSOIDAL'
      WRITE(6,*)'6=STEP'
      WRITE(6,*)'7=ZERO INFLOW'
      WRITE(6,*)'8=STEADY-STATE HISTORY, VARIABLE TEMP'
      WRITE(6,*)'9=STEADY-STATE HISTORY, CONSTANT HIGH TEMP'
      WRITE(6,*)'10=STEADY-STATE HISTORY, CONSTANT LOW TEMP'
      READ(5,*)INCHOICE

      WRITE(6,*)
      WRITE(6,*)
      IF(INCHOICE .EQ. 1)THEN
         WRITE(6,*)'YOU HAVE CHOSEN F(QI)= A*X+B'
         WRITE(6,*)'ENTER A VALUE FOR ''A'', AND ''B'''
      ELSE IF(INCHOICE .EQ. 2)THEN
         WRITE(6,*)'YOU HAVE CHOSEN F(QI)= B*EXP(A*X)'
         WRITE(6,*)'ENTER A VALUE FOR ''A'', AND ''B'''
      ELSE IF(INCHOICE .EQ. 3)THEN
         WRITE(6,*)'YOU HAVE CHOSEN F(QI)= B+A*LOG(X)'
         WRITE(6,*)'ENTER A VALUE FOR ''A'', AND ''B'''
      ELSE IF(INCHOICE .EQ. 4)THEN
         WRITE(6,*)'YOU HAVE CHOSEN F(QI)= B+A*(X**C)'
         WRITE(6,*)'ENTER VALUES FOR ''A'',''B'', AND ''C'''
      ELSE IF(INCHOICE .EQ. 5)THEN
         WRITE(6,*)'YOU HAVE CHOSEN F(QI)= B+A*SIN(C*X)'
         WRITE(6,*)'ENTER VALUES FOR ''A'',''B'', AND ''C'''
      ELSE IF(INCHOICE .EQ. 6)THEN
         WRITE(6,*)'YOU HAVE CHOSEN F(QI)= B+(A*B)'
         WRITE(6,*)'ENTER A VALUE FOR ''A'', AND ''B'''
      ELSE IF(INCHOICE .EQ. 7)THEN
         WRITE(6,*)'YOU HAVE CHOSEN ZERO INFLOW'
         WRITE(6,*)'GRAB YOUR CANTEEN AND HEAD FOR THE SHADE'
      ELSE IF(INCHOICE .EQ. 8)THEN
         WRITE(6,*)'YOU HAVE CHOSEN STEADY STATE "GUESS"'
         WRITE(6,*)'VARIABLE TEMPERATURE HISTORY'
      ELSE IF(INCHOICE .EQ. 9)THEN
         WRITE(6,*)'YOU HAVE CHOSEN STEADY STATE "GUESS"'
         WRITE(6,*)'CONSTANT HIGH TEMPERATURE HISTORY'
      ELSE IF(INCHOICE .EQ. 10)THEN
         WRITE(6,*)'YOU HAVE CHOSEN STEADY STATE "GUESS"'
         WRITE(6,*)'CONSTANT LOW TEMPERATURE HISTORY'
      ELSE
         WRITE(6,*)
         WRITE(6,*)
         WRITE(6,*)'NOT A VALID CHOICE MULLET-HEAD'
         WRITE(6,*)
         WRITE(6,*)
         GOTO 115
      ENDIF
```

```
      IF(INCHOICE .EQ. 8)THEN

         WRITE(6,*)'READING INFLOW HISTORY'
         OPEN(UNIT=29,FILE='CASEA1.DAT',STATUS='OLD')
         DO 450 I=1,1000
            READ(29,*,END=451)QITIME(I),F2,QIHIST(I)
            QITIME(I)=QITIME(I)*1.D6
450      CONTINUE
451      WRITE(6,*)'READ',I-1,' POINTS FROM INFLOW HISTORY'
         NQPTS=I-1

      ELSEIF(INCHOICE .EQ. 9)THEN

         WRITE(6,*)'READING INFLOW HISTORY'
         OPEN(UNIT=28,FILE='CASEB1.DAT',STATUS='OLD')
         DO 460 I=1,1000
            READ(28,*,END=461)QITIME(I),F2,QIHIST(I)
            QITIME(I)=QITIME(I)*1.D6
460      CONTINUE
461      WRITE(6,*)'READ',I-1,' POINTS FROM INFLOW HISTORY'
         NQPTS=I-1

      ELSEIF(INCHOICE .EQ. 10)THEN

         WRITE(6,*)'READING INFLOW HISTORY'
         OPEN(UNIT=27,FILE='CASEC1.DAT',STATUS='OLD')
         DO 470 I=1,1000
            READ(27,*,END=471)QITIME(I),F2,QIHIST(I)
            QITIME(I)=QITIME(I)*1.D6
470      CONTINUE
471      WRITE(6,*)'READ',I-1,' POINTS FROM INFLOW HISTORY'
         NQPTS=I-1

      ELSEIF(INCHOICE .EQ. 4 .OR. INCHOICE .EQ. 5)THEN
         READ(5,*)A,B,C
      ELSE IF(INCHOICE .NE. 7)THEN
         READ(5,*)A,B
      ENDIF

      WRITE(6,*)
      WRITE(6,*)'ENTER STARTING TIME AND ENDING TIME'
      WRITE(6,*)'0 CORRESPONDS TO PRESENT, 2.0 IS 2 MILLION YRS AGO'
      WRITE(6,*)
      WRITE(6,*)
      WRITE(6,*)'MANAGEMENT ACCEPTS NO RESPONSIBILITY FOR PEOPLE WHO'
      WRITE(6,*)'RUN THE MODEL BACKWARD IN TIME'
      WRITE(6,*)
      WRITE(6,*)
      READ(5,*)TBEG, TEND
      TBEG=TBEG*1.D6
      TEND=TEND*1.D6
      WRITE(6,*)

***********************************************************************
** TELL THE MODEL HOW MUCH TIME TO PUT BETWEEN SAVED DATA POINTS.    **
** THIS MAKES THE MODEL RUN MUCH FASTER AND REDUCES THE SIZE OF THE **
** DATA FILES.                                                       **
***********************************************************************

      WRITE(6,*)
      WRITE(6,*)'ENTER THE MINIMUM TIME BETWEEN SAVED DATA POINTS'
      WRITE(6,*)
      READ(5,*)DATSAV

************************************************************************
```

```
** SET "PEAK AND VALLEY" DETECTORS SO DATSAV SPACING DOESN'T MISS ANY **
** MAXIMA OR MINIMA                                                   **
************************************************************************

      SDC=X_CS(4)
      SDP1=X_CS(4)
      SDP2=X_CS(4)

********************************
** CONSTANT PARAMETER OPTION **
********************************

      CPARAM=.FALSE.
      WRITE(6,*)
      WRITE(6,*)'DO YOU WANT TO RUN THE PROGRAM WITH CONSTANT PARAMETE
     +RS ?'
      WRITE(6,*)'1=YES   0=NO'
      WRITE(6,*)
      READ(5,*)INPARAM

      IF(INPARAM .EQ. 1)THEN
         CPARAM=.TRUE.

 490     WRITE(6,*)
         WRITE(6,*)'WOULD YOU LIKE UPPER LIMIT, LOWER LIMIT, OR CUSTOM
     + '
         WRITE(6,*)'1 = UPPER LIMIT'
         WRITE(6,*)'2 = LOWER LIMIT'
         WRITE(6,*)'3 = CUSTOM'
         WRITE(6,*)
         READ(5,*)LIMCHOICE

         IF(LIMCHOICE .EQ. 1)THEN
            OPEN(UNIT=68,FILE='UPPER.INP',STATUS='OLD')
            WRITE(6,*)'READING CONSTANT PARAMETERS'
            WRITE(6,*)
            READ(68,*)TEMPC,TEMPC_C,TEMPC_S,
     +           PRECIPC,PRECIPC_C,PRECIPC_S,EVAPC,EVAPC_C,EVAPC_S,
     +           EVAPC_P,EVAPC_D,DELAC,DELAC_C,DELAC_S,DELIC,DELPC,
     +           DELPC_C,DELPC_S
            CLOSE(UNIT=68)
         ELSE IF(LIMCHOICE .EQ. 2)THEN
            OPEN(UNIT=67,FILE='LOWER.INP',STATUS='OLD')
            WRITE(6,*)'READING CONSTANT PARAMETERS'
            WRITE(6,*)
            READ(67,*)TEMPC,TEMPC_C,TEMPC_S,
     +           PRECIPC,PRECIPC_C,PRECIPC_S,EVAPC,EVAPC_C,EVAPC_S,
     +           EVAPC_P,EVAPC_D,DELAC,DELAC_C,DELAC_S,DELIC,DELPC,
     +           DELPC_C,DELPC_S
            CLOSE(UNIT=67)
         ELSE IF(LIMCHOICE .EQ. 3)THEN

****************************
** PICK A TEMP, ANY TEMP **
****************************

            WRITE(6,*)
            WRITE(6,*)'ENTER THE TEMP FOR OWENS (DEGREES C)'
            WRITE(6,*)
            READ(5,*)TEMPC
            WRITE(6,*)

            WRITE(6,*)
            WRITE(6,*)'ENTER THE TEMP FOR CHINA (DEGREES C)'
            WRITE(6,*)
            READ(5,*)TEMPC_C
```

```
      WRITE(6,*)

      WRITE(6,*)
      WRITE(6,*)'ENTER THE TEMP FOR SEARLES (DEGREES C)'
      WRITE(6,*)
      READ(5,*)TEMPC_S
      WRITE(6,*)

*****************************
** CONSTANT PRECIPITATION **
*****************************

      WRITE(6,*)
      WRITE(6,*)'ENTER THE PRECIPITATION FOR OWENS(METERS/YR)'
      WRITE(6,*)
      READ(5,*)PRECIPC
      WRITE(6,*)

      WRITE(6,*)
      WRITE(6,*)'ENTER THE PRECIPITATION FOR CHINA(METERS/YR)'
      WRITE(6,*)
      READ(5,*)PRECIPC_C
      WRITE(6,*)

      WRITE(6,*)
      WRITE(6,*)'ENTER THE PRECIPITATION FOR SEARLES(METERS/YR)'
      WRITE(6,*)
      READ(5,*)PRECIPC_S
      WRITE(6,*)

***************************
** CONSTANT EVAPORATION **
***************************

      WRITE(6,*)
      WRITE(6,*)'ENTER THE EVAPORATION FOR OWENS(METERS/YR)'
      WRITE(6,*)
      READ(5,*)EVAPC
      WRITE(6,*)

      WRITE(6,*)
      WRITE(6,*)'ENTER THE EVAPORATION FOR CHINA(METERS/YR)'
      WRITE(6,*)
      READ(5,*)EVAPC_C
      WRITE(6,*)

      WRITE(6,*)
      WRITE(6,*)'ENTER THE EVAPORATION FOR SEARLES(METERS/YR)'
      WRITE(6,*)
      READ(5,*)EVAPC_S
      WRITE(6,*)

      WRITE(6,*)
      WRITE(6,*)'ENTER THE EVAPORATION FOR PANAMINT(METERS/YR)'
      WRITE(6,*)
      READ(5,*)EVAPC_P
      WRITE(6,*)

      WRITE(6,*)
      WRITE(6,*)'ENTER THE EVAPORATION FOR DEATH VALLEY(METERS/Y
     +R)'
      WRITE(6,*)
      READ(5,*)EVAPC_D
      WRITE(6,*)

******************************
```

```
** CONSTANT DEL ATMOSPHERE **
******************************

            WRITE(6,*)
            WRITE(6,*)'ENTER DEL ATMOSPHERE FOR OWENS (PER MIL)'
            WRITE(6,*)
            READ(5,*)DELAC
            WRITE(6,*)

            WRITE(6,*)
            WRITE(6,*)'ENTER DEL ATMOSPHERE FOR CHINA (PER MIL)'
            WRITE(6,*)
            READ(5,*)DELAC_C
            WRITE(6,*)

            WRITE(6,*)
            WRITE(6,*)'ENTER DEL ATMOSPHERE FOR SEARLES (PER MIL)'
            WRITE(6,*)
            READ(5,*)DELAC_S
            WRITE(6,*)

**************************
** CONSTANT DEL INFLOW  **
**************************

            WRITE(6,*)
            WRITE(6,*)'ENTER DEL INFLOW FOR OWENS (PER MIL)'
            WRITE(6,*)
            READ(5,*)DELIC
            WRITE(6,*)

**************************
** CONSTANT DEL PRECIP  **
**************************

            WRITE(6,*)
            WRITE(6,*)'ENTER DEL PRECIPITATION FOR OWENS (PER MIL)'
            WRITE(6,*)
            READ(5,*)DELPC
            WRITE(6,*)

            WRITE(6,*)
            WRITE(6,*)'ENTER DEL PRECIPITATION FOR CHINA (PER MIL)'
            WRITE(6,*)
            READ(5,*)DELPC_C
            WRITE(6,*)

            WRITE(6,*)
            WRITE(6,*)'ENTER DEL PRECIPITATION FOR SEARLES (PER MIL)'
            WRITE(6,*)
            READ(5,*)DELPC_S
            WRITE(6,*)

         ELSE
            WRITE(6,*)'NO,NO,NOOOOOOO...'
            WRITE(6,*)'PICK 1,2, OR 3 '
            WRITE(6,*)
            GO TO 490
         ENDIF


         OPEN(UNIT=80,FILE='HUMHIST.DAT',STATUS='OLD')
         DO 1750 I=1,400
            READ(80,*,END=1751)HUMTIME(I),SHUM(I)
            OHUM(I)=SHUM(I)
            CHUM(I)=SHUM(I)
```

```
1750      CONTINUE

1751      NHPTS=I-1
          WRITE(6,*)'READ',NHPTS,' FROM HUMHIST.DAT'
          CLOSE(80)

          DO 1760 I=1,NHPTS
             HUMTIME(I)=HUMTIME(I)*1.D6
1760      CONTINUE

       ENDIF

********************
** GRAPHICS STUFF **
********************

       GRAF=.FALSE.

***************************************************
** SAVE VALUES FOR MODEL STARTUP AT A GIVEN TIME **
***************************************************

       SU=.FALSE.

       WRITE(6,*)
       WRITE(6,*)'DO YOU WANT TO SAVE INFO TO RESTART THE MODEL'
       WRITE(6,*)'AT A SPECIFIC TIME'
       WRITE(6,*)' 1=YES    0=NO'
       WRITE(6,*)

       READ(5,*)ISU

       IF(ISU .EQ. 1)THEN
          SU=.TRUE.
          WRITE(6,*)
          WRITE(6,*)'HOW MANY STARTUP TIMES DO YOU WISH ?'
          WRITE(6,*)
          READ(5,*)NUMST

          DO 1900 I=1,NUMST
             WRITE(6,*)
             WRITE(6,*)'ENTER STARTUP TIME'
             READ(5,*)SUT(I)
             WRITE(6,*)
 1900     CONTINUE

          SAVETIME=SUT(1)
          ISTCNT=1

       ENDIF



*****************************************************************************
*     ALL WE'RE DOING HERE IS READING DATA FILES, THE GOOD STUFF IS LATER   *
*****************************************************************************

       IF(.NOT. CPARAM)THEN

          WRITE(6,*)
          WRITE(6,*)
          WRITE(6,*)'READING DATA FILES ...'
          WRITE(6,*)
          WRITE(6,*)
```

```
      OPEN(UNIT=20,FILE='OWENS.UF',STATUS='OLD',
     +     FORM='UNFORMATTED')

      DO 500 I = 1, 2000
         READ (20,END=501) WTIME(I),OTEMP(I),OEVAP(I),GBG1,
     +        OPRECIP(I)
500   CONTINUE

501   NPTS=I-1

      DO 600 I = 1,NPTS
         WTIME(I)=WTIME(I)*1.0D6
600   CONTINUE

      DO 650 I=1,NPTS
         IF(WTIME(I).GT.TBEG)THEN
            ILAST=I
            GOTO 651
         ENDIF
 650  CONTINUE

 651  DO 660 I=1,NPTS
         IF(WTIME(I).GT.TEND)THEN
            IFIRST=I-1
            GOTO 661
         ENDIF
 660  CONTINUE

 661  CLOSE (UNIT=20)

      OPEN(UNIT=40,FILE='SEARLES.UF',STATUS='OLD',
     +     FORM='UNFORMATTED')

      DO 800 I = 1, NPTS
         READ (40) STEMP(I),SEVAP(I),GBG2,SPRECIP(I)
800   CONTINUE

      CLOSE (UNIT=40)

      DO 850 I=1,NPTS
         CTEMP(I)=STEMP(I)
         CEVAP(I)=SEVAP(I)
         CPRECIP(I)=SPRECIP(I)
 850  CONTINUE

      OPEN(UNIT=21,FILE='ODELP.UF',STATUS='OLD',
     +     FORM='UNFORMATTED')
      DO 900 I = 1, NPTS
         READ(21)ODELP(I)
900   CONTINUE

      CLOSE(UNIT=21)

      OPEN(UNIT=41,FILE='SDELP.UF',STATUS='OLD',
     +     FORM='UNFORMATTED')
      DO 1100 I = 1, NPTS
         READ(41)SDELP(I)
1100  CONTINUE

      CLOSE(UNIT=41)

      DO 1150 I=1,NPTS
         CDELP(I)=SDELP(I)
 1150 CONTINUE

      OPEN(UNIT=22,FILE='ODELA.UF',STATUS='OLD',
```

```
     +          FORM='UNFORMATTED')
           DO 1200 I = 1, NPTS
              READ(22) ODELA(I)
1200       CONTINUE

           CLOSE(UNIT=22)

           OPEN(UNIT=32,FILE='CSDELA.UF',STATUS='OLD',
     +          FORM='UNFORMATTED')
           DO 1300 I = 1, NPTS
              READ(32) CSDELA(I)
1300       CONTINUE

           CLOSE(UNIT=32)

           OPEN(UNIT=23,FILE='ODELI.UF',STATUS='OLD',
     +          FORM='UNFORMATTED')
           DO 1400 I = 1,NPTS
              READ(23)ODELI(I)
1400       CONTINUE

           CLOSE(UNIT=23)

           OPEN(UNIT=85,FILE='PEVAP.UF',STATUS='OLD',
     +          FORM='UNFORMATTED')
           DO 1500 I = 1,NPTS
              READ(85)PEVAP(I)
1500       CONTINUE

           CLOSE(UNIT=85)

           OPEN(UNIT=84,FILE='DEVAP.UF',STATUS='OLD',
     +          FORM='UNFORMATTED')
           DO 1600 I = 1,NPTS
              READ(84)DEVAP(I)
1600       CONTINUE

           CLOSE(UNIT=84)

           OPEN(UNIT=80,FILE='HUMHIST.DAT',STATUS='OLD')
           DO 1700 I=1,400
              READ(80,*,END=1701)HUMTIME(I),SHUM(I)
              OHUM(I)=SHUM(I)
              CHUM(I)=SHUM(I)
 1700      CONTINUE

 1701      NHPTS=I-1

           CLOSE(UNIT=80)

           DO 1800 I=1,NHPTS
              HUMTIME(I)=HUMTIME(I)*1.D6
1800       CONTINUE

        ENDIF

***************************************
** CONVERT DEL DOLOMITE TO DEL WATER **
***************************************

           IF(CPARAM)THEN
              X(2)= W_DEL(X(2),TEMPC)
              X_CS(2)=W_DEL(X_CS(2),TEMPC_C)
              X_CS(4)=W_DEL(X_CS(4),TEMPC_S)
           ELSE
```

```
          CALL FINDT(NPTS,TBEG,NDX,FOUND,WTIME)
          TOTEMP=(OTEMP(NDX+1)-OTEMP(NDX))/(WTIME(NDX+1)-
     +         WTIME(NDX))*(TBEG-WTIME(NDX))+OTEMP(NDX)
          TSTEMP=(STEMP(NDX+1)-STEMP(NDX))/(WTIME(NDX+1)-
     +         WTIME(NDX))*(TBEG-WTIME(NDX))+STEMP(NDX)
          TCTEMP=(CTEMP(NDX+1)-CTEMP(NDX))/(WTIME(NDX+1)-
     +         WTIME(NDX))*(TBEG-WTIME(NDX))+CTEMP(NDX)

          X(2)=W_DEL(X(2),TOTEMP)
          X_CS(2)=W_DEL(X_CS(2),TCTEMP)
          X_CS(4)=W_DEL(X_CS(4),TSTEMP)
        ENDIF

****************************
** START THE BALL ROLLING **
****************************



      CALL RKF(N,X,TBEG,TEND,TOL,DTMAX,DTMIN,ITMAX)


********************************************
** CALL SOLVER FOR CHINA AND SEARLES LAKE **
********************************************


      GUESS=.TRUE.
      GUESS2=.TRUE.

      CALL RKF_CS(NCS,X_CS,TBEG,TEND,TOL_CS,DTMAX_CS,
     +     DTMIN_CS,ITMAX_CS)



      END



      SUBROUTINE RKF(N,X,TBEG,TEND,TOL,DTMAX,DTMIN,ITMAX)
*******************************************************************************
*      SOLVE A SYSTEM OF PARTIAL DIFFERENTIAL EQUATION OF THE FORM:           *
*                         F(T,X)= X'                                          *
*       BETWEEN T1,T2, GIVEN THE INITIAL CONDITION XO(T1)                     *
*******************************************************************************
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      PARAMETER(NUMA=25000,NUMB=2000)
      PARAMETER (VOLMAX=30.02D9,QCL_IN=1.67D8)

      LOGICAL PASS,GRAF,ONLY1,ZEROVOL,ZEROCHK,SU

      SAVE

      COMMON/BOTH/CL(NUMA),TSOD(10),TCL(10),OTIME(NUMA),TCO3(10),
     + OQO(NUMA),QO(10),CONC_CL(NUMA),DOLTEMP,DELDOL,TODELI,TOTEMP,
     + ODEL_OUT(NUMA),OCL_OUT(NUMA),PEVAP(NUMB),DEVAP(NUMB),
     + SUM_PCL_DEP,AREA_P,AREA_D,
     + AREA(NUMA),SOAREA,ALLAREA,CONC_CL_P,
     + CL_P

      COMMON/BOTH2/CL_C,TSOD_C(10),TCL_C(10),TIME,TCO3_C(10),
     + CQO(10),GRAF,CONC_CL_C,DOLTEMP_C,DELDOL_C,
     + TCDELI,TCTEMP,CDEL_OUT,CCL_OUT,CL_S,NOPTS,
     + TSOD_S(10),TCL_S(10),TCO3_S(10),SQO(10),CONC_CL_S,
     + DOLTEMP_S,DELDOL_S,TSDELI,TSTEMP,SCL_DEP,TTLCL_IN,NPTS,
```

```
     + SUM_SCL_DEP,QI_C,QI_S,PQI

       COMMON/PREV/PCL_P,PCONC_CL_P,PSUM_PCL_DEP,PCL_C,PCONC_CL_C,
     + PCDEL_OUT,PCL_S,PCONC_CL_S,PSUM_SCL_DEP,PALLAREA,DATSAV,
     + SDC,SDP1,SDP2

       COMMON/HIST/QIHIST(1000),HUMTIME(500),NHPTS,SU,SUT(15),
     + SAVETIME,NUMST,QITIME(1000),NQPTS

       COMMON/FINAL/FINDEL,FINVOL,FINAREA,FINCL

       COMMON/INFLOW/INCHOICE,A,B,C

       DIMENSION X(3),RK1(3),RK2(3),RK3(3),RK4(3),RK5(3),RK6(3),R(3)
       DIMENSION TERM(3),DEL(3),TOL(3)

**     OPEN(UNIT=95,FILE='QI.OUT',STATUS='UNKNOWN')
       OPEN(UNIT=70,FILE='START.OUT',STATUS='UNKNOWN')

       STEP=DTMAX
       KOUNT=1
       OTIME(KOUNT)=TBEG
       ODEL_OUT(KOUNT)=X(2)
       OCL_OUT(KOUNT)=0.D0
       ONLY1=.TRUE.

*********************************
** TO PROTECT YOU FROM DRYNESS **
*********************************

       ZEROVOL=.FALSE.
       ZEROCHK=.FALSE.

*************************************
** THE SOLVING ROUTINE BEGINS HERE **
*************************************

********************************************
** INITIALIZE PARAMETERS TO RESTART MODEL **
********************************************

       SAVETIME=SUT(1)
       ISTCNT=1
       ITER=0

       WRITE(6,*)
       WRITE(6,*)
       WRITE(6,*)'SOLVING DIFFERENTIAL EQUATIONS FOR OWENS'
       WRITE(6,*)
       WRITE(6,*)

       WHILE(KOUNT .LT. NUMA)DO
          ITER=ITER+1
          IF(OTIME(KOUNT).GT.TEND)THEN
             KNT=1
             T=OTIME(KOUNT)
             DO 200 I=1,N
                RK1(I)=STEP*F(I,X,T,KNT,KOUNT,TBEG,TEND)
200          CONTINUE

***********************************************************************
** "TERM" IS AN ARRAY WHICH STORES APPROXIMATIONS OF VOL AND DEL O-18 **
** WHICH WILL BE USED IN FINAL CALCULATIONS IF ERRORS WITHIN THE STEP **
** ARE LESS THAN THE GIVEN TOLERANCES.                                **
***********************************************************************
```

```
      T=OTIME(KOUNT)-STEP/4.D0
      KNT=2
      DO 300 I=1,N
         TERM(I)=X(I)+ RK1(I)/4.D0
300   CONTINUE

******************************************************************************
** WHEN BOTH "IF'S" ARE SATISFIED YOU ARE AS CLOSE TO ZERO VOLUME AS THE   **
** SOLVER IS GOING TO GET WITH THE GIVEN CONSTRAINTS, SO GO TO THE END OF **
** THE RKF AND MAKE VOLUME=0 AND DEL=DELINFLOW                              **
******************************************************************************

      IF(TERM(1) .LE. 10.D0)THEN
         IF(DABS(STEP-DTMIN).LT.1.D-6)THEN
            ZEROVOL=.TRUE.
            PASS=.TRUE.
            GOTO 1375
         ELSE
            ZEROCHK=.TRUE.
         ENDIF
      ENDIF
      DO 400 I=1,N
         RK2(I)=STEP*F(I,TERM,T,KNT,KOUNT,TBEG,TEND)
400   CONTINUE
      T=OTIME(KOUNT)-3.D0*STEP/8.D0
      KNT=3
      DO 500 I=1,N
         TERM(I)=X(I)+(3.D0*RK1(I)+9.D0*RK2(I))/32.D0
500   CONTINUE
      IF(TERM(1) .LE. 10.D0)THEN
         IF(DABS(STEP-DTMIN).LT.1.D-6)THEN
            ZEROVOL=.TRUE.
            PASS=.TRUE.
            GOTO 1375
         ELSE
            ZEROCHK=.TRUE.
         ENDIF
      ENDIF
      DO 600 I=1,N
         RK3(I)=STEP*F(I,TERM,T,KNT,KOUNT,TBEG,TEND)
600   CONTINUE
      T=OTIME(KOUNT)-12.D0*STEP/13.D0
      KNT=4
      DO 700 I=1,N
         TERM(I)=X(I) + (1932.D0*RK1(I)-7200.D0*RK2(I)+
     +         7296.D0*RK3(I))/2197.D0
700   CONTINUE
      IF(TERM(1) .LE. 10.D0)THEN
         IF(DABS(STEP-DTMIN).LT.1.D-6)THEN
            ZEROVOL=.TRUE.
            PASS=.TRUE.
            GOTO 1375
         ELSE
            ZEROCHK=.TRUE.
         ENDIF
      ENDIF
      DO 800 I=1,N
         RK4(I)=STEP*F(I,TERM,T,KNT,KOUNT,TBEG,TEND)
800   CONTINUE
      T=OTIME(KOUNT)-STEP
      KNT=5
      DO 900 I=1,N
         TERM(I)=X(I) +439.D0*RK1(I)/216.D0-
     +         8.D0*RK2(I)+3680.D0*RK3(I)/513.D0-
     +         845.D0*RK4(I)/4104.D0
900   CONTINUE
```

```
          IF(TERM(1) .LE. 10.D0)THEN
             IF(DABS(STEP-DTMIN).LT.1.D-6)THEN
                ZEROVOL=.TRUE.
                PASS=.TRUE.
                GOTO 1375
             ELSE
                ZEROCHK=.TRUE.
             ENDIF
          ENDIF
          DO 1000 I=1,N
             RK5(I)=STEP*F(I,TERM,T,KNT,KOUNT,TBEG,TEND)
1000      CONTINUE
          T=OTIME(KOUNT)-STEP/2.D0
          KNT=6
          DO 1100 I=1,N
             TERM(I)=X(I)-8.D0*RK1(I)/27.D0+
     +            2.D0*RK2(I)-3544.D0*RK3(I)/2565.D0+
     +            1859.D0*RK4(I)/4104.D0-11.D0*RK5(I)/40.D0
1100      CONTINUE
          IF(TERM(1) .LE. 10.D0)THEN
             IF(DABS(STEP-DTMIN).LT.1.D-6)THEN
                ZEROVOL=.TRUE.
                PASS=.TRUE.
                GOTO 1375
             ELSE
                ZEROCHK=.TRUE.
             ENDIF
          ENDIF
          DO 1200 I=1,N
             RK6(I)=STEP*F(I,TERM,T,KNT,KOUNT,TBEG,TEND)
1200      CONTINUE
          IF(TERM(1) .LE. 10.D0)THEN
             IF(DABS(STEP-DTMIN).LT.1.D-6)THEN
                ZEROVOL=.TRUE.
                PASS=.TRUE.
                GOTO 1375
             ELSE
                ZEROCHK=.TRUE.
             ENDIF
          ENDIF
          PASS=.TRUE.

***********************************************
** CALCULATE ERRORS RESULTING FROM STEP SIZE **
***********************************************

          DO 1300 I=1,N
             R(I)=DABS(RK1(I)/360.D0 -128.D0*RK3(I)/4275.D0-
     +            2197.D0*RK4(I)/75240.D0+RK5(I)/50.D0+
     +            2.D0*RK6(I)/55.D0)/STEP
             IF(R(I).GT.TOL(I)) PASS=.FALSE.
1300      CONTINUE

**************************************************************************
** MAKE SURE THE SOLVER ISN'T "STUCK" BECAUSE OF THE ERROR TOLERANCES  **
**************************************************************************

          IF(DABS(R(1)-RIPREV).LT.1.D-6 .AND.
     +       DABS(STEP-DTMIN).LT.1.D-6)THEN
             IF(ZEROCHK)THEN
                PASS=.TRUE.
                ZEROVOL=.TRUE.
                GOTO 1375
             ELSE
                WRITE(6,*)
                WRITE(6,*)'THE CURRENT RUN IS "STUCK" BUT WE HAVE
```

```
     +FORCED IT TO MOVE ON'
                  WRITE(6,*)'DESPITE THE GIVEN TOLERANCES'
                  WRITE(6,*)
                  PASS=.TRUE.
                  GOTO 1375
               ENDIF
            ELSE
               RIPREV=R(1)
            ENDIF

            DO 1310 I=1,N
               IF(R(I) .LT. 1.D-15)R(I)=.1
1310        CONTINUE
            DELMIN=4.D0

*******************************************************
** 'DEL' IS A VARIABLE USED TO UPDATE THE STEP SIZE **
*******************************************************

            DO 1350 I = 1,N
               DEL(I)=0.84*(TOL(I)/R(I))**(1.D0/4.D0)
               DELMIN=DMIN1(DEL(I),DELMIN)
1350        CONTINUE

*********************************************************
** IF THE ERROR IS LESS THAN THE GIVEN TOLERANCES ... **
*********************************************************

1375        IF(PASS)THEN
               IF(OTIME(KOUNT)-STEP.GE.TEND)THEN
                  KOUNT=KOUNT+1
                  OTIME(KOUNT)=OTIME(KOUNT-1)-STEP
                  IF(ZEROVOL)THEN
                     X(1)=0.D0
                     X(2)=TODELI
                     ZEROVOL=.FALSE.
                     ZEROCHK=.FALSE.
                  ELSE

************************************
** CALCULATE VOLUME AND DEL O-18 **
************************************

                     DO 1400 I=1,N
                        X(I)=X(I)+25.D0*RK1(I)/216.D0+
     +                       1408.D0*RK3(I)/2565.D0+
     +                       2197.D0*RK4(I)/4104.D0- RK5(I)/5.D0
1400                 CONTINUE
                  ENDIF

**************************************************
** SALT BALANCE STUFF FOR THE ENTIRE TIME-STEP **
**************************************************

                  IF(X(1) .GT. VOLMAX)THEN
**************
** OVERFLOW **
**************
                     VOL_OUT=X(1)-VOLMAX
                     CL(KOUNT)=CL(KOUNT-1)+QCL_IN*STEP-
     +                    VOL_OUT*1.D3*CONC_CL(KOUNT-1)
                     IF(CL(KOUNT).LT.0.D0)CL(KOUNT)=3.95D-4*30.02D9*
     +                    1.D3
                     X(1)=VOLMAX
                     CONC_CL(KOUNT)=CL(KOUNT)/(X(1)*1.D3)
```

```
*******************************************************************************
** "OCL_OUT" RECORDS THE SUM MOLES OF CL THAT LEAVE DURING EACH TIME-STEP **
*******************************************************************************

                  OCL_OUT(KOUNT)= VOL_OUT*1.D3*CONC_CL(KOUNT-1)+
     +                 OCL_OUT(KOUNT-1)

               ELSE IF(X(1) .LE. 10.D0)THEN
**************
** DRY LAKE **
**************
                  OCL_OUT(KOUNT)=OCL_OUT(KOUNT-1)
                  CL(KOUNT)=0.D0
                  X(1)=0.D0
                  CONC_CL(KOUNT)=0.D0
               ELSE

******************************
** BETWEEN DRY AND OVERFLOW **
******************************

                  OCL_OUT(KOUNT)=OCL_OUT(KOUNT-1)
                  CL(KOUNT)=CL(KOUNT-1)+QCL_IN*STEP
                  CONC_CL(KOUNT)=CL(KOUNT)/(X(1)*1.D3)

******************************
** CHECK FOR CL SATURATION **
******************************

                  IF(CONC_CL(KOUNT) .GT. 6.1D0)THEN
                     CONC_CL(KOUNT)=6.1D0
                     CL(KOUNT)=6.1D0*(X(1)*1.D3)
                  ENDIF
               ENDIF

**************************************************
** CALCULATE DEL DOLOMITE FROM DEL WATER, X(2)**
**************************************************

               ODEL_OUT(KOUNT)=X(2)
               DELDOL=FDDOL(DOLTEMP,X(2))

*******************************************************
** STORE VALUES IN ARRAYS FOR THE PLOTTING ROUTINE **
*******************************************************

               CALL FINDQT(NQPTS,OTIME(KOUNT),IQNDEX,QITIME)

               IF(INCHOICE .GE. 8)THEN
                  CALL QINTERP(OTIME(KOUNT),IQNDEX,QI,QIHIST,
     +                 QITIME)
               ELSE
                  QI=FQI(TBEG-OTIME(KOUNT))
               ENDIF
               IF(QI .LE. 0.D0)QI=0.D0

***************************
** WRITE RESULTS TO FILE **
***************************

               AREA(KOUNT)=OAREA(X(1))
*               WRITE(96,*)OTIME(KOUNT),AREA(KOUNT)
*               WRITE(99,*)OTIME(KOUNT),DELDOL
**              WRITE(95,*)OTIME(KOUNT),QI

               IF(SU)THEN
```

```
                  IF(OTIME(KOUNT) .LT. SAVETIME .AND. ISTCNT .LE.
     +                  NUMST)THEN
                     WRITE(70,*)'OWENS LAKE'
                     WRITE(70,*)'STARTUP TIME #',ISTCNT
                     WRITE(70,*)OTIME(KOUNT),X(1),DELDOL,
     +                       CONC_CL(KOUNT)
                     WRITE(70,*)
                     ISTCNT=ISTCNT+1
                     SAVETIME=SUT(ISTCNT)
                  ENDIF
               ENDIF

               STEP=STEP+0.5D0*STEP
               IF(STEP .GT. DTMAX)STEP=DTMAX
               GOTO 100

***********************************
** MAKE SURE THE SOLVER DOESN'T **
** OVERSTEP DESIGNATED END TIME **
***********************************

*********************************************************************
** THIS ELSE IF CORRESPONDS TO "IF(OTIME(KOUNT)-STEP.GE.TEND)THEN **
*********************************************************************

            ELSEIF(OTIME(KOUNT)-STEP.LT.TEND)THEN
               DTMAX=OTIME(KOUNT)-TEND
               STEP=DTMAX
               GOTO 100
            ENDIF

***************************************
** ADJUST THE SIZE OF THE TIME STEP **
***************************************

***********************************************
** THIS ELSEIF CORRESPONDS TO "IF(PASS)THEN" **
***********************************************

         ELSEIF(DELMIN .LE. 0.1)THEN
            STEP=STEP*1.0D-1
         ELSEIF(DELMIN .GE. 4.D0)THEN
            STEP=4.D0*STEP
         ELSE
            STEP=DELMIN*STEP
         ENDIF
         IF(STEP.GT.DTMAX)STEP=DTMAX
         IF(STEP.LT.DTMIN)STEP=DTMIN

**************************************************************
** THIS ELSE CORRESPONDS TO "IF(OTIME(KOUNT).GT.TEND)THEN" **
**************************************************************

      ELSE
         IF(.NOT. GRAF)THEN
            WRITE(6,*)
            WRITE(6,*)'THE RKF SOLVED',KOUNT,' POINTS IN',ITER,' IT
     +ERATIONS'
            WRITE(6,*)
            WRITE(6,*)'FINAL DEL FOR OWENS =',DELDOL
            WRITE(6,*)'FINAL AREA FOR OWENS  =',AREA(KOUNT)
            WRITE(6,*)'FINAL VOLUME FOR OWENS =',X(1)
            WRITE(6,*)'FINAL CL CONC, OWENS =',CONC_CL(KOUNT)
            FINDEL=DELDOL
            FINAREA=AREA(KOUNT)
            FINVOL=X(1)
```

```
              FINCL=CONC_CL(KOUNT)
           ENDIF

           NOPTS=KOUNT
           RETURN
        ENDIF
100   ENDWHILE
      NOPTS=KOUNT
**    IF(ITER .GT. ITMAX)WRITE(6,*)'MAX # OF ITERATIONS EXCEEDED'
      IF(.NOT. GRAF)THEN
         WRITE(6,*)
         WRITE(6,*)'THE RKF SOLVED',KOUNT,' POINTS IN',ITER,' IT
     +ERATIONS'
         WRITE(6,*)
         WRITE(6,*)'FINAL DEL FOR OWENS =',DELDOL
         WRITE(6,*)'FINAL AREA FOR OWENS  =',AREA(KOUNT)
         WRITE(6,*)'FINAL VOLUME FOR OWENS =',X(1)
         WRITE(6,*)'FINAL CL CONC, OWENS =',CONC_CL(KOUNT)
         FINDEL=DELDOL
         FINAREA=AREA(KOUNT)
         FINVOL=X(1)
         FINCL=CONC_CL(KOUNT)
         WRITE(6,*)
      ENDIF

      END


********************************************************************************
********************************************************************************
*    BELOW LIES A CHAOTIC CONVOLUTION OF ESOTERIC ENIGMAS THAT HOPEFULLY      *
*    ACCOMPLISH THE ISOTOPIC AND LAKE LEVEL VOODOO WE SET OUT TOODOO.         *
*    ACTUALLY THIS PART OF THE PROGRAM CALCULATES THE DERIVATIVES OF LAKE     *
*    VOLUME AND ISOTOPIC COMPOSITION WITH RESPECT TO TIME.                    *
********************************************************************************
********************************************************************************

      DOUBLE PRECISION FUNCTION F(I,TERM,T,KNT,KOUNT,TBEG,TEND)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      LOGICAL FOUND,GUESS,GRAF,CPARAM,GUESS2,SU

      PARAMETER(NUMA=25000,NUMB=2000)
      PARAMETER(QCL_IN=1.67D8,VOLMAX=30.02D9)

      SAVE

      COMMON/BOTH/CL(NUMA),TSOD(10),TCL(10),OTIME(NUMA),TCO3(10),
     + OQO(NUMA),QO(10),CONC_CL(NUMA),DOLTEMP,DELDOL,TODELI,TOTEMP,
     + ODEL_OUT(NUMA),OCL_OUT(NUMA),PEVAP(NUMB),DEVAP(NUMB),
     + SUM_PCL_DEP,AREA_P,AREA_D,
     + AREA(NUMA),SOAREA,ALLAREA,CONC_CL_P,
     + CL_P

      COMMON/BOTH2/CL_C,TSOD_C(10),TCL_C(10),TIME,TCO3_C(10),
     + CQO(10),GRAF,CONC_CL_C,DOLTEMP_C,DELDOL_C,
     + TCDELI,TCTEMP,CDEL_OUT,CCL_OUT,CL_S,NOPTS,
     + TSOD_S(10),TCL_S(10),TCO3_S(10),SQO(10),CONC_CL_S,
     + DOLTEMP_S,DELDOL_S,TSDELI,TSTEMP,SCL_DEP,TTLCL_IN,NPTS,
     + SUM_SCL_DEP,QI_C,QI_S,PQI

      COMMON/PREV/PCL_P,PCONC_CL_P,PSUM_PCL_DEP,PCL_C,PCONC_CL_C,
     + PCDEL_OUT,PCL_S,PCONC_CL_S,PSUM_SCL_DEP,PALLAREA,DATSAV,
     + SDC,SDP1,SDP2
```

```
      COMMON/FIRST/WTIME(NUMB),OTEMP(NUMB),OEVAP(NUMB),
     + OHUM(500),OPRECIP(NUMB),ODELP(NUMB),ODELA(NUMB),ODELI(NUMB),
     + GUESS,TEMPC,EVAPC,PRECIPC,DELAC,DELIC,CPARAM,DELPC,
     + EVAPC_P,EVAPC_D

      COMMON/SECOND/CTEMP(NUMB),CEVAP(NUMB),CHUM(500),CPRECIP(NUMB),
     + CDELP(NUMB),CSDELA(NUMB),GUESS2,TEMPC_C,EVAPC_C,
     + PRECIPC_C,DELAC_C,DELPC_C,STEMP(NUMB),SEVAP(NUMB),SHUM(500),
     + SPRECIP(NUMB),SDELP(NUMB),TEMPC_S,EVAPC_S,
     + PRECIPC_S,DELAC_S,DELPC_S

      COMMON/HIST/QIHIST(1000),HUMTIME(500),NHPTS,SU,SUT(15),
     + SAVETIME,NUMST,QITIME(1000),NQPTS

      COMMON/INFLOW/INCHOICE,A,B,C

      DIMENSION TERM(3),TCONC_CL(10),V_OUT(10)

      IF(I.EQ.1)THEN

************************************
** CALCULATE DV/DT FOR OWENS LAKE **
************************************

         VOL = TERM(1)
         IF(VOL .LE. 10.D0)VOL=0.D0


*******************************
** CONSTANT PARAMETER OPTION **
*******************************

         IF(CPARAM)THEN
            TODELI=DELIC
            TOTEMP=TEMPC
            TOEVAP=EVAPC
            TOPRECIP=PRECIPC
            TODELP=DELPC
            TODELA=DELAC
            CALL FINDHT(NHPTS,T,IHNDEX,HUMTIME)
            CALL FINDQT(NQPTS,T,IQNDEX,QITIME)
            CALL HUMTERP(T,IHNDEX,TOHUM,OHUM,HUMTIME)
         ELSE

**********************************************************************
** DETERMINE VALUES OF NECESSARY PARAMETERS BY ASSIGNING VALUES OR  **
** INTERPOLATING BETWEEN GIVEN VALUES                               **
**********************************************************************

            CALL FINDT(NPTS,T,INDEX,FOUND,WTIME)
            CALL FINDHT(NHPTS,T,IHNDEX,HUMTIME)
            CALL FINDQT(NQPTS,T,IQNDEX,QITIME)

            IF(FOUND)THEN
               TODELI = ODELI(INDEX)
               TOTEMP = OTEMP(INDEX)
               TOEVAP = OEVAP(INDEX)
               TOPRECIP = OPRECIP(INDEX)
               TODELP = ODELP(INDEX)
               TODELA = ODELA(INDEX)
            ELSE
               CALL OINTERP(T,INDEX,TODELI,TOTEMP,TOEVAP,TOPRECIP,
     +              TODELP,TODELA)
            ENDIF
            CALL HUMTERP(T,IHNDEX,TOHUM,OHUM,HUMTIME)
         ENDIF
```

```
*********************************************************************
** "REMEMBER" TEMP AT END OF TIMESTEP TO CALCULATE DEL DOLOMITE **
*********************************************************************

         IF(KNT .EQ. 5)DOLTEMP=TOTEMP

**************************************************
** TRACK TOTAL ELAPSED TIME TO CALCULATE INFLOW**
**************************************************

         ETIME = TBEG-T

******************************************************
** ALSO NEED TO KNOW DEL TIME WITHIN THE TIME-STEP **
******************************************************

         DTIME = OTIME(KOUNT)-T

****************************
** CALCULATE AREA OF LAKE **
****************************

         AREA(KOUNT) = OAREA(VOL)

*****************************
** CALCULATE PRECIPITATION **
*****************************

         QP = AREA(KOUNT)*TOPRECIP

**********************************
** THE INFAMOUS "SALT" BALANCE **
**********************************

         IF(DABS(T-OTIME(KOUNT)).LT.1.D-8)THEN
            TCL(KNT)=CL(KOUNT)
            TCONC_CL(KNT)=CONC_CL(KOUNT)
            TCO3(KNT) = TCL(KNT)*1.03D0
            TSOD(KNT)=TCL(KNT)+TCO3(KNT)
****************************************************
** CALCULATE TOTAL OUTFLOW VOLUME FROM TIME TO T **
** AND ADJUST IF VOL EXCEEDS VOLMAX              **
****************************************************

         ELSE
            IF(VOL .LE. VOLMAX)THEN
               V_OUT(KNT)=0.D0
            ELSE
               V_OUT(KNT) = VOL-VOLMAX
               VOL=VOLMAX
            ENDIF

**********************************************************
**  CALCULATE CONCENTRATIONS FOR INTERMEDIATE TIME "T" **
**********************************************************

            TCL(KNT) = (OTIME(KOUNT)-T)*QCL_IN+CL(KOUNT)-V_OUT(KNT)*
     +           CONC_CL(KOUNT)*1.D3
            IF(TCL(KNT).LT.0.D0)TCL(KNT)=3.95D-4*30.02D9*1.D3

**********************************************************************
** CONCENTRATION UNITS ARE MOLARITY SO VOL MUST BE MULT BY 1000 TO **
** CONVERT M^3 TO LITERS **
**********************************************************************
```

```
          IF(VOL .GT. 10.D0)THEN
             TCONC_CL(KNT)=TCL(KNT)/(VOL*1.0D3)
          ELSE
             TCONC_CL(KNT)=0.D0
             TCL(KNT)=0.D0
          ENDIF

***********************************
** CHECK FOR CHLORIDE SATURATION **
***********************************

          IF(TCONC_CL(KNT) .GT. 6.1D0)THEN
             TCONC_CL(KNT)=6.1D0
             TCL(KNT)=6.1D0*(VOL*1.D3)
          ENDIF

***************************************************
** TOTAL CO3 IS KEPT AT A CONSTANT RATIO WITH CL **
***************************************************

          TCO3(KNT) = TCL(KNT)*1.03D0

********************************************************************
** AMT OF SODIUM IS THE AMT NECESSARY TO ACHIEVE ELECTRONEUTALITY **
********************************************************************

          TSOD(KNT)=TCL(KNT)+TCO3(KNT)
        ENDIF

********************************************
**  CALCULATE BACK-CONDENSATION FLUX (QC) **
********************************************

        IF(VOL .GT. 10.D0 .AND. TCONC_CL(KNT) .GT. 0.D0)THEN
           PHI=FPHI(TCONC_CL(KNT),SUM)
        ELSE
           PHI=0.D0
        ENDIF

        AW=DEXP(-18.D0*PHI*SUM*0.5D0/1.D3)

********************************
** CALCULATE EVAPORATION (QE) **
********************************

        QE = AREA(KOUNT)*TOEVAP*AW

        IF(VOL .GT. 10.D0)THEN
           QC=(TOHUM*QE)/AW
        ELSE
           QC=0.D0
        ENDIF

*************************************
**  CALCULATE DV/DT IF VOL IS ZERO **
*************************************

        IF(TERM(1) .LE. 10.D0)THEN
           IF(INCHOICE .GE. 8)THEN
              CALL QINTERP(T,IQNDEX,QI,QIHIST,QITIME)
           ELSE
              QI=FQI(ETIME)
           ENDIF
           IF(QI .LT. 0.D0)QI=0.D0
           IF(GUESS)THEN
*             WRITE(96,*)TBEG,AREA(KOUNT)
```

```
**             WRITE(95,*)TBEG,QI
               DELDOL=FDDOL(TOTEMP,TERM(2))
*              WRITE(99,*)TBEG,DELDOL
               GUESS=.FALSE.
            ENDIF

            QO(KNT)=0.D0

*****************************************
** ASSUME QO FOR OWENS = QI FOR CHINA **
*****************************************

            IF(KNT .EQ. 1)THEN
               OQO(KOUNT)=QO(KNT)
*              WRITE(97,*)T,OQO(KOUNT)
            ENDIF
            IF(DABS(T-TEND).LT.1.D-8)THEN
               OQO(KOUNT+1)=QO(KNT)
*              WRITE(97,*)T,OQO(KOUNT)
            ENDIF

            F=QI

            IF(F .LE. 0.D0)THEN
               DV_DT=0.D0
            ELSE
               DV_DT=F
            ENDIF

**************************************************
**  CALCULATE DV/DT IF VOL IS LESS THAN VOLMAX **
**************************************************

         ELSE IF(TERM(1)  .LT. VOLMAX)THEN

****************************
** CALCULATE INFLOW (QI) **
****************************

            IF(INCHOICE .GE. 8)THEN
               CALL QINTERP(T,IQNDEX,QI,QIHIST,QITIME)
            ELSE
               QI=FQI(ETIME)
            ENDIF
            IF(QI .LT. 0.D0)QI=0.D0
            IF(GUESS)THEN
*              WRITE(96,*)TBEG,AREA(KOUNT)
**             WRITE(95,*)TBEG,QI
               DELDOL=FDDOL(TOTEMP,TERM(2))
*              WRITE(99,*)TBEG,DELDOL
               GUESS=.FALSE.
            ENDIF

***********************
**  CALCULATE DV/DT **
***********************

            F=QI+QC-QE+QP
            DV_DT=F
            QO(KNT)=0.D0
            IF(KNT .EQ. 1)THEN
               OQO(KOUNT)=QO(KNT)
*              WRITE(97,*)T,OQO(KOUNT)
            ENDIF
            IF(DABS(T-TEND).LT.1.D-8)THEN
               OQO(KOUNT+1)=QO(KNT)
```

```
*                WRITE(97,*)T,OQO(KOUNT)
              ENDIF

********************************************
**  CALCULATE DV/DT IF OWENS LAKE IS FULL **
********************************************

           ELSE

******************
** CALCULATE QI **
******************

              IF(INCHOICE .GE. 8)THEN
                 CALL QINTERP(T,IQNDEX,QI,QIHIST,QITIME)
              ELSE
                 QI=FQI(ETIME)
              ENDIF

              IF(GUESS)THEN
*                 WRITE(96,*)TBEG,AREA(KOUNT)
**               WRITE(95,*)TBEG,QI
                 DELDOL=FDDOL(TOTEMP,TERM(2))
*                WRITE(99,*)TBEG,DELDOL
                 GUESS=.FALSE.
              ENDIF

******************
** CALCULATE QO **
******************

              QO(KNT)=QI+QC-QE+QP

*********************
** CALCULATE DV/DT **
*********************

              IF(QO(KNT) .LT. 0.D0)THEN
                 QO(KNT)=0.D0
                 IF(KNT .EQ. 1)THEN
                    OQO(KOUNT)=QO(KNT)
*                   WRITE(97,*)T,OQO(KOUNT)
                 ENDIF
                 IF(DABS(T-TEND).LT.1.D-8)THEN
                    OQO(KOUNT+1)=QO(KNT)
*                   WRITE(97,*)T,OQO(KOUNT)
                 ENDIF
                 F=QI+QC-QE+QP
                 DV_DT=F
              ELSE
                 F=QO(KNT)
                 IF(KNT .EQ. 1)THEN
                    OQO(KOUNT)=QO(KNT)
*                   WRITE(97,*)T,OQO(KOUNT)
                 ENDIF
                 IF(DABS(T-TEND).LT.1.D-8)THEN
                    OQO(KOUNT+1)=QO(KNT)
*                   WRITE(97,*)T,OQO(KOUNT)
                 ENDIF
                 DV_DT=0.D0
              ENDIF
           ENDIF

******************************************************************************
*    A FUNCTION SUBROUTINE TO CALCULATE THE ISOTOPIC HISTORY OF OWENS LAKE    *
******************************************************************************
```

```
      ELSE IF(I.EQ.2)THEN

         DELL = TERM(2)

************************
** CALCULATE DDEL/DT  **
************************

         IF(VOL .LE. 10.D0)THEN
            F=0.D0

         ELSE

******************************************
** CALCULATE ISOTOPIC ENRICHMENT FACTOR **
******************************************

            EPS = FEPS(TOTEMP)

********************************************
** CALCULATE DEL OF THE BACK-CONDENSATION **
********************************************

            ODELC =EPS*(1.D0+(TODELA/1.D3))+TODELA

**************************************
** CALCULATE DEL OF THE EVAPORATION **
**************************************

            ODELE = DELE(DELL,EPS,TOHUM)

*****************************************************
** SET DEL OF THE OUTFLOW EQUAL TO DEL OF THE LAKE **
*****************************************************

            ODELO = DELL

            F=(QI*TODELI+QC*ODELC+QP*TODELP-QO(KNT)*ODELO-QE*
     +           ODELE-DELL*DV_DT)/VOL

         ENDIF

      END IF
      RETURN
      END



      SUBROUTINE RKF_CS(NCS,X_CS,TBEG,TEND,TOL_CS,DTMAX_CS,DTMIN_CS,
     +     ITMAX_CS)
********************************************************************************
*     SOLVE A SYSTEM OF PARTIAL DIFFERENTIAL EQUATION OF THE FORM:             *
*                       F(T,X)= X'                                              *
*      BETWEEN T1,T2, GIVEN THE INITIAL CONDITION XO(T1)                        *
********************************************************************************
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      PARAMETER(NUMA=25000,NUMB=2000)
      PARAMETER (VOLMAX_C=0.696D9,VOLMAX_S=85.28D9,AREAMAX_P=.727D9,
     +     AREAMAX_D=0.583D9,SLTCONST=0.0127)

      LOGICAL PASS,GRAF,ONLY1,ZEROVOL_C,ZEROCHK_C,ZEROVOL_S,
     + ZEROCHK_S,COAL,DECOUP,COUP,FOUND,CPARAM,GUESS,SU

      SAVE
```

```
      COMMON/FIRST/WTIME(NUMB),OTEMP(NUMB),OEVAP(NUMB),
     + OHUM(500),OPRECIP(NUMB),ODELP(NUMB),ODELA(NUMB),ODELI(NUMB),
     + GUESS,TEMPC,EVAPC,PRECIPC,DELAC,DELIC,CPARAM,DELPC,
     + EVAPC_P,EVAPC_D

      COMMON/BOTH/CL(NUMA),TSOD(10),TCL(10),OTIME(NUMA),TCO3(10),
     + OQO(NUMA),QO(10),CONC_CL(NUMA),DOLTEMP,DELDOL,TODELI,TOTEMP,
     + ODEL_OUT(NUMA),OCL_OUT(NUMA),PEVAP(NUMB),DEVAP(NUMB),
     + SUM_PCL_DEP,AREA_P,AREA_D,
     + AREA(NUMA),SOAREA,ALLAREA,CONC_CL_P,
     + CL_P

      COMMON/BOTH2/CL_C,TSOD_C(10),TCL_C(10),TIME,TCO3_C(10),
     + CQO(10),GRAF,CONC_CL_C,DOLTEMP_C,DELDOL_C,
     + TCDELI,TCTEMP,CDEL_OUT,CCL_OUT,CL_S,NOPTS,
     + TSOD_S(10),TCL_S(10),TCO3_S(10),SQO(10),CONC_CL_S,
     + DOLTEMP_S,DELDOL_S,TSDELI,TSTEMP,SCL_DEP,TTLCL_IN,NPTS,
     + SUM_SCL_DEP,QI_C,QI_S,PQI

      COMMON/PREV/PCL_P,PCONC_CL_P,PSUM_PCL_DEP,PCL_C,PCONC_CL_C,
     + PCDEL_OUT,PCL_S,PCONC_CL_S,PSUM_SCL_DEP,PALLAREA,DATSAV,
     + SDC,SDP1,SDP2

      COMMON/NEW/OPREV,CPREV,SPREV,PPREV,DPREV

      COMMON/HIST/QIHIST(1000),HUMTIME(500),NHPTS,SU,SUT(15),
     + SAVETIME,NUMST,QITIME(1000),NQPTS

      COMMON/FINAL/FINDEL,FINVOL,FINAREA,FINCL

      DIMENSION X_CS(4),RK1(4),RK2(4),RK3(4),RK4(4),RK5(4),RK6(4),R(4)
      DIMENSION TERM(4),DEL(4),TOL_CS(4)

      OPEN(UNIT=87,FILE='CONC_S.OUT',STATUS='UNKNOWN')
      OPEN(UNIT=81,FILE='ALLAREA.OUT',STATUS='UNKNOWN')
      OPEN(UNIT=78,FILE='DEL_S.OUT',STATUS='UNKNOWN')
      OPEN(UNIT=77,FILE='AREA_S.OUT',STATUS='UNKNOWN')
      OPEN(UNIT=76,FILE='QI_S.OUT',STATUS='UNKNOWN')
      OPEN(UNIT=74,FILE='SUMCL.OUT',STATUS='UNKNOWN')

      OPEN(UNIT=91,FILE='QO_S.OUT',STATUS='UNKNOWN')
**    OPEN(UNIT=83,FILE='PCL_DEP.OUT',STATUS='UNKNOWN')
      OPEN(UNIT=55,FILE='RES.OUT',STATUS='UNKNOWN')
      OPEN(UNIT=56,FILE='UPDATE.OUT',STATUS='UNKNOWN')


      XMIN=TBEG
      XMAX=TEND

      TIME=TBEG
      WRTM=TBEG
      STEP=DTMAX_CS
      KOUNT=1

      PCDEL_OUT=X_CS(2)
      SCL_DEP=0.D0
      CCL_OUT=0.D0

***********************************
** WRITE INITIAL VALUES TO FILE **
***********************************

      WRITE(74,*)TBEG/1.D6,PSUM_SCL_DEP
**    WRITE(83,*)TBEG/1.D6,PSUM_PCL_DEP
      WRITE(87,*)TBEG/1.D6,PCONC_CL_S
```

```
      ONLY1=.TRUE.
      COAL=.FALSE.
      DECOUP=.FALSE.
      COUP=.FALSE.
      ZEROVOL_C=.FALSE.
      ZEROCHK_C=.FALSE.
      ZEROVOL_S=.FALSE.
      ZEROCHK_S=.FALSE.

      IF(X_CS(3) .GT. 65.87D9)COAL=.TRUE.

*********************************************
** INITIALIZE PARAMETERS TO RESTART MODEL **
*********************************************

      SAVETIME=SUT(1)
      ISTCNT=1
      ITER=0

*************************************
** THE SOLVING ROUTINE BEGINS HERE **
*************************************

      WRITE(6,*)
      WRITE(6,*)
      WRITE(6,*)'SOLVING DIFFERENTIAL EQUATIONS'
      WRITE(6,*)'FOR CHINA AND SEARLES LAKE'
      WRITE(6,*)
      WRITE(6,*)

      WHILE(ITER .LT. 50000)DO
         ITER=ITER+1
         IF(TIME.GT.TEND)THEN

            IF(COAL)THEN
               NST=3
            ELSE
               NST=1
            ENDIF

            KNT=1
            T=TIME

            ZEROVOL_C=.FALSE.
            ZEROCHK_C=.FALSE.
            ZEROVOL_S=.FALSE.
            ZEROCHK_S=.FALSE.

            DO 200 I=NST,NCS
               RK1(I)=STEP*F_CS(I,X_CS,T,KNT,KOUNT,TBEG,COAL)
200         CONTINUE

***********************************************************************
** "TERM" IS AN ARRAY WHICH STORES APPROXIMATIONS OF VOL AND DEL O-18 **
** WHICH WILL BE USED IN FINAL CALCULATIONS IF ERRORS WITHIN THE STEP **
** ARE LESS THAN THE GIVEN TOLERANCES.                                **
***********************************************************************

            T=TIME-STEP/4.D0
            KNT=2
            DO 300 I=NST,NCS
               TERM(I)=X_CS(I)+ RK1(I)/4.D0
300         CONTINUE
            IF(TERM(1) .LE. 10.D0 .AND. .NOT. COAL)THEN
               IF(DABS(STEP-DTMIN_CS).LT.1.D-8)THEN
```

```
                ZEROVOL_C=.TRUE.
                PASS=.TRUE.
             ELSE
                ZEROCHK_C=.TRUE.
             ENDIF
          ENDIF
          IF(TERM(3) .LE. 10.D0)THEN
             IF(DABS(STEP-DTMIN_CS).LT.1.D-8)THEN
                ZEROVOL_S=.TRUE.
                PASS=.TRUE.
             ELSE
                ZEROCHK_S=.TRUE.
             ENDIF
          ENDIF
          DO 400 I=NST,NCS
             RK2(I)=STEP*F_CS(I,TERM,T,KNT,KOUNT,TBEG,COAL)
400       CONTINUE
          T=TIME-3.D0*STEP/8.D0
          KNT=3
          DO 500 I=NST,NCS
             TERM(I)=X_CS(I)+(3.D0*RK1(I)+9.D0*RK2(I))/32.D0
500       CONTINUE

          IF(TERM(1) .LE. 10.D0 .AND. .NOT. COAL)THEN
             IF(DABS(STEP-DTMIN_CS).LT.1.D-8)THEN
                ZEROVOL_C=.TRUE.
                PASS=.TRUE.
             ELSE
                ZEROCHK_C=.TRUE.
             ENDIF
          ENDIF
          IF(TERM(3) .LE. 10.D0)THEN
             IF(DABS(STEP-DTMIN_CS).LT.1.D-8)THEN
                ZEROVOL_S=.TRUE.
                PASS=.TRUE.
             ELSE
                ZEROCHK_S=.TRUE.
             ENDIF
          ENDIF

          DO 600 I=NST,NCS
             RK3(I)=STEP*F_CS(I,TERM,T,KNT,KOUNT,TBEG,COAL)
600       CONTINUE
          T=TIME-12.D0*STEP/13.D0
          KNT=4
          DO 700 I=NST,NCS
             TERM(I)=X_CS(I)+(1932.D0*RK1(I)-7200.D0*RK2(I)+
     +                  7296.D0*RK3(I))/2197.D0
700       CONTINUE

          IF(TERM(1) .LE. 10.D0 .AND. .NOT. COAL)THEN
             IF(DABS(STEP-DTMIN_CS).LT.1.D-8)THEN
                ZEROVOL_C=.TRUE.
                PASS=.TRUE.
             ELSE
                ZEROCHK_C=.TRUE.
             ENDIF
          ENDIF
          IF(TERM(3) .LE. 10.D0)THEN
             IF(DABS(STEP-DTMIN_CS).LT.1.D-8)THEN
                ZEROVOL_S=.TRUE.
                PASS=.TRUE.
             ELSE
                ZEROCHK_S=.TRUE.
             ENDIF
          ENDIF
```

```
          DO 800 I=NST,NCS
             RK4(I)=STEP*F_CS(I,TERM,T,KNT,KOUNT,TBEG,COAL)
800       CONTINUE
          T=TIME-STEP
          KNT=5
          DO 900 I=NST,NCS
             TERM(I)=X_CS(I)+439.D0*RK1(I)/216.D0-
     +              8.D0*RK2(I)+3680.D0*RK3(I)/513.D0-
     +              845.D0*RK4(I)/4104.D0
900       CONTINUE

          IF(TERM(1) .LE. 10.D0 .AND. .NOT. COAL)THEN
             IF(DABS(STEP-DTMIN_CS).LT.1.D-8)THEN
                ZEROVOL_C=.TRUE.
                PASS=.TRUE.
             ELSE
                ZEROCHK_C=.TRUE.
             ENDIF
          ENDIF
          IF(TERM(3) .LE. 10.D0)THEN
             IF(DABS(STEP-DTMIN_CS).LT.1.D-8)THEN
                ZEROVOL_S=.TRUE.
                PASS=.TRUE.
             ELSE
                ZEROCHK_S=.TRUE.
             ENDIF
          ENDIF

          DO 1000 I=NST,NCS
             RK5(I)=STEP*F_CS(I,TERM,T,KNT,KOUNT,TBEG,COAL)
1000      CONTINUE
          T=TIME-STEP/2.D0
          KNT=6
          DO 1100 I=NST,NCS
             TERM(I)=X_CS(I)-8.D0*RK1(I)/27.D0+
     +              2.D0*RK2(I)-3544.D0*RK3(I)/2565.D0+
     +              1859.D0*RK4(I)/4104.D0-11.D0*RK5(I)/40.D0
1100      CONTINUE

          IF(TERM(1) .LE. 10.D0 .AND. .NOT. COAL)THEN
             IF(DABS(STEP-DTMIN_CS).LT.1.D-8)THEN
                ZEROVOL_C=.TRUE.
                PASS=.TRUE.
             ELSE
                ZEROCHK_C=.TRUE.
             ENDIF
          ENDIF
          IF(TERM(3) .LE. 10.D0)THEN
             IF(DABS(STEP-DTMIN_CS).LT.1.D-8)THEN
                ZEROVOL_S=.TRUE.
                PASS=.TRUE.
             ELSE
                ZEROCHK_S=.TRUE.
             ENDIF
          ENDIF

          DO 1200 I=NST,NCS
             RK6(I)=STEP*F_CS(I,TERM,T,KNT,KOUNT,TBEG,COAL)
1200      CONTINUE

          IF(ZEROVOL_C .OR. ZEROVOL_S)GOTO 1375

          PASS=.TRUE.

************************************************
```

```
** CALCULATE ERRORS RESULTING FROM STEP SIZE **
************************************************

            DO 1300 I=NST,NCS
               R(I)=DABS(RK1(I)/360.D0 -128.D0*RK3(I)/4275.D0-
     +              2197.D0*RK4(I)/75240.D0+RK5(I)/50.D0+
     +              2.D0*RK6(I)/55.D0)/STEP

               IF(R(I).GT.TOL_CS(I))PASS=.FALSE.

               IF(R(I).GT.TOL_CS(I).AND. DABS(STEP-DTMIN_CS)
     +             .LT.1.D-8)THEN
                  WRITE(6,*)
                  WRITE(6,*)'STUCK ...',I,TOL_CS(I),R(I)
                  WRITE(6,*)
               endif

1300        CONTINUE

******************************************************************************
** MAKE SURE THE SOLVER ISN'T "STUCK" BECAUSE OF THE ERROR TOLERANCES **
******************************************************************************

            IF(DABS(R(1)-RIPREV).LT.1.D-6 .AND.
     +         DABS(STEP-DTMIN_CS).LT.1.D-6)THEN
               IF(ZEROCHK_C)ZEROVOL_C=.TRUE.
               IF(ZEROCHK_S)ZEROVOL_S=.TRUE.
               IF(ZEROVOL_C .OR. ZEROVOL_S .AND. .NOT. COAL)THEN
                  PASS=.TRUE.
                  GOTO 1375
               ELSE
                  PASS=.TRUE.
                  GOTO 1375
               ENDIF
            ELSE
               RIPREV=R(1)
            ENDIF

            DO 1310 I=NST,NCS
               IF(R(I) .LT. 1.0D-3)R(I)=.1
1310        CONTINUE
            DELMIN=4.D0

*********************************************************
** 'DEL' IS A VARIABLE USED TO UPDATE THE STEP SIZE **
*********************************************************

            DO 1350 I = NST,NCS
               DEL(I)=0.84*(TOL_CS(I)/R(I))**(1.D0/4.D0)
               DELMIN=DMIN1(DEL(I),DELMIN)
1350        CONTINUE

**********************************************************
** IF THE ERROR IS LESS THAN THE GIVEN TOLERANCES ... **
**********************************************************

1375        IF(PASS)THEN

               IF(TIME-STEP.GE.TEND)THEN
                  KOUNT=KOUNT+1
                  SLTCORR=SLTCONST*STEP
                  TIME=TIME-STEP

                  IF(COAL)THEN
```

```
************************************************************
** CHECK TO SEE IF CHINA AND SEARLES ARE STILL COALESCED **
************************************************************

                  DO 1700 I=3,4
                     X_CS(I)=X_CS(I)+25.D0*RK1(I)/216.D0+
     +               1408.D0*RK3(I)/2565.D0+
     +               2197.D0*RK4(I)/4104.D0- RK5(I)/5.D0
1700              CONTINUE

                  IF(X_CS(3) .LE. 65.87D9)THEN
                     COAL=.FALSE.
                     X_CS(1)=0.696D9
                     X_CS(2)=X_CS(4)
                     X_CS(3)=X_CS(3)-X_CS(1)
                     WRITE(6,*)
                     WRITE(6,*)'CHINA AND SEARLES HAVE DECOUPLED'
                     WRITE(6,*)'AT',TIME
                     WRITE(6,*)
                     DECOUP=.TRUE.
                  ELSE
                     X_CS(1)=0.D0
                  ENDIF

               ELSE IF(ZEROVOL_C .AND. ZEROVOL_S)THEN

***********************************************************
** CALCULATE VOLUME AND DEL O-18 IF BOTH LAKES ARE  DRY**
***********************************************************

                  X_CS(1)=0.D0
                  X_CS(2)=TCDELI
                  ZEROVOL_C=.FALSE.
                  ZEROCHK_C=.FALSE.
                  X_CS(3)=0.D0
                  X_CS(4)=PCDEL_OUT
                  ZEROVOL_S=.FALSE.
                  ZEROCHK_S=.FALSE.

               ELSE IF(ZEROVOL_C .AND. .NOT. ZEROVOL_S)THEN

*********************************************************
** CALCULATE VOLUME AND DEL O-18 IF CHINA LAKE IS DRY**
*********************************************************

                  X_CS(1)=0.D0
                  X_CS(2)=TCDELI
                  ZEROVOL_C=.FALSE.
                  ZEROCHK_C=.FALSE.
                  DO 1400 I=3,4
                     X_CS(I)=X_CS(I)+25.D0*RK1(I)/216.D0+
     +                    1408.D0*RK3(I)/2565.D0+
     +                    2197.D0*RK4(I)/4104.D0- RK5(I)/5.D0
1400              CONTINUE

               ELSE IF(ZEROVOL_S .AND. .NOT. ZEROVOL_C)THEN

***********************************************************
** CALCULATE VOLUME AND DEL O-18 IF SEARLES LAKE IS DRY**
***********************************************************

                  X_CS(3)=0.D0
                  X_CS(4)=PCDEL_OUT
                  ZEROVOL_S=.FALSE.
                  ZEROCHK_S=.FALSE.
                  DO 1500 I=1,2
```

```
                         X_CS(I)=X_CS(I)+25.D0*RK1(I)/216.D0+
     +                           1408.D0*RK3(I)/2565.D0+
     +                           2197.D0*RK4(I)/4104.D0- RK5(I)/5.D0
1500                  CONTINUE

                    ELSE

************************************************************
** CALCULATE VOLUME AND DEL O-18 IF NEITHER LAKE IS DRY **
** AND CHECK TO SEE IF CHINA AND SEARLES COALESCE       **
************************************************************

                      DO 1600 I=1,NCS
                         X_CS(I)=X_CS(I)+25.D0*RK1(I)/216.D0+
     +                           1408.D0*RK3(I)/2565.D0+
     +                           2197.D0*RK4(I)/4104.D0- RK5(I)/5.D0
1600                  CONTINUE

                      IF(X_CS(3) .GT. 65.87D9)THEN
                         COAL=.TRUE.
                         COUP=.TRUE.
                         X_CS(3)=X_CS(3)+0.696D9
                         CPCNT=0.696D9/X_CS(3)
                         SPCNT=1.D0-CPCNT
                         X_CS(4)=CPCNT*X_CS(2)+SPCNT*X_CS(4)
                         X_CS(1)=0.D0
                         X_CS(2)=X_CS(4)
                         WRITE(6,*)
                         WRITE(6,*)'CHINA AND SEARLES HAVE COALESCED'
                         WRITE(6,*)'AT',TIME
                         WRITE(6,*)
                      ENDIF

                    ENDIF

                    IF(X_CS(1) .LT. 0.D0)THEN
                       X_CS(1)=0.D0
                       X_CS(2)=TCDELI
                    ENDIF

                    IF(X_CS(3) .LT. 0.D0)THEN
                       X_CS(3)=0.D0
                       X_CS(4)=PCDEL_OUT
                    ENDIF

****************************************************
** SALT BALANCE STUFF FOR THE ENTIRE TIME-STEP **
****************************************************

*********************************************
** IF THE LAKES HAVE JUST BEEN DECOUPLED **
*********************************************
                    IF(DECOUP)THEN
                       ALL_CL=PCL_S +TTLCL_IN
                       ALL_VOL=X_CS(1)+X_CS(3)
                       ALL_CONC=ALL_CL/(ALL_VOL*1.D3)
** CHINA **
                       CONC_CL_C =ALL_CONC
                       IF(CONC_CL_C  .GT. 6.1D0)THEN
                          CONC_CL_C =6.1D0
                       ENDIF
                       CL_C =CONC_CL_C *X_CS(1)*1.D3
                       CCL_OUT=0.D0
** SEARLES **
                       CONC_CL_S =ALL_CONC
                       IF(CONC_CL_S  .GT. 6.1D0)THEN
```

```
                        CLDEP=CONC_CL_S -6.1D0
                        CONC_CL_S =6.1D0
                        SCL_DEP=CLDEP*X_CS(3)*35.453D0*
     +                       3.22D-8
                     ELSE
                        SCL_DEP=0.D0
                     ENDIF

                     SUM_SCL_DEP=PSUM_SCL_DEP +
     +                    SCL_DEP+SLTCORR
                     CL_S =X_CS(3)*1.D3*CONC_CL_S
                     DECOUP=.FALSE.

**********************************************
** IF CHINA AND SEARLES HAVE JUST COALESCED **
**********************************************

                  ELSE IF(COUP)THEN
                     CL_S =PCL_C +PCL_S +TTLCL_IN
                     CONC_CL_S =CL_S /(X_CS(3)*1.D3)
                     IF(CONC_CL_S  .GT. 6.1D0)THEN
                        CLDEP=CONC_CL_S -6.1D0
                        CONC_CL_S =6.1D0
                        SCL_DEP=CLDEP*X_CS(3)*35.453D0*
     +                       3.22D-8
                     ELSE
                        SCL_DEP=0.D0
                     ENDIF

                     SUM_SCL_DEP=PSUM_SCL_DEP +
     +                    SCL_DEP+SLTCORR

                     COUP=.FALSE.

*******************************************************************
** IF CHINA AND SEARLES ARE STILL COALESCED FROM THE LAST TIME STEP **
*******************************************************************

                  ELSE IF(COAL)THEN
                     IF(X_CS(3) .GT. VOLMAX_S)THEN
                        VOL_OUT_S=X_CS(3)-VOLMAX_S
                        CL_S =PCL_S +TTLCL_IN-
     +                       VOL_OUT_S*1.D3*PCONC_CL_S
                        IF(CL_S .LT.0.D0)CL_S =3.95D-4*
     +                       85.28D9*1.0D3
                        X_CS(3)=VOLMAX_S
                        CONC_CL_S =CL_S /(X_CS(3)*1.D3)
                        SCL_DEP=0.D0
                        SUM_SCL_DEP=PSUM_SCL_DEP +
     +                       SCL_DEP+SLTCORR
                     ELSE
                        CL_S =PCL_S +TTLCL_IN
                        CONC_CL_S =CL_S /(X_CS(3)*1.D3)
** SATURATION CHECK **
                        IF(CONC_CL_S  .GT. 6.1D0)THEN
                           CLDEP=CONC_CL_S -6.1D0
                           CONC_CL_S =6.1D0
                           CL_S =X_CS(3)*1.D3*CONC_CL_S
                           SCL_DEP=CLDEP*X_CS(3)*35.453D0*
     +                          3.22D-8
                        ELSE
                           SCL_DEP=0.D0
                        ENDIF

                        SUM_SCL_DEP=PSUM_SCL_DEP +
     +                       SCL_DEP+SLTCORR
```

```
                    ENDIF

**********************************************************************
** IF CHINA AND SEARLES AREN'T DOING ANY OF THE COUP/DECOUP STUFF **
**********************************************************************

                 ELSE IF(X_CS(1).GT.VOLMAX_C.AND.X_CS(3).GT.VOLMAX_S)
     +                   THEN
**************
** OVERFLOW **  ** CHINA **
**************


                    VOL_OUT_C=X_CS(1)-VOLMAX_C
                    CL_C =PCL_C +TTLCL_IN-
     +                   VOL_OUT_C*1.D3*PCONC_CL_C
                    IF(CL_C .LT.0.D0)CL_C =3.95D-4*
     +                   0.696D9*1.0D3
                    X_CS(1)=VOLMAX_C
                    CONC_CL_C =CL_C /(X_CS(1)*1.D3)
                    CCL_OUT= VOL_OUT_C*1.D3*PCONC_CL_C
**************
** OVERFLOW **  ** SEARLES **
**************

                    VOL_OUT_S=X_CS(3)-VOLMAX_S
                    CL_S =PCL_S +CCL_OUT-
     +                   VOL_OUT_S*1.D3*PCONC_CL_S
                    IF(CL_S .LT.0.D0)CL_S =3.95D-4*
     +                   85.28D9*1.0D3
                    X_CS(3)=VOLMAX_S
                    CONC_CL_S =CL_S /(X_CS(3)*1.D3)
                    SCL_DEP=0.D0
                    SUM_SCL_DEP=PSUM_SCL_DEP +
     +                   SCL_DEP+SLTCORR

************************************************************
** CHINA LAKE IS OVERFLOWING, SEARLES IS STILL FILLING **
************************************************************

                 ELSE IF(X_CS(1).GT.VOLMAX_C.AND.X_CS(3).GT.10.D0)
     +                   THEN
**************
** OVERFLOW **  ** CHINA **
**************
                    VOL_OUT_C=X_CS(1)-VOLMAX_C
                    CL_C =PCL_C +TTLCL_IN-
     +                   VOL_OUT_C*1.D3*PCONC_CL_C
                    IF(CL_C .LT.0.D0)CL_C =3.95D-4*
     +                   0.696D9*1.0D3
                    X_CS(1)=VOLMAX_C
                    CONC_CL_C =CL_C /(X_CS(1)*1.D3)
                    CCL_OUT= VOL_OUT_C*1.D3*PCONC_CL_C
*************
** FILLING ** ** SEARLES **
*************
                    CL_S =PCL_S +CCL_OUT
                    CONC_CL_S =CL_S /(X_CS(3)*1.D3)
** SATURATION CHECK **
                    IF(CONC_CL_S  .GT. 6.1D0)THEN
                       CLDEP=CONC_CL_S -6.1D0
                       CONC_CL_S =6.1D0
                       CL_S =X_CS(3)*1.D3*CONC_CL_S
                       SCL_DEP=CLDEP*X_CS(3)*35.453D0*
     +                      3.22D-8
```

```
                 ELSE
                    SCL_DEP=0.D0
                 ENDIF

                 SUM_SCL_DEP=PSUM_SCL_DEP +
     +                SCL_DEP+SLTCORR

****************************************************************
** CHINA LAKE IS BETWEEN OVERFLOW & DRY, SEARLES LAKE IS DRY **
****************************************************************

              ELSE IF(X_CS(1) .GT. 10.D0 .AND. X_CS(3) .LT. 10.D0)
     +                THEN

** CHINA **
                 CL_C =PCL_C +TTLCL_IN
                 CONC_CL_C =CL_C /(X_CS(1)*1.D3)
                 IF(CONC_CL_C .GT.6.1D0)THEN
                    CONC_CL_C =6.1D0
                    CL_C =6.1D0*X_CS(1)*1.D3
                 ENDIF
                 CCL_OUT=0.D0

** SEARLES **

                 CL_S =0.D0
                 X_CS(3)=0.D0
                 CONC_CL_S =0.D0
                 SCL_DEP=PCL_S *35.453D0/1.0D3*
     +                3.22D-8
                 SUM_SCL_DEP=PSUM_SCL_DEP +
     +                SCL_DEP+SLTCORR

*********************************************************************************
** LET'S ASSUME THAT IF CHINA LAKE IS DRY, SEARLES LAKE WON'T BE OVERFLOWING **
*********************************************************************************

*****************************************************
** CHINA LAKE IS DRY, SEARLES LAKE IS NOT YET DRY **
*****************************************************

              ELSE IF(X_CS(1) .LE. 10.D0 .AND. X_CS(3) .GT. 10.D0)
     +                THEN

**************
** DRY LAKE ** ** CHINA **
**************

                 CL_C =0.D0
                 X_CS(1)=0.D0
                 CONC_CL_C =0.D0
                 CCL_OUT=0.D0

*****************
** NOT YET DRY ** SEARLES **
*****************

                 CL_S =PCL_S
                 CONC_CL_S =CL_S /(X_CS(3)*1.D3)
** SATURATION CHECK **
                 IF(CONC_CL_S  .GT. 6.1D0)THEN
                    CLDEP=CONC_CL_S -6.1D0
                    CONC_CL_S =6.1D0
                    CL_S =X_CS(3)*1.D3*CONC_CL_S
                    SCL_DEP=CLDEP*X_CS(3)*35.453D0*
     +                   3.22D-8
```

```
                      ELSE
                         SCL_DEP=0.D0
                      ENDIF

                      SUM_SCL_DEP=PSUM_SCL_DEP +
     +                     SCL_DEP+SLTCORR

********************************************
** CHINA LAKE IS DRY, SEARLES LAKE IS DRY **
********************************************

                   ELSE IF(X_CS(1) .LE. 10.D0 .AND. X_CS(3) .LE. 10.D0)
     +                     THEN

**************
** DRY LAKE ** ** CHINA **
**************

                      CL_C =0.D0
                      X_CS(1)=0.D0
                      CONC_CL_C =0.D0
                      CCL_OUT=0.D0
**************
** DRY LAKE ** ** SEARLES **
**************
                      CL_S =0.D0
                      X_CS(1)=0.D0
                      CONC_CL_S =0.D0
                      SCL_DEP=PCL_S *35.453D0/1.0D3*
     +                    3.22D-8
                      SUM_SCL_DEP=PSUM_SCL_DEP +
     +                     SCL_DEP+SLTCORR

                   ELSE
*****************************************
** BOTH LAKES BETWEEN DRY AND OVERFLOW **
*****************************************

** CHINA **
                      CL_C =PCL_C +TTLCL_IN
                      CONC_CL_C =CL_C /(X_CS(1)*1.D3)
                      IF(CONC_CL_C .GT.6.1D0)THEN
                         CONC_CL_C =6.1D0
                         CL_C =6.1D0*X_CS(1)*1.D3
                      ENDIF
                      CCL_OUT=0.D0

** SEARLES **
                      CL_S =PCL_S
                      CONC_CL_S =CL_S /(X_CS(3)*1.D3)
**SATURATION CHECK**
                      IF(CONC_CL_S  .GT. 6.1D0)THEN
                         CLDEP=CONC_CL_S -6.1D0
                         CONC_CL_S =6.1D0
                         CL_S =X_CS(3)*1.D3*CONC_CL_S
                         SCL_DEP=CLDEP*X_CS(3)*35.453D0*
     +                       3.22D-8
                      ELSE
                         SCL_DEP=0.D0
                      ENDIF
                      SUM_SCL_DEP=PSUM_SCL_DEP +
     +                     SCL_DEP+SLTCORR
                   ENDIF

********************************************
** CALCULATE DEL DOLOMITE FROM DEL WATER **
```

```
********************************************

                    CDEL_OUT =X_CS(2)
                    DELDOL_C=FDDOL(DOLTEMP_C,X_CS(2))
                    DELDOL_S=FDDOL(DOLTEMP_S,X_CS(4))

****************************************************************************
** CALCULATE SURFACE AREAS AND SALT BALANCE FOR PANAMINT AND DEATH VALLEY **
****************************************************************************

                    IF(CPARAM)THEN
                       TPEVAP=EVAPC_P
                       TDEVAP=EVAPC_D
                    ELSE
                       CALL FINDT(NPTS,TIME,INDEX,FOUND,WTIME)

                       IF(FOUND)THEN
                          TPEVAP = PEVAP(INDEX)
                          TDEVAP = DEVAP(INDEX)
                       ELSE
                          CALL PDINTERP(TIME,INDEX,TPEVAP,TDEVAP)
                       ENDIF
                    ENDIF

                    AREA_P=PQI/TPEVAP

                    PQO=0.D0

                    IF(AREA_P .GT. AREAMAX_P)THEN
                       PQO=PQI-TPEVAP*AREAMAX_P
                       VOL_OUT_P=PQO*STEP
                       AREA_P=AREAMAX_P
                    ELSE IF(AREA_P .GE. 0.D0)THEN
                       PQO=0.D0
                    ENDIF

                    VOL_P=PVOL(AREA_P)

                    IF(VOL_P .GT. 0.D0)THEN

                       CL_P =PCL_P +VOL_OUT_S*1.D3*
     +                      PCONC_CL_S -VOL_OUT_P*1.D3*
     +                      PCONC_CL_P
                       CONC_CL_P =CL_P /(1.D3*VOL_P)
**********************
** SATURATION CHECK **
**********************
                       IF(CONC_CL_P  .GT. 6.1D0)THEN
                          CLDEP=CONC_CL_P -6.1D0
                          CONC_CL_P =6.1D0
                          PCL_DEP=2.D0*CLDEP*1.D3*VOL_P*35.453D0/1.D3*
     +                         3.22D-8
                          SUM_PCL_DEP =PSUM_PCL_DEP +
     +                         PCL_DEP
                       ELSE
                          SUM_PCL_DEP =PSUM_PCL_DEP
                       ENDIF
                    ELSE
                       CL_P =0.D0
                       CONC_CL_P=0.D0
                       PCL_DEP=PCL_P *2.D0*35.453/1.0D3*3.22D-8
                       SUM_PCL_DEP =PSUM_PCL_DEP +PCL_DEP
                    ENDIF
******************
** DEATH VALLEY **
******************
```

```
                  DQI=PQO
                  AREA_D=DQI/TDEVAP

                  IF(AREA_D .GT. AREAMAX_D)THEN
                     DQO=DQI-TDEVAP*AREAMAX_D
                     VOL_OUT_D=DQO*STEP
                     AREA_D=AREAMAX_D
                  ELSE IF(AREA_D .GT. 0.D0)THEN
                     DQO=0.D0
                  ENDIF

                  VOL_D=DVOL(AREA_D)

                  IF(VOL_D.GT. 0.D0)THEN

                  ENDIF

***************************************************
** WRITE PANAMINT/DEATH VALLEY STUFF TO FILE **
***************************************************

**                WRITE(83,*)TIME/1.D6,SUM_PCL_DEP

**********************************************
** CALCULATE SUMMATION OF ALL LAKE AREAS **
**********************************************

                  AREA_C=CAREA(X_CS(1))
                  AREA_S=SAREA(X_CS(3))

                  DELTA_O=SOAREA-OPREV
                  DELTA_C=AREA_C-CPREV
                  DELTA_S=AREA_S-SPREV
                  DELTA_P=AREA_P-PPREV
                  DELTA_D=AREA_D-DPREV


                  ALLAREA =DELTA_O+DELTA_C+DELTA_S+DELTA_P
     +                 +DELTA_D+PALLAREA

                  IF(ALLAREA  .LT. 0.D0)ALLAREA =0.D0

                  OPREV=SOAREA
                  CPREV=AREA_C
                  SPREV=AREA_S
                  PPREV=AREA_P
                  DPREV=AREA_D

******************************************
** UPDATE "PEAK AND VALLEY INDICATORS **
******************************************

                  SDP2=SDP1
                  SDP1=SDC
                  SDC=DELDOL_S
                  DIF1=DABS(SDC-SDP1)
                  DIF2=DABS(SDP1-SDP2)
                  DIFMAX=DMAX1(DIF1,DIF2)

************************************
** CHLORIDE SATURATION "WARNING" **
************************************

                  IF(DABS(CONC_CL_S -6.1D0).LT.1.0D-5)THEN
                     RTIME=TIME/1.D6
                     WRITE(6,*)
```

```
                  WRITE(6,'(A,2X,F8.5)')'SEARLES Cl SAT. AT'
     +                 ,RTIME
                  WRITE(6,*)
               ENDIF
***************************
** WRITE RESULTS TO FILE **
***************************

               IF((WRTM-TIME).GT. DATSAV)THEN
                  WRTM=TIME
                  WRITE(87,*)TIME/1.D6,CONC_CL_S
** SQO = PQI **
                  WRITE(91,*)TIME/1.D6,PQI
                  WRITE(77,*)TIME/1.D6,AREA_S/1.D9
                  WRITE(78,*)TIME/1.D6,DELDOL_S
                  WRITE(76,*)TIME/1.D6,QI_S
                  WRITE(74,*)TIME/1.D6,SUM_SCL_DEP
                  WRITE(81,*)TIME/1.D6,ALLAREA/1.D9
               ELSE IF(SDC .GT. SDP1 .AND. SDP2 .GT. SDP1)THEN
                  WRTM=TIME
                  WRITE(87,*)TIME/1.D6,CONC_CL_S
** SQO = PQI **
                  WRITE(91,*)TIME/1.D6,PQI
                  WRITE(77,*)TIME/1.D6,AREA_S/1.D9
                  WRITE(78,*)TIME/1.D6,DELDOL_S
                  WRITE(76,*)TIME/1.D6,QI_S
                  WRITE(74,*)TIME/1.D6,SUM_SCL_DEP
                  WRITE(81,*)TIME/1.D6,ALLAREA/1.D9
               ELSE IF(SDC .LT. SDP1 .AND. SDP2 .LT. SDP1)THEN
                  WRTM=TIME
                  WRITE(87,*)TIME/1.D6,CONC_CL_S
** SQO = PQI **
                  WRITE(91,*)TIME/1.D6,PQI
                  WRITE(77,*)TIME/1.D6,AREA_S/1.D9
                  WRITE(78,*)TIME/1.D6,DELDOL_S
                  WRITE(76,*)TIME/1.D6,QI_S
                  WRITE(74,*)TIME/1.D6,SUM_SCL_DEP
                  WRITE(81,*)TIME/1.D6,ALLAREA/1.D9
               ELSE IF(DABS(TIME-TEND).LT.1.D-5)THEN
                  WRITE(87,*)TIME/1.D6,CONC_CL_S
** SQO = PQI **
                  WRITE(91,*)TIME/1.D6,PQI
                  WRITE(77,*)TIME/1.D6,AREA_S/1.D9
                  WRITE(78,*)TIME/1.D6,DELDOL_S
                  WRITE(76,*)TIME/1.D6,QI_S
                  WRITE(74,*)TIME/1.D6,SUM_SCL_DEP
                  WRITE(81,*)TIME/1.D6,ALLAREA /1.D9
               ENDIF
               IF(SU)THEN
                  IF(TIME .LT. SAVETIME .AND. ISTCNT
     +                 .LE. NUMST)THEN
                     WRITE(70,*)'CHINA LAKE'
                     WRITE(70,*)'STARTUP TIME #',ISTCNT
                     WRITE(70,*)TIME,X_CS(1),DELDOL_C,
     +                    CONC_CL_C
                     WRITE(70,*)'SEARLES LAKE'
                     WRITE(70,*)TIME,X_CS(3),AREA_S,DELDOL_S,
     +                    CONC_CL_S ,SUM_SCL_DEP
                     WRITE(70,*)'PANAMINT LAKE'
                     WRITE(70,*)TIME,AREA_P,
     +                    CONC_CL_P ,SUM_PCL_DEP
                     WRITE(70,*)'DEATH VALLEY'
                     WRITE(70,*)TIME,AREA_D
                     WRITE(70,*)
                     ISTCNT=ISTCNT+1
                     SAVETIME=SUT(ISTCNT)
```

```
                 ENDIF
              ENDIF

              PALLAREA=ALLAREA
              PCL_P=CL_P
              PCONC_CL_P=CONC_CL_P
              PSUM_PCL_DEP=SUM_PCL_DEP
              PCL_C=CL_C
              PCONC_CL_C=CONC_CL_C
              PCDEL_OUT=CDEL_OUT
              PCL_S=CL_S
              PCONC_CL_S=CONC_CL_S
              PSUM_SCL_DEP=SUM_SCL_DEP

              STEP=STEP+STEP*0.5D0
              IF(STEP .GT. DTMAX_CS)STEP=DTMAX_CS
              IF(TIME-STEP.LT.TEND)THEN
                 DTMAX_CS=TIME-TEND
                 STEP=DTMAX_CS
              ENDIF

              GOTO 100

**********************************
** MAKE SURE THE SOLVER DOESN'T **
** OVERSTEP DESIGNATED END TIME **
**********************************

           ELSEIF(TIME-STEP.LT.TEND)THEN
              DTMAX_CS=TIME-TEND
              STEP=DTMAX_CS
              GOTO 100
           ENDIF

**************************************
** ADJUST THE SIZE OF THE TIME STEP **
**************************************

        ELSEIF(DELMIN .LE. 0.1)THEN
           STEP=STEP*1.0D-1
        ELSEIF(DELMIN .GE. 4.D0)THEN
           STEP=4.D0*STEP
        ELSE
           STEP=DELMIN*STEP
        ENDIF
        IF(STEP.GT.DTMAX_CS)STEP=DTMAX_CS
        IF(STEP.LT.DTMIN_CS)STEP=DTMIN_CS
     ELSE

        CLOSE(UNIT=92)
        CLOSE(UNIT=91)
        CLOSE(UNIT=87)
        CLOSE(UNIT=85)
        CLOSE(UNIT=84)
        CLOSE(UNIT=83)
        CLOSE(UNIT=81)
        CLOSE(UNIT=78)
        CLOSE(UNIT=77)
        CLOSE(UNIT=76)
        CLOSE(UNIT=75)
        CLOSE(UNIT=74)

        IF( .NOT. GRAF)THEN
           WRITE(6,*)
           WRITE(6,*)'THE RKF SOLVED',KOUNT,' POINTS IN',ITER,' IT
    +ERATIONS'
```

```
          WRITE(6,*)'FOR CHINA AND SEARLES'
          WRITE(6,*)
          WRITE(6,*)'FINAL DEL FOR OWENS =',FINDEL
          WRITE(6,*)'FINAL VOLUME FOR OWENS =',FINVOL/1.D9
          WRITE(6,*)'FINAL CL CONC, OWENS =',FINCL
          WRITE(6,*)
          WRITE(6,*)'FINAL DEL FOR CHINA =',DELDOL_C
          WRITE(6,*)'FINAL VOLUME FOR CHINA =',X_CS(1)/1.D9
          WRITE(6,*)'FINAL CL CONC, CHINA =',CONC_CL_C
          WRITE(6,*)
          WRITE(6,*)'FINAL DEL FOR SEARLES =',DELDOL_S
          WRITE(6,*)'FINAL VOLUME FOR SEARLES =',X_CS(3)/1.D9
          WRITE(6,*)'FINAL AREA FOR SEARLES =',AREA_S/1.D9
          WRITE(6,*)'FINAL CL CONC, SEARLES =',CONC_CL_S
          WRITE(6,*)'TOTAL CL DEPOSITED IN SEARLES',
     +         SUM_SCL_DEP
          WRITE(6,*)
          WRITE(6,*)'FINAL AREA OF PANAMINT',AREA_P/1.D9
          WRITE(6,*)'FINAL CL CONC, PANAMINT',CONC_CL_P
          WRITE(6,*)'TOTAL CL DEPOSITED IN PANAMINT',
     +         SUM_PCL_DEP
          WRITE(6,*)'FINAL AREA OF LAKE MANLY',AREA_D/1.D9
          WRITE(6,*)'FINAL TIME IS:',TIME

          WRITE(55,*)
          WRITE(55,*)'THE RKF SOLVED',KOUNT,' POINTS IN',ITER,' I
     +TERATIONS'
          WRITE(55,*)'FOR CHINA AND SEARLES'
          WRITE(55,*)
          WRITE(55,*)'FINAL DEL FOR OWENS =',FINDEL
          WRITE(55,*)'FINAL VOLUME FOR OWENS =',FINVOL/1.D9
          WRITE(55,*)'FINAL CL CONC, OWENS =',FINCL
          WRITE(55,*)
          WRITE(55,*)'FINAL DEL FOR CHINA =',DELDOL_C
          WRITE(55,*)'FINAL VOLUME FOR CHINA =',X_CS(1)/1.D9
          WRITE(55,*)'FINAL CL CONC, CHINA =',CONC_CL_C
          WRITE(55,*)
          WRITE(55,*)'FINAL DEL FOR SEARLES =',DELDOL_S
          WRITE(55,*)'FINAL VOLUME FOR SEARLES =',X_CS(3)/1.D9
          WRITE(55,*)'FINAL AREA FOR SEARLES =',AREA_S/1.D9
          WRITE(55,*)'FINAL CL CONC, SEARLES =',CONC_CL_S
          WRITE(55,*)'TOTAL CL DEPOSITED IN SEARLES',
     +         SUM_SCL_DEP
          WRITE(55,*)
          WRITE(55,*)'FINAL AREA OF PANAMINT',AREA_P/1.D9
          WRITE(55,*)'FINAL CL CONC, PANAMINT',CONC_CL_P
          WRITE(55,*)'TOTAL CL DEPOSITED IN PANAMINT',
     +         SUM_PCL_DEP
          WRITE(55,*)'FINAL AREA OF LAKE MANLY',AREA_D/1.D9
          WRITE(55,*)'FINAL TIME IS:',TIME

*********************************************************
** WRITE NUMBERS TO FILE TO UPDATE STARTING PARAMETERS **
*********************************************************

          WRITE(56,*)FINDEL,FINVOL,FINCL,DELDOL_C,X_CS(1),
     +    CONC_CL_C,DELDOL_S,X_CS(3),CONC_CL_S,AREA_P,
     +    CONC_CL_P,SUM_PCL_DEP,AREA_D
          WRITE(56,*)
          WRITE(55,*)'TIME IS:',TIME
        ENDIF

        RETURN
      ENDIF

100   ENDWHILE
```

```
**      IF(ITER .GT. ITMAX_CS)WRITE(6,*)'MAX # OF ITERATIONS EXCEEDED'
        IF( .NOT. GRAF)THEN
           WRITE(6,*)
           WRITE(6,*)'THE RKF SOLVED',KOUNT,' POINTS IN',ITER,' IT
     +ERATIONS'
           WRITE(6,*)'FOR CHINA AND SEARLES'
           WRITE(6,*)
           WRITE(6,*)'FINAL DEL FOR OWENS =',FINDEL
           WRITE(6,*)'FINAL VOLUME FOR OWENS =',FINVOL/1.D9
           WRITE(6,*)'FINAL CL CONC, OWENS =',FINCL
           WRITE(6,*)
           WRITE(6,*)'FINAL DEL FOR CHINA =',DELDOL_C
           WRITE(6,*)'FINAL VOLUME FOR CHINA =',X_CS(1)/1.D9
           WRITE(6,*)'FINAL CL CONC, CHINA =',CONC_CL_C
           WRITE(6,*)
           WRITE(6,*)'FINAL DEL FOR SEARLES =',DELDOL_S
           WRITE(6,*)'FINAL VOLUME FOR SEARLES =',X_CS(3)/1.D9
           WRITE(6,*)'FINAL AREA FOR SEARLES =',AREA_S/1.D9
           WRITE(6,*)'FINAL CL CONC, SEARLES =',CONC_CL_S
           WRITE(6,*)'TOTAL CL DEPOSITED IN SEARLES',
     +          SUM_SCL_DEP
           WRITE(6,*)
           WRITE(6,*)'FINAL AREA OF PANAMINT',AREA_P/1.D9
           WRITE(6,*)'FINAL CL CONC, PANAMINT',CONC_CL_P
           WRITE(6,*)'TOTAL CL DEPOSITED IN PANAMINT',
     +          SUM_PCL_DEP
           WRITE(6,*)'FINAL AREA OF LAKE MANLY',AREA_D/1.D9
           WRITE(6,*)'FINAL TIME IS:',TIME

           WRITE(55,*)
           WRITE(55,*)'THE RKF SOLVED',KOUNT,' POINTS IN',ITER,' ITERATI
     +ONS'
           WRITE(55,*)'FOR CHINA AND SEARLES'
           WRITE(55,*)
           WRITE(55,*)'FINAL DEL FOR OWENS =',FINDEL
           WRITE(55,*)'FINAL VOLUME FOR OWENS =',FINVOL/1.D9
           WRITE(55,*)'FINAL CL CONC, OWENS =',FINCL
           WRITE(55,*)
           WRITE(55,*)'FINAL DEL FOR CHINA =',DELDOL_C
           WRITE(55,*)'FINAL VOLUME FOR CHINA =',X_CS(1)/1.D9
           WRITE(55,*)'FINAL CL CONC, CHINA =',CONC_CL_C
           WRITE(55,*)
           WRITE(55,*)'FINAL DEL FOR SEARLES =',DELDOL_S
           WRITE(55,*)'FINAL VOLUME FOR SEARLES =',X_CS(3)/1.D9
           WRITE(55,*)'FINAL AREA FOR SEARLES =',AREA_S/1.D9
           WRITE(55,*)'FINAL CL CONC, SEARLES =',CONC_CL_S
           WRITE(55,*)'TOTAL CL DEPOSITED IN SEARLES',
     +                SUM_SCL_DEP
           WRITE(55,*)
           WRITE(55,*)'FINAL AREA OF PANAMINT',AREA_P/1.D9
           WRITE(55,*)'FINAL CL CONC, PANAMINT',CONC_CL_P
           WRITE(55,*)'TOTAL CL DEPOSITED IN PANAMINT',
     +                SUM_PCL_DEP
           WRITE(55,*)'FINAL AREA OF LAKE MANLY',AREA_D/1.D9
           WRITE(55,*)'FINAL TIME IS:',TIME

**********************************************************
** WRITE NUMBERS TO FILE TO UPDATE STARTING PARAMETERS **
**********************************************************

           WRITE(56,*)FINDEL,FINVOL,FINCL,DELDOL_C,X_CS(1),
     +           CONC_CL_C,DELDOL_S,X_CS(3),CONC_CL_S,AREA_P,
     +           CONC_CL_P,SUM_PCL_DEP,AREA_D
           WRITE(56,*)
           WRITE(55,*)'TIME IS:',TIME
        ENDIF
```

```
      END


***************************************************************************
***************************************************************************
** A FUNCTION TO CALCULATE DV/DT AND DDEL/DT FOR CHINA AND SEARLES LAKE **
***************************************************************************
***************************************************************************

      DOUBLE PRECISION FUNCTION F_CS(I,TERM,T,KNT,KOUNT,TBEG,COAL)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      LOGICAL FOUND,GUESS,GRAF,CPARAM,GUESS2,FOUND2,COAL,SU

      PARAMETER(NUMA=25000,NUMB=2000)
      PARAMETER(VOLMAX_C=0.696D9,VOLMAX_S=85.28D9)

      SAVE

      COMMON/BOTH/CL(NUMA),TSOD(10),TCL(10),OTIME(NUMA),TCO3(10),
     + OQO(NUMA),QO(10),CONC_CL(NUMA),DOLTEMP,DELDOL,TODELI,TOTEMP,
     + ODEL_OUT(NUMA),OCL_OUT(NUMA),PEVAP(NUMB),DEVAP(NUMB),
     + SUM_PCL_DEP,AREA_P,AREA_D,
     + AREA(NUMA),SOAREA,ALLAREA,CONC_CL_P,
     + CL_P

      COMMON/BOTH2/CL_C,TSOD_C(10),TCL_C(10),TIME,TCO3_C(10),
     + CQO(10),GRAF,CONC_CL_C,DOLTEMP_C,DELDOL_C,
     + TCDELI,TCTEMP,CDEL_OUT,CCL_OUT,CL_S,NOPTS,
     + TSOD_S(10),TCL_S(10),TCO3_S(10),SQO(10),CONC_CL_S,
     + DOLTEMP_S,DELDOL_S,TSDELI,TSTEMP,SCL_DEP,TTLCL_IN,NPTS,
     + SUM_SCL_DEP,QI_C,QI_S,PQI

      COMMON/PREV/PCL_P,PCONC_CL_P,PSUM_PCL_DEP,PCL_C,PCONC_CL_C,
     + PCDEL_OUT,PCL_S,PCONC_CL_S,PSUM_SCL_DEP,PALLAREA,DATSAV,
     + SDC,SDP1,SDP2

      COMMON/FIRST/WTIME(NUMB),OTEMP(NUMB),OEVAP(NUMB),
     + OHUM(500),OPRECIP(NUMB),ODELP(NUMB),ODELA(NUMB),ODELI(NUMB),
     + GUESS,TEMPC,EVAPC,PRECIPC,DELAC,DELIC,CPARAM,DELPC,
     + EVAPC_P,EVAPC_D

      COMMON/SECOND/CTEMP(NUMB),CEVAP(NUMB),CHUM(500),CPRECIP(NUMB),
     + CDELP(NUMB),CSDELA(NUMB),GUESS2,TEMPC_C,EVAPC_C,
     + PRECIPC_C,DELAC_C,DELPC_C,STEMP(NUMB),SEVAP(NUMB),SHUM(500),
     + SPRECIP(NUMB),SDELP(NUMB),TEMPC_S,EVAPC_S,
     + PRECIPC_S,DELAC_S,DELPC_S

      COMMON/NEW/OPREV,CPREV,SPREV,PPREV,DPREV

      COMMON/HIST/QIHIST(1000),HUMTIME(500),NHPTS,SU,SUT(15),
     + SAVETIME,NUMST,QITIME(1000),NQPTS

      DIMENSION TERM(4),TCONC_CL_C(10),V_OUT_C(10),
     + TCONC_CL_S(10),V_OUT_S(10)

      IF(KOUNT .EQ. 1)THEN
         ALLAREA=PALLAREA
         CL_C=PCL_C
         CONC_CL_C=PCONC_CL_C
         CDEL_OUT=PCDEL_OUT
         CL_S=PCL_S
         CONC_CL_S=PCONC_CL_S
         SUM_SCL_DEP=PSUM_SCL_DEP
```

```
      ENDIF

      IF(I.EQ.1)THEN

************************************
** CALCULATE DV/DT FOR CHINA LAKE **
************************************

         VOL_C = TERM(1)
         IF(VOL_C .LE. 10.D0)VOL_C=0.D0


*******************************
** CONSTANT PARAMETER OPTION **
*******************************

         IF(CPARAM)THEN
            TCTEMP=TEMPC_C
            TCEVAP=EVAPC_C
            TCPRECIP=PRECIPC_C
            TCDELP=DELPC_C
            TCDELA=DELAC_C
            CALL FINDHT(NHPTS,T,IHNDEX,HUMTIME)
            CALL HUMTERP(T,IHNDEX,TCHUM,CHUM,HUMTIME)
         ELSE

**********************************************************************
** DETERMINE VALUES OF NECESSARY PARAMETERS BY ASSIGNING VALUES OR  **
** INTERPOLATING BETWEEN GIVEN VALUES                               **
**********************************************************************

            CALL FINDT(NPTS,T,INDEX,FOUND,WTIME)
            CALL FINDHT(NHPTS,T,IHNDEX,HUMTIME)


            IF(FOUND)THEN
               TCTEMP = CTEMP(INDEX)
               TCEVAP = CEVAP(INDEX)
               TCPRECIP = CPRECIP(INDEX)
               TCDELP = CDELP(INDEX)
               TCDELA = CSDELA(INDEX)
            ELSE
               CALL CINTERP(T,INDEX,TCTEMP,TCEVAP,TCPRECIP,
     +                  TCDELP,TCDELA)
            ENDIF
            CALL HUMTERP(T,IHNDEX,TCHUM,CHUM,HUMTIME)
         ENDIF

*******************************************************************************
** INTERPOLATE TO FIND VALUES FOR PARAMETERS CALCULATED DURING OWENS ROUTINE **
*******************************************************************************

         CALL FINDT2(NOPTS,T,INDEX2,FOUND2,OTIME)
         IF(FOUND2)THEN
            QI=OQO(INDEX2)
            TCDELI=ODEL_OUT(INDEX2)
            TAREA=AREA(INDEX2)
         ELSE
            CALL C2INTERP(T,INDEX2,TCDELI,QI,TAREA)
         ENDIF

         IF(QI .LT. 0.D0)QI=0.D0

*****************************************************************
** "REMEMBER" TEMP AT END OF TIMESTEP TO CALCULATE DEL DOLOMITE **
*****************************************************************
```

```
          IF(KNT .EQ. 5)DOLTEMP_C=TCTEMP

          IF(KNT .EQ. 5)SOAREA=TAREA

*********************************************************
** "REMEMBER" QI AT END OF TIMESTEP FOR GRAPHICS ARRAY **
*********************************************************

          IF(KNT .EQ. 5)QI_C=QI

          ETIME = TBEG-T

*****************************
** CALCULATE AREA OF LAKE **
*****************************

          AREA_C = CAREA(VOL_C)

******************************
** CALCULATE PRECIPITATION **
******************************

          QP = AREA_C*TCPRECIP

**********************************
** THE INFAMOUS "SALT" BALANCE **
**********************************

          IF(DABS(T-TIME).LT.1.D-8)THEN
             CALL C3INTERP(T,INDEX2,TIMECL_IN)
             TCL_C(KNT)=CL_C
             TCONC_CL_C(KNT)=CONC_CL_C

***************************************************
** CALCULATE TOTAL OUTFLOW VOLUME FROM TIME TO T **
** AND ADJUST IF VOL EXCEEDS VOLMAX             **
***************************************************

          ELSE
             IF(VOL_C .LE. VOLMAX_C)THEN
                V_OUT_C(KNT)=0.D0
             ELSE
                V_OUT_C(KNT) = VOL_C-VOLMAX_C
                VOL_C=VOLMAX_C
             ENDIF

*********************************************************
**  CALCULATE CONCENTRATIONS FOR INTERMEDIATE TIME "T" **
*********************************************************

*******************************************************
** CALCULATE HOW MUCH 'SALT' CAME IN FROM OWENS LAKE **
*******************************************************

             CALL C3INTERP(T,INDEX2,CL_IN)
             TCL_IN=CL_IN-TIMECL_IN
             IF(KNT .EQ. 5)TTLCL_IN=TCL_IN
             TCL_C(KNT) = TCL_IN+CL_C -V_OUT_C(KNT)*
     +            CONC_CL_C *1.D3
             IF(TCL_C(KNT).LT.0.D0)TCL_C(KNT)=3.95D-4*0.696D9*1.D3

*******************************************************************
** CONCENTRATION UNITS ARE MOLARITY SO VOL MUST BE MULT BY 1000 TO **
** CONVERT M^3 TO LITERS **
*******************************************************************
```

```
          IF(VOL_C .GT. 10.D0)THEN
             TCONC_CL_C(KNT)=TCL_C(KNT)/(VOL_C*1.0D3)
          ELSE
             TCONC_CL_C(KNT)=0.D0
             TCL_C(KNT)=0.D0
          ENDIF

***********************************
** CHECK FOR CHLORIDE SATURATION **
***********************************

          IF(TCONC_CL_C(KNT) .GT. 6.1D0)THEN
             TCONC_CL_C(KNT)=6.1D0
             TCL_C(KNT)=6.1D0*(VOL_C*1.D3)
          ENDIF

************************************************************
** CALCULATE HOW MUCH SALT GOES TO SEARLES FROM TIME TO T **
************************************************************


          TCCL_OUT=V_OUT_C(KNT)*1.D3*CONC_CL_C

       ENDIF

********************************************
**  CALCULATE BACK-CONDENSATION FLUX (QC) **
********************************************

       IF(VOL_C .GT. 10.D0 .AND. TCONC_CL_C(KNT) .GT. 0.D0)THEN
          PHI=FPHI(TCONC_CL_C(KNT),SUM)
       ELSE
          PHI=0.D0
       ENDIF
       AW=DEXP(-18.D0*PHI*SUM*0.5D0/1.D3)

********************************
** CALCULATE EVAPORATION (QE) **
********************************

       QE = AREA_C*TCEVAP*AW

       IF(VOL_C .GT. 10.D0)THEN
          QC=(TCHUM*QE)/AW
       ELSE
          QC=0.D0
       ENDIF

*************************************
**  CALCULATE DV/DT IF VOL IS ZERO **
*************************************

       IF(TERM(1) .LE. 10.D0)THEN
          IF(GUESS)THEN
             DELDOL_C=FDDOL(TCTEMP,TERM(2))
             GUESS=.FALSE.
          ENDIF

          CQO(KNT)=0.D0

          F_CS=QI

          IF(F_CS .LE. 0.D0)THEN
             DV_DT=0.D0
          ELSE
```

```
              DV_DT=F_CS
           ENDIF

**************************************************
**  CALCULATE DV/DT IF VOL IS < VOLMAX BUT > 0 **
**************************************************

         ELSE IF(TERM(1)  .LT. VOLMAX_C)THEN

**********************
**  CALCULATE DV/DT **
**********************

           F_CS=QI+QC-QE+QP
           DV_DT=F_CS
           CQO(KNT)=0.D0

*****************************************************
**  CALCULATE DV/DT IF VOL IS GREATER THAN VOLMAX **
*****************************************************

         ELSE

******************
** CALCULATE QO **
******************

          CQO(KNT)=QI+QC-QE+QP

*********************
** CALCULATE DV/DT **
*********************

           IF(CQO(KNT) .LT. 0.D0)THEN
              CQO(KNT)=0.D0
              F_CS=QI+QC-QE+QP
              DV_DT=F_CS
           ELSE
              F_CS=CQO(KNT)
              DV_DT=0.D0
           ENDIF
         ENDIF

***************************************
** CALCULATE DDEL/DT FOR CHINA LAKE **
***************************************

      ELSE IF(I.EQ.2)THEN

         DELL_C = TERM(2)

         IF(VOL_C .LE. 10.D0)THEN
           F_CS=0.D0

         ELSE

*******************************************
** CALCULATE ISOTOPIC ENRICHMENT FACTOR **
*******************************************

           EPS = FEPS(TCTEMP)

*********************************************
** CALCULATE DEL OF THE BACK-CONDENSATION **
*********************************************
```

```
            CDELC =EPS*(1.D0+(TCDELA/1.D3))+TCDELA

**************************************
** CALCULATE DEL OF THE EVAPORATION **
**************************************

            CDELE = DELE(DELL_C,EPS,TCHUM)

*****************************************************
** SET DEL OF THE OUTFLOW EQUAL TO DEL OF THE LAKE **
*****************************************************

            CDELO = DELL_C

            F_CS=(QI*TCDELI+QC*CDELC+QP*TCDELP-CQO(KNT)*CDELO-QE*
     +           CDELE-DELL_C*DV_DT)/VOL_C

         ENDIF

      ELSE IF(I.EQ.3)THEN

**************************************
** CALCULATE DV/DT FOR SEARLES LAKE **
**************************************

         VOL_S = TERM(3)
         IF(VOL_S .LE. 10.D0)VOL_S=0.D0

*******************************
** CONSTANT PARAMETER OPTION **
*******************************

         IF(CPARAM)THEN
            TSTEMP=TEMPC_S
            TSEVAP=EVAPC_S
            TSPRECIP=PRECIPC_S
            TSDELP=DELPC_S
            TSDELA=DELAC_S
            IF(COAL)THEN
               CALL FINDT(NPTS,T,INDEX,FOUND,WTIME)
               CALL FINDHT(NHPTS,T,IHNDEX,HUMTIME)
            ENDIF
            CALL HUMTERP(T,IHNDEX,TSHUM,SHUM,HUMTIME)
         ELSE

**********************************************************************
** DETERMINE VALUES OF NECESSARY PARAMETERS BY ASSIGNING VALUES OR  **
** INTERPOLATING BETWEEN GIVEN VALUES                               **
**********************************************************************

            IF(COAL)THEN
               CALL FINDT(NPTS,T,INDEX,FOUND,WTIME)
               CALL FINDHT(NHPTS,T,IHNDEX,HUMTIME)
            ENDIF

            IF(FOUND)THEN
               TSTEMP = STEMP(INDEX)
               TSEVAP = SEVAP(INDEX)
               TSPRECIP = SPRECIP(INDEX)
               TSDELP = SDELP(INDEX)
            ELSE

*******************************************************************************
** INTERPOLATE TO FIND VALUES FOR PARAMETERS CALCULATED DURING OWENS ROUTINE **
*******************************************************************************
```

```
            CALL SINTERP(T,INDEX,TSTEMP,TSEVAP,TSPRECIP,
     +                TSDELP,TSDELA)

          ENDIF
          CALL HUMTERP(T,IHNDEX,TSHUM,SHUM,HUMTIME)
        ENDIF

*********************************************************************
** THESE SEARLES PARAMETERS ARE EQUAL TO THEIR CHINA COUNTERPARTS **
*********************************************************************

        IF(COAL)THEN
          CALL FINDT2(NOPTS,T,INDEX2,FOUND2,OTIME)
          IF(FOUND2)THEN
            QI=OQO(INDEX2)
            TSDELI=ODEL_OUT(INDEX2)
            TAREA=AREA(INDEX2)
          ELSE
            CALL C2INTERP(T,INDEX2,TSDELI,QI,TAREA)
          ENDIF
          IF(QI .LT. 0.D0)QI=0.D0
        ELSE
          TSDELI = CDELO
        ENDIF

*******************************************************************
** "REMEMBER" TEMP AT END OF TIMESTEP TO CALCULATE DEL DOLOMITE **
*******************************************************************

        IF(KNT .EQ. 5)DOLTEMP_S=TSTEMP

        IF(KNT .EQ. 5)SOAREA=TAREA

        ETIME = TBEG-T

****************************
** CALCULATE AREA OF LAKE **
****************************

        AREA_S = SAREA(VOL_S)

*****************************
** CALCULATE PRECIPITATION **
*****************************

        QP = AREA_S*TSPRECIP

************************
** QI = QO FROM CHINA **
************************

        IF(.NOT. COAL)THEN
          QI =CQO(KNT)
        ENDIF

        IF(KNT .EQ. 5)QI_S=QI

*********************************
** THE INFAMOUS "SALT" BALANCE **
*********************************

        IF(DABS(T-TIME).LT.1.D-8)THEN

**************************************************************
** IF CHINA AND SEARLES COALESCE, THE INITIAL SALT BALANCE **
** IS CALCULATED IN THE RKF_CS SOLVER                       **
```

```
*************************************************************

             TCL_S(KNT)=CL_S
             TCONC_CL_S(KNT)=CONC_CL_S

***************************************************
** CALCULATE TOTAL OUTFLOW VOLUME FROM TIME TO T **
** AND ADJUST IF VOL EXCEEDS VOLMAX              **
***************************************************

          ELSE
             IF(VOL_S .LE. VOLMAX_S)THEN
                V_OUT_S(KNT)=0.D0
             ELSE
                V_OUT_S(KNT) = VOL_S-VOLMAX_S
                VOL_S=VOLMAX_S
             ENDIF

*********************************************************
**  CALCULATE CONCENTRATIONS FOR INTERMEDIATE TIME "T" **
*********************************************************

*******************************************************
** CALCULATE HOW MUCH 'SALT' CAME IN FROM CHINA LAKE **
*******************************************************

             IF(COAL)THEN
                CALL C3INTERP(TIME,INDEX2,TIMECL_IN)
                CALL C3INTERP(T,INDEX2,CL_IN)
                TCL_IN=CL_IN-TIMECL_IN
                IF(KNT .EQ. 5)TTLCL_IN=TCL_IN
             ELSE
                TCL_IN=TCCL_OUT
             ENDIF

             TCL_S(KNT) = TCL_IN+CL_S -V_OUT_S(KNT)*
     +             CONC_CL_S *1.D3

             IF(TCL_S(KNT).LT.0.D0)TCL_S(KNT)=3.95D-4*85.28D9*1.D3

*********************************************************************
** CONCENTRATION UNITS ARE MOLARITY SO VOL MUST BE MULT BY 1000 TO **
** CONVERT M^3 TO LITERS **
*********************************************************************

             IF(VOL_S .GT. 10.D0)THEN
                TCONC_CL_S(KNT)=TCL_S(KNT)/(VOL_S*1.0D3)
             ELSE
                TCONC_CL_S(KNT)=0.D0
             ENDIF

***********************************
** CHECK FOR CHLORIDE SATURATION **
***********************************

             IF(TCONC_CL_S(KNT) .GT. 6.1D0)THEN
                TCONC_CL_S(KNT)=6.1D0
                TCL_S(KNT)=6.1D0*(VOL_S*1.D3)
             ENDIF

          ENDIF

*********************************************
**  CALCULATE BACK-CONDENSATION FLUX (QC) **
*********************************************
```

```
          IF(VOL_S .GT. 10.D0 .AND. TCONC_CL_S(KNT) .GT. 0.D0)THEN
             PHI=FPHI(TCONC_CL_S(KNT),SUM)
          ELSE
             PHI=0.D0
          ENDIF
          AW=DEXP(-18.D0*PHI*SUM*0.5D0/1.D3)

********************************
** CALCULATE EVAPORATION (QE) **
********************************

          QE = AREA_S*TSEVAP*AW

          IF(VOL_S .GT. 10.D0)THEN
             QC=(TSHUM*QE)/AW
          ELSE
             QC=0.D0
          ENDIF

*************************************
**  CALCULATE DV/DT IF VOL IS ZERO **
*************************************

          IF(TERM(3) .LE. 10.D0)THEN
             IF(GUESS2)THEN
                WRITE(77,*)TBEG/1.D6,AREA_S/1.D9
                WRITE(76,*)TBEG/1.D6,QI
                WRITE(91,*)TBEG/1.D6,0.0
                DELDOL_S=FDDOL(TSTEMP,TERM(4))
                WRITE(78,*)TBEG/1.D6,DELDOL_S
                GUESS2=.FALSE.
                PALLAREA=AREA(1)+AREA_S+AREA_C+AREA_P+AREA_D
                OPREV=AREA(1)
                CPREV=AREA_C
                SPREV=AREA_S
                PPREV=AREA_P
                DPREV=AREA_P
                WRITE(81,*)TBEG/1.D6,PALLAREA/1.D9
             ENDIF

             SQO(KNT)=0.D0

**************************************************
** SET SEARLES OUTFLUX EQUAL TO PANAMINT INFLUX **
**************************************************

             IF(KNT .EQ. 5)THEN
                PQI=SQO(KNT)
             ENDIF

             IF(QI .LT. 0.D0)QI=0.D0

             F_CS=QI

             IF(F_CS .LE. 0.D0)THEN
                DV_DT=0.D0
             ELSE
                DV_DT=F_CS
             ENDIF

*************************************************
**  CALCULATE DV/DT IF VOL IS < VOLMAX BUT > 0 **
*************************************************

          ELSE IF(TERM(3)  .LT. VOLMAX_S)THEN
```

```
***********************************
** WRITE INITIAL VALUES TO FILE **
***********************************

              IF(GUESS2)THEN
                 WRITE(77,*)TBEG/1.D6,AREA_S/1.D9
                 WRITE(76,*)TBEG/1.D6,QI
                 WRITE(91,*)TBEG/1.D6,0.0
                 DELDOL_S=FDDOL(TSTEMP,TERM(4))
                 WRITE(78,*)TBEG/1.D6,DELDOL_S
                 PALLAREA=AREA(1)+AREA_S+AREA_C+AREA_P+AREA_D
                 OPREV=AREA(1)
                 CPREV=AREA_C
                 SPREV=AREA_S
                 PPREV=AREA_P
                 DPREV=AREA_D
                 GUESS2=.FALSE.
                 WRITE(81,*)TBEG/1.D6,PALLAREA/1.D9
              ENDIF


**********************
**  CALCULATE DV/DT **
**********************

              F_CS=QI+QC-QE+QP
              DV_DT=F_CS
              SQO(KNT)=0.D0
              IF(KNT .EQ. 5)THEN
                 PQI=SQO(KNT)
              ENDIF


****************************************************
**  CALCULATE DV/DT IF VOL IS GREATER THAN VOLMAX **
****************************************************

           ELSE

              IF(GUESS2)THEN
                 WRITE(77,*)TBEG/1.D6,AREA_S/1.D9
                 WRITE(76,*)TBEG/1.D6,QI
                 WRITE(91,*)TBEG/1.D6,QI+QC-QE+QP
                 DELDOL_S=FDDOL(TSTEMP,TERM(4))
                 WRITE(78,*)TBEG/1.D6,DELDOL_S
                 PALLAREA=AREA(1)+AREA_S+AREA_C+AREA_P+AREA_D
                 GUESS2=.FALSE.
                 OPREV=AREA(1)
                 CPREV=AREA_C
                 SPREV=AREA_S
                 PPREV=AREA_P
                 DPREV=AREA_D
                 WRITE(81,*)TBEG/1.D6,PALLAREA/1.D9
              ENDIF


******************
** CALCULATE QO **
******************

             SQO(KNT)=QI+QC-QE+QP

*********************
** CALCULATE DV/DT **
*********************
```

```
            IF(SQO(KNT) .LT. 0.D0)THEN
               SQO(KNT)=0.D0
               IF(KNT .EQ. 5)PQI=SQO(KNT)
               F_CS=QI+QC-QE+QP
               DV_DT=F_CS
            ELSE
               F_CS=SQO(KNT)
               IF(KNT .EQ. 5)THEN
                  PQI=SQO(KNT)
               ENDIF
               DV_DT=0.D0
            ENDIF
         ENDIF

**************************************
** CALCULATE DDEL/DT FOR SEARLES LAKE **
**************************************

      ELSE IF(I.EQ.4)THEN

         DELL_S = TERM(4)

         IF(VOL_S .LE. 10.D0)THEN
            F_CS=0.D0

         ELSE

******************************************
** CALCULATE ISOTOPIC ENRICHMENT FACTOR **
******************************************

            EPS = FEPS(TSTEMP)

********************************************
** CALCULATE DEL OF THE BACK-CONDENSATION **
********************************************

            SDELC =EPS*(1.D0+(TSDELA/1.D3))+TSDELA

**************************************
** CALCULATE DEL OF THE EVAPORATION **
**************************************

            SDELE = DELE(DELL_S,EPS,TSHUM)

*****************************************************
** SET DEL OF THE OUTFLOW EQUAL TO DEL OF THE LAKE **
*****************************************************

            SDELO = DELL_S

            F_CS=(QI*TSDELI+QC*SDELC+QP*TSDELP-SQO(KNT)*SDELO-QE*
     +            SDELE-DELL_S*DV_DT)/VOL_S

         ENDIF

      END IF
      RETURN
      END


******************************************************************************
******************************************************************************
**         A FUNCTION SUBPROGRAM TO CALCULATE THE AREA OF OWENS LAKE        **
******************************************************************************
```

```
********************************************************************************

      DOUBLE PRECISION FUNCTION OAREA (VOL)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      TVOL=VOL/1.0D9

      IF(TVOL .GE. 0.0D0 .AND. TVOL .LT. 0.16D0) THEN
         OAREA = 1.25D0*TVOL
      ELSE IF (TVOL .GE. 0.16D0 .AND. TVOL .LT. 3.15D0) THEN
         OAREA = 3.01D-2*(TVOL-0.16D0)+0.2D0
      ELSE IF(TVOL .GE. 3.15D0 .AND. TVOL .LT. 30.02D0) THEN
         OAREA = 1.5035D-2*(TVOL-3.15D0)+0.29D0
      ELSE
         OAREA = 0.694D0
      END IF
      OAREA=OAREA*1.0D9
      RETURN
      END


********************************************************************************
********************************************************************************
**       A FUNCTION SUBPROGRAM TO CALCULATE THE AREA OF CHINA LAKE            **
********************************************************************************
********************************************************************************

      DOUBLE PRECISION FUNCTION CAREA (VOL)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      TVOL=VOL/1.0D9
      IF(TVOL .GE. 0.0D0 .AND. TVOL .LT. 0.036D0) THEN
         CAREA = 0.75D0*TVOL
      ELSE IF(TVOL .GE. 0.036D0 .AND. TVOL .LT. 0.696D0) THEN
         CAREA =(.128D0/.66D0)*(TVOL-0.036D0)+0.027D0
      ELSE
         CAREA = 0.155D0
      END IF
      CAREA=CAREA*1.0D9
      RETURN
      END


********************************************************************************
********************************************************************************
**      A FUNCTION SUBPROGRAM TO CALCULATE THE AREA OF SEARLES LAKE           **
********************************************************************************
********************************************************************************

      DOUBLE PRECISION FUNCTION SAREA (VOL)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      TVOL=VOL/1.0D9

      IF(TVOL .GE. 0.0D0 .AND. TVOL .LT. 2.04D0) THEN
         SAREA =(0.245D0/2.04D0)*TVOL
      ELSE IF(TVOL .GE. 2.04D0 .AND. TVOL .LT. 10.75D0) THEN
         SAREA=(0.055D0/8.71D0)*(TVOL-2.04D0)+0.245D0
      ELSE IF(TVOL .GE. 10.75D0 .AND. TVOL .LT. 20.82D0) THEN
         SAREA=(0.05D0/10.07D0)*(TVOL-10.75D0)+0.3D0
      ELSE IF(TVOL .GE. 20.82D0 .AND. TVOL .LT. 33.46D0) THEN
         SAREA=(0.092D0/12.64D0)*(TVOL-20.82D0)+0.35D0
      ELSE IF(TVOL .GE. 33.46D0 .AND. TVOL .LT. 46.6D0) THEN
         SAREA=(0.091D0/13.14D0)*(TVOL-33.46D0)+0.442D0
```

```
      ELSE IF(TVOL .GE. 46.6D0 .AND. TVOL .LT. 65.87D0) THEN
         SAREA=(0.182D0/19.27D0)*(TVOL-46.6D0)+0.533D0
      ELSE IF(TVOL .GE. 65.87D0 .AND. TVOL .LT. 85.28D0) THEN
         SAREA=(0.124D0/19.41D0)*(TVOL-65.87D0)+0.87D0
      ELSE
         SAREA=0.994D0
      ENDIF
      SAREA=SAREA*1.0D9
      RETURN
      END


*******************************************************************************
*******************************************************************************
**    A FUNCTION SUBPROGRAM TO CALCULATE THE VOLUME OF PANAMINT LAKE        **
*******************************************************************************
*******************************************************************************

      DOUBLE PRECISION FUNCTION PVOL(AREA)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      TAREA=AREA/1.0D9

      IF(TAREA .GE. 0.0D0 .AND. TAREA .LT. 0.118D0) THEN
         PVOL =(0.71D0/.118D0)*TAREA
      ELSE IF(TAREA .GE. 0.118D0 .AND. TAREA .LT. 0.175D0) THEN
         PVOL=(2.91D0/0.057D0)*(TAREA-0.118D0)+0.71D0
      ELSE IF(TAREA .GE. 0.175D0 .AND. TAREA .LT. 0.189D0) THEN
         PVOL=(2.D0/0.014D0)*(TAREA-0.1750)+3.62D0
      ELSE IF(TAREA .GE. 0.189D0 .AND. TAREA .LT. 0.242D0) THEN
         PVOL=(6.45D0/0.53D0)*(TAREA-0.189D0)+5.62D0
      ELSE IF(TAREA .GE. 0.242D0 .AND. TAREA .LT. 0.289D0) THEN
         PVOL=(8.22D0/0.047D0)*(TAREA-0.242D0)+12.07D0
      ELSE IF(TAREA .GE. 0.289D0 .AND. TAREA .LT. 0.329D0) THEN
         PVOL=(9.26D0/0.04D0)*(TAREA-0.289D0)+20.29D0
      ELSE IF(TAREA .GE. 0.329D0 .AND. TAREA .LT. 0.369D0) THEN
         PVOL=(10.81D0/0.04D0)*(TAREA-0.329D0)+29.55D0
      ELSE IF(TAREA .GE. 0.369D0 .AND. TAREA .LT. 0.428D0) THEN
         PVOL=(13.93D0/0.059D0)*(TAREA-0.369D0)+40.36D0
      ELSE IF(TAREA .GE. 0.428D0 .AND. TAREA .LT. 0.488D0) THEN
         PVOL=(9.15D0/0.06D0)*(TAREA-0.428D0)+54.29D0
      ELSE IF(TAREA .GE. 0.488D0 .AND. TAREA .LT. 0.524D0) THEN
         PVOL=(7.59D0/0.036D0)*(TAREA-0.488D0)+63.44D0
      ELSE IF(TAREA .GE. 0.524D0 .AND. TAREA .LT. 0.568D0) THEN
         PVOL=(10.92D0/0.044D0)*(TAREA-0.524D0)+71.03D0
      ELSE IF(TAREA .GE. 0.568D0 .AND. TAREA .LT. 0.638D0) THEN
         PVOL=(15.67D0/0.07D0)*(TAREA-0.568D0)+81.95D0
      ELSE IF(TAREA .GE. 0.638D0 .AND. TAREA .LT. 0.727D0) THEN
         PVOL=(10.23D0/0.089D0)*(TAREA-0.638D0)+97.62D0
      ELSE
         PVOL=107.85D0
      ENDIF
      PVOL=PVOL*1.0D9
      RETURN
      END


*******************************************************************************
*******************************************************************************
**       A FUNCTION SUBPROGRAM TO CALCULATE THE VOLUME OF MANLY LAKE        **
*******************************************************************************
*******************************************************************************

      DOUBLE PRECISION FUNCTION DVOL(AREA)
```

```
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      TAREA=AREA/1.0D9

      IF(TAREA .GE. 0.0D0 .AND. TAREA .LT. 0.05D0) THEN
         DVOL =(0.65D0/.05D0)*TAREA
      ELSE IF(TAREA .GE. 0.05D0 .AND. TAREA .LT. 47.0D0) THEN
         DVOL=(46.35D0/0.533D0)*(TAREA-0.05D0)+0.65D0
      ELSE
         DVOL=47.D0
      ENDIF

      DVOL=DVOL*1.0D9

      RETURN
      END


*******************************************************************************
*******************************************************************************
*  THIS IS A FUNCTION SUBPROGRAM TO CALCULATE THE ISOTOPIC ENRICHMENT FACTOR  *
*******************************************************************************
*******************************************************************************

      DOUBLE PRECISION FUNCTION FEPS(TEMP)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)

**************************
**CONVERT TEMP TO KELVIN**
**************************

      TEMPK=TEMP+273.15D0
      FEPS =(DEXP((1.534D0*(1.0D6/(TEMPK)**2)-3.206D0*(1.0D3/TEMPK)+
     +       2.644D0)/1.D3)-1.D0)*1.D3
      RETURN
      END


*******************************************************************************
*******************************************************************************
*     THIS IS A FUNCTION SUBPROGRAM TO CALCULATE THE OSMOTIC COEFFICIENT      *
*******************************************************************************
*******************************************************************************

      DOUBLE PRECISION FUNCTION FPHI(CL_M,SUM)

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

*********************************
**CALCULATE MOLARITIES FOR IONS**
*********************************

      SOD_M=CL_M
      SUM=CL_M+SOD_M

*************************************
**CALCULATE OSMOTIC COEFFICIENT, PHI**
*************************************

      XI = (SOD_M+CL_M+4.D0)/2.D0
      XF = 0.392D0*(DSQRT(XI)/(1.D0+1.2D0*DSQRT(XI)))
      BCL = 0.0765D0 + 0.2664D0 * DEXP(-2.D0*DSQRT(XI))
      BCO3 = 0.18975D0+0.846D0*DEXP(-2.D0*DSQRT(XI))
      DSUM1 = 2.D0*SOD_M*CL_M*(BCL+SOD_M*0.00127D0)
```

```
      DSUM2 = 2.D0*SOD_M*(BCO3-0.048032D0*SOD_M/DSQRT(2.D0))
      FPHI = 1.4121D0*(1.D0+(1.D0/SUM*(2.D0*XI*XF+DSUM1+DSUM2)))
      RETURN
      END


*******************************************************************************
*******************************************************************************
*     A SUBROUTINE TO LINEARLY INTERPOLATE BETWEEN POINTS IN DATA SETS        *
*                   CONTAINING OWENS LAKE PARAMETERS                          *
*******************************************************************************
*******************************************************************************

      SUBROUTINE OINTERP(TIME,NDX,TODELI,TOTEMP,TOEVAP,TOPRECIP,
     +     TODELP,TODELA)

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      LOGICAL GUESS,CPARAM,GUESS2

      PARAMETER(NUMB=2000)
      COMMON/FIRST/WTIME(NUMB),OTEMP(NUMB),OEVAP(NUMB),
     + OHUM(500),OPRECIP(NUMB),ODELP(NUMB),ODELA(NUMB),ODELI(NUMB),
     + GUESS,TEMPC,EVAPC,PRECIPC,DELAC,DELIC,CPARAM,DELPC,
     + EVAPC_P,EVAPC_D

      COMMON/SECOND/CTEMP(NUMB),CEVAP(NUMB),CHUM(500),CPRECIP(NUMB),
     + CDELP(NUMB),CSDELA(NUMB),GUESS2,TEMPC_C,EVAPC_C,
     + PRECIPC_C,DELAC_C,DELPC_C,STEMP(NUMB),SEVAP(NUMB),SHUM(500),
     + SPRECIP(NUMB),SDELP(NUMB),TEMPC_S,EVAPC_S,
     + PRECIPC_S,DELAC_S,DELPC_S

      TODELI=(ODELI(NDX+1)-ODELI(NDX))/(WTIME(NDX+1)-WTIME(NDX))*
     +     (TIME-WTIME(NDX))+ODELI(NDX)

      TOTEMP=(OTEMP(NDX+1)-OTEMP(NDX))/(WTIME(NDX+1)-WTIME(NDX))*
     +     (TIME-WTIME(NDX))+OTEMP(NDX)

      TOEVAP=(OEVAP(NDX+1)-OEVAP(NDX))/(WTIME(NDX+1)-WTIME(NDX))*
     +     (TIME-WTIME(NDX))+OEVAP(NDX)

      TOPRECIP=(OPRECIP(NDX+1)-OPRECIP(NDX))/(WTIME(NDX+1)-
     +     WTIME(NDX))*(TIME-WTIME(NDX))+OPRECIP(NDX)

      TODELP = (ODELP(NDX+1)-ODELP(NDX))/(WTIME(NDX+1)-WTIME(NDX))*
     +     (TIME-WTIME(NDX))+ODELP(NDX)

      TODELA = (ODELA(NDX+1)-ODELA(NDX))/(WTIME(NDX+1)-WTIME(NDX))*
     +     (TIME-WTIME(NDX))+ODELA(NDX)

      RETURN
      END


*******************************************************************************
*******************************************************************************
*        A SUBROUTINE TO CALCULATE HUMIDITY FOR ALL OF THE LAKES             **
*******************************************************************************
*******************************************************************************


      SUBROUTINE HUMTERP(TIME,NDX,T_HUM,HUM,HUMTIME)

      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
```

```
      PARAMETER(NUM=500)
      DOUBLE PRECISION HUM(NUM),HUMTIME(NUM)

      T_HUM=(HUM(NDX+1)-HUM(NDX))/(HUMTIME(NDX+1)-HUMTIME(NDX))*
     +      (TIME-HUMTIME(NDX))+HUM(NDX)


      RETURN
      END


*******************************************************************************
*******************************************************************************
*         A SUBROUTINE TO CALCULATE INFLOW FROM HISTORY                     **
*******************************************************************************
*******************************************************************************


      SUBROUTINE QINTERP(TIME,NDX,QI,QHIST,QITIME)

      IMPLICIT DOUBLE PRECISION(A-H,O-Z)

      DOUBLE PRECISION QHIST(1000),QITIME(1000)

      QI=(QHIST(NDX+1)-QHIST(NDX))/(QITIME(NDX+1)-QITIME(NDX))*
     +      (TIME-QITIME(NDX))+QHIST(NDX)


      RETURN
      END


*******************************************************************************
*******************************************************************************
*     A SUBROUTINE TO LINEARLY INTERPOLATE BETWEEN POINTS IN DATA SETS       *
*                   CONTAINING CHINA LAKE PARAMETERS                         *
*******************************************************************************
*******************************************************************************

      SUBROUTINE CINTERP(TIME,NDX,TCTEMP,TCEVAP,TCPRECIP,
     +      TCDELP,TCDELA)

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      LOGICAL GUESS,CPARAM,GUESS2

      PARAMETER(NUMB=2000)
      COMMON/FIRST/WTIME(NUMB),OTEMP(NUMB),OEVAP(NUMB),
     + OHUM(500),OPRECIP(NUMB),ODELP(NUMB),ODELA(NUMB),ODELI(NUMB),
     + GUESS,TEMPC,EVAPC,PRECIPC,DELAC,DELIC,CPARAM,DELPC,
     + EVAPC_P,EVAPC_D

      COMMON/SECOND/CTEMP(NUMB),CEVAP(NUMB),CHUM(500),CPRECIP(NUMB),
     + CDELP(NUMB),CSDELA(NUMB),GUESS2,TEMPC_C,EVAPC_C,
     + PRECIPC_C,DELAC_C,DELPC_C,STEMP(NUMB),SEVAP(NUMB),SHUM(500),
     + SPRECIP(NUMB),SDELP(NUMB),TEMPC_S,EVAPC_S,
     + PRECIPC_S,DELAC_S,DELPC_S


      TCTEMP=(CTEMP(NDX+1)-CTEMP(NDX))/(WTIME(NDX+1)-WTIME(NDX))*
     +      (TIME-WTIME(NDX))+CTEMP(NDX)

      TCEVAP=(CEVAP(NDX+1)-CEVAP(NDX))/(WTIME(NDX+1)-WTIME(NDX))*
     +      (TIME-WTIME(NDX))+CEVAP(NDX)
```

```
      TCPRECIP=(CPRECIP(NDX+1)-CPRECIP(NDX))/(WTIME(NDX+1)-
     +     WTIME(NDX))*(TIME-WTIME(NDX))+CPRECIP(NDX)

      TCDELP=(CDELP(NDX+1)-CDELP(NDX))/(WTIME(NDX+1)-WTIME(NDX))*
     +     (TIME-WTIME(NDX))+CDELP(NDX)

      TCDELA=(CSDELA(NDX+1)-CSDELA(NDX))/(WTIME(NDX+1)-WTIME(NDX))*
     +     (TIME-WTIME(NDX))+CSDELA(NDX)

      RETURN
      END


*************************************************************************
*************************************************************************
*     A SUBROUTINE TO LINEARLY INTERPOLATE BETWEEN POINTS IN DATA SETS  *
*                FROM THE OWENS LAKE CALCULATIONS                       *
*************************************************************************
*************************************************************************

      SUBROUTINE C2INTERP(TIME,NDX,TCDELI,CQI,TAREA)

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      LOGICAL GUESS,CPARAM,GUESS2

      PARAMETER(NUMA=25000,NUMB=2000)

      COMMON/BOTH/CL(NUMA),TSOD(10),TCL(10),OTIME(NUMA),TCO3(10),
     + OQO(NUMA),QO(10),CONC_CL(NUMA),DOLTEMP,DELDOL,TODELI,TOTEMP,
     + ODEL_OUT(NUMA),OCL_OUT(NUMA),PEVAP(NUMB),DEVAP(NUMB),
     + SUM_PCL_DEP,AREA_P,AREA_D,
     + AREA(NUMA),SOAREA,ALLAREA,CONC_CL_P,
     + CL_P

      COMMON/FIRST/WTIME(NUMB),OTEMP(NUMB),OEVAP(NUMB),
     + OHUM(500),OPRECIP(NUMB),ODELP(NUMB),ODELA(NUMB),ODELI(NUMB),
     + GUESS,TEMPC,EVAPC,PRECIPC,DELAC,DELIC,CPARAM,DELPC,
     + EVAPC_P,EVAPC_D

      COMMON/SECOND/CTEMP(NUMB),CEVAP(NUMB),CHUM(500),CPRECIP(NUMB),
     + CDELP(NUMB),CSDELA(NUMB),GUESS2,TEMPC_C,EVAPC_C,
     + PRECIPC_C,DELAC_C,DELPC_C,STEMP(NUMB),SEVAP(NUMB),SHUM(500),
     + SPRECIP(NUMB),SDELP(NUMB),TEMPC_S,EVAPC_S,
     + PRECIPC_S,DELAC_S,DELPC_S


      TCDELI = (ODEL_OUT(NDX+1)-ODEL_OUT(NDX))/(OTIME(NDX+1)-
     +     OTIME(NDX))*(TIME-OTIME(NDX))+ODEL_OUT(NDX)

      CQI = (OQO(NDX+1)-OQO(NDX))/(OTIME(NDX+1)-OTIME(NDX))*
     +     (TIME-OTIME(NDX))+OQO(NDX)

      TAREA = (AREA(NDX+1)-AREA(NDX))/(OTIME(NDX+1)-OTIME(NDX))*
     +     (TIME-OTIME(NDX))+AREA(NDX)

      RETURN
      END


*************************************************************************
*************************************************************************
*     A SUBROUTINE TO LINEARLY INTERPOLATE BETWEEN POINTS IN AN ARRAY   *
```

```
*                FROM THE OWENS LAKE SALT OUTFLOW HISTORY                     *
*******************************************************************************
*******************************************************************************

      SUBROUTINE C3INTERP(TIME,NDX,SLTNUM)

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      LOGICAL GUESS,CPARAM,GUESS2

      PARAMETER(NUMA=25000,NUMB=2000)

      COMMON/BOTH/CL(NUMA),TSOD(10),TCL(10),OTIME(NUMA),TCO3(10),
     + OQO(NUMA),QO(10),CONC_CL(NUMA),DOLTEMP,DELDOL,TODELI,TOTEMP,
     + ODEL_OUT(NUMA),OCL_OUT(NUMA),PEVAP(NUMB),DEVAP(NUMB),
     + SUM_PCL_DEP,AREA_P,AREA_D,
     + AREA(NUMA),SOAREA,ALLAREA,CONC_CL_P,
     + CL_P

      COMMON/FIRST/WTIME(NUMB),OTEMP(NUMB),OEVAP(NUMB),
     + OHUM(500),OPRECIP(NUMB),ODELP(NUMB),ODELA(NUMB),ODELI(NUMB),
     + GUESS,TEMPC,EVAPC,PRECIPC,DELAC,DELIC,CPARAM,DELPC,
     + EVAPC_P,EVAPC_D

      COMMON/SECOND/CTEMP(NUMB),CEVAP(NUMB),CHUM(500),CPRECIP(NUMB),
     + CDELP(NUMB),CSDELA(NUMB),GUESS2,TEMPC_C,EVAPC_C,
     + PRECIPC_C,DELAC_C,DELPC_C,STEMP(NUMB),SEVAP(NUMB),SHUM(500),
     + SPRECIP(NUMB),SDELP(NUMB),TEMPC_S,EVAPC_S,
     + PRECIPC_S,DELAC_S,DELPC_S


      SLTNUM = (OCL_OUT(NDX+1)-OCL_OUT(NDX))/(OTIME(NDX+1)-
     +     OTIME(NDX))*(TIME-OTIME(NDX))+OCL_OUT(NDX)

      RETURN
      END


*******************************************************************************
*******************************************************************************
*     A SUBROUTINE TO LINEARLY INTERPOLATE BETWEEN POINTS IN DATA SETS        *
*                  CONTAINING SEARLES LAKE PARAMETERS                         *
*******************************************************************************
*******************************************************************************

      SUBROUTINE SINTERP(TIME,NDX,TSTEMP,TSEVAP,TSPRECIP,
     +     TSDELP,TSDELA)

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      LOGICAL GUESS,CPARAM,GUESS2

      PARAMETER(NUMB=2000)
      COMMON/FIRST/WTIME(NUMB),OTEMP(NUMB),OEVAP(NUMB),
     + OHUM(500),OPRECIP(NUMB),ODELP(NUMB),ODELA(NUMB),ODELI(NUMB),
     + GUESS,TEMPC,EVAPC,PRECIPC,DELAC,DELIC,CPARAM,DELPC,
     + EVAPC_P,EVAPC_D

      COMMON/SECOND/CTEMP(NUMB),CEVAP(NUMB),CHUM(500),CPRECIP(NUMB),
     + CDELP(NUMB),CSDELA(NUMB),GUESS2,TEMPC_C,EVAPC_C,
     + PRECIPC_C,DELAC_C,DELPC_C,STEMP(NUMB),SEVAP(NUMB),SHUM(500),
     + SPRECIP(NUMB),SDELP(NUMB),TEMPC_S,EVAPC_S,
     + PRECIPC_S,DELAC_S,DELPC_S
```

```
      TSTEMP=(STEMP(NDX+1)-STEMP(NDX))/(WTIME(NDX+1)-WTIME(NDX))*
     +       (TIME-WTIME(NDX))+STEMP(NDX)

      TSEVAP=(SEVAP(NDX+1)-SEVAP(NDX))/(WTIME(NDX+1)-WTIME(NDX))*
     +       (TIME-WTIME(NDX))+SEVAP(NDX)

      TSPRECIP=(SPRECIP(NDX+1)-SPRECIP(NDX))/(WTIME(NDX+1)-
     +       WTIME(NDX))*(TIME-WTIME(NDX))+SPRECIP(NDX)

      TSDELP=(SDELP(NDX+1)-SDELP(NDX))/(WTIME(NDX+1)-WTIME(NDX))*
     +       (TIME-WTIME(NDX))+SDELP(NDX)

      TSDELA=(CSDELA(NDX+1)-CSDELA(NDX))/(WTIME(NDX+1)-WTIME(NDX))*
     +       (TIME-WTIME(NDX))+CSDELA(NDX)

      RETURN
      END


*****************************************************************************
*****************************************************************************
*     A SUBROUTINE TO LINEARLY INTERPOLATE BETWEEN POINTS IN AN ARRAY       *
*                FROM THE OWENS LAKE SALT OUTFLOW HISTORY                   *
*****************************************************************************
*****************************************************************************

      SUBROUTINE PDINTERP(TIME,NDX,TPEVAP,TDEVAP)

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      LOGICAL GUESS,CPARAM,GUESS2

      PARAMETER(NUMA=25000,NUMB=2000)

      COMMON/BOTH/CL(NUMA),TSOD(10),TCL(10),OTIME(NUMA),TCO3(10),
     + OQO(NUMA),QO(10),CONC_CL(NUMA),DOLTEMP,DELDOL,TODELI,TOTEMP,
     + ODEL_OUT(NUMA),OCL_OUT(NUMA),PEVAP(NUMB),DEVAP(NUMB),
     + SUM_PCL_DEP,AREA_P,AREA_D,
     + AREA(NUMA),SOAREA,ALLAREA,CONC_CL_P,
     + CL_P

      COMMON/FIRST/WTIME(NUMB),OTEMP(NUMB),OEVAP(NUMB),
     + OHUM(500),OPRECIP(NUMB),ODELP(NUMB),ODELA(NUMB),ODELI(NUMB),
     + GUESS,TEMPC,EVAPC,PRECIPC,DELAC,DELIC,CPARAM,DELPC,
     + EVAPC_P,EVAPC_D

      COMMON/SECOND/CTEMP(NUMB),CEVAP(NUMB),CHUM(500),CPRECIP(NUMB),
     + CDELP(NUMB),CSDELA(NUMB),GUESS2,TEMPC_C,EVAPC_C,
     + PRECIPC_C,DELAC_C,DELPC_C,STEMP(NUMB),SEVAP(NUMB),SHUM(500),
     + SPRECIP(NUMB),SDELP(NUMB),TEMPC_S,EVAPC_S,
     + PRECIPC_S,DELAC_S,DELPC_S


      TPEVAP = (PEVAP(NDX+1)-PEVAP(NDX))/(WTIME(NDX+1)-
     +       WTIME(NDX))*(TIME-WTIME(NDX))+PEVAP(NDX)

      TDEVAP = (DEVAP(NDX+1)-DEVAP(NDX))/(WTIME(NDX+1)-
     +       WTIME(NDX))*(TIME-WTIME(NDX))+DEVAP(NDX)

      RETURN
      END


*****************************************************************************
```

```
*******************************************************************************
**  A FUNCTION SUBPROGRAM TO CALCULATE THE RELATIVE ISOTOPIC ENRICHMENT OF   **
**  THE EVAPORATING WATER.                                                   **
*******************************************************************************
*******************************************************************************

      DOUBLE PRECISION FUNCTION DELE(DELL,EPS,HUM)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      PARAMETER(C=6.8D-3)

      DELE=(((1.D0+1.D-3*DELL)*(1.D0-C))/((1.D0+1.D-3*EPS)*(1.D0-C*
     +     HUM))-1.D0)*1.0D3

      RETURN
      END


*******************************************************************************
*******************************************************************************
**         A FUNCTION SUBPROGRAM TO CALCULATE THE INFLOW, QI(T)             **
*******************************************************************************
*******************************************************************************

      DOUBLE PRECISION FUNCTION FQI(X)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)

      COMMON/INFLOW/INCHOICE,A,B,C

      IF(INCHOICE .EQ. 1)THEN
         FQI=A*X+B
      ELSE IF(INCHOICE .EQ. 2)THEN
         FQI=B*DEXP(A*X)
      ELSE IF(INCHOICE .EQ. 3)THEN
         IF(X .LT. 1.D0)X=1.D0
         FQI=B+A*DLOG10(X)
      ELSE IF(INCHOICE .EQ. 4)THEN
         FQI=B+A*X**C
      ELSE IF(INCHOICE .EQ. 5)THEN
         FQI=B+A*DSIN(C*X)
      ELSE IF(INCHOICE .EQ. 6)THEN
         IF(X .LT. 1.D0)THEN
            FQI=B
         ELSE
            FQI=B+A*B
         ENDIF
      ELSE IF(INCHOICE .EQ. 7)THEN
         FQI=0.D0
      ENDIF
      IF(FQI .LT. 0.D0)FQI=0.D0
      RETURN
      END


*******************************************************************************
*******************************************************************************
*         A SUBROUTINE TO FIND THE TIME INDEX WITH A BINARY SEARCH           *
*******************************************************************************
*******************************************************************************

      SUBROUTINE FINDT(NPTS,TM,INDEX,FOUND,TIME)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      LOGICAL FOUND
      PARAMETER(NUMB=2000)
      DIMENSION TIME(NUMB)
```

```
      COMMON/SEARCH/IFIRST,ILAST,HIFIRST,HILAST

      TFST=IFIRST
      TLST=ILAST
      FOUND = .FALSE.
      DO 200 I = 1,100
         IF(TLST .LT. TFST) THEN
            INDEX=TLST
            GOTO 210
         ENDIF
         IF( TFST .LE. TLST .AND. .NOT. FOUND)THEN
            MIDDLE = (TFST+TLST)/2
            TEST = ABS(TM-TIME(MIDDLE))
            IF( TEST .LT. 1.0D-1) THEN
               FOUND = .TRUE.
               INDEX=MIDDLE
               GOTO 210
            ELSE IF(TM .LT. TIME(MIDDLE))THEN
               TLST = MIDDLE-1
            ELSE
               TFST = MIDDLE+1
            END IF
         END IF
200   CONTINUE

210   RETURN
      END


*******************************************************************************
*******************************************************************************
*     A SUBROUTINE TO FIND THE HUMIDITY TIME INDEX WITH A BINARY SEARCH      *
*******************************************************************************
*******************************************************************************

      SUBROUTINE FINDHT(NPTS,TM,INDEX,TIME)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      LOGICAL FOUND
      DIMENSION TIME(500)
      IFIRST=1
      ILAST=NPTS
      FOUND = .FALSE.
      DO 200 I = 1,100
         IF(ILAST .LT. IFIRST) THEN
            INDEX=ILAST
            GOTO 210
         ENDIF
         IF( IFIRST .LE. ILAST .AND. .NOT. FOUND)THEN
            MIDDLE = (IFIRST+ILAST)/2
            TEST = ABS(TM-TIME(MIDDLE))
            IF( TEST .LT. 1.0D-1) THEN
               FOUND = .TRUE.
               INDEX=MIDDLE
               GOTO 210
            ELSE IF(TM .LT. TIME(MIDDLE))THEN
               ILAST = MIDDLE-1
            ELSE
               IFIRST = MIDDLE+1
            END IF
         END IF
200   CONTINUE

210   RETURN
      END
```

```
******************************************************************************
******************************************************************************
*     A SUBROUTINE TO FIND THE INFLOW TIME INDEX WITH A BINARY SEARCH        *
******************************************************************************
******************************************************************************

        SUBROUTINE FINDQT(NPTS,TM,INDEX,TIME)
        IMPLICIT DOUBLE PRECISION(A-H,O-Z)
        LOGICAL FOUND
        DIMENSION TIME(1000)
        IFIRST=1
        ILAST=NPTS
        FOUND = .FALSE.
        DO 200 I = 1,100
           IF(ILAST .LT. IFIRST) THEN
              INDEX=ILAST
              GOTO 210
           ENDIF
           IF( IFIRST .LE. ILAST .AND. .NOT. FOUND)THEN
              MIDDLE = (IFIRST+ILAST)/2
              TEST = ABS(TM-TIME(MIDDLE))
              IF( TEST .LT. 1.0D-1) THEN
                 FOUND = .TRUE.
                 INDEX=MIDDLE
                 GOTO 210
              ELSE IF(TM .LT. TIME(MIDDLE))THEN
                 ILAST = MIDDLE-1
              ELSE
                 IFIRST = MIDDLE+1
              END IF
           END IF
200     CONTINUE

210     RETURN
        END


******************************************************************************
******************************************************************************
*          A SUBROUTINE TO FIND THE TIME INDEX WITH A BINARY SEARCH          *
******************************************************************************
******************************************************************************

        SUBROUTINE FINDT2(NPTS,TM,INDEX2,FOUND2,TIME)
        IMPLICIT DOUBLE PRECISION(A-H,O-Z)
        LOGICAL FOUND2
        PARAMETER(NUMA=25000)
        DIMENSION TIME(NUMA)
        IFIRST=1
        ILAST=NPTS
        FOUND2 = .FALSE.
        DO 200 I = 1,100
           IF(ILAST .LT. IFIRST) THEN
              INDEX2=ILAST
              GOTO 210
           ENDIF
           IF( IFIRST .LE. ILAST .AND. .NOT. FOUND2) THEN
              MIDDLE = (IFIRST+ILAST)/2
              TEST = ABS(TM-TIME(MIDDLE))
              IF( TEST .LT. 1.0D-1) THEN
                 FOUND2 = .TRUE.
                 INDEX2=MIDDLE
                 GOTO 210
```

```
            ELSE IF(TM .GT. TIME(MIDDLE))THEN
               ILAST = MIDDLE-1
            ELSE
               IFIRST = MIDDLE+1
            END IF
          END IF
200     CONTINUE

210     RETURN
        END


***********************************************************
***********************************************************
**    A FUNCTION SUBPROGRAM TO  CALCULATE DEL DOLOMITE **
***********************************************************
***********************************************************

        DOUBLE PRECISION FUNCTION FDDOL(TEMP,DELH2O)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)

*************************************
** CONVERT TEMP TO DEGREES KELVIN **
*************************************

        TEMPK = TEMP+273.15D0

***********************************************
** CALCULATE EPSILON FOR DOLOMITE AND WATER **
***********************************************

        EPSDOL=(DEXP((3.2D0*(1.D6/TEMPK**2)-4.3D0)/1.D3)-1.D0)*1.D3

        FDDOL=EPSDOL+DELH2O*(EPSDOL/1.0D3+1.D0)

        RETURN
        END


***********************************************************
***********************************************************
**    A FUNCTION SUBPROGRAM TO  CALCULATE DEL WATER     **
***********************************************************
***********************************************************

        DOUBLE PRECISION FUNCTION W_DEL(DELDOL,TEMP)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)

*************************************
** CONVERT TEMP TO DEGREES KELVIN **
*************************************

        TEMPK = TEMP+273.15D0

***********************************************
** CALCULATE EPSILON FOR DOLOMITE AND WATER **
***********************************************

        EPSDOL=(DEXP((3.2D0*(1.D6/TEMPK**2)-4.3D0)/1.D3)-1.D0)*1.D3

        W_DEL=(DELDOL-EPSDOL)/((EPSDOL/1.D3)+1.D0)

        RETURN
        END
```

1. Introduction

One goal of this project was to quantitatively reconstruct lake surface-area histories for the paleo-Owens River system and San Agustin lake, using the stable isotope records. Examples of lake-level and stable isotope evolution models are presented in Chapter II, Section , using simple analytic solutions to simulate the histories. Although this modeling approach is instructive, it is not suitable for reconstructing long lake-level histories because continuous variations in model input parameters cannot readily be incorporated into analytical solutions. We have therefore constructed several numerical models for this purpose.

The main tool for the lake-level reconstructions is a transient numerical model. The equations used were initially presented by Gonfiantini (1965). The model is based on the one described in Phillips and others(1986), but differs from that model in being formulated on a time-derivative basis rather than a volume-derivative basis. The model consists of linked water mass balance and isotope mass balance equations. In order to compute the mass balances the model requires histories of temperature, humidity, precipitation rate, evaporation rate, isotopic composition of inflow water, isotopic composition of precipitation on the lake surface, and isotopic composition of the atmospheric humidity. The independent parameter that "drives" the model is the inflow to the lake or lakes. The dependent variables calculated by the model are the surface area of the lake(s) and the isotopic

1. Introduction

In recent years, analysis of continuous marine sediment cores has allowed great advances in the understanding of marine and polar climate (Shackleton and Opdyke, 1973; Hays et

al., 1977 ... t been made in the
understand ... ie to a lack of old,
continuous ... table isotope record
developed ... Kerr-McGee KM-3 core
from Sear ... tudy addresses this
problem by ... ig lake surface-area
histories ... em. The paleo-Owens
River clos ... rised of Owens Lake,
China Lake, ... and Lake Manly (Death

Ind Study Version
L 13.9°C
H 18.8°C
Searsim.
searcon

Valley). Water levels in closed-basin lakes are a direct measure of the basin water balance, and thus comparison of the water-level reconstruction with independent climatic records

will enable dete ... the hydrologic
system to climat

Ind. Study Version

The main to ... ructions is a
transient numeri ... were initially
presented by Gon ... s based on the
one described in ... t differs from
that model in b ... rivative basis
rather than a vol ... lel consists of
linked water mass ... nce equations.

In order to compute the mass balances the model requires histories of temperature, humidity, precipitation rate,

OLD MODEL

```
***********************************************************
**                              PROGRAM ISO.FOR
***********************************************************

***********************************************************
***********************************************************
* THIS PROGRAM CALCULATES CHANGES IN LAKE VOLUME AND ISOTOPI
* WITH RESPECT TO TIME.
***********************************************************
***********************************************************

***********************************************************
*       A PROGRAM TO CALL THE RUNGE-KUTTA-FEHLBERG ORDER 4 R
*       A SYSTEM OF PARTIAL DIFFERENTIAL EQUATION OF THE FOR
***********************************************************
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        DIMENSION X(3),TOL(3),X_CS(4),TOL_CS(4)

        LOGICAL GUESS,GRAF,CPARAM,GUESS2,FOUND,SU

        PARAMETER(NUMA=15000,VOLMAX=30.02D9,VOLMAX_C=.696D9,
     +       VOLMAX_S=85.28D9,NUMB=2500)

        COMMON/FIRST/A,B,WTIME(NUMB),OTEMP(NUMB),OEVAP(NUMB),
     +  OHUM(500),OPRECIP(NUMB),ODELP(NUMB),ODELA(NUMB),ODELI(NUMB),
     +  GUESS,C,TEMPC,EVAPC,PRECIPC,DELAC,DELIC,CPARAM,DELPC,
     +  EVAPC_P,EVAPC_D

        COMMON/SECOND/CTEMP(NUMB),CEVAP(NUMB),CHUM(500),CPRECIP(NUMB),
     +  CDELP(NUMB),CSDELA(NUMB),GUESS2,TEMPC_C,EVAPC_C,
     +  PRECIPC_C,DELAC_C,DELPC_C,STEMP(NUMB),SEVAP(NUMB),SHUM(500),
     +  SPRECIP(NUMB),SDELP(NUMB),TEMPC_S,EVAPC_S,
     +  PRECIPC_S,DELAC_S,DELPC_S

        COMMON/BOTH/CL(NUMA),TSOD(10),TCL(10),OTIME(NUMA),TCO3(10),
     +  OQO(NUMA),QO(10),CONC_CL(NUMA),DOLTEMP,DELDOL,TODELI,TOTEMP,
     +  ODEL_OUT(NUMA),OCL_OUT(NUMA),PEVAP(NUMB),DEVAP(NUMB),
     +  SUM_PCL_DEP(NUMA),SUM_DCL_DEP(NUMA),AREA_P,AREA_D,
     +  AREA(NUMA),SOAREA,ALLAREA(NUMA),CONC_CL_P(NUMA),CONC_CL_D(NUMA),
     +  CL_P(NUMA),CL_D(NUMA),INCHOICE

        COMMON/BOTH2/CL_C(NUMA),TSOD_C(10),TCL_C(10),TIME,TCO3_C(10),
     +  CQO(10),GRAF,CONC_CL_C(NUMA),DOLTEMP_C,DELDOL_C,
     +  TCDELI,TCTEMP,CDEL_OUT(NUMA),CCL_OUT(NUMA),CL_S(NUMA),NOPTS,
     +  TSOD_S(10),TCL_S(10),TCO3_S(10),SQO(10),CONC_CL_S(NUMA),
     +  DOLTEMP_S,DELDOL_S,TSDELI,TSTEMP,SCL_DEP(NUMA),TTLCL_IN,NPTS,
     +  SUM_SCL_DEP(NUMA),QI_C,QI_S,PQI

        COMMON/HIST/QIHIST(1000),HUMTIME(500),NHPTS,SU,SUT(15),
     +  SAVETIME,NUMST,QITIME(1000),NQPTS

        OPEN(UNIT=98,FILE='OWENS.INP',STATUS='OLD')
        WRITE(6,*)
        WRITE(6,*)'READING OWENS LAKE STARTING PARAMETERS'
        READ(98,*)ITMAX,N,(X(I),I=1,N),DTMAX,DTMIN,(TOL(I),I=1,N),
     +       CONC_CL(1)

        OPEN(UNIT=89,FILE='C_S.INP',STATUS='OLD')
        WRITE(6,*)
        WRITE(6,*)'READING CHINA & SEARLES LAKE STARTING PARAMETERS'
        READ(89,*)ITMAX_CS,NCS,(X_CS(I),I=1,NCS),DTMAX_CS,DTMIN_CS,
     +       (TOL_CS(I),I=1,NCS),CONC_CL_C(1),CONC_CL_S(1)
```

ISO.FOR

OLD MODEL

```
      OPEN(UNIT=69,FILE='P_D.INP',STATUS='OLD')
      WRITE(6,*)
      WRITE(6,*)'READING PANAMINT AND DEATH VALLEY STUFF'
      READ(69,*)AREA_P,CONC_CL_P(1),SUM_PCL_DEP(1),AREA_D,
     +     CONC_CL_D(1),SUM_DCL_DEP(1)

*****************************************
** A CHANCE TO ADJUST INPUT PARAMETERS **
*****************************************

      WRITE(6,*)
      WRITE(6,*)'THE CURRENT OWENS LAKE PARAMETERS ARE:'
      WRITE(6,*)'   1 : MAX ITERATIONS =',ITMAX
      WRITE(6,*)'   2 : LAKE VOL =',X(1)
      WRITE(6,*)'   3 : DEL O-18 DOLOMITE =',X(2)
      WRITE(6,*)'   4 : MAX TIME STEP =',DTMAX
      WRITE(6,*)'   5 : MIN TIME STEP =',DTMIN
      WRITE(6,*)'   6 : LAKE VOL TOLERANCE=',TOL(1)
      WRITE(6,*)'   7 : O-18 TOLERANCE=',TOL(2)
      WRITE(6,*)'   8 : CHLORIDE CONC=',CONC_CL(1)

      WRITE(6,*)
      WRITE(6,*)'DO YOU WANT TO CHANGE ANY OF THE STARTING PARAMETERS ?'
      WRITE(6,*)'1=YES    0=NO'
      READ(5,*)ISEE
      WRITE(6,*)

      IF(ISEE .EQ. 1)THEN

         WRITE(6,*)
         WRITE(6,*)'HOW MANY PARAMETERS WOULD YOU LIKE TO CHANGE ?'
         READ(5,*)K
         WRITE(6,*)

         DO 250 I = 1,K
            WRITE(6,*)
300         WRITE(6,*)'ENTER THE PARAMETER NUMBER'
            WRITE(6,*)
            WRITE(6,*)'   1 : MAX ITERATIONS ='
            WRITE(6,*)'   2 : LAKE VOL ='
            WRITE(6,*)'   3 : DEL O-18 DOLOMITE ='
            WRITE(6,*)'   4 : MAX TIME STEP ='
            WRITE(6,*)'   5 : MIN TIME STEP ='
            WRITE(6,*)'   6 : LAKE VOL TOLERANCE='
            WRITE(6,*)'   7 : O-18 TOLERANCE='
            WRITE(6,*)'   8 : CHLORIDE CONC='
            READ(5,*)NUMP
            IF(NUMP .EQ. 1)THEN
               WRITE(6,*)
               WRITE(6,*)'MAX ITERATIONS :',ITMAX
               WRITE(6,*)'ENTER NEW VALUE'
               READ(5,*)ITMAX
            ELSE IF(NUMP .EQ. 2)THEN
               WRITE(6,*)
               WRITE(6,'(A,D15.7)')' LAKE VOLUME :',X(1)
               WRITE(6,*)'NOTE: MAX LAKE VOL IS 30.02D9 (M^3)'
               WRITE(6,*)'ENTER NEW VALUE'
               READ(5,*)X(1)
            ELSE IF(NUMP .EQ. 3)THEN
               WRITE(6,*)
               WRITE(6,*)'DEL O-18 DOLOMITE :',X(2)
               WRITE(6,*)'ENTER NEW VALUE'
               READ(5,*)X(2)
            ELSE IF(NUMP .EQ. 4)THEN
               WRITE(6,*)
               WRITE(6,*)'MAX TIME STEP :',DTMAX
```

```
               WRITE(6,*)'ENTER NEW VALUE'
               READ(5,*)DTMAX
            ELSE IF(NUMP .EQ. 5)THEN
               WRITE(6,*)
               WRITE(6,*)'MIN TIME STEP :',DTMIN
               WRITE(6,*)'ENTER NEW VALUE'
               READ(5,*)DTMIN
            ELSE IF(NUMP .EQ. 6)THEN
               WRITE(6,*)
               WRITE(6,*)'LAKE VOL TOLERANCE :',TOL(1)
               WRITE(6,*)'ENTER NEW VALUE'
               READ(5,*)TOL(1)
            ELSE IF(NUMP .EQ. 7)THEN
               WRITE(6,*)
               WRITE(6,*)'DEL O-18 TOLERANCE :',TOL(2)
               WRITE(6,*)'ENTER NEW VALUE'
               READ(5,*)TOL(2)
            ELSE IF(NUMP .EQ. 8)THEN
               WRITE(6,*)
               WRITE(6,*)'CHLORIDE CONC :',CONC_CL(1)
               WRITE(6,*)'NOTE: 6.1 IS SATURATION'
               WRITE(6,*)'ENTER NEW VALUE'
               READ(5,*)CONC_CL(1)
            ELSE
               WRITE(6,*)
 21             WRITE(6,*)'YOU SUFFER FROM CALCULATOR DEPENDENCY'
               WRITE(6,*)'PICK A NUMBER BETWEEN 1 AND 8'
               WRITE(6,*)
               GO TO 300
            ENDIF
250      CONTINUE

************************************
** WRITE NEW VALUES TO INPUT FILE **
************************************

         REWIND 98
         WRITE(98,*)ITMAX,N,(X(I),I=1,N),DTMAX,DTMIN,(TOL(I),
     +        I=1,N),CONC_CL(1)
         CLOSE(UNIT=98)
      ENDIF

****************************************************************
** A LOGICAL VARIABLE USED TO WRITE STARTING VALUES TO FILES **
****************************************************************

      GUESS=.TRUE.

***************************************************
** CALCULATE MOLES OF CHLORIDE FROM CONC AND VOL **
***************************************************

      IF(X(1) .LE. 0.D0)THEN
         CL(1)=0.D0
         CONC_CL(1)=0.D0
      ELSE
         CL(1)=CONC_CL(1)*(X(1)*1.D3)
      ENDIF


********************************************************
** A CHANCE TO ADJUST INPUT PARAMETERS FOR CHINA LAKE **
********************************************************

      WRITE(6,*)
      WRITE(6,*)'THE CURRENT CHINA LAKE PARAMETERS ARE:'
```

```
      WRITE(6,*)'   1 : MAX ITERATIONS =',ITMAX_CS
      WRITE(6,*)'   2 : LAKE VOL =',X_CS(1)
      WRITE(6,*)'   3 : DEL O-18 DOLOMITE =',X_CS(2)
      WRITE(6,*)'   4 : MAX TIME STEP =',DTMAX_CS
      WRITE(6,*)'   5 : MIN TIME STEP =',DTMIN_CS
      WRITE(6,*)'   6 : LAKE VOL TOLERANCE=',TOL_CS(1)
      WRITE(6,*)'   7 : O-18 TOLERANCE=',TOL_CS(2)
      WRITE(6,*)'   8 : CHLORIDE CONC=',CONC_CL_C(1)

      WRITE(6,*)
      WRITE(6,*)'DO YOU WANT TO CHANGE ANY OF THE STARTING PARAMETERS ?'
      WRITE(6,*)'1=YES    0=NO'
      READ(5,*)ISEE
      WRITE(6,*)

      IF(ISEE .EQ. 1)THEN

         WRITE(6,*)
         WRITE(6,*)'HOW MANY PARAMETERS WOULD YOU LIKE TO CHANGE ?'
         READ(5,*)K
         WRITE(6,*)

         DO 350 I = 1,K
            WRITE(6,*)
360         WRITE(6,*)'ENTER THE PARAMETER NUMBER'
            WRITE(6,*)
            WRITE(6,*)'   1 : MAX ITERATIONS ='
            WRITE(6,*)'   2 : LAKE VOL ='
            WRITE(6,*)'   3 : DEL O-18 DOLOMITE ='
            WRITE(6,*)'   4 : MAX TIME STEP ='
            WRITE(6,*)'   5 : MIN TIME STEP ='
            WRITE(6,*)'   6 : LAKE VOL TOLERANCE='
            WRITE(6,*)'   7 : O-18 TOLERANCE='
            WRITE(6,*)'   8 : CHLORIDE CONC='
            READ(5,*)NUMP
            IF(NUMP .EQ. 1)THEN
               WRITE(6,*)
               WRITE(6,*)'MAX ITERATIONS :',ITMAX_CS
               WRITE(6,*)'ENTER NEW VALUE'
               READ(5,*)ITMAX_CS
            ELSE IF(NUMP .EQ. 2)THEN
               WRITE(6,*)
               WRITE(6,'(A,D15.7)')' LAKE VOLUME :',X_CS(1)
               WRITE(6,*)'NOTE: MAX LAKE VOL IS 0.696D9 (M^3)'
               WRITE(6,*)'ENTER NEW VALUE'
               READ(5,*)X_CS(1)
            ELSE IF(NUMP .EQ. 3)THEN
               WRITE(6,*)
               WRITE(6,*)'DEL O-18 DOLOMITE :',X_CS(2)
               WRITE(6,*)'ENTER NEW VALUE'
               READ(5,*)X_CS(2)
            ELSE IF(NUMP .EQ. 4)THEN
               WRITE(6,*)
               WRITE(6,*)'MAX TIME STEP :',DTMAX_CS
               WRITE(6,*)'ENTER NEW VALUE'
               READ(5,*)DTMAX_CS
            ELSE IF(NUMP .EQ. 5)THEN
               WRITE(6,*)
               WRITE(6,*)'MIN TIME STEP :',DTMIN_CS
               WRITE(6,*)'ENTER NEW VALUE'
               READ(5,*)DTMIN_CS
            ELSE IF(NUMP .EQ. 6)THEN
               WRITE(6,*)
               WRITE(6,*)'LAKE VOL TOLERANCE :',TOL_CS(1)
               WRITE(6,*)'ENTER NEW VALUE'
               READ(5,*)TOL_CS(1)
```

```
            ELSE IF(NUMP .EQ. 7)THEN
               WRITE(6,*)
               WRITE(6,*)'DEL O-18 TOLERANCE :',TOL_CS(2)
               WRITE(6,*)'ENTER NEW VALUE'
               READ(5,*)TOL_CS(2)
            ELSE IF(NUMP .EQ. 8)THEN
               WRITE(6,*)
               WRITE(6,*)'CHLORIDE CONC :',CONC_CL_C(1)
               WRITE(6,*)'NOTE: 6.1 IS SATURATION'
               WRITE(6,*)'ENTER NEW VALUE'
               READ(5,*)CONC_CL_C(1)
            ELSE
               WRITE(6,*)
               WRITE(6,*)'OBVIOUSLY MATH IS NOT YOUR FORTE'
               WRITE(6,*)'PICK A NUMBER BETWEEN 1 AND 8'
               WRITE(6,*)
               GO TO 360
            ENDIF
350      CONTINUE
      ENDIF

****************************************************
** CALCULATE MOLES OF CHLORIDE FROM CONC AND VOL **
****************************************************

      IF(X_CS(1) .LE. 0.D0)THEN
         CL_C(1)=0.D0
         CONC_CL_C(1)=0.D0
      ELSE
         CL_C(1)=CONC_CL_C(1)*(X_CS(1)*1.D3)
      ENDIF


***********************************************************
** A CHANCE TO ADJUST INPUT PARAMETERS FOR SEARLES LAKE **
***********************************************************

      WRITE(6,*)
      WRITE(6,*)'THE CURRENT SEARLES LAKE PARAMETERS ARE:'
      WRITE(6,*)'   1 : MAX ITERATIONS =',ITMAX_CS
      WRITE(6,*)'   2 : LAKE VOL =',X_CS(3)
      WRITE(6,*)'   3 : DEL O-18 DOLOMITE =',X_CS(4)
      WRITE(6,*)'   4 : MAX TIME STEP =',DTMAX_CS
      WRITE(6,*)'   5 : MIN TIME STEP =',DTMIN_CS
      WRITE(6,*)'   6 : LAKE VOL TOLERANCE=',TOL_CS(3)
      WRITE(6,*)'   7 : O-18 TOLERANCE=',TOL_CS(4)
      WRITE(6,*)'   8 : CHLORIDE CONC=',CONC_CL_S(1)

      WRITE(6,*)
      WRITE(6,*)'DO YOU WANT TO CHANGE ANY OF THE STARTING PARAMETERS ?'
      WRITE(6,*)'1=YES    0=NO'
      READ(5,*)ISEE
      WRITE(6,*)

      IF(ISEE .EQ. 1)THEN

         WRITE(6,*)
         WRITE(6,*)'HOW MANY PARAMETERS WOULD YOU LIKE TO CHANGE ?'
         READ(5,*)K
         WRITE(6,*)

         DO 370 I = 1,K
            WRITE(6,*)
380         WRITE(6,*)'ENTER THE PARAMETER NUMBER'
            WRITE(6,*)
            WRITE(6,*)'   1 : MAX ITERATIONS ='
```

```
         WRITE(6,*)'   2 : LAKE VOL ='
         WRITE(6,*)'   3 : DEL O-18 DOLOMITE='
         WRITE(6,*)'   4 : MAX TIME STEP ='
         WRITE(6,*)'   5 : MIN TIME STEP ='
         WRITE(6,*)'   6 : LAKE VOL TOLERANCE='
         WRITE(6,*)'   7 : O-18 TOLERANCE='
         WRITE(6,*)'   8 : CHLORIDE CONC='
         READ(5,*)NUMP
         IF(NUMP .EQ. 1)THEN
            WRITE(6,*)
            WRITE(6,*)'MAX ITERATIONS :',ITMAX_CS
            WRITE(6,*)'ENTER NEW VALUE'
            READ(5,*)ITMAX_CS
         ELSE IF(NUMP .EQ. 2)THEN
            WRITE(6,*)
            WRITE(6,'(A,D15.7)')' LAKE VOLUME :',X_CS(3)
            WRITE(6,*)'NOTE: MAX LAKE VOL IS 85.28D9 (M^3)'
            WRITE(6,*)'ENTER NEW VALUE'
            READ(5,*)X_CS(3)
         ELSE IF(NUMP .EQ. 3)THEN
            WRITE(6,*)
            WRITE(6,*)'DEL O-18 DOLOMITE :',X_CS(4)
            WRITE(6,*)'ENTER NEW VALUE'
            READ(5,*)X_CS(4)
         ELSE IF(NUMP .EQ. 4)THEN
            WRITE(6,*)
            WRITE(6,*)'MAX TIME STEP :',DTMAX_CS
            WRITE(6,*)'ENTER NEW VALUE'
            READ(5,*)DTMAX_CS
         ELSE IF(NUMP .EQ. 5)THEN
            WRITE(6,*)
            WRITE(6,*)'MIN TIME STEP :',DTMIN_CS
            WRITE(6,*)'ENTER NEW VALUE'
            READ(5,*)DTMIN_CS
         ELSE IF(NUMP .EQ. 6)THEN
            WRITE(6,*)
            WRITE(6,*)'LAKE VOL TOLERANCE :',TOL_CS(3)
            WRITE(6,*)'ENTER NEW VALUE'
            READ(5,*)TOL_CS(3)
         ELSE IF(NUMP .EQ. 7)THEN
            WRITE(6,*)
            WRITE(6,*)'DEL O-18 TOLERANCE :',TOL_CS(4)
            WRITE(6,*)'ENTER NEW VALUE'
            READ(5,*)TOL_CS(4)
         ELSE IF(NUMP .EQ. 8)THEN
            WRITE(6,*)
            WRITE(6,*)'CHLORIDE CONC :',CONC_CL_S(1)
            WRITE(6,*)'NOTE: 6.1 IS SATURATION'
            WRITE(6,*)'ENTER NEW VALUE'
            READ(5,*)CONC_CL_S(1)
         ELSE
            WRITE(6,*)
            WRITE(6,*)'GET A CLUE KELP BREATH'
            WRITE(6,*)'PICK A NUMBER BETWEEN 1 AND 8'
            WRITE(6,*)
            GO TO 380
         ENDIF
370     CONTINUE
      ENDIF

************************************
** WRITE NEW VALUES TO INPUT FILE **
************************************

      REWIND 89
      WRITE(89,*)ITMAX_CS,NCS,(X_CS(I),I=1,NCS),DTMAX_CS,
```

```
     +       DTMIN_CS,(TOL_CS(I),I=1,NCS),CONC_CL_C(1),
     +       CONC_CL_S(1)
        CLOSE(UNIT=89)

***************************************************
** CALCULATE MOLES OF CHLORIDE FROM CONC AND VOL **
***************************************************

        IF(X_CS(3) .LE. 0.D0)THEN
           CL_S(1)=0.D0
           CONC_CL_S(1)=0.D0
        ELSE
           CL_S(1)=CONC_CL_S(1)*(X_CS(3)*1.D3)
        ENDIF

*****************************************************************
** TOTAL MASS OF CHLORIDE DEPOSITED IN SEARLES AT START OF RUN **
*****************************************************************

        WRITE(6,*)
        WRITE(6,*)'ENTER THE MASS OF CL DEPOSITED IN SEARLES LAKE'
        WRITE(6,*)'AT THE BEGINNING OF THIS RUN (KG/M^2)'
        WRITE(6,*)
        READ(5,*)SUM_SCL_DEP(1)

***********************************************************************
** A CHANCE TO ADJUST INPUT PARAMETERS FOR PANAMINT AND DEATH VALLEY **
***********************************************************************

        WRITE(6,*)
        WRITE(6,*)'THE CURRENT PANAMINT LAKE PARAMETERS ARE:'
        WRITE(6,*)'   1 : SURFACE AREA =',AREA_P
        WRITE(6,*)'   2 : CHLORIDE CONC=',CONC_CL_P(1)
        WRITE(6,*)'   3 : CHLORIDE DEPOSITED (KG/M^2)',SUM_PCL_DEP(1)

        WRITE(6,*)
        WRITE(6,*)'DO YOU WANT TO CHANGE ANY OF THE STARTING PARAMETERS ?'
        WRITE(6,*)'1=YES    0=NO'
        READ(5,*)ISEE
        WRITE(6,*)

        IF(ISEE .EQ. 1)THEN

           WRITE(6,*)
           WRITE(6,*)'HOW MANY PARAMETERS WOULD YOU LIKE TO CHANGE ?'
           READ(5,*)K
           WRITE(6,*)

           DO 400 I = 1,K
              WRITE(6,*)
 410          WRITE(6,*)'ENTER THE PARAMETER NUMBER'
              WRITE(6,*)
              WRITE(6,*)'   1 : SURFACE AREA'
              WRITE(6,*)'   2 : CHLORIDE CONC'
              WRITE(6,*)'   3 : CHLORIDE DEPOSITED (KG/M^2)'
              READ(5,*)NUMP
              IF(NUMP .EQ. 1)THEN
                 WRITE(6,*)
                 WRITE(6,'(A,D15.7)')' SURFACE AREA =',AREA_P
                 WRITE(6,*)'NOTE: MAX SURFACE AREA IS 0.727D9'
                 WRITE(6,*)'ENTER NEW VALUE'
                 READ(5,*)AREA_P
              ELSE IF(NUMP .EQ. 2)THEN
                 WRITE(6,*)
                 WRITE(6,*)'CHLORIDE CONC :',CONC_CL_P(1)
                 WRITE(6,*)'NOTE: 6.1 IS SATURATION'
```

```
            WRITE(6,*)'ENTER NEW VALUE'
            READ(5,*)CONC_CL_P(1)
         ELSE IF(NUMP .EQ. 3)THEN
            WRITE(6,*)
            WRITE(6,*)'SUM OF CHLORIDE DEPOSITED',SUM_PCL_DEP(1)
            WRITE(6,*)'ENTER NEW VALUE'
            READ(5,*)SUM_PCL_DEP(1)
         ELSE
            WRITE(6,*)
            WRITE(6,*)'EVERY DAY MUST BE MONDAY FOR SOMEONE LIKE
     +YOU'
            WRITE(6,*)'PICK A NUMBER BETWEEN 1 AND 8'
            WRITE(6,*)
            GO TO 410
         ENDIF
400      CONTINUE
      ENDIF

***************************************************
** CALCULATE MOLES OF CHLORIDE FROM CONC AND VOL **
***************************************************

      IF(CONC_CL_P(1) .LE. 0.D0)THEN
         CL_P(1)=0.D0
      ELSE
         VOL_P=PVOL(AREA_P)
         CL_P(1)=CONC_CL_P(1)/(VOL_P*1.D3)
      ENDIF

************************
** DEATH VALLEY STUFF **
************************

      WRITE(6,*)
      WRITE(6,*)'THE CURRENT DEATH VALLEY PARAMETERS ARE:'
      WRITE(6,*)'   1 : SURFACE AREA =',AREA_D
      WRITE(6,*)'   2 : CHLORIDE CONC=',CONC_CL_D(1)
      WRITE(6,*)'   3 : CHLORIDE DEPOSITED (KG/M^2)',SUM_DCL_DEP(1)

      WRITE(6,*)
      WRITE(6,*)'DO YOU WANT TO CHANGE ANY OF THE STARTING PARAMETERS ?'
      WRITE(6,*)'1=YES    0=NO'
      READ(5,*)ISEE
      WRITE(6,*)

      IF(ISEE .EQ. 1)THEN

         WRITE(6,*)
         WRITE(6,*)'HOW MANY PARAMETERS WOULD YOU LIKE TO CHANGE ?'
         READ(5,*)K
         WRITE(6,*)

         DO 420 I = 1,K
            WRITE(6,*)
 430        WRITE(6,*)'ENTER THE PARAMETER NUMBER'
            WRITE(6,*)
            WRITE(6,*)'   1 : SURFACE AREA'
            WRITE(6,*)'   2 : CHLORIDE CONC'
            WRITE(6,*)'   3 : CHLORIDE DEPOSITED (KG/M^2)'
            READ(5,*)NUMP
            IF(NUMP .EQ. 1)THEN
               WRITE(6,*)
               WRITE(6,'(A,D15.7)')' SURFACE AREA =',AREA_D
               WRITE(6,*)'NOTE: MAX SURFACE AREA IS 0.583D9'
               WRITE(6,*)'ENTER NEW VALUE'
               READ(5,*)AREA_D
```

```
            ELSE IF(NUMP .EQ. 2)THEN
               WRITE(6,*)
               WRITE(6,*)'CHLORIDE CONC :',CONC_CL_D(1)
               WRITE(6,*)'NOTE: 6.1 IS SATURATION'
               WRITE(6,*)'ENTER NEW VALUE'
               READ(5,*)CONC_CL_D(1)
            ELSE IF(NUMP .EQ. 3)THEN
               WRITE(6,*)
               WRITE(6,*)'SUM OF CHLORIDE DEPOSITED',SUM_DCL_DEP(1)
               WRITE(6,*)'ENTER NEW VALUE'
               READ(5,*)SUM_DCL_DEP(1)
            ELSE
               WRITE(6,*)
               WRITE(6,*)'DON''T GIVE UP YOUR DAY JOB'
               WRITE(6,*)'PICK A NUMBER BETWEEN 1 AND 8'
               WRITE(6,*)
               GO TO 430
            ENDIF
420      CONTINUE

***************************************************
** CALCULATE MOLES OF CHLORIDE FROM CONC AND VOL **
***************************************************

        IF(CONC_CL_D(1) .LE. 0.D0)THEN
           CL_D(1)=0.D0
        ELSE
           VOL_D=DVOL(AREA_D)
           CL_D(1)=CONC_CL_D(1)*(VOL_D*1.D3)
        ENDIF

************************************
** WRITE NEW VALUES TO INPUT FILE **
************************************

           REWIND 69
           WRITE(69,*)AREA_P,CONC_CL_P(1),SUM_PCL_DEP(1),AREA_D,
     +          CONC_CL_D(1),SUM_DCL_DEP(1)
           CLOSE(UNIT=69)
        ENDIF

******************************************************************************
*             LET'S CHOOSE AN INFLOW FUNCTION AND TIME INTERVAL              *
******************************************************************************


        WRITE(6,*)
        WRITE(6,*)
115     WRITE(6,*)'WHICH INFLOW FUNCTION WOULD YOU LIKE FOR OWENS LAKE?'
        WRITE(6,*)'1=LINEAR'
        WRITE(6,*)'2=EXPONENTIAL'
        WRITE(6,*)'3=LOGARITHMIC'
        WRITE(6,*)'4=POWER'
        WRITE(6,*)'5=SINUSOIDAL'
        WRITE(6,*)'6=STEP'
        WRITE(6,*)'7=ZERO INFLOW'
        WRITE(6,*)'8=STEADY-STATE HISTORY, VARIABLE TEMP'
        WRITE(6,*)'9=STEADY-STATE HISTORY, CONSTANT HIGH TEMP'
        WRITE(6,*)'10=STEADY-STATE HISTORY, CONSTANT LOW TEMP'
        READ(5,*)INCHOICE

        WRITE(6,*)
        WRITE(6,*)
        IF(INCHOICE .EQ. 1)THEN
           WRITE(6,*)'YOU HAVE CHOSEN F(QI)= A*X+B'
           WRITE(6,*)'ENTER A VALUE FOR ''A'', AND ''B'''
```

```
      ELSE IF(INCHOICE .EQ. 2)THEN
         WRITE(6,*)'YOU HAVE CHOSEN F(QI)= B*EXP(A*X)'
         WRITE(6,*)'ENTER A VALUE FOR ''A'', AND ''B'''
      ELSE IF(INCHOICE .EQ. 3)THEN
         WRITE(6,*)'YOU HAVE CHOSEN F(QI)= B+A*LOG(X)'
         WRITE(6,*)'ENTER A VALUE FOR ''A'', AND ''B'''
      ELSE IF(INCHOICE .EQ. 4)THEN
         WRITE(6,*)'YOU HAVE CHOSEN F(QI)= B+A*(X**C)'
         WRITE(6,*)'ENTER VALUES FOR ''A'',''B'', AND ''C'''
      ELSE IF(INCHOICE .EQ. 5)THEN
         WRITE(6,*)'YOU HAVE CHOSEN F(QI)= B+A*SIN(C*X)'
         WRITE(6,*)'ENTER VALUES FOR ''A'',''B'', AND ''C'''
      ELSE IF(INCHOICE .EQ. 6)THEN
         WRITE(6,*)'YOU HAVE CHOSEN F(QI)= B+(A*B)'
         WRITE(6,*)'ENTER A VALUE FOR ''A'', AND ''B'''
      ELSE IF(INCHOICE .EQ. 7)THEN
         WRITE(6,*)'YOU HAVE CHOSEN ZERO INFLOW'
         WRITE(6,*)'GRAB YOUR CANTEEN AND HEAD FOR THE SHADE'
      ELSE IF(INCHOICE .EQ. 8)THEN
         WRITE(6,*)'YOU HAVE CHOSEN STEADY STATE "GUESS"'
         WRITE(6,*)'VARIABLE TEMPERATURE HISTORY'
      ELSE IF(INCHOICE .EQ. 9)THEN
         WRITE(6,*)'YOU HAVE CHOSEN STEADY STATE "GUESS"'
         WRITE(6,*)'CONSTANT HIGH TEMPERATURE HISTORY'
      ELSE IF(INCHOICE .EQ. 10)THEN
         WRITE(6,*)'YOU HAVE CHOSEN STEADY STATE "GUESS"'
         WRITE(6,*)'CONSTANT LOW TEMPERATURE HISTORY'
      ELSE
         WRITE(6,*)
         WRITE(6,*)
         WRITE(6,*)'NOT A VALID CHOICE MULLET-HEAD'
         WRITE(6,*)
         WRITE(6,*)
         GOTO 115
      ENDIF

      IF(INCHOICE .EQ. 8)THEN

         WRITE(6,*)'READING INFLOW HISTORY'
         OPEN(UNIT=29,FILE='CASEA1.DAT',STATUS='OLD')
         DO 450 I=1,1000
            READ(29,*,END=451)QITIME(I),F2,QIHIST(I)
            QITIME(I)=QITIME(I)*1.D6
450      CONTINUE
451      WRITE(6,*)'READ',I-1,' POINTS FROM INFLOW HISTORY'
         NQPTS=I-1

      ELSEIF(INCHOICE .EQ. 9)THEN

         WRITE(6,*)'READING INFLOW HISTORY'
         OPEN(UNIT=28,FILE='CASEB1.DAT',STATUS='OLD')
         DO 460 I=1,1000
            READ(28,*,END=461)QITIME(I),F2,QIHIST(I)
            QITIME(I)=QITIME(I)*1.D6
460      CONTINUE
461      WRITE(6,*)'READ',I-1,' POINTS FROM INFLOW HISTORY'
         NQPTS=I-1

      ELSEIF(INCHOICE .EQ. 10)THEN

         WRITE(6,*)'READING INFLOW HISTORY'
         OPEN(UNIT=27,FILE='CASEC1.DAT',STATUS='OLD')
         DO 470 I=1,1000
            READ(27,*,END=471)QITIME(I),F2,QIHIST(I)
            QITIME(I)=QITIME(I)*1.D6
470      CONTINUE
```

```
 471      WRITE(6,*)'READ',I-1,' POINTS FROM INFLOW HISTORY'
          NQPTS=I-1

       ELSEIF(INCHOICE .EQ. 4 .OR. INCHOICE .EQ. 5)THEN
          READ(5,*)A,B,C
       ELSE IF(INCHOICE .NE. 7)THEN
          READ(5,*)A,B
       ENDIF

       WRITE(6,*)
       WRITE(6,*)'ENTER STARTING TIME AND ENDING TIME'
       WRITE(6,*)'0 CORRESPONDS TO PRESENT, 2.0E6 IS 2 MILLION YRS AGO'
       WRITE(6,*)
       WRITE(6,*)
       WRITE(6,*)'MANAGEMENT ACCEPTS NO RESPONSIBILITY FOR PEOPLE WHO'
       WRITE(6,*)'RUN THE MODEL BACKWARD IN TIME'
       WRITE(6,*)
       WRITE(6,*)
       READ(5,*)TBEG, TEND
       WRITE(6,*)

********************************
** CONSTANT PARAMETER OPTION **
********************************

       CPARAM=.FALSE.
       WRITE(6,*)
       WRITE(6,*)'DO YOU WANT TO RUN THE PROGRAM WITH CONSTANT PARAMETE
     +RS ?'
       WRITE(6,*)'1=YES   0=NO'
       WRITE(6,*)
       READ(5,*)INPARAM

       IF(INPARAM .EQ. 1)THEN
          CPARAM=.TRUE.

 490      WRITE(6,*)
          WRITE(6,*)'WOULD YOU LIKE UPPER LIMIT, LOWER LIMIT, OR CUSTOM
     + '
          WRITE(6,*)'1 = UPPER LIMIT'
          WRITE(6,*)'2 = LOWER LIMIT'
          WRITE(6,*)'3 = CUSTOM'
          WRITE(6,*)
          READ(5,*)LIMCHOICE

          IF(LIMCHOICE .EQ. 1)THEN
             OPEN(UNIT=68,FILE='UPPER.INP',STATUS='OLD')
             WRITE(6,*)'READING CONSTANT PARAMETERS'
             WRITE(6,*)
             READ(68,*)TEMPC,TEMPC_C,TEMPC_S,
     +            PRECIPC,PRECIPC_C,PRECIPC_S,EVAPC,EVAPC_C,EVAPC_S,
     +            EVAPC_P,EVAPC_D,DELAC,DELAC_C,DELAC_S,DELIC,DELPC,
     +            DELPC_C,DELPC_S
             CLOSE(UNIT=68)
          ELSE IF(LIMCHOICE .EQ. 2)THEN
             OPEN(UNIT=67,FILE='LOWER.INP',STATUS='OLD')
             WRITE(6,*)'READING CONSTANT PARAMETERS'
             WRITE(6,*)
             READ(67,*)TEMPC,TEMPC_C,TEMPC_S,
     +            PRECIPC,PRECIPC_C,PRECIPC_S,EVAPC,EVAPC_C,EVAPC_S,
     +            EVAPC_P,EVAPC_D,DELAC,DELAC_C,DELAC_S,DELIC,DELPC,
     +            DELPC_C,DELPC_S
             CLOSE(UNIT=67)
          ELSE IF(LIMCHOICE .EQ. 3)THEN

***************************
```

```
** PICK A TEMP, ANY TEMP **
****************************

              WRITE(6,*)
              WRITE(6,*)'ENTER THE TEMP FOR OWENS (DEGREES C)'
              WRITE(6,*)
              READ(5,*)TEMPC
              WRITE(6,*)

              WRITE(6,*)
              WRITE(6,*)'ENTER THE TEMP FOR CHINA (DEGREES C)'
              WRITE(6,*)
              READ(5,*)TEMPC_C
              WRITE(6,*)

              WRITE(6,*)
              WRITE(6,*)'ENTER THE TEMP FOR SEARLES (DEGREES C)'
              WRITE(6,*)
              READ(5,*)TEMPC_S
              WRITE(6,*)

*****************************
** CONSTANT PRECIPITATION **
*****************************

              WRITE(6,*)
              WRITE(6,*)'ENTER THE PRECIPITATION FOR OWENS(METERS/YR)'
              WRITE(6,*)
              READ(5,*)PRECIPC
              WRITE(6,*)

              WRITE(6,*)
              WRITE(6,*)'ENTER THE PRECIPITATION FOR CHINA(METERS/YR)'
              WRITE(6,*)
              READ(5,*)PRECIPC_C
              WRITE(6,*)

              WRITE(6,*)
              WRITE(6,*)'ENTER THE PRECIPITATION FOR SEARLES(METERS/YR)'
              WRITE(6,*)
              READ(5,*)PRECIPC_S
              WRITE(6,*)

***************************
** CONSTANT EVAPORATION **
***************************

              WRITE(6,*)
              WRITE(6,*)'ENTER THE EVAPORATION FOR OWENS(METERS/YR)'
              WRITE(6,*)
              READ(5,*)EVAPC
              WRITE(6,*)

              WRITE(6,*)
              WRITE(6,*)'ENTER THE EVAPORATION FOR CHINA(METERS/YR)'
              WRITE(6,*)
              READ(5,*)EVAPC_C
              WRITE(6,*)

              WRITE(6,*)
              WRITE(6,*)'ENTER THE EVAPORATION FOR SEARLES(METERS/YR)'
              WRITE(6,*)
              READ(5,*)EVAPC_S
              WRITE(6,*)

              WRITE(6,*)
```

```
          WRITE(6,*)'ENTER THE EVAPORATION FOR PANAMINT(METERS/YR)'
          WRITE(6,*)
          READ(5,*)EVAPC_P
          WRITE(6,*)

          WRITE(6,*)
          WRITE(6,*)'ENTER THE EVAPORATION FOR DEATH VALLEY(METERS/Y
     +R)'
          WRITE(6,*)
          READ(5,*)EVAPC_D
          WRITE(6,*)

******************************
** CONSTANT DEL ATMOSPHERE **
******************************

          WRITE(6,*)
          WRITE(6,*)'ENTER DEL ATMOSPHERE FOR OWENS (PER MIL)'
          WRITE(6,*)
          READ(5,*)DELAC
          WRITE(6,*)

          WRITE(6,*)
          WRITE(6,*)'ENTER DEL ATMOSPHERE FOR CHINA (PER MIL)'
          WRITE(6,*)
          READ(5,*)DELAC_C
          WRITE(6,*)

          WRITE(6,*)
          WRITE(6,*)'ENTER DEL ATMOSPHERE FOR SEARLES (PER MIL)'
          WRITE(6,*)
          READ(5,*)DELAC_S
          WRITE(6,*)

**************************
** CONSTANT DEL INFLOW  **
**************************

          WRITE(6,*)
          WRITE(6,*)'ENTER DEL INFLOW FOR OWENS (PER MIL)'
          WRITE(6,*)
          READ(5,*)DELIC
          WRITE(6,*)

**************************
** CONSTANT DEL PRECIP  **
**************************

          WRITE(6,*)
          WRITE(6,*)'ENTER DEL PRECIPITATION FOR OWENS (PER MIL)'
          WRITE(6,*)
          READ(5,*)DELPC
          WRITE(6,*)

          WRITE(6,*)
          WRITE(6,*)'ENTER DEL PRECIPITATION FOR CHINA (PER MIL)'
          WRITE(6,*)
          READ(5,*)DELPC_C
          WRITE(6,*)

          WRITE(6,*)
          WRITE(6,*)'ENTER DEL PRECIPITATION FOR SEARLES (PER MIL)'
          WRITE(6,*)
          READ(5,*)DELPC_S
          WRITE(6,*)
```

```
          ELSE
             WRITE(6,*)'NO,NO,NOOOOOOO...'
             WRITE(6,*)'PICK 1,2, OR 3 '
             WRITE(6,*)
             GO TO 490
          ENDIF

          OPEN(UNIT=80,FILE='HUMHIST.DAT',STATUS='OLD')
          DO 1750 I=1,400
             READ(80,*,END=1751)HUMTIME(I),SHUM(I)
             OHUM(I)=SHUM(I)
             CHUM(I)=SHUM(I)
1750      CONTINUE

1751      NHPTS=I-1
          WRITE(6,*)'READ',NHPTS,' FROM HUMHIST.DAT'
          CLOSE(80)

          DO 1760 I=1,NHPTS
             HUMTIME(I)=HUMTIME(I)*1.D6
1760      CONTINUE

        ENDIF

********************
** GRAPHICS STUFF **
********************

        GRAF=.FALSE.

        WRITE(6,*)
        WRITE(6,*)'DO YOU WANT TO PLOT THIS RUN ON THE SCREEN'
        WRITE(6,*)'1=YES    0=NO'
        READ(5,*)IGRAF
        WRITE(6,*)

        IF(IGRAF .EQ. 1)THEN
           GRAF=.TRUE.
        ENDIF

****************************************************
** SAVE VALUES FOR MODEL STARTUP AT A GIVEN TIME **
****************************************************

        SU=.FALSE.

        WRITE(6,*)
        WRITE(6,*)'DO YOU WANT TO SAVE INFO TO RESTART THE MODEL'
        WRITE(6,*)'AT A SPECIFIC TIME'
        WRITE(6,*)' 1=YES    0=NO'
        WRITE(6,*)

        READ(5,*)ISU

        IF(ISU .EQ. 1)THEN
           SU=.TRUE.
           WRITE(6,*)
           WRITE(6,*)'HOW MANY STARTUP TIMES DO YOU WISH ?'
           WRITE(6,*)
           READ(5,*)NUMST

           DO 1900 I=1,NUMST
              WRITE(6,*)
              WRITE(6,*)'ENTER STARTUP TIME'
              READ(5,*)SUT(I)
              WRITE(6,*)
```

```
 1900     CONTINUE

          SAVETIME=SUT(1)
          ISTCNT=1

       ENDIF



********************************************************************************
*     ALL WE'RE DOING HERE IS READING DATA FILES, THE GOOD STUFF IS LATER      *
********************************************************************************

       IF(.NOT. CPARAM)THEN

          WRITE(6,*)
          WRITE(6,*)
          WRITE(6,*)'READING DATA FILES, GOOD TIME TO GET MUNCHIES'
          WRITE(6,*)
          WRITE(6,*)

          OPEN(UNIT=20,FILE='OWENS.CLIM_UF',STATUS='OLD',
     +          FORM='UNFORMATTED')

          DO 500 I = 1, 2000
             READ (20,END=501) WTIME(I),OTEMP(I),OEVAP(I),GBG1,
     +          OPRECIP(I)
 500      CONTINUE

 501      NPTS=I-1

          WRITE(6,*)'READ',5*NPTS,' DATA POINTS FROM OWENS.CLIM'

          DO 600 I = 1,NPTS
             WTIME(I)=WTIME(I)*1.0D6
 600      CONTINUE


          CLOSE (UNIT=20)

          OPEN(UNIT=40,FILE='SEARLES.CLIM_UF',STATUS='OLD',
     +          FORM='UNFORMATTED')

          DO 800 I = 1, NPTS
             READ (40) STEMP(I),SEVAP(I),GBG2,SPRECIP(I)
 800      CONTINUE

          WRITE(6,*)'READ',4*NPTS,' DATA POINTS FROM SEARLES.CLIM'

          CLOSE (UNIT=40)

          DO 850 I=1,NPTS
             CTEMP(I)=STEMP(I)
             CEVAP(I)=SEVAP(I)
             CPRECIP(I)=SPRECIP(I)
 850      CONTINUE

          WRITE(6,*)'READ CHINA VALUES FROM SEARLES VALUES'

          OPEN(UNIT=21,FILE='ODELP.DAT_UF',STATUS='OLD',FORM='UNFORMATTED')
          DO 900 I = 1, NPTS
             READ(21)ODELP(I)
 900      CONTINUE

          WRITE(6,*)'READ',NPTS,' DATA POINTS FROM ODELP.DAT'
```

```
      CLOSE(UNIT=21)

C     WRITE(6,*)'READ',NPTS,' DATA POINTS FROM CDELP.DAT'

C     CLOSE(UNIT=31)

      OPEN(UNIT=41,FILE='SDELP.DAT_UF',STATUS='OLD',FORM='UNFORMATTED')
      DO 1100 I = 1, NPTS
         READ(41)SDELP(I)
1100  CONTINUE

      WRITE(6,*)'READ',NPTS,' DATA POINTS FROM SDELP.DAT'

      CLOSE(UNIT=41)

      DO 1150 I=1,NPTS
         CDELP(I)=SDELP(I)
 1150 CONTINUE

      WRITE(6,*)'READ CDELP DATA'

      OPEN(UNIT=22,FILE='ODELA.DAT_UF',STATUS='OLD',FORM='UNFORMATTED')
      DO 1200 I = 1, NPTS
         READ(22) ODELA(I)
1200  CONTINUE

      WRITE(6,*)'READ',NPTS,' DATA POINTS FROM ODELA.DAT'

      CLOSE(UNIT=22)

      OPEN(UNIT=32,FILE='CSDELA.DAT_UF',STATUS='OLD',FORM='UNFORMATTED')
      DO 1300 I = 1, NPTS
         READ(32) CSDELA(I)
1300  CONTINUE

      WRITE(6,*)'READ',NPTS,' DATA POINTS FROM CSDELA.DAT'

      CLOSE(UNIT=32)

      OPEN(UNIT=23,FILE='ODELI.DAT_UF',STATUS='OLD',FORM='UNFORMATTED')
      DO 1400 I = 1,NPTS
         READ(23)ODELI(I)
1400  CONTINUE

      WRITE(6,*) 'READ',NPTS,' DATA POINTS FROM ODELI.DAT'

      CLOSE(UNIT=23)

      OPEN(UNIT=85,FILE='PEVAP.DAT_UF',STATUS='OLD',FORM='UNFORMATTED')
      DO 1500 I = 1,NPTS
         READ(85)PEVAP(I)
1500  CONTINUE

      WRITE(6,*) 'READ',NPTS,' DATA POINTS FROM PEVAP.DAT'

      CLOSE(UNIT=85)

      OPEN(UNIT=84,FILE='DEVAP.DAT_UF',STATUS='OLD',FORM='UNFORMATTED')
      DO 1600 I = 1,NPTS
         READ(84)DEVAP(I)
1600  CONTINUE

      WRITE(6,*) 'READ',NPTS,' DATA POINTS FROM DEVAP.DAT'

      CLOSE(UNIT=84)
```

```
          OPEN(UNIT=80,FILE='HUMHIST.DAT',STATUS='OLD')
          DO 1700 I=1,400
             READ(80,*,END=1701)HUMTIME(I),SHUM(I)
             OHUM(I)=SHUM(I)
             CHUM(I)=SHUM(I)
 1700     CONTINUE

 1701     NHPTS=I-1
          WRITE(6,*)'READ',NHPTS,' FROM HUMHIST.DAT'
          CLOSE(UNIT=80)

          DO 1800 I=1,NHPTS
             HUMTIME(I)=HUMTIME(I)*1.D6
1800      CONTINUE

        ENDIF


****************************************
** CONVERT DEL DOLOMITE TO DEL WATER **
****************************************

          IF(CPARAM)THEN
             X(2)= W_DEL(X(2),TEMPC)
             X_CS(2)=W_DEL(X_CS(2),TEMPC_C)
             X_CS(4)=W_DEL(X_CS(4),TEMPC_S)
          ELSE
             CALL FINDT(NPTS,TBEG,NDX,FOUND,WTIME)
             TOTEMP=(OTEMP(NDX+1)-OTEMP(NDX))/(WTIME(NDX+1)-
     +            WTIME(NDX))*(TBEG-WTIME(NDX))+OTEMP(NDX)
             TSTEMP=(STEMP(NDX+1)-STEMP(NDX))/(WTIME(NDX+1)-
     +            WTIME(NDX))*(TBEG-WTIME(NDX))+STEMP(NDX)
             TCTEMP=(CTEMP(NDX+1)-CTEMP(NDX))/(WTIME(NDX+1)-
     +            WTIME(NDX))*(TBEG-WTIME(NDX))+CTEMP(NDX)

             X(2)=W_DEL(X(2),TOTEMP)
             X_CS(2)=W_DEL(X_CS(2),CTEMP(1))
             X_CS(4)=W_DEL(X_CS(4),STEMP(1))
          ENDIF

****************************
** START THE BALL ROLLING **
****************************

        CALL RKF(N,X,TBEG,TEND,TOL,DTMAX,DTMIN,ITMAX)

*       WRITE(99,*)
*       WRITE(96,*)
*       WRITE(95,*)
*       CLOSE(UNIT=99)
*       CLOSE(UNIT=97)
*       CLOSE(UNIT=96)
*       CLOSE(UNIT=95)

*********************************************
** CALL SOLVER FOR CHINA AND SEARLES LAKE **
*********************************************


        GUESS=.TRUE.
        GUESS2=.TRUE.

        CALL RKF_CS(NCS,X_CS,TBEG,TEND,TOL_CS,DTMAX_CS,
     +       DTMIN_CS,ITMAX_CS)
```

```
      END

      SUBROUTINE RKF(N,X,TBEG,TEND,TOL,DTMAX,DTMIN,ITMAX)
******************************************************************************
*     SOLVE A SYSTEM OF PARTIAL DIFFERENTIAL EQUATION OF THE FORM:          *
*                       F(T,X)= X'                                           *
*      BETWEEN T1,T2, GIVEN THE INITIAL CONDITION XO(T1)                     *
******************************************************************************
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      PARAMETER(NUMA=15000,NUMB=2500)
      PARAMETER (VOLMAX=30.02D9,QCL_IN=1.67D8)

      LOGICAL PASS,GRAF,ONLY1,ZEROVOL,ZEROCHK,SU

      COMMON/BOTH/CL(NUMA),TSOD(10),TCL(10),OTIME(NUMA),TCO3(10),
     + OQO(NUMA),QO(10),CONC_CL(NUMA),DOLTEMP,DELDOL,TODELI,TOTEMP,
     + ODEL_OUT(NUMA),OCL_OUT(NUMA),PEVAP(NUMB),DEVAP(NUMB),
     + SUM_PCL_DEP(NUMA),SUM_DCL_DEP(NUMA),AREA_P,AREA_D,
     + AREA(NUMA),SOAREA,ALLAREA(NUMA),CONC_CL_P(NUMA),CONC_CL_D(NUMA),
     + CL_P(NUMA),CL_D(NUMA),INCHOICE

      COMMON/BOTH2/CL_C(NUMA),TSOD_C(10),TCL_C(10),TIME,TCO3_C(10),
     + CQO(10),GRAF,CONC_CL_C(NUMA),DOLTEMP_C,DELDOL_C,
     + TCDELI,TCTEMP,CDEL_OUT(NUMA),CCL_OUT(NUMA),CL_S(NUMA),NOPTS,
     + TSOD_S(10),TCL_S(10),TCO3_S(10),SQO(10),CONC_CL_S(NUMA),
     + DOLTEMP_S,DELDOL_S,TSDELI,TSTEMP,SCL_DEP(NUMA),TTLCL_IN,NPTS,
     + SUM_SCL_DEP(NUMA),QI_C,QI_S,PQI

      COMMON/HIST/QIHIST(1000),HUMTIME(500),NHPTS,SU,SUT(15),
     + SAVETIME,NUMST,QITIME(1000),NQPTS

      COMMON/FINAL/FINDEL,FINVOL,FINAREA,FINCL

      REAL XMIN,XMAX,YMIN,YMAX,YDEL(NUMA),XPLT(NUMA),YQI(NUMA),
     + YAREA(NUMA),YTEMP(NUMA)

      DIMENSION X(3),RK1(3),RK2(3),RK3(3),RK4(3),RK5(3),RK6(3),R(3)
      DIMENSION TERM(3),DEL(3),TOL(3)


*     OPEN(UNIT=99,FILE='DEL.OUT',STATUS='NEW',CARRIAGE CONTROL=
*    +     'LIST')
*     OPEN(UNIT=97,FILE='OQO.OUT',STATUS='NEW',CARRIAGE CONTROL=
*    +     'LIST')
*     OPEN(UNIT=96,FILE='AREA.OUT',STATUS='NEW',CARRIAGE CONTROL=
*    +     'LIST')
      OPEN(UNIT=95,FILE='QI.OUT',STATUS='NEW',CARRIAGE CONTROL=
     +     'LIST')
      OPEN(UNIT=70,FILE='START.OUT',STATUS='UNKNOWN',CARRIAGE CONTROL=
     +     'LIST')
10    FORMAT(A,D12.5)

** FOR GRAPH SCALING **
      XMIN=TBEG
      XMAX=TEND

      STEP=DTMAX
      KOUNT=1
      OTIME(KOUNT)=TBEG
      ODEL_OUT(KOUNT)=X(2)
      OCL_OUT(KOUNT)=0.D0
      ONLY1=.TRUE.

**********************************
** TO PROTECT YOU FROM DRYNESS **
```

```
**********************************

        ZEROVOL=.FALSE.
        ZEROCHK=.FALSE.

**************************************
** THE SOLVING ROUTINE BEGINS HERE **
**************************************

*********************************************
** INITIALIZE PARAMETERS TO RESTART MODEL **
*********************************************

        SAVETIME=SUT(1)
        ISTCNT=1

        WRITE(6,*)
        WRITE(6,*)
        WRITE(6,*)'START SOLVING DIFFERENTIAL EQUATIONS FOR OWENS'
        WRITE(6,*)
        WRITE(6,*)

        DO 100 ITER=1,ITMAX
           IF(OTIME(KOUNT).GT.TEND)THEN
              KNT=1
              T=OTIME(KOUNT)
              DO 200 I=1,N
                 RK1(I)=STEP*F(I,X,T,KNT,KOUNT,TBEG,TEND)
200           CONTINUE

*********************************************
** STORE INITIAL VALUES IN GRAPHICS ARRAY **
*********************************************

              IF(ONLY1)THEN
                 YDEL(1)=DELDOL
                 XPLT(1)=TBEG
                 YAREA(1)=OAREA(X(1))

                 CALL FINDQT(NQPTS,OTIME(KOUNT),IQNDEX,QITIME)
                 IF(INCHOICE .GE. 8)THEN
                    CALL QINTERP(OTIME(KOUNT),IQNDEX,QI,QIHIST,
     +                   QITIME)
                 ELSE
                    QI=FQI(TBEG-OTIME(KOUNT))
                 ENDIF
                 IF(QI .LE. 0.D0)QI=0.D0
                 YQI(1)=FQI(0.D0)

                 YTEMP(1)=TOTEMP
                 ONLY1=.FALSE.
              ENDIF

************************************************************************
** "TERM" IS AN ARRAY WHICH STORES APPROXIMATIONS OF VOL AND DEL O-18 **
** WHICH WILL BE USED IN FINAL CALCULATIONS IF ERRORS WITHIN THE STEP **
** ARE LESS THAN THE GIVEN TOLERANCES.                                **
************************************************************************

              T=OTIME(KOUNT)-STEP/4.D0
              KNT=2
              DO 300 I=1,N
                 TERM(I)=X(I)+ RK1(I)/4.D0
300           CONTINUE

*****************************************************************************
```

```
** WHEN BOTH "IF'S" ARE SATISFIED YOU ARE AS CLOSE TO ZERO VOLUME AS THE  **
** SOLVER IS GOING TO GET WITH THE GIVEN CONSTRAINTS, SO GO TO THE END OF **
** THE RKF AND MAKE VOLUME=0 AND DEL=DELINFLOW                             **
****************************************************************************

              IF(TERM(1) .LE. 10.D0)THEN
                 IF(STEP .EQ. DTMIN)THEN
                    ZEROVOL=.TRUE.
                    PASS=.TRUE.
                    GOTO 1375
                 ELSE
                    ZEROCHK=.TRUE.
                 ENDIF
              ENDIF
              DO 400 I=1,N
                 RK2(I)=STEP*F(I,TERM,T,KNT,KOUNT,TBEG,TEND)
400           CONTINUE
              T=OTIME(KOUNT)-3.D0*STEP/8.D0
              KNT=3
              DO 500 I=1,N
                 TERM(I)=X(I)+(3.D0*RK1(I)+9.D0*RK2(I))/32.D0
500           CONTINUE
              IF(TERM(1) .LE. 10.D0)THEN
                 IF(STEP .EQ. DTMIN)THEN
                    ZEROVOL=.TRUE.
                    PASS=.TRUE.
                    GOTO 1375
                 ELSE
                    ZEROCHK=.TRUE.
                 ENDIF
              ENDIF
              DO 600 I=1,N
                 RK3(I)=STEP*F(I,TERM,T,KNT,KOUNT,TBEG,TEND)
600           CONTINUE
              T=OTIME(KOUNT)-12.D0*STEP/13.D0
              KNT=4
              DO 700 I=1,N
                 TERM(I)=X(I) + (1932.D0*RK1(I)-7200.D0*RK2(I)+
     +                7296.D0*RK3(I))/2197.D0
700           CONTINUE
              IF(TERM(1) .LE. 10.D0)THEN
                 IF(STEP .EQ. DTMIN)THEN
                    ZEROVOL=.TRUE.
                    PASS=.TRUE.
                    GOTO 1375
                 ELSE
                    ZEROCHK=.TRUE.
                 ENDIF
              ENDIF
              DO 800 I=1,N
                 RK4(I)=STEP*F(I,TERM,T,KNT,KOUNT,TBEG,TEND)
800           CONTINUE
              T=OTIME(KOUNT)-STEP
              KNT=5
              DO 900 I=1,N
                 TERM(I)=X(I) +439.D0*RK1(I)/216.D0-
     +                8.D0*RK2(I)+3680.D0*RK3(I)/513.D0-
     +                845.D0*RK4(I)/4104.D0
900           CONTINUE
              IF(TERM(1) .LE. 10.D0)THEN
                 IF(STEP .EQ. DTMIN)THEN
                    ZEROVOL=.TRUE.
                    PASS=.TRUE.
                    GOTO 1375
                 ELSE
                    ZEROCHK=.TRUE.
```

```
                 ENDIF
              ENDIF
              DO 1000 I=1,N
                 RK5(I)=STEP*F(I,TERM,T,KNT,KOUNT,TBEG,TEND)
1000          CONTINUE
              T=OTIME(KOUNT)-STEP/2.D0
              KNT=6
              DO 1100 I=1,N
                 TERM(I)=X(I)-8.D0*RK1(I)/27.D0+
     +                2.D0*RK2(I)-3544.D0*RK3(I)/2565.D0+
     +                1859.D0*RK4(I)/4104.D0-11.D0*RK5(I)/40.D0
1100          CONTINUE
              IF(TERM(1) .LE. 10.D0)THEN
                 IF(STEP .EQ. DTMIN)THEN
                    ZEROVOL=.TRUE.
                    PASS=.TRUE.
                    GOTO 1375
                 ELSE
                    ZEROCHK=.TRUE.
                 ENDIF
              ENDIF
              DO 1200 I=1,N
                 RK6(I)=STEP*F(I,TERM,T,KNT,KOUNT,TBEG,TEND)
1200          CONTINUE
              IF(TERM(1) .LE. 10.D0)THEN
                 IF(STEP .EQ. DTMIN)THEN
                    ZEROVOL=.TRUE.
                    PASS=.TRUE.
                    GOTO 1375
                 ELSE
                    ZEROCHK=.TRUE.
                 ENDIF
              ENDIF
              PASS=.TRUE.

************************************************
** CALCULATE ERRORS RESULTING FROM STEP SIZE **
************************************************

              DO 1300 I=1,N
                 R(I)=DABS(RK1(I)/360.D0 -128.D0*RK3(I)/4275.D0-
     +                2197.D0*RK4(I)/75240.D0+RK5(I)/50.D0+
     +                2.D0*RK6(I)/55.D0)/STEP
                 IF(R(I).GT.TOL(I)) PASS=.FALSE.
1300          CONTINUE

****************************************************************************
** MAKE SURE THE SOLVER ISN'T "STUCK" BECAUSE OF THE ERROR TOLERANCES  **
****************************************************************************

              IF(R(1) .EQ. RIPREV .AND. STEP .EQ. DTMIN)THEN
                 IF(ZEROCHK)THEN
                    PASS=.TRUE.
                    ZEROVOL=.TRUE.
                    GOTO 1375
                 ELSE
                    WRITE(6,*)
                    WRITE(6,*)'THE CURRENT RUN IS "STUCK" BUT WE HAVE
     +FORCED IT TO MOVE ON'
                    WRITE(6,*)'DESPITE THE GIVEN TOLERANCES'
                    WRITE(6,*)
                    PASS=.TRUE.
                    GOTO 1375
                 ENDIF
              ELSE
                 RIPREV=R(1)
```

```
            ENDIF

            DO 1310 I=1,N
               IF(R(I) .EQ. 0.D0)R(I)=.1
1310        CONTINUE
            DELMIN=4.D0

*******************************************************
** 'DEL' IS A VARIABLE USED TO UPDATE THE STEP SIZE **
*******************************************************

            DO 1350 I = 1,N
               DEL(I)=0.84*(TOL(I)/R(I))**(1.D0/4.D0)
               DELMIN=DMIN1(DEL(I),DELMIN)
1350        CONTINUE

*********************************************************
** IF THE ERROR IS LESS THAN THE GIVEN TOLERANCES ... **
*********************************************************

1375        IF(PASS)THEN
               IF(OTIME(KOUNT)-STEP.GE.TEND)THEN
                  KOUNT=KOUNT+1
                  OTIME(KOUNT)=OTIME(KOUNT-1)-STEP
                  IF(ZEROVOL)THEN
                     X(1)=0.D0
                     X(2)=TODELI
                     ZEROVOL=.FALSE.
                     ZEROCHK=.FALSE.
                  ELSE

***********************************
** CALCULATE VOLUME AND DEL O-18 **
***********************************

                     DO 1400 I=1,N
                        X(I)=X(I)+25.D0*RK1(I)/216.D0+
     +                       1408.D0*RK3(I)/2565.D0+
     +                       2197.D0*RK4(I)/4104.D0- RK5(I)/5.D0
1400                 CONTINUE
                  ENDIF

**************************************************
** SALT BALANCE STUFF FOR THE ENTIRE TIME-STEP **
**************************************************

                  IF(X(1) .GT. VOLMAX)THEN
**************
** OVERFLOW **
**************
                     VOL_OUT=X(1)-VOLMAX
                     CL(KOUNT)=CL(KOUNT-1)+QCL_IN*STEP-
     +                    VOL_OUT*1.D3*CONC_CL(KOUNT-1)
                     IF(CL(KOUNT).LT.0.D0)CL(KOUNT)=3.95D-4*30.02D9*
     +                    1.D3
                     X(1)=VOLMAX
                     CONC_CL(KOUNT)=CL(KOUNT)/(X(1)*1.D3)

******************************************************************************
** "OCL_OUT" RECORDS THE SUM MOLES OF CL THAT LEAVE DURING EACH TIME-STEP **
******************************************************************************

                     OCL_OUT(KOUNT)= VOL_OUT*1.D3*CONC_CL(KOUNT-1)+
     +                    OCL_OUT(KOUNT-1)

                  ELSE IF(X(1) .LE. 10.D0)THEN
```

```
**************
** DRY LAKE **
**************
                     OCL_OUT(KOUNT)=OCL_OUT(KOUNT-1)
                     CL(KOUNT)=0.D0
                     X(1)=0.D0
                     CONC_CL(KOUNT)=0.D0
                  ELSE

******************************
** BETWEEN DRY AND OVERFLOW **
******************************

                     OCL_OUT(KOUNT)=OCL_OUT(KOUNT-1)
                     CL(KOUNT)=CL(KOUNT-1)+QCL_IN*STEP
                     CONC_CL(KOUNT)=CL(KOUNT)/(X(1)*1.D3)

*****************************
** CHECK FOR CL SATURATION **
*****************************

                     IF(CONC_CL(KOUNT) .GT. 6.1D0)THEN
                        CONC_CL(KOUNT)=6.1D0
                        CL(KOUNT)=6.1D0*(X(1)*1.D3)
                     ENDIF
                  ENDIF

************************************************
** CALCULATE DEL DOLOMITE FROM DEL WATER, X(2)**
************************************************

                  ODEL_OUT(KOUNT)=X(2)
                  DELDOL=FDDOL(DOLTEMP,X(2))

*****************************************************
** STORE VALUES IN ARRAYS FOR THE PLOTTING ROUTINE **
*****************************************************

                  CALL FINDQT(NQPTS,OTIME(KOUNT),IQNDEX,QITIME)

                  IF(INCHOICE .GE. 8)THEN
                     CALL QINTERP(OTIME(KOUNT),IQNDEX,QI,QIHIST,
     +                    QITIME)
                  ELSE
                     QI=FQI(TBEG-OTIME(KOUNT))
                  ENDIF
                  IF(QI .LE. 0.D0)QI=0.D0

                  IF(GRAF)THEN
                     YQI(KOUNT)=QI
                     YTEMP(KOUNT)=DOLTEMP
                     YDEL(KOUNT)=DELDOL
                     YAREA(KOUNT)=OAREA(X(1))
                     XPLT(KOUNT)=OTIME(KOUNT)
                  ENDIF

***************************
** WRITE RESULTS TO FILE **
***************************

                  AREA(KOUNT)=OAREA(X(1))
*                  WRITE(96,*)OTIME(KOUNT),AREA(KOUNT)
*                  WRITE(99,*)OTIME(KOUNT),DELDOL
                   WRITE(95,*)OTIME(KOUNT),QI

                  IF(SU)THEN
```

```
                   IF(OTIME(KOUNT) .LT. SAVETIME .AND. ISTCNT .LE.
     +                  NUMST)THEN
                      WRITE(70,*)'OWENS LAKE'
                      WRITE(70,*)'STARTUP TIME #',ISTCNT
                      WRITE(70,*)OTIME(KOUNT),X(1),DELDOL,
     +                     CONC_CL(KOUNT)
                      WRITE(70,*)
                      ISTCNT=ISTCNT+1
                      SAVETIME=SUT(ISTCNT)
                   ENDIF
                ENDIF

                STEP=DTMAX
                GOTO 100

**********************************
** MAKE SURE THE SOLVER DOESN'T **
** OVERSTEP DESIGNATED END TIME **
**********************************

*******************************************************************
** THIS ELSE IF CORRESPONDS TO "IF(OTIME(KOUNT)-STEP.GE.TEND)THEN **
*******************************************************************

             ELSEIF(OTIME(KOUNT)-STEP.LT.TEND)THEN
                DTMAX=OTIME(KOUNT)-TEND
                STEP=DTMAX
                GOTO 100
             ENDIF

**************************************
** ADJUST THE SIZE OF THE TIME STEP **
**************************************

***********************************************
** THIS ELSEIF CORRESPONDS TO "IF(PASS)THEN" **
***********************************************

          ELSEIF(DELMIN .LE. 0.1)THEN
             STEP=STEP*1.0D-1
          ELSEIF(DELMIN .GE. 4.D0)THEN
             STEP=4.D0*STEP
          ELSE
             STEP=DELMIN*STEP
          ENDIF
          IF(STEP.GT.DTMAX)STEP=DTMAX
          IF(STEP.LT.DTMIN)STEP=DTMIN

************************************************************
** THIS ELSE CORRESPONDS TO "IF(OTIME(KOUNT).GT.TEND)THEN" **
************************************************************

       ELSE
          IF(.NOT. GRAF)THEN
             WRITE(6,*)
             WRITE(6,*)'THE RKF SOLVED',KOUNT,' POINTS IN',ITER,' IT
     +ERATIONS'
             WRITE(6,*)'FOR OWENS LAKE'
             WRITE(6,*)
             WRITE(6,*)'FINAL DEL FOR OWENS =',DELDOL
             WRITE(6,*)'FINAL AREA FOR OWENS  =',AREA(KOUNT)
             WRITE(6,*)'FINAL VOLUME FOR OWENS =',X(1)
             WRITE(6,*)'FINAL CL CONC, OWENS =',CONC_CL(KOUNT)
             FINDEL=DELDOL
             FINAREA=AREA(KOUNT)
             FINVOL=X(1)
```

```
            FINCL=CONC_CL(KOUNT)
         ENDIF
         IF(GRAF)THEN

            WRITE(6,*)'CALLING PLOTTING ROUTINE'
            WRITE(6,*)
            WRITE(6,*)'THE RKF SOLVED',KOUNT,' POINTS IN',ITER,' IT
     +ERATIONS'
            WRITE(6,*)
            WRITE(6,*)'FINAL DEL FOR OWENS =',DELDOL
            WRITE(6,*)'FINAL AREA FOR OWENS  =',AREA(KOUNT)
            WRITE(6,*)'FINAL VOLUME FOR OWENS =',X(1)
            WRITE(6,*)'FINAL CL CONC, OWENS =',CONC_CL(KOUNT)
            FINDEL=DELDOL
            FINAREA=AREA(KOUNT)
            FINVOL=X(1)
            FINCL=CONC_CL(KOUNT)
            CALL PLOTEM(XMIN,XMAX,XPLT,YDEL,YAREA,YQI,YTEMP,KOUNT)

*************************************
** FINISH PLOTTING STUFF FOR OWENS **
*************************************

            CALL ENDPL(0)
            CALL DONEPL
            WRITE(6,*)'FINISHED WITH OWENS LAKE'

         ENDIF
         NOPTS=KOUNT
         RETURN
      ENDIF
100   CONTINUE
      NOPTS=KOUNT
      IF(ITER .GT. ITMAX)WRITE(6,*)'MAX # OF ITERATIONS EXCEEDED'
      IF(.NOT. GRAF)THEN
         WRITE(6,*)
         WRITE(6,*)'THE RKF SOLVED',KOUNT,' POINTS IN',ITER,' IT
     +ERATIONS'
         WRITE(6,*)'FOR OWENS LAKE'
         WRITE(6,*)
         WRITE(6,*)'FINAL DEL FOR OWENS =',DELDOL
         WRITE(6,*)'FINAL AREA FOR OWENS  =',AREA(KOUNT)
         WRITE(6,*)'FINAL VOLUME FOR OWENS =',X(1)
         WRITE(6,*)'FINAL CL CONC, OWENS =',CONC_CL(KOUNT)
         FINDEL=DELDOL
         FINAREA=AREA(KOUNT)
         FINVOL=X(1)
         FINCL=CONC_CL(KOUNT)
         WRITE(6,*)
      ENDIF
      IF(GRAF)THEN
         WRITE(6,*)'CALLING PLOTTING ROUTINE'
         WRITE(6,*)
         WRITE(6,*)'THE RKF SOLVED',KOUNT,' POINTS IN',ITER,' ITERATIO
     +NS'
         WRITE(6,*)

         CALL PLOTEM(XMIN,XMAX,XPLT,YDEL,YAREA,YQI,YTEMP,KOUNT)

*************************************
** FINISH PLOTTING STUFF FOR OWENS **
*************************************

         CALL ENDPL(0)
         CALL DONEPL
         WRITE(6,*)'FINISHED WITH OWENS LAKE'
```

```
          WRITE(6,*)
          WRITE(6,*)'FINAL DEL FOR OWENS =',DELDOL
          WRITE(6,*)'FINAL AREA FOR OWENS  =',AREA(KOUNT)
          WRITE(6,*)'FINAL VOLUME FOR OWENS =',X(1)
          WRITE(6,*)'FINAL CL CONC, OWENS =',CONC_CL(KOUNT)
          FINDEL=DELDOL
          FINAREA=AREA(KOUNT)
          FINVOL=X(1)
          FINCL=CONC_CL(KOUNT)
          WRITE(6,*)
        ENDIF

        END

*******************************************************************************
*******************************************************************************
*    BELOW LIES A CHAOTIC CONVOLUTION OF ESOTERIC ENIGMAS THAT HOPEFULLY      *
*    ACCOMPLISH THE ISOTOPIC AND LAKE LEVEL VOODOO WE SET OUT TOODOO.         *
*    ACTUALLY THIS PART OF THE PROGRAM CALCULATES THE DERIVATIVES OF LAKE     *
*    VOLUME AND ISOTOPIC COMPOSITION WITH RESPECT TO TIME.                    *
*******************************************************************************
*******************************************************************************

        DOUBLE PRECISION FUNCTION F(I,TERM,T,KNT,KOUNT,TBEG,TEND)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)

        LOGICAL FOUND,GUESS,GRAF,CPARAM,GUESS2,SU

        PARAMETER(NUMA=15000,NUMB=2500)
        PARAMETER(QCL_IN=1.67D8,VOLMAX=30.02D9)

        COMMON/BOTH/CL(NUMA),TSOD(10),TCL(10),OTIME(NUMA),TCO3(10),
     +  OQO(NUMA),QO(10),CONC_CL(NUMA),DOLTEMP,DELDOL,TODELI,TOTEMP,
     +  ODEL_OUT(NUMA),OCL_OUT(NUMA),PEVAP(NUMB),DEVAP(NUMB),
     +  SUM_PCL_DEP(NUMA),SUM_DCL_DEP(NUMA),AREA_P,AREA_D,
     +  AREA(NUMA),SOAREA,ALLAREA(NUMA),CONC_CL_P(NUMA),CONC_CL_D(NUMA),
     +  CL_P(NUMA),CL_D(NUMA),INCHOICE

        COMMON/BOTH2/CL_C(NUMA),TSOD_C(10),TCL_C(10),TIME,TCO3_C(10),
     +  CQO(10),GRAF,CONC_CL_C(NUMA),DOLTEMP_C,DELDOL_C,
     +  TCDELI,TCTEMP,CDEL_OUT(NUMA),CCL_OUT(NUMA),CL_S(NUMA),NOPTS,
     +  TSOD_S(10),TCL_S(10),TCO3_S(10),SQO(10),CONC_CL_S(NUMA),
     +  DOLTEMP_S,DELDOL_S,TSDELI,TSTEMP,SCL_DEP(NUMA),TTLCL_IN,NPTS,
     +  SUM_SCL_DEP(NUMA),QI_C,QI_S,PQI

        COMMON/FIRST/A,B,WTIME(NUMB),OTEMP(NUMB),OEVAP(NUMB),
     +  OHUM(500),OPRECIP(NUMB),ODELP(NUMB),ODELA(NUMB),ODELI(NUMB),
     +  GUESS,C,TEMPC,EVAPC,PRECIPC,DELAC,DELIC,CPARAM,DELPC,
     +  EVAPC_P,EVAPC_D

        COMMON/SECOND/CTEMP(NUMB),CEVAP(NUMB),CHUM(500),CPRECIP(NUMB),
     +  CDELP(NUMB),CSDELA(NUMB),GUESS2,TEMPC_C,EVAPC_C,
     +  PRECIPC_C,DELAC_C,DELPC_C,STEMP(NUMB),SEVAP(NUMB),SHUM(500),
     +  SPRECIP(NUMB),SDELP(NUMB),TEMPC_S,EVAPC_S,
     +  PRECIPC_S,DELAC_S,DELPC_S

        COMMON/HIST/QIHIST(1000),HUMTIME(500),NHPTS,SU,SUT(15),
     +  SAVETIME,NUMST,QITIME(1000),NQPTS

        DIMENSION TERM(3),TCONC_CL(10),V_OUT(10),TDTIME(10),TQI(10)


        IF(I.EQ.1)THEN

*************************************
** CALCULATE DV/DT FOR OWENS LAKE **
```

```
************************************

          VOL = TERM(1)
          IF(VOL .LE. 10.D0)VOL=0.D0

10        FORMAT(A,D12.5)

*******************************
** CONSTANT PARAMETER OPTION **
*******************************

          IF(CPARAM)THEN
             TODELI=DELIC
             TOTEMP=TEMPC
             TOEVAP=EVAPC
             TOPRECIP=PRECIPC
             TODELP=DELPC
             TODELA=DELAC
             CALL FINDHT(NHPTS,T,IHNDEX,HUMTIME)
             CALL FINDQT(NQPTS,T,IQNDEX,QITIME)
             CALL HUMTERP(T,IHNDEX,TOHUM,OHUM,HUMTIME)
          ELSE

**********************************************************************
** DETERMINE VALUES OF NECESSARY PARAMETERS BY ASSIGNING VALUES OR  **
** INTERPOLATING BETWEEN GIVEN VALUES                               **
**********************************************************************

             CALL FINDT(NPTS,T,INDEX,FOUND,WTIME)
             CALL FINDHT(NHPTS,T,IHNDEX,HUMTIME)
             CALL FINDQT(NQPTS,T,IQNDEX,QITIME)

             IF(FOUND)THEN
                TODELI = ODELI(INDEX)
                TOTEMP = OTEMP(INDEX)
                TOEVAP = OEVAP(INDEX)
                TOPRECIP = OPRECIP(INDEX)
                TODELP = ODELP(INDEX)
                TODELA = ODELA(INDEX)
             ELSE
                CALL OINTERP(T,INDEX,TODELI,TOTEMP,TOEVAP,TOPRECIP,
     +               TODELP,TODELA)
             ENDIF
             CALL HUMTERP(T,IHNDEX,TOHUM,OHUM,HUMTIME)
          ENDIF

******************************************************************
** "REMEMBER" TEMP AT END OF TIMESTEP TO CALCULATE DEL DOLOMITE **
******************************************************************

          IF(KNT .EQ. 5)DOLTEMP=TOTEMP

************************************************
** TRACK TOTAL ELAPSED TIME TO CALCULATE INFLOW**
************************************************

          ETIME = TBEG-T

****************************************************
** ALSO NEED TO KNOW DEL TIME WITHIN THE TIME-STEP **
****************************************************

          DTIME = OTIME(KOUNT)-T

***************************
** CALCULATE AREA OF LAKE **
```

```
****************************

          AREA(KOUNT) = OAREA(VOL)

*****************************
** CALCULATE PRECIPITATION **
*****************************

          QP = AREA(KOUNT)*TOPRECIP

*********************************
** THE INFAMOUS "SALT" BALANCE **
*********************************

          IF(T .EQ. OTIME(KOUNT))THEN
             TCL(KNT)=CL(KOUNT)
             TCONC_CL(KNT)=CONC_CL(KOUNT)

***************************************************
** CALCULATE TOTAL OUTFLOW VOLUME FROM TIME TO T **
** AND ADJUST IF VOL EXCEEDS VOLMAX              **
***************************************************

          ELSE
             IF(VOL .LE. VOLMAX)THEN
                V_OUT(KNT)=0.D0
             ELSE
                V_OUT(KNT) = VOL-VOLMAX
                VOL=VOLMAX
             ENDIF

*********************************************************
**  CALCULATE CONCENTRATIONS FOR INTERMEDIATE TIME "T" **
*********************************************************

             TCL(KNT) = (OTIME(KOUNT)-T)*QCL_IN+CL(KOUNT)-V_OUT(KNT)*
     +            CONC_CL(KOUNT)*1.D3
             IF(TCL(KNT).LT.0.D0)TCL(KNT)=3.95D-4*30.02D9*1.D3

*********************************************************************
** CONCENTRATION UNITS ARE MOLARITY SO VOL MUST BE MULT BY 1000 TO **
** CONVERT M^3 TO LITERS **
*********************************************************************

             IF(VOL .GT. 10.D0)THEN
                TCONC_CL(KNT)=TCL(KNT)/(VOL*1.0D3)
             ELSE
                TCONC_CL(KNT)=0.D0
                TCL(KNT)=0.D0
             ENDIF

***********************************
** CHECK FOR CHLORIDE SATURATION **
***********************************

             IF(TCONC_CL(KNT) .GT. 6.1D0)THEN
                TCONC_CL(KNT)=6.1D0
                TCL(KNT)=6.1D0*(VOL*1.D3)
             ENDIF

********************************************************************
** AMT OF SODIUM IS THE AMT NECESSARY TO ACHIEVE ELECTRONEUTALITY **
********************************************************************

             TSOD(KNT)=TCL(KNT)
          ENDIF
```

```
*********************************************
**  CALCULATE BACK-CONDENSATION FLUX (QC) **
*********************************************

          IF(VOL .GT. 10.D0 .AND. TCONC_CL(KNT) .GT. 0.D0)THEN
             PHI=FPHI(TCONC_CL(KNT),SUM)
          ELSE
             PHI=0.D0
          ENDIF

          AW=DEXP(-18.D0*PHI*SUM*0.5D0/1.D3)

*********************************
** CALCULATE EVAPORATION (QE) **
*********************************

          QE = AREA(KOUNT)*TOEVAP*AW

          IF(VOL .GT. 10.D0)THEN
             QC=(TOHUM*QE)/AW
          ELSE
             QC=0.D0
          ENDIF

*************************************
**  CALCULATE DV/DT IF VOL IS ZERO **
*************************************

          IF(TERM(1) .LE. 10.D0)THEN
             IF(INCHOICE .GE. 8)THEN
                CALL QINTERP(T,IQNDEX,QI,QIHIST,QITIME)
             ELSE
                QI=FQI(ETIME)
             ENDIF
             IF(QI .LT. 0.D0)QI=0.D0
             IF(GUESS)THEN
                WRITE(6,*)
                WRITE(6,*)'FYI,  DV/DT = ',QI
                WRITE(6,*)
*               WRITE(96,*)TBEG,AREA(KOUNT)
                WRITE(95,*)TBEG,QI
                DELDOL=FDDOL(TOTEMP,TERM(2))
*               WRITE(99,*)TBEG,DELDOL
                GUESS=.FALSE.
             ENDIF

             QO(KNT)=0.D0

****************************************
** ASSUME QO FOR OWENS = QI FOR CHINA **
****************************************

             IF(KNT .EQ. 1)THEN
                OQO(KOUNT)=QO(KNT)
*               WRITE(97,*)T,OQO(KOUNT)
             ENDIF
             IF(T .EQ. TEND)THEN
                OQO(KOUNT+1)=QO(KNT)
*               WRITE(97,*)T,OQO(KOUNT)
             ENDIF

             F=QI

             IF(F .LE. 0.D0)THEN
                DV_DT=0.D0
```

```
              ELSE
                 DV_DT=F
              ENDIF

*************************************************
**  CALCULATE DV/DT IF VOL IS LESS THAN VOLMAX **
*************************************************

           ELSE IF(TERM(1)  .LT. VOLMAX)THEN

***************************
** CALCULATE INFLOW (QI) **
***************************

              IF(INCHOICE .GE. 8)THEN
                 CALL QINTERP(T,IQNDEX,QI,QIHIST,QITIME)
              ELSE
                 QI=FQI(ETIME)
              ENDIF
              IF(QI .LT. 0.D0)QI=0.D0
              IF(GUESS)THEN
                 WRITE(6,*)
                 WRITE(6,*)'FYI, DV/DT =',QI+QP+QC-QE
                 WRITE(6,*)
*                WRITE(96,*)TBEG,AREA(KOUNT)
                 WRITE(95,*)TBEG,B
                 DELDOL=FDDOL(TOTEMP,TERM(2))
*                WRITE(99,*)TBEG,DELDOL
                 GUESS=.FALSE.
              ENDIF

**********************
**  CALCULATE DV/DT **
**********************

              F=QI+QC-QE+QP
              DV_DT=F
              QO(KNT)=0.D0
              IF(KNT .EQ. 1)THEN
                 OQO(KOUNT)=QO(KNT)
*                WRITE(97,*)T,OQO(KOUNT)
              ENDIF
              IF(T .EQ. TEND)THEN
                 OQO(KOUNT+1)=QO(KNT)
*                WRITE(97,*)T,OQO(KOUNT)
              ENDIF

********************************************
**  CALCULATE DV/DT IF OWENS LAKE IS FULL **
********************************************

           ELSE

******************
** CALCULATE QI **
******************

              IF(INCHOICE .GE. 8)THEN
                 CALL QINTERP(T,IQNDEX,QI,QIHIST,QITIME)
              ELSE
                 QI=FQI(ETIME)
              ENDIF

              IF(GUESS)THEN
                 WRITE(6,*)
                 WRITE(6,*)'FYI, QO=',QI+QP+QC-QE
```

```
*                  WRITE(96,*)TBEG,AREA(KOUNT)
                  WRITE(95,*)TBEG,B
                  DELDOL=FDDOL(TOTEMP,TERM(2))
*                 WRITE(99,*)TBEG,DELDOL
                  GUESS=.FALSE.
               ENDIF

******************
** CALCULATE QO **
******************

               QO(KNT)=QI+QC-QE+QP

*********************
** CALCULATE DV/DT **
*********************

               IF(QO(KNT) .LT. 0.D0)THEN
                  QO(KNT)=0.D0
                  IF(KNT .EQ. 1)THEN
                     OQO(KOUNT)=QO(KNT)
*                    WRITE(97,*)T,OQO(KOUNT)
                  ENDIF
                  IF(T .EQ. TEND)THEN
                     OQO(KOUNT+1)=QO(KNT)
*                    WRITE(97,*)T,OQO(KOUNT)
                  ENDIF
                  F=QI+QC-QE+QP
                  DV_DT=F
               ELSE
                  F=QO(KNT)
                  IF(KNT .EQ. 1)THEN
                     OQO(KOUNT)=QO(KNT)
*                    WRITE(97,*)T,OQO(KOUNT)
                  ENDIF
                  IF(T .EQ. TEND)THEN
                     OQO(KOUNT+1)=QO(KNT)
*                    WRITE(97,*)T,OQO(KOUNT)
                  ENDIF
                  DV_DT=0.D0
               ENDIF
            ENDIF

******************************************************************************
*   A FUNCTION SUBROUTINE TO CALCULATE THE ISOTOPIC HISTORY OF OWENS LAKE    *
******************************************************************************

         ELSE IF(I.EQ.2)THEN

            DELL = TERM(2)

************************
** CALCULATE DDEL/DT  **
************************

            IF(VOL .LE. 10.D0)THEN
               F=0.D0

            ELSE

******************************************
** CALCULATE ISOTOPIC ENRICHMENT FACTOR **
******************************************

               EPS = FEPS(TOTEMP)
```

```
********************************************
** CALCULATE DEL OF THE BACK-CONDENSATION **
********************************************

              ODELC =EPS*(1.D0+(TODELA/1.D3))+TODELA

**************************************
** CALCULATE DEL OF THE EVAPORATION **
**************************************

              ODELE = DELE(DELL,EPS,TOHUM)

*****************************************************
** SET DEL OF THE OUTFLOW EQUAL TO DEL OF THE LAKE **
*****************************************************

              ODELO = DELL

              F=(QI*TODELI+QC*ODELC+QP*TODELP-QO(KNT)*ODELO-QE*
     +             ODELE-DELL*DV_DT)/VOL

           ENDIF

        END IF
        RETURN
        END




        SUBROUTINE RKF_CS(NCS,X_CS,TBEG,TEND,TOL_CS,DTMAX_CS,DTMIN_CS,
     +       ITMAX_CS)
********************************************************************************
*       SOLVE A SYSTEM OF PARTIAL DIFFERENTIAL EQUATION OF THE FORM:          *
*                          F(T,X)= X'                                          *
*        BETWEEN T1,T2, GIVEN THE INITIAL CONDITION XO(T1)                     *
********************************************************************************
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)

        PARAMETER(NUMA=15000,NUMB=2500)
        PARAMETER (VOLMAX_C=0.696D9,VOLMAX_S=85.28D9,AREAMAX_P=.727D9,
     +       AREAMAX_D=0.583D9,SLTCONST=0.0127)

        LOGICAL PASS,GRAF,ONLY1,ZEROVOL_C,ZEROCHK_C,ZEROVOL_S,
     + ZEROCHK_S,COAL,DECOUP,COUP,FOUND,CPARAM,GUESS,SU,GU,GD

        COMMON/FIRST/A,B,WTIME(NUMB),OTEMP(NUMB),OEVAP(NUMB),
     + OHUM(500),OPRECIP(NUMB),ODELP(NUMB),ODELA(NUMB),ODELI(NUMB),
     + GUESS,C,TEMPC,EVAPC,PRECIPC,DELAC,DELIC,CPARAM,DELPC,
     + EVAPC_P,EVAPC_D

        COMMON/BOTH/CL(NUMA),TSOD(10),TCL(10),OTIME(NUMA),TCO3(10),
     + OQO(NUMA),QO(10),CONC_CL(NUMA),DOLTEMP,DELDOL,TODELI,TOTEMP,
     + ODEL_OUT(NUMA),OCL_OUT(NUMA),PEVAP(NUMB),DEVAP(NUMB),
     + SUM_PCL_DEP(NUMA),SUM_DCL_DEP(NUMA),AREA_P,AREA_D,
     + AREA(NUMA),SOAREA,ALLAREA(NUMA),CONC_CL_P(NUMA),CONC_CL_D(NUMA),
     + CL_P(NUMA),CL_D(NUMA),INCHOICE

        COMMON/BOTH2/CL_C(NUMA),TSOD_C(10),TCL_C(10),TIME,TCO3_C(10),
     + CQO(10),GRAF,CONC_CL_C(NUMA),DOLTEMP_C,DELDOL_C,
     + TCDELI,TCTEMP,CDEL_OUT(NUMA),CCL_OUT(NUMA),CL_S(NUMA),NOPTS,
     + TSOD_S(10),TCL_S(10),TCO3_S(10),SQO(10),CONC_CL_S(NUMA),
     + DOLTEMP_S,DELDOL_S,TSDELI,TSTEMP,SCL_DEP(NUMA),TTLCL_IN,NPTS,
     + SUM_SCL_DEP(NUMA),QI_C,QI_S,PQI

        COMMON/NEW/OPREV,CPREV,SPREV,PPREV,DPREV
```

```
      COMMON/HIST/QIHIST(1000),HUMTIME(500),NHPTS,SU,SUT(15),
     + SAVETIME,NUMST,QITIME(1000),NQPTS

      COMMON/FINAL/FINDEL,FINVOL,FINAREA,FINCL

      COMMON/FIX/GU,GD,AVGIN,ITGCNT

      REAL XMIN,XMAX,YDEL_C(NUMA),XPLT(NUMA),YQI_C(NUMA),
     + YAREA_C(NUMA),YTEMP_C(NUMA),YDEL_S(NUMA),YQI_S(NUMA),
     + YAREA_S(NUMA),YTEMP_S(NUMA),YCL_DEP(NUMA),YSUM_SCL_DEP(NUMA),
     + YCONC(NUMA),YCL_INV(NUMA)

      DIMENSION X_CS(4),RK1(4),RK2(4),RK3(4),RK4(4),RK5(4),RK6(4),R(4)
      DIMENSION TERM(4),DEL(4),TOL_CS(4)


*     OPEN(UNIT=88,FILE='DEL_C.OUT',STATUS='NEW',CARRIAGE CONTROL=
*     +      'LIST')
      OPEN(UNIT=87,FILE='CONC_CL_S.OUT',STATUS='NEW',CARRIAGE CONTROL=
     +      'LIST')
*     OPEN(UNIT=86,FILE='QI_C.OUT',STATUS='NEW',CARRIAGE CONTROL=
*     +      'LIST')

      OPEN(UNIT=78,FILE='DEL_S.OUT',STATUS='NEW',CARRIAGE CONTROL=
     +      'LIST')
      OPEN(UNIT=77,FILE='AREA_S.OUT',STATUS='NEW',CARRIAGE CONTROL=
     +      'LIST')
      OPEN(UNIT=76,FILE='QI_S.OUT',STATUS='NEW',CARRIAGE CONTROL=
     +      'LIST')
      OPEN(UNIT=74,FILE='SUMCL_DEP.OUT',STATUS='NEW',CARRIAGE
     +       CONTROL='LIST')

      OPEN(UNIT=91,FILE='QO_S.OUT',STATUS='NEW',CARRIAGE
     +       CONTROL='LIST')
*     OPEN(UNIT=73,FILE='AREA_P.OUT',STATUS='NEW',CARRIAGE
*     +       CONTROL='LIST')
*     OPEN(UNIT=72,FILE='AREA_D.OUT',STATUS='NEW',CARRIAGE
*     +       CONTROL='LIST')
      OPEN(UNIT=83,FILE='SUM_PCL_DEP.OUT',STATUS='NEW',CARRIAGE
     +       CONTROL='LIST')

10    FORMAT(A,D12.5)

      XMIN=TBEG
      XMAX=TEND

      TIME=TBEG
      STEP=DTMAX_CS
      KOUNT=1

      CDEL_OUT(KOUNT)=X_CS(2)
      SCL_DEP(KOUNT)=0.D0
      CCL_OUT(KOUNT)=0.D0

**********************************
** WRITE INITIAL VALUES TO FILE **
**********************************

      WRITE(74,*)TBEG,SUM_SCL_DEP(KOUNT)
*     WRITE(73,*)TBEG,AREA_P
*     WRITE(72,*)TBEG,AREA_D
      WRITE(83,*)TBEG,SUM_PCL_DEP(KOUNT)
*     WRITE(82,*)TBEG,SUM_DCL_DEP(KOUNT)

      WRITE(87,*)TBEG,CONC_CL_S(KOUNT)
```

```
      ONLY1=.TRUE.
      COAL=.FALSE.
      DECOUP=.FALSE.
      COUP=.FALSE.
      GU=.FALSE.
      GD=.FALSE.
      ZEROVOL_C=.FALSE.
      ZEROCHK_C=.FALSE.
      ZEROVOL_S=.FALSE.
      ZEROCHK_S=.FALSE.

      IF(X_CS(3) .GT. 65.87D9)COAL=.TRUE.

********************************************
** INITIALIZE PARAMETERS TO RESTART MODEL **
********************************************

      SAVETIME=SUT(1)
      ISTCNT=1

*************************************
** THE SOLVING ROUTINE BEGINS HERE **
*************************************

      WRITE(6,*)
      WRITE(6,*)
      WRITE(6,*)'START SOLVING DIFFERENTIAL EQUATIONS'
      WRITE(6,*)'FOR CHINA AND SEARLES LAKE'
      WRITE(6,*)
      WRITE(6,*)

      DO 100 ITER=1,ITMAX_CS
         IF(TIME.GT.TEND)THEN

         IF(COAL)THEN
            NST=3
         ELSE
            NST=1
         ENDIF

            KNT=1
            T=TIME
            DO 200 I=NST,NCS
               RK1(I)=STEP*F_CS(I,X_CS,T,KNT,KOUNT,TBEG,COAL)
200         CONTINUE

********************************************
** STORE INITIAL VALUES IN GRAPHICS ARRAY **
********************************************

            IF(ONLY1)THEN
               YDEL_C(1)=DELDOL_C
               YDEL_S(1)=DELDOL_S
               YSUM_SCL_DEP(1)=SUM_SCL_DEP(1)
               XPLT(1)=TBEG
               YAREA_C(1)=CAREA(X_CS(1))
               YAREA_S(1)=SAREA(X_CS(3))
               YQI_C(1)=OQO(1)
               YQI_S(1)=CQO(1)
               YTEMP_C(1)=TCTEMP
               YTEMP_S(1)=TSTEMP
               YCONC(1)=CONC_CL_S(1)
               YCL_INV(1)=CL_S(1)*0.035453D0
               ONLY1=.FALSE.
            ENDIF
```

```
***************************************************************************
** "TERM" IS AN ARRAY WHICH STORES APPROXIMATIONS OF VOL AND DEL O-18 **
** WHICH WILL BE USED IN FINAL CALCULATIONS IF ERRORS WITHIN THE STEP **
** ARE LESS THAN THE GIVEN TOLERANCES.                                **
***************************************************************************

              T=TIME-STEP/4.D0
              KNT=2
              DO 300 I=NST,NCS
                 TERM(I)=X_CS(I)+ RK1(I)/4.D0
300           CONTINUE
              IF(TERM(1) .LE. 10.D0 .AND. .NOT. COAL)THEN
                 IF(STEP .EQ. DTMIN_CS)THEN
                    ZEROVOL_C=.TRUE.
                    PASS=.TRUE.
                 ELSE
                    ZEROCHK_C=.TRUE.
                 ENDIF
              ENDIF
              IF(TERM(3) .LE. 10.D0)THEN
                 IF(STEP .EQ. DTMIN_CS)THEN
                    ZEROVOL_S=.TRUE.
                    PASS=.TRUE.
                 ELSE
                    ZEROCHK_S=.TRUE.
                 ENDIF
              ENDIF
              DO 400 I=NST,NCS
                 RK2(I)=STEP*F_CS(I,TERM,T,KNT,KOUNT,TBEG,COAL)
400           CONTINUE
              T=TIME-3.D0*STEP/8.D0
              KNT=3
              DO 500 I=NST,NCS
                 TERM(I)=X_CS(I)+(3.D0*RK1(I)+9.D0*RK2(I))/32.D0
500           CONTINUE

              IF(TERM(1) .LE. 10.D0 .AND. .NOT. COAL)THEN
                 IF(STEP .EQ. DTMIN_CS)THEN
                    ZEROVOL_C=.TRUE.
                    PASS=.TRUE.
                 ELSE
                    ZEROCHK_C=.TRUE.
                 ENDIF
              ENDIF
              IF(TERM(3) .LE. 10.D0)THEN
                 IF(STEP .EQ. DTMIN_CS)THEN
                    ZEROVOL_S=.TRUE.
                    PASS=.TRUE.
                 ELSE
                    ZEROCHK_S=.TRUE.
                 ENDIF
              ENDIF

              DO 600 I=NST,NCS
                 RK3(I)=STEP*F_CS(I,TERM,T,KNT,KOUNT,TBEG,COAL)
600           CONTINUE
              T=TIME-12.D0*STEP/13.D0
              KNT=4
              DO 700 I=NST,NCS
                 TERM(I)=X_CS(I)+(1932.D0*RK1(I)-7200.D0*RK2(I)+
    +                    7296.D0*RK3(I))/2197.D0
700           CONTINUE

              IF(TERM(1) .LE. 10.D0 .AND. .NOT. COAL)THEN
                 IF(STEP .EQ. DTMIN_CS)THEN
```

```
            ZEROVOL_C=.TRUE.
            PASS=.TRUE.
         ELSE
            ZEROCHK_C=.TRUE.
         ENDIF
      ENDIF
      IF(TERM(3) .LE. 10.D0)THEN
         IF(STEP .EQ. DTMIN_CS)THEN
            ZEROVOL_S=.TRUE.
            PASS=.TRUE.
         ELSE
            ZEROCHK_S=.TRUE.
         ENDIF
      ENDIF

      DO 800 I=NST,NCS
         RK4(I)=STEP*F_CS(I,TERM,T,KNT,KOUNT,TBEG,COAL)
800   CONTINUE
      T=TIME-STEP
      KNT=5
      DO 900 I=NST,NCS
         TERM(I)=X_CS(I)+439.D0*RK1(I)/216.D0-
     +           8.D0*RK2(I)+3680.D0*RK3(I)/513.D0-
     +           845.D0*RK4(I)/4104.D0
900   CONTINUE

      IF(TERM(1) .LE. 10.D0 .AND. .NOT. COAL)THEN
         IF(STEP .EQ. DTMIN_CS)THEN
            ZEROVOL_C=.TRUE.
            PASS=.TRUE.
         ELSE
            ZEROCHK_C=.TRUE.
         ENDIF
      ENDIF
      IF(TERM(3) .LE. 10.D0)THEN
         IF(STEP .EQ. DTMIN_CS)THEN
            ZEROVOL_S=.TRUE.
            PASS=.TRUE.
         ELSE
            ZEROCHK_S=.TRUE.
         ENDIF
      ENDIF

      DO 1000 I=NST,NCS
         RK5(I)=STEP*F_CS(I,TERM,T,KNT,KOUNT,TBEG,COAL)
1000  CONTINUE
      T=TIME-STEP/2.D0
      KNT=6
      DO 1100 I=NST,NCS
         TERM(I)=X_CS(I)-8.D0*RK1(I)/27.D0+
     +           2.D0*RK2(I)-3544.D0*RK3(I)/2565.D0+
     +           1859.D0*RK4(I)/4104.D0-11.D0*RK5(I)/40.D0
1100  CONTINUE

      IF(TERM(1) .LE. 10.D0 .AND. .NOT. COAL)THEN
         IF(STEP .EQ. DTMIN_CS)THEN
            ZEROVOL_C=.TRUE.
            PASS=.TRUE.
         ELSE
            ZEROCHK_C=.TRUE.
         ENDIF
      ENDIF
      IF(TERM(3) .LE. 10.D0)THEN
         IF(STEP .EQ. DTMIN_CS)THEN
            ZEROVOL_S=.TRUE.
            PASS=.TRUE.
```

```
                ELSE
                   ZEROCHK_S=.TRUE.
                ENDIF
             ENDIF

             DO 1200 I=NST,NCS
                RK6(I)=STEP*F_CS(I,TERM,T,KNT,KOUNT,TBEG,COAL)
1200         CONTINUE

             IF(TERM(1) .LE. 10.D0 .AND. .NOT. COAL)THEN
                IF(STEP .EQ. DTMIN_CS)THEN
                   ZEROVOL_C=.TRUE.
                   PASS=.TRUE.
                ELSE
                   ZEROCHK_C=.TRUE.
                ENDIF
             ENDIF
             IF(TERM(3) .LE. 10.D0)THEN
                IF(STEP .EQ. DTMIN_CS)THEN
                   ZEROVOL_S=.TRUE.
                   PASS=.TRUE.
                ELSE
                   ZEROCHK_S=.TRUE.
                ENDIF
             ENDIF

             IF(ZEROVOL_C .OR. ZEROVOL_S)GOTO 1375

             PASS=.TRUE.

***********************************************
** CALCULATE ERRORS RESULTING FROM STEP SIZE **
***********************************************

             DO 1300 I=NST,NCS
                R(I)=DABS(RK1(I)/360.D0 -128.D0*RK3(I)/4275.D0-
     +                2197.D0*RK4(I)/75240.D0+RK5(I)/50.D0+
     +                2.D0*RK6(I)/55.D0)/STEP
                IF(R(I).GT.TOL_CS(I)) PASS=.FALSE.
1300         CONTINUE

**********************************************************************
** MAKE SURE THE SOLVER ISN'T "STUCK" BECAUSE OF THE ERROR TOLERANCES **
**********************************************************************

             IF(R(1) .EQ. RIPREV .AND. STEP .EQ. DTMIN_CS)THEN
                IF(ZEROCHK_C)ZEROVOL_C=.TRUE.
                IF(ZEROCHK_S)ZEROVOL_S=.TRUE.
                IF(ZEROVOL_C .OR. ZEROVOL_S .AND. .NOT. COAL)THEN
                   PASS=.TRUE.
                   GOTO 1375
                ELSE
                   WRITE(6,*)
                   WRITE(6,*)'THE CURRENT RUN IS "STUCK" BUT WE HAVE
     +FORCED IT TO MOVE ON'
                   WRITE(6,*)'DESPITE THE GIVEN TOLERANCES'
                   WRITE(6,*)
                   PASS=.TRUE.
                   GOTO 1375
                ENDIF
             ELSE
                RIPREV=R(1)
             ENDIF

             DO 1310 I=NST,NCS
                IF(R(I) .EQ. 0.D0)R(I)=.1
```

```
1310          CONTINUE
              DELMIN=4.D0

*******************************************************
** 'DEL' IS A VARIABLE USED TO UPDATE THE STEP SIZE **
*******************************************************

              DO 1350 I = NST,NCS
                 DEL(I)=0.84*(TOL_CS(I)/R(I))**(1.D0/4.D0)
                 DELMIN=DMIN1(DEL(I),DELMIN)
1350          CONTINUE

*********************************************************
** IF THE ERROR IS LESS THAN THE GIVEN TOLERANCES ... **
*********************************************************

1375          IF(PASS)THEN
                 IF(TIME-STEP.GE.TEND)THEN
                    KOUNT=KOUNT+1
                    SLTCORR=SLTCONST*STEP
                    TIME=TIME-STEP

                    IF(GU .OR. GD)THEN
                       AVGIN=QI_S
                       ITGCNT=ITGCNT-1
                       IF(ITGCNT .EQ. 0)THEN
                          GU=.FALSE.
                          GD=.FALSE.
                       ENDIF
                    ENDIF

                    IF(COAL)THEN

************************************************************
** CHECK TO SEE IF CHINA AND SEARLES ARE STILL COALESCED **
************************************************************

                       DO 1700 I=3,4
                          X_CS(I)=X_CS(I)+25.D0*RK1(I)/216.D0+
     +                    1408.D0*RK3(I)/2565.D0+
     +                    2197.D0*RK4(I)/4104.D0- RK5(I)/5.D0
1700                   CONTINUE

                       IF(X_CS(3) .LE. 66.566D9)THEN
                          COAL=.FALSE.
                          X_CS(1)=0.696D9
                          X_CS(2)=X_CS(4)
                          X_CS(3)=X_CS(3)-X_CS(1)
                          WRITE(6,*)
                          WRITE(6,*)'CHINA AND SEARLES HAVE DECOUPLED'
                          WRITE(6,*)'AT',TIME
                          WRITE(6,*)'CONGRATULATIONS, YOU HAVE TWINS'
                          WRITE(6,*)
                          DECOUP=.TRUE.
                       ELSE
                          X_CS(1)=0.D0
                       ENDIF

                    ELSE IF(ZEROVOL_C .AND. ZEROVOL_S)THEN

**********************************************************
** CALCULATE VOLUME AND DEL O-18 IF BOTH LAKES ARE  DRY**
**********************************************************

                       X_CS(1)=0.D0
                       X_CS(2)=TCDELI
```

```
                  ZEROVOL_C=.FALSE.
                  ZEROCHK_C=.FALSE.
                  X_CS(3)=0.D0
                  X_CS(4)=CDEL_OUT(KOUNT-1)
                  ZEROVOL_S=.FALSE.
                  ZEROCHK_S=.FALSE.

               ELSE IF(ZEROVOL_C .AND. .NOT. ZEROVOL_S)THEN

*********************************************************
** CALCULATE VOLUME AND DEL O-18 IF CHINA LAKE IS DRY**
*********************************************************

                  X_CS(1)=0.D0
                  X_CS(2)=TCDELI
                  ZEROVOL_C=.FALSE.
                  ZEROCHK_C=.FALSE.
                  DO 1400 I=3,4
                     X_CS(I)=X_CS(I)+25.D0*RK1(I)/216.D0+
     +                    1408.D0*RK3(I)/2565.D0+
     +                    2197.D0*RK4(I)/4104.D0- RK5(I)/5.D0
1400              CONTINUE

               ELSE IF(ZEROVOL_S .AND. .NOT. ZEROVOL_C)THEN

***********************************************************
** CALCULATE VOLUME AND DEL O-18 IF SEARLES LAKE IS DRY**
***********************************************************

                  X_CS(3)=0.D0
                  X_CS(4)=CDEL_OUT(KOUNT-1)
                  ZEROVOL_S=.FALSE.
                  ZEROCHK_S=.FALSE.
                  DO 1500 I=1,2
                     X_CS(I)=X_CS(I)+25.D0*RK1(I)/216.D0+
     +                    1408.D0*RK3(I)/2565.D0+
     +                    2197.D0*RK4(I)/4104.D0- RK5(I)/5.D0
1500              CONTINUE

               ELSE

************************************************************
** CALCULATE VOLUME AND DEL O-18 IF NEITHER LAKE IS DRY **
** AND CHECK TO SEE IF CHINA AND SEARLES COALESCE        **
************************************************************

                  DO 1600 I=1,NCS
                     X_CS(I)=X_CS(I)+25.D0*RK1(I)/216.D0+
     +                    1408.D0*RK3(I)/2565.D0+
     +                    2197.D0*RK4(I)/4104.D0- RK5(I)/5.D0
1600              CONTINUE

                  IF(X_CS(3) .GT. 65.87D9)THEN
                     COAL=.TRUE.
                     COUP=.TRUE.
                     X_CS(3)=X_CS(3)+0.696D9
                     CPCNT=0.696D9/X_CS(3)
                     SPCNT=1.D0-CPCNT
                     X_CS(4)=CPCNT*X_CS(2)+SPCNT*X_CS(4)
                     X_CS(1)=0.D0
                     X_CS(2)=X_CS(4)
                     WRITE(6,*)
                     WRITE(6,*)'CHINA AND SEARLES HAVE COALESCED'
                     WRITE(6,*)'AT',TIME
                     WRITE(6,*)'IT''S PREHISTORIC COLD FUSION !!!'
                     WRITE(6,*)
```

```
                  ENDIF

               ENDIF

               IF(X_CS(1) .LT. 0.D0)THEN
                  X_CS(1)=0.D0
                  X_CS(2)=TCDELI
               ENDIF

               IF(X_CS(3) .LT. 0.D0)THEN
                  X_CS(3)=0.D0
                  X_CS(4)=CDEL_OUT(KOUNT-1)
               ENDIF

*************************************************
** SALT BALANCE STUFF FOR THE ENTIRE TIME-STEP **
*************************************************

*******************************************
** IF THE LAKES HAVE JUST BEEN DECOUPLED **
*******************************************
               IF(DECOUP)THEN
                  ALL_CL=CL_S(KOUNT-1)+TTLCL_IN
                  ALL_VOL=X_CS(1)+X_CS(3)
                  ALL_CONC=ALL_CL/(ALL_VOL*1.D3)
** CHINA **
                  CONC_CL_C(KOUNT)=ALL_CONC
                  IF(CONC_CL_C(KOUNT) .GT. 6.1D0)THEN
                     CONC_CL_C(KOUNT)=6.1D0
                  ENDIF
                  CL_C(KOUNT)=CONC_CL_C(KOUNT)*X_CS(1)*1.D3
                  CCL_OUT(KOUNT)=0.D0
** SEARLES **
                  CONC_CL_S(KOUNT)=ALL_CONC
                  IF(CONC_CL_S(KOUNT) .GT. 6.1D0)THEN
                     CLDEP=CONC_CL_S(KOUNT)-6.1D0
                     CONC_CL_S(KOUNT)=6.1D0
                     SCL_DEP(KOUNT)=CLDEP*X_CS(3)*35.453D0*
     +                   3.22D-8
                  ELSE
                     SCL_DEP(KOUNT)=0.D0
                  ENDIF

                  GD=.TRUE.
                  AVGIN=QI_S
                  ITGCNT=10

                  SUM_SCL_DEP(KOUNT)=SUM_SCL_DEP(KOUNT-1)+
     +                  SCL_DEP(KOUNT)+SLTCORR
                  CL_S(KOUNT)=X_CS(3)*1.D3*CONC_CL_S(KOUNT)
                  DECOUP=.FALSE.

**********************************************
** IF CHINA AND SEARLES HAVE JUST COALESCED **
**********************************************

               ELSE IF(COUP)THEN
                  CL_S(KOUNT)=CL_C(KOUNT-1)+CL_S(KOUNT-1)+TTLCL_IN
                  CONC_CL_S(KOUNT)=CL_S(KOUNT)/(X_CS(3)*1.D3)
                  IF(CONC_CL_S(KOUNT) .GT. 6.1D0)THEN
                     CLDEP=CONC_CL_S(KOUNT)-6.1D0
                     CONC_CL_S(KOUNT)=6.1D0
                     SCL_DEP(KOUNT)=CLDEP*X_CS(3)*35.453D0*
     +                   3.22D-8
                  ELSE
                     SCL_DEP(KOUNT)=0.D0
```

```
                    ENDIF

                    GU=.TRUE.
                    AVGIN=QI_S
                    ITGCNT=10

                    SUM_SCL_DEP(KOUNT)=SUM_SCL_DEP(KOUNT-1)+
     +                   SCL_DEP(KOUNT)+SLTCORR

                    COUP=.FALSE.

*********************************************************************
** IF CHINA AND SEARLES ARE STILL COALESCED FROM THE LAST TIME STEP **
*********************************************************************

                 ELSE IF(COAL)THEN
                    IF(X_CS(3) .GT. VOLMAX_S)THEN
                       VOL_OUT_S=X_CS(3)-VOLMAX_S
                       CL_S(KOUNT)=CL_S(KOUNT-1)+TTLCL_IN-
     +                      VOL_OUT_S*1.D3*CONC_CL_S(KOUNT-1)
                       IF(CL_S(KOUNT).LT.0.D0)CL_S(KOUNT)=3.95D-4*
     +                      85.28D9*1.0D3
                       X_CS(3)=VOLMAX_S
                       CONC_CL_S(KOUNT)=CL_S(KOUNT)/(X_CS(3)*1.D3)
                       SCL_DEP(KOUNT)=0.D0
                       SUM_SCL_DEP(KOUNT)=SUM_SCL_DEP(KOUNT-1)+
     +                      SCL_DEP(KOUNT)+SLTCORR
                    ELSE
                       CL_S(KOUNT)=CL_S(KOUNT-1)+TTLCL_IN
                       CONC_CL_S(KOUNT)=CL_S(KOUNT)/(X_CS(3)*1.D3)
** SATURATION CHECK **
                       IF(CONC_CL_S(KOUNT) .GT. 6.1D0)THEN
                          CLDEP=CONC_CL_S(KOUNT)-6.1D0
                          CONC_CL_S(KOUNT)=6.1D0
                          CL_S(KOUNT)=X_CS(3)*1.D3*CONC_CL_S(KOUNT)
                          SCL_DEP(KOUNT)=CLDEP*X_CS(3)*35.453D0*
     +                         3.22D-8
                       ELSE
                          SCL_DEP(KOUNT)=0.D0
                       ENDIF

                       SUM_SCL_DEP(KOUNT)=SUM_SCL_DEP(KOUNT-1)+
     +                      SCL_DEP(KOUNT)+SLTCORR

                    ENDIF

********************************************************************
** IF CHINA AND SEARLES AREN'T DOING ANY OF THE COUP/DECOUP STUFF **
********************************************************************

                 ELSE IF(X_CS(1).GT.VOLMAX_C.AND.X_CS(3).GT.VOLMAX_S)
     +                   THEN
**************
** OVERFLOW **  ** CHINA **
**************


                    VOL_OUT_C=X_CS(1)-VOLMAX_C
                    CL_C(KOUNT)=CL_C(KOUNT-1)+TTLCL_IN-
     +                   VOL_OUT_C*1.D3*CONC_CL_C(KOUNT-1)
                    IF(CL_C(KOUNT).LT.0.D0)CL_C(KOUNT)=3.95D-4*
     +                   0.696D9*1.0D3
                    X_CS(1)=VOLMAX_C
                    CONC_CL_C(KOUNT)=CL_C(KOUNT)/(X_CS(1)*1.D3)
                    CCL_OUT(KOUNT)= VOL_OUT_C*1.D3*CONC_CL_C(KOUNT-1)
**************
```

```
** OVERFLOW **  ** SEARLES **
**************

                    VOL_OUT_S=X_CS(3)-VOLMAX_S
                    CL_S(KOUNT)=CL_S(KOUNT-1)+CCL_OUT(KOUNT)-
     +                   VOL_OUT_S*1.D3*CONC_CL_S(KOUNT-1)
                    IF(CL_S(KOUNT).LT.0.D0)CL_S(KOUNT)=3.95D-4*
     +                   85.28D9*1.0D3
                    X_CS(3)=VOLMAX_S
                    CONC_CL_S(KOUNT)=CL_S(KOUNT)/(X_CS(3)*1.D3)
                    SCL_DEP(KOUNT)=0.D0
                    SUM_SCL_DEP(KOUNT)=SUM_SCL_DEP(KOUNT-1)+
     +                   SCL_DEP(KOUNT)+SLTCORR

**********************************************************
** CHINA LAKE IS OVERFLOWING, SEARLES IS STILL FILLING **
**********************************************************

                 ELSE IF(X_CS(1).GT.VOLMAX_C.AND.X_CS(3).GT.10.D0)
     +                   THEN
**************
** OVERFLOW **  ** CHINA **
**************
                    VOL_OUT_C=X_CS(1)-VOLMAX_C
                    CL_C(KOUNT)=CL_C(KOUNT-1)+TTLCL_IN-
     +                   VOL_OUT_C*1.D3*CONC_CL_C(KOUNT-1)
                    IF(CL_C(KOUNT).LT.0.D0)CL_C(KOUNT)=3.95D-4*
     +                   0.696D9*1.0D3
                    X_CS(1)=VOLMAX_C
                    CONC_CL_C(KOUNT)=CL_C(KOUNT)/(X_CS(1)*1.D3)
                    CCL_OUT(KOUNT)= VOL_OUT_C*1.D3*CONC_CL_C(KOUNT-1)
*************
** FILLING ** ** SEARLES **
*************
                    CL_S(KOUNT)=CL_S(KOUNT-1)+CCL_OUT(KOUNT)
                    CONC_CL_S(KOUNT)=CL_S(KOUNT)/(X_CS(3)*1.D3)
** SATURATION CHECK **
                    IF(CONC_CL_S(KOUNT) .GT. 6.1D0)THEN
                       CLDEP=CONC_CL_S(KOUNT)-6.1D0
                       CONC_CL_S(KOUNT)=6.1D0
                       CL_S(KOUNT)=X_CS(3)*1.D3*CONC_CL_S(KOUNT)
                       SCL_DEP(KOUNT)=CLDEP*X_CS(3)*35.453D0*
     +                      3.22D-8
                    ELSE
                       SCL_DEP(KOUNT)=0.D0
                    ENDIF

                    SUM_SCL_DEP(KOUNT)=SUM_SCL_DEP(KOUNT-1)+
     +                   SCL_DEP(KOUNT)+SLTCORR

****************************************************************
** CHINA LAKE IS BETWEEN OVERFLOW & DRY, SEARLES LAKE IS DRY **
****************************************************************

                 ELSE IF(X_CS(1) .GT. 10.D0 .AND. X_CS(3) .LT. 10.D0)
     +                   THEN

** CHINA **
                    CL_C(KOUNT)=CL_C(KOUNT-1)+TTLCL_IN
                    CONC_CL_C(KOUNT)=CL_C(KOUNT)/(X_CS(1)*1.D3)
                    IF(CONC_CL_C(KOUNT).GT.6.1D0)THEN
                       CONC_CL_C(KOUNT)=6.1D0
                       CL_C(KOUNT)=6.1D0*X_CS(1)*1.D3
                    ENDIF
                    CCL_OUT(KOUNT)=0.D0
```

```
** SEARLES **

                   CL_S(KOUNT)=0.D0
                   X_CS(3)=0.D0
                   CONC_CL_S(KOUNT)=0.D0
                   SCL_DEP(KOUNT)=CL_S(KOUNT-1)*35.453D0/1.0D3*
     +                 3.22D-8
                   SUM_SCL_DEP(KOUNT)=SUM_SCL_DEP(KOUNT-1)+
     +                 SCL_DEP(KOUNT)+SLTCORR

*****************************************************************************
** LET'S ASSUME THAT IF CHINA LAKE IS DRY, SEARLES LAKE WON'T BE OVERFLOWING **
*****************************************************************************

****************************************************
** CHINA LAKE IS DRY, SEARLES LAKE IS NOT YET DRY **
****************************************************

                ELSE IF(X_CS(1) .LE. 10.D0 .AND. X_CS(3) .GT. 10.D0)
     +                 THEN

**************
** DRY LAKE ** ** CHINA **
**************

                   CL_C(KOUNT)=0.D0
                   X_CS(1)=0.D0
                   CONC_CL_C(KOUNT)=0.D0
                   CCL_OUT(KOUNT)=0.D0

*****************
** NOT YET DRY ** SEARLES **
*****************

                   CL_S(KOUNT)=CL_S(KOUNT-1)
                   CONC_CL_S(KOUNT)=CL_S(KOUNT)/(X_CS(3)*1.D3)
** SATURATION CHECK **
                   IF(CONC_CL_S(KOUNT) .GT. 6.1D0)THEN
                      CLDEP=CONC_CL_S(KOUNT)-6.1D0
                      CONC_CL_S(KOUNT)=6.1D0
                      CL_S(KOUNT)=X_CS(3)*1.D3*CONC_CL_S(KOUNT)
                      SCL_DEP(KOUNT)=CLDEP*X_CS(3)*35.453D0*
     +                    3.22D-8
                   ELSE
                      SCL_DEP(KOUNT)=0.D0
                   ENDIF

                   SUM_SCL_DEP(KOUNT)=SUM_SCL_DEP(KOUNT-1)+
     +                 SCL_DEP(KOUNT)+SLTCORR

********************************************
** CHINA LAKE IS DRY, SEARLES LAKE IS DRY **
********************************************

                ELSE IF(X_CS(1) .LE. 10.D0 .AND. X_CS(3) .LE. 10.D0)
     +                 THEN

**************
** DRY LAKE ** ** CHINA **
**************

                   CL_C(KOUNT)=0.D0
                   X_CS(1)=0.D0
                   CONC_CL_C(KOUNT)=0.D0
                   CCL_OUT(KOUNT)=0.D0
**************
```

```
** DRY LAKE ** ** SEARLES **
**************
                        CL_S(KOUNT)=0.D0
                        X_CS(1)=0.D0
                        CONC_CL_S(KOUNT)=0.D0
                        SCL_DEP(KOUNT)=CL_S(KOUNT-1)*35.453D0/1.0D3*
     +                       3.22D-8
                        SUM_SCL_DEP(KOUNT)=SUM_SCL_DEP(KOUNT-1)+
     +                       SCL_DEP(KOUNT)+SLTCORR

                     ELSE
*******************************************
** BOTH LAKES BETWEEN DRY AND OVERFLOW **
*******************************************

** CHINA **
                        CL_C(KOUNT)=CL_C(KOUNT-1)+TTLCL_IN
                        CONC_CL_C(KOUNT)=CL_C(KOUNT)/(X_CS(1)*1.D3)
                        IF(CONC_CL_C(KOUNT).GT.6.1D0)THEN
                           CONC_CL_C(KOUNT)=6.1D0
                           CL_C(KOUNT)=6.1D0*X_CS(1)*1.D3
                        ENDIF
                        CCL_OUT(KOUNT)=0.D0

** SEARLES **
                        CL_S(KOUNT)=CL_S(KOUNT-1)
                        CONC_CL_S(KOUNT)=CL_S(KOUNT)/(X_CS(3)*1.D3)
**SATURATION CHECK**
                        IF(CONC_CL_S(KOUNT) .GT. 6.1D0)THEN
                           CLDEP=CONC_CL_S(KOUNT)-6.1D0
                           CONC_CL_S(KOUNT)=6.1D0
                           CL_S(KOUNT)=X_CS(3)*1.D3*CONC_CL_S(KOUNT)
                           SCL_DEP(KOUNT)=CLDEP*X_CS(3)*35.453D0*
     +                          3.22D-8
                        ELSE
                           SCL_DEP(KOUNT)=0.D0
                        ENDIF
                        SUM_SCL_DEP(KOUNT)=SUM_SCL_DEP(KOUNT-1)+
     +                       SCL_DEP(KOUNT)+SLTCORR
                     ENDIF

********************************************
** CALCULATE DEL DOLOMITE FROM DEL WATER **
********************************************

                     CDEL_OUT(KOUNT)=X_CS(2)
                     DELDOL_C=FDDOL(DOLTEMP,X_CS(2))
                     DELDOL_S=FDDOL(DOLTEMP,X_CS(4))

*****************************************************************************
** CALCULATE SURFACE AREAS AND SALT BALANCE FOR PANAMINT AND DEATH VALLEY **
*****************************************************************************

                     IF(CPARAM)THEN
                        TPEVAP=EVAPC_P
                        TDEVAP=EVAPC_D
                     ELSE
                        CALL FINDT(NPTS,T,INDEX,FOUND,WTIME)
                        IF(FOUND)THEN
                           TPEVAP = PEVAP(INDEX)
                           TDEVAP = DEVAP(INDEX)
                        ELSE
                           CALL PDINTERP(TIME,INDEX,TPEVAP,TDEVAP)
                        ENDIF
                     ENDIF
```

```
                  AREA_P=PQI/TPEVAP

                  IF(AREA_P .GT. AREAMAX_P)THEN
                     PQO=PQI-TPEVAP*AREAMAX_P
                     VOL_OUT_P=PQO*STEP
                     AREA_P=AREAMAX_P
                  ELSE IF(AREA_P .GT. 0.D0)THEN
                     PQO=0.D0
                  ENDIF

                  VOL_P=PVOL(AREA_P)

                  IF(VOL_P .GT. 0.D0)THEN
                     CL_P(KOUNT)=CL_P(KOUNT-1)+VOL_OUT_S*1.D3*
     +                    CONC_CL_S(KOUNT-1)-VOL_OUT_P*1.D3*
     +                    CONC_CL_P(KOUNT-1)
                     CONC_CL_P(KOUNT)=CL_P(KOUNT)/(1.D3*VOL_P)
**********************
** SATURATION CHECK **
**********************
                     IF(CONC_CL_P(KOUNT) .GT. 6.1D0)THEN
                        CLDEP=CONC_CL_P(KOUNT)-6.1D0
                        CONC_CL_P(KOUNT)=6.1D0
                        PCL_DEP=2.D0*CLDEP*1.D3*VOL_P*35.453D0/1.D3*
     +                       3.22D-8
                        SUM_PCL_DEP(KOUNT)=SUM_PCL_DEP(KOUNT-1)+
     +                       PCL_DEP
                     ELSE
                        SUM_PCL_DEP(KOUNT)=SUM_PCL_DEP(KOUNT-1)
                     ENDIF
                  ELSE
                     CL_P(KOUNT)=0.D0
                     PCL_DEP=CL_P(KOUNT-1)*2.D0*35.453/1.0D3*3.22D-8
                     SUM_PCL_DEP(KOUNT)=SUM_PCL_DEP(KOUNT-1)+PCL_DEP
                  ENDIF
******************
** DEATH VALLEY **
******************
                  DQI=PQO
                  AREA_D=DQI/TDEVAP

                  IF(AREA_D .GT. AREAMAX_D)THEN
                     DQO=DQI-TDEVAP*AREAMAX_D
                     VOL_OUT_D=DQO*STEP
                     AREA_D=AREAMAX_D
                  ELSE IF(AREA_D .GT. 0.D0)THEN
                     DQO=0.D0
                  ENDIF

                  VOL_D=DVOL(AREA_D)

                  IF(VOL_D.GT. 0.D0)THEN
                     CL_D(KOUNT)=CL_D(KOUNT-1)+VOL_OUT_P*1.D3*
     +                    CONC_CL_P(KOUNT-1)-VOL_OUT_D*1.D3*
     +                    CONC_CL_D(KOUNT-1)
                     CONC_CL_D(KOUNT)=CL_D(KOUNT)/(1.D3*VOL_D)
**********************
** SATURATION CHECK **
**********************
                     IF(CONC_CL_D(KOUNT) .GT. 6.1D0)THEN
                        CLDEP=CONC_CL_D(KOUNT)-6.D0
                        CONC_CL_D(KOUNT)=6.1D0
                        DCL_DEP=CLDEP*1.D3*VOL_D*35.453D0/1.D3

                        SUM_DCL_DEP(KOUNT)=SUM_DCL_DEP(KOUNT-1)+
     +                       DCL_DEP
```

```
                  ELSE
                     SUM_DCL_DEP(KOUNT)=SUM_DCL_DEP(KOUNT-1)
                  ENDIF
               ELSE
                  CL_D(KOUNT)=0.D0
                  DCL_DEP=CL_D(KOUNT-1)*35.453/1.0D3
                  SUM_DCL_DEP(KOUNT)=SUM_DCL_DEP(KOUNT-1)+DCL_DEP
               ENDIF

**************************************************
** WRITE PANAMINT/DEATH VALLEY STUFF TO FILE **
**************************************************

*              WRITE(73,*)TIME,AREA_P
*              WRITE(72,*)TIME,AREA_D
               WRITE(83,*)TIME,SUM_PCL_DEP(KOUNT)
*              WRITE(82,*)TIME,SUM_DCL_DEP(KOUNT)

*********************************************
** CALCULATE SUMMATION OF ALL LAKE AREAS **
*********************************************

               AREA_C=CAREA(X_CS(1))
               AREA_S=SAREA(X_CS(3))

               DELTA_O=SOAREA-OPREV
               DELTA_C=AREA_C-CPREV
               DELTA_S=AREA_S-SPREV
               DELTA_P=AREA_P-PPREV
               DELTA_D=AREA_D-DPREV


               ALLAREA(KOUNT)=DELTA_O+DELTA_C+DELTA_S+DELTA_P
     +              +DELTA_D+ALLAREA(KOUNT-1)

               IF(ALLAREA(KOUNT) .LT. 0.D0)ALLAREA(KOUNT)=0.D0

               OPREV=SOAREA
               CPREV=AREA_C
               SPREV=AREA_S
               PPREV=AREA_P
               DPREV=AREA_D

********************************************************
** STORE VALUES IN ARRAYS FOR THE PLOTTING ROUTINE **
********************************************************

               IF(GRAF)THEN
                  YQI_C(KOUNT)=QI_C
                  YQI_S(KOUNT)=QI_S
                  YDEL_C(KOUNT)=DELDOL_C
                  YDEL_S(KOUNT)=DELDOL_S
                  YAREA_C(KOUNT)=AREA_C
                  YAREA_S(KOUNT)=AREA_S
                  YSUM_SCL_DEP(KOUNT)=SUM_SCL_DEP(KOUNT)
                  YCONC(KOUNT)=CONC_CL_S(KOUNT)
                  YCL_INV(KOUNT)=CL_S(KOUNT)*0.035453D0
                  XPLT(KOUNT)=TIME
               ENDIF

***************************
** WRITE RESULTS TO FILE **
***************************

               WRITE(87,*)TIME,CONC_CL_S(KOUNT)
** SQO = PQI **
```

```
                  WRITE(91,*)TIME,PQI
                  WRITE(77,*)TIME,AREA_S
*                 WRITE(88,*)TIME,DELDOL_C
                  WRITE(78,*)TIME,DELDOL_S
*                 WRITE(86,*)TIME,QI_C
                  WRITE(76,*)TIME,QI_S
                  WRITE(74,*)TIME,SUM_SCL_DEP(KOUNT)

                  IF(SU)THEN
                     IF(TIME .LT. SAVETIME .AND. ISTCNT
     +                   .LE. NUMST)THEN
                       WRITE(70,*)'CHINA LAKE'
                       WRITE(70,*)'STARTUP TIME #',ISTCNT
                       WRITE(70,*)TIME,X_CS(1),DELDOL_C,
     +                      CONC_CL_C(KOUNT)
                       WRITE(70,*)'SEARLES LAKE'
                       WRITE(70,*)TIME,X_CS(3),AREA_S,DELDOL_S,
     +                      CONC_CL_S(KOUNT),SUM_SCL_DEP(KOUNT)
                       WRITE(70,*)'PANAMINT LAKE'
                       WRITE(70,*)TIME,AREA_P,
     +                      CONC_CL_P(KOUNT),SUM_PCL_DEP(KOUNT)
                       WRITE(70,*)'DEATH VALLEY'
                       WRITE(70,*)TIME,AREA_D,
     +                      CONC_CL_D(KOUNT),SUM_DCL_DEP(KOUNT)
                       WRITE(70,*)
                       ISTCNT=ISTCNT+1
                       SAVETIME=SUT(ISTCNT)
                     ENDIF
                  ENDIF

                  STEP=DTMAX_CS
                  GOTO 100

**********************************
** MAKE SURE THE SOLVER DOESN'T **
** OVERSTEP DESIGNATED END TIME **
**********************************

               ELSEIF(TIME-STEP.LT.TEND)THEN
                  DTMAX_CS=TIME-TEND
                  STEP=DTMAX_CS
                  GOTO 100
               ENDIF

**************************************
** ADJUST THE SIZE OF THE TIME STEP **
**************************************

            ELSEIF(DELMIN .LE. 0.1)THEN
               STEP=STEP*1.0D-1
            ELSEIF(DELMIN .GE. 4.D0)THEN
               STEP=4.D0*STEP
            ELSE
               STEP=DELMIN*STEP
            ENDIF
            IF(STEP.GT.DTMAX_CS)STEP=DTMAX_CS
            IF(STEP.LT.DTMIN_CS)STEP=DTMIN_CS
         ELSE

            CLOSE(UNIT=92)
            CLOSE(UNIT=91)
*           CLOSE(UNIT=88)
            CLOSE(UNIT=87)
*           CLOSE(UNIT=86)
            CLOSE(UNIT=85)
            CLOSE(UNIT=84)
```

```
            CLOSE(UNIT=83)
*           CLOSE(UNIT=82)
            WRITE(78,*)
            CLOSE(UNIT=78)
            CLOSE(UNIT=77)
            CLOSE(UNIT=76)
            CLOSE(UNIT=75)
            CLOSE(UNIT=74)
*           CLOSE(UNIT=73)
*           CLOSE(UNIT=72)


            IF( .NOT. GRAF)THEN
               WRITE(6,*)
               WRITE(6,*)'THE RKF SOLVED',KOUNT,' POINTS IN',ITER,' IT
     +ERATIONS'
               WRITE(6,*)'FOR CHINA AND SEARLES'
               WRITE(6,*)
               WRITE(6,*)'FINAL DEL FOR OWENS =',FINDEL
               WRITE(6,*)'FINAL VOLUME FOR OWENS =',FINVOL
               WRITE(6,*)'FINAL CL CONC, OWENS =',FINCL
               WRITE(6,*)
               WRITE(6,*)'FINAL DEL FOR CHINA =',DELDOL_C
               WRITE(6,*)'FINAL VOLUME FOR CHINA =',X_CS(1)
               WRITE(6,*)'FINAL CL CONC, CHINA =',CONC_CL_C(KOUNT)
               WRITE(6,*)
               WRITE(6,*)'FINAL DEL FOR SEARLES =',DELDOL_S
               WRITE(6,*)'FINAL VOLUME FOR SEARLES =',X_CS(3)
               WRITE(6,*)'FINAL AREA FOR SEARLES =',AREA_S
               WRITE(6,*)'FINAL CL CONC, SEARLES =',CONC_CL_S(KOUNT)
               WRITE(6,*)'TOTAL CL DEPOSITED IN SEARLES',
     +              SUM_SCL_DEP(KOUNT)
               WRITE(6,*)
               WRITE(6,*)'FINAL AREA OF PANAMINT',AREA_P
               WRITE(6,*)'FINAL CL CONC, PANAMINT',CONC_CL_P(KOUNT)
               WRITE(6,*)'TOTAL CL DEPOSITED IN PANAMINT',
     +              SUM_PCL_DEP(KOUNT)
               WRITE(6,*)'FINAL AREA OF LAKE MANLY',AREA_D
               WRITE(6,*)'FINAL TIME IS:',TIME
            ENDIF
            IF(GRAF)THEN
               WRITE(6,*)'CALLING PLOTTING ROUTINE FOR CHINA LAKE'
               WRITE(6,*)
               WRITE(6,*)'THE RKF SOLVED',KOUNT,' POINTS IN',ITER,' IT
     +ERATIONS'

               WRITE(6,*)
               CALL PLOTEMC(XMIN,XMAX,XPLT,YDEL_C,YAREA_C,YQI_C,KOUNT)

               CALL ENDPL(0)
               CALL DONEPL

               WRITE(6,*)'CALLING PLOTTING ROUTINE FOR SEARLES LAKE'
               WRITE(6,*)
               WRITE(6,*)'THE RKF SOLVED',KOUNT,' POINTS IN',ITER,' IT
     +ERATIONS'
               WRITE(6,*)
               CALL PLOTEMS(XMIN,XMAX,XPLT,YDEL_S,YAREA_S,YQI_S,KOUNT)

               CALL ENDPL(0)
               CALL DONEPL

**********************************************************************
** PLOT CL CONCENTRATION, INVENTORY & DEPOSITION FOR SEARLES LAKE **
**********************************************************************
```

```
              CALL PLOTEM_INV(XMIN,XMAX,XPLT,YCONC,YCL_INV,
     +             YSUM_SCL_DEP,KOUNT)

              CALL ENDPL(0)
              CALL DONEPL

****************************
** PLOT SUM OF LAKE AREAS **
****************************

              CALL PLOTEM_SUM(XMIN,XMAX,XPLT,ALLAREA,KOUNT)

              CALL ENDPL(0)
              CALL DONEPL

 1801         WRITE(6,*)
              WRITE(6,*)'WOULD YOU LIKE TO SEE SOME OF THE PLOTS
     +AGAIN ?  [1=YES, 0=NO]'
              READ(5,*)IREPLOT

              IF(IREPLOT .EQ. 1)THEN
                 WRITE(6,*)
                 WRITE(6,*)'PICK A PLOT'
                 WRITE(6,*)'[1] CHINA LAKE DEL, AREA, INFLOW'
                 WRITE(6,*)'[2] SEARLES LAKE DEL, AREA, INFLOW'
                 WRITE(6,*)'[3] SEARLES LAKE SALT STUFF'
                 WRITE(6,*)'[4] TOTAL SURFACE AREA'
                 WRITE(6,*)
                 READ(5,*)IPLOTNUM
                 IF(IPLOTNUM .EQ. 1)THEN
                    WRITE(6,*)'CALL PLOTTING ROUTINE FOR CHINA LAKE'
                    WRITE(6,*)
                    CALL PLOTEMC(XMIN,XMAX,XPLT,YDEL_C,YAREA_C,
     +                   YQI_C,KOUNT)
                    CALL ENDPL(0)
                    CALL DONEPL
                 ELSE IF(IPLOTNUM .EQ. 2)THEN
                    WRITE(6,*)'CALL PLOT ROUTINE FOR SEARLES LAKE'
                    WRITE(6,*)
                    CALL PLOTEMS(XMIN,XMAX,XPLT,YDEL_S,YAREA_S,
     +                   YQI_S,KOUNT)
                    CALL ENDPL(0)
                    CALL DONEPL
                 ELSE IF(IPLOTNUM .EQ. 3)THEN
                    CALL PLOTEM_INV(XMIN,XMAX,XPLT,YCONC,YCL_INV,
     +                   YSUM_SCL_DEP,KOUNT)
                    CALL ENDPL(0)
                    CALL DONEPL
                 ELSE IF(IPLOTNUM .EQ. 4)THEN
                    CALL PLOTEM_SUM(XMIN,XMAX,XPLT,ALLAREA,KOUNT)
                    CALL ENDPL(0)
                    CALL DONEPL
                 ENDIF
                 GOTO 1801
              ENDIF


              WRITE(6,*)
              WRITE(6,*)'FINAL DEL FOR OWENS =',FINDEL
              WRITE(6,*)'FINAL VOLUME FOR OWENS =',FINVOL
              WRITE(6,*)'FINAL CL CONC, OWENS =',FINCL
              WRITE(6,*)
              WRITE(6,*)'FINAL DEL FOR CHINA =',DELDOL_C
              WRITE(6,*)'FINAL VOLUME FOR CHINA =',X_CS(1)
              WRITE(6,*)'FINAL CL CONC, CHINA =',CONC_CL_C(KOUNT)
              WRITE(6,*)
```

```
                WRITE(6,*)'FINAL DEL FOR SEARLES =',DELDOL_S
                WRITE(6,*)'FINAL VOLUME FOR SEARLES =',X_CS(3)
                WRITE(6,*)'FINAL AREA FOR SEARLES =',AREA_S
                WRITE(6,*)'FINAL CL CONC, SEARLES =',CONC_CL_S(KOUNT)
                WRITE(6,*)'TOTAL CL DEPOSITED IN SEARLES',
     +               SUM_SCL_DEP(KOUNT)
                WRITE(6,*)
                WRITE(6,*)'FINAL AREA OF PANAMINT',AREA_P
                WRITE(6,*)'FINAL CL CONC, PANAMINT',CONC_CL_P(KOUNT)
                WRITE(6,*)'TOTAL CL DEPOSITED IN PANAMINT',
     +               SUM_PCL_DEP(KOUNT)
                WRITE(6,*)'FINAL AREA OF LAKE MANLY',AREA_D
                WRITE(6,*)'FINAL TIME IS:',TIME
             ENDIF
             RETURN
          ENDIF
100    CONTINUE
       IF(ITER .GT. ITMAX)WRITE(6,*)'MAX # OF ITERATIONS EXCEEDED'
       IF( .NOT. GRAF)THEN
          WRITE(6,*)
          WRITE(6,*)'THE RKF SOLVED',KOUNT,' POINTS IN',ITER,' IT
     +ERATIONS'
          WRITE(6,*)'FOR CHINA AND SEARLES'
          WRITE(6,*)
          WRITE(6,*)'FINAL DEL FOR OWENS =',FINDEL
          WRITE(6,*)'FINAL VOLUME FOR OWENS =',FINVOL
          WRITE(6,*)'FINAL CL CONC, OWENS =',FINCL
          WRITE(6,*)
          WRITE(6,*)'FINAL DEL FOR CHINA =',DELDOL_C
          WRITE(6,*)'FINAL VOLUME FOR CHINA =',X_CS(1)
          WRITE(6,*)'FINAL CL CONC, CHINA =',CONC_CL_C(KOUNT)
          WRITE(6,*)
          WRITE(6,*)'FINAL DEL FOR SEARLES =',DELDOL_S
          WRITE(6,*)'FINAL VOLUME FOR SEARLES =',X_CS(3)
          WRITE(6,*)'FINAL AREA FOR SEARLES =',AREA_S
          WRITE(6,*)'FINAL CL CONC, SEARLES =',CONC_CL_S(KOUNT)
          WRITE(6,*)'TOTAL CL DEPOSITED IN SEARLES',
     +          SUM_SCL_DEP(KOUNT)
          WRITE(6,*)
          WRITE(6,*)'FINAL AREA OF PANAMINT',AREA_P
          WRITE(6,*)'FINAL CL CONC, PANAMINT',CONC_CL_P(KOUNT)
          WRITE(6,*)'TOTAL CL DEPOSITED IN PANAMINT',
     +          SUM_PCL_DEP(KOUNT)
          WRITE(6,*)'FINAL AREA OF LAKE MANLY',AREA_D
          WRITE(6,*)'FINAL TIME IS:',TIME

       ENDIF
             CLOSE(UNIT=92)
*            CLOSE(UNIT=91)
*            CLOSE(UNIT=88)
*            CLOSE(UNIT=87)
*            CLOSE(UNIT=86)
             CLOSE(UNIT=85)
             CLOSE(UNIT=84)
             CLOSE(UNIT=83)
*            CLOSE(UNIT=82)
             CLOSE(UNIT=78)
             CLOSE(UNIT=77)
             CLOSE(UNIT=76)
             CLOSE(UNIT=75)
             CLOSE(UNIT=74)
*            CLOSE(UNIT=73)
*            CLOSE(UNIT=72)


          IF(GRAF)THEN
```

```
            WRITE(6,*)'CALLING PLOTTING ROUTINE FOR CHINA LAKE'
            WRITE(6,*)
            WRITE(6,*)'THE RKF SOLVED',KOUNT,' POINTS IN',ITER,' ITERA
     +TIONS'
            WRITE(6,*)
            CALL PLOTEMC(XMIN,XMAX,XPLT,YDEL_C,YAREA_C,YQI_C,KOUNT)

*************************************
** FINISH PLOTTING STUFF FOR CHINA **
*************************************

            CALL ENDPL(0)
            CALL DONEPL

            WRITE(6,*)'CALLING PLOTTING ROUTINE FOR SEARLES LAKE'
            WRITE(6,*)
            WRITE(6,*)'THE RKF SOLVED',KOUNT,' POINTS IN',ITER,' ITERA
     +TIONS'
            WRITE(6,*)
            CALL PLOTEMS(XMIN,XMAX,XPLT,YDEL_S,YAREA_S,YQI_S,KOUNT)

***************************************
** FINISH PLOTTING STUFF FOR SEARLES **
***************************************

            CALL ENDPL(0)
            CALL DONEPL


*********************************************************************
** PLOT CL CONCENTRATION, INVENTORY & DEPOSITION FOR SEARLES LAKE **
*********************************************************************

            CALL PLOTEM_INV(XMIN,XMAX,XPLT,YCONC,YCL_INV,
     +           YSUM_SCL_DEP,KOUNT)

            CALL ENDPL(0)
            CALL DONEPL

****************************
** PLOT SUM OF LAKE AREAS **
****************************

               CALL PLOTEM_SUM(XMIN,XMAX,XPLT,ALLAREA,KOUNT)

               CALL ENDPL(0)
               CALL DONEPL

               WRITE(6,*)
               WRITE(6,*)'WOULD YOU LIKE TO SEE SOME OF THE PLOTS
     +AGAIN ?  [1=YES, 0=NO]'
               READ(5,*)IREPLOT

               IF(IREPLOT .EQ. 1)THEN
                  WRITE(6,*)
                  WRITE(6,*)'PICK A PLOT'
                  WRITE(6,*)'[1] CHINA LAKE DEL, AREA, INFLOW'
                  WRITE(6,*)'[2] SEARLES LAKE DEL, AREA, INFLOW'
                  WRITE(6,*)'[3] SEARLES LAKE SALT STUFF'
                  WRITE(6,*)'[4] TOTAL SURFACE AREA'
                  WRITE(6,*)
                  READ(5,*)IPLOTNUM
                  IF(IPLOTNUM .EQ. 1)THEN
                     WRITE(6,*)'CALL PLOTTING ROUTINE FOR CHINA LAKE'
                     WRITE(6,*)
                     CALL PLOTEMC(XMIN,XMAX,XPLT,YDEL_C,YAREA_C,
```

```
     +                  YQI_C,KOUNT)
                    CALL ENDPL(0)
                    CALL DONEPL
                 ELSE IF(IPLOTNUM .EQ. 2)THEN
                    WRITE(6,*)'CALL PLOT ROUTINE FOR SEARLES LAKE'
                    WRITE(6,*)
                    CALL PLOTEMS(XMIN,XMAX,XPLT,YDEL_S,YAREA_S,
     +                  YQI_S,KOUNT)
                    CALL ENDPL(0)
                    CALL DONEPL
                 ELSE IF(IPLOTNUM .EQ. 3)THEN
                    CALL PLOTEM_INV(XMIN,XMAX,XPLT,YCONC,YCL_INV,
     +                  YSUM_SCL_DEP,KOUNT)
                    CALL ENDPL(0)
                    CALL DONEPL
                 ELSE IF(IPLOTNUM .EQ. 4)THEN
                    CALL PLOTEM_SUM(XMIN,XMAX,XPLT,ALLAREA,KOUNT)
                    CALL ENDPL(0)
                    CALL DONEPL
                 ENDIF
                 GOTO 1801
              ENDIF

              WRITE(6,*)
              WRITE(6,*)'FINAL DEL FOR OWENS =',FINDEL
              WRITE(6,*)'FINAL VOLUME FOR OWENS =',FINVOL
              WRITE(6,*)'FINAL CL CONC, OWENS =',FINCL
              WRITE(6,*)
              WRITE(6,*)'THE FINAL DEL FOR CHINA =',DELDOL_C
              WRITE(6,*)'FINAL VOLUME FOR CHINA =',X_CS(1)
              WRITE(6,*)'FINAL CL CONC, CHINA =',CONC_CL_C(KOUNT)
              WRITE(6,*)
              WRITE(6,*)'FINAL DEL FOR SEARLES =',DELDOL_S
              WRITE(6,*)'FINAL VOLUME FOR SEARLES =',X_CS(3)
              WRITE(6,*)'FINAL AREA FOR SEARLES =',AREA_S
              WRITE(6,*)'FINAL CL CONC, SEARLES =',CONC_CL_S(KOUNT)
              WRITE(6,*)'TOTAL CL DEPOSITED IN SEARLES',
     +            SUM_SCL_DEP(KOUNT)
              WRITE(6,*)
              WRITE(6,*)'FINAL AREA OF PANAMINT',AREA_P
              WRITE(6,*)'FINAL CL CONC, PANAMINT',CONC_CL_P(KOUNT)
              WRITE(6,*)'TOTAL CL DEPOSITED IN PANAMINT',
     +            SUM_PCL_DEP(KOUNT)
              WRITE(6,*)'FINAL AREA OF LAKE MANLY',AREA_D
              WRITE(6,*)'FINAL TIME IS:',TIME

         ENDIF
      END

*****************************************************************************
*****************************************************************************
** A FUNCTION TO CALCULATE DV/DT AND DDEL/DT FOR CHINA AND SEARLES LAKE **
*****************************************************************************
*****************************************************************************

      DOUBLE PRECISION FUNCTION F_CS(I,TERM,T,KNT,KOUNT,TBEG,COAL)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      LOGICAL FOUND,GUESS,GRAF,CPARAM,GUESS2,FOUND2,COAL,GU,GD,SU

      PARAMETER(NUMA=15000,NUMB=2500)
      PARAMETER(VOLMAX_C=0.696D9,VOLMAX_S=85.28D9)

      COMMON/BOTH/CL(NUMA),TSOD(10),TCL(10),OTIME(NUMA),TCO3(10),
     + OQO(NUMA),QO(10),CONC_CL(NUMA),DOLTEMP,DELDOL,TODELI,TOTEMP,
     + ODEL_OUT(NUMA),OCL_OUT(NUMA),PEVAP(NUMB),DEVAP(NUMB),
```

```
     +  SUM_PCL_DEP(NUMA),SUM_DCL_DEP(NUMA),AREA_P,AREA_D,
     +  AREA(NUMA),SOAREA,ALLAREA(NUMA),CONC_CL_P(NUMA),CONC_CL_D(NUMA),
     +  CL_P(NUMA),CL_D(NUMA),INCHOICE

        COMMON/BOTH2/CL_C(NUMA),TSOD_C(10),TCL_C(10),TIME,TCO3_C(10),
     +  CQO(10),GRAF,CONC_CL_C(NUMA),DOLTEMP_C,DELDOL_C,
     +  TCDELI,TCTEMP,CDEL_OUT(NUMA),CCL_OUT(NUMA),CL_S(NUMA),NOPTS,
     +  TSOD_S(10),TCL_S(10),TCO3_S(10),SQO(10),CONC_CL_S(NUMA),
     +  DOLTEMP_S,DELDOL_S,TSDELI,TSTEMP,SCL_DEP(NUMA),TTLCL_IN,NPTS,
     +  SUM_SCL_DEP(NUMA),QI_C,QI_S,PQI

        COMMON/FIRST/A,B,WTIME(NUMB),OTEMP(NUMB),OEVAP(NUMB),
     +  OHUM(500),OPRECIP(NUMB),ODELP(NUMB),ODELA(NUMB),ODELI(NUMB),
     +  GUESS,C,TEMPC,EVAPC,PRECIPC,DELAC,DELIC,CPARAM,DELPC,
     +  EVAPC_P,EVAPC_D

        COMMON/SECOND/CTEMP(NUMB),CEVAP(NUMB),CHUM(500),CPRECIP(NUMB),
     +  CDELP(NUMB),CSDELA(NUMB),GUESS2,TEMPC_C,EVAPC_C,
     +  PRECIPC_C,DELAC_C,DELPC_C,STEMP(NUMB),SEVAP(NUMB),SHUM(500),
     +  SPRECIP(NUMB),SDELP(NUMB),TEMPC_S,EVAPC_S,
     +  PRECIPC_S,DELAC_S,DELPC_S

        COMMON/NEW/OPREV,CPREV,SPREV,PPREV,DPREV

        COMMON/HIST/QIHIST(1000),HUMTIME(500),NHPTS,SU,SUT(15),
     +  SAVETIME,NUMST,QITIME(1000),NQPTS

        COMMON/FIX/GU,GD,AVGIN,ITGCNT

        DIMENSION TERM(4),TCONC_CL_C(10),V_OUT_C(10),
     +  TQI_C(10),TCONC_CL_S(10),V_OUT_S(10),TQI_S(10)


        IF(I.EQ.1)THEN

************************************
** CALCULATE DV/DT FOR CHINA LAKE **
************************************

           VOL_C = TERM(1)
           IF(VOL_C .LE. 10.D0)VOL_C=0.D0

10         FORMAT(A,D12.5)

*******************************
** CONSTANT PARAMETER OPTION **
*******************************

           IF(CPARAM)THEN
              TCTEMP=TEMPC_C
              TCEVAP=EVAPC_C
              TCPRECIP=PRECIPC_C
              TCDELP=DELPC_C
              TCDELA=DELAC_C
              CALL FINDHT(NHPTS,T,IHNDEX,HUMTIME)
              CALL HUMTERP(T,IHNDEX,TCHUM,CHUM,HUMTIME)
           ELSE

**********************************************************************
** DETERMINE VALUES OF NECESSARY PARAMETERS BY ASSIGNING VALUES OR  **
** INTERPOLATING BETWEEN GIVEN VALUES                               **
**********************************************************************

              CALL FINDT(NPTS,T,INDEX,FOUND,WTIME)
              CALL FINDHT(NHPTS,T,IHNDEX,HUMTIME)
```

```
            IF(FOUND)THEN
               TCTEMP = CTEMP(INDEX)
               TCEVAP = CEVAP(INDEX)
               TCPRECIP = CPRECIP(INDEX)
               TCDELP = CDELP(INDEX)
               TCDELA = CSDELA(INDEX)
            ELSE
               CALL CINTERP(T,INDEX,TCTEMP,TCEVAP,TCPRECIP,
     +                  TCDELP,TCDELA)
            ENDIF
            CALL HUMTERP(T,IHNDEX,TCHUM,CHUM,HUMTIME)
         ENDIF

***********************************************************************************
** INTERPOLATE TO FIND VALUES FOR PARAMETERS CALCULATED DURING OWENS ROUTINE **
***********************************************************************************

         CALL FINDT2(NOPTS,T,INDEX2,FOUND2,OTIME)
         IF(FOUND2)THEN
            QI=OQO(INDEX2)
            TCDELI=ODEL_OUT(INDEX2)
            TAREA=AREA(INDEX2)
         ELSE
            CALL C2INTERP(T,INDEX2,TCDELI,QI,TAREA)
         ENDIF

         IF(QI .LT. 0.D0)QI=0.D0


**********************************************************************
** "REMEMBER" TEMP AT END OF TIMESTEP TO CALCULATE DEL DOLOMITE **
**********************************************************************

         IF(KNT .EQ. 5)DOLTEMP_C=TCTEMP

         IF(KNT .EQ. 5)SOAREA=TAREA
************************************************************
** "REMEMBER" QI AT END OF TIMESTEP FOR GRAPHICS ARRAY **
************************************************************

         IF(KNT .EQ. 5)QI_C=QI

         ETIME = TBEG-T

*****************************
** CALCULATE AREA OF LAKE **
*****************************

         AREA_C = CAREA(VOL_C)

******************************
** CALCULATE PRECIPITATION **
******************************

         QP = AREA_C*TCPRECIP

***********************************
** THE INFAMOUS "SALT" BALANCE **
***********************************

         IF(T .EQ. TIME)THEN
            CALL C3INTERP(T,INDEX2,TIMECL_IN)
            TCL_C(KNT)=CL_C(KOUNT)
            TCONC_CL_C(KNT)=CONC_CL_C(KOUNT)
```

```
****************************************************
** CALCULATE TOTAL OUTFLOW VOLUME FROM TIME TO T **
** AND ADJUST IF VOL EXCEEDS VOLMAX             **
****************************************************

           ELSE
              IF(VOL_C .LE. VOLMAX_C)THEN
                 V_OUT_C(KNT)=0.D0
              ELSE
                 V_OUT_C(KNT) = VOL_C-VOLMAX_C
                 VOL_C=VOLMAX_C
              ENDIF

**********************************************************
**  CALCULATE CONCENTRATIONS FOR INTERMEDIATE TIME "T" **
**********************************************************

********************************************************
** CALCULATE HOW MUCH 'SALT' CAME IN FROM OWENS LAKE **
********************************************************

              CALL C3INTERP(T,INDEX2,CL_IN)
              TCL_IN=CL_IN-TIMECL_IN
              IF(KNT .EQ. 5)TTLCL_IN=TCL_IN
              TCL_C(KNT) = TCL_IN+CL_C(KOUNT)-V_OUT_C(KNT)*
     +             CONC_CL_C(KOUNT)*1.D3
              IF(TCL_C(KNT).LT.0.D0)TCL_C(KNT)=3.95D-4*0.696D9*1.D3

**********************************************************************
** CONCENTRATION UNITS ARE MOLARITY SO VOL MUST BE MULT BY 1000 TO **
** CONVERT M^3 TO LITERS **
**********************************************************************

              IF(VOL_C .GT. 10.D0)THEN
                 TCONC_CL_C(KNT)=TCL_C(KNT)/(VOL_C*1.0D3)
              ELSE
                 TCONC_CL_C(KNT)=0.D0
                 TCL_C(KNT)=0.D0
              ENDIF

************************************
** CHECK FOR CHLORIDE SATURATION **
************************************

              IF(TCONC_CL_C(KNT) .GT. 6.1D0)THEN
                 TCONC_CL_C(KNT)=6.1D0
                 TCL_C(KNT)=6.1D0*(VOL_C*1.D3)
              ENDIF

**************************************************************
** CALCULATE HOW MUCH SALT GOES TO SEARLES FROM TIME TO T **
**************************************************************


              TCCL_OUT=V_OUT_C(KNT)*1.D3*CONC_CL_C(KOUNT)

           ENDIF

*********************************************
**  CALCULATE BACK-CONDENSATION FLUX (QC) **
*********************************************

           IF(VOL_C .GT. 10.D0 .AND. TCONC_CL_C(KNT) .GT. 0.D0)THEN
              PHI=FPHI(TCONC_CL_C(KNT),SUM)
           ELSE
              PHI=0.D0
```

```
          ENDIF
          AW=DEXP(-18.D0*PHI*SUM*0.5D0/1.D3)

********************************
** CALCULATE EVAPORATION (QE) **
********************************

          QE = AREA_C*TCEVAP*AW

          IF(VOL_C .GT. 10.D0)THEN
             QC=(TCHUM*QE)/AW
          ELSE
             QC=0.D0
          ENDIF

*************************************
**  CALCULATE DV/DT IF VOL IS ZERO **
*************************************

          IF(TERM(1) .LE. 10.D0)THEN
             IF(GUESS)THEN
**********************************
** WRITE INITIAL VALUES TO FILE **
**********************************
*               WRITE(87,*)TBEG,AREA_C
*               WRITE(86,*)TBEG,OQO(1)
                DELDOL_C=FDDOL(TCTEMP,TERM(2))
*               WRITE(88,*)TBEG,DELDOL_C
                GUESS=.FALSE.
             ENDIF

             CQO(KNT)=0.D0

             F_CS=QI

             IF(F_CS .LE. 0.D0)THEN
                DV_DT=0.D0
             ELSE
                DV_DT=F_CS
             ENDIF

*************************************************
**  CALCULATE DV/DT IF VOL IS < VOLMAX BUT > 0 **
*************************************************

          ELSE IF(TERM(1)  .LT. VOLMAX_C)THEN

             IF(GUESS)THEN
**********************************
** WRITE INITIAL VALUES TO FILE **
**********************************
*               WRITE(87,*)TBEG,AREA_C
*               WRITE(86,*)TBEG,OQO(1)
                DELDOL_C=FDDOL(TCTEMP,TERM(2))
*               WRITE(88,*)TBEG,DELDOL_C
                GUESS=.FALSE.
             ENDIF


**********************
**  CALCULATE DV/DT **
**********************

             F_CS=QI+QC-QE+QP
             DV_DT=F_CS
             CQO(KNT)=0.D0
```

```
****************************************************
**  CALCULATE DV/DT IF VOL IS GREATER THAN VOLMAX **
****************************************************

           ELSE

              IF(GUESS)THEN
**********************************
** WRITE INITIAL VALUES TO FILE **
**********************************
*                WRITE(87,*)TBEG,AREA_C
*                WRITE(86,*)TBEG,OQO(1)
                 DELDOL_C=FDDOL(TCTEMP,TERM(2))
*                WRITE(88,*)TBEG,DELDOL_C
                 GUESS=.FALSE.
              ENDIF


******************
** CALCULATE QO **
******************

             CQO(KNT)=QI+QC-QE+QP

*********************
** CALCULATE DV/DT **
*********************

              IF(CQO(KNT) .LT. 0.D0)THEN
                 CQO(KNT)=0.D0
                 F_CS=QI+QC-QE+QP
                 DV_DT=F_CS
              ELSE
                 F_CS=CQO(KNT)
                 DV_DT=0.D0
              ENDIF
           ENDIF

***************************************
** CALCULATE DDEL/DT FOR CHINA LAKE **
***************************************

        ELSE IF(I.EQ.2)THEN

           DELL_C = TERM(2)

           IF(VOL_C .LE. 10.D0)THEN
              F_CS=0.D0

           ELSE

******************************************
** CALCULATE ISOTOPIC ENRICHMENT FACTOR **
******************************************

              EPS = FEPS(TCTEMP)

********************************************
** CALCULATE DEL OF THE BACK-CONDENSATION **
********************************************

              CDELC =EPS*(1.D0+(TCDELA/1.D3))+TCDELA

***************************************
** CALCULATE DEL OF THE EVAPORATION **
```

```
***************************************

             CDELE = DELE(DELL_C,EPS,TCHUM)

*****************************************************
** SET DEL OF THE OUTFLOW EQUAL TO DEL OF THE LAKE **
*****************************************************

             CDELO = DELL_C

             F_CS=(QI*TCDELI+QC*CDELC+QP*TCDELP-CQO(KNT)*CDELO-QE*
     +            CDELE-DELL_C*DV_DT)/VOL_C

          ENDIF

       ELSE IF(I.EQ.3)THEN

***************************************
** CALCULATE DV/DT FOR SEARLES LAKE **
***************************************

          VOL_S = TERM(3)
          IF(VOL_S .LE. 10.D0)VOL_S=0.D0

*******************************
** CONSTANT PARAMETER OPTION **
*******************************

          IF(CPARAM)THEN
             TSTEMP=TEMPC_S
             TSEVAP=EVAPC_S
             TSPRECIP=PRECIPC_S
             TSDELP=DELPC_S
             TSDELA=DELAC_S
             IF(COAL)THEN
                CALL FINDT(NPTS,T,INDEX,FOUND,WTIME)
                CALL FINDHT(NHPTS,T,IHNDEX,HUMTIME)
             ENDIF
             CALL HUMTERP(T,IHNDEX,TSHUM,SHUM,HUMTIME)
          ELSE

*********************************************************************
** DETERMINE VALUES OF NECESSARY PARAMETERS BY ASSIGNING VALUES OR  **
** INTERPOLATING BETWEEN GIVEN VALUES                               **
*********************************************************************

             IF(COAL)THEN
                CALL FINDT(NPTS,T,INDEX,FOUND,WTIME)
                CALL FINDHT(NHPTS,T,IHNDEX,HUMTIME)
             ENDIF

             IF(FOUND)THEN
                TSTEMP = STEMP(INDEX)
                TSEVAP = SEVAP(INDEX)
                TSPRECIP = SPRECIP(INDEX)
                TSDELP = SDELP(INDEX)
             ELSE

******************************************************************************
** INTERPOLATE TO FIND VALUES FOR PARAMETERS CALCULATED DURING OWENS ROUTINE **
******************************************************************************

                CALL SINTERP(T,INDEX,TSTEMP,TSEVAP,TSPRECIP,
     +               TSDELP,TSDELA)
             ENDIF
             CALL HUMTERP(T,IHNDEX,TSHUM,SHUM,HUMTIME)
```

```
          ENDIF

******************************************************************
** THESE SEARLES PARAMETERS ARE EQUAL TO THEIR CHINA COUNTERPARTS **
******************************************************************

          IF(COAL)THEN
             CALL FINDT2(NOPTS,T,INDEX2,FOUND2,OTIME)
             IF(FOUND2)THEN
                QI=OQO(INDEX2)
                TSDELI=ODEL_OUT(INDEX2)
                TAREA=AREA(INDEX2)
             ELSE
                CALL C2INTERP(T,INDEX2,TSDELI,QI,TAREA)
             ENDIF
             IF(QI .LT. 0.D0)QI=0.D0
             IF(GU)THEN
                QI=((QI+AVGIN)/2.D0)*(1.D0-(ITGCNT/100.D0))
             ENDIF
          ELSE
             TSDELI = CDELO
          ENDIF

*****************************************************************
** "REMEMBER" TEMP AT END OF TIMESTEP TO CALCULATE DEL DOLOMITE **
*****************************************************************

          IF(KNT .EQ. 5)DOLTEMP_S=TCTEMP

          IF(KNT .EQ. 5)SOAREA=TAREA

          ETIME = TBEG-T

****************************
** CALCULATE AREA OF LAKE **
****************************

          AREA_S = SAREA(VOL_S)

*****************************
** CALCULATE PRECIPITATION **
*****************************

          QP = AREA_S*TSPRECIP

************************
** QI = QO FROM CHINA **
************************

          IF(.NOT. COAL)THEN
             QI =CQO(KNT)
          ENDIF

          IF(GD)THEN
             QI=((QI+AVGIN)/2.D0)*(1.D0-(ITGCNT/100.D0))
          ENDIF

          IF(KNT .EQ. 5)QI_S=QI

*********************************
** THE INFAMOUS "SALT" BALANCE **
*********************************

          IF(T .EQ. TIME)THEN

**************************************************************
```

```
** IF CHINA AND SEARLES COALESCE, THE INITIAL SALT BALANCE **
** IS CALCULATED IN THE RKF_CS SOLVER                      **
*************************************************************

              TCL_S(KNT)=CL_S(KOUNT)
              TCONC_CL_S(KNT)=CONC_CL_S(KOUNT)

***************************************************
** CALCULATE TOTAL OUTFLOW VOLUME FROM TIME TO T **
** AND ADJUST IF VOL EXCEEDS VOLMAX              **
***************************************************

           ELSE
              IF(VOL_S .LE. VOLMAX_S)THEN
                 V_OUT_S(KNT)=0.D0
              ELSE
                 V_OUT_S(KNT) = VOL_S-VOLMAX_S
                 VOL_S=VOLMAX_S
              ENDIF

*********************************************************
**  CALCULATE CONCENTRATIONS FOR INTERMEDIATE TIME "T" **
*********************************************************

*******************************************************
** CALCULATE HOW MUCH 'SALT' CAME IN FROM CHINA LAKE **
*******************************************************

              IF(COAL)THEN
                 CALL C3INTERP(TIME,INDEX2,TIMECL_IN)
                 CALL C3INTERP(T,INDEX2,CL_IN)
                 TCL_IN=CL_IN-TIMECL_IN
                 IF(KNT .EQ. 5)TTLCL_IN=TCL_IN
              ELSE
                 TCL_IN=TCCL_OUT
              ENDIF

              TCL_S(KNT) = TCL_IN+CL_S(KOUNT)-V_OUT_S(KNT)*
     +           CONC_CL_S(KOUNT)*1.D3

              IF(TCL_S(KNT).LT.0.D0)TCL_S(KNT)=3.95D-4*85.28D9*1.D3

*********************************************************************
** CONCENTRATION UNITS ARE MOLARITY SO VOL MUST BE MULT BY 1000 TO **
** CONVERT M^3 TO LITERS **
*********************************************************************

              IF(VOL_S .GT. 10.D0)THEN
                 TCONC_CL_S(KNT)=TCL_S(KNT)/(VOL_S*1.0D3)
              ELSE
                 TCONC_CL_S(KNT)=0.D0
              ENDIF

*********************************
** CHECK FOR CHLORIDE SATURATION **
*********************************

              IF(TCONC_CL_S(KNT) .GT. 6.1D0)THEN
                 TCONC_CL_S(KNT)=6.1D0
                 TCL_S(KNT)=6.1D0*(VOL_S*1.D3)
              ENDIF

           ENDIF

*******************************************
**  CALCULATE BACK-CONDENSATION FLUX (QC) **
```

```
*********************************************

          IF(VOL_S .GT. 10.D0 .AND. TCONC_CL_S(KNT) .GT. 0.D0)THEN
             PHI=FPHI(TCONC_CL_S(KNT),SUM)
          ELSE
             PHI=0.D0
          ENDIF
          AW=DEXP(-18.D0*PHI*SUM*0.5D0/1.D3)

*********************************
** CALCULATE EVAPORATION (QE) **
*********************************

          QE = AREA_S*TSEVAP*AW

          IF(VOL_S .GT. 10.D0)THEN
             QC=(TSHUM*QE)/AW
          ELSE
             QC=0.D0
          ENDIF

**************************************
**  CALCULATE DV/DT IF VOL IS ZERO **
**************************************

          IF(TERM(3) .LE. 10.D0)THEN
             IF(GUESS2)THEN
                WRITE(77,*)TBEG,AREA_S
                WRITE(76,*)TBEG,QI
                WRITE(91,*)TBEG,0.0
                DELDOL_S=FDDOL(TSTEMP,TERM(4))
                WRITE(78,*)TBEG,DELDOL_S
                GUESS2=.FALSE.
                ALLAREA(1)=AREA(1)+AREA_S+AREA_C+AREA_P+AREA_D
                OPREV=AREA(1)
                CPREV=AREA_C
                SPREV=AREA_S
                PPREV=AREA_P
                DPREV=AREA_P
             ENDIF

             SQO(KNT)=0.D0

***************************************************
** SET SEARLES OUTFLUX EQUAL TO PANAMINT INFLUX **
***************************************************

             IF(KNT .EQ. 5)PQI=SQO(KNT)

             IF(QI .LT. 0.D0)QI=0.D0

             F_CS=QI

             IF(F_CS .LE. 0.D0)THEN
                DV_DT=0.D0
             ELSE
                DV_DT=F_CS
             ENDIF

**************************************************
**  CALCULATE DV/DT IF VOL IS < VOLMAX BUT > 0 **
**************************************************

          ELSE IF(TERM(3)  .LT. VOLMAX_S)THEN

**********************************
```

```
** WRITE INITIAL VALUES TO FILE **
**********************************

              IF(GUESS2)THEN
                 WRITE(77,*)TBEG,AREA_S
                 WRITE(76,*)TBEG,QI
                 WRITE(91,*)TBEG,0.0
                 DELDOL_S=FDDOL(TSTEMP,TERM(4))
                 WRITE(78,*)TBEG,DELDOL_S
                 ALLAREA(1)=AREA(1)+AREA_S+AREA_C+AREA_P+AREA_D
                 OPREV=AREA(1)
                 CPREV=AREA_C
                 SPREV=AREA_S
                 PPREV=AREA_P
                 DPREV=AREA_P
                 GUESS2=.FALSE.
              ENDIF

**********************
**  CALCULATE DV/DT **
**********************

              F_CS=QI+QC-QE+QP
              DV_DT=F_CS
              SQO(KNT)=0.D0
              IF(KNT .EQ. 5)PQI=SQO(KNT)

****************************************************
**  CALCULATE DV/DT IF VOL IS GREATER THAN VOLMAX **
****************************************************

           ELSE

              IF(GUESS2)THEN
                 WRITE(77,*)TBEG,AREA_S
                 WRITE(76,*)TBEG,QI
                 WRITE(91,*)TBEG,QI+QC-QE+QP
                 DELDOL_S=FDDOL(TSTEMP,TERM(4))
                 WRITE(78,*)TBEG,DELDOL_S
                 ALLAREA(1)=AREA(1)+AREA_S+AREA_C+AREA_P+AREA_D
                 GUESS2=.FALSE.
                 OPREV=AREA(1)
                 CPREV=AREA_C
                 SPREV=AREA_S
                 PPREV=AREA_P
                 DPREV=AREA_P
              ENDIF

******************
** CALCULATE QO **
******************

             SQO(KNT)=QI+QC-QE+QP

*********************
** CALCULATE DV/DT **
*********************

              IF(SQO(KNT) .LT. 0.D0)THEN
                 SQO(KNT)=0.D0
                 IF(KNT .EQ. 5)PQI=SQO(KNT)
                 F_CS=QI+QC-QE+QP
                 DV_DT=F_CS
              ELSE
```

```
              F_CS=SQO(KNT)
              IF(KNT .EQ. 5)PQI=SQO(KNT)
              DV_DT=0.D0
           ENDIF
        ENDIF

**************************************
** CALCULATE DDEL/DT FOR SEARLES LAKE **
**************************************

      ELSE IF(I.EQ.4)THEN

         DELL_S = TERM(4)

         IF(VOL_S .LE. 10.D0)THEN
            F_CS=0.D0

         ELSE

******************************************
** CALCULATE ISOTOPIC ENRICHMENT FACTOR **
******************************************

            EPS = FEPS(TSTEMP)

********************************************
** CALCULATE DEL OF THE BACK-CONDENSATION **
********************************************

            SDELC =EPS*(1.D0+(TSDELA/1.D3))+TSDELA

**************************************
** CALCULATE DEL OF THE EVAPORATION **
**************************************

            SDELE = DELE(DELL_S,EPS,TSHUM)

*****************************************************
** SET DEL OF THE OUTFLOW EQUAL TO DEL OF THE LAKE **
*****************************************************

            SDELO = DELL_S

            F_CS=(QI*TSDELI+QC*SDELC+QP*TSDELP-SQO(KNT)*SDELO-QE*
     +           SDELE-DELL_S*DV_DT)/VOL_S

         ENDIF

      END IF
      RETURN
      END

********************************************************************************
********************************************************************************
**          A FUNCTION SUBPROGRAM TO CALCULATE THE AREA OF OWENS LAKE         **
********************************************************************************
********************************************************************************

      DOUBLE PRECISION FUNCTION OAREA (VOL)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      TVOL=VOL/1.0D9

      IF(TVOL .GE. 0.0D0 .AND. TVOL .LT. 0.16D0) THEN
         OAREA = 1.25D0*TVOL
```

```
      ELSE IF (TVOL .GE. 0.16D0 .AND. TVOL .LT. 3.15D0) THEN
         OAREA = 3.01D-2*(TVOL-0.16D0)+0.2D0
      ELSE IF(TVOL .GE. 3.15D0 .AND. TVOL .LT. 30.02D0) THEN
         OAREA = 1.5035D-2*(TVOL-3.15D0)+0.29D0
      ELSE
         OAREA = 0.694D0
      END IF
      OAREA=OAREA*1.0D9
      RETURN
      END

******************************************************************************
******************************************************************************
**         A FUNCTION SUBPROGRAM TO CALCULATE THE AREA OF CHINA LAKE        **
******************************************************************************
******************************************************************************

      DOUBLE PRECISION FUNCTION CAREA (VOL)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      TVOL=VOL/1.0D9
      IF(TVOL .GE. 0.0D0 .AND. TVOL .LT. 0.036D0) THEN
         CAREA = 0.75D0*TVOL
      ELSE IF(TVOL .GE. 0.036D0 .AND. TVOL .LT. 0.696D0) THEN
         CAREA =(.128D0/.66D0)*(TVOL-0.036D0)+0.027D0
      ELSE
         CAREA = 0.155D0
      END IF
      CAREA=CAREA*1.0D9
      RETURN
      END

******************************************************************************
******************************************************************************
**       A FUNCTION SUBPROGRAM TO CALCULATE THE AREA OF SEARLES LAKE        **
******************************************************************************
******************************************************************************

      DOUBLE PRECISION FUNCTION SAREA (VOL)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      TVOL=VOL/1.0D9

      IF(TVOL .GE. 0.0D0 .AND. TVOL .LT. 2.04D0) THEN
         SAREA =(0.245D0/2.04D0)*TVOL
      ELSE IF(TVOL .GE. 2.04D0 .AND. TVOL .LT. 10.75D0) THEN
         SAREA=(0.055D0/8.71D0)*(TVOL-2.04D0)+0.245D0
      ELSE IF(TVOL .GE. 10.75D0 .AND. TVOL .LT. 20.82D0) THEN
         SAREA=(0.05D0/10.07D0)*(TVOL-10.75D0)+0.3D0
      ELSE IF(TVOL .GE. 20.82D0 .AND. TVOL .LT. 33.46D0) THEN
         SAREA=(0.092D0/12.64D0)*(TVOL-20.82D0)+0.35D0
      ELSE IF(TVOL .GE. 33.46D0 .AND. TVOL .LT. 46.6D0) THEN
         SAREA=(0.091D0/13.14D0)*(TVOL-33.46D0)+0.442D0
      ELSE IF(TVOL .GE. 46.6D0 .AND. TVOL .LT. 65.87D0) THEN
         SAREA=(0.182D0/19.27D0)*(TVOL-46.6D0)+0.533D0
      ELSE IF(TVOL .GE. 65.87D0 .AND. TVOL .LT. 85.28D0) THEN
         SAREA=(0.279D0/19.41D0)*(TVOL-65.87D0)+0.715D0
      ELSE
         SAREA=0.994D0
      ENDIF
      SAREA=SAREA*1.0D9
      RETURN
      END
```

```
*******************************************************************************
*******************************************************************************
**    A FUNCTION SUBPROGRAM TO CALCULATE THE VOLUME OF PANAMINT LAKE        **
*******************************************************************************
*******************************************************************************

      DOUBLE PRECISION FUNCTION PVOL(AREA)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      TAREA=AREA/1.0D9

      IF(TAREA .GE. 0.0D0 .AND. TAREA .LT. 0.118D0) THEN
        PVOL =(0.71D0/.118D0)*TAREA
      ELSE IF(TAREA .GE. 0.118D0 .AND. TAREA .LT. 0.175D0) THEN
        PVOL=(2.91D0/0.057D0)*(TAREA-0.118D0)+0.71D0
      ELSE IF(TAREA .GE. 0.175D0 .AND. TAREA .LT. 0.189D0) THEN
        PVOL=(2.D0/0.014D0)*(TAREA-0.1750)+3.62D0
      ELSE IF(TAREA .GE. 0.189D0 .AND. TAREA .LT. 0.242D0) THEN
        PVOL=(6.45D0/0.53D0)*(TAREA-0.189D0)+5.62D0
      ELSE IF(TAREA .GE. 0.242D0 .AND. TAREA .LT. 0.289D0) THEN
        PVOL=(8.22D0/0.047D0)*(TAREA-0.242D0)+12.07D0
      ELSE IF(TAREA .GE. 0.289D0 .AND. TAREA .LT. 0.329D0) THEN
        PVOL=(9.26D0/0.04D0)*(TAREA-0.289D0)+20.29D0
      ELSE IF(TAREA .GE. 0.329D0 .AND. TAREA .LT. 0.369D0) THEN
        PVOL=(10.81D0/0.04D0)*(TAREA-0.329D0)+29.55D0
      ELSE IF(TAREA .GE. 0.369D0 .AND. TAREA .LT. 0.428D0) THEN
        PVOL=(13.93D0/0.059D0)*(TAREA-0.369D0)+40.36D0
      ELSE IF(TAREA .GE. 0.428D0 .AND. TAREA .LT. 0.488D0) THEN
        PVOL=(9.15D0/0.06D0)*(TAREA-0.428D0)+54.29D0
      ELSE IF(TAREA .GE. 0.488D0 .AND. TAREA .LT. 0.524D0) THEN
        PVOL=(7.59D0/0.036D0)*(TAREA-0.488D0)+63.44D0
      ELSE IF(TAREA .GE. 0.524D0 .AND. TAREA .LT. 0.568D0) THEN
        PVOL=(10.92D0/0.044D0)*(TAREA-0.524D0)+71.03D0
      ELSE IF(TAREA .GE. 0.568D0 .AND. TAREA .LT. 0.638D0) THEN
        PVOL=(15.67D0/0.07D0)*(TAREA-0.568D0)+81.95D0
      ELSE IF(TAREA .GE. 0.638D0 .AND. TAREA .LT. 0.727D0) THEN
        PVOL=(10.23D0/0.089D0)*(TAREA-0.638D0)+97.62D0
      ELSE
        PVOL=107.85D0
      ENDIF
      PVOL=PVOL*1.0D9
      RETURN
      END

*******************************************************************************
*******************************************************************************
**       A FUNCTION SUBPROGRAM TO CALCULATE THE VOLUME OF MANLY LAKE        **
*******************************************************************************
*******************************************************************************

      DOUBLE PRECISION FUNCTION DVOL(AREA)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      DAREA=AREA/1.0D9

      IF(TAREA .GE. 0.0D0 .AND. TAREA .LT. 0.05D0) THEN
        DVOL =(0.65D0/.05D0)*TAREA
      ELSE IF(TAREA .GE. 0.05D0 .AND. TAREA .LT. 47.0D0) THEN
        DVOL=(46.35D0/0.533D0)*(TAREA-0.05D0)+0.65D0
      ELSE
        DVOL=47.D0
      ENDIF

      DVOL=DVOL*1.0D9
```

```
      RETURN
      END

********************************************************************************
********************************************************************************
*  THIS IS A FUNCTION SUBPROGRAM TO CALCULATE THE ISOTOPIC ENRICHMENT FACTOR  *
********************************************************************************
********************************************************************************

      DOUBLE PRECISION FUNCTION FEPS(TEMP)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)

**************************
**CONVERT TEMP TO KELVIN**
**************************

      TEMPK=TEMP+273.15D0
      FEPS =(DEXP((1.534D0*(1.0D6/(TEMPK)**2)-3.206D0*(1.0D3/TEMPK)+
     +     2.644D0)/1.D3)-1.D0)*1.D3
      RETURN
      END

********************************************************************************
********************************************************************************
*     THIS IS A FUNCTION SUBPROGRAM TO CALCULATE THE OSMOTIC COEFFICIENT     *
********************************************************************************
********************************************************************************

      DOUBLE PRECISION FUNCTION FPHI(CL_M,SUM)

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

*********************************
**CALCULATE MOLARITIES FOR IONS**
*********************************

 **   CO3_M=CL_M*1.03D0
      CO3_M=0.D0
      SOD_M=CL_M+CO3_M
      SUM=CL_M+SOD_M+CO3_M

**************************************
**CALCULATE OSMOTIC COEFFICIENT, PHI**
**************************************

      XI = (SOD_M+CL_M+4.D0*CO3_M)/2.D0
      XF = 0.392D0*(DSQRT(XI)/(1.D0+1.2D0*DSQRT(XI)))
      BCL = 0.0765D0 + 0.2664D0 * DEXP(-2.D0*DSQRT(XI))
      BCO3 = 0.18975D0+0.846D0*DEXP(-2.D0*DSQRT(XI))
      DSUM1 = 2.D0*SOD_M*CL_M*(BCL+SOD_M*0.00127D0)
      DSUM2 = 2.D0*SOD_M*CO3_M*(BCO3-0.048032D0*SOD_M/DSQRT(2.D0))
      FPHI = 1.4121D0*(1.D0+(1.D0/SUM*(2.D0*XI*XF+DSUM1+DSUM2)))
      RETURN
      END

********************************************************************************
********************************************************************************
*     A SUBROUTINE TO LINEARLY INTERPOLATE BETWEEN POINTS IN DATA SETS        *
*                    CONTAINING OWENS LAKE PARAMETERS                         *
********************************************************************************
********************************************************************************

      SUBROUTINE OINTERP(TIME,NDX,TODELI,TOTEMP,TOEVAP,TOPRECIP,
     +     TODELP,TODELA)

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
```

```
      LOGICAL GUESS,CPARAM,GUESS2

      PARAMETER(NUMA=15000,NUMB=2500)
      COMMON/FIRST/A,B,WTIME(NUMB),OTEMP(NUMB),OEVAP(NUMB),
     + OHUM(500),OPRECIP(NUMB),ODELP(NUMB),ODELA(NUMB),ODELI(NUMB),
     + GUESS,C,TEMPC,EVAPC,PRECIPC,DELAC,DELIC,CPARAM,DELPC,
     + EVAPC_P,EVAPC_D

      COMMON/SECOND/CTEMP(NUMB),CEVAP(NUMB),CHUM(500),CPRECIP(NUMB),
     + CDELP(NUMB),CSDELA(NUMB),GUESS2,TEMPC_C,EVAPC_C,
     + PRECIPC_C,DELAC_C,DELPC_C,STEMP(NUMB),SEVAP(NUMB),SHUM(500),
     + SPRECIP(NUMB),SDELP(NUMB),TEMPC_S,EVAPC_S,
     + PRECIPC_S,DELAC_S,DELPC_S

      TODELI=(ODELI(NDX+1)-ODELI(NDX))/(WTIME(NDX+1)-WTIME(NDX))*
     +      (TIME-WTIME(NDX))+ODELI(NDX)

      TOTEMP=(OTEMP(NDX+1)-OTEMP(NDX))/(WTIME(NDX+1)-WTIME(NDX))*
     +      (TIME-WTIME(NDX))+OTEMP(NDX)

      TOEVAP=(OEVAP(NDX+1)-OEVAP(NDX))/(WTIME(NDX+1)-WTIME(NDX))*
     +      (TIME-WTIME(NDX))+OEVAP(NDX)

      TOPRECIP=(OPRECIP(NDX+1)-OPRECIP(NDX))/(WTIME(NDX+1)-
     +      WTIME(NDX))*(TIME-WTIME(NDX))+OPRECIP(NDX)

      TODELP = (ODELP(NDX+1)-ODELP(NDX))/(WTIME(NDX+1)-WTIME(NDX))*
     +      (TIME-WTIME(NDX))+ODELP(NDX)

      TODELA = (ODELA(NDX+1)-ODELA(NDX))/(WTIME(NDX+1)-WTIME(NDX))*
     +      (TIME-WTIME(NDX))+ODELA(NDX)

      RETURN
      END

*****************************************************************************
*****************************************************************************
*          A SUBROUTINE TO CALCULATE HUMIDITY FOR ALL OF THE LAKES        **
*****************************************************************************
*****************************************************************************


      SUBROUTINE HUMTERP(TIME,NDX,T_HUM,HUM,HUMTIME)

      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      PARAMETER(NUM=500)
      DOUBLE PRECISION HUM(NUM),HUMTIME(NUM)

      T_HUM=(HUM(NDX+1)-HUM(NDX))/(HUMTIME(NDX+1)-HUMTIME(NDX))*
     +      (TIME-HUMTIME(NDX))+HUM(NDX)


      RETURN
      END

*****************************************************************************
*****************************************************************************
*          A SUBROUTINE TO CALCULATE INFLOW FROM HISTORY                  **
*****************************************************************************
*****************************************************************************

      SUBROUTINE QINTERP(TIME,NDX,QI,QHIST,QITIME)

      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
```

```
      DOUBLE PRECISION QHIST(1000),QITIME(1000)

      QI=(QHIST(NDX+1)-QHIST(NDX))/(QITIME(NDX+1)-QITIME(NDX))*
     +      (TIME-QITIME(NDX))+QHIST(NDX)


      RETURN
      END

*******************************************************************************
*******************************************************************************
*      A SUBROUTINE TO LINEARLY INTERPOLATE BETWEEN POINTS IN DATA SETS      *
*                  CONTAINING CHINA LAKE PARAMETERS                          *
*******************************************************************************
*******************************************************************************

      SUBROUTINE CINTERP(TIME,NDX,TCTEMP,TCEVAP,TCPRECIP,
     +      TCDELP,TCDELA)

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      LOGICAL GUESS,CPARAM,GUESS2

      PARAMETER(NUMA=15000,NUMB=2500)
      COMMON/FIRST/A,B,WTIME(NUMB),OTEMP(NUMB),OEVAP(NUMB),
     + OHUM(500),OPRECIP(NUMB),ODELP(NUMB),ODELA(NUMB),ODELI(NUMB),
     + GUESS,C,TEMPC,EVAPC,PRECIPC,DELAC,DELIC,CPARAM,DELPC,
     + EVAPC_P,EVAPC_D

      COMMON/SECOND/CTEMP(NUMB),CEVAP(NUMB),CHUM(500),CPRECIP(NUMB),
     + CDELP(NUMB),CSDELA(NUMB),GUESS2,TEMPC_C,EVAPC_C,
     + PRECIPC_C,DELAC_C,DELPC_C,STEMP(NUMB),SEVAP(NUMB),SHUM(500),
     + SPRECIP(NUMB),SDELP(NUMB),TEMPC_S,EVAPC_S,
     + PRECIPC_S,DELAC_S,DELPC_S


      TCTEMP=(CTEMP(NDX+1)-CTEMP(NDX))/(WTIME(NDX+1)-WTIME(NDX))*
     +      (TIME-WTIME(NDX))+CTEMP(NDX)

      TCEVAP=(CEVAP(NDX+1)-CEVAP(NDX))/(WTIME(NDX+1)-WTIME(NDX))*
     +      (TIME-WTIME(NDX))+CEVAP(NDX)

      TCHUM=(CHUM(NDX+1)-CHUM(NDX))/(WTIME(NDX+1)-WTIME(NDX))*
     +      (TIME-WTIME(NDX))+CHUM(NDX)

      TCPRECIP=(CPRECIP(NDX+1)-CPRECIP(NDX))/(WTIME(NDX+1)-
     +      WTIME(NDX))*(TIME-WTIME(NDX))+CPRECIP(NDX)

      TCDELP=(CDELP(NDX+1)-CDELP(NDX))/(WTIME(NDX+1)-WTIME(NDX))*
     +      (TIME-WTIME(NDX))+CDELP(NDX)

      TCDELA=(CSDELA(NDX+1)-CSDELA(NDX))/(WTIME(NDX+1)-WTIME(NDX))*
     +      (TIME-WTIME(NDX))+CSDELA(NDX)

      RETURN
      END

*******************************************************************************
*******************************************************************************
*      A SUBROUTINE TO LINEARLY INTERPOLATE BETWEEN POINTS IN DATA SETS      *
*               FROM THE OWENS LAKE CALCULATIONS                             *
*******************************************************************************
*******************************************************************************
```

```
      SUBROUTINE C2INTERP(TIME,NDX,TCDELI,CQI,TAREA)

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      LOGICAL GUESS,CPARAM,GUESS2

      PARAMETER(NUMA=15000,NUMB=2500)

      COMMON/BOTH/CL(NUMA),TSOD(10),TCL(10),OTIME(NUMA),TCO3(10),
     + OQO(NUMA),QO(10),CONC_CL(NUMA),DOLTEMP,DELDOL,TODELI,TOTEMP,
     + ODEL_OUT(NUMA),OCL_OUT(NUMA),PEVAP(NUMB),DEVAP(NUMB),
     + SUM_PCL_DEP(NUMA),SUM_DCL_DEP(NUMA),AREA_P,AREA_D,
     + AREA(NUMA),SOAREA,ALLAREA(NUMA),CONC_CL_P(NUMA),CONC_CL_D(NUMA),
     + CL_P(NUMA),CL_D(NUMA),INCHOICE

      COMMON/FIRST/A,B,WTIME(NUMB),OTEMP(NUMB),OEVAP(NUMB),
     + OHUM(500),OPRECIP(NUMB),ODELP(NUMB),ODELA(NUMB),ODELI(NUMB),
     + GUESS,C,TEMPC,EVAPC,PRECIPC,DELAC,DELIC,CPARAM,DELPC,
     + EVAPC_P,EVAPC_D

      COMMON/SECOND/CTEMP(NUMB),CEVAP(NUMB),CHUM(500),CPRECIP(NUMB),
     + CDELP(NUMB),CSDELA(NUMB),GUESS2,TEMPC_C,EVAPC_C,
     + PRECIPC_C,DELAC_C,DELPC_C,STEMP(NUMB),SEVAP(NUMB),SHUM(500),
     + SPRECIP(NUMB),SDELP(NUMB),TEMPC_S,EVAPC_S,
     + PRECIPC_S,DELAC_S,DELPC_S


      TCDELI = (ODEL_OUT(NDX+1)-ODEL_OUT(NDX))/(OTIME(NDX+1)-
     +      OTIME(NDX))*(TIME-OTIME(NDX))+ODEL_OUT(NDX)

      CQI = (OQO(NDX+1)-OQO(NDX))/(OTIME(NDX+1)-OTIME(NDX))*
     +      (TIME-OTIME(NDX))+OQO(NDX)

      TAREA = (AREA(NDX+1)-AREA(NDX))/(OTIME(NDX+1)-OTIME(NDX))*
     +      (TIME-OTIME(NDX))+AREA(NDX)

      RETURN
      END

******************************************************************************
******************************************************************************
*     A SUBROUTINE TO LINEARLY INTERPOLATE BETWEEN POINTS IN AN ARRAY        *
*               FROM THE OWENS LAKE SALT OUTFLOW HISTORY                     *
******************************************************************************
******************************************************************************

      SUBROUTINE C3INTERP(TIME,NDX,SLTNUM)

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      LOGICAL GUESS,CPARAM,GUESS2

      PARAMETER(NUMA=15000,NUMB=2500)

      COMMON/BOTH/CL(NUMA),TSOD(10),TCL(10),OTIME(NUMA),TCO3(10),
     + OQO(NUMA),QO(10),CONC_CL(NUMA),DOLTEMP,DELDOL,TODELI,TOTEMP,
     + ODEL_OUT(NUMA),OCL_OUT(NUMA),PEVAP(NUMB),DEVAP(NUMB),
     + SUM_PCL_DEP(NUMA),SUM_DCL_DEP(NUMA),AREA_P,AREA_D,
     + AREA(NUMA),SOAREA,ALLAREA(NUMA),CONC_CL_P(NUMA),CONC_CL_D(NUMA),
     + CL_P(NUMA),CL_D(NUMA),INCHOICE

      COMMON/FIRST/A,B,WTIME(NUMB),OTEMP(NUMB),OEVAP(NUMB),
     + OHUM(500),OPRECIP(NUMB),ODELP(NUMB),ODELA(NUMB),ODELI(NUMB),
     + GUESS,C,TEMPC,EVAPC,PRECIPC,DELAC,DELIC,CPARAM,DELPC,
```

```
     +  EVAPC_P,EVAPC_D

        COMMON/SECOND/CTEMP(NUMB),CEVAP(NUMB),CHUM(500),CPRECIP(NUMB),
     +  CDELP(NUMB),CSDELA(NUMB),GUESS2,TEMPC_C,EVAPC_C,
     +  PRECIPC_C,DELAC_C,DELPC_C,STEMP(NUMB),SEVAP(NUMB),SHUM(500),
     +  SPRECIP(NUMB),SDELP(NUMB),TEMPC_S,EVAPC_S,
     +  PRECIPC_S,DELAC_S,DELPC_S


        SLTNUM = (OCL_OUT(NDX+1)-OCL_OUT(NDX))/(OTIME(NDX+1)-
     +       OTIME(NDX))*(TIME-OTIME(NDX))+OCL_OUT(NDX)

        RETURN
        END


*******************************************************************************
*******************************************************************************
*      A SUBROUTINE TO LINEARLY INTERPOLATE BETWEEN POINTS IN DATA SETS      *
*                    CONTAINING SEARLES LAKE PARAMETERS                       *
*******************************************************************************
*******************************************************************************

        SUBROUTINE SINTERP(TIME,NDX,TSTEMP,TSEVAP,TSPRECIP,
     +       TSDELP,TSDELA)

        IMPLICIT DOUBLE PRECISION (A-H,O-Z)

        LOGICAL GUESS,CPARAM,GUESS2

        PARAMETER(NUMA=15000,NUMB=2500)
        COMMON/FIRST/A,B,WTIME(NUMB),OTEMP(NUMB),OEVAP(NUMB),
     +  OHUM(500),OPRECIP(NUMB),ODELP(NUMB),ODELA(NUMB),ODELI(NUMB),
     +  GUESS,C,TEMPC,EVAPC,PRECIPC,DELAC,DELIC,CPARAM,DELPC,
     +  EVAPC_P,EVAPC_D

        COMMON/SECOND/CTEMP(NUMB),CEVAP(NUMB),CHUM(500),CPRECIP(NUMB),
     +  CDELP(NUMB),CSDELA(NUMB),GUESS2,TEMPC_C,EVAPC_C,
     +  PRECIPC_C,DELAC_C,DELPC_C,STEMP(NUMB),SEVAP(NUMB),SHUM(500),
     +  SPRECIP(NUMB),SDELP(NUMB),TEMPC_S,EVAPC_S,
     +  PRECIPC_S,DELAC_S,DELPC_S


        TSTEMP=(STEMP(NDX+1)-STEMP(NDX))/(WTIME(NDX+1)-WTIME(NDX))*
     +       (TIME-WTIME(NDX))+STEMP(NDX)

        TSEVAP=(SEVAP(NDX+1)-SEVAP(NDX))/(WTIME(NDX+1)-WTIME(NDX))*
     +       (TIME-WTIME(NDX))+SEVAP(NDX)

        TSHUM=(SHUM(NDX+1)-SHUM(NDX))/(WTIME(NDX+1)-WTIME(NDX))*
     +       (TIME-WTIME(NDX))+SHUM(NDX)

        TSPRECIP=(SPRECIP(NDX+1)-SPRECIP(NDX))/(WTIME(NDX+1)-
     +       WTIME(NDX))*(TIME-WTIME(NDX))+SPRECIP(NDX)

        TSDELP=(SDELP(NDX+1)-SDELP(NDX))/(WTIME(NDX+1)-WTIME(NDX))*
     +       (TIME-WTIME(NDX))+SDELP(NDX)

        TSDELA=(CSDELA(NDX+1)-CSDELA(NDX))/(WTIME(NDX+1)-WTIME(NDX))*
     +       (TIME-WTIME(NDX))+CSDELA(NDX)

        RETURN
        END
```

```
******************************************************************************
******************************************************************************
*      A SUBROUTINE TO LINEARLY INTERPOLATE BETWEEN POINTS IN AN ARRAY       *
*                FROM THE OWENS LAKE SALT OUTFLOW HISTORY                    *
******************************************************************************
******************************************************************************

      SUBROUTINE PDINTERP(TIME,NDX,TPEVAP,TDEVAP)

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      LOGICAL GUESS,CPARAM,GUESS2

      PARAMETER(NUMA=15000,NUMB=2500)

      COMMON/BOTH/CL(NUMA),TSOD(10),TCL(10),OTIME(NUMA),TCO3(10),
     + OQO(NUMA),QO(10),CONC_CL(NUMA),DOLTEMP,DELDOL,TODELI,TOTEMP,
     + ODEL_OUT(NUMA),OCL_OUT(NUMA),PEVAP(NUMB),DEVAP(NUMB),
     + SUM_PCL_DEP(NUMA),SUM_DCL_DEP(NUMA),AREA_P,AREA_D,
     + AREA(NUMA),SOAREA,ALLAREA(NUMA),CONC_CL_P(NUMA),CONC_CL_D(NUMA),
     + CL_P(NUMA),CL_D(NUMA),INCHOICE

      COMMON/FIRST/A,B,WTIME(NUMB),OTEMP(NUMB),OEVAP(NUMB),
     + OHUM(500),OPRECIP(NUMB),ODELP(NUMB),ODELA(NUMB),ODELI(NUMB),
     + GUESS,C,TEMPC,EVAPC,PRECIPC,DELAC,DELIC,CPARAM,DELPC,
     + EVAPC_P,EVAPC_D

      COMMON/SECOND/CTEMP(NUMB),CEVAP(NUMB),CHUM(500),CPRECIP(NUMB),
     + CDELP(NUMB),CSDELA(NUMB),GUESS2,TEMPC_C,EVAPC_C,
     + PRECIPC_C,DELAC_C,DELPC_C,STEMP(NUMB),SEVAP(NUMB),SHUM(500),
     + SPRECIP(NUMB),SDELP(NUMB),TEMPC_S,EVAPC_S,
     + PRECIPC_S,DELAC_S,DELPC_S


      TPEVAP = (PEVAP(NDX+1)-PEVAP(NDX))/(WTIME(NDX+1)-
     +     WTIME(NDX))*(TIME-WTIME(NDX))+PEVAP(NDX)

      TDEVAP = (DEVAP(NDX+1)-DEVAP(NDX))/(WTIME(NDX+1)-
     +     WTIME(NDX))*(TIME-WTIME(NDX))+DEVAP(NDX)

      RETURN
      END


******************************************************************************
******************************************************************************
** A FUNCTION SUBPROGRAM TO CALCULATE THE RELATIVE ISOTOPIC ENRICHMENT OF   **
** THE EVAPORATING WATER.                                                   **
******************************************************************************
******************************************************************************

      DOUBLE PRECISION FUNCTION DELE(DELL,EPS,HUM)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      PARAMETER(C=6.8D-3)

      DELE=(((1.D0+1.D-3*DELL)*(1.D0-C))/((1.D0+1.D-3*EPS)*(1.D0-C*
     +     HUM))-1.D0)*1.0D3

      RETURN
      END

******************************************************************************
******************************************************************************
**         A FUNCTION SUBPROGRAM TO CALCULATE THE INFLOW, QI(T)             **
******************************************************************************
```

```
*******************************************************************************

      DOUBLE PRECISION FUNCTION FQI(X)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)

      LOGICAL GUESS,CPARAM,GUESS2

      PARAMETER(NUMA=15000,NUMB=2500)

      COMMON/FIRST/A,B,WTIME(NUMB),OTEMP(NUMB),OEVAP(NUMB),
     + OHUM(500),OPRECIP(NUMB),ODELP(NUMB),ODELA(NUMB),ODELI(NUMB),
     + GUESS,C,TEMPC,EVAPC,PRECIPC,DELAC,DELIC,CPARAM,DELPC,
     + EVAPC_P,EVAPC_D

      COMMON/SECOND/CTEMP(NUMB),CEVAP(NUMB),CHUM(500),CPRECIP(NUMB),
     + CDELP(NUMB),CSDELA(NUMB),GUESS2,TEMPC_C,EVAPC_C,
     + PRECIPC_C,DELAC_C,DELPC_C,STEMP(NUMB),SEVAP(NUMB),SHUM(500),
     + SPRECIP(NUMB),SDELP(NUMB),TEMPC_S,EVAPC_S,
     + PRECIPC_S,DELAC_S,DELPC_S


      IF(INCHOICE .EQ. 1)THEN
         FQI=A*X+B
      ELSE IF(INCHOICE .EQ. 2)THEN
         FQI=B*DEXP(A*X)
      ELSE IF(INCHOICE .EQ. 3)THEN
         IF(X .LT. 1.D0)X=1.D0
         FQI=B+A*DLOG10(X)
      ELSE IF(INCHOICE .EQ. 4)THEN
         FQI=B+A*X**C
      ELSE IF(INCHOICE .EQ. 5)THEN
         FQI=B+A*DSIN(C*X)
      ELSE IF(INCHOICE .EQ. 6)THEN
         IF(X .LT. 1.D0)THEN
            FQI=B
         ELSE
            FQI=B+A*B
         ENDIF
      ELSE IF(INCHOICE .EQ. 7)THEN
         FQI=0.D0
      ENDIF
      IF(FQI .LT. 0.D0)FQI=0.D0
      RETURN
      END


*******************************************************************************
*******************************************************************************
*        A SUBROUTINE TO FIND THE TIME INDEX WITH A BINARY SEARCH             *
*******************************************************************************
*******************************************************************************

      SUBROUTINE FINDT(NPTS,TM,INDEX,FOUND,TIME)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      LOGICAL FOUND
      PARAMETER(NUMA=15000,NUMB=2500)
      DIMENSION TIME(NUMB)

      TFST=1
      TLST=NPTS
      FOUND = .FALSE.
      DO 200 I = 1,100
         IF(TLST .LT. TFST) THEN
            INDEX=TLST
            GOTO 210
         ENDIF
```

```
            IF( TFST .LE. TLST .AND. .NOT. FOUND)THEN
               MIDDLE = (TFST+TLST)/2
               TEST = ABS(TM-TIME(MIDDLE))
               IF( TEST .LT. 1.0D-1) THEN
                  FOUND = .TRUE.
                  INDEX=MIDDLE
                  GOTO 210
               ELSE IF(TM .LT. TIME(MIDDLE))THEN
                  TLST = MIDDLE-1
               ELSE
                  TFST = MIDDLE+1
               END IF
            END IF
200      CONTINUE

210      RETURN
         END

*******************************************************************************
*******************************************************************************
*     A SUBROUTINE TO FIND THE HUMIDITY TIME INDEX WITH A BINARY SEARCH       *
*******************************************************************************
*******************************************************************************

         SUBROUTINE FINDHT(NPTS,TM,INDEX,TIME)
         IMPLICIT DOUBLE PRECISION(A-H,O-Z)
         LOGICAL FOUND
         DIMENSION TIME(500)
         TFIRST=1
         TLAST=NPTS
         FOUND = .FALSE.
         DO 200 I = 1,100
            IF(TLAST .LT. TFIRST) THEN
               INDEX=TLAST
               GOTO 210
            ENDIF
            IF( TFIRST .LE. TLAST .AND. .NOT. FOUND)THEN
               MIDDLE = (TFIRST+TLAST)/2
               TEST = ABS(TM-TIME(MIDDLE))
               IF( TEST .LT. 1.0D-1) THEN
                  FOUND = .TRUE.
                  INDEX=MIDDLE
                  GOTO 210
               ELSE IF(TM .LT. TIME(MIDDLE))THEN
                  TLAST = MIDDLE-1
               ELSE
                  TFIRST = MIDDLE+1
               END IF
            END IF
200      CONTINUE

210      RETURN
         END

*******************************************************************************
*******************************************************************************
*     A SUBROUTINE TO FIND THE INFLOW TIME INDEX WITH A BINARY SEARCH         *
*******************************************************************************
*******************************************************************************

         SUBROUTINE FINDQT(NPTS,TM,INDEX,TIME)
         IMPLICIT DOUBLE PRECISION(A-H,O-Z)
         LOGICAL FOUND
         DIMENSION TIME(1000)
         TFIRST=1
         TLAST=NPTS
```

```
      FOUND = .FALSE.
      DO 200 I = 1,100
         IF(TLAST .LT. TFIRST) THEN
            INDEX=TLAST
            GOTO 210
         ENDIF
         IF( TFIRST .LE. TLAST .AND. .NOT. FOUND)THEN
            MIDDLE = (TFIRST+TLAST)/2
            TEST = ABS(TM-TIME(MIDDLE))
            IF( TEST .LT. 1.0D-1) THEN
               FOUND = .TRUE.
               INDEX=MIDDLE
               GOTO 210
            ELSE IF(TM .LT. TIME(MIDDLE))THEN
               TLAST = MIDDLE-1
            ELSE
               TFIRST = MIDDLE+1
            END IF
         END IF
200   CONTINUE

210   RETURN
      END


*******************************************************************************
*******************************************************************************
*          A SUBROUTINE TO FIND THE TIME INDEX WITH A BINARY SEARCH           *
*******************************************************************************
*******************************************************************************

      SUBROUTINE FINDT2(NPTS,TM,INDEX2,FOUND2,TIME)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      LOGICAL FOUND2
      PARAMETER(NUMA=15000,NUMB=2500)
      DIMENSION TIME(NUMA)
      TFIRST=1
      TLAST=NPTS
      FOUND2 = .FALSE.
      DO 200 I = 1,100
         IF(TLAST .LT. TFIRST) THEN
            INDEX2=TLAST
            GOTO 210
         ENDIF
         IF( TFIRST .LE. TLAST .AND. .NOT. FOUND2) THEN
            MIDDLE = (TFIRST+TLAST)/2
            TEST = ABS(TM-TIME(MIDDLE))
            IF( TEST .LT. 1.0D-1) THEN
               FOUND2 = .TRUE.
               INDEX2=MIDDLE
               GOTO 210
            ELSE IF(TM .GT. TIME(MIDDLE))THEN
               TLAST = MIDDLE-1
            ELSE
               TFIRST = MIDDLE+1
            END IF
         END IF
200   CONTINUE

210   RETURN
      END


      SUBROUTINE PLOTEM(XMIN,XMAX,XPLT,YDEL,YAREA,YQI,YTEMP,KOUNT)
```

```
******************************************************************************
******************************************************************************
**  A SUBROUTINE TO PLOT THE RESULTS OF THE OWENS LAKE SIMULATION.  THIS    **
**  SUBROUTINE PLOTS INFLOW, SURFACE AREA, DEL O18, AND TEMPERATURE.        **
**  GRAPHICS CALLS ARE MADE TO DISPLAY, THE GRAPHICS SOFTWARE ON THE VAX.   **
******************************************************************************
******************************************************************************

*****************************************************************************
*       SET UP THE GRAPHICS WINDOW FOR PLOTTING                             *
*****************************************************************************

        PARAMETER(NUMA=15000,NUMB=2500)
        REAL XMIN,XMAX,YMIN,YMAX,XPLT(NUMA),YAREA(NUMA),YQI(NUMA),
     +       YDEL(NUMA),YTEMP(NUMA)

        CALL UIS

        CALL PAGE(33.,22.)
        CALL COMPLX
        CALL HEIGHT(.35)

C$$$        FIND THE POSITION TO PUT THE GRAPH
        DO 300 I=1,4
           IF(I .EQ. 1)THEN
              CALL PHYSOR(4.,17.)
           ELSE IF(I .EQ. 2)THEN
              CALL PHYSOR(4.,12.)
           ELSE IF(I .EQ. 3)THEN
              CALL PHYSOR(4.,7.)
           ELSE IF(I .EQ. 4)THEN
              CALL PHYSOR(4.,2.)
           ENDIF

           CALL AREA2D (25.,3.)
           CALL GAPWID(.001)

C$$$   PUT THE HEADING ON THE PLOT

           IF(I .EQ. 1)THEN
              CALL YNAME ('DEL O-18$',100)
              CALL XNAME ('TIME$',100)
           ELSEIF(I .EQ. 2)THEN
              CALL YNAME ('AREA$',100)
              CALL XNAME ('TIME$',100)
           ELSEIF(I .EQ. 3)THEN
              CALL YNAME ('INFLOW $',100)
              CALL XNAME ('TIME$',100)
           ELSEIF(I .EQ. 4)THEN
              CALL YNAME ('TEMP $',100)
              CALL XNAME ('TIME$',100)
           ENDIF

*****************************
** CALCULATE YMAX AND YMIN **
*****************************

           IF(I .EQ. 1)THEN
              YMIN=YDEL(1)
              YMAX=YDEL(1)
              DO 1500 IMM = 2,KOUNT
                 IF(YDEL(IMM) .LT. YMIN)YMIN=YDEL(IMM)
                 IF(YDEL(IMM) .GT. YMAX)YMAX=YDEL(IMM)
1500          CONTINUE
              YMIN=NINT(YMIN)-0.5
              YMAX=NINT(YMAX)+0.5
```

```
         ELSE IF(I .EQ. 2)THEN
            YMIN=YAREA(1)
            YMAX=YAREA(1)
            DO 1600 IMM = 2,KOUNT
               IF(YAREA(IMM) .LT. YMIN)YMIN=YAREA(IMM)
               IF(YAREA(IMM) .GT. YMAX)YMAX=YAREA(IMM)
1600        CONTINUE
            YMIN=YMIN-0.1D0*YMIN
            YMAX=YMAX+0.1D0*YMAX
         ELSE IF(I .EQ. 3)THEN
            YMIN=YQI(1)
            YMAX=YQI(1)
            DO 1700 IMM = 2,KOUNT
               IF(YQI(IMM) .LT. YMIN)YMIN=YQI(IMM)
               IF(YQI(IMM) .GT. YMAX)YMAX=YQI(IMM)
1700        CONTINUE
            YMIN=YMIN-0.1D0*YMIN
            YMAX=YMAX+0.1D0*YMIN
         ELSE IF(I .EQ. 4)THEN
            YMIN=YTEMP(1)
            YMAX=YTEMP(1)
            DO 1800 IMM = 2,KOUNT
               IF(YTEMP(IMM) .LT. YMIN)YMIN=YTEMP(IMM)
               IF(YTEMP(IMM) .GT. YMAX)YMAX=YTEMP(IMM)
1800        CONTINUE
            YMIN=YMIN-0.1D0*YMIN
            YMAX=YMAX+0.1D0*YMIN
         ENDIF

         CALL GRAF (XMIN,XMAX,XMAX,YMIN,YMAX,YMAX)

         CALL FRAME
         CALL THKCRV(.01)
         CALL MARKER(3)
         CALL SCLPIC(.3)

         IF(I .EQ. 1)THEN
            CALL CURVE(XPLT,YDEL,KOUNT,-1)
         ELSE IF(I .EQ. 2)THEN
            CALL CURVE(XPLT,YAREA,KOUNT,-1)
         ELSE IF(I .EQ. 3)THEN
            CALL CURVE(XPLT,YQI,KOUNT,-1)
         ELSE IF(I .EQ. 4)THEN
            CALL CURVE(XPLT,YTEMP,KOUNT,-1)
         ENDIF

         CALL ENDGR(IPLOT)

 300  CONTINUE

      RETURN
      END


      SUBROUTINE PLOTEMC(XMIN,XMAX,XPLT,YDEL_C,YAREA_C,YQI_C,KOUNT)
*********************************************************************
*     SET UP THE GRAPHICS WINDOW FOR PLOTTING                       *
*********************************************************************

      PARAMETER(NUMA=15000,NUMB=2500)
      REAL XMIN,XMAX,YMIN,YMAX,XPLT(NUMA),YAREA_C(NUMA),YQI_C(NUMA),
     +     YDEL_C(NUMA)

      CALL UIS
```

```
      CALL PAGE(33.,22.)
      CALL COMPLX
      CALL HEIGHT(.35)

C$$$        FIND THE POSITION TO PUT THE GRAPH
      DO 300 I=1,3
         IF(I .EQ. 1)THEN
            CALL PHYSOR(4.,12.)
         ELSE IF(I .EQ. 2)THEN
            CALL PHYSOR(4.,7.)
         ELSE IF(I .EQ. 3)THEN
            CALL PHYSOR(4.,2.)
         ENDIF

         IF(I .EQ. 1)THEN
            CALL AREA2D (25.,8.)
         ELSE
            CALL AREA2D (25.,3.)
         ENDIF
         CALL GAPWID(.001)

C$$$   PUT THE HEADING ON THE PLOT

         IF(I .EQ. 1)THEN
            CALL YNAME ('DEL O-18$',100)
            CALL XNAME ('TIME$',100)
         ELSEIF(I .EQ. 2)THEN
            CALL YNAME ('AREA$',100)
            CALL XNAME ('TIME$',100)
         ELSEIF(I .EQ. 3)THEN
            CALL YNAME ('INFLOW $',100)
            CALL XNAME ('TIME$',100)
         ENDIF

******************************
** CALCULATE YMAX AND YMIN **
******************************

         IF(I .EQ. 1)THEN
            YMIN=YDEL_C(1)
            YMAX=YDEL_C(1)
            DO 1500 IMM = 2,KOUNT
               IF(YDEL_C(IMM) .LT. YMIN)YMIN=YDEL_C(IMM)
               IF(YDEL_C(IMM) .GT. YMAX)YMAX=YDEL_C(IMM)
1500        CONTINUE
            YMIN=YMIN-0.5
            YMAX=YMAX+0.5
         ELSE IF(I .EQ. 2)THEN
            YMIN=YAREA_C(1)
            YMAX=YAREA_C(1)
            DO 1600 IMM = 2,KOUNT
               IF(YAREA_C(IMM) .LT. YMIN)YMIN=YAREA_C(IMM)
               IF(YAREA_C(IMM) .GT. YMAX)YMAX=YAREA_C(IMM)
1600        CONTINUE
            YMIN=YMIN-0.1D0*YMIN
            YMAX=YMAX+0.1D0*YMAX
         ELSE IF(I .EQ. 3)THEN
            YMIN=YQI_C(1)
            YMAX=YQI_C(1)
            DO 1700 IMM = 2,KOUNT
               IF(YQI_C(IMM) .LT. YMIN)YMIN=YQI_C(IMM)
               IF(YQI_C(IMM) .GT. YMAX)YMAX=YQI_C(IMM)
1700        CONTINUE
            YMIN=YMIN-0.1D0*YMIN
            YMAX=YMAX+0.1D0*YMIN
         ENDIF
```

```
         CALL GRAF (XMIN,XMAX,XMAX,YMIN,YMAX,YMAX)

         CALL FRAME
         CALL THKCRV(.02)
         CALL MARKER(3)
         CALL SCLPIC(.7)

         IF(I .EQ. 1)THEN
            CALL CURVE(XPLT,YDEL_C,KOUNT,-1)
         ELSE IF(I .EQ. 2)THEN
            CALL CURVE(XPLT,YAREA_C,KOUNT,-1)
         ELSE IF(I .EQ. 3)THEN
            CALL CURVE(XPLT,YQI_C,KOUNT,-1)
         ENDIF

         CALL ENDGR(IPLOT)

 300  CONTINUE

      RETURN
      END


      SUBROUTINE PLOTEMS(XMIN,XMAX,XPLT,YDEL_S,YAREA_S,YQI_S,KOUNT)
****************************************************************************
*     SET UP THE GRAPHICS WINDOW FOR PLOTTING                              *
****************************************************************************

      PARAMETER(NUMA=15000,NUMB=2500)
      REAL XMIN,XMAX,YMIN,YMAX,XPLT(NUMA),YAREA_S(NUMA),YQI_S(NUMA),
     +     YDEL_S(NUMA),STIME(NUMA),SEARISO(NUMA)

      WRITE(6,*)
      WRITE(6,*)'READING SEARLES ISOTOPE DATA'
      WRITE(6,*)

      OPEN(UNIT=60,FILE='REALISO.DAT',STATUS='OLD')

      DO 100 I=1,500
         READ(60,*,END=101)STIME(I),SEARISO(I)
100   CONTINUE

101   NSPTS=I-1

      CLOSE(UNIT=60)

      WRITE(6,*)'READ',NSPTS,' POINTS FROM SEARLES DATA'

      DO 150 I=1,NSPTS
         STIME(I)=STIME(I)*1.0E6
150   CONTINUE

      CALL UIS

      CALL PAGE(33.,22.)
      CALL COMPLX
      CALL HEIGHT(.35)

C$$$       FIND THE POSITION TO PUT THE GRAPH
      DO 300 I=1,3
         IF(I .EQ. 1)THEN
            CALL PHYSOR(4.,12.)
         ELSE IF(I .EQ. 2)THEN
            CALL PHYSOR(4.,7.)
         ELSE IF(I .EQ. 3)THEN
```

```
            CALL PHYSOR(4.,2.)
         ENDIF

         IF(I .EQ. 1)THEN
            CALL AREA2D (25.,8.)
         ELSE
            CALL AREA2D (25.,3.)
         ENDIF
         CALL GAPWID(.001)

C$$$  PUT THE HEADING ON THE PLOT

         IF(I .EQ. 1)THEN
            CALL YNAME ('DEL O-18$',100)
            CALL XNAME ('TIME$',100)
         ELSEIF(I .EQ. 2)THEN
            CALL YNAME ('AREA$',100)
            CALL XNAME ('TIME$',100)
         ELSEIF(I .EQ. 3)THEN
            CALL YNAME ('INFLOW $',100)
            CALL XNAME ('TIME$',100)
         ENDIF

*****************************
** CALCULATE YMAX AND YMIN **
*****************************

         IF(I .EQ. 1)THEN
            YMIN=YDEL_S(1)
            YMAX=YDEL_S(1)
            DO 1500 IMM = 2,KOUNT
               IF(YDEL_S(IMM) .LT. YMIN)YMIN=YDEL_S(IMM)
               IF(YDEL_S(IMM) .GT. YMAX)YMAX=YDEL_S(IMM)
1500        CONTINUE
            YMIN=YMIN-0.5
            YMAX=YMAX+0.5
         ELSE IF(I .EQ. 2)THEN
            YMIN=YAREA_S(1)
            YMAX=YAREA_S(1)
            DO 1600 IMM = 2,KOUNT
               IF(YAREA_S(IMM) .LT. YMIN)YMIN=YAREA_S(IMM)
               IF(YAREA_S(IMM) .GT. YMAX)YMAX=YAREA_S(IMM)
1600        CONTINUE
            YMIN=YMIN-0.1D0*YMIN
            YMAX=YMAX+0.1D0*YMAX
         ELSE IF(I .EQ. 3)THEN
            YMIN=YQI_S(1)
            YMAX=YQI_S(1)
            DO 1700 IMM = 2,KOUNT
               IF(YQI_S(IMM) .LT. YMIN)YMIN=YQI_S(IMM)
               IF(YQI_S(IMM) .GT. YMAX)YMAX=YQI_S(IMM)
1700        CONTINUE
            YMIN=YMIN-0.1D0*YMIN
            YMAX=YMAX+0.1D0*YMIN
         ENDIF

         CALL GRAF (XMIN,XMAX,XMAX,YMIN,YMAX,YMAX)

         CALL FRAME
         CALL THKCRV(.01)
         CALL MARKER(3)
         CALL SCLPIC(.3)

         IF(I .EQ. 1)THEN
            CALL CURVE(XPLT,YDEL_S,KOUNT,-1)
            CALL CURVE(STIME,SEARISO,NSPTS,0)
```

```
         ELSE IF(I .EQ. 2)THEN
            CALL CURVE(XPLT,YAREA_S,KOUNT,-1)
         ELSE IF(I .EQ. 3)THEN
            CALL CURVE(XPLT,YQI_S,KOUNT,-1)
         ENDIF

         CALL ENDGR(IPLOT)

 300  CONTINUE

      RETURN
      END


      SUBROUTINE PLOTEM_SUM(XMIN,XMAX,XPLT,ALLAREA,KOUNT)
******************************************************************************
*     SET UP THE GRAPHICS WINDOW FOR PLOTTING                                *
******************************************************************************

      PARAMETER(NUMA=15000,NUMB=2500)

      REAL XMIN,XMAX,YMIN,YMAX,XPLT(NUMA),RAREA(NUMA),XC(2),OC(2),
     +     CC(2),SC(2),PC(2),DC(2)
      DOUBLE PRECISION ALLAREA(NUMA)


      DATA (OC(I),I=1,2)/.694E3,.694E3/
     +     (CC(I),I=1,2)/.849E3,.849E3/(SC(I),I=1,2)/1.688E3,1.688E3/
     +     (PC(I),I=1,2)/2.415E3,2.415E3/(DC(I),I=1,2)/
     +     2.998E3,2.998E3/

      OPEN(UNIT=81,FILE='ALLAREA.OUT',STATUS='UNKNOWN',CARRIAGE
     +     CONTROL='LIST')

      XC(1)=XMIN
      XC(2)=XMAX

      WRITE(6,*)
      WRITE(6,*)'***************************'
      WRITE(6,*)'PLOTTING TOTAL SURFACE AREA'
      WRITE(6,*)'***************************'
      WRITE(6,*)

      DO 100 I=1,KOUNT
         WRITE(81,*)XPLT(I),ALLAREA(I)
         RAREA(I)=REAL(ALLAREA(I)*1.D-6)
100   CONTINUE

      CALL UIS

      CALL PAGE(33.,22.)
      CALL COMPLX
      CALL HEIGHT(.35)

C$$$      FIND THE POSITION TO PUT THE GRAPH

      CALL PHYSOR(4.,4.)

      CALL AREA2D (25.,15.)
      CALL GAPWID(.001)

C$$$  PUT THE HEADING ON THE PLOT

      CALL YNAME ('SURFACE AREA (KM^2)$',100)
      CALL XNAME ('TIME$',100)
```

```
*****************************
** CALCULATE YMAX AND YMIN **
*****************************
        YMIN=0.0
        YMAX=3.0E3

        CALL GRAF (XMIN,XMAX,XMAX,YMIN,YMAX,YMAX)

        CALL FRAME
        CALL THKCRV(.01)
        CALL MARKER(3)
        CALL SCLPIC(.1)

        CALL CURVE(XPLT,RAREA,KOUNT,-1)
        CALL CURVE(XC,OC,2,1)
        CALL CURVE(XC,CC,2,1)
        CALL CURVE(XC,SC,2,1)
        CALL CURVE(XC,PC,2,1)
        CALL CURVE(XC,CC,2,1)
        CALL CURVE(XC,DC,2,1)

        CALL ENDGR(IPLOT)

        RETURN
        END

**********************************************************
**********************************************************
**   A FUNCTION SUBPROGRAM TO  CALCULATE DEL DOLOMITE **
**********************************************************
**********************************************************

        DOUBLE PRECISION FUNCTION FDDOL(TEMP,DELH2O)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)

************************************
** CONVERT TEMP TO DEGREES KELVIN **
************************************

        TEMPK = TEMP+273.15D0

**********************************************
** CALCULATE EPSILON FOR DOLOMITE AND WATER **
**********************************************

        EPSDOL=(DEXP((3.2D0*(1.D6/TEMPK**2)-4.3D0)/1.D3)-1.D0)*1.D3

        FDDOL=EPSDOL+DELH2O*(EPSDOL/1.0D3+1.D0)

        RETURN
        END


**********************************************************
**********************************************************
**   A FUNCTION SUBPROGRAM TO  CALCULATE DEL WATER    **
**********************************************************
**********************************************************

        DOUBLE PRECISION FUNCTION W_DEL(DELDOL,TEMP)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)

************************************
** CONVERT TEMP TO DEGREES KELVIN **
************************************
```

```
      TEMPK = TEMP+273.15D0

*********************************************
** CALCULATE EPSILON FOR DOLOMITE AND WATER **
*********************************************

      EPSDOL=(DEXP((3.2D0*(1.D6/TEMPK**2)-4.3D0)/1.D3)-1.D0)*1.D3

      W_DEL=(DELDOL-EPSDOL)/((EPSDOL/1.D3)+1.D0)

      RETURN
      END


      SUBROUTINE PLOTEM_INV(XMIN,XMAX,XPLT,YCONC,YCL_INV,
     +     YSUM_SCL_DEP,KOUNT)
**************************************************************************
*     SET UP THE GRAPHICS WINDOW FOR PLOTTING                            *
**************************************************************************

      PARAMETER(NUMA=15000,NUMB=2500)
      REAL XMIN,XMAX,YMIN,YMAX,XPLT(NUMA),YCONC(NUMA),
     +     YCL_INV(NUMA),YSUM_SCL_DEP(NUMA),SALT(500),STIME(500)

      OPEN(UNIT=61,FILE='SLTDEP.DAT',STATUS='OLD')

      WRITE(6,*)
      WRITE(6,*)'READING SEARLES SALT DATA'
      WRITE(6,*)

      DO 100 I=1,500
         READ(61,*,END=101)STIME(I),SALT(I)
100   CONTINUE

101   NSPTS=I-1

      CLOSE(UNIT=61)

      WRITE(6,*)'READ',NSPTS,' POINTS FROM SALT DATA'

      DO 150 I=1,NSPTS
         STIME(I)=STIME(I)*1.0E6
150   CONTINUE

      CALL UIS

      CALL PAGE(33.,22.)
      CALL COMPLX
      CALL HEIGHT(.35)

C$$$       FIND THE POSITION TO PUT THE GRAPH
      DO 300 I=1,3
         IF(I .EQ. 1)THEN
            CALL PHYSOR(4.,12.)
         ELSE IF(I .EQ. 2)THEN
            CALL PHYSOR(4.,7.)
         ELSE IF(I .EQ. 3)THEN
            CALL PHYSOR(4.,2.)
         ENDIF

         IF(I .EQ. 1)THEN
            CALL AREA2D (25.,8.)
         ELSE
            CALL AREA2D (25.,3.)
         ENDIF
```

```
          CALL GAPWID(.001)

C$$$   PUT THE HEADING ON THE PLOT

          IF(I .EQ. 1)THEN
             CALL YNAME ('CL (KG/M^2)$',100)
             CALL XNAME ('TIME$',100)
          ELSEIF(I .EQ. 2)THEN
             CALL YNAME ('CL CONC(MOL/LTR)$',100)
             CALL XNAME ('TIME$',100)
          ELSEIF(I .EQ. 3)THEN
             CALL YNAME ('CL INV (KG)$',100)
             CALL XNAME ('TIME$',100)
          ENDIF

******************************
** CALCULATE YMAX AND YMIN **
******************************

          IF(I .EQ. 1)THEN
             YMIN=YSUM_SCL_DEP(1)
             YMAX=YSUM_SCL_DEP(1)
             DO 1100 IMM = 2,KOUNT
                IF(YSUM_SCL_DEP(IMM) .LT. YMIN)YMIN=YSUM_SCL_DEP(IMM)
                IF(YSUM_SCL_DEP(IMM) .GT. YMAX)YMAX=YSUM_SCL_DEP(IMM)
1100         CONTINUE
             YMIN=YMIN-0.1D0*YMIN
             YMAX=YMAX+0.1D0*YMAX
          ELSEIF(I .EQ. 2)THEN
             YMIN=0.0
             YMAX=6.1
          ELSE IF(I .EQ. 3)THEN
             YMIN=YCL_INV(1)
             YMAX=YCL_INV(1)
             DO 1600 IMM = 2,KOUNT
                IF(YCL_INV(IMM) .LT. YMIN)YMIN=YCL_INV(IMM)
                IF(YCL_INV(IMM) .GT. YMAX)YMAX=YCL_INV(IMM)
1600         CONTINUE
             YMIN=YMIN-0.1D0*YMIN
             YMAX=YMAX+0.1D0*YMAX
          ENDIF

          CALL GRAF (XMIN,XMAX,XMAX,YMIN,YMAX,YMAX)

          CALL FRAME
          CALL THKCRV(.01)
          CALL MARKER(3)
          CALL SCLPIC(.3)

          IF(I .EQ. 1)THEN
             CALL CURVE(XPLT,YSUM_SCL_DEP,KOUNT,-1)
             CALL CURVE(STIME,SALT,NSPTS,0)
          ELSEIF(I .EQ. 2)THEN
             CALL CURVE(XPLT,YCONC,KOUNT,-1)
          ELSE IF(I .EQ. 3)THEN
             CALL CURVE(XPLT,YCL_INV,KOUNT,-1)
          ENDIF

          CALL ENDGR(IPLOT)

 300    CONTINUE

        RETURN
        END
```