A UNIVERSITY OF UTAH
DIRECT CONTACT GEOTHERMAL POWER
PROJECT REPORT

A Computer Program
for Determining the Thermodynamic
Properties of Water

Prepared for the
Energy Research and Development Administration
Under

Grant No. AER 75-01039

by

David H. Riemer
Harold R. Jacobs
Robert F. Boehm

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

JUN 11 1976
15598
DISCLAIMER

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

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

This work was prepared as a part of ERDA Grant No. AER-75-01039 which was originally funded as a National Science Foundation RANN Grant entitled, "Feasibility Study of the Application of Direct Contact Heat Exchangers to Power Cycles Utilizing Geothermal Brines." The work was conducted in the Mechanical and Industrial Engineering Department at the University of Utah under the direction of Drs. Harold R. Jacobs and Robert F. Boehm.

Appreciation for supplemental funding and encouragement is given to Dr. J. Kunze of Aerojet Nuclear Company, Idaho Falls, Idaho, for ERDA support under contract AT(10-1)1523.
NOMENCLATURE

\( A_{ij} \) \hspace{1cm} \text{constants required for properties}

\( C_i \) \hspace{1cm} \text{constants required for } \psi_o

\( F_i \) \hspace{1cm} \text{constants required for saturation pressure}

\( H \) \hspace{1cm} \text{enthalpy per lbm}

\( i \) \hspace{1cm} \text{index of summation}

\( j \) \hspace{1cm} \text{index of summation}

\( P \) \hspace{1cm} \text{pressure}

\( P_c \) \hspace{1cm} \text{critical pressure}

\( P_s \) \hspace{1cm} \text{saturation pressure}

\( Q \) \hspace{1cm} \text{part of } \psi, \text{ function of density and temperature}

\( R \) \hspace{1cm} \text{universal gas constant}

\( S \) \hspace{1cm} \text{entropy per lbm}

\( t \) \hspace{1cm} \text{temperature in } °C

\( T \) \hspace{1cm} \text{temperature in } °K

\( t_c \) \hspace{1cm} \text{critical temperature in } °C

\( U \) \hspace{1cm} \text{internal energy per lbm}

\( V \) \hspace{1cm} \text{specific volume per lbm}

\( \tau \) \hspace{1cm} \text{dimensionless temperature } 1000°K/T°K

\( \rho \) \hspace{1cm} \text{density}

\( \tau_i \) \hspace{1cm} \text{constant dependent on } j

\( \rho_{aj} \) \hspace{1cm} \text{constant dependent on } j

\( \psi \) \hspace{1cm} \text{Helmholtz free energy}

\( \psi_o \) \hspace{1cm} \text{Helmholtz free energy which is only temperature dependent}
ABSTRACT

This program was written to be used as a subroutine. The program determines the thermodynamic properties of water given any of the following pairs of knowns to define a thermodynamic state: pressure and entropy, pressure and enthalpy, pressure and quality, temperature and pressure, or temperature and quality. These five pairs of knowns allow the user to evaluate any thermodynamic cycle using water, as a working fluid. The basic equations came from Keenan, Keyes, Hill and Moore, Steam Tables, John Wiley and Sons, 1969. This report includes a complete derivation of equations, program listing, program symbol description, a complete set of flow charts and a sample steam turbine calculation.
INTRODUCTION

This subroutine computes the thermodynamic properties of water. The subroutine was written to allow for a wide range of applications such that it may be used to determine the properties of water in a conventional steam power plant or, as the authors developed it, for use in evaluating geothermal power cycles. The routine can be treated as a "black box" for the user with a limited background in programming or the program may be modified to the user's needs. This manual describes the general use of subroutine WATER and appendices contain a detailed derivation and explanation of WATER.

DEVELOPMENT OF PROPERTIES

The basic equation used in deriving the thermodynamic properties of water is the Helmhotz free energy equation since all properties can be determined from it.\(^1\)

\[
\psi = \psi_0(T) + RT (\ln \rho + \rho Q(p,T)) \tag{1}
\]

where

\[
\psi_0(T) = \sum_{i=1}^{6} C_i / \tau^{i-1} + C_7 \ln \tau + \frac{C_8 \ln \tau}{\tau} \tag{2}
\]

and

\[
Q = (\tau - \tau_c) \sum_{j=1}^{7} (\tau - \tau_{a j})^{j-2} \left[ \sum_{i=1}^{8} A_{i j}(\rho - \rho_{a j})^{i-1} \right.
\]
\[+ \ e^{-p_{0}} \sum_{i=9}^{10} A_{i j} \rho^{i-9} \left. \right] \tag{3}
\]

As stated previously all of the thermodynamic properties can be developed from (1) and they are as follows:
For a complete derivation of all equations see Appendix I. Recall that a thermodynamic state requires that two properties be known and for saturation states pressure and temperature are not independent. For evaluating an ideal power cycle the following processes are employed, constant entropy (turbine and pump), constant enthalpy (expansion valve), constant pressure (boiler or condenser). In order to determine properties for these processes, properties must be determined with one of the following pairs known: temperature and pressure, temperature and quality, pressure and quantities, a search technique must be used to find the density. Considering one of these at a time, first for temperature and pressure as known qualities, a search technique must be used to find the density in order that the remaining properties can be determined. A search is required because the pressure is a function of temperature and density and the equation cannot be solved explicitly for density. For all searches in WATER the Newton-Raphson iteration technique was used. This technique was selected because of its generally rapid convergence. Second, the saturation states can be solved in a similar manner if the saturation pressure were a function of temperature, such a function is:

\[
P(\rho, T) = \rho RT(1 + \rho Q(\rho, T) + \rho^2 \frac{\partial Q(\rho, T)}{\partial \rho})
\]

\[
U(\rho, T) = 1000R\rho \left( \frac{\partial Q(\rho, T)}{\partial \rho} \right) + \frac{d(\psi_0 \tau)}{d\tau}
\]

\[
H(\rho, T) = RT(\rho T(\frac{\partial Q(\rho, T)}{\partial \rho}) + 1 + Q(\rho, T) + \rho^2 \frac{\partial Q(\rho, T)}{\partial \rho}) + \frac{d(\psi_0 \tau)}{d\tau}
\]

\[
S(\rho, T) = -R(\ln \rho + \rho Q(\rho, T) - \rho \frac{\partial Q(\rho, T)}{\partial \rho}) - \frac{d\psi_0 \tau}{d\tau}
\]
This equation allows the saturation states to be solved just as the temperature and pressure case, by searching for both the saturated liquid and saturated vapor roots to Equation (4). After finding the saturated liquid and vapor properties the effect of a given quality can be determined. Third, for cases of pressure and entropy or enthalpy a modified version of the Newton-Raphson method was used. In these cases a search for two unknowns must be carried out, temperature and density, (see Appendix I for a complete derivation).

PROGRAM DEVELOPMENT

Subroutine WATER was written in FORTRAN V for the UNIVAC 1108 series computer but should easily be adapted to any similar FORTRAN compiler. Mixed mode arithmetic (real and integers) is required for subroutine PROPER, however, this can be easily changed for compilers which do not allow mixed mode calculations. The subroutine allows the user to use the program at two levels, first as a "black box" (the main text describes the use and general development of the program), second the program may be modified to best suit the user's needs, the appendices contain a complete documentation to allow the user to modify WATER. Subroutines were used throughout to allow for easy modification and as a means of determining any possible problems. The names of the subroutines used and a brief description appear in Table 1. All quantities required by these subroutines are global variables (common to all subroutines) placed in blank common. The general flow chart for WATER and detailed flow charts for all subroutines are contained in Appendix II. The program listing and symbol table are in Appendix III.
Table 1. Listing of Subroutines Used in WATER and a Description of Their Functions

<table>
<thead>
<tr>
<th>SUBROUTINE NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>WATER</td>
<td>Main routine—determines process to be evaluated and calls proper routines</td>
</tr>
<tr>
<td>PSATUR</td>
<td>Computes saturation pressure given temperature</td>
</tr>
<tr>
<td>TSATUR</td>
<td>Determines saturation temperature given pressure by using Newton-Raphson method</td>
</tr>
<tr>
<td>DFINN</td>
<td>Determines density given pressure and temperature by using Newton-Raphson method</td>
</tr>
<tr>
<td>PROPER</td>
<td>Calculates all properties and their derivatives for use in search routines</td>
</tr>
<tr>
<td>LIQDEN</td>
<td>Sets up search for saturated liquid properties</td>
</tr>
<tr>
<td>VAPDEN</td>
<td>Sets of search for saturated vapor properties</td>
</tr>
<tr>
<td>PHFIND</td>
<td>Performs search for temperature and density given pressure and enthalpy for superheated vapor or compressed liquid by using a modified Newton-Raphson method</td>
</tr>
<tr>
<td>PSFIND</td>
<td>Performs search for temperature and density given pressure and entropy for superheated vapor or compressed liquid by using a modified Newton-Raphson method</td>
</tr>
<tr>
<td>PRINT</td>
<td>Prints out results only if NPRINT=1</td>
</tr>
</tbody>
</table>

**USE OF WATER**

In order to use WATER the user must supply to the subroutine the following information (in the order that they appear in the parameter list):

1. The process for which properties are to be determined (CYCLE)
   - The allowable values of CYCLE are the following pairs of knowns:
     - TP—temperature and pressure
     - TX—temperature and quality
PX-pressure and quality
PH-pressure and enthalpy
PS-pressure and entropy

These are the only allowable values of CYCLE and the order is important, i.e., PT or SP, etc., are illegal and will generate an error termination of WATER

2. The value of the first known specified by the value of CYCLE(FGIVEN).

3. The value of the second known specified by the value of CYCLE(SGIVEN)

4. The line number from which the call on WATER originated.
   (This is used to aid the user in locating an error should WATER terminate due to improper data)

5. State number. This is an arbitrary number assigned by the user to a location in his thermodynamic cycle (used only if print out is desired, to inform the user which state in the thermodynamic cycle properties are being determined).

6. Print out selection--if =1 then print out, ≠1 then no print out.

WATER will return the following information (in the order they appear in the parameter list:

1. Quality-applies to saturation states (if this is greater than 1 then it indicates superheat and if it is less than 0 it indicates a compressed liquid state)

2. Entropy (BTU/lbm°R)

3. Enthalpy (BTU/lbm)

4. Specific volume (ft³/lbm)
5. Internal energy (BTU/lbm)
6. Temperature (°F)
7. Pressure (psia)
8. Saturated liquid entropy (BTU/lbm-°R)
9. Saturated liquid enthalpy (BTU/lbm)
10. Saturated liquid specific volume (ft³/lbm)
11. Saturated liquid internal energy (BTU/lbm)
12. Saturated vapor entropy (BTU/lbm-°R)
13. Saturated vapor enthalpy (BTU/lbm)
14. Saturated vapor specific volume (ft³/lbm)
15. Saturated vapor internal energy (BTU/lbm)
16. Error termination flag = 1 error termination

Note that all saturation properties for the cases of superheated vapor and compressed liquid are at the pressure given. A typical call on WATER would look like the following:

```
10 CALL WATER ('TP', TEMP, PRESS, 10, 1, 1, X, S, H, V, U, T, P, SL, HL, VL, UL, SV, HV, VV, UV, NERROR)
```

This call on WATER will evaluate properties given temperature and pressure. Subroutine WATER was written so as to be separate from the calling program except for the passage of parameters, therefore no external variables need be in common.

Appendix IV contains a sample run to illustrate the use of WATER. A sample steam turbine is evaluated using WATER to show the use of WATER in cycle evaluations.
For a complete analysis of the accuracy of this generalized equation for thermodynamic properties see Reference 1. The values found from this program may vary slightly with those tabulated in Reference 1 due to the fact that on convergence tests a tolerance of ± 0.5 percent was used.
APPENDIX I

DERIVATION OF ALL EQUATIONS REQUIRED FOR WATER
A. Derivation of Properties

From Reference (1)

$$\psi = \psi_0(T) + RT(\ln \rho + \rho Q(\rho, \tau))$$  \hspace{1cm} (1)

where

$$\psi_0(T) = \sum_{i=1}^{6} \frac{C_i}{\tau^{i-1}} + C_7 \ln \tau + \frac{C_8 \ln T}{\tau}$$  \hspace{1cm} (2)

and

$$Q = (\tau - \tau_c) \sum_{j=1}^{7} (\tau - \tau_{aj})^{j-2} \left[ \sum_{i=1}^{8} A_{ij}(\rho - \rho_{aj})^{i-1} + e^{-E_\rho} \sum_{i=9}^{10} A_{ij}\rho^{i-9} \right]$$  \hspace{1cm} (3)

From Reference (3)

$$S = - (\frac{\partial \psi}{\partial T})_\rho$$  \hspace{1cm} (AI.1)

$$P = \rho \left( \frac{\partial \psi}{\partial \rho} \right)_T$$  \hspace{1cm} (AI.2)

$$U = \left( \frac{\partial (\psi T)}{\partial T} \right)_\rho$$  \hspace{1cm} (AI.3)

To determine $S$ substitute Equation (2) into Equation (AI.1)

$$S = - \frac{\partial}{\partial T} \left( \psi_0(T) + RT(\ln \rho + \rho Q(\rho, \tau)) \right)$$

differentiating

$$S = - \left( \frac{d\psi_0}{dT}(T) + (\ln \rho + \rho Q(\rho, \tau) + RT(\frac{\partial Q(\rho, \tau)}{\partial T})_\rho \right)$$

differentiating $\tau$

$$d\tau = -\frac{T}{T} dT$$

substituting in

$$\frac{\partial Q(\rho, \tau)}{\partial \tau} = \frac{\partial Q(\rho, \tau)}{\partial \tau}; \; \frac{d\tau}{dT} = -\frac{T}{T} \frac{\partial Q(\rho, \tau)}{\partial \tau}$$

$$S = -R(\ln \rho + \rho Q(\rho, \tau) - \rho_T(\frac{\partial Q(\rho, \tau)}{\partial T})_\rho) - \frac{d\psi_0(T)}{dT}$$  \hspace{1cm} (AI.4)
To determine \( P \), substitute (2) into (AI.2)

\[
P = \rho^2 \frac{\partial}{\partial \rho} \left( \psi_o(T) + RT(\ln \rho + \rho Q(\rho,\tau)) \right)
\]

Differentiating

\[
P = \rho^2 RT\left(1 + \frac{1}{\rho} \frac{\partial Q(\rho,\tau)}{\partial \rho} \right) + \rho \frac{\partial Q(\rho,\tau)}{\partial \rho}
\]

Simplifying

\[
P = \rho RT \left(1 + \frac{1}{\rho} \frac{\partial Q(\rho,\tau)}{\partial \rho} \right) + \rho^2 \frac{\partial Q(\rho,\tau)}{\partial \rho}
\]

To determine \( U \), multiplying (2) by \( \tau \)

\[
\psi_T = \tau \psi_o(T) + RT (\ln \rho + \rho Q(\rho,\tau))
\]

Substituting \( \tau = 1000/T \)

\[
\psi_T = \tau \psi_o(T) + 1000R \ln \rho + \rho Q(\rho,\tau)
\]

Substituting this into (2) and differentiating

\[
U = \frac{d(\psi_T(T)\tau)}{dT} + 1000 \ln \rho + \rho \frac{\partial Q(\rho,\tau)}{\partial \rho}
\]

To determine \( H \), use definition of \( H = u + pv \) or \( H = u + p/\rho \)

Substituting in \( U \) and \( P \)

\[
H = RT \left(1 + \frac{\partial Q(\rho,\tau)}{\partial \rho} \right) + 1 + \rho Q(\rho,\tau) + \rho^2 \frac{\partial Q(\rho,\tau)}{\partial \rho} + \frac{d(\psi_T(T)\tau)}{dT}
\]

To determine \( P, U \) and \( S \) requires the following derivatives \( \frac{\partial Q}{\partial \rho}, \frac{\partial Q}{\partial \rho} \),

\[
\frac{d(\psi_T(T)\tau)}{dT}, \frac{d\psi_T(T)}{dT}
\]
1. \( \frac{\partial Q}{\partial \rho} \)\(_\tau\)

Differentiating (3)

\[
\frac{\partial Q}{\partial \rho} = (\tau-\tau_c) \sum_{j=1}^{7} (\tau-\tau_{aj})^{j-2} \left( \sum_{i=1}^{8} A_{ij}(i-1)\ (\rho-\rho_{aj})^{i-2} \right)
\]

\[
-\Delta e^{-\Delta \rho} \sum_{i=9}^{10} A_{ij} \rho^{i-9} + e^{-\Delta \rho} \sum_{i=9}^{10} A_{ij} (i-9) \rho^{i-10}
\]

evaluating \( \sum_{i=9}^{10} A_{ij}(i-9)\rho^{i-10} = A_{10,j} \)

collecting terms

\[
\frac{\partial Q}{\partial \rho} = (\tau-\tau_c) \sum_{j=1}^{7} (\tau-\tau_{aj})^{j-2} \left( \sum_{i=1}^{8} A_{ij}(i-1)\ (\rho-\rho_{aj})^{i-2} \right)
\]

\[
+ e^{-\Delta \rho} (-E \sum_{i=9}^{10} A_{ij} \rho^{i-9} + A_{10,j}) \]

(\(Q\)I.8)

II. \( \frac{\partial Q}{\partial \tau} \)\(_\rho\)

Let \( B_j = \sum_{i=1}^{8} A_{ij} (\rho-\rho_{aj})^{i-1} + \sum_{i=9}^{10} e^{-\Delta \rho} A_{ij} \rho^{i-9} \)

therefore \( Q = (\tau-\tau_c) \sum_{j=1}^{7} (\tau-\tau_{aj})^{j-2} B_j \)

differentiating

\[
\frac{\partial Q}{\partial \tau} = \sum_{j=1}^{7} (\tau-\tau_{aj})^{j-2} B_j + (\tau-\tau_c) \sum_{j=1}^{7} (j-2)(\tau-\tau_{aj})^{j-3} B_j
\]

collecting terms and substituting in \( B_j \)
\[
\frac{\partial (\rho/\partial \tau)_0}{\partial \tau} = \sum_{j=1}^{7} [(\tau-\tau_c)(j-2)(\tau-\tau_{a,j})^{j-3}(\tau-\tau_{a,j})^{j-2}] \left( \sum_{i=1}^{8} A_{ij} \right) \\
(\rho - \rho_{a,j})^{i-1} + \sum_{i=9}^{10} e^{-E_P A_{ij} \rho^{i-9}} \quad (AI.9)
\]

III. \( \frac{d(\psi(T)\tau)}{d\tau} \)

multiplying (2) by \( \tau \)

\[
\psi_0(\tau) = \sum_{i=1}^{6} \frac{C_i}{\tau^{i-2}} + C_1 \tau \ln T + C_8 \ln T \]

noting that \( \ln T = \ln \frac{1000}{T} = \ln 1000 - \ln T \)

differentiating and collecting terms

\[
\frac{d(\psi_0(\tau)\tau)}{d\tau} = \sum_{i=1}^{6} \frac{C_i(2-i)}{\tau^{i-1}} + C_1(\ln T - 1) - \frac{C_8}{\tau} \quad (AI.10)
\]

IV. \( \frac{d\psi_0(\tau)}{dT} \)

substituting \( T = \frac{1000}{\tau} \) into (1), and differentiating and simplifying

\[
\frac{d\psi_0(\tau)}{dT} = \sum_{i=1}^{6} \frac{C_i(i-1)2^{2-1}}{1000} + \frac{C_7}{T} + \frac{C_8}{1000} (\ln T + 1) \quad (AI.11)
\]

Derivation of density search given temperature and pressure (DFIND)

basic iteration Newton-Raphson equation

\[
\rho_{i+1} = \rho_i - \left[ \frac{P(\rho_i, T) - P\text{ known}}{(\partial P/\partial \rho)_{T_i}} \right]
\]
evaluating \( \frac{\partial P}{\partial \rho} \) by differentiating AI.5 and simplifying

\[
\frac{\partial P}{\partial \rho} |_T = RT (1 + 2\rho Q + 4\rho^2 \frac{\partial Q}{\partial \rho} |_T + \rho^3 \frac{\partial^2 Q}{\partial \rho^2} |_T)
\]
(A.12)

\( \frac{\partial P}{\partial \rho} \) requires \( \frac{\partial^2 Q}{\partial \rho^2} \)

V. \( \frac{\partial^2 Q}{\partial \rho^2} \)

differentiating AI.8

\[
\frac{\partial^2 Q}{\partial \rho^2} = (\tau - \tau_c) \sum_{j=1}^{7} (\tau - \tau_{a_j})^{j-2} \left[ \sum_{i=1}^{8} A_{ij}(i-1)(i-2)(\rho - \rho_{a_j})^{i-3} \right]
\]

\[
-\frac{E \cdot \rho^2}{\tau - \tau_c} \sum_{i=y}^{10} A_{ij} + A_{10, j} + e^{E \cdot \rho} \left[ -E A_{10, j} \right]
\]

collecting terms

\[
\frac{\partial^2 Q}{\partial \rho^2} |_T = (\tau - \tau_c) \sum_{j=1}^{7} (\tau - \tau_{a_j})^{j-2} \left[ \sum_{i=1}^{8} A_{ij} (i-1)(i-2)(\rho - \rho_{a_j})^{i-3} + e^{E \cdot \rho} \right]
\]

\[
\left[ E^2 \sum_{i=9}^{10} A_{ij} \rho - 2E A_{10, j} \right]
\]
(AI.13)

Derivation of saturated temperature search given pressure (TSATUR).

basic Newton-Raphson iteration equation

\[
T_{i+1} = T_i - \left[ \frac{P_s(T_i)}{\frac{\partial P_s}{\partial T_i}} \right] \]

\[
\frac{\partial P}{\partial T} = P_c \exp[\tau 10^{-5} (t_c - t) \sum_{i=1}^{8} F_i (0.65 - 0.01 t)^{i-1}] \\
\cdot \left(1000 \cdot 10^{-5} \frac{(t_c - t) \sum_{i=1}^{8} F_i (0.65 - 0.01 t)^{i-1}}{dT}\right)
\]

simplifying

\[
\frac{dP}{dT} = \frac{P_S}{100} \cdot \left(1000 \cdot 10^{-5} \frac{(t_c - t) \sum_{i=1}^{8} F_i (0.65 - 0.01 t)^{i-1}}{dT}\right)
\]

differentiating and simplifying

\[
\frac{dP}{dT} = \frac{P_S}{100} \cdot \left(1000 \cdot 10^{-5} \frac{(t_c - t) \sum_{i=1}^{8} F_i (0.65 - 0.01 t)^{i-1}}{dT}\right)
\]

\[
+ \left[0.65 - 0.01 t\right] . \sum_{i=1}^{8} F_i (0.65 - 0.01 t)^{i-2}\]

(AI.14)

Derivation of temperature and density search given pressure and entropy or enthalpy (PSFIND or PHFIND)

Basic iteration equations

\[
\rho_{i+1} = \rho_i - \left[\frac{(P - P_{\text{known}}) \left(\frac{\partial I}{\partial P}\right)}{J(P, I)} - \left(I-I_{\text{known}}\right) \left(\frac{\partial P}{\partial T}\right) \rho\right]_i
\]

\[
T_{i+1} = T_i - \left[\frac{(I-I_{\text{known}}) \left(\frac{\partial P}{\partial T}\right) - (P-P_{\text{known}}) \left(\frac{\partial I}{\partial P}\right)}{J(P, I)}\right]_i
\]

where \(I\) is either \(H\) or \(S\)

and \(J(P, I) = \left(\frac{\partial P}{\partial P}\right)_T \left(\frac{\partial I}{\partial T}\right) - \left(\frac{\partial I}{\partial P}\right)_T \left(\frac{\partial P}{\partial T}\right)\)

A. \(\left(\frac{\partial P}{\partial T}\right)\)

Differentiating AI.5

\[
\left(\frac{\partial P}{\partial T}\right) = \rho R (1 + \rho Q + \rho^2 \left(\frac{\partial Q}{\partial P}\right)_T + T \rho \left(\frac{\partial Q}{\partial T}\right)_\rho + \left(\frac{2}{\partial P \partial T}\right)\right)
\]
substituting in \((\frac{\partial Q}{\partial T})_\rho = \frac{\partial Q}{\partial \tau} \frac{\rho}{T} \)

\[(\frac{\partial P}{\partial T})_\rho = \rho R (1 + \rho Q + \rho^2 \frac{\partial Q}{\partial \rho}_T) + T \rho \left( \frac{\partial Q}{\partial \tau} \rho + \rho \left( \frac{\partial^2 Q}{\partial \rho \partial \tau} \right) \right) \]  \( (AI.15) \)

B. \((\frac{\partial P}{\partial \rho})_T \) - was evaluated for density search (DFIND)

C. \((\frac{\partial S}{\partial T})_\rho \)

differentiating equation \(AI.4\)

\[\frac{\partial S}{\partial T} = \frac{dT}{\rho} \frac{\partial S}{\partial \rho} \]

\[= - \frac{T}{\rho} \left( -R \left( \frac{\partial Q}{\partial \tau} \right)_\rho - \rho \left( \frac{\partial Q}{\partial \tau} \right)_\rho + 2 \rho \left( \frac{\partial^2 Q}{\partial \tau^2} \right)_\rho \right) \]

simplifying

\[-\frac{T}{\rho} \left( -R \frac{\partial^2 Q}{\partial \tau^2} + \frac{T}{\rho} \frac{dT}{\partial \rho} \left( \frac{\partial^2 Q}{\partial \tau^2} \right)_\rho \right) \]

substituting in

\[\frac{\partial S}{\partial \rho} \left( \frac{dT}{\partial \rho} \right)_\rho \frac{\partial^2 Q}{\partial \tau^2} - \frac{\partial^2 Q}{\partial \tau^2} \]

\[= -\frac{T}{\rho} R \frac{\partial^2 Q}{\partial \tau^2} + \frac{T}{\rho} \frac{dT}{\partial \rho} \frac{\partial^2 Q}{\partial \tau^2} \]

\[\frac{\partial S}{\partial T} = -\frac{T}{\rho} R \frac{\partial^2 Q}{\partial \tau^2} - \frac{\partial^2 Q}{\partial \tau^2} \]

\[ (AI.16) \]

D. \((\frac{\partial S}{\partial \rho})_T \)

differentiating equation \(AI.4\)

\[\frac{\partial S}{\partial \rho}_T = -R \left( \frac{1}{\rho} + \rho \left( \frac{\partial Q}{\partial \rho} \right)_T \right) + Q - \tau \left( \frac{\partial Q}{\partial \tau} \right)_\rho - \rho \left( \frac{\partial^2 Q}{\partial \tau \partial \rho} \right) \]

noting that

\[\frac{\partial^2 Q}{\partial \tau \partial \rho} = \frac{\partial}{\partial \rho} \left( \frac{\partial Q}{\partial \tau} \right) = \frac{\partial}{\partial \rho} \left( \frac{dT}{\partial \tau} \left( \frac{\partial Q}{\partial \rho} \right) \right) \]

where

\[\frac{dT}{\partial \tau} = \frac{T}{\tau} \]

\[\frac{\partial S}{\partial \rho}_T = -R \left( \frac{1}{\rho} + Q + \rho \left( \frac{\partial Q}{\partial \rho} \right)_T - \tau \left( \frac{\partial Q}{\partial \tau} \right)_\rho + T \rho \left( \frac{\partial^2 Q}{\partial \rho \partial \tau} \right) \right) \]

\[ (AI.17) \]
E. \( \frac{\partial H}{\partial T} \rho \)

\[ \frac{\partial H}{\partial T} = \frac{dt}{d\tau} \left( \frac{\partial H}{\partial \tau} \right) \]

differentiating AI.7 with respect to \( \tau \)

\[ \frac{\partial H}{\partial \tau} = -\frac{R(1000)}{\tau^2} \left( \rho \frac{\partial Q}{\partial \tau} \rho + 1 + \rho Q + \rho^2 \left( \frac{\partial Q}{\partial \rho} \right)_T \right) \]

\[ + \frac{R(1000)}{\tau} \left( 2\rho \frac{\partial Q}{\partial \rho} + \rho T \frac{\partial^2 Q}{\partial^2 T} + \rho^2 \frac{\partial^2 Q}{\partial \rho \partial T} \right) + \frac{d^2 (\psi Q(T)T)}{dT^2} \]

collecting terms and substituting

\[ \frac{\partial^2 Q}{\partial \rho \partial T} = -\frac{T}{\tau} \frac{\partial^2 Q}{\partial T \partial \rho} \]

\[ \frac{dt}{d\tau} = -\frac{T}{\tau} \]

\[ \left( \frac{\partial H}{\partial \tau} \right) \rho = -R \left( 1 + \rho Q + \rho^2 \left( \frac{\partial Q}{\partial \rho} \right)_T - \rho T \frac{\partial^2 Q}{\partial \rho \partial T} \right) \]

\[ + \rho^2 \left( \frac{\partial^2 Q}{\partial \rho^2} \right)_T - \frac{T}{\tau} \frac{d^2 (\psi Q(T)T)}{dT^2} \]  

(AII.18)

F. \( \frac{\partial H}{\partial \rho} \) _T

differentiating AI.7

\[ \left( \frac{\partial H}{\partial \rho} \right)_T = R \left( \tau \frac{\partial Q}{\partial \tau} \rho + \rho T \frac{\partial^2 Q}{\partial^2 T} \right) + Q + 3\rho \left( \frac{\partial Q}{\partial \rho} \right)_T \]

\[ + \rho^2 \left( \frac{\partial^2 Q}{\partial \rho^2} \right)_T \]

substituting in \( \frac{\partial^2 Q}{\partial T \partial \rho} = -\frac{T}{\tau} \frac{\partial^2 Q}{\partial T \partial \rho} \).

\[ \left( \frac{\partial H}{\partial \rho} \right)_T = R \left( \tau \frac{\partial Q}{\partial \tau} \rho - T \rho \frac{\partial^2 Q}{\partial T \partial \rho} \right) + Q + 3\rho \left( \frac{\partial Q}{\partial \rho} \right)_T + \rho^2 \left( \frac{\partial^2 Q}{\partial \rho^2} \right)_T \]  

(AII.19)
All of the following derivatives of $Q$ are required for the preceding
derivatives.

VI. \( \frac{\partial^2 Q}{\partial \tau^2} \rho \)

differentiating AI.9

let 
\[ M_\rho = \sum_{i=1}^{8} A_{ij} (\rho - c_{aj})^{i-1} + e^{-Ep} \sum_{i=9}^{10} A_{ij} \rho^{i-9} \]

\[ \frac{\partial^2 Q}{\partial \tau^2} \rho = \sum_{j=1}^{7} [(j-2)(\tau - \tau_{aj})^{j-3} + (\tau - \tau_c)(j-2)(j-3) (\tau - \tau_{aj})^{j-4} \]

\[ + (j-2) (\tau - \tau_{aj})^{j-3}] [M_\rho] \]

collecting terms

\[ \frac{\partial^2 Q}{\partial \tau^2} \rho = \sum_{j=1}^{7} [2(j-2)(\tau - \tau_{aj})^{j-3} + (\tau - \tau_c)(j-2)(j-3) (\tau - \tau_{aj})^{j-4}] \]

\[ [\sum_{i=1}^{8} A_{ij} (\rho - c_{aj})^{i-1} + e^{-Ep} \sum_{i=9}^{10} A_{ij} \rho^{i-9}] \quad (AI.20) \]

VII. \( \frac{\partial^2 Q}{\partial \rho \partial \tau} \)

differentiating AI.8 with respect to $\tau$

\[ \frac{\partial^2 Q}{\partial \rho \partial \tau} \tau = -\frac{1}{T} \frac{\partial^2 Q}{\partial \rho \partial \tau} \]

let 
\[ N_\rho = \sum_{i=1}^{8} A_{ij} (i-1)(i-2)(\rho - c_{aj})^{i-2} + e^{-E} \sum_{i=9}^{10} A_{ij} \rho^{i-9} + A_{10,j} \]
\[ \frac{\partial^2 Q}{\partial \rho \partial \tau} = \sum_{j=1}^{7} (\tau - \tau_{aj})^{j-2} \rho + (\tau - \tau_{C}) \sum_{j=1}^{7} (j-2)(\tau - \tau_{aj})^{j-2} \rho \]

collecting terms

\[ \frac{\partial^2 Q}{\partial \rho \partial \tau} = \sum_{j=1}^{7} \left[ (\tau - \tau_{aj})^{j-2} + (\tau - \tau_{C})(j-2)(\tau - \tau_{aj})^{j-2} \right] \]

\[ \left[ \sum_{i=1}^{8} A_{ij} (i-1)(i-2)(\rho - \rho_{aj})^{i-2} + e^{-E\rho} \sum_{i=9}^{10} A_{i,j} \rho^{i-9} + A_{10,j} \right] \] (AI.21)

VIII. \( \frac{d^2 \psi_{\phi}(T)}{dT^2} \)

differentiating AI.11

\[ \frac{d^2 \psi_{\phi}(T)}{dT^2} = \sum_{i=1}^{6} \frac{(i-1)(i-2)}{10^6} C_i t^{3-i} - \frac{C_7}{T^2} + \frac{C_8}{1000T} \] (AI.22)

IX. \( \frac{d^2 (\psi_{\phi}(T)\tau)}{dT^2} \)

differentiating AI.10

\[ \frac{d^2 (\psi_{\phi}(T)\tau)}{dT^2} = \sum_{i=1}^{6} (2-i)(1-i) \frac{C_i}{\tau^i} - \frac{C_7}{\tau} + \frac{C_8}{\tau^2} \] (AI.23)
APPENDIX II

General Flow Chart of Routine
WATER and Detailed Flow Charts of all Subroutines
GENERALIZED FLOWCHART OF SUBROUTINE WATER

START

DETERMINE PAIRS OF KNOWNS

TP

NO

T<T_c

YES

P>P_c

NO

P=PSAT

YES

FIND PSAT

NO

RETURN

YES

SUPERHEATED VAPOR STATE

RETURN

NO

P>P_c

YES

FIND PSAT

NO

ERROR

RETURN

SATURATION STATE

NO

P & T ARE NOT INDEPENDENT

RETURN

PX

CHECK X

YES

T<T_c

NO

P=P_c

YES

ERROR

RETURN

SATURATION STATE

NO

P<P_c

YES

SATURATED LIQUID

RETURN

YES

H>H_sat

NO

SUPERHEATED STATE

RETURN

SATURATED LIQUID

NO

H<H_sat

NO

COMPRESSED STATE

RETURN

SATURATION STATE

NO

SATURATED VAPOR

RETURN

YES

SATURATED VAPOR

RETURN

FIND TSAT

SATURATION STATE

RETURN

NO

P<P_c

NO

FIND TSAT

YES

TEMPERATURE AND VOLUME

RETURN

SATURATED VAPOR

SATURATION STATE

RETURN

RETURN
3

QUAL=SGIVEN

NO

QUAL=0.0

ERROR

QUAL=0.0

YES

TEMPF=FGIVEN
PRESS=SGIVEN
TEMPK=(TEMPF+459.6)/1.8

YES

TEMPF.705
NO

YES

PSATUR

YES

PRESS=PSAT
NO

YES

PRESS=PSAT
NO

YES

QUAL=2.0

DGUESS=1.0x10^-5
DENTOL=1.0x10^-7

QUAL=2.0

DGUESS=0.99
DENTOL=1.0x10^-7

YES

TEMPF.705
NO

QUAL=3.0

YES

RETURN

NO

CYCLE='TX'

NO

ERROR

QUAL=1.0

YES

ERROR

QUAL=1.0

YES

5

6

7
ENTR*(1.0-QUAL) * LSATS + QUAL * VSATS

TEMPF+FGIVEN

YES

TEMPK > 705

NO

TEMPK = (TEMPF + 460) / 1.8

ERROR

RETURN

PRESS+PSAT

NO

PSATUR

YES

TEMPK = 705

ERROR

RETURN

ENTH = (1.0-QUAL) * LSATH + QUAL * VSATH

VOLUME = (1.0-QUAL) * LSATV + QUAL * VSATV

RETURN

PRESS+FGIVEN

PKnown+FGIVEN

YES

PRESS+PC

NO

ERROR

RETURN

DFINO

ENTH = VLCAL

PRINT

RETURN
DETAILED FLOWCHART OF SUBROUTINE LIQDEN

START

DGUESS=0.96
DTOLERANCE=1x10^{-7}

DFIND

VF = V
UF = U
HF = H
SF = S

RETURN

DETAILED FLOWCHART OF SUBROUTINE VAPDEN

START

DGUESS=1x10^{-6}
DTOLERANCE=1x10^{-12}

DFIND

VF = V
UF = U
HF = H
SF = S

RETURN
DETAILED FLOWCHART OF SUBROUTINE PHFIND

START

LASTD=D
LASTT=T

PROPER

I=1
I=LIMT

YES

RETURN

PHFIND FAILED TO CONVERGE

D=LASTD - \( \frac{(P-P\text{KNOWN})H^T}{T^T} - \frac{(H-H\text{KNOWN})P^T}{T^T} \) JACOBIAN

T=LASTT - \( \frac{(H-H\text{KNOWN})P}{T} - \frac{(P-P\text{KNOWN})H}{T} \) JACOBIAN

LASTT=T
LASTD=D

PROPER

NO

CHECK TOLERANCE

NO

YES

RETURN
DETAILED FLOWCHART OF SUBROUTINE PRINT

START

CONVERT UNITS

NPRINT=0

YES

NO

RETURN

STATE, T F PRESSURE

YES

NO

QUAL<1.0

SATURATED PROPERTIES

YES

QUAL>1.0

SUPERHEATED VAPOR

NO

2.5<QUAL<0

COMPRRESSED LIQUID

NO

QUAL<2.5

YES

FINAL PROPERTIES

NO

ABOVE CRITICAL POINT
DETAILED FLOWCHART OF SUBROUTINE PSATUR

START

\[ r = \frac{1000}{T} \]
\[ t = T - 273.15 \]

SUM1 = 0.0
SUM2 = 0.0

\[ i = 1 \]
\[ i = i + 1 \]

YES

SUM1 = SUM1 + F(i) * (0.65 - 0.01t) \( t^{-1} \)
SUM2 = SUM2 + F(i)(i - 1) * (0.65 - 0.01t) \( t^{-2}/100 \)

NO

\[ \frac{dPSAT}{dT} = \text{PSAT}(\frac{t_c + \text{SUM1}}{t^2} - \text{SUM2}(\frac{t_c}{t_c} - 1))/100 \]

RETURN
DETAILED FLOWCHART OF SUBROUTINE PSFIND

START

LASTD=D

LASTT=T

PROPER

I=1

I>LMIT

YES

NO

PSFIND FAILED TO CONVERGE

RETURN

JACOBIAN = [(P-KNOWN) T S - (S-KNOWN) T S] / JACOBIAN

D = LASTD - JACOBIAN

T = LASTT - JACOBIAN

LASTT=T

LASTD=D

PROPER

CHECK TOLERANCE

NO

YES

RETURN
DETAILED FLOWCHART OF SUBROUTINE TSATUR

START

TSAT=340.0
TEMPK=340.0

PSATUR

I=1
I=I+1
I>LIMIT

TEMPK=TSAT-(PSAT-PKNOWN) / GPsat dY

TSAT=TEMPK

TEMPK=TEMPK/2

YES

PSATUR

NO

TEMPK=TSAT/10

RETURN

TSAT FAILED TO CONVERGE

RETURN

YES

CHECK TOLERANCE

YES

RETURN

NO
APPENDIX III
Complete Program Listing and Symbol Table Defining All Variables
Used in WATER and Its Subroutines
This subroutine computes the thermodynamic properties of water by the method developed by Kleinan, Keyes, Hill, and Moore in their book steam tables 1909. The types of states which can be evaluated are given the follow pairs:

1) TEMPERATURE  -  PRESSURE
2) TEMPERATURE  -  QUALITY
3) PRESSURE    -  ENTROPY
4) PRESSURE    -  ENTHALPY
5) PRESSURE    -  QUALITY

This allows for the analysis of all processes in a steam cycle to be analyzed. The program was written by David H. Riemer.

Subroutine WATER (CYCLE, GIVEN, CALLIN, NSTATI, NPRAWA, QUAL1, ENTR1, ENTH1, VOL1, UINT, TEMP1, PRESS1, LSATS1, LSAT1, LSATVL1, LSATUV1, VSA1, VSAT1, VSA1T1, VSATVL1, VSA1TUV1, VSA1TUVERROR)

DATA ITITER, UTOLER, CONVR, 1.005, 0.995, 0.4299229/
DATA ICRIT, PCRIT, TAUCRT, GAS1, GAS1, 374.136, 3203.6, 6.1, 5.44912, 66.9, 9366,
1 0.42151/
DATA ITERK1/200/
COMMON VSATS1, VSAT1, VSATUV1, LSATS1, LSAT1, LSATV1, TEMP1, PCRIT, TAUCRT, TGUESS, GAS1, GAS1, CONVR,
COMMON DENS1T, DENS1, DENS1H, DENS1D, DENS1HO, TEMP1, PRESS1, UINT, ENTH1, VOL1,
COMMON ENTR1, ENTH1, UDHRH1, UPDT1, SCAL1, HCAL1, DSDRH1, SJOULE1, SKNOWN1,
COMMON/ITSTAT, PCSTAT, BTOLER, PCAL, QUAL, PKnown, DSOT
COMMON/PRIN1A, /PRINT1, /STATE
PARAMETER M2=6, M1=5
IMPLICIT REAL(C)
INTEGER CALLIN
NSTATE=NSTATI
NPRAWA=NPRAWA
TEMPF=TEMPF1
ENTR=ENTRI
ENTH=ENTHI
PRESS=PRESS1
LSATH=0.0
LSA1=0.0
LSAT=0.0
VSAT=0.0
VSA1=0.0
NERR=0.0
WRITE(M2,9000)
IF (CYCLE.EQ.'TP') GO TO 1930
IF (CYCLE.EQ.'TX'.OR.CYCLE.EQ.'PA') GO TO 1720
IF (CYCLE.EQ.'PS') GO TO 1100
IF (CYCLE.EQ.'PH') GO TO 1330
WRITE(M2,9010) CYCLE, CALLIN
NERR=1
RETURN

C----------------------PRESSURE-ENTROPY-------------------------------------

1100 SKNUM=SGIVEN*1.0
PARNUM=PGIVEN
PRESS=PGIVEN
ENTR=SGIVEN
WRITE(M2,9050)
IF (PRESS.GT.PCM17) GO TO 1200
TEMP=500.0
IF (ENTR.GT.1.0380) DENSIT=1.0E-05
IF (ENTR.GT.1.0580) GO TO 1550
DENSIT=0.90
GO TO 1550
1200 CALL TSATUR
CALL VAPL
IF (ENTR/SGAL.GT.1100E-0) GO TO 1550
CALL LIQDEN
IF (ENTH/SCAL .LT. DTOLER) GO TO 1630
QUAL = (ENTH - LSATS) / (VSATS - LSATS)
ENTH = (1.0 - QUAL) * LSATH + QUAL * VSATH
QINT = (1.0 - QUAL) * LSATU + QUAL * VSATU
VOL = (1.0 - QUAL) * LSATV + QUAL * VSATV
CALL PRINT
GO TO 2200

C------------------------PRESSURE-ENTHALPY--------------------------

1000 IF (known=given)
    IF (known=given)
        PRESS = given
    END IF
    ENTH = given
    WRITE (N2, 9(60))
    IF (PRESS .LT. PCRIT) GO TO 1400
    TEMP = 500.0
    IF (ENTH .GT. 902.5) DENSIT=1.0E-05
    IF (ENTH .GT. 902.5) GO TO 1550
    DENSIT=0.90
    GO TO 1650

1300 CALL TSATUR
131 CALL VAPOR
    IF (known=.LT.HCAL.GT.TTOLER) GO TO 1550
    CALL LIQDEN
    IF (known=.LT.HCAL.LT.BTOLER) GO TO 1630
    QUAL = (known - LSATH) / (VSATH - LSATH)
    ENTH = (1.0 - QUAL) * LSATH + QUAL * VSATH
    QINT = (1.0 - QUAL) * LSATU + QUAL * VSATU
    VOL = (1.0 - QUAL) * LSATV + QUAL * VSATV
1351 CALL PRINT
136 GO TO 2200

1500 DENTOL=1.0E-10
150 IF (CICLE='G','PH') CALL PHFIND
155 IF (CICLE='G','PS') CALL PSFIND
156 ENTH=HCLAL
157 ENTH=SCAL
108     IF(PRESS.GT.PCRIT) QUAL=-3.0
109     IF(PRESS.LE.PCRIT) QUAL=+2.0
110     CALL PRINT
111     GO TO 2200
112  1630     DENTOL=1.0E-05
113     IF(CYCLE.EQ.1) GO TO 1680
114     TEMPK=160.0*15.4+9.630626*ENTR-2.640989*SRM(ENTR)-0.10592*ENTR**2
115  116     DENSIT=1.0/(62.4203*(0.12  -527.6609*ENTR+.2874426*SRM(ENTR))
117     CALL PSFIND
118     GO TO 1700
119  1680     TEMPK=1.8*(51.0+1.378531*ENTH+4.737845*SRM(ENTH)+4.540239E-04*
120     ENTH**2)
121     DENSIT=1.0E+04/(62.4203*(149.2-.185676*ENTH+2.671739*SRM(ENTH))
122     CALL PSFIND
123  1700     ENTH=HCAL
124     ENTR=SCAL
125     IF(PRESS.GT.PCRIT) QUAL=-3.0
126     IF(PRESS.LE.PCRIT) QUAL=-2.0
127     CALL PRINT
128     GO TO 2200
129  1720     QUAL=GETLEN
130     IF(QUAL.LE.0.0) GO TO 1780
131     WRITE (M2,9020)
132     QUA L=0.0
133     1780    IF(QUAL.LE.1.0) GO TO 1610
134     WRITE (M2,9030)
135     QUAL=1.0
136     C-------------------PRESSURE-QUALITY---------------------------------
137  1610    IF(CYCLE.EQ.1) GO TO 1870
138     PRESS =GIVEN
139     WRITE (M2,9070)
140     IF(PRESS.LE.PCRIT*0.99) GO TO 1890

NERRCR=1
WRITE (M2,9100)
RETURN
1890 CALL TSATUR
1892 GO TO 2120
C---------------------TEMPERATURE-QUALITY
1870 TEMPF=FGIVEN
1871 WRITE (M2,9080)
1872 IF (TEMFP.LE.705.0) GO TO 1900
1873 NERROR=1
1874 WRITE (M2,9110)
1875 RETURN
1876 1900 TEMPK=(TEMFP+459.67)/1.8
1877 CALL PSATUR
1878 PRESS =PSAT
1879 GO TO 2120
C---------------------PRESSURE-TEMPERATURE
1930 TEMPF=FGIVEN
1931 PRESS =SGIVEN
1932 WRITE (M2,9090)
1933 TEMPK=(TEMFP+459.67)/1.8
1934 IF (TEMFP.LT.705.0) GO TO 1960
1935 IF (PRESS.LT.PCH1T) GO TO 2060
1936 GO TO 2035
1938 1960 CALL PSATUR
1939 IF (PRESS /PSAT.GT.ITOIER.OR.PRESS /PSAT.LT.RTOIER) GO TO 2030
1940 WRITE (M2,9040) CALLIN,TEMPF,PRESS
1941 NERROR=1
1942 RETURN
1944 2030 IF (PRESS /PSAT.GT.ITOIER) GO TO 2080
1945 2035 SCAL=2.0
1946 DGUESS=1.0E-05
1947 DENTOL=1.0E-10
1948 GO TO 2040
1949 2060 DGUESS=0.99
1950 DENTOL=1.0E-07
160 IF (TEMPF.LT.705.) QUAL=-2.0
161 IF (TEMPF.GE.705.) QUAL=-3.0
162 2090 CALL DFIND
163 ENTHER=CAL
164 ENTR=SCAL
165 CALL PRINT
166 GO TO 2200
167 2120 IF (ABS(QUAL).LT.0.000001) GO TO 2150
168 CALL VAPDEN
169 IF (ABS(QUAL-1.0).LT.0.000001) GO TO 2160
170 2150 CALL LIQDEN
171 2160 ENTR=(1.0-QUAL)*LSATS + QUAL*VSATS
172 UINT=(1.0-QUAL)*LSATU + QUAL*VSATU
173 ENTH=(1.0-QUAL)*LSATH + QUAL*VSATH
174 VOLS=(1.0-QUAL)*LSATV + QUAL*VSATV
175 CALL PRINT
176 2200 TEMPF=TEMPF
177 VOL1=VOL
178 ENTR1=ENTR
179 ENTH1=ENTH
200 UINT1=UINT
201 PRESS1=PRESS
202 QUAL1=QUAL
203 LSATS1=LSATS
204 LSATU1=LSATU
205 LSATH1=LSATH
206 LSATV1=LSATV
207 VSATS1=VSATS
208 VSATH1=VSATH
209 VSATU1=VSATU
210 VSATV1=VSATV
211 9000 FORMAT(1H1,///13X,******** THERMODYNAMIC PROPERTIES OF WATER
212 1 **********,///)
213 9010 FORMAT(1H1,4X,**ERROR** CYCLE = 'N3', CYCLE MUST EQUAL IP, TX,
214 1 FX, PS, PH, LAST CALL ON WATER CAME FROM LINE',16,///,10X,'WATER T
215 ZEROMINATED')
9020 FORMAT(1H,'*X,***ERROR*** A QUALITY OF < 0 IS NOT DEFINED. ASSUME
10 TO BE ZERO*)
9030 FORMAT(1H,'*X,***ERROR*** A QUALITY OF > 1 IS NOT DEFINED. ASSUME
10 TO BE ONE*)
9040 FORMAT(1H,'*X,***ERROR*** TEMPERATURE AND PRESSURE ARE NOT INDEP
ENT UNDER THE SATURATION DOME,"/",*X,"WATER TERMINATED, LAST CAL
2L CAME FROM LINE *,*L/,*10X,"TEMPERATURE = ",F15.5,*10X,"PRESSURE = 
3 *,*F15.5)
9050 FORMAT(1H,*,16X,"PRESSURE AND ENTROPY")
9060 FORMAT(1H,*,16X,"PRESSURE AND ENTHALPY")
9070 FORMAT(1H,*,16X,"PRESSURE AND QUALITY")
9080 FORMAT(1H,*,16X,"TEMPERATURE AND QUALITY")
9090 FORMAT(1H,*,16X,"TEMPERATURE AND PRESSURE")
9100 FORMAT(1H,'*X,***ERROR*** A SATURATION STATE DOES NOT EXIST FOR P
PRESSURES ABOVE THE CRITICAL PRESSURE(3203,APSIA), WATER TERMINATED
2 *)
9110 FORMAT(1H,'*X,***ERROR*** A SATURATION STATE DOES NOT EXIST FOR T
TEMPERATURES ABOVE THE CRITICAL TEMPERATURE(705 DEG.F), WATER TERM
2 NATED*)
200 RETURN
END
DEFN A

USES NEWTON-RAPHSON ITERATION TO LOCATE DENSITY GIVEN PRESSURE
AND TEMPERATURE.

SUBROUTINE DFNU

PARAMETER M2=6

COMMON VSATS, VSATH, VSATU, VSATV, LSATS, LSATH, LSATU, LSATV,
1 TEMPK, TCRT, PCRT, TAUER, TAUERG, GAS, GAS1, CONVR,
2 DPSUT, UGUESS, DENSIT, DPORHO, TEMPF,PRESS, VIN, ENTH, VOL,
3 ENTR, DDT, DRRHO, DDT, SCAL, RSCP, DPORHO, SIG, JOLLE, SKNWN,
4 ITER, PSAT, DENTOL, TTOLER, RTOLER, PCAL, QCAL, PKNWN, DSNT

IMPLICIT REAL(L)

DENSIT=UGUESS
LASTD2=UGUESS
CALL PROPER
DO 2560, I=1, ITER

2510 DENSIT=LSTD2-(PCAL-PRESS)*DPORHO
21 IF(DENSIT.LE.1.0E-10) DENSIT=LSTD2/10.0
22 CALL PROPER
23 IF(PCAL/PRESS .LT.RTOLER.AND.PCAL/PRESS .LT.TTOLER.OR.
24 .ABS(LSTD2-DENSIT).LT.DENTOL) RETURN
25 LASTD2=DENSIT
26 CONTINUE
27 2560 CONTINUE
28 ARIT((K,9800)) ITER
29 9800 FORMAT(*** DENSITY SEARCH FAILED TO CONVERGE IN:IT, ITERATIONS
30 1: RESULTS MAY NOT BE VALID:****)
31 RETURN
32 END
C******************************************************************************************
C
C L I Q U E N

C SET UP SEARCH FOR SATURATED LIQUID DENSITY

C******************************************************************************************
C SUBROUTINE LIQUEN
COMMON VSATW, VSAITH, VSATV, LSATW, LSATH, LSATV,
1 TEMPK, TCRIT, PCRIT, TAUCRT, TGUSS, GAS, GAS1, CONVR,
2 DPDUR, DGUSS, DENSIT, DPDUR1, TEMP1, PRESS, UINT, ENTH, VOL,
3 ENTR, JBOT, JDRHO, LRDT, SCAL, HCAL, DSDKHO, SJOULE, SKNOWN,
4 ITERAT, PSAT, DENTOL, TTULEK, BTOLER, PCAL, GAUL, PKNOWN, DSST

IMPLICIT REAL (L)
UGUSS=0.96
DENTOL=1.0E-07
CALL DFIND
LSATV=VOL
LSATU=UINT
LSATS=SCAL
LSATN=HCAL
RETURN
END
C SUBROUTINE PHFIND
C IMPLICIT REAL(L)
C PARAMETER M2=6
C COMMON VSATS, VSATH, VSATV, LSATS, LSATH, LSATU, LSATV,
C TEMPK, TCRT, PCRT, TAUCRT, TGUESS, GAS, GAS1, CONVR,
C DPSAT, DGUESS, DENSIT, DPHRHO, TEMPF, PRESS, UINT, ENTH, VOL,
C ENTR, DHT, DHRHO, DPDT, SCAL, HCAL, DSDPHO, SJOULE, SKNOWN,
C ITERAT, PSAT, DENTOL, TTOLER, BTOLER, PCAL, GUAL, PKNOWN, DSDT
C REAL JACOB
C I Last=DENSIT
C TLAST=TEMPK
C CALL PROPER
C DO 3470 I=1, ITLHAT
C JACOB=DPHRHO*DHT-DHRHO*DPDT/JACOB
C DENSIT=DLAST-(PCAL-PRESS )*DHT-(HCAL-ENTH)*DPDT)/JACOB
C IF (DENSIT.LE.1.0E-10) DENSIT=DLAST/10.0
C TEMPK=TLAST-(HCAL-ENTH)*DPHRHO-(PCAL-PRESS )*DHRHO)/JACOB
C IF (TEMPK.LE.1.0E-10) TEMPK=300.0
C CALL PROPER
C IF (PCAL/PRESS .LT.TTOLER.AND.PCAL/PRESS .GT.BTOLER.AND.
C HCAL/ENTH.LT.TTOLER.AND.HCAL/ENTH.GT.BTOLER) RETURN
C IF (ABS(DLAST-DENSIT).LT.DENTOL.AND.ABS(TLAST-TEMPK).LT.1.0E-04)
C RETURN
C TLAST=TEMPK
C 3470 CONTINUE
36 WRITE(92,9000) ITERAT
37 9000 FORMAT('***TEMPERATURE-DENSITY SEARCH, GIVEN PRESSURE & ENTHALPY, ')
38     'FAILED TO CONVERGE IN ',15,' ITERATIONS, RESULTS MAY NOT BE VALID')
39 2 J
40 RETURN
41 END
COMMON VSAT, VSATU, VSATV, LSATS, LSATH, LSATU, LSATV,
1 ILMKP, TCKT, PCKT, TUCKT, TGUSS, GAS, GASI, CONVR,
2 DPSUI, UGUES, DENS, DPHRU, TEMPF, PRESS, UNIT, ENTH, VOL,
3 ENTR, JMT, JHORH, LPET, SCAL, MCAL, DSDRHU, SJOLUE, SKNOWN,
4 ITERAT, PSAT, DENTOL, TTOLEW, BTOLER, PCAL, QUAL, PKNOWN, DSDT
COMMON/PRINTA/PRINTA/STATE
PARAMETER KZ=0
IMPLICIT REAL (L)
TEMPF=TEMPF+1.6-459.67
IF(NPRINT.LT.0) RETURN
WRITE(MZ,5040) NSTATE
WRITE(MZ,5050) TEMPF, PRESS
IF (QUAL.LT.1.0.AND.QUAL.GE.0.0) GO TO 3280
IF (QUAL.GT.1.0) WRITE(MZ,9060)
IF (QUAL.LT.0.0.AND.QUAL.GT.-2.5) WRITE(MZ,9070)
IF (QUAL.LT.-2.5) WRITE(MZ,9030)
GO TO 3370
3280 WRITE(MZ,9060) QUAL
WRITE(MZ,9130)
WRITE(MZ,9060) VSATU,LSATU
WRITE(MZ,9100) VSATH,LSATH
WRITE(MZ,9110) VSATS,LSATS
WRITE(MZ,9120) VSATV,LSATV
3370 WRITE(MZ,9140)
WRITE(MZ,9090) UNIT
WRITE(MZ,9160) ENTH
WRITE(MZ,9170) ENTR
WRITE(MZ,9120) VOL
Y000 FORMAT(1H,20X,'CRITICAL SOLUTION',/INDEX/)
35 9040 FORMAT (1H, 10(''), 16X, 'PROPERTIES OF WATER AT STATE ' ,I4,5(''))
37 9050 FORMAT (1H, 20X, 'TEMPERATURE = ', F7,2, ' DEG,F', '/')
39     1 21X, 'PRESSURE = ', F10,5, ' PSIA')
41 9060 FORMAT (1H, 20X, 'SUPERHEATED VAPOR', '////////')
42 9070 FORMAT (1H, 20X, 'COMPRESSED LIQUID', '////////')
44 9080 FORMAT (1H, 20X, 'SATURATED LIQUID-VAPOR EQUILIBRIUM', '21X,
45     1 'QUALITY = ', F6,4)
47 9090 FORMAT (1H, 20X, 'INTERNAL ENERGY (BTU/LBM)', 5X, F9,3, 'X', F9,3)
50 9100 FORMAT (1H, 20X, 'ENTHALPY', 5X, (BTU/LBM)', 5X, F9,3, 8X, F9,3)
53 9110 FORMAT (1H, 20X, 'SPECIFIC VOLUME (CU FT/LBM)', 5X, F9,4, 8X, F9,4, '////////
55     1 ///)
57 9120 FORMAT (1H, 20X, 'ENTROPY', 9X, '(BTU/LBM- R)', 5X, F9,6, 8X, F9,6)
59 9130 FORMAT (1H, '///', 51X, 'SATURATED', 8X, 'SATURATED', '///', 51X, 'VAPOR', '12X,
62     1 'LIQUID', '21X, 60(''R''))
64 9140 FORMAT (1H, '////////')
66  RETURN
68  END
COMPUTES PROPERTIES AND THEIR DERIVATIVES.

SUBROUTINE PROPER
COMM VSA, VSAI, VSAI, VSAT, LSAT, LSAT, LSAT, LSAT,
1 TEMPA, TCRIT, PCRIT, TAUCLT, TQGUESS, GAS, GAI, CONVR,
2 DPHI, DGUES, DENSE, DPH, DPH, DPH, DPH, DPH,
3 ENTH, ENTH, ENTH, ENTH, ENTH, ENTH, ENTH, ENTH,
4 SCAL, SCAL, SCAL, SCAL, SCAL, SCAL, SCAL, SCAL,
5 CAL, CAL, CAL, CAL, CAL, CAL, CAL, CAL,
6 DSDH, DSDH, DSDH, DSDH, DSDH, DSDH, DSDH, DSDH,
7 SJOULE, SJOULE, SJOULE, SJOULE, SJOULE, SJOULE, SJOULE, SJOULE,
8 SKNOW, SKNOW, SKNOW, SKNOW, SKNOW, SKNOW, SKNOW, SKNOW,
9 DSDT, DSDT, DSDT, DSDT, DSDT, DSDT, DSDT, DSDT,
10 IMPLICIT REAL(8)
11 DIMENSION A(I0, J), C(J)
12 DATA (A(I,J), C(J)) / 29.49293, -5.19856, 6.83353354,
13 1 -6.1564104, -0.3972405, -3.9861411, -0.60644554, -152.11317,
14 2 7.77741c, -0.149751, -0.72566108, -6.490282, 15.430501,
15 3 2.7407416, 274.64632, -33.301902, 65.326996, -9.2734289,
16 4 -47.746374, -29.142247, -5.1028207, -360.93828, -16.254022,
17 5 -20.109799, 4.312534, 56.325311, 29.568796, 3.963689, 342.1843,
18 6 1, -17.21107, 0.0, 0.0, 0.0, 0.0, -244.55042, 127.48742,
19 7 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
20 8 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
21 9 -410.3666, 307.3113, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
22 10 79.94797, 13.041253, -410.0566, -209.88866, -733.96648,
23 11 10.10171, 65.8150, 399.1757, 71.531353 /
24 DATA (C(J)) / 1857.065, 3229.12, -419.465, 36.6649, -20.5516,
25 1 4.85233, 40.0, -101.2497
26 VOL=1.0/(64., 4253*VENSIT)
27 TAU=1.0000/TEMPK
28 ∂3SUM=0.0
29 ∂4SUM=0.0
30 ∂5SUM=0.0
31 ∂6SUM=0.0
32 ∂7SUM=0.0

30  WSUM=0.0
37  ZSUM=0.0
30  DO 2840 J=1,7
39  IF(J.EQ.1) GO TO 2680
40  UCON=1.0
41  TAUCON=2.5
42  GO TO 2700
43  -2000 UCON=0.034
44  TAUCON=TAUCRT
45  2700 CONTINUE
46  ZSUM=0.0
47  ZSUM=0.0
48  WSUM=0.0
49  DO 2740 I=1,6
50  ZSUM=ZSUM+A(I,J)*(DENSIT-DCON)**(I-1)
51  ZSUM=ZSUM+A(I,J)*(I-1)*(DENSIT-DCON)**(I-2)
52  WSUM=WSUM+A(I,J)*(I-1)*(I-2)*(DENSIT-DCON)**(I-3)
53  2740 CONTINUE
54  ZSUM=0.0
55  DO 2780 I=7,10
56  ZSUM=ZSUM+A(I,J)*DENSIT**(J-9)
57  2780 CONTINUE
58  ZSUM=ZSUM*EXP(-4.8*DENSIT)
59  WSUM=WSUM*3.04
60  ZSUM=ZSUM+(TAU-1)*((SUM+ZSUM)
61  ZSUM=ZSUM+(TAU-1)*((J-2)*ZSUM+ZSUM)
62  _**((J-3)*ZSUM+ZSUM)
63  ZSUM=ZSUM+(TAU-1)*((J-2)*(TAU-1)*((J-3)*ZSUM+ZSUM+
64  1)*EXP(-4.8*DENSIT)*(J-3)**((J-3)*ZSUM+ZSUM)
65  ZSUM=ZSUM+(TAU-1)*((J-2)*(TAU-1)*((J-3)*ZSUM+ZSUM+
66  1)*EXP(-4.8*DENSIT)*(J-3)**((J-3)*ZSUM+ZSUM)
67  ZSUM=ZSUM+(TAU-1)*((J-2)*ZSUM+ZSUM)
68  ZSUM=ZSUM+(TAU-1)*((J-2)*WSUM+WSUM*9.6*EXP(-4.8*DENSIT)
69  _*(J-3)**((J-3)*ZSUM+ZSUM)
70  2690 CONTINUE
71  2690 CONTINUE
u = (TAU - TAUCR1) * z5SUM

DGURHO = (TAU - TAUCR1) * z4SUM

DGUTAU = z5SUM

z6SUM = 0.0

z7SUM = 0.0

z8SUM = 0.0

A7SUM = 0.0

U2 2990 1

A7SUM = A7SUM + (2 - 1) * (1 - 1) * C(1) / TAU**1

z6SUM = z6SUM + (2 - 1) * C(1) / TAU**1

z7SUM = C(1) * (1 - 1) * TAU**2 / 1000.0 + z7SUM

z8SUM = z8SUM + C(1) * (1 - 1) * (1 - 2) * TAU**3

CONTINUE

U2CUT2 = z8SUM + 1.0 - 0.6 * C(7) / (TEMPK**2 + C(8)) / (1000.0 * TEMPK)

U2DUD2 = (TAU - TAUCR1) * z2SUM

U2CUT = x7SUM - C(7) / (TAU + C(6) / TAU**2

U2DUD = z7SUM + C(7) / (7 * TEMPK**2) + C(8) / TAU

U2CUT = z7SUM + C(7) / (7 * TEMPK**2) + C(8) / TAU

PCAL = DENSIT * GAS1 * TEMPK * (1.0 + U*DENSIT + DENSIT**2*DQDRHO)

U1INT = CONVR*(1000.0*GAS*DENSIT*DGUTAU+DCTDT)

SCAL = CONVR*(-DAS*(LOG(DENSIT)+DENSIT**G*DENSIT*TAU*DQDTAU))

1 - UUDT / 1.0

SUULE = SCAL1*U

HCAL = CONVR*(GAS1*TEMPPK*(DENSIT*TAU*DQDTAU + 1.0 + DENSIT**G +

1 + DENSIT**2*DQURHO) + UCTDT)

1 - UDDT / (2*SUPH+2*SUM) - (-TAU/TEMPPK)

UPDHU = GAS1*TEMPPK*(1.0 + 2.0*G*DENSIT + 4.0 + DENSIT**2*DQDRHO

1 + DENSIT**3*U2DD2)

UHDT = CONVR*(GAS1*(1.0 + DENSIT**G + DENSIT**2*DQDRHO - TAU*DENSIT*DQDTAU

1 - DENSIT*TAU**2*D2DQA + TEMPK*DENSIT**2*DQDTAU) + (-TAU/TEMPPK*D2CDDT)

U2DUDH = GAS1*CONVR*TEMPPK*(TAU*DQDTAU - TEMPPK*DENSIT*DQDUT +

1 + 3.0*DENSIT*U2DUD2)

U2DUDH = GAS1*CONVR*TEMPPK*(TAU*DQDTAU - TEMPPK*DENSIT*DQDUD +

1 + 3.0*DENSIT*U2DUD2)

U2DUDH = GAS1*CONVR*TEMPPK*(TAU*DQDTAU - TEMPPK*DENSIT*DQDUD +

1 + 3.0*DENSIT*U2DUD2)

U2DUDH = GAS1*CONVR*TEMPPK*(TAU*DQDTAU - TEMPPK*DENSIT*DQDUD +

1 + 3.0*DENSIT*U2DUD2)
103  USDTE=COVR*(GAS*TAU**2/TEMP*DENS*O2GUTA+O2GUT2)/1.8
109  RETURN
110  END
SUBROUTINE PSATUR

COMMON VSAT1, VSAT2, VSAT3, LSATS, LSATH, LSATU, LSATV,

1 TEMP1, TCRIT, PCRIT, TACRIT, TGUSS, GAS, GAS1, CONVR,

2 UPSAT, DGUSS, DENS1, DPDRHO, TEMPF, PRESS, UINT, ENTH, VOL,

3 ENTR, SHOT, UNDRHO, UPDT, SCAL, HCAL, DSDRHO, SJOULE, SKNOWN,

4 ITERAT, PSAT, DENTOL, TTOLER, BTOLER, PCAL, QUAL, PKNOWN, DSĐT

IMPLICIT REAL(L)

DIMENSION F(8)

DATA (F(I), I=1,8) /-741.9242, -29.721, -11.52586, -0.86656,85,

1 0.1096, 0.439953, 0.2520658, 0.05218684/1

TAU=1.0000, 0/TEMPK

TEMPK=TEMPK-273.15

SUM1=0.0

SUM2=0.0

DO 2420 I=1,8

SUM1=SUM1+(1)*(0.65-0.01*TEMPK)**(I-1)

SUM2=SUM2+(1)*(I-1)*0.01*(0.65-0.01*TEMPK)**(I-2)

2420 CONTINUE

PSAT=EXP(TAU*1.0L-05*(TCRIT-TEMPK)*SUM1)*PCRIT

UPSAT=PSAT/100.0*((((TEMPK-TCRIT)/TEMPK)**2-1.0/TEMPK)*SUM1

1 -(TCRIT/TEMPK-TEMPK/TMPK)*SUM2)

RETURN

END
SUBROUTINE PSFIND
IMPLICIT REAL(L)
PARAMETER M2=6
COMMON VSATS, VSATH, VSATU, VSATV, LSATS, LSATH, LSATU, LSATV,
1 TEMPK, TCRT, PCRT, TAUER, TGUES, GAI, GAST, CONVR,
2 DPSDT, DGGUESS, DENSIT, DPURHO, TEMPF, PRESS, DTGT, ENTH, VOL,
3 ENTR, DHDT, DHRHO, DDPDT, SCAL, HCAL, DSDFHO, SJUOLE, SKNOW,
4 ITERAT, PSAT, DENTOL, TTOLER, ATOLER, PCAL, QUAL, PKNOW, DSDF
REAL JACOB,
DLAST=DENSIT
TLAST=TEMPK
CALL PROPER
DO 4000 I=1, ITERAT
3790 JACOB=DPURHO*DSDFHO*DPDT
24 DENSIT=DLAST-((PCAL-PRESS)*DSDF-(SJUOLE-SKNOW)*DDPDT)/JACOB
25 IF (DENSIT.LE.1.0E-10) DENSIT=DLAST/10.0
26 TEMPK=DLAST-(SJUOLE-SKNOW)*DPURHO-(PCAL-PRESS)*DSDFHO)/JACOB
27 IF (TEMPK.LE.0.0) TEMPK=360.0
28 CALL PROPER
29 IF (PCAL/PRESS.LT.TTOLER.AND.PCAL/PRESS.GT.ATOLER.AND.SCAL/ENTR.LT.
1 TTOLER.AND.SCAL/ENTR.GT.ATOLER) RETURN
31 IF (ABS(DLAST-DENSIT).LT.DENTOL.AND.ABS(TLAST-TEMPK).LT.1.0E-04)
32 1 RETURN
33 DLAST=DENSIT
34 TLAST=TEMPK
4000 CONTINUE
36  WRITE(*2,9000) ITERAT
37  9000 FORMAT(' ***TEMPERATURE-DENSITY SEARCH GIVEN PRESSURE AND ENTROPY
38      FAILED TO CONVERGE IN '15,' ITERATIONS. RESULTS MAY NOT BE VALID.'
39      2 )
40  RETURN
41  END
USES NEWTON-RAPHSON ITERATION TO LOCATE SATURATION TEMPERATURES GIVEN SATURATION PRESSURE.

SUBROUTINE TSATUR
PARAMETER M2=0
COMMON VSATS, VSATH, VSATU, VSATV, LSA, LSAH, LSAT, LSATV,
1 TEMPK, TCRIT, TCRI, TAU, TCRIT, TGM, GAS, GAS1, CONVR,
2 DSDT, DGUESS, DEHSIT, DSRHO, TEMPF, PRESS, UINT, ENTH, VOL,
3 ENTH, ENTH, DHRC, DFDT, SCAL, HCA, DSRHO, SJOULE, SKNOW,
4 ITER, IER, DDT, TTOL, BTOL, PCAL, QUAL, PKNOW, DSDT
IMPLICIT REAL(*)
TSAT=0.0
TEMPK=TSAT
CALL PSATUR
DO 2320 I=1, ITER
2200 TEMPK=TSAT=(PSAT-PKNOW)/LPSET
2210 IF (TEMPK-2.15,0.1,TCRIT) TEMPK=TCRIT+2.15
23 CALL PSATUR
24 IF (PSAT/PKNOW,GT,BTOL) AND (PSAT/PKNOW,LT,TTOL) GO TO 2330
25 TSAT=TEMPK
2520 CONTINUE
27 WHILE (M2,9000) ITER
2990 FORMAT(1,10SATURATED TEMPERATURE SEARCH FAILED TO CONVERGE IN
1.15. I. ITERATIONS; RESULTS MAY NOT BE VALID!!!)
2330 TSAT=TEMPK
31 RETURN
32 END
SUBROUTINE VAPDEN
IMPLICIT REAL(L)
COMMON VSAT, VSATV, VSAUV, VSAT, LSATS, LSAT, LSATV
1 TEMP, TCUR, P1, T1, P1, T1, T2, T2, T3, T3, T4, T4, T5, T5,
2 UPS, UD, GUESS, DENS, OPCO2, TEMPF, PRESS, UINT, ENTH, VOL,
3 ENTR, ENTH, OPCO2, OPDT, SCAL, HCAL, DORH, SJUDE, SKNOW,
4 ITER, PSAT, DENTOL, TVLEK, BTOL, PCAL, QCAL, PKNOW, DSCT
5 GUESS = 1.0E+00
6 DENTOL = 1.0E+12
7 CALL UFING
8 VSATV = VOL
9 VSAT = UINT
10 VSATS = SCAL
11 VSATH = HCAL
12 RETURN
13 END
<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A(I,J)</td>
<td>Constants required for Q</td>
</tr>
<tr>
<td>BTOLER</td>
<td>Lower tolerance on convergence and saturation tests</td>
</tr>
<tr>
<td>C(I)</td>
<td>Constants required for ( \Psi_0 )</td>
</tr>
<tr>
<td>CALLIN</td>
<td>Line number from which call on Water originated</td>
</tr>
<tr>
<td>CONVR</td>
<td>Conversion factor—converts joules/g to BTU/lbm = 0.4299229</td>
</tr>
<tr>
<td>CYCLE</td>
<td>Type of calculation to be made, CYCLE has the following possible pairs of knowns:</td>
</tr>
<tr>
<td></td>
<td>1) Temperature and Pressure = TP</td>
</tr>
<tr>
<td></td>
<td>2) Temperature and Quality = TX</td>
</tr>
<tr>
<td></td>
<td>3) Pressure and Quality = PX</td>
</tr>
<tr>
<td></td>
<td>4) Pressure and Enthalpy = PH</td>
</tr>
<tr>
<td></td>
<td>5) Pressure and Entropy = PS</td>
</tr>
<tr>
<td>DCDT</td>
<td>( \frac{d\Psi_0}{dT} )</td>
</tr>
<tr>
<td>DCON</td>
<td>( \rho_{ij} ) [0.634 for ( j=1 ), 1.0 for ( j&gt;1 )]</td>
</tr>
<tr>
<td>DCTDT</td>
<td>( \frac{d\Psi_{OT}}{dT} )</td>
</tr>
<tr>
<td>DENSIT</td>
<td>Density</td>
</tr>
<tr>
<td>DENTOL</td>
<td>Convergence tolerance on density in density search (DFIND)</td>
</tr>
<tr>
<td>DGUESS</td>
<td>Initial guess for density in DFIND when Temperature and Pressure are known</td>
</tr>
<tr>
<td>DHDRHO</td>
<td>( \left( \frac{\partial h}{\partial \rho} \right)_T )</td>
</tr>
<tr>
<td>DHDT</td>
<td>( \left( \frac{\partial h}{\partial T} \right)_\rho )</td>
</tr>
<tr>
<td>DLAST</td>
<td>Density from last iteration to be used in next iteration in density search, given pressure and entropy or enthalpy</td>
</tr>
<tr>
<td>DPDRHO</td>
<td>( \left( \frac{\partial P}{\partial \rho} \right)_T )</td>
</tr>
<tr>
<td>DPDT</td>
<td>( \left( \frac{\partial P}{\partial T} \right)_\rho )</td>
</tr>
<tr>
<td>DPSDT</td>
<td>( \frac{dP_{sat}}{dT} )</td>
</tr>
</tbody>
</table>
\[
\frac{\partial^2 Q}{\partial \rho \partial T}
\]
\[
\frac{\partial Q}{\partial \rho} T
\]
\[
\frac{\partial Q}{\partial T} \rho
\]
\[
\frac{\partial S}{\partial \rho} T
\]
\[
\frac{\partial S}{\partial T} \rho
\]
\[
\frac{d^2 \psi_0}{dT^2}
\]
\[
\frac{d^2 (\psi \tau)}{d\tau^2}
\]
\[
\frac{\partial^2 Q}{\partial \rho^2} T
\]
\[
\frac{\partial^2 Q}{\partial T^2}
\]

**ENTH**  
Final enthalpy (BTU/lbm)

**ENTHL**  
Final enthalpy, value returned from WATER (BTU/lbm)

**ENTR**  
Final entropy (BTU/lbm\(^{\circ}\)R)

**ENTR1**  
Final entropy, value returned from WATER (BTU/lbm\(^{\circ}\)R)

**F(I)**  
Constants required for saturated pressure calculation used in PSATUR

**FGIVEN**  
First known passed to WATER, i.e. Temperature for values of CYCLE of TP, TX and pressure for values of CYCLE of PX, PS, and PH

**GAS**  
Universal gas constant=0.46141 joule/g\(^{\circ}\)K

**GAS1**  
Universal gas constant=66.9366 psia-cc/g\(^{\circ}\)K

**HCAL**  
Enthalpy calculated in PROPER

**I**  
Index

**ITERAT**  
Maximum number of iterations allowed in any one search routine

**J**  
Index
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JACOB</td>
<td>Jacobian</td>
</tr>
<tr>
<td>LASTD2</td>
<td>Density from last iteration, in density search (DFIND) given temperature and pressure</td>
</tr>
<tr>
<td>LH</td>
<td>Enthalpy of saturated liquid, value returned from WATER (BTU/lbm)</td>
</tr>
<tr>
<td>LS</td>
<td>Entropy of saturated liquid, value returned from WATER (BTU/lbm^°R)</td>
</tr>
<tr>
<td>LSATH</td>
<td>Enthalpy of saturated liquid (BTU/lbm)</td>
</tr>
<tr>
<td>LSATS</td>
<td>Entropy of saturated liquid (BTU/lbm^°R)</td>
</tr>
<tr>
<td>LSATU</td>
<td>Internal energy of saturated liquid (BTU/lbm^°R)</td>
</tr>
<tr>
<td>LSATV</td>
<td>Specific volume of saturated liquid (ft^3/lbm)</td>
</tr>
<tr>
<td>LU</td>
<td>Internal energy of saturated liquid, value returned from WATER (BTU/lbm)</td>
</tr>
<tr>
<td>LV</td>
<td>Specific volume of saturated liquid, value returned from WATER (ft^3/lbm)</td>
</tr>
<tr>
<td>NERROR</td>
<td>Error flag, set if data is incorrect</td>
</tr>
<tr>
<td>NPRINA</td>
<td>NPRINA=1 print out results, NPRINA#1 no print out</td>
</tr>
<tr>
<td>NPRINT</td>
<td>NPRINT=1 print out results, NPRINT#1 no print out</td>
</tr>
<tr>
<td>NSTATE</td>
<td>State number to which properties are desired</td>
</tr>
<tr>
<td>NSTAT1</td>
<td>State number to which properties are desired</td>
</tr>
<tr>
<td>PCAL</td>
<td>Pressure calculated in PROPER(psia)</td>
</tr>
<tr>
<td>PCRIT</td>
<td>Critical pressure = 3203.6(psia)</td>
</tr>
<tr>
<td>PKNOWN</td>
<td>Known pressure(psia)</td>
</tr>
<tr>
<td>PRESS</td>
<td>Pressure (psia)</td>
</tr>
<tr>
<td>PRESS1</td>
<td>Pressure, value returned from WATER (psia)</td>
</tr>
<tr>
<td>PSAT</td>
<td>Saturated pressure calculated in PSATUR (psia)</td>
</tr>
<tr>
<td>Q</td>
<td>Q - part of Ψ</td>
</tr>
<tr>
<td>QUAL</td>
<td>Quality</td>
</tr>
<tr>
<td>QUAL1</td>
<td>Quality, value returned from WATER</td>
</tr>
</tbody>
</table>
SCAL
Entropy calculated in PROPER (BTU/lbm°R)

SGIVEN
Second known passed to WATER, i.e. quality for CYCLE=TX or PX; pressure for CYCLE=TP; enthalpy for CYCLE=PH; entropy for CYCLE=PS

SJOULE
Entropy (BTU/lbm°K)

SUM1
Intermediate result for saturated pressure calculation
\[ \sum_{i=1}^{8} F_i (0.65-0.01t)^{i-1} \]

SUM2
Intermediate result for saturated pressure calculation
\[ \sum_{i=1}^{8} F_i (0.65-0.01t)^{i-2}(i-1)/100 \]

TAU
1000/ T^φ(τ)

TAUCON
\[ \tau_{aj} = \begin{cases} \tau_c & \text{for } j=1 \\ 2.5 & \text{for } j>1 \end{cases} \]

TAUCRT
Critical \( \tau = \tau_c = 1.544912 \)

TCRIT
Critical temperature (\( t_c \)) = 347.136°C

TEMPC
Temperature (°C)

TEMPP
Temperature (°F)

TEMPF1
Temperature (°F), value returned from WATER

TEMPK
Temperature (°K)

TGUESS
Initial guess for saturation temperature search (TSATUR) given pressure (°K)

TLAST
Temperature after last iteration to be used for next iteration in (PHFIND or PSFIND) search given pressure and enthalpy or entropy (°)

TSAT
Saturation temperature (°C)

TTOLER
Upper tolerance on convergence and saturation condition tests

UINT
Internal energy (BTU/lbm)

UINT1
Final internal energy, value returned from WATER (BTU/lbm)

VH
Saturated vapor enthalpy, value returned from WATER (BTU/lbm)

VOL
Specific volume (ft³/lbm)
VOLL: Specific volume, value returned from WATER (ft^3/lbm)
VS: Saturated vapor entropy (BTU/lbm°R), value returned from WATER
VSATH: Saturated vapor enthalpy (BTU/lbm)
VSATS: Saturated vapor entropy (BTU/lbm°R)
VSATU: Saturated vapor internal energy (BTU/lbm)
VSATV: Saturated vapor specific volume (ft^3/lbm)
VU: Saturated vapor internal energy, value returned from WATER (BTU/lbm)
VV: Saturated vapor specific volume, value returned from WATER (ft^3/lbm)

All of the following variables are intermediate results

WSUM
\[ \sum_{i=1}^{8} A_{ij}(i-1)(i-2)(\rho_\omega \rho)_{i-3} \]
\[ + \sum_{i=1}^{10} e^{-E_\rho} \sum_{i=9}^{10} A_{ij}\rho \]

W1SUM
\[ \sum_{j=1}^{7} (\tau-\tau_{aj})^{j-2} (WSUM+W1SUM-2E_\rho A_{10,j}) \]

X7SUM
\[ \sum_{i=1}^{6} (2-i)(1-i) C_i /\tau \]

ZSUM
\[ \sum_{i=1}^{8} A_{ij}(\rho_\omega \rho)_{i-1} \]

Z1SUM
\[ e^{-E_\rho} \sum_{i=9}^{10} A_{ij}\rho \]

Z2SUM
\[ \sum_{i=1}^{8} A_{ij}(i-1)(\rho_\omega \rho)_{i-2} \]

Z3SUM
\[ \sum_{j=1}^{7} (\tau-\tau_{aj})^{j-2} \left[ \sum_{i=1}^{8} A_{ij}(\rho_\omega \rho)_{i-1} + e^{-E_\rho} \sum_{i=9}^{10} A_{ij}\rho \right] \]

Z4SUM
\[ \sum_{j=1}^{7} (\tau-\tau_{aj})^{j-2} \left( \sum_{i=1}^{8} A_{ij}(i-1)(\rho_\omega \rho)_{i-2} - e^{-E_\rho} \sum_{i=9}^{10} A_{ij}\rho \right) \]
\[ Z5SUM \sum_{j=1}^{7} (\tau - \tau_{aj})^{j-2} \times \sum_{j=1}^{8} \frac{(\tau - \tau_{aj})(j-2)(\tau - \tau_{aj})^{j-3}}{\rho_{ij}}(\rho_{aj})^{i-1} e^{-\epsilon}\sum_{i=1}^{10} A_{ij} \ker i - 9) \]

\[ Z6SUM \sum_{i=1}^{6} \frac{(2 - i) C_i}{\tau} \]

\[ Z7SUM \sum_{i=1}^{6} \frac{(1 - i) C_i \tau^{2 - 1}/1000}{\tau} \]

\[ Z8SUM \sum_{i=1}^{6} \frac{C_i (i - 1)(i - 2) \tau^{3 - 1}/10^6}{\tau} \]

\[ Z9SUM (\tau - \tau_c) \sum_{j=1}^{7} (j - 2)(\tau - \tau_{aj})^{j-3}(Z2SUM - 4.8ZSUM + e^{-4.8} A_{10,j}) \]

\[ A_{ij} \ker \]
APPENDIX IV

Sample run of WATER to illustrate
Its Use in Evaluating a Steam Turbine

The Program Listing and Output are Contained
TEST OF WATER
SAMPLE RUN OF SUBROUTINE WATER TO ILLUSTRATE ITS USE IN EVALUATING
A STEAM TURBINE (ISENTROPIC EXPANSION)

IMPLICIT_REAL(L)
READ(5,9000) PINLET,TINLET,PEXIST
10 CALL WATER('TP',TINLET,PINLET,10,1,i,x,s,h,v,u,t,p,l,s,l,h,l,v,l,u,v,
 1 v,h,v,v,u,v,n,ERROR)
  IF(NERROR.EQ.1) GO TO 60
  SINLET=5
  TINLET=H
  VINLET=V
  UINLET=U
20 CALL WATER('PS',PEXIST,SINLET,20,2,i,x,s,h,v,u,t,p,l,s,l,h,l,v,l,u,v,
 1 v,h,v,v,u,v,n,ERROR)
  IF(NERROR.EQ.1) GO TO 60
  HEXIT=H
  SEXIT=S
  VEXIT=V
  UEXIT=U
  TEXIT=T
50 DELTAH=HINLET-EXIT
WRITE(6,9100) TINLET,PINLET,EXIT,PEXIST,DELTAH
STOP
60 WRITE(6,9200)
9000 FORMAT(3F10.2)
9100 FORMAT(1H1,10x,'SAMPLE STEAM TURBINE CALCULATION USING SUBROUTINE
1water',/,'INLET TEMPERATURE = ',F15.5,/,15x,'INLET PRESSURE =
2 ',F15.5,/,15x,'EXIT TEMPERATURE = ',F15.5,/,15x,'EXIT PRESSURE
3 = ',F15.5,/,11x,'TOTAL WORK OUTPUT(PTU/LBM) = ',F15.5)
9200 FORMAT(1H1,10x,'SAMPLE TURBINE RUN FAILED TO CONVERGE!')
END
**Thermodynamic Properties of Water**

**Temperature and Pressure:**

**Properties of Water at State 1**

Temperature = 1008.00 deg F  
Pressure = 1400.0000 PSIA  
Saturated Vapor

<table>
<thead>
<tr>
<th>Property</th>
<th>Value (BTU/LBM)</th>
<th>Value (BTU/LBM-K)</th>
<th>Value (CU FT/LBM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal Energy</td>
<td>1342.836</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enthalpy</td>
<td>1493.560</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Entropy</td>
<td>1.609442</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specific Volume</td>
<td></td>
<td></td>
<td>0.5817</td>
</tr>
</tbody>
</table>
**THERMODYNAMIC PROPERTIES OF WATER**

**PRESSURE AND ENTROPY**

**PROPERTIES OF WATER AT STATE**

**TEMPERATURE** = 75.12 DEG.F
**PRESSURE** = .43000 PSIA
**SATURATED LIQUID-VAPOR EQUILIBRIUM**
**QUALITY** = .7759

<p>| SATURATED | SATURATED |</p>
<table>
<thead>
<tr>
<th>VAPOR</th>
<th>LIQUID</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTERNAL ENERGY (BTU/LBM)</td>
<td>1035.405</td>
</tr>
<tr>
<td>ENTHALPY    (BTU/LBM)</td>
<td>1094.278</td>
</tr>
<tr>
<td>ENTROPY     (BTU/LBM-K)</td>
<td>2.049976</td>
</tr>
<tr>
<td>SPECIFIC VOLUME (CU.FT/LBM)</td>
<td>740.7535</td>
</tr>
</tbody>
</table>

**INTERNAL ENERGY** (BTU/LBM) | 813.048
**ENTHALPY** (BTU/LBM)  | 858.727
**ENTROPY** (BTU/LBM-K) | 1.609442
**SPECIFIC VOLUME** (CU.FT/LBM) | 574.7490
SAMPLE STEAM TURBINE CALCULATION USING SUBROUTINE WATER

INLET TEMPERATURE = 1000.0000
INLET PRESSURE = 1400.0000

EXIT TEMPERATURE = .78,12062
EXIT PRESSURE = .45600

TOTAL WORK OUTPUT (BTU/HR) = 634.77387
REFERENCES


