Forschungszentrum Karlsruhe
Technik und Umwelt

Wissenschaftliche Berichte
FZKA 5588

Thermal Properties of Water

K. Thurnay
Institut für Neutronenphysik und Reaktortechnik
Projekt Nukleare Sicherheitsforschung

Juni 1995

Forschungszentrum Karlsruhe
Technik und Umwelt
Wissenschaftliche Berichte
FZKA 5588

Thermal Properties of Water

K. Thurnay
Institut für Neutronenphysik und Reaktortechnik
Projekt Nukleare Sicherheitsforschung

Forschungszentrum Karlsruhe GmbH, Karlsruhe
1995

Forschungszentrum Karlsruhe GmbH
Postfach 3640, 76021 Karlsruhe

ISSN 0947-8620

Abstract

The report describes AQUA, a code developed in the Forschungszentrum Karlsruhe to calculate thermal properties of the water in steam explosions. AQUA bases on the H.G.K. water code, yet supplies - besides of the pressure and heat capacity functions - also the thermal conductivity, viscosity and the surface tension. AQUA calculates in a new way the thermal properties in the two phase region, which is more realistic as the one used in the H.G.K. code. AQUA is equipped with new, fast runnig routines to convert temperature-density dependent states into temperature-pressure ones.

AQUA has a version to be used on line and versions adapted for batch calculations.

A complete description of the code is included.

Thermische Eigenschaften des Wassers.

Zusammenfassung.

Der Bericht befaßt sich mit dem Code AQUA. AQUA wurde im Forschungszentrum Karlsruhe entwickelt um bei der Untersuchung von Dampfexplosionen die thermischen Eigenschaften des Wassers zu liefern. AQUA ist eine Fortentwicklung des H.G.K.-Wassercodes, aber er berechnet - neben Druck- und Wärmeeigenschaften - auch die Wärmeleitfähigket, die Viskosität und die Oberflächenspannung. Im Zweiphasengebiet beschreibt AQUA die thermischen Eigenschaften mit einer neuen Methode, die realistischer ist, als das in der H.G.K.-Code dargebotene Verfahren. AQUA verfügt über schnell laufende Hilfsprogramme, die die Umwandlung der Temperatur-Dichte Darstellungen in die Druck-Dichte Darstellungen ermöglichen.

AQUA kann man sowohl im Dialogbetrieb benutzen als auch beim rechnen im Stapelverfahren.

Eine ausführliche Beschreibung des Programmsystems rundet den Bericht ab.

Glossary

Name	Code	Description
A	A	specific free energy [J/kg]
A^0		$A/(R_{gas} \cdot T)$
α	ALF	-0.1, critical exponent for the heat capacity
β	BET	0.325, critical exponent for the density
C_V	CV	heat capacity at constant volume [$J/(kg \cdot K)$]
	CVL	heat capacity at constant volume of the liquid [R_{gas}]
	CVV	heat capacity at constant volume of the vapor [R_{gas}]
C_V^0	CV0	C_V/R_{gas}
C_P	CP	heat capacity at constant pressure [$J/(kg \cdot K)$]
C_P^0	CP0	C_P/R_{gas}
	CPC	C_P/C_V
γ	GAM	1.24, critical exponent for $\partial P/\partial \rho$
	DL	density of the liquid [g/cm^3]
	DV	density of the vapor [g/cm^3]
	DX	density in g/cm^3
η	ETA	dynamic viscosity [$Pa \cdot$ sec]
η_c	ETC	critical viscosity, $3.95 \cdot 10^{-5}\,Pa \cdot$ sec
	ETL	dynamic viscosity of the liquid [$Pa \cdot$ sec]
	ETV	dynamic viscosity of the vapor [$Pa \cdot$ sec]
G	G	specific Gibbs function [J/kg]
G^x	GL	G in the saturated states
G^0	G0	$G/(R_{gas} \cdot T)$
	GASCON	the gas-law constant of the water in $J/g \cdot K$
H	H	specific enthalpy [J/kg]
H^0	H0	$H/(R_{gas} \cdot T)$
H_P	H1P	$\partial H/\partial P$, pressure derivative of H
H_ρ		$\partial H/\partial \rho$, density derivative of H

Name	**Code**	**Description**
χ_T	CHIT	reduced isothermal compressibility (s. Eq. [3.9])
IAPS		The International Association for the Properties of Steam
λ	XLMB	thermal conductivity [$W/(m \cdot K)$]
	XLL	thermal conductivity of the liquid [$W/(m \cdot K)$]
	XLV	thermal conductivity of the vapor [$W/(m \cdot K)$]
	MW	Molal mass (18.0152 g/mol)
P	P	pressure [Pa]
P_c	PC	critical pressure, 22.0549 MPa
$P^{\times}$	PS	vapor pressure
P_{SL}	PSL	pressure on the liquid spinodal
P_{SV}	PSV	pressure on the vapor spinodal
P_ρ	P1R	$\partial P/\partial \rho$, density derivative of the pressure
	PL1R	density derivative of the pressure in the saturated liquid
	PV1R	density derivative of the pressure in the saturated vapor
P_ρ^0	P1R0	$P_\rho/(R_{gas} \cdot T)$
$\partial P/\partial \rho \vert_S$	P1RS	density derivative of the pressure at constant entropy
P_T	P1T	$\partial P/\partial T$, temperature derivative of the pressure
	PL1T	temperature derivative of the pressure in the saturated liquid
	PV1T	temperature derivative of the pressure in the saturated vapor
P_T^0	P1T0	$P_T/(R_{gas} \cdot \rho)$
$P^{\#}$	PRN	Prandtl number, $C_P \cdot (\eta/\lambda)$
	Q	pressure in MPa-s
R_{gas}	RGAS	gas-law constant of the water, 461.522 $J/(kg \cdot K)$
ρ	RH	density [kg/m^3]
ρ_c	RHC	critical density, 322. kg/m^3
ρ_{SL}	RHSL	density of the liquid spinodal
ρ_{SV}	RHSV	density of the vapor spinodal
ρ_T^0	R1T0	reduced temperature-derivative of the density, $T/\rho \cdot \partial \rho/\partial T$

Name	Code	Description
r_L	RL1T0	reduced temperature-derivative of the density of the saturated liquid, $T/\rho \cdot d\rho_L/dT$
r_V	RV1T0	reduced temperature-derivative of the density of the saturated vapor, $T/\rho \cdot d\rho_V/dT$
S	S	specific entropy [$J/(kg \cdot K)$]
S_c	SC	critical entropy, $9.43 \cdot R_{gas} = 435.244\ J/(kg \cdot K)$
σ	SRFTS	surface tension [N/M]
	SL	specific entropy of the liquid [$J/(kg \cdot K)$]
	SREF	$7.6180802 \cdot R_{gas}$
S^0	S0	S/R_{gas}
T	T	temperature [K]
T_c	TC	critical temperature, 647.126 K
	TPS1T	$T \cdot dP^x/dT$ [Pa]
	TTPS2T	$T \cdot T \cdot d^2P^x/dT^2$ [Pa]
dT/dP^x	TS1P	inverse of the derivative of the vapor pressure [K/Pa]
τ	Y	scaled temperature, $1 - T/T_c$
U	U	specific internal energy [J/kg]
	UREF	$-4328.455039 \cdot R_{gas} \cdot T$
U^0	U0	$U/(R_{gas} \cdot T)$
U_ρ	U1R	$\partial U/\partial \rho$, density derivative of U
V	V	specific volume, $1/\rho$
v_s	VS	velocity of sound, $v_s^2 = \partial P/\partial \rho \|_S$
	VQ	vapor quality, $\dfrac{m_V}{m_L + m_V}$
w	W	$\sqrt{\tau}$
Z	Z	factor of reality, $(P \cdot V)/(R_{gas} \cdot T)$

Contents

Figures

Tables

1.0 Introduction

Water is the most commonly used working substance in chemistry and engineering. It is also the cooling medium of the pressurized and boiling water nuclear reactors. Therefore a solid knowledge of the thermophysical properties of the water in the range of possible temperatures, pressures and densities is essential to calculate hazardous events - which may emerge once in a while in a normally operating reactor.

In the Institut für Neutronenphysik und Reaktortechnik we are presently developing a thermohydrodynamical code, IVA3, /21/ to be able to calculate the consequences of some hypothesized reactor failures. To supply IVA3 with the thermophysical properties of the water we adopted the H.G.K.-equation of state, developed in the years 1981-84 by Haar, Gallagher and Kell, /10/ and recommended by the IAPS (s. /12/). The H.G.K.-equation of state calculates the various thermal properties via a continuous Helmholtz free energy function, which is defined over the whole range of the fluid states of water and is fitted to the available experimental data.

The corresponding computer code described in /10/ has some drawbacks and doesn't fit well to our IVA3-code. Item 1 : the code doesn't describe the properties viscosity, thermal conductivity and surface tension. Item 2 : the description of the properties in the mixed state is unphysical. Item 3 : IVA3 needs the thermophysical properties as functions of the temperature and pressure, whereas the H.G.K.-code supplies them as functions of the temperature and density. Although there is a modul in the H.G.K.-code, DFIND, which turns temperature-pressure states of water into corresponding temperature-density ones, yet DFIND consumes too much time during the iteration - needed for the conversion of these states.

In the following the code AQUA will be presented.

AQUA is an extension of the H.G.K.-code, modified to suit our special conditions. In the largest part of the temperature-density domain AQUA returns the same thermophysical property values as the H.G.K.-code, but in the unstable part of the two phase region the H.G.K.-properties are substituted with physically meaningful two phase state values.

AQUA calculates at a given (T , ρ) state besides of the pressure also the following thermal properties of the water :

- S, the specific entropy,
- the energy functions
    - U, the specific energy ,
    - H, the specific enthalpy ,
    - A, the specific free energy ,
    - G, the specific free enthalpy ,
- the pressure derivatives $\partial P/\partial \rho$ and $\partial P/\partial T$,
- the heat capacities C_P and C_V,
- the derivatives $\partial H/\partial P$ and $\partial \rho/\partial T$
- the viscosity η and the thermal conductivity λ,
- the Joule Thomson coefficient, $\partial T/\partial P \mid H$,
- the surface tension, the velocity of sound and the Prandtl number.

AQUA calculates the properties also as temperature-pressure dependent functions. It incorporates additional functions and procedures to speed up the (T , P) → (T , ρ) conversion.

AQUA has a set of routines (SAET01, WASSER, DAMPF, RDRDP, VD1) which serve to adjust AQUA to the conditions of the code IVA3.

AQUA needs and returns all the properties in SI-units.

AQUA calculates reasonable property-values if the temperature exceeds 260.00 K and if the density remains below 1200 kg/m^3. Trying to convert pressures to densities above 5000 K may lead to dissatisfaction.

2.0 A short description of the H.G.K.-code

2.1 The Helmholtz function

Haar, Gallagher and Kell developed and tested - in close cooperation with the International Association for the Properties of Steam (IAPS) - an analytic description for the thermal and caloric properties of the water, /10/ to replace the standard tables used formerly. The resulting properties are the Provisional IAPS Formulation 1982 for the Thermodynamic Properties of Ordinary Water Substance for Scientific and General Use.

Haar, Gallagher and Kell constructed their algorithm by developing a mathematical description for the Helmholtz function (alias free energy) $A(\rho, T)$. Using Eq. [A.4 - A.5] one can develop from A all the needed thermal and caloric properties (s. Appendix A, "Some thermal relations").

The description of the Helmholtz function consist from a simple base function to describe the liquid and supercritical states, of an additional residual function for a proper fitting in the mixed states and of a solely temperature-dependent term for the ideal gas states.

$$A(\rho, T) = A_{base}(\rho, T) + A_{residual}(\rho, T) + A_{idealgas}(T) \quad . \qquad [2.1]$$

2.2 The base function

is a simple analytic equation

$$A_{base}(\rho, T) = R_{gas} \cdot T \left[-\ln(1-y) - \frac{\beta - 1}{1-y} + \frac{\alpha + \beta + 1}{2 \cdot (1-y)^2} + \right.$$
$$\left. + 4y\left(\frac{\bar{B}}{b} - \gamma\right) - \frac{\alpha - \beta + 3}{2} + \ln\left(\frac{\rho \cdot R_{gas} \cdot T}{P_0}\right)\right] \quad . \qquad [2.2]$$

In the above eq.

$$y = \frac{b(T) \cdot \rho}{4} \qquad [2.3]$$

is a dimensionless density,

$$b = b_1 - b_2 \cdot \ln \frac{T_0}{T} + \sum_{j=3}^{7} b_j \cdot \left(\frac{T_0}{T}\right)^{j-2} \qquad [2.4]$$

is the 2nd virial and

$$\bar{B} = B_1 + \sum_{j=3}^{7} B_j \cdot \left(\frac{T_0}{T}\right)^{j-2} \qquad [2.5]$$

is the excluded volume. The pressure and temperature coefficients are

$$P_0 = 1.01325 \text{ bar} \qquad \text{resp.} \qquad T_0 = 647.073 \text{ K} \qquad [2.6]$$

and the greek letters mean

$$\alpha = 11. \quad , \quad \beta = 133/3 \quad \text{and} \quad \gamma = 3.5 \quad .$$

The coefficients of the Eq.s [2.4] , [2.5] are the following ones :

Table 1. The coeficients of the base function

b_j (cm^3 / g)	j	B_j (cm^3 / g)
0.7478629	1	1.1278334
-0.3540782	2	0.0
0.0	3	-0.5944001
0.0	4	-5.010996
0.007159876	5	0.0
0.0	6	0.63684256
-0.003528426	7	0.0

Note: The descriptions given in /10/ for the 2nd virial and for the excluded volume are not in accordance with those, used in the routine BB. I describe here the properties in accordance with the routine BB.

2.3 The ideal gas function

has the following form :

$$A_{idealgas}(T) = -R_{gas} \cdot T \left[1 + \left(\frac{C_1}{T_R} + C_2 \right) \ln(T_R) + \sum_{i=3}^{18} C_i \cdot T_R^{\,i-6} \right] \qquad [2.8]$$

$$\text{with} \qquad T_R \equiv T / 100\,K \quad .$$

The C-s are given in the table below.

Table 2. The coeficients of the ideal gas function

j	C_j	j	C_j
1	19.730271018	10	$4.1238460633 \cdot 10^{-3}$
2	20.9662681977	11	$-2.7929052852 \cdot 10^{-4}$
3	-0.483429455355	12	$1.4481695261 \cdot 10^{-6}$
4	6.05743189245	13	$-5.6473658748 \cdot 10^{-7}$
5	22.56023885	14	$1.6200446 \cdot 10^{-8}$
6	-9.87532442	15	$-3.303822796 \cdot 10^{-10}$
7	-4.3135538513	16	$4.51916067368 \cdot 10^{-12}$
8	$4.58155781 \cdot 10^{-1}$	17	$-3.70734122708 \cdot 10^{-14}$
9	$-4.7754901883 \cdot 10^{-2}$	18	$1.37546068238 \cdot 10^{-16}$

2.4 *The residual function*

has two parts :

$$A_{resi}(\rho, T) = \sum_{i=1}^{36} \frac{g_i}{k(i)} \left(\frac{647.073}{T} \right)^{l(i)} \left[1 - e^{-\rho} \right]^{k(i)} + \qquad [2.9]$$

$$+ \sum_{i=37}^{40} g_i \delta_i^{l(i)} \cdot \exp\left(-\alpha_i \delta_i^{k(i)} - \beta_i \tau_i^2 \right)$$

The first 36 terms are needed to fit the description to the experimental data in the low temperature mixed states, the remainig 3 terms correct the function in the vicinity of the critical point.

$$\delta_i = \frac{\rho - \rho_i}{\rho_i} \quad \text{and} \quad \tau_i = \frac{T - T_i}{T_i} \qquad [2.10]$$

are reduced densities and temperatures. The coefficients used by the residual function are given in the following tables :

Table 3. The exponents i(1) - i(40) of the residual function

0	0	0	0	1	1	1	1	2	2
2	2	3	3	3	3	4	4	4	4
5	5	5	5	6	6	6	6	8	8
8	8	2	2	0	4	2	2	2	4

Table 4. The exponents k(1) - k(40) of the residual function

2	3	5	7	2	3	5	7	2	3
5	7	2	3	5	7	2	3	5	7
2	3	5	7	2	3	5	7	2	3
5	7	1	4	4	4	0	2	0	0

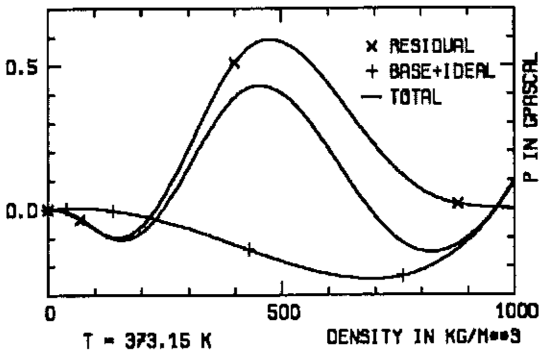
Table 5. The coefficients of the near-critical description in the residual function

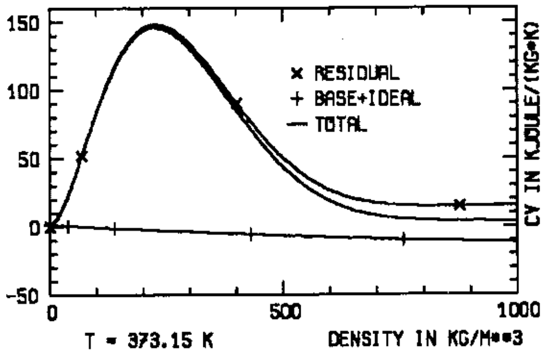
i	**37**	**38**	**39**	**40**
α_i	34.	40.	30.	1050.
β_i	20000	20000	40000	25.
ρ_i	0.319	0.319	0.319	1.550
T_i	640.	640.	641.6	270.

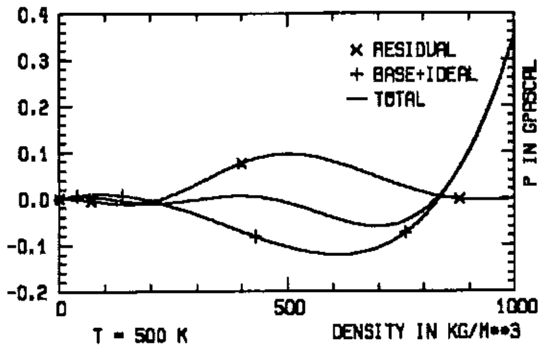
Table 6. The coefficients g(1) - g(40) of the residual function

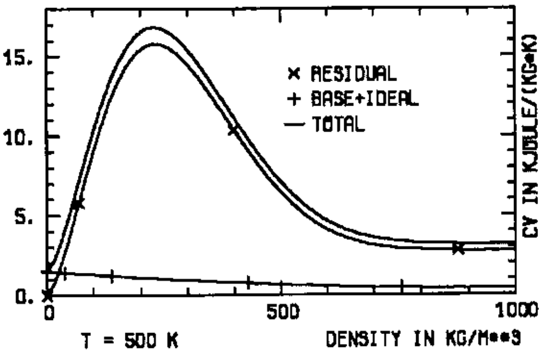
$-.53062968529023 \cdot 10^{3}$	$.22744901424408 \cdot 10^{4}$	$.78779333020687 \cdot 10^{3}$	$-.69830527374994 \cdot 10^{2}$	$.17863832875422 \cdot 10^{5}$
$-.39514731563338 \cdot 10^{5}$	$.33803884280753 \cdot 10^{5}$	$-.13855050202703 \cdot 10^{5}$	$-.25637436613260 \cdot 10^{6}$	$.48212575981415 \cdot 10^{6}$
$-.34183016969660 \cdot 10^{6}$	$.12223156417448 \cdot 10^{6}$	$.11797433655832 \cdot 10^{7}$	$-.21734810110373 \cdot 10^{7}$	$.10829952168620 \cdot 10^{7}$
$-.25441998064049 \cdot 10^{6}$	$-.31377774947767 \cdot 10^{7}$	$.52911910757704 \cdot 10^{7}$	$-.13802577177877 \cdot 10^{7}$	$-.25109914369001 \cdot 10^{6}$
$.46561826115608 \cdot 10^{7}$	$-.72752773275387 \cdot 10^{7}$	$.41774246148294 \cdot 10^{6}$	$.14016358244614 \cdot 10^{7}$	$-.31555231392127 \cdot 10^{7}$
$.47929666384584 \cdot 10^{7}$	$.40912664781209 \cdot 10^{6}$	$-.13626369388386 \cdot 10^{7}$	$.69625220862664 \cdot 10^{6}$	$-.10834900096447 \cdot 10^{7}$
$-.22722827401688 \cdot 10^{6}$	$.38365486000660 \cdot 10^{6}$	$.68833257944332 \cdot 10^{4}$	$.21757245522644 \cdot 10^{5}$	$-.26627944829770 \cdot 10^{4}$
$-.70730418082074 \cdot 10^{5}$	-.225	-1.68	0.055	-93.0

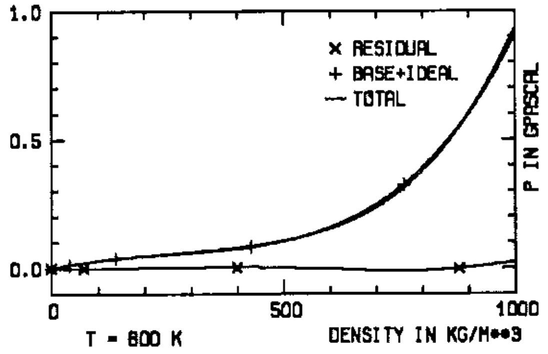
The following figures display the contributions of the basic, ideal and residual functions to the total pressure respective to the total heat capacity at the boiling point and at the temperatures 500 K and 800 K.











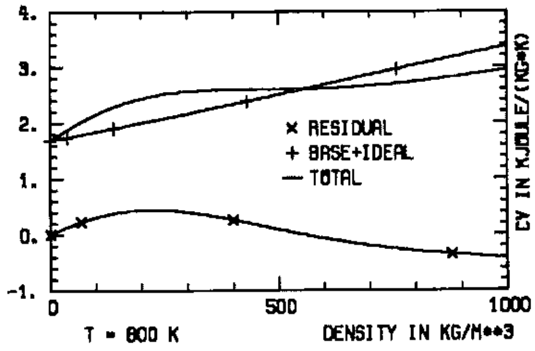


Figure 3 on page 14 displays the water pressure calculated by the H.G.K.-code as a surface of the temperature and the density.

2.5 The critical data

The H.G.K.-code uses the following values as critical data of the water :

$$T_c = 647.126\,K \quad , \quad P_c = 22.0549\,MPa \quad , \quad \rho_c = 322\,kg/m^3 \quad . \qquad [2.11]$$

2.6 The validity domain

Haar and Gallagher, /13/ give - in accordance with the IAPS - the the following validity limits for the H.G.K.-code :

$$\begin{aligned} 273\,K \leq\ &T \leq 1273\,K \\ P &\leq 1500\,MPa \end{aligned} \qquad [2.12]$$

Note: The density should be lower than 1200 kg/m^3 especially at low temperatures to avoid getting negative values for the heat capacities and the pressure derivatives $\partial P/\partial T$!

3.0 Transport properties of the water.

For the description of the viscosity, of the thermal conductivity and of the surface tension AQUA uses the IAPS-82 equations, recommended on the VIII. International Conference on the Properties of Steam /9/ (see also /10/, /11/ and /16/).

Reduced variables

In describing the transport properties of the water it is convenient to introduce the following reduced variables :

$$\begin{aligned} \tilde{r} &\equiv \frac{\rho}{\rho_k} \quad , \quad \rho_k = 317.763\, kg/m^3 \\ \tilde{t} &\equiv \frac{T}{T_k} \quad , \quad T_k = 647.27\, K \\ \tilde{p} &\equiv \frac{P}{P_k} \quad , \quad P_k = 22.115\, MPa \end{aligned} \qquad [3.1]$$

along with the "ideal temperature" :

$$T_{id} \equiv \frac{P_k}{R_{gas} \cdot \rho_k} = 150.7965\ K \quad . \qquad [3.2]$$

3.1 Viscosity

With the reduced variables the viscosity can be described as

$$\eta(\rho, T) \equiv \eta_0(T) \cdot \exp\left[\tilde{r} \cdot \sum_{j=0}^{5} \sum_{m=0}^{4} B_{jm} \cdot \left(\frac{1}{\tilde{t}} - 1 \right)^j \cdot (\tilde{r} - 1)^m \right] \qquad [3.3]$$

with the temperature-dependent amplitude :

$$\eta_0(T) \equiv \frac{10^{-6} \cdot \sqrt{\tilde{t}}}{\sum_{l=0}^{3} \frac{A_l}{\tilde{t}^l}} \quad . \qquad [3.4]$$

The coefficients of the amplitude are

`A0 = 0.0181583 , A1 = 0.0177624 , A2 = 0.0105287 , A3 = -0.0036744 .`

For the B-s in Eq. [3.3] see the table supplied.

Table 7. The coefficients B (j,m) of the viscosity					
j	m = 0	1	2	3	4
0	0.501938	0.235622	-0.274637	0.145831	-0.0270448
1	0.162888	0.789393	-0.743539	0.263129	-0.0273093
2	-0.130353	0.673665	-0.959456	0.347247	-0.0267758
3	0.907919	1.207552	-0.687343	0.213486	-0.0822904
4	-0.551119	0.0670665	-0.497089	0.100754	0.0602253
5	0.146543	-0.084337	0.195286	-0.032932	-0.0202595

3.2 Thermal conductivity

The thermal conductivity of all fluids reaches in the vicinity of the critical temperature anomalously high values (s. e. g. /4/). The following figure of Sengers - presented in /16/ - shows the thermal conductivity in the water on isobars, nearing the critical one from the high-pressure side. The data points - compared here with the corresponding IAPS calculations - are measurements of Sirota and al., /7/.

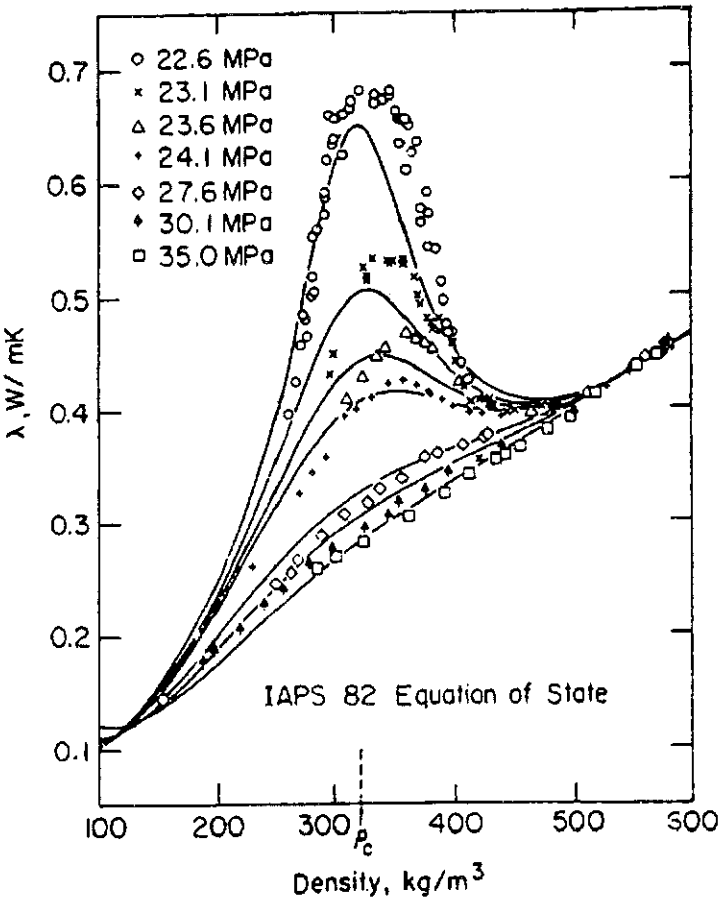


Figure 1. Thermal conductivity of the steam in the critical region as a function of density at constant pressures

Cosequently Basu and Sengers proposed a two-termed description for the thermal conductivity. The first - regular - term of this description has the same form as the viscosity, the second term describes the near critical inflation of the thermal conductivity :

$$\lambda(\rho,T) \equiv \lambda_R(\rho,T) + \Delta\lambda(\rho,T) \quad . \tag{3.5}$$

The regular term is

$$\lambda_R(\rho,T) \equiv \lambda_0(T) \cdot \exp\left[\, \tilde{r} \cdot \sum_{i=0}^{4} \sum_{n=0}^{5} b_{in} \cdot \left(\frac{1}{\tilde{t}} - 1 \right)^{i} \cdot (\tilde{r} - 1)^{n} \right] \tag{3.6}$$

with the amplitude

$$\lambda_0(T) \equiv \frac{\sqrt{\tilde{t}}}{\sum_{k=0}^{3} \frac{a_k}{\tilde{t}^{k}}} \quad . \tag{3.7}$$

The coefficients of the amplitude are

`a0 = 2.02223 , a1 = 14.11166 , a2 = 5.25597 , a3 = -2.01870` .

The b-s of the Eq. [3.6] are tabulated as follows :

Table 8. The coefficients b (i,n) of the thermal conductivity

n	i = 0	1	2	3	4
0	1.329304600	1.7018363	5.2246158	8.7127675	-1.8525999
1	-0.404524370	-2.2156845	-10.124111	-9.5000611	0.9340469
2	0.244094900	1.6511057	4.9874687	4.3786606	0.0
3	0.018660751	-0.76736002	-0.27297694	-0.91783782	0.0
4	-0.129610680	0.37283344	-0.43083393	0.0	0.0
5	0.044809953	-0.11203160	0.13333849	0.0	0.0

The nonregular term of the thermal conductivity is

$$\Delta\lambda(\rho,T) \equiv$$

$$\frac{C}{\eta(\rho,T)} \cdot \left(\frac{\tilde{t}}{\tilde{r}} \cdot \frac{\partial \tilde{p}}{\partial \tilde{t}} \right)^{2} \cdot (\chi_T)^{\omega} \cdot \sqrt{\tilde{r}} \cdot \exp\left[-A \cdot (\tilde{t} - 1)^{2} - B \cdot (\tilde{r} - 1)^{4} \right] \quad . \tag{3.8}$$

Sengers call the property

$$\chi_T \equiv \tilde{r} \cdot \frac{\partial \tilde{r}}{\partial \tilde{p}} \tag{3.9}$$

"reduced isothermal compressibility". Actually, χ_T is the factor of the nonregular term, which inflates the thermal conductivity, since

$$\frac{\partial \tilde{r}}{\partial \tilde{p}} \propto \frac{1}{\partial P / \partial \rho}$$

turns infinite at the critical point.

With the relation

$$\frac{\partial \tilde{p}}{\partial \tilde{r}} = \frac{\rho_k}{P_k} \cdot \frac{\partial P}{\partial \rho}$$

χ_T can be converted to

$$\chi_T = \frac{P_k}{\rho_k^2} \cdot \frac{\rho}{\partial P / \partial \rho} = \frac{\tilde{r}}{(T/T_{id}) \cdot P_\rho^0} \quad , \qquad [3.10]$$

and the pressure derivative in Eq. [3.8] to

$$\frac{\tilde{t}}{\tilde{r}} \cdot \frac{\partial \tilde{p}}{\partial \tilde{t}} = \frac{T}{\tilde{r} \cdot P_k} \frac{\partial P}{\partial T} = \frac{T}{T_{id}} \cdot P_T^0 \quad . \qquad [3.11]$$

The coefficients of the Eq. [3.8] are

A = 18.66 , B = 1.0 , C = 3.7711 $\cdot 10^{-8}$ Pa*sec * W/(K*m) , ω = 0.4678 .

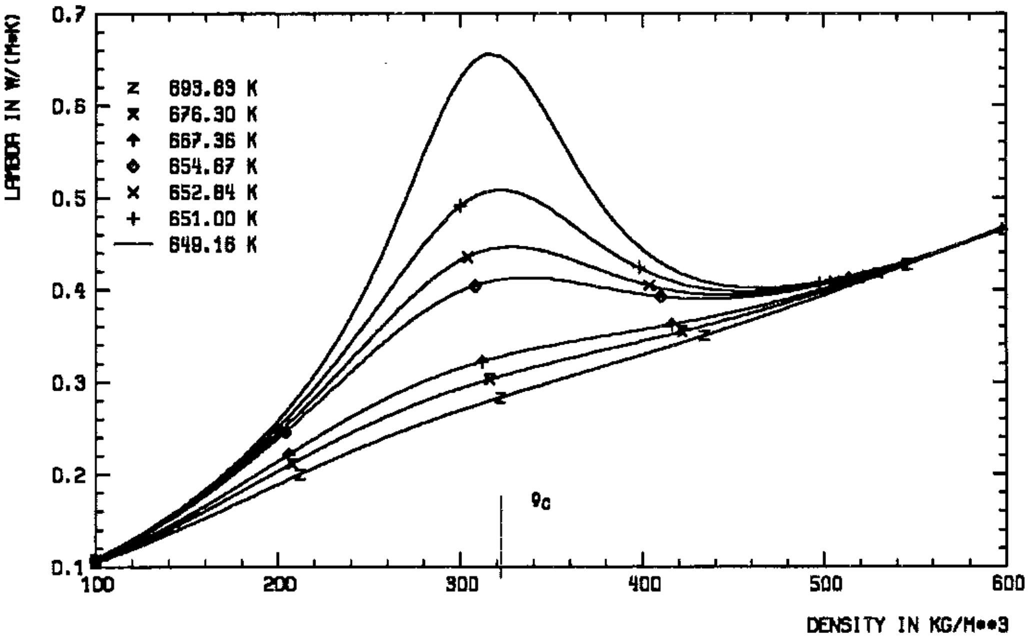


Figure 2. Thermal conductivity of the water on supercritical isotherms

Figure 2 shows the thermal conductivity on some supercritical isotherms. The isothermal pressures at the critical density are :

T / K	693.63	676.36	667.36	654.67	652.84	651.00	649.16
P322 / MPa	35.0	30.1	27.6	24.1	23.6	23.1	22.6

3.3 Surface tension

According to the IAPS Working Group III, 1975, /9/ the surface tension of the water can be described as

$$\sigma(T) = B \cdot \left(\frac{T_q - T}{T_q} \right)^{\mu} \cdot \left(1 + b \, \frac{T_q - T}{T_c} \right) \qquad [3.12]$$

with T_q = 647.15 K, B = 0.2358 *N/M*, b = -0.625 and μ = 1.256 .

Note: AQUA uses as well in Eq. [3.1] as in Eq. [3.12] as critical temperature the now usually accepted value T_c = 647.126 K. An attempt, to replace in Eq. [3.1] ρ_k = 317.763 kg/m^3 with the correct value ρ_c = 322. kg/m^3 resulted in a drastic decrease of the thermal conductivity of the water at high densities and had been given up.

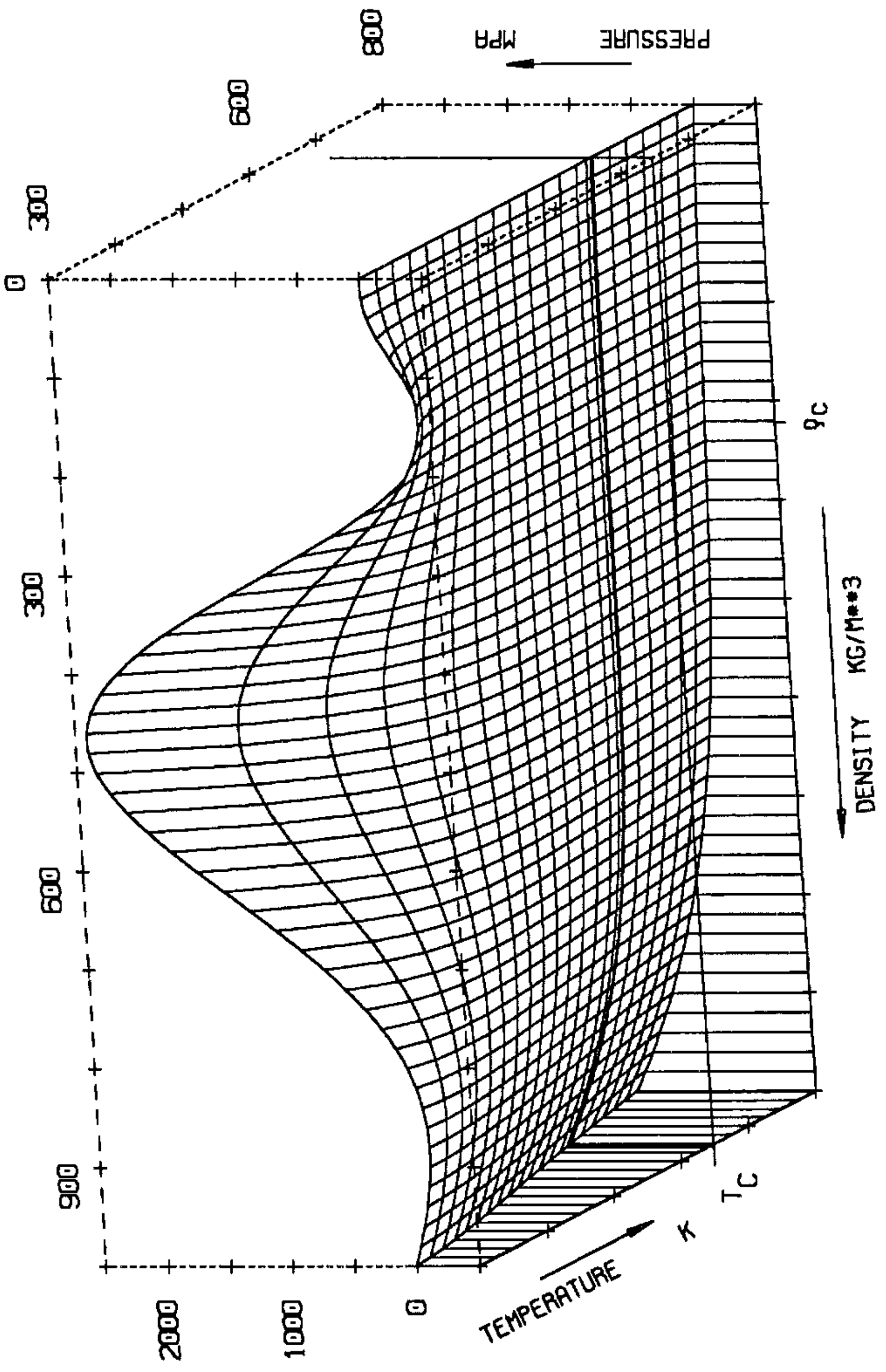


Figure 3. Pressure surface of the water. H.G.K.-code

4.0 Thermal properties in the two phase region

4.1 Thermal states of the subcritical water

At temperatures above the melting and below the critical Point water can be present not only as liquid or vapor but also as a mixture of both of them. This part of the subregion, the two phase region, or the "saturation dome" is bounded by the "saturation line ", which is a collection of the saturated liquid and the saturated vapor states. Outside the saturation dome the water is either a compressed liquid or an expanded vapor.

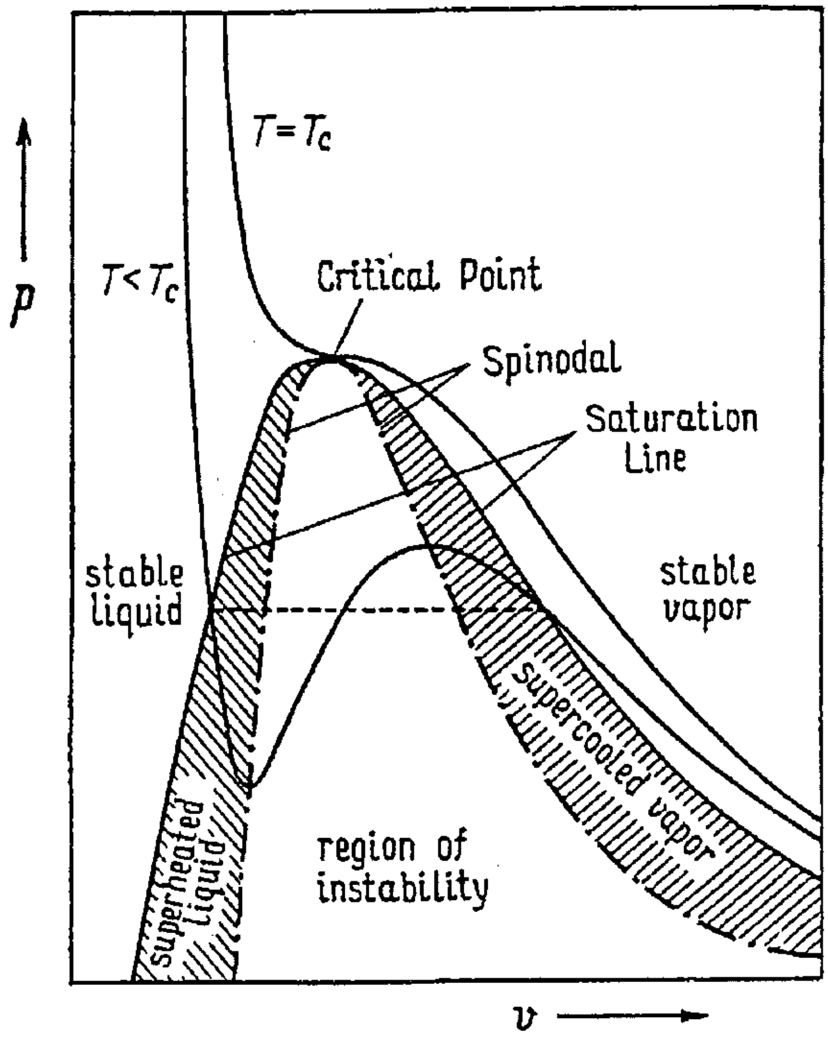


Figure 4. Sketch of P,V diagram with saturation and spinodal lines. $V = 1 / \rho$

Normally an isothermal expansion of the water beyond the saturated density ($\rho = \rho_L$) results in vaporizing the water. A part of the water - as large, as needed to restore the vapor pressure, $P^{\times}$ (dashed line on the sketch) - turns into vapor, to suffice the equilibrium conditions :

$$P^{\times}(T) = P(\rho_L, T) = P(\rho_V, T) \quad , \qquad [4.1]$$

$$G(\rho_L, T) = G(\rho_V, T) \quad . \qquad [4.2]$$

Under particular circumstances (rapid expansion, very clean water) one can expand the saturated water to a distinct degree, without turning it into vapor and reach a state with a pressure below the $P^{\times}$ (for a detailed description s e. g. /15/). This expanded water is known as superheated (liquid) water. The superheated water is metastable : it tends to turn into a mixture of saturated liquid and vapor with the equilibrium pressure $P^{\times}$. The theoretical limit of the expasion of the liquid water is given by the highest density value on the isotherm, for which it holds :

$$\frac{\partial P}{\partial \rho}(T, \rho) = 0 \qquad [4.3]$$

and is known as the (liquid) spinodal density, ρ_{SL}.

Similarly normal compression of the saturated vapor results in turning a part of it into liquid water. By carefully chosen conditions the vapor also can be compressed in some extent above the saturated vapor density without developing water droplets in it and the vapor can reach an overheated state with a pressure higher as $P^{\times}$. Again, the limiting density is the smallest density, which fulfills Eq. [4.3] and is called the (vapor) spinodal density, ρ_{SV}.

On the remaining part of the isotherm - lying between the spinodal densities (region of instability on Figure 4) - the water can occur only as a mixture of saturated liquid and vapor.

4.2 Water properties in the region of instability

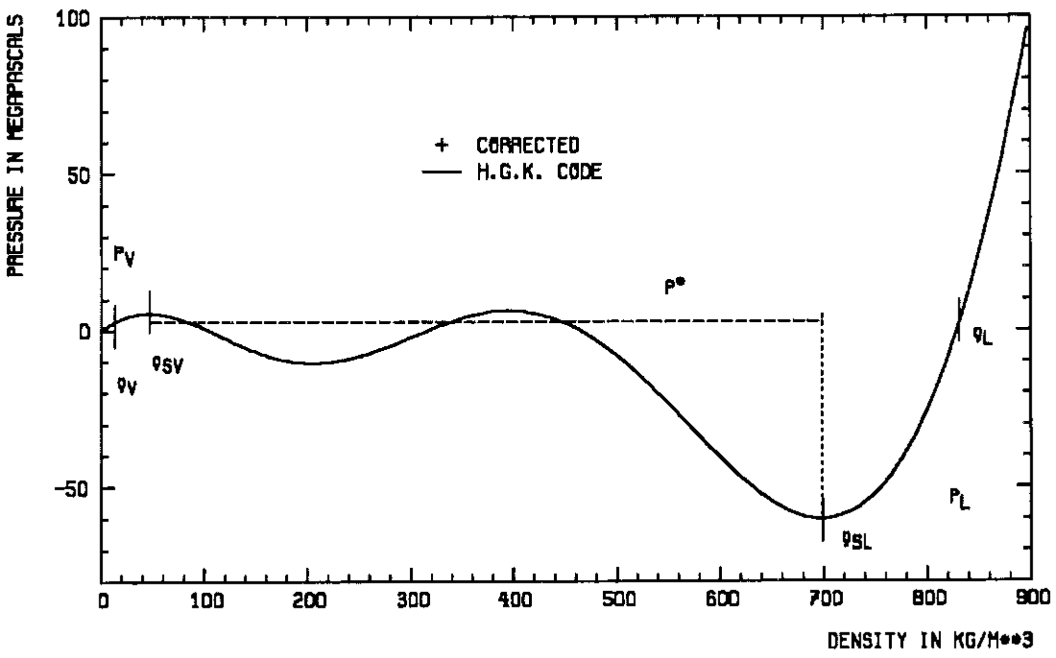


Figure 5. Pressure of the water in the two phase region. T = 500 K

Figure 5 shows the water pressure along a typical isotherm as calculated by the codes H.G.K. resp. AQUA.

The code AQUA follows the pressure description of the H.G.K.-code from zero density till the density of the vapor-spinodal and from the liquid spinodal onward. But in the instable region, between the spinodal points AQUA describes the water as a two phase mixture :

$$\begin{aligned} P(\rho, T) &= P^{x}(T) \\ \frac{\partial P}{\partial T}(\rho, T) &= \frac{dP^{x}(T)}{dT} \\ \frac{\partial P}{\partial \rho}(\rho, T) &= 0 \end{aligned} \qquad [4.4]$$

(s. dashed line on Figure 5). A 3D-diagram of the pressure of the water - as AQUA calculates it - is depicted on Figure 14 on page 24.

Note: AQUA still idealizes the progress of the water pressure along the isotherm. In reality no liquid can be streched to reach ρ_{sL} and no vapor can be compressed to have the density ρ_{sV}. The limiting values of the superheated pressure in the water can be seen on Figure 6. The measured points are from /3/.

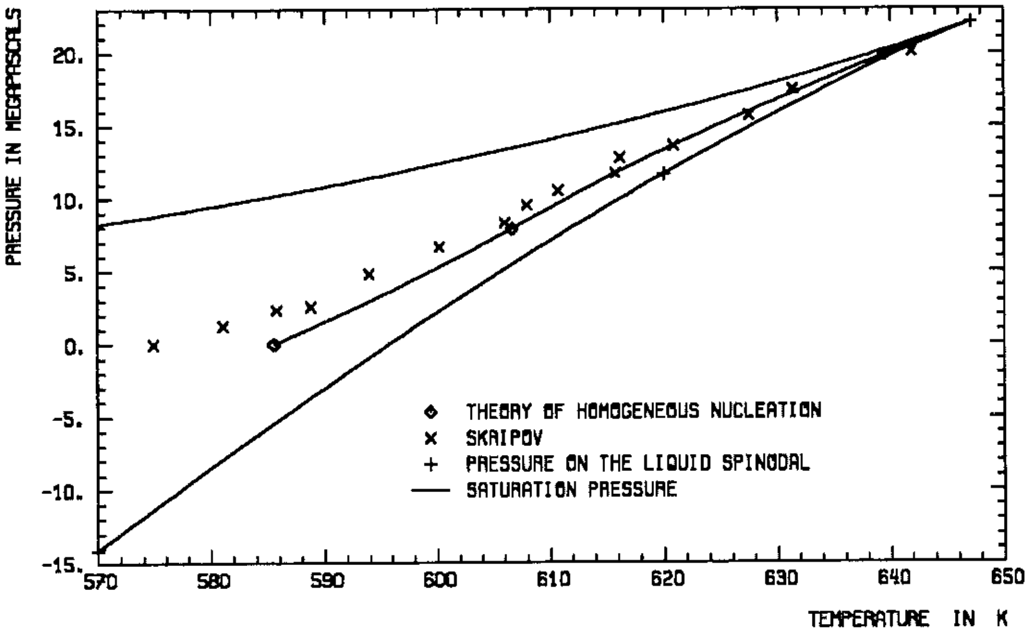


Figure 6. Water pressures in the superheated liquid state

The Figure 7 on page 18 shows the free enthalpy of the water along the same isotherm. Here again outside of the region of instability AQUA and H.G.K. calculate identical values, between the spinodal points AQUA sets G equal to the common free enthalpy value :

$$G(\rho, T) = G_L(T) = G_V(T) \ . \qquad [4.5]$$

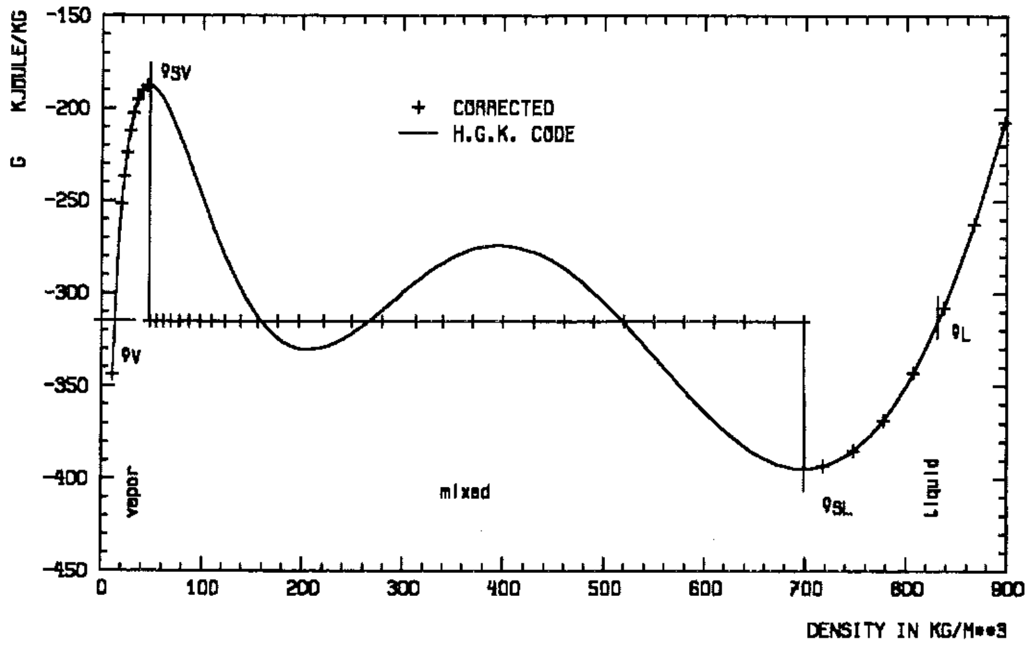


Figure 7. Free enthalpy of the water in the two phase region. T = 500 K

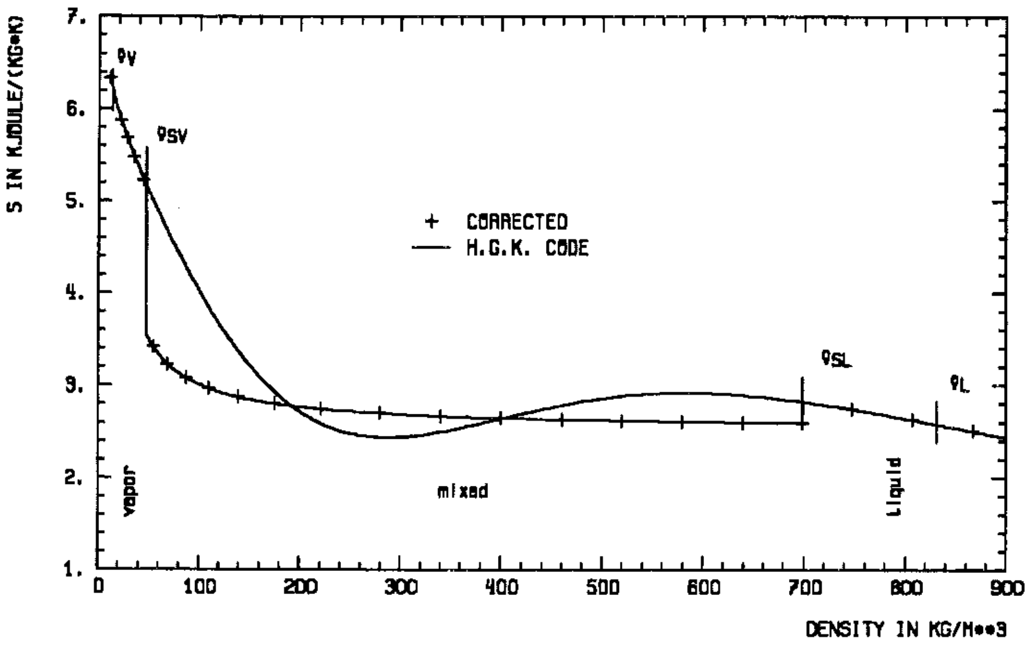


Figure 8. Entropy of the water in the two phase region. T = 500 K

As regards to the entropy, one can calculate this property in the mixed state using the density derivative [A.22]

$$\frac{\partial S}{\partial \rho} = -\frac{1}{\rho^2}\frac{\partial P}{\partial T} = -\frac{1}{\rho^2}\frac{dP^x}{dT} \qquad [4.6]$$

which gives in the region of instability the following density dependence :

$$S(\rho, T) = S_L(T) + [\,V - V_L(T)\,]\cdot\frac{dP^{\times}(T)}{dT} \quad . \tag{4.7}$$

The free energy in the region of instability (s. Figure 9) AQUA calculates from the pressure and free enthalpy via

$$A = G - P/\rho \quad . \tag{4.8}$$

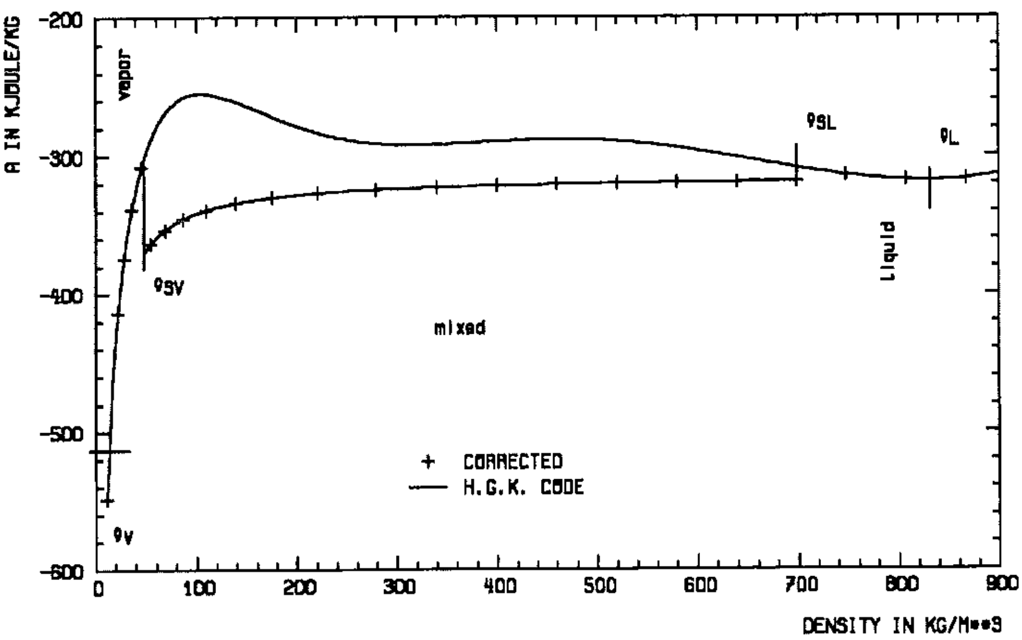


Figure 9. Free energy of the water in the two phase region. T = 500 K

For the enthalpy and the energy AQUA uses the equations

$$H = G + S\cdot T \quad , \tag{4.9}$$

respectively

$$U = H - P/\rho \quad . \tag{4.10}$$

The density derivative of the energy is given by

$$\frac{\partial U}{\partial \rho} = \frac{1}{\rho^2}\left(P - T\cdot\frac{\partial P}{\partial T}\right) = \frac{1}{\rho^2}\left(P^{\times} - T\cdot\frac{dP^{\times}}{dT}\right) \quad . \tag{4.11}$$

In the two phase region the derivatives $\partial H/\partial P$ and C_P are not defined altogether.

For the sonic velocity in the region of instability the first version of the Eq. [A.23] can be used :

$$\left.\frac{\partial P}{\partial \rho}\right|_S = \frac{\partial P}{\partial \rho} + \frac{T}{C_V}\left(\frac{\partial P/\partial T}{\rho}\right)^2 = \frac{T}{C_V}\left(\frac{1}{\rho}\,\frac{dP^{\times}}{dT}\right)^2 \tag{4.12}$$

with

$$v_s = \sqrt{\frac{\partial P}{\partial \rho}\Big|_S} = \frac{1}{\rho}\,\frac{dP^{\times}}{dT}\,\sqrt{\frac{T}{C_V}} \quad . \tag{4.13}$$

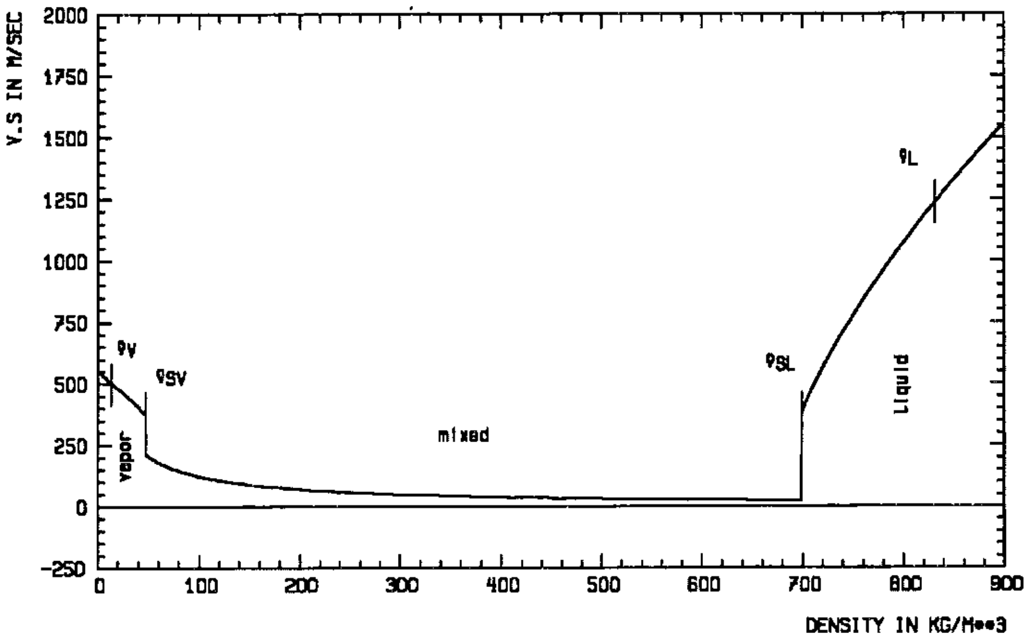


Figure 10. Sonic velocity of the water in the two phase region. T = 500 K

Not so easy is to calculate the constant volume heat capacity in the region of instability. The density derivative in the two phase region

$$\frac{\partial C_V}{\partial \rho} = \frac{\partial (T \cdot \partial S/\partial T)}{\partial \rho} = -\frac{T}{\rho^2}\,\frac{\partial^2 P}{\partial T^2} = -\frac{T}{\rho^2}\,\frac{d^2 P^{\times}}{dT^2} \tag{4.14}$$

suplies here the difference term :

$$\Delta C_V(\rho, \rho_L) = T\,\frac{d^2 P^{\times}}{dT^2}\,(V - V_L) \quad . \tag{4.15}$$

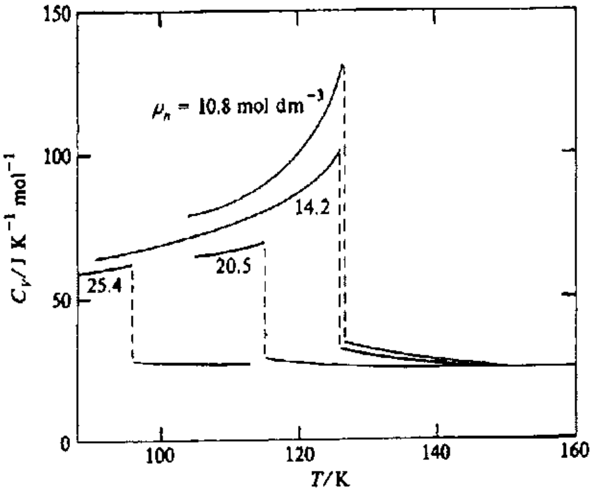
Besides of the difference term, one has also to account for the C_V - jumps, at the places, where the isotherm crosses the saturation line resp. where the liquid or the vapor turns into a vapor-liquid mixture (s. /1/, /2/, /5/, /8/). Since $\partial P/\partial T$ shows here a discontinuity, so does also C_V. At the saturated liquid this discontinuity amounts to

$$C_V(\rho_L - 0) - C_V(\rho_L) = \frac{r_L}{\rho_L}\left[\frac{dP^{\times}}{dT} - \frac{\partial P}{\partial T}(\rho_L)\right] \quad . \tag{4.16}$$

r_L in the above eq. is the logarithmic derivative of the saturation line :

$$r_L \equiv \frac{T}{\rho_L} \cdot \frac{d\rho_L}{dT} \quad . \tag{4.17}$$

The following figure displays such C_V-jumps, measured in nitrogen along different isochores.



One- and two phase C_V results for nitrogen at several amount-of-substance densities.

after L. A. Weber, /6/.

The Eqs. [4.15], [4.16] together give the following heat capacity function in the region of instability :

$$C_V(\rho) = C_V(\rho_L) + \frac{r_L}{\rho_L}\left[\frac{dP^x}{dT} - \frac{\partial P}{\partial T}(\rho_L)\right] + \left(\frac{1}{\rho} - \frac{1}{\rho_L}\right) T \,\frac{d^2P^x}{dT^2} \quad . \qquad [4.18]$$

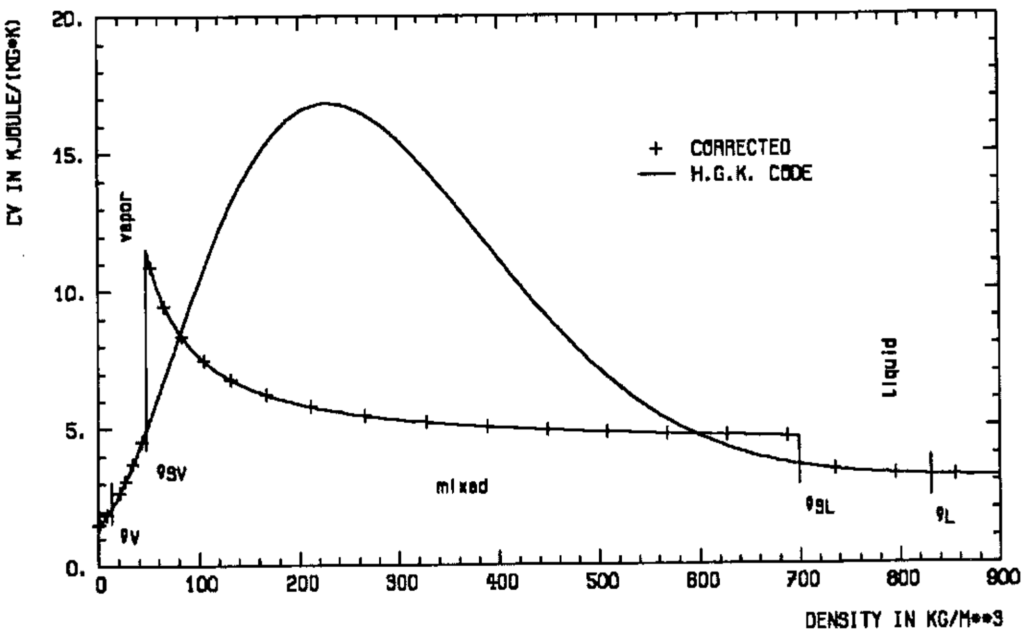


Figure 11. Heat capacity - C_V - of the water in the two phase region. T = 500 K

For a (T , ρ) - surface of the heat capacity of the water - calculated by AQUA - see Figure 15 on page 25 resp. Figure 16 on page 26.

In calculating the thermal conductivity in the two phase region AQUA uses in the different subregions different algorithms. In the superheated liquid and subcooled vapor the function - described in 3.0, "Transport properties of the water." , Eq. [3.5] ff. - is used as well as outside the saturation dome.

In the region of instability AQUA calculates the thermal conductivity using the following approximation : the liquid part of the water in this subregion is distributed homogeneously - as a multitude of tiny droplets - in the vapor part of the water. This mixture is furthermore static in its behaviour, so the convective part of the heat transfer can be neglected. The approximation above allows to calculate in the region of instability the thermal conductivity from the thermal conductivities of the saturated states - λ_L , λ_V - as follows :

$$\frac{1}{\lambda} = \frac{1-x}{\lambda_L} + \frac{x}{\lambda_V + x^2\cdot(\lambda_L - \lambda_V)} \tag*{[4.19]}$$

(s. /5/, Eq. 71). x in the eq. above is

$$x = \sqrt[3]{\frac{\rho - \rho_V}{\rho_L - \rho_V}} \quad . \tag*{[4.20]}$$

Figure 12 shows the thermal conductivity of the water along the 500 K - isotherm. The spikes in the vicinity of the spinodal points are a consequence of the description [3.8], using $\partial P/\partial\rho$ - which vanishes at the spinodals - in the denominator.

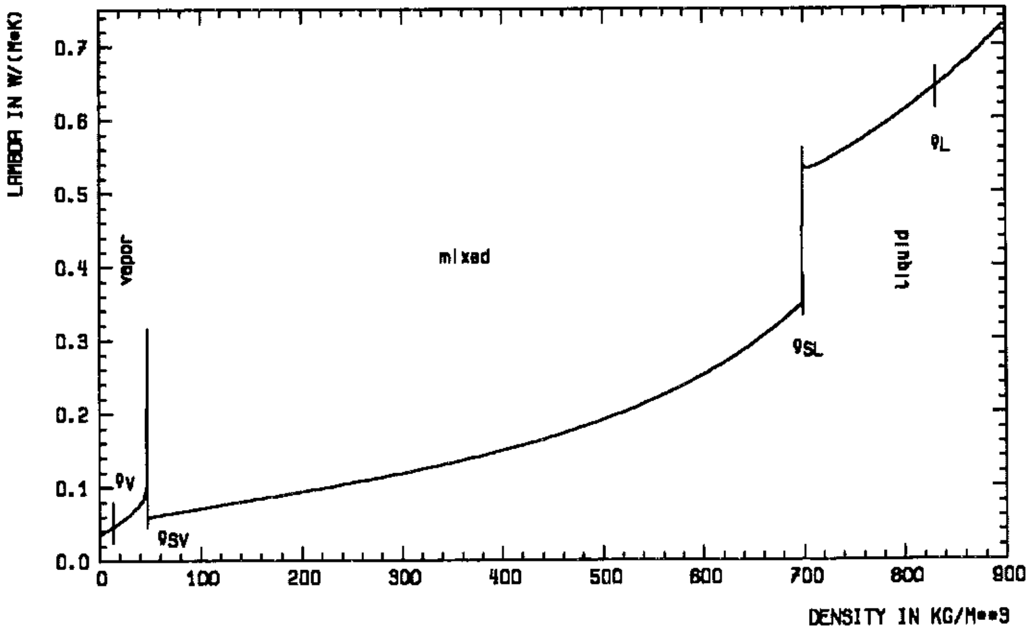


Figure 12. Thermal conductivity of the water in the two phase region. T = 500 K

As well as the thermal conductivity, the viscosity is also a transport process, the transport of the monentum. So AQUA calculates the viscosity by a procedure, very similar to the one, used for the calculation of the thermal conductivity.

At the saturation densities the η function Eq. [3.3]- [3.4] - will be continued until the spinodals are reached. In the region of instability AQUA uses the same fluid-model, as in the case of the thermal conductivity to calculate the viscosity from the saturated viscosities η_L , η_V. The resulting equation is :

$$\frac{1}{\eta} = \frac{1-x}{\eta_L} + \frac{x}{\eta_V + x^2 \cdot (\eta_L - \eta_V)} \quad . \qquad [4.21]$$

Figure 13 shows the water-viscosity as calculated by AQUA along the 500 K - isotherm.

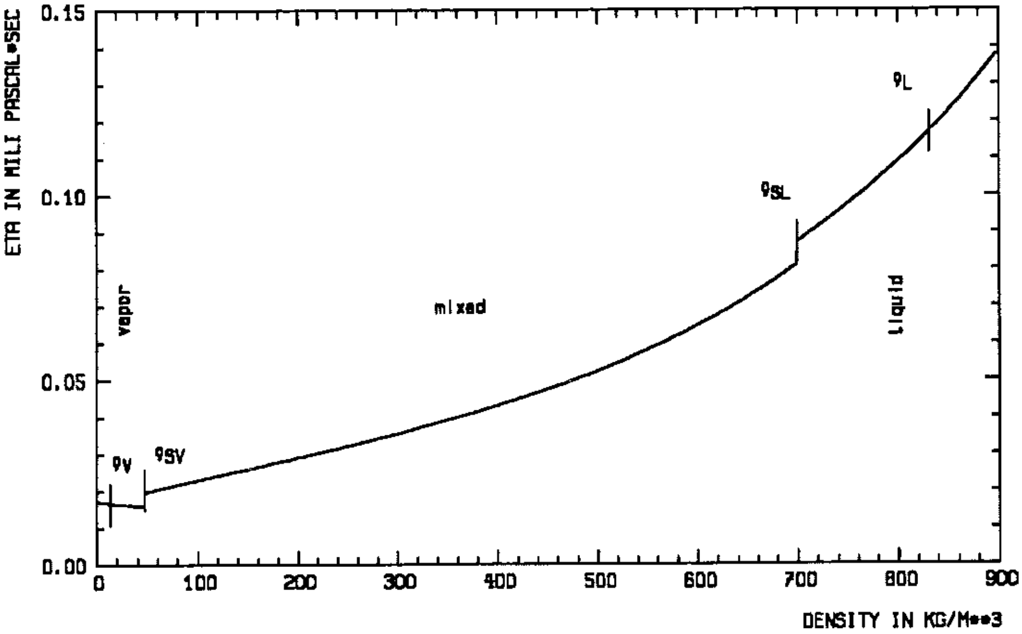


Figure 13. Viscosity of the water in the two phase region. T = 500 K

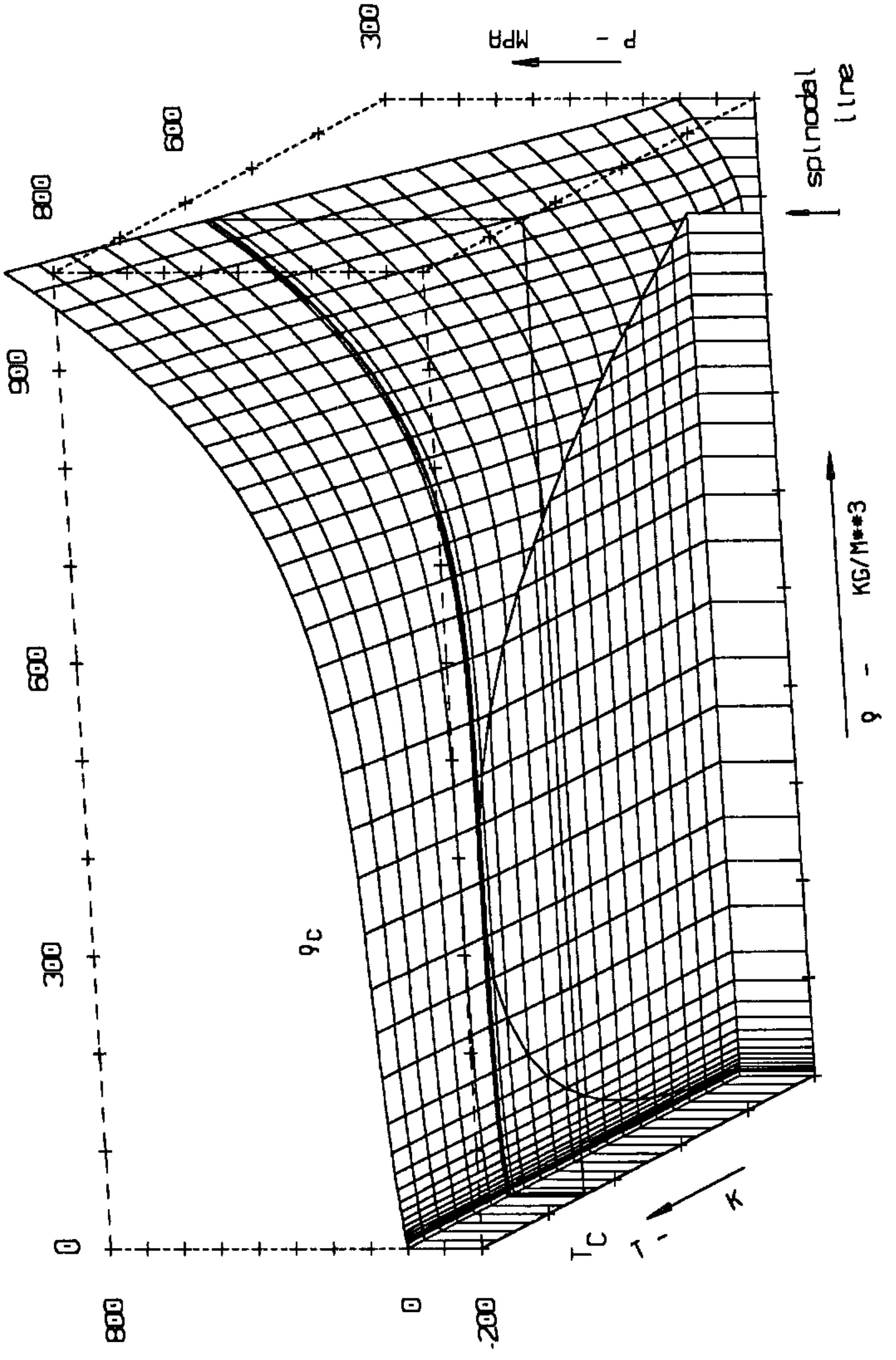


Figure 14. Pressure surface of the water. AQUA

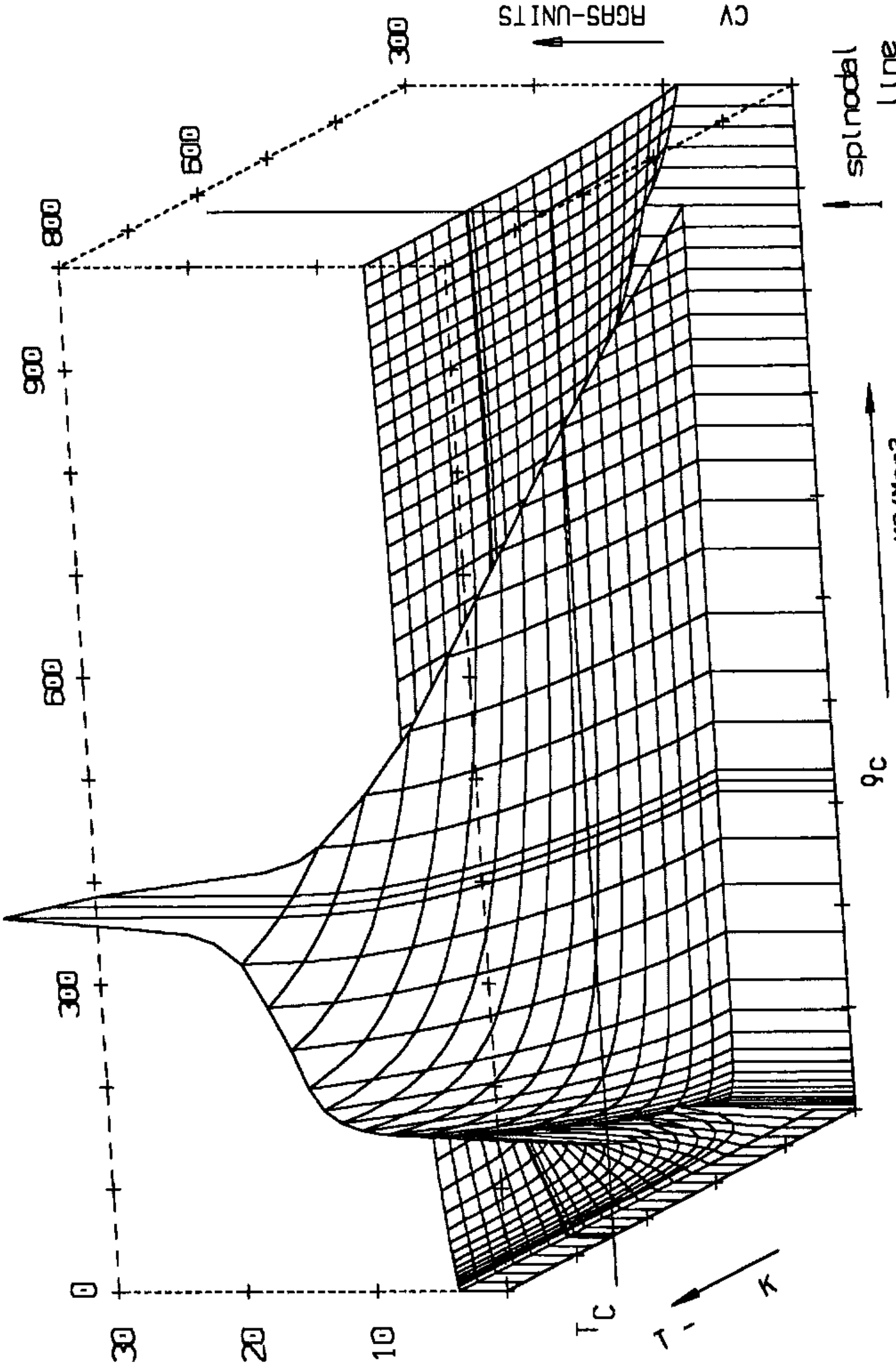


Figure 15. Heat capacity surface of the water. AQUA. View from the cold side.

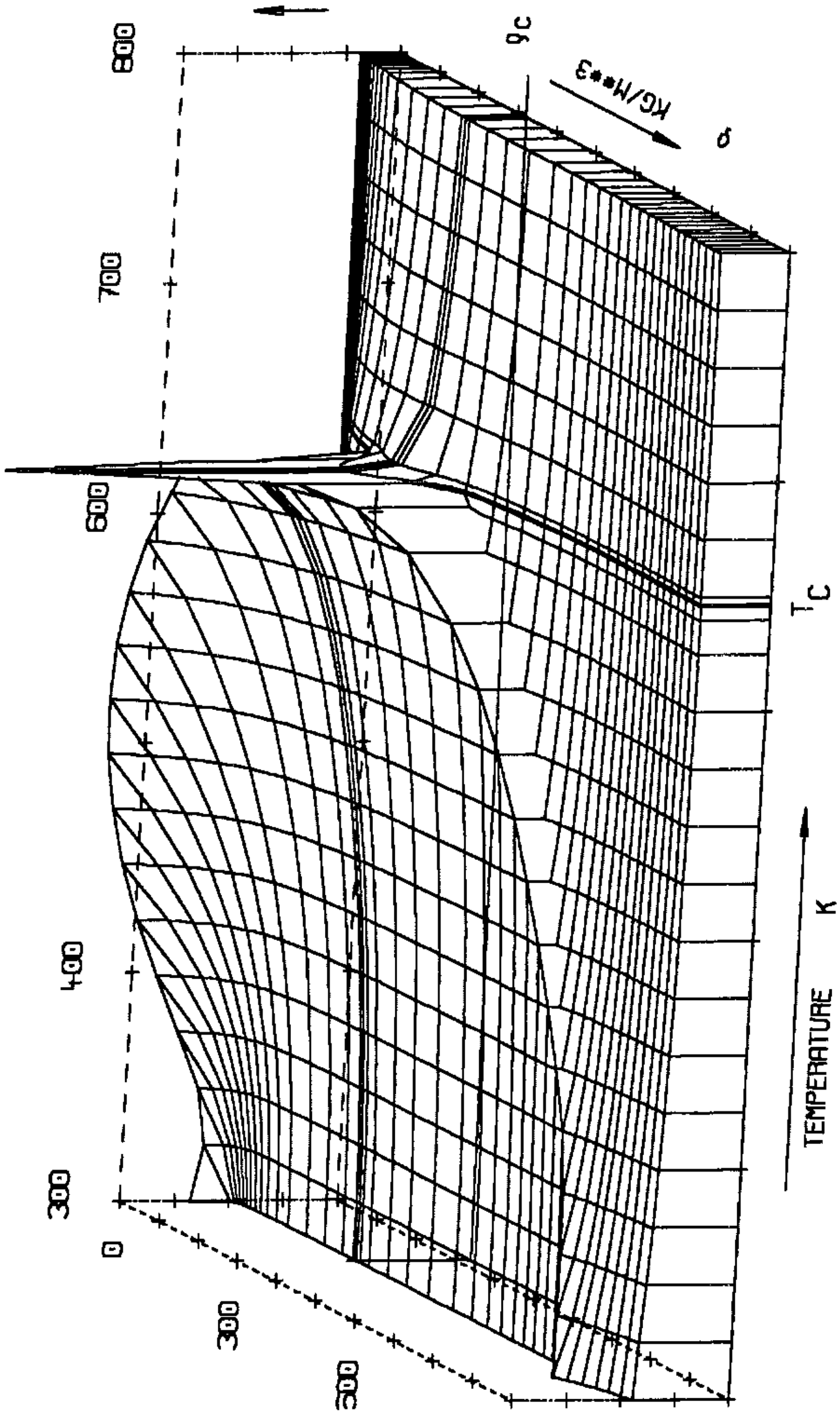


Figure 16. Heat capacity surface of the water. AQUA. View from the liquid side.

5.0 Describing the water properties in the saturated states

In order to be able to calculate the water properties in the region of instability AQUA needs a series of thermal properties of the water in the saturated states (s. e.g. Eq. [4.18]). To speed up the calculations AQUA doesn't calculate these properties (P^x, ρ, $\partial P/\partial \rho$, $\partial P/\partial T$, C_V, λ, etc) directly, but use pre-defined temperature-functions of the corresponding properties. In constructing these temperature-functions great care was taken

to have negligible deviations to the H.G.K values and

to show the correct behaviour in the vicinity of the critical point.

The remaining part of this chapter describes the temperature-descriptions of these properties of the water in the saturated states.

5.1 *Describing the saturation line of the water*

The saturation pressure

Haar, Gallagher and Kell describe the vapor pressure of the water (at temperatures, exceeding 314 K) as a function of the scaled temperature

$$\tau \equiv 1 - \frac{T}{T_c} \qquad [5.1]$$

rsp. of the square root of it

$$w \equiv \sqrt{\tau} \qquad [5.2]$$

as follows :

$$\ln P^x = \ln P_c + \frac{\tau}{1-\tau} \sum_{j=1}^{8} A_j \bullet w^{j-1} \ . \qquad [5.3]$$

In the above Eq. H.G.K. use critical data

$$T_c = 647.25\,K \quad , \quad P_c = 22.093\,MPa \qquad [5.4]$$

which differ markedly from their regulary used values Eq. [2.11] .

The polynomial coefficients, A_j are :

Table 9. The coefficients A(1) - A(8) of the HGK-vapor pressure equation

1	-7.8889166	2	2.5514255	3	-6.716169
4	33.239495	5	-105.38479	6	174.35319
7	-148.39348	8	48.631602		

AQUA uses, instead of the Eq. [5.3] a description, proposed by Wagner and Saul at the conference in Moscow, /14/. The W&S-equation needs only 6 coefficients and yet represent the measured vapor pressure data within their experimental uncertainity (see Figure 17).

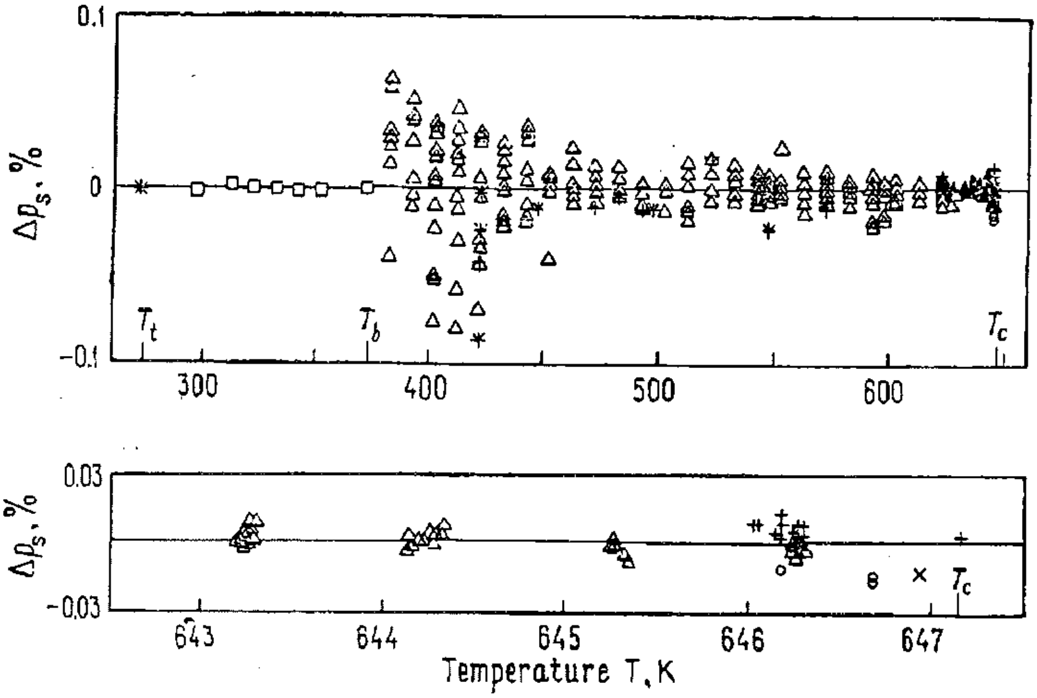


Fig. 1. Relative deviations $\Delta p_s = (p_{s,exp} - p_{s,calc})/p_{s,exp}$ of the experimental vapour pressures from values calculated with Eq. (5). △ Osborne et al. [14], + Rivkin et al. [16], □ Stimson [13], ✻ Guildner et al. [12], ○ Hanafusa et al. [17], × Kawai et al. [18], ⊻ Kell et al. [19]

Figure 17. Vapor pressure formula of Wagner and Saul

The W&S-equation has the same form as the H.G.K. one, Eq. [5.3] ,

$$\ln P^{x} = \ln P_c + \frac{\tau}{1-\tau} \sum_{j=1}^{13} N_j \cdot w^{j-1} \qquad [5.5]$$

only the critical data and the coefficients differ. They use

$$T_c = 647.14\,K \quad , \quad P_c = 22.064\,MPa \qquad [5.6]$$

as critical data and

$$\begin{array}{llllll} N_1 = & -7.85823 \quad , & N_2 = & 1.83991 \quad , & N_5 = & -11.7811 \quad , \\ N_6 = & 22.6705 \quad , & N_7 = & -15.9393 \quad , & N_{13} = & 1.77516 \quad , \end{array} \qquad [5.7]$$

as polynomial-coefficients. In calculating the vapor pressure AQUA uses here also the critical data of Haar, Gallagher and Kell (Eq. [2.11]) instead of the data [5.6] .

As well as in the H.G.K.-code, P^{x} of Eq. [5.5] is for AQUA only an approximated pressure. The true vapor pressure is calculated from this first value by correcting the saturated densities till the Eq.s [4.1 - 4.2] are satisfied.

The saturation line

The saturated densities are needed not only as limiting borders, but also as properties of their own value. The caclulation of various thermal properties in the region of instability requires mostly the saturated densities and in some cases their temperature derivatives too (s. e.g. Eq. [4.18]). Consequently one needs not only smooth functions for the saturated densities, but functions with smooth temperature derivatives.

The H.G.K.-code calculates - below 646.3 K - P^x, ρ_L and ρ_V by searching for states (T,ρ_L) resp. (T,ρ_V) with equal free enthalpy values (Eq.s [4.1 - 4.2]). The corresponding equal pressures define then P^x(T). Above 646.3 K Haar, Gallagher, and Kell use "scaled" equations for the saturated densities :

$$\frac{\rho_L(T)}{\rho_c} = 1 + 2.04077 \cdot \tau^{\beta}$$
$$\frac{\rho_V(T)}{\rho_c} = 1 - 2.04077 \cdot \tau^{\beta} \qquad [5.8]$$

with the critical exponent $\beta = 0.325$, without correcting the densities for common free enthalpy.

AQUA describes - as well as the vapor pressure - the saturation line also with the T_c-related variables τ and w, Eq.s [5.1] , [5.2] .

Below 620 K the density of the saturated liquid is described with a polynomial of w,

$$\rho_L(T) = \rho_c \cdot \sum_{j=1}^{12} A_j \cdot w^{j-1} \quad . \qquad [5.9]$$

Table 10. The coefficients A(1) - A(12) in the Eq. of the saturated liquid

1	6.072746418	2	-149.9456089	3	2121.007381
4	-17164.83301	5	90176.84684	6	-323045.6094
7	805443.5472	8	-1398515.305	9	1658696.365
10	-1281195.416	11	580741.4278	12	-117181.9097

In the case of the saturated vapor instead of the density, the factor of reality is described by a polynomial of w :

$$\rho_V(T) = \frac{P^x(T)}{R_{gas} \cdot T \cdot Z_V(T)} \quad , \quad Z_V(T) = \sum_{j=1}^{12} Z_j \cdot w^{j-1} \quad . \qquad [5.10]$$

Table 11. The coefficients Z(1) - Z(12) in the Eq. of the saturated vapor

1	-1.125999398	2	42.15284538	3	-538.0831003
4	4150.173161	5	-20809.47751	6	71295.07069
7	-170444.6088	8	284498.9747	9	-325135.7165
10	242459.309	11	-106255.203	12	20747.6267

As well for the liquid, as for the vapor the polynomial-coefficients were fitted to properties, calculated by the H.G.K-code. The vapor pressure in Eq. [5.10] was calculated by the W&S formula.

Above 620 K AQUA uses practically the same scaled description, as the H.G.K code (Eq. [5.8]) only fitting tails had been added :

$$\begin{aligned} \rho_L(T) &= \rho_c \bullet \left[1 + 2.03 \bullet \tau^{\beta} + \tau \left(U_1 + \tau \left(U_2 + \tau^2 U_4 \right) \right) \right] \quad , \\ \rho_V(T) &= \rho_c \bullet \left[1 - 2.03 \bullet \tau^{\beta} + \tau \left(W_1 + \tau \left(W_2 + \tau^2 W_4 \right) \right) \right] \quad . \end{aligned} \qquad [5.11]$$

The coefficients of the fitting tails

$$U_1 = 3.075668853 \quad , \quad U_2 = -19.58060683 \quad , \quad U_4 = 1383.086689$$
$$W_1 = 1.451962241 \quad , \quad W_2 = -5.041325559 \quad , \quad W_4 = 611.1810749$$

had ben calculated by demanding a double-smooth connection

$$f_-(T_s) = f_+(T_s) \quad , \quad f'_-(T_s) = f'_+(T_s) \quad , \quad f''_-(T_s) = f''_+(T_s) \qquad [5.12]$$

for them to the low-T formulas at $T_s = 620\,K$. (see Appendix B, "Derivatives of some saturated properties" on page 71).

The saturated densities, calculated by the Eq.s [5.9 - 5.11] are again only approximated ones. AQUA corrects them - below 646.0 K - to fulfil the Eq.s [4.1 - 4.2] (s. Appendix C, "Adjusting the saturated states" on page 75).

5.2 *Describing the spinodals*

Calculating the spinodals of the water.

The spinodals of the water are essential for the code AQUA : the spinodal densities are needed as domain borders for the calculations in the saturation dome, the spinodal pressures are neded as pressure bounds in converting the pressures to densities.

In order to calculate the densities of the spinodals one has to find the nearest roots of the Eq. [4.3] on the isotherms in the vicinity of the saturated densities. In the code-package AQUA there is a program HGKSPN wich calculates - using the routines QUADO, MONIKA and SORBET - the densities and pressures of the spinodals of the water.

HGKSPN calculates $\partial P / \partial \rho$ at different densities on an isotherm using the H.G.K.-code.

QUADO selects 3 adjacent densities nearest to the root of Eq. [4.3] and fits a 2. grade polynomial - Q_2 - to them.

MONIKA is needed only to ensure, that the densities are properly ordered. QUADO solves the resulting quadratic equation

$$Q_2(\rho) = 0$$

and selects the appropriate root as the first approximation for the spinodal density, ρ_{SL1}.

SORBET discards now one of the first 3 density values - the density, with the worst fitting - and replaces it with ρ_{SL1}. These 3 densities are then returned to QUADO and a new root for $\partial P / \partial \rho$, ρ_{SL2} will be calculated. The procedure endures, till Eq. [4.3] is fulfilled with sufficient accuracy.

The spinodal densities

Figure 18. shows the near-critical part of the calculated spinodal densities. As one can see, above the temperature of 635 K the spinodal lines get more and more irregular in shape. Above T = 646.3 K the H.G.K. code prohibites altogether the calculation of the spinodals.

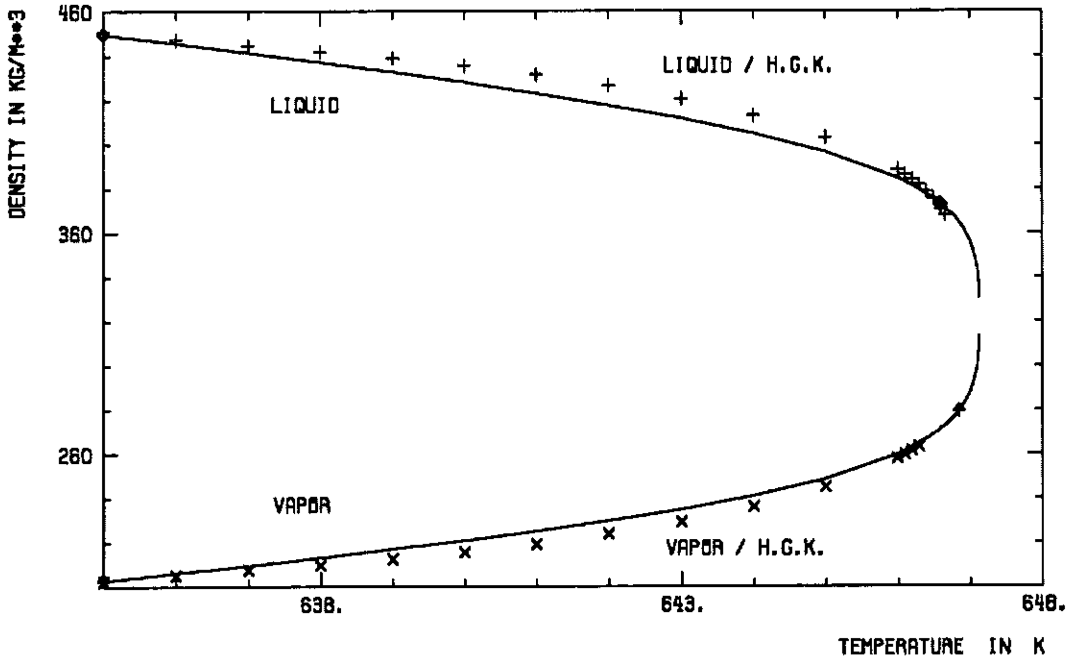


Figure 18. Spinodal densities in the vicinity of the critical point

AQUA describes the spinodal densities - in the same way as the saturated densities - as functions of w resp. τ (s. Eq. [5.1 - 5.2]) .

Below 620 K the density as well of the liquid, as of the vapor spinodal is approximated with polynomials

$$\rho_{SL}(T) = \rho_c \cdot \sum_{j=1}^{9} A_j \cdot w^{j-1} \quad , \qquad \rho_{SV}(T) = \rho_c \cdot \sum_{j=1}^{7} C_j \cdot w^{j-1} \quad . \qquad [5.13]$$

Table 12. Polynomial coefficients A(1) - A(9) of the density of the liquid spinodal

1	1.911888766	2	-16.54157899	3	179.7358194
4	-940.1374463	5	2957.964417	6	-5752.429056
7	6772.987553	8	-4428.001759	9	1234.441322

Table 13. Polynomial coefficients C(1) - C(7) of the density of the vapor spinodal

1	0.9201508683	2	-2.159701944	3	-0.2932760492
4	6.324940268	5	-10.49597891	6	8.78261245
7	-3.119252324				

The coefficients in Eq. [5.13] were fitted to the liquid resp. vapor densities calculated by HGKSPN.

Above 620 K the spinodal densities of the H.G.K.-isotherms are not reliable to be used as a fitting source. Instead of using them, a [5.11] - type near-critical extension

$$\begin{aligned} \rho_{SL}(T) &= \rho_c \cdot \left[1 + 1.40 \cdot \tau^{\beta} + \tau \left(U_1 + \tau \left(U_2 + \tau^2 U_4 \right) \right) \right] \quad , \\ \rho_{SV}(T) &= \rho_c \cdot \left[1 - 1.40 \cdot \tau^{\beta} + \tau \left(W_1 + \tau \left(W_2 + \tau^2 W_4 \right) \right) \right] \end{aligned} \qquad [5.14]$$

was constructed.

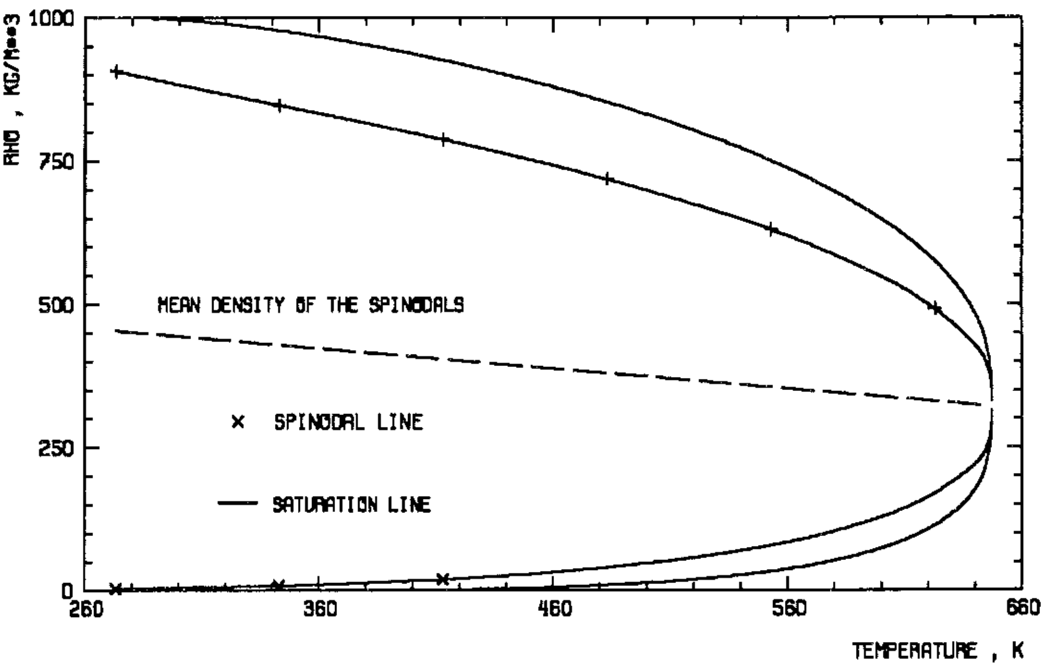


Figure 19. Saturation and spinodal densities of the water

ρ_c and β in Eq. [5.14] ensure, that for all near-critical isotherms the density-relation

$$\rho_V < \rho_{SV} < \rho_c < \rho_{SL} < \rho_L$$

holds. The critical amplitude - 1.4 in Eq. [5.14] - was found by comparing the results of Eq. [5.14] with the H.G.K.-data at temperatures above 640 K. The coefficients of the fitting tails

$$U_1 = 0.4871717941 \quad , \quad U_2 = 24.26899415 \quad , \quad U_4 = -2461.494028$$

$$W_1 = 1.132819731 \quad , \quad W_2 = -30.70131059 \quad , \quad W_4 = 3324.944936$$

are calculated again - as in the case of the saturated densities - by demanding double smooth connections at 620 K.

Figure 19 displays the saturated and the spinodal densities as functions of the temperature. It is interesting to note that the mean spinodal

$$\rho_{SM} = (\rho_{SL} + \rho_{SV})/2$$

is a linear function of the temperature.

The spinodal pressures

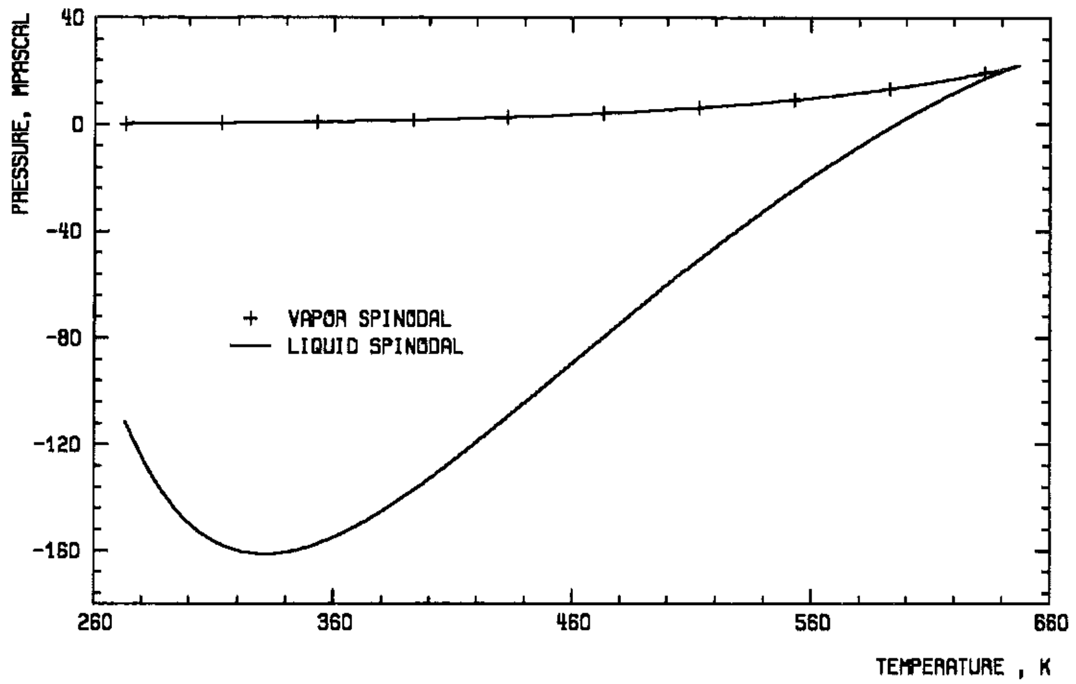


Figure 20. Pressures on the spinodals of the water

In the vicinity of the critical point the spinodal pressures - like the density - are not accessible via the H.G.K.-code. Moreover, at every subcritical T the pressure relation

$$P_{SL}(T) < P^{x}(T) < P_{SV}(T)$$

must hold. To cope with these conditions AQUA approximatates below 620 K the spinodal pressures - calculated by HGKSPN - with τ- polynomials and describes - near T_c - their deviations from P^x.

Below 620 K the pressures are described as :

$$P_{SL}(T) = \sum_{j=1}^{9} A_j \cdot \tau^{j-1} \quad , \quad P_{SV}(T) = \sum_{j=1}^{8} C_j \cdot \tau^{j-1} \quad . \qquad [5.16]$$

Table 14. Polynomial coefficients A(1) - A(9) of the pressure of the liquid spinodal

1	-0.001387361801	2	0.1510341901	3	1.980521347
4	2.921317875	5	-38.3240633	6	162.2329521
7	-384.0027285	8	479.3961544	9	-264.1248035

Table 15. Polynomial coefficients C(1) - C(8) of the pressure of the vapor spinodal					
1	-0.0007497146775	2	0.162845372	3	-0.3190428782
4	-1.582269493	5	6.787420254	6	-10.41333815
7	7.599390008	8	-2.285884902		

Above 620 K the descriptions are :

$$P_{SL}(T) = P^{x}(T) - 4.98 \cdot 10^{8} \cdot \sum_{j=1}^{3} U_j \cdot \tau^j \quad ,$$

$$P_{SV}(T) = P^{x}(T) + 2.00 \cdot 10^{8} \cdot \sum_{j=1}^{3} W_j \cdot \tau^j \quad . \qquad [5.17]$$

The polynomials in Eq. [5.17] vanish at the critical point as τ. Their coefficients

$U_1 = 0.05235363429$, $U_2 = 4.419437236$, $U_3 = -18.86727538$

$W_1 = 0.1103469546$, $W_2 = 0.913150777$, $W_3 = -10.87001272$

had been gained by using double smooth connections at 620 K.

5.3 *The entropy on the saturation line*

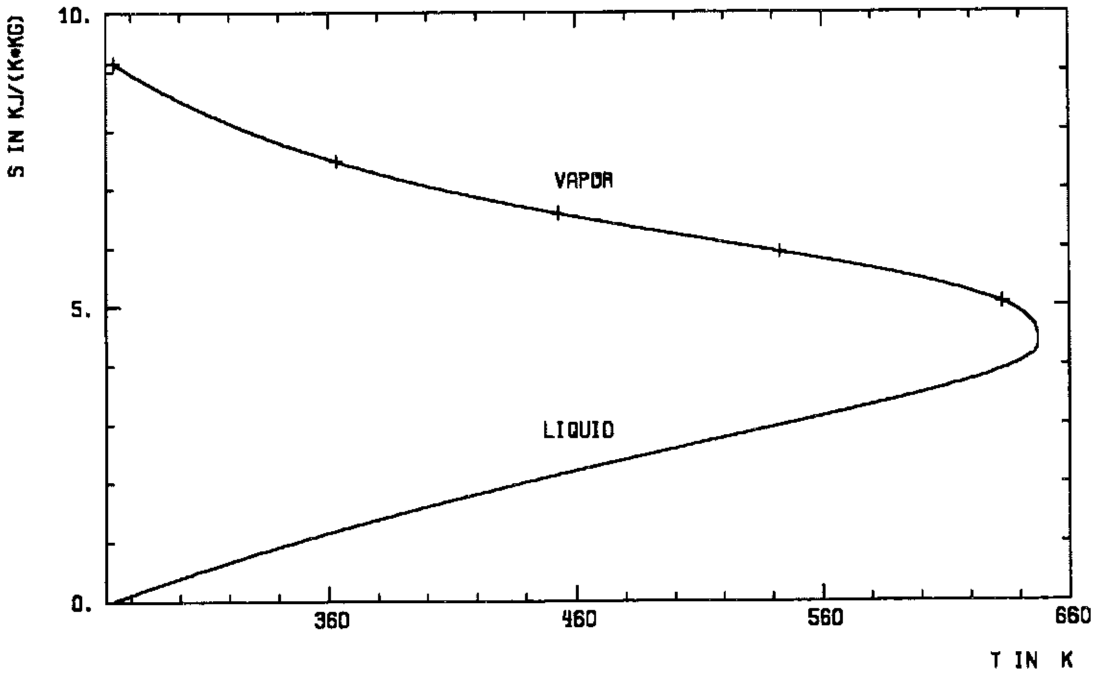


Figure 21. Entropy on the saturation line of the water

The entropy in the saturated states show a course similar to the course of the saturated densities. Consequently AQUA employs - for the description of the entropy of the saturated liquid - a formula, similar to the one used for the description of the saturated densities.

Below 620 K the entropy of the saturated liquid - calculated by the H.G.K. code - is described with a w - polynomial (s. Eq. [5.1 - 5.2])

$$S_L(T) = S_c \cdot \sum_{j=1}^{9} A_j \cdot w^{j-1} \quad . \qquad [5.18]$$

Table 16. Polynomial coefficients A(1) - A(9) of the entropy of the saturated liquid

1	0.9528621116	2	0.2725494325	3	-8.457375782
4	44.57631748	5	-150.7441559	6	311.7914429
7	-391.8528748	8	273.5529633	9	-82.2848053

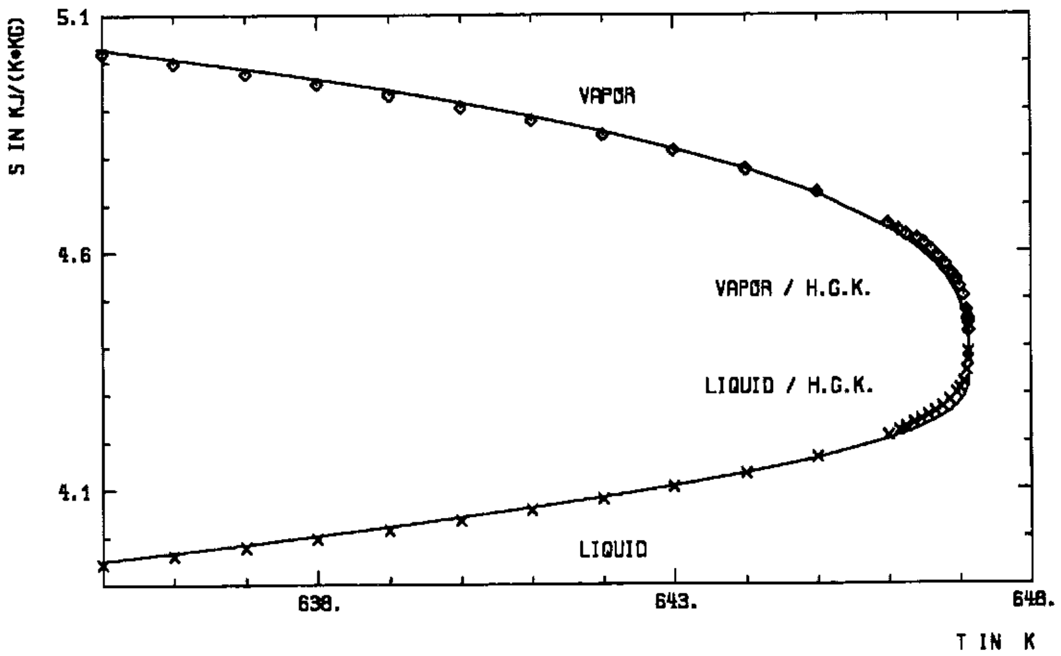


Figure 22. Entropy of the saturated states in the vicinity of T_c

Above 620 K the entropy of the liquid is described with a "vapor-like" scaled formula :

$$S_L(T) = S_c \cdot \left[1 - 0.25 \cdot \tau^{\beta} + \tau \left(U_1 + \tau \left(U_2 + \tau^2 U_4 \right) \right) \right] \quad . \qquad [5.19]$$

The critical entropy and the scaled amplitude

$$S_c = 9.43 \cdot R_{gas} \qquad \text{and} \qquad B_s = 0.25 \qquad [5.20]$$

had been determined by fitting as well Eq. [5.19], as the corresponding formula for the entropy of the vapor to the H.G.K.-values above 635 K. The coefficients

$$U_1 = -1.23383958 \quad , \quad U_2 = -1.656615363 \quad , \quad U_4 = 911.5375812$$

result from using a double smooth extension to higher temperatures at the switching point.

The entropy on the saturated vapor needs no extra description, it can be calculated from S_L by the eq. of Clausius & Clapeyron

$$S_V = S_L + \left(\frac{1}{\rho_V} - \frac{1}{\rho_L}\right) \cdot \frac{dP^x}{dT} \quad .$$

5.4 The transport properties on the saturation line

The following descriptions for the viscosity resp. thermal conductivity in the saturated states were constructed by calculatig the properties on the saturation line with the algorithms, described in 3.0, "Transport properties of the water." on page 9 and approximating these values with adequate functions.

The viscosity

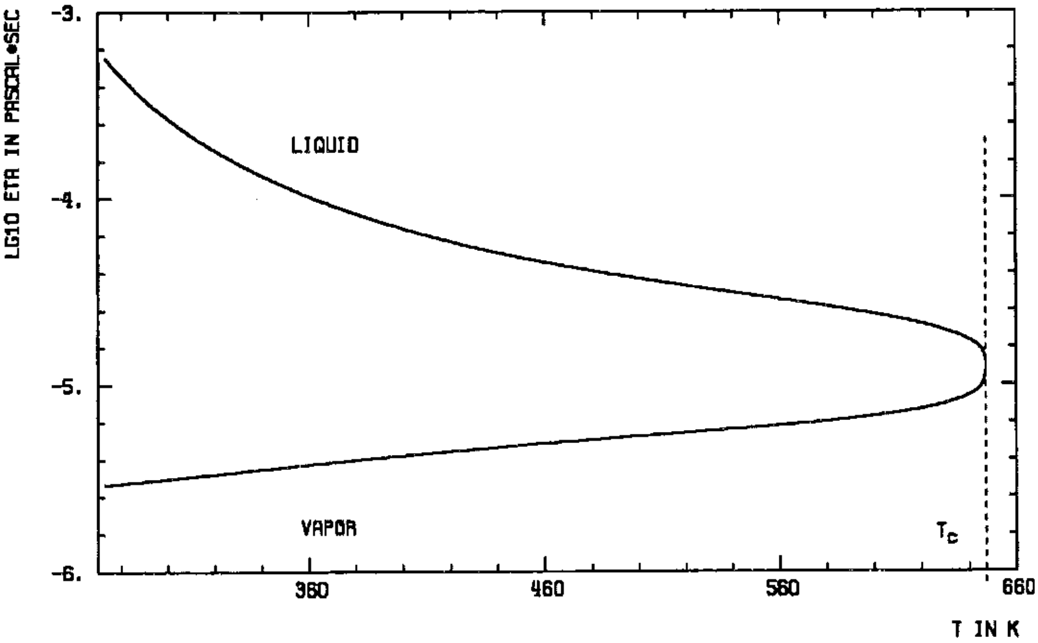


Figure 23. Viscosity of the water in the saturated states

In describing the viscosities in the cold liquid resp. vapor different functions of the critical variable w (Eq. [5.2]) were used :

$$\eta_L(T) = \frac{\eta_c}{\sum_{j=1}^{11} A_j \cdot w^{j-1}} \quad , \quad \eta_V(T) = \eta_c \cdot \sum_{j=1}^{9} C_j \cdot w^{j-1} \quad . \qquad [5.21]$$

Table 17. Polynomial coefficients A(1) - A(11) of the viscosity in the saturated liquid

1	0.945822019	2	-3.50567042	3	23.61506749
4	-176.2217806	5	960.5599634	6	-3533.866457
7	8609.716158	8	-13637.7187	9	13435.90472
10	-7453.116048	11	1776.272412		

Table 18. Polynomial coefficients C(1) - C(9) of the viscosity in the saturated vapor

1	0.9733280423	2	-4.061585251	3	20.72025593
4	-80.06616536	5	214.64856	6	-384.6194719
7	433.0818384	8	-278.1018786	9	78.25010044

In the vicinity of T_c density-like formulas were adapted :

$$\begin{aligned} \eta_L(T) &= \eta_c \bullet \left[1 + 1.50 \bullet \tau^{\beta} + \tau \left(U_1 + \tau \left(U_2 + \tau^2 U_4 \right) \right) \right] \quad , \\ \eta_V(T) &= \eta_c \bullet \left[1 - 1.50 \bullet \tau^{\beta} + \tau \left(W_1 + \tau \left(W_2 + \tau^2 W_4 \right) \right) \right] \quad . \end{aligned} \qquad [5.22]$$

The polynomial coefficients

$$\begin{aligned} U_1 &= 4.669845392 \quad , \quad U_2 = -19.58930391 \quad , \quad U_4 = 1768.652432 \\ W_1 &= 4.262680597 \quad , \quad W_2 = -34.89896734 \quad , \quad W_4 = 2452.105565 \end{aligned}$$

were calculated using the restrictions of double-smooth connections at 620 K to the "cold" descriptions. By comparing the functions [5.22] with the calculated viscosities above 635 K the best fitting gives the following critical viscosity :

$$\eta_c = 3.95 \bullet 10^{-5} \quad \text{Pa} \bullet \text{sec} \quad . \qquad [5.23]$$

The thermal conductivity

AQUA describes the thermal conductivities in the whole T-region as function of τ. Below 620 K the following polynomials were used :

$$\lambda_L(T) = \sum_{j=1}^{8} A_j \bullet \tau^{j-1} \quad , \quad \lambda_V(T) = \sum_{j=1}^{9} C_j \bullet \tau^{j-1} \quad . \qquad [5.24]$$

Table 19. Coefficients A(1) - A(8) of the thermal conductivity in the saturated liquid

1	0.397252283	2	1.361430056	3	0.2886579476
4	-1.137153088	5	-75.20016424	6	333.88123
7	-546.0632435	8	311.575529		

Table 20. Coefficients C(1) - C(9) of the thermal conductivity in the saturated vapor

1	0.2012600289	2	-2.546881317	3	19.24073027
4	-79.8925815	5	178.6322576	6	-195.5137779
7	62.06051011	8	38.86696488	9	-15.11624233

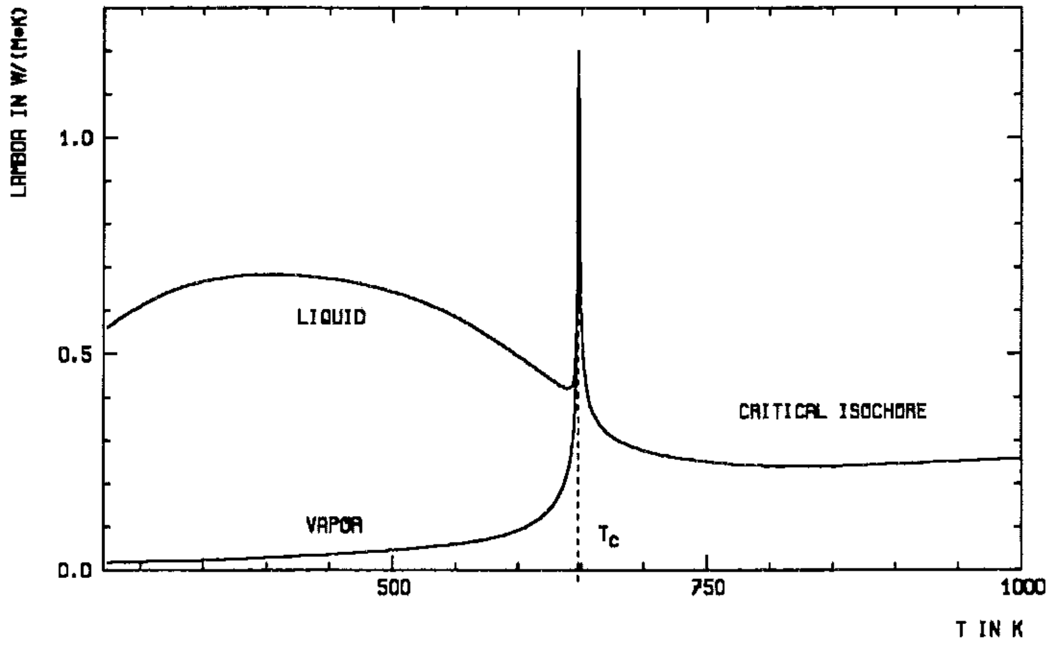


Figure 24. Thermal conductivity of the water as function of the temperature

At the critical point the thermal conductivity approaches infinty as

$$\lambda \propto \tau^{-0.5} \qquad [5.25]$$

Consequently above 620 K the following near-critical descriptions had been applied :

$$\lambda_L(T) = \tau^{-0.5} \cdot \sum_{j=1}^{5} U_j \cdot \tau^{j-1} \quad , \quad \lambda_V(T) = \tau^{-0.5} \cdot \sum_{j=1}^{5} W_j \cdot \tau^{j-1} \quad . \qquad [5.26]$$

The first coefficients of the appended polynomials are equal :

$$U_1 = W_1 = 0.02 \, .$$

For the remaining coefficients the condition of the double smooth continuation at 620 K gives :

$U_2 = 2.484618723$, $U_3 = -31.16080409$,
$U_4 = 421.0825394$, $U_5 = -2390.608760$,
$W_2 = 0.7221916099$, $W_3 = -29.169893490$,
$W_4 = 495.59365880$, $W_5 = -3169.2527940$.

5.5 Other saturated properties

The following properties are all described in AQUA as functions of the scaled temperature τ, only the form of the functions differ in the cold (T < 620 K) and in the near-critical (T > 620 K) part of saturation line. Below 620 K τ-polynomials were fitted to the property-values calculated with the H.G.K. code. Above 620 K scaled descriptions (s. e.g. Eq. [5.30]) are used. The values of the critical parameters in these formula are gained by fitting the descriptions to the H.G.K.-functions in the region

$$635\,K < T < T_c \quad . \qquad [5.27]$$

The remaining coefficients of the high-T descriptions are defined by demanding a double-smooth connection (Eq. [5.12]) for them to the low-T formulas at $T_s = 620\,K$.

In all the cases the descriptions had been developed for the reduced properties - P_ρ^0 , P_T^0 , C_V^0 and G^0 - since AQUA caltulates internally with these functions.

The density derivative of the pressure

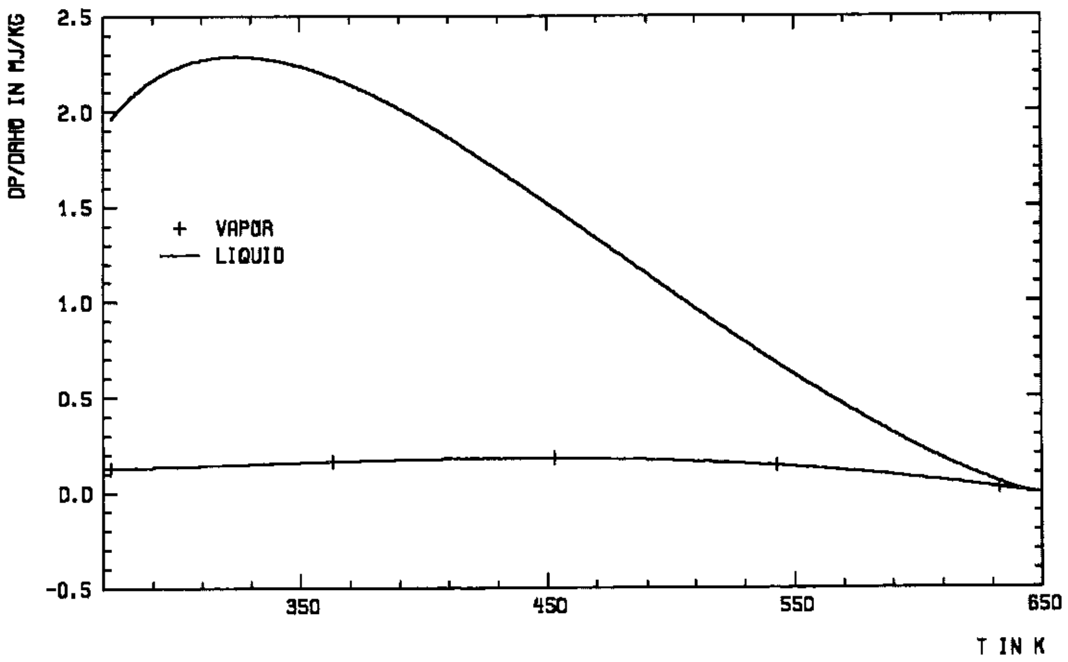


Figure 25. Density derivatives of the pressure in the saturated states

Table 21. Polynomial coefficients A(1) - A(12) of $\partial P/\partial\rho$ in the saturated liquid

1	0.09168326854	2	0.1410358804	3	293.745773
4	-3694.718543	5	35315.9487	6	-224505.1913
7	968102.1275	8	-2837321.019	9	5560661.663
10	-6973719.089	11	5058540.246	12	-1615244.963

Table 22. Polynomial coefficients C(1) - C(9) of $\partial P/\partial\rho$ in the saturated vapor					
1	0.001435429119	2	3.738553917	3	3.552096947
4	-55.88507759	5	217.5743793	6	-535.2445068
7	812.4224243	8	-670.9285889	9	229.0827484

Table 21 and Table 22 contain the coefficients of the low temperature description od the density-derivatives for the saturated liquid resp. for the saturated vapor

$$\frac{\partial P}{\partial\rho}_L(T) = R_{gas}\bullet T\sum_{j=1}^{12} A_j\bullet\tau^{j-1} \quad , \quad \frac{\partial P}{\partial\rho}_V(T) = R_{gas}\bullet T\sum_{j=1}^{9} C_j\bullet\tau^{j-1} \quad . \qquad [5.28]$$

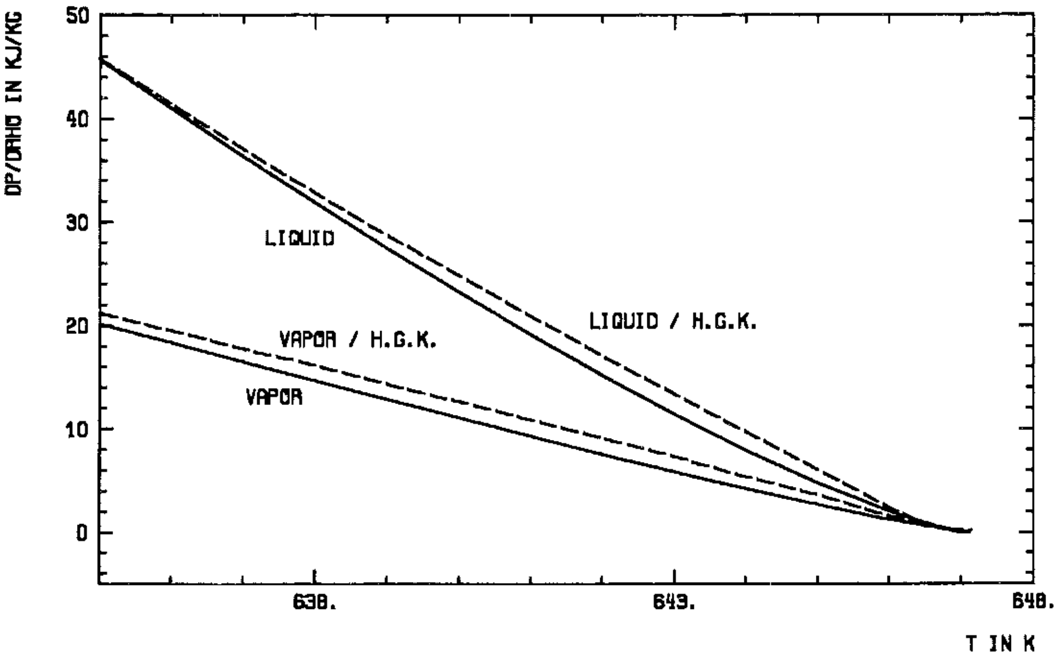


Figure 26. Density derivatives of the pressure in the vicinity of T_c

At the critical point the density-derivatives vanish as

$$\frac{\partial P}{\partial\rho}_L \propto \rho_L\bullet\tau^{\gamma} \qquad \text{resp.} \qquad \frac{\partial P}{\partial\rho}_V \propto \rho_V\bullet\tau^{\gamma} \qquad [5.29]$$

with the critical exponent $\gamma = 1.24$ (s. e.g. /17/) so the following near-critical descriptions had been applied :

$$\frac{\partial P}{\partial\rho}_L(T) = R_{gas}\bullet T\bullet\frac{\rho_L}{\rho_c}\bullet\tau^{\gamma}\sum_{j=1}^{4} U_j\bullet\tau^{j-1} \quad ,$$

$$\frac{\partial P}{\partial\rho}_V(T) = R_{gas}\bullet T\bullet\frac{\rho_V}{\rho_c}\bullet\tau^{\gamma}\sum_{j=1}^{4} W_j\bullet\tau^{j-1} \quad . \qquad [5.30]$$

The first coefficients of the polynomials in Eq. [5.30] are the same :

$$U_1 = W_1 = 15$$

and here are the remaining ones :

$$U_2 = -79.73057836 \quad , \quad U_3 = -430.1476885 \quad , \quad U_4 = 14139.7632 \quad ,$$
$$W_2 = 369.6750971 \quad , \quad W_3 = -5110.089071 \quad , \quad W_4 = 46121.0908 \quad .$$

The temperature derivative of the pressure

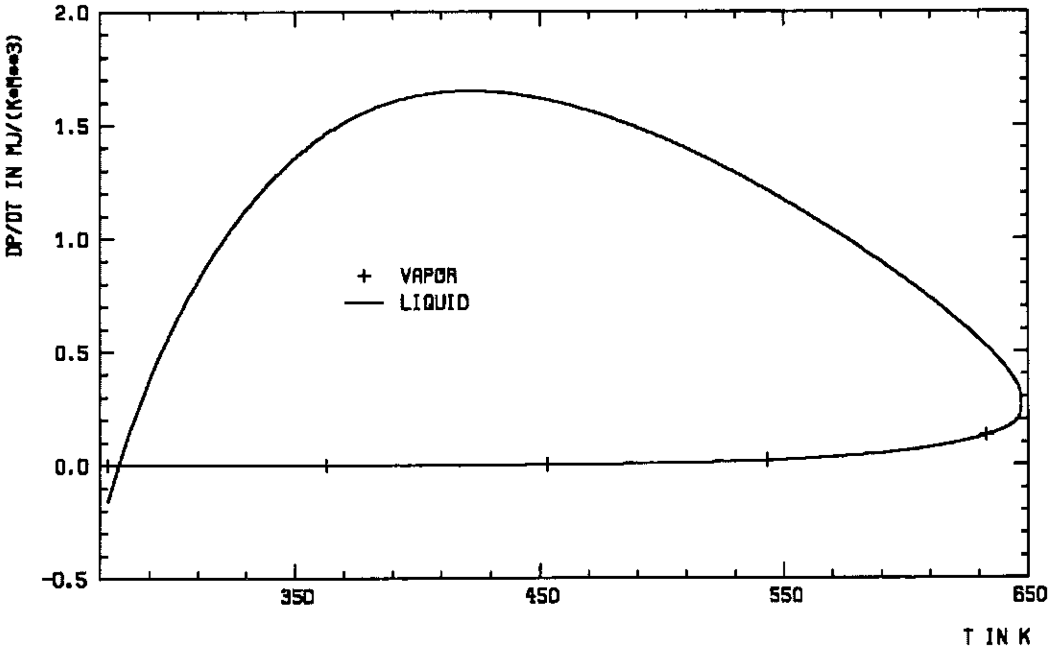


Figure 27. Temperature derivatives of the pressure in the saturated states

The temperature derivatives show quite a regular behaviour at T_c, so one can use polynomials as well on the cold saturation line as on the near critical one. Below 620 K the polynomial fittings resulted in

$$\frac{\partial P}{\partial T}_L(T) = R_{gas} \cdot \rho_L \sum_{j=1}^{12} A_j \cdot \tau^{j-1} \quad , \quad \frac{\partial P}{\partial T}_V(T) = R_{gas} \cdot \rho_V \sum_{j=1}^{10} C_j \cdot \tau^{j-1} \quad . [5.31]$$

Table 23. Polynomial coefficients A(1) - A(12) of $\partial P/\partial T$ in the saturated liquid

1	1.818884544	2	16.40737736	3	-65.85455759
4	52.04371914	5	4031.238729	6	-47257.0313
7	277423.4268	8	-988332.4003	9	2218919.758
10	-3077677.674	11	2414555.699	12	-821000.2244

Table 24. Polynomial coefficients C(1) - C(10) of $\partial P/\partial T$ in the saturated vapor					
1	2.029700729	2	-0.1255470649	3	-61.54407176
4	544.8751175	5	-2848.188899	6	9708.977783
7	-21262.51929	8	28710.96533	9	-21755.27246
10	7082.224609				

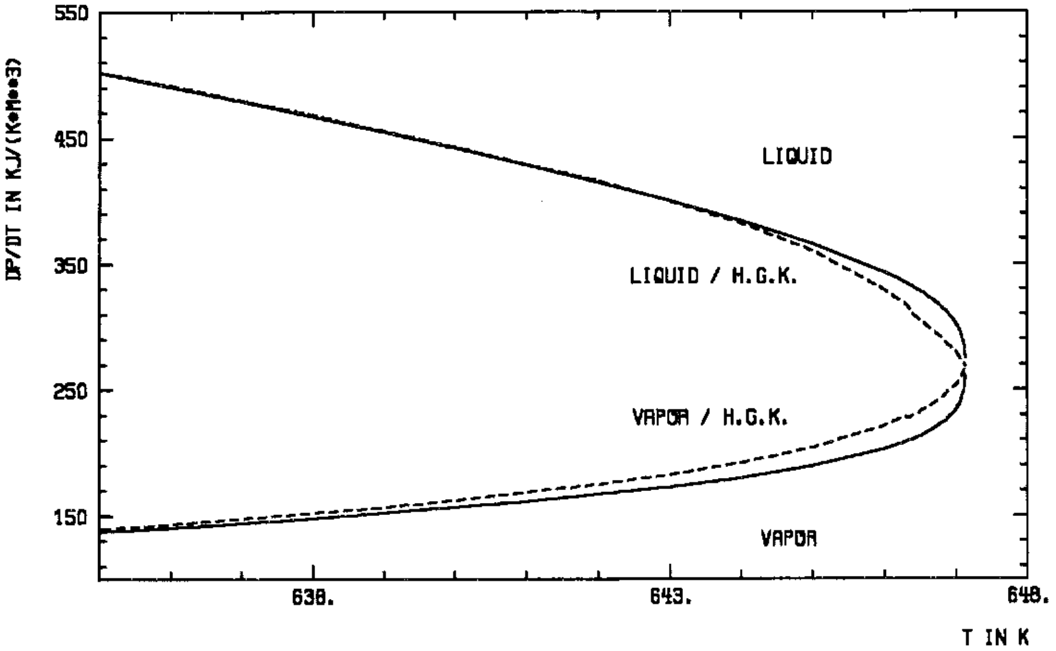


Figure 28. Temperature derivatives of the pressure in the vicinity of T_c

Above 620 K the double smooth extension gives

$$\frac{\partial P}{\partial T}_L(T) = R_{gas} \cdot \rho_L \sum_{j=1}^{4} U_j \cdot \tau^{j-1} \quad ,$$

$$\frac{\partial P}{\partial T}_V(T) = R_{gas} \cdot \rho_V \sum_{j=1}^{4} W_j \cdot \tau^{j-1} \quad . \qquad [5.32]$$

The first coefficients of the polynomials in Eq. [5.32] must be the same

$$U_1 = W_1 = 1.80215732$$

to fulfil

$$\frac{\partial P}{\partial T}_V(T_c) = \frac{dP^{\times}(T_c)}{dT} = \frac{\partial P}{\partial T}_L(T_c) \quad . \qquad [5.34]$$

The remainig ones are :

$U_2 = 17.63135647$, $U_3 = -97.95806903$, $U_4 = 451.8053522$,

$W_2 = 16.00172011$, $W_3 = -439.3063303$, $W_4 = 3359.729201$.

The heat capacities

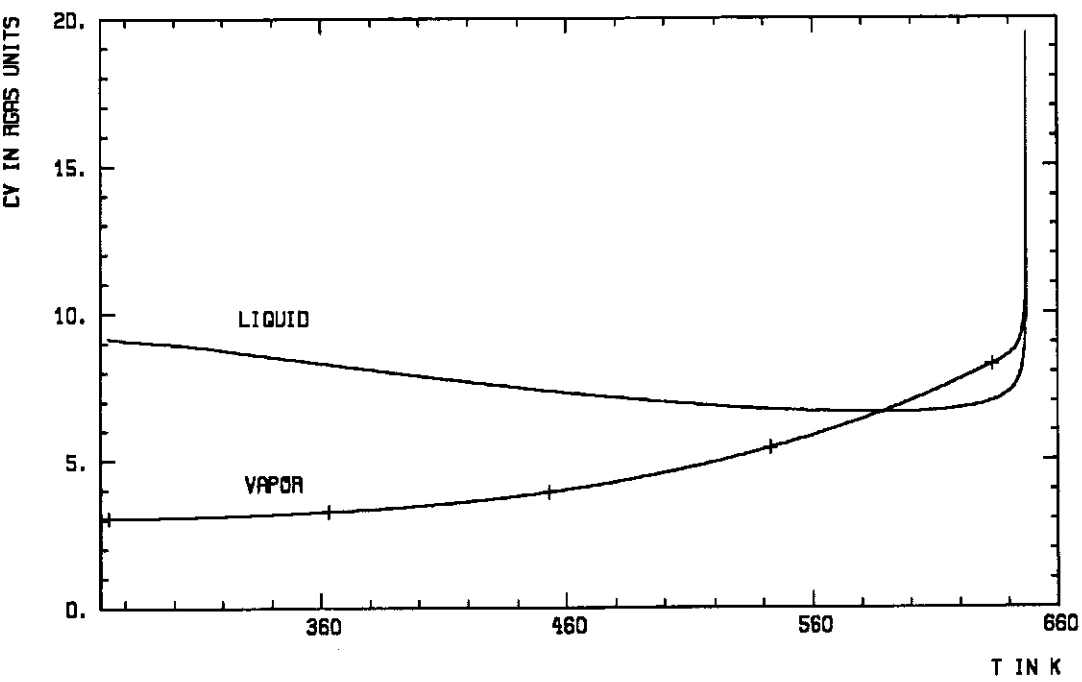


Figure 29. Heat capacities of the water in the saturated states.

AQUA describes the heat capacities in R_{gas} units. Below 620 K polynomials were used :

$$C_{VL}(T) = R_{gas} \cdot \sum_{j=1}^{12} A_j \cdot \tau^{j-1} \quad , \quad C_{VV}(T) = R_{gas} \cdot \sum_{j=1}^{10} C_j \cdot \tau^{j-1} \quad . \qquad [5.35]$$

Table 25. Polynomial coefficients A(1) - A(12) of C_V in the saturated liquid

1	7.4305055	2	-24.93618016	3	195.5654567
4	1986.485797	5	-53305.43411	6	505697.1723
7	-2724774.677	8	9167737.673	9	-19622033.78
10	25984725.33	11	-19419431.35	12	6263206.554

Table 26. Polynomial coefficients C(1) - C(10) of C_V in the saturated vapor

1	8.956404735	2	-33.97230774	3	126.5786602
4	-513.7462523	5	1521.138693	6	-2182.266721
7	-73.57828067	8	4483.361889	9	-5407.324042
10	2099.821164				

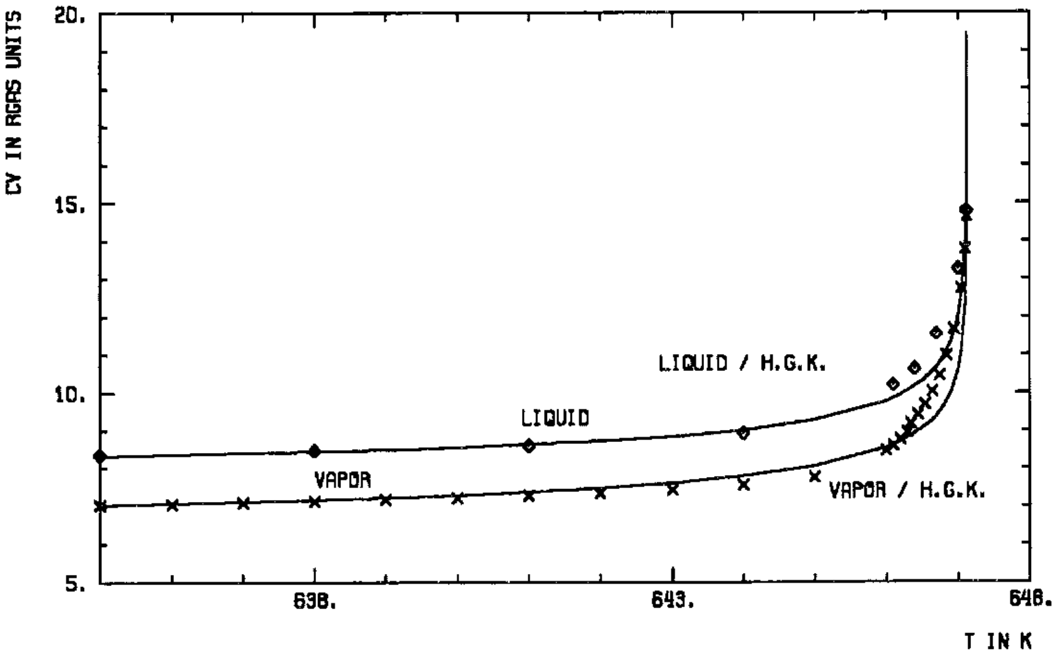


Figure 30. Heat capacities of the water in the vicinity of T_c

At the critical point the heat capacity approaches infinty as

$$C_V \propto \tau^{\alpha} \qquad [5.37]$$

with the critical exponent $\alpha = -0.1$ (s. e.g. /17/). Consequently above 620 K the following near-critical descriptions had been applied :

$$C_{VL}(T) = R_{gas} \cdot \tau^{\alpha} \cdot \sum_{j=1}^{4} U_j \cdot \tau^{j-1} \quad , \quad C_{VV}(T) = R_{gas} \cdot \tau^{\alpha} \cdot \sum_{j=1}^{4} W_j \cdot \tau^{j-1} \quad . \qquad [5.38]$$

The double smooth continuation at 620 K results in the polynomial-coefficients :

$U_1 = 4.5$, $U_2 = 15.21452007$, $U_3 = -167.2951053$, $U_4 = 1042.333618$,

$W_1 = 5.1$, $W_2 = 42.35451412$, $W_3 = -1002.675527$, $W_4 = 6911.203183$.

The free enthalpy, G

The Gibbs function shows not even a trace of irregularity at T_c. A single polynomial suffices therefore, to describe it in the whole temperature region :

$$G_L(T) = G_V(T) = R_{gas} \cdot T \sum_{j=1}^{9} A_j \cdot \tau^{j-1} \quad . \qquad [5.39]$$

Table 27. Polynomial coefficients A(1) - A(9) of the free enthalpy of the saturated states					
1	-2.569516966	2	5.187876405	3	0.6408677408
4	1.374106295	5	-18.01831251	6	68.53337198
7	-176.2024879	8	226.296896	9	-130.4997179

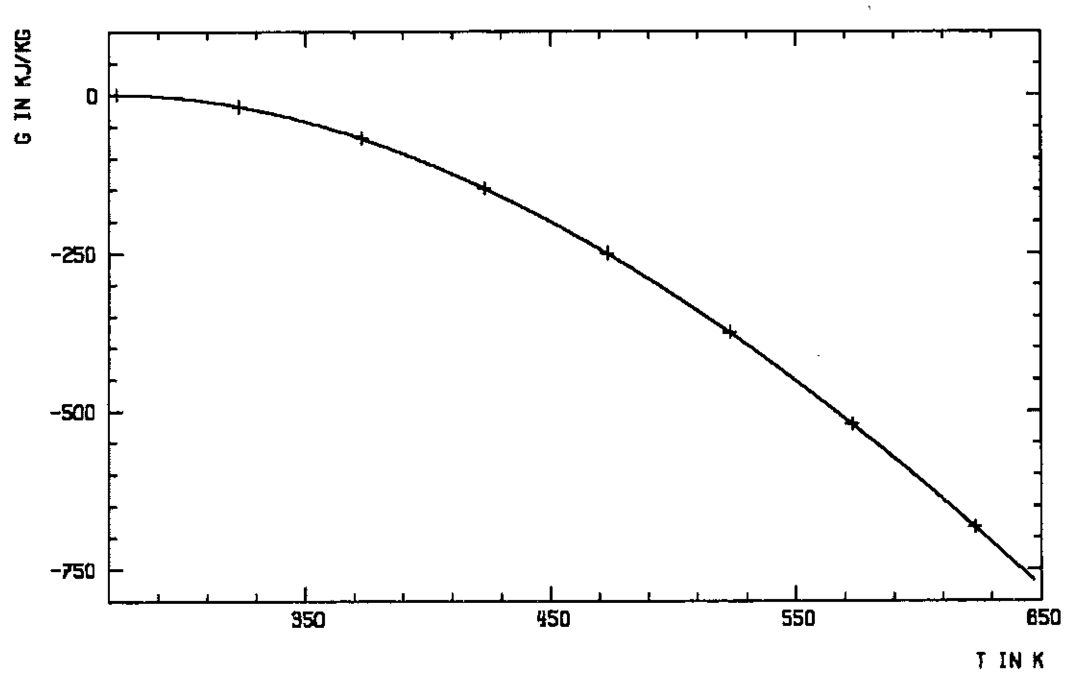


Figure 31. Free enthalpy - G - of the water in the saturated states

6.0 Calculating the densities from the pressures

IVA3 supplies the equation of state with the temperature and the pressure as independent variables, the H.G.K.-routines calculate the thermophysical properties as temperature-density functions. So obviously a procedure is needed to calculate the variable-pair (T, ρ) from the variable-pair (T, P). AQUA uses three routines to do this task : DLFIND seeks ρ(T, P) in the liquid, DVFIND in the vapor, and DGFIND in the supercritical water.

6.1 Calculating small density changes

All of these routines

DGFIND(T,P,ρ,IECCH)
DLFIND(T,P,ρ,IECCH)
DVFIND(T,P,ρ,IECCH)

use the same searching-algorithm : beginning with an initial density, ρ_0 - supplied by the user - a first pressure, P_0 and $\partial P/\partial\rho$ are calculated, then a corrected density is derived - from the difference, P - P_0 - by Newton's method.

Table 28. Coefficients of P_{Lmax} (T).

$$P_{Lmax} = A_0 + A_1 T^1 + A_2 T^2 + \ \ldots$$

T >	0 K	373.15 K	647.126 K
A_0	$128.0317635 \cdot 10^9$	$1.121372093 \cdot 10^9$	$-731.6836907 \cdot 10^6$
A_1	$-1.503427271 \cdot 10^9$	$-6.278371391 \cdot 10^6$	$6.597512847 \cdot 10^6$
A_2	$6.608885188 \cdot 10^6$	$26.44577782 \cdot 10^3$	$-7.08407094 \cdot 10^3$
A_3	$-12.83080180 \cdot 10^3$	-33.36474460	5.764846806
A_4	9.305682808	$14.81761478 \cdot 10^{-3}$	$-3.188721412 \cdot 10^{-3}$
A_5			$1.193485372 \cdot 10^{-6}$
A_6			$-2.946300976 \cdot 10^{-10}$
A_7			$4.575561580 \cdot 10^{-14}$
A_8			$-4.040003535 \cdot 10^{-18}$
A_9			$1.543953233 \cdot 10^{-22}$

Before beginnig with the iteration, the routines check, if there is a posibility at all to find a density, corresponding to the supplied pressure, P in the given area. To this aim P is compared with a pressure-window,

$$P_{\text{min}} \leq P \leq P_{\text{max}}$$

and if P don't fit into it, the routines end with an error message (for the error handling s. Appendix L, "Return codes" on page 151). The borders of the pressure-window are

$$P_{\text{min}} = 0 \quad , \quad P_{\text{max}} = P_{Lmax}(T) \qquad [6.1]$$

in the supercritical water,

$$P_{\min} = 0 \quad , \quad P_{\max} = P_{SV}(T) \tag{6.2}$$

in the vapor, and

$$P_{\min} = P_{SL}(T) \quad , \quad P_{\max} = P_{Lmax}(T) \tag{6.3}$$

in the liquid respectively.

P_{Lmax} in the above eq.s is the pressure at the liquid edge of the validity-domain

$$P_{Lmax}(T) = P(T,\ 1200\,kg/m^3) \quad . \tag{6.4}$$

AQUA calculates P_{Lmax} with the temperature polynomials given in Table 28 (routine PLMAX). The description is valid in the temperature range

$$273.15\,K \leq T \leq 5000\,K \quad .$$

The remaining two temperature-functions in Eq. [6.1 - 6.3], P_{SV} (T) resp. P_{SL} (T) are the spinodal pressures in the vapor resp. in the liquid, Eq. [5.16 - 5.17]. At least as long the temperatures remain below 635. K. Abowe this point the shape of the H.G.K.-isotherms begins to differ markedly from the shape, one would expect in the neighbourhood of the critical point. As well the pressure, as the density - particularly of the liquid spinodal - differ from the descriptions given in 5.0, "Describing the water properties in the saturated states" on page 27 (compare also Figure 18 on page 31). Figure 32 shows a massively enlarged part of the 647 K isotherm of the water. On this isotherm the liquid spinodal lies in the vapor region already.

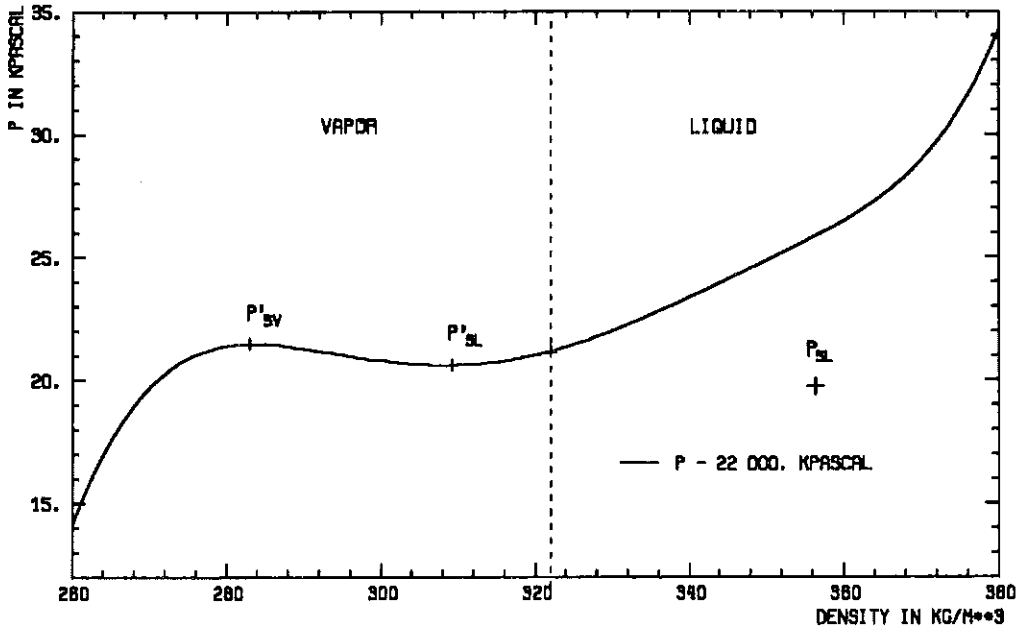


Figure 32. Water pressures on the 647 K isotherm

AQUA uses additional τ-polynomials to describe the real pressure minima and maxima P'_{SL}, P'_{SV} in the region above 635 K :

$$P'_{SL}(T) = P_{SL}(T) + \sum_{j=1}^{9} U_j \cdot \tau^j \quad ,$$
$$P'_{SV}(T) = P_{SV}(T) - \sum_{j=1}^{6} W_j \cdot \tau^j \quad . \qquad [6.5]$$

with the coefficients

Table 29. Polynomial coefficients U(1) - U(9) of the correcting term for P'_{SL}

1	40.49363544	2	$-20.73773694 \cdot 10^3$	3	$7.607260219 \cdot 10^6$
4	$-1.796520658 \cdot 10^9$	5	$258.9287847 \cdot 10^9$	6	$-22.71838331 \cdot 10^{12}$
7	$1.182066587 \cdot 10^{15}$	8	$-33.45912597 \cdot 10^{15}$	9	$396.4473916 \cdot 10^{15}$

resp.

Table 30. Polynomial coefficients W(1) - W(6) of the correcting term for P'_{SV}

1	17.17830396	2	$-3.434969255 \cdot 10^3$	3	$448.7956073 \cdot 10^3$
4	$-38.08457011 \cdot 10^6$	5	$1.646319316 \cdot 10^9$	6	$-27.19071374 \cdot 10^9$

(routines DILE resp. DIVE).

In a thin region before the critical temperature,

$$647.0\,K < T < T_c$$

it can happen, that the liquid pressures of an isotherm remain always above the vapor pressures

$$P'_{SV} < P'_{SL} \quad .$$

This would lead to the unacceptable state, that at a distinct (near critical) temperature there would be a pressure, finding no corresponding density either in the vapor, or in the liquid. To avoid this unphisical state DLFIND checks at near critical temperatures the spinodal pressures and uses the smaller of them as P_{min} in the window [6.3].

6.2 Calculating large density changes

If the supplied density, ρ_0 differs massively from the final density, ρ (T, P), then in some cases the 19 iteration-steps, allowed in the FIND-routines will not suffice to calculate the final density. The user can circumvent this difficulty by supplying in this case a zero initial density,

$$\rho_0 = 0$$

to the FIND-routines. Finding a zero initial density the routines search - after checking the pressure, yet before beginning with the iteration - for an adequate approximated density, which will then serve as an initial density for the iteration with Newton's method. To find this density, the routines use an algorithm, similar to the one, used to calculate the spinodal densities (page 30).

DVFIND selects three densities with the corresponding pressures

$$\begin{aligned} \rho_1 &= 0 &, \quad P_1 &= 0 \\ \rho_2 &= \rho_V &, \quad P_2 &= P^x \\ \rho_3 &= \rho_{SV} &, \quad P_3 &= P_{SV} \end{aligned} \qquad [6.6]$$

and calls the routine DINVA to find a density, corresponding the supplied pressure, P with the help of these points. DINVA supplies the points to the routine QUADO, which constructs from them a polynomial of second grade and solves it for ρ_0 (P). DINVA then calculates the pressure, P (ρ_0) and checks, if the ρ_0 is good enough to begin with. If P (ρ_0) differs to much from P, then DINVA uses SORBET to discard the "worst" of the four points

$$\rho_1 \quad , \quad \rho_2 \quad , \quad \rho_3 \quad , \quad \rho_0$$

and seeks with the remainig three a better ρ_0.

DLFIND uses

$$\begin{aligned} \rho_1 &= \rho_{SL} &, \quad P_1 &= P_{SL} \\ \rho_2 &= \rho_L &, \quad P_2 &= P^x \\ \rho_3 &= 1200 kg/m^3 &, \quad P_3 &= P_{Lmax} \end{aligned} \qquad [6.7]$$

as pivotal points and calls the routine DINLI to find the density, ρ_0. DINLI works the same way as DINVA.

DGFIND has a more complicated task to fulfill, as either DVFIND or DLFIND. Since at supercritical temperatures there are no ready-made density-pressure relations to use - as the saturation line or the spinodals - some density-pressure relations must be constructed.

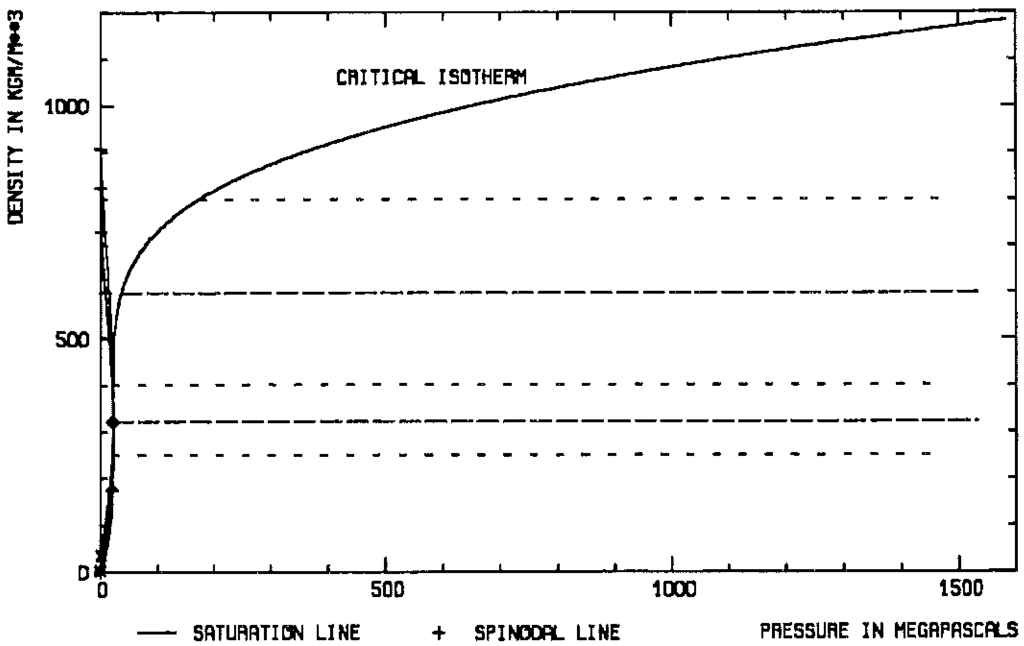


Figure 33. Water. Density-pressure chart

To this end DGFIND divides the supercritical region in three sub-areas:

$$\text{"LD"}: \quad 0 \leq \rho \leq \rho_c$$

the area of low-densities,

$$\text{"MD"}: \quad \rho_c \leq \rho \leq 600\,kg/m^3$$

the area of moderate-densities and

$$\text{"HD"}: \quad 600\,kg/m^3 \leq \rho \leq 1200\,kg/m^3$$

the area of high-densities (s. Figure 33). AQUA supplies DGFIND with the following isochoral pressures :

$$P_{250}(T) = P(T,\ 250\,kg/m^3) \qquad [6.8]$$

Table 31. Polynomial coefficients A(1) - A(8) of P_{250} (T). $P_{250} = A_0 + A_1 \bullet T + A_2 \bullet T^2 + \ldots$

1	$-161.6042550 \bullet 10^{6}$	2	$370.3861090 \bullet 10^{3}$	3	-183.5936079
4	$93.96635527 \bullet 10^{-3}$	5	$-30.07808886 \bullet 10^{-6}$	6	$5.819644183 \bullet 10^{-9}$
7	$-619.7247350 \bullet 10^{-15}$	8	$27.81950727 \bullet 10^{-18}$		

$$P_{322}(T) = P(T,\ \rho_c) \qquad [6.9]$$

Table 32. Polynomial coefficients A(1) - A(10) of P_{322} (T). $P_{322} = A_0 + A_1 \bullet T + A_2 \bullet T^2 + \ldots$

1	$-115.3411242 \bullet 10^{6}$	2	$-14.61390831 \bullet 10^{3}$	3	713.8479434
4	$-843.0259976 \bullet 10^{-3}$	5	$558.6511529 \bullet 10^{-6}$	6	$-228.9982666 \bullet 10^{-9}$
7	$59.25205508 \bullet 10^{-12}$	8	$-9.423230731 \bullet 10^{-15}$	9	$841.0373279 \bullet 10^{-21}$
10	$-32.24390358 \bullet 10^{-24}$				

$$P_{400}(T) = P(T,\ 400\,kg/m^3) \qquad [6.10]$$

Table 33. Polynomial coefficients A(1) - A(10) of P_{400} (T). $P_{400} = A_0 + A_1 \bullet T + A_2 \bullet T^2 + \ldots$

1	$-48.22391500 \bullet 10^{6}$	2	$-534.0793594 \bullet 10^{3}$	3	$1.893391113 \bullet 10^{3}$
4	-2.052518308	5	$1.304520061 \bullet 10^{-3}$	6	$-521.2344512 \bullet 10^{-9}$
7	$132.5272444 \bullet 10^{-12}$	8	$-20.80959374 \bullet 10^{-15}$	9	$1.839373352 \bullet 10^{-18}$
10	$-69.98460931 \bullet 10^{-24}$				

$$P_{600}(T) = P(T,\ 600\,kg/m^3) \qquad [6.11]$$

Table 34. Polynomial coefficients A(1) - A(10) of P_{600} (T). $P_{600} = A_0 + A_1 \cdot T + A_2 \cdot T^2 + \ldots$

1	$-193.5182531 \cdot 10^{6}$	2	$-662.0824295 \cdot 10^{3}$	3	$3.041300733 \cdot 10^{3}$
4	$--3.331588396$	5	$2.091973967 \cdot 10^{-3}$	6	$-822.3881849 \cdot 10^{-9}$
7	$205.8600153 \cdot 10^{-12}$	8	$-31.88132588 \cdot 10^{-15}$	9	$2.784677333 \cdot 10^{-18}$
10	$-104.8772468 \cdot 10^{-24}$				

and $$P_{800}(T) = P(T,\ 800\,kg/m^3) \qquad [6.12]$$

Table 35. Polynomial coefficients A(1) - A(10) of P_{800} (T). $P_{800} = A_0 + A_1 \cdot T + A_2 \cdot T^2 + \ldots$

1	$-576.9907617 \cdot 10^{6}$	2	$229.0441320 \cdot 10^{3}$	3	$3.275463692 \cdot 10^{3}$
4	-4.331244582	5	$3.008662103 \cdot 10^{-3}$	6	$-1.272178994 \cdot 10^{-6}$
7	$337.9386420 \cdot 10^{-12}$	8	$-55.07102702 \cdot 10^{-15}$	9	$5.030253110 \cdot 10^{-18}$
10	$-197.1414663 \cdot 10^{-24}$				

(routines P250, P322, P400, P600 and P800). The above polynomials are valid in the temperature range

$$647.126\,K \leq T \leq 5000\,K \quad .$$

If now, in the beginnig DGFIND finds a zero for the initial density, then it will compare the supplied pressure with three pressure-windows :

$$0 \leq P \leq P_{322} \quad , \qquad [6.13]$$

$$P_{322} \leq P \leq P_{600} \quad \text{and} \qquad [6.14]$$

$$P_{600} \leq P \leq P_{Lmax} \qquad [6.15]$$

to decide, in which of the three sub-areas to expect the density. If the pressure P lies in window [6.13] (sub-area "LD"), then DGFIND selects the following pivotal points

$$\begin{aligned} \rho_1 &= 0 &, \quad P_1 &= 0 \\ \rho_2 &= 250\,kg/m^3 &, \quad P_2 &= P_{250} \\ \rho_3 &= \rho_c &, \quad P_3 &= P_{322} \end{aligned} \qquad [6.16]$$

(s. Figure 33) and supplies them to the routine DINSU to get the initial density ρ_0. If P suffices the inequality [6.14] (sub-area "MD"), then the supplied corner-points will be

$$\begin{aligned} \rho_1 &= \rho_c &, \quad P_1 &= P_{322} \\ \rho_2 &= 400\,kg/m^3 &, \quad P_2 &= P_{400} \\ \rho_3 &= 600\,kg/m^3 &, \quad P_3 &= P_{400} \end{aligned} \qquad [6.17]$$

and if P lies inside the window [6.15] (sub-area "HD"), then DGFIND uses the points

$$\begin{aligned} \rho_1 &= 600\ kg/m^3 \quad , \quad P_1 = P_{600} \\ \rho_2 &= 800\ kg/m^3 \quad , \quad P_2 = P_{800} \\ \rho_3 &= 1200\ kg/m^3 \quad , \quad P_3 = P_{Lmax} \end{aligned} \qquad [6.18]$$

to get an initial density, ρ_0.

DINSU calculates ρ_0 the same way as DINVA or DINLI.

7.0 Using AQUA

7.1 The codesystem AQUA

can be used in different ways :

1. the user can call the interactive procedure WATER, to get the water properties at a given (T,ρ) resp. (T,P) point displayed on the screen imediately,
2. it is also possible to acquire water properties for a whole string of points, either along a number of isotherms or along a set of isochores in a TSO-session by calling the procedure KAISOT or KAISOK,
3. finally, there are also the routines WASSER, DAMPF, RDRDP, SAET01 and VD1 which supply the water properties to the code IVA3 and which could be used by other codes too.

SERVUS : auxiliary datasets and routines

The procedures KAISOT and KAISOK calculate the water properties along a string of density resp. temperature points. To be able to do that, they need

- a SERVUS-dataset to supply a sample isotherm resp. a sample isochore for these routines and to receive the calculated properties,
- some routines from the code SERVUS to read, inspect, handle, sort and store the functions appearing in the procedures.

The code SERVUS (s. /19/) resides in the dataset ***INR105.SERVUS.LOAD***.

The sample functions are either in the dataset ***INR105.AQUA.DATA*** or in ***INR105.AQUAT.DATA***. AQUA holds sample isotherms with some 300 density points in the range $\rho < 1200\ kg/m^3$. In AQUAT there are temperature functions with 400 temperature points between the melting and the critical points.

The user can acquire the above datasets by copying and adapting the procedure ***INR105.SERVUS.CNTL(IEBGENER)*** to ***userid.SERVUS.CNTL(IEBGENER)*** (s. Figure 34).

```
//userida1 JOB (0000,999,abcde),dombrowsky,MSGLEVEL=(1,1),
// NOTIFY=userid,MSGCLASS=H
//IEBG EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=INR,DSNAME=INR105.AQUA.DATA,DISP=SHR
//SYSUT2 DD UNIT=iii,DSNAME=userid.AQUA.DATA,DISP=(,CATLG),
// SPACE=(TRK,(50,10),RLSE),DCB=(RECFM=VBS,LRECL=32760,BLKSIZE=19069)
//SYSIN DD DUMMY
//
```

Figure 34. The procedure userid.SERVUS.CNTL(IEBGENER).

The "string"-routines KAISOT and KAISOK need, besides of the SERVUS routines also the AQUA-routines SIGNAT, NAPSZM and MUSTER.

- MUSTER helps in selecting the sample-function in the SERVUS dataset,
- SIGNAT sets the chracteristic numbers and words - needed in registering the function in the dataset - for the properties calculated and
- NUMBER returns the number of day in the year.

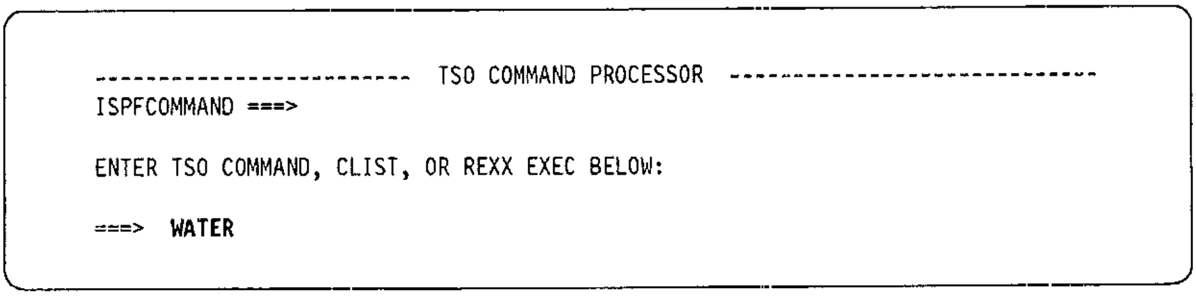
7.2 WATER

WATER calculates the water properties either as pressure-temperature or as density-temperature functions. In the stable and metastable states WATER returns property-values, corresponding to the ones, calculated by the H.G.K.-code. In the region of instability WATER calculates mixed state properties, as described in 4.0, "Thermal properties in the two phase region."

WATER needs the "FIND"-routines for the (T,P) ⇒ (T, ρ) conversion and HYDOR to calculate the water properties in the (ρ,T) - dependence (s. Appendix D, "Module trees" on page 77).

WATER resides in the dataset 'INR105.KATHER.LOAD' and can be activated by typing simply "WATER".

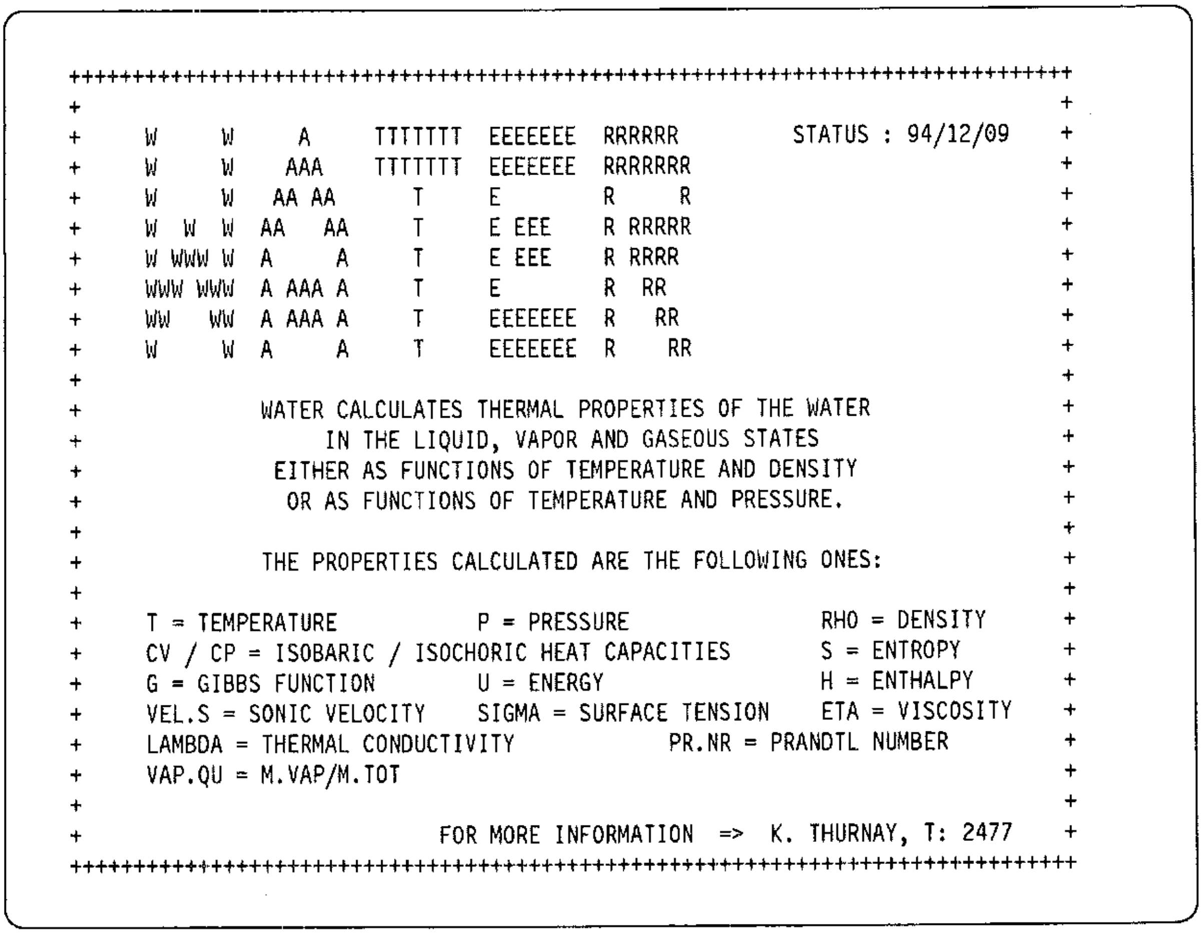
The following frozen screens exemplify a typical conversation with WATER :



```
------------------------- TSO COMMAND PROCESSOR -----------------------------
ISPFCOMMAND ===>

ENTER TSO COMMAND, CLIST, OR REXX EXEC BELOW:

===>  WATER
```



```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+                                                                              +
+     W     W     A     TTTTTTT  EEEEEEE  RRRRRR         STATUS : 94/12/09     +
+     W     W    AAA    TTTTTTT  EEEEEEE  RRRRRRR                              +
+     W     W   AA AA      T     E        R     R                              +
+     W  W  W   AA  AA     T     E EEE    R RRRRR                              +
+     W WWW W  A     A     T     E EEE    R RRRR                               +
+     WWW WWW  A AAA A     T     E        R  RR                                +
+     WW   WW  A AAA A     T     EEEEEEE  R   RR                               +
+     W     W  A     A     T     EEEEEEE  R    RR                              +
+                                                                              +
+              WATER CALCULATES THERMAL PROPERTIES OF THE WATER                +
+                   IN THE LIQUID, VAPOR AND GASEOUS STATES                    +
+               EITHER AS FUNCTIONS OF TEMPERATURE AND DENSITY                 +
+                OR AS FUNCTIONS OF TEMPERATURE AND PRESSURE.                  +
+                                                                              +
+              THE PROPERTIES CALCULATED ARE THE FOLLOWING ONES:               +
+                                                                              +
+     T = TEMPERATURE           P = PRESSURE               RHO = DENSITY       +
+     CV / CP = ISOBARIC / ISOCHORIC HEAT CAPACITIES       S = ENTROPY         +
+     G = GIBBS FUNCTION        U = ENERGY                 H = ENTHALPY        +
+     VEL.S = SONIC VELOCITY    SIGMA = SURFACE TENSION    ETA = VISCOSITY     +
+     LAMBDA = THERMAL CONDUCTIVITY          PR.NR = PRANDTL NUMBER            +
+     VAP.QU = M.VAP/M.TOT                                                     +
+                                                                              +
+                            FOR MORE INFORMATION  =>  K. THURNAY, T: 2477     +
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
ENTER TEMPERATURE ( IN K ) /   373.150 / =:       ( STOP WTIH T = 0 )
 00011 ?
373.15
```

```
DO YOU WANT THE PROPERTIES AS
     1 = TEMPERATURE-DENSITY FUNCTIONS
     2 = TEMPERATURE-PRESSURE FUNCTIONS                ( STOP WITH 0 )
 00014 ?
2
```

```
ENTER PRESSURE ( IN  MPASCAL ) /  0.1013250     / =:    ( STOP WTIH P = 0 )
 00019 ?
0.1013250
```

```
LIQUID ( 1 ) OR VAPOR ( 2 ) STATE ? ( 0 = STOP ) =:
 00028 ?
1
```

```
T =   373.150            RHO =  958.3926           P =  0.1013458
  K                        KG/M**3                   MPASCAL
DP/DRHO =  2125.38       DP/DT =  1.52786          DRHO/DT = -0.718864
  KJ/KG                    MPASCAL/K                 KG/(K*M**3
CV =  3.77087            CP =  4.21707             S =   1.30688
  KJ/(KG*K)                KJ/(KG*K)                 KJ/(KG*K)
U =   418.958            H =   419.064             G =  -68.5997
  KJ/KG                    KJ/KG                     KJ/KG
DU/DRHO = -0.620588      DH/DP =  0.751373E-03     VEL.S =  1541.71
  KJ*M**3/KG**2            M**3/KG                   M/SEC
ETA = 0.282103E-03       LAMBDA = 0.679215         SIGMA = 0.589152E-01
  KG/M/SEC                 W/M/K                     KG/SEC**2
PR.NR =  1.75150         VAP.QU = 0.352389E-07     PS = 0.1013127
                                                     MPASCAL
RH.V = 0.5975324         RH.SV =  12.67164         P.SV =  1.21103
  KG/M**3                  KG/M**3                   MPASCAL
RH.L =  958.4467         RH.SL =  823.0227         P.SL = -149.037
  KG/M**3                  KG/M**3                   MPASCAL
```

```
ENTER PRESSURE ( IN  MPASCAL ) /  0.1013250     / =:     ( STOP WTIH P = 0 )
 00019 ?
0
```

```
DO YOU WANT THE PROPERTIES AS
     1 = TEMPERATURE-DENSITY FUNCTIONS
     2 = TEMPERATURE-PRESSURE FUNCTIONS                ( STOP WITH 0 )
 00014 ?
0
```

```
ENTER TEMPERATURE ( IN K ) /   373.150 / =:       ( STOP WTIH T = 0 )
 00011 ?
0
```

7.3 *String-routines*

The routines KAISOT and KAISOK are placed in the dataset 'INR105.AQUA.LOAD'

KAISOT

KAISOT calculates the water properties as density-temperature functions along isotherms.

KAISOT needs (s. Appendix D, "Module trees" on page 77) :

- HYDOR for the calculation of the (ρ,T)-properties,
- the AQUA-routines SIGNAT, NAPSZM, MUSTER
- the SERVUS-routines SERDIO, SDEXG8, FENST8, SDING8
- the dataset ***userid.AQUA.DATA*** and
- the procedure **userid.AQUA.CLIST(KAISOT)** (s. Figure 35).

```
ALLOC DA('userid.AQUA.DATA') F(FT15F001) SHR REU
ALLOC DA('userid.AQUA.DATA') F(FT16F001) SHR REU
CALL 'INR105.AQUA.LOAD(KAISOT)'
FREE DA('userid.AQUA.DATA')
FREE DA('userid.AQUA.DATA')
```

Figure 35. The procedure userid.AQUA.CLIST(KAISOT)

A sample conversation with KAISOT is something, like this :

```
 READY
ex aqua(kaisot)
```

```
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+                                                                             +
+      PROGRAM KAISOT                                  STATUS : 94/11/30      +
+                                                                             +
+      CALCULATES THERMAL PROPERTIES OF THE WATER ALONG AN ISOTHERM           +
+                        AS FUNCTIONS OF THE DENSITY                          +
+            USING A SAMPLE (RH,FRH)-FUNCTION AS A DENSITY-PATTERN            +
+                                                                             +
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++



+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+                                                                             +
+     12.12.94     LIST OF THE GRAPHS OF THE DATASET        AQUA    .DATA     +
+                                                                             +
+                       STORAGE-NORM IS      "GRAPHIC8"                       +
+                                                                             +
+     DATE OF THE ALLOCATION:   23. 5.91      LAST SCRATCH:      27. 9.94     +
+                                                                             +
+     THE DATASET CONTAINS   275 GRAPHS                 TIME: 16H 57' 11"     +
+                                                                             +
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++




+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+                                                                             +
+   12.12.94    THE GRAPHS WILL BE COPIED INTO THE DATASET    AQUA    .DATA   +
+                                                                             +
+                       STORAGE-NORM IS      "GRAPHIC8"                       +
+                                                                             +
+     DATE OF THE ALLOCATION:   23. 5.91      LAST SCRATCH:      27. 9.94     +
+                                                                             +
+     THE DATASET CONTAINS   275 GRAPHS                 TIME: 16H 57' 11"     +
+                                                                             +
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
 ENTER TEMPERATURE OF THE ISOTHERM =: ( 0 = END )
 00016 ?
373.15
```

```
 TABLE OF THE PROPERTY-NUMBERS : ( END = 0 )
    |  1      2      3      4      5      6      7      8
 _________________________________________________________
  0 |   P*    P*¯   DP/DT  DT/DP    T*
 10 |   RH    RH¯   RH.SP           P     P.SP    V      Z
 20 |  P1R    P1T    R1P    R1T           CV     CP
 30 |  V.S    SIG           ETA  LAMBDA   PRN
 40 |   A      G      U      H      S     U1R    H1P
 ENTER NUMBER OF THE PROPERTY /   0 / =:
 00012 ?
26
```

```
 ENTER SERIAL NUMBER OF THE (T,FT) MASTER FUNCTION =: ( 0 = END )
 00008 ?
22
```

```
  #     22   GRAPH(1015/ 5000)  X : +DENSITY +KG/M**3 +  F : +     P.T+PASCAL  +
 X1= 0.5550E-03,X(293)=  1190.      ,  0.0000E+00 < X <  1200.      , U=0.10E+03
 F1=  128.1     ,F(293)= 0.1283E+10 , -0.6100E+08 < F < 0.1300E+10 , U=0.10E+09
 FUNCTION OK =: ( 0 = YES | 9 = NO )
 00012 ?
0
```

```
 ENTER BEGIN, END & INCREMENTUM ( 0 = END ) /  1 ,1000 ,  1 / =: , =: , =:
 00016 ?
1 300 1
```

```
 FUNCTION TO BE SAVED ? ( 0 = NO ) =:
 00024 ?
3
```

```
 ===>  173   GRAPH(  26/ 3731)  X : +DENSITY +KG/M**3 +  F : +  CV.ITH+J/KG/K  +
 X1= 0.5550E-03,X(293)=  1190.      ,  0.0000E+00 < X <  1200.      , U=0.10E+03
 F1=  1428.     ,F(293)=  3412.      ,   1400.     < F <  7000.      , U=0.10E+04
```

```
 TABLE OF THE PROPERTY-NUMBERS : ( END = 0 )
    |  1      2      3      4      5      6      7      8
--------------------------------------------------------------
  0 |   P*     P*"   DP/DT  DT/DP    T*
 10 |   RH     RH"   RH.SP          P     P.SP    V      Z
 20 |  P1R    P1T    R1P    R1T           CV     CP
 30 |  V.S    SIG           ETA  LAMBDA   PRN
 40 |   A      G      U      H      S     U1R    H1P
 ENTER NUMBER OF THE PROPERTY /   0 / =:
 00012 ?
0
```

```
 ENTER TEMPERATURE OF THE ISOTHERM =: ( 0 = END )
 00016 ?
0
```

KAISOK

KAISOK calculates the water properties as density-temperature functions along isochores.
KAISOK uses, similarly to KAISOT :

- HYDOR for the calculation of the (ρ,T)-properties,
- the AQUA-routines SIGNAT, NAPSZM, MUSTER
- the SERVUS-routines SERDIO, SDEXG8, FENST8, SDING8
- the dataset ***userid.AQUAT.DATA*** and
- the procedure **userid.AQUA.CLIST(KAISOK)** (s. Figure 35).

```
ALLOC DA('userid.AQUAT.DATA') F(FT15F001) SHR REU
ALLOC DA('userid.AQUAT.DATA') F(FT16F001) SHR REU
CALL 'INR105.AQUA.LOAD(KAISOK)'
FREE DA('userid.AQUAT.DATA')
FREE DA('userid.AQUAT.DATA')
```

Figure 36. The procedure userid.AQUA.CLIST(KAISOK)

A dialogue with KAISOK runs in the same way, as one with KAISOT.

```
 READY
ex aqua(kaisok)
```

etc.

7.4 IVA-routines

A set of the following routines - completed with all the other AQUA-routines needed by them - are deposited in the dataset ***INR105.IVHZO.FORT***.

WASSER

WASSER calculates thermophysical properties of the liquid water as functions ot the temperature and pressure for the codesystem IVA3. WASSER calculates only in the stable and metastable states of the liquid, so the supplied P must be higher as the spinodal-pressure of the liquid. In calling WASSER

```
        CALL WASSER(T,P
                ,RW,HW,SW,CPW,VSW,ETW,XLMBW,PRNW,HW1P,RW1T,RW1P,SRFTS
                ,LCATAW)
```

1. T, the temperature and
2. P, the pressure

should be supplied. WASSER then returns

3. RW, the density,
4. HW, the specific enthalpy,
5. SW, the specific entropy,
6. CPW, the heat capacity at constant pressure,
7. VSW, the velocity of sound,
8. ETW, the viscosity,
9. XLMBW, the thermal conductivity,
10. PRND, the Prandtl number of the water ,
11. HW1P, the pressure derivative of the enthalpy, $\partial H/\partial P$,
12. RW1T, the temperature derivative of the density, $\partial \rho/\partial T$,
13. RW1P, the pressure derivative of the density, $\partial \rho/\partial P$ and
14. SRFTS, the surface tension.

All of the above properties are in SI-units.

LCATAW, a logical variable will be set .true. if WASSER fails to find the density `RF(P,T)`.

DAMPF

DAMPF calculates thermophysical properties of the vaporized water as functions ot the temperature and pressure for the codesystem IVA3. DAMPF calculates only in the stable and metastable vapor states, so the supplied P must be lower as the spinodal-pressure of the vapor. DAMPF

```
        CALL DAMPF(T,P
                ,RGAS,WGM,RD,HD,SD,CPD,VSD,ETD,XLMBD,PRND,HD1P,RD1T,RD1P
                ,LCATAD)
```

needs initial values for

1. T, the temperature and
2. P, the pressure

and it returns the following properties :

3. RGAS, the gas-law constant of the water,
4. WGM, the molal mass,
5. RD, the density,
6. HD, the specific enthalpy,
7. SD, the specific entropy,
8. CPD, the heat capacity at constant pressure,
9. VSD, the velocity of sound,
10. ETD, the viscosity,
11. XLMBD, the thermal conductivity,
12. PRND, the Prandtl number of the steam
13. HD1P, the pressure derivative of the enthalpy, $\partial H / \partial P$,
14. RD1T, the temperature derivative of the density, $\partial \rho / \partial T$ and
15. RD1P, the pressure derivative of the density, $\partial \rho / \partial P$.

All of the above properties - as in the case of WASSER - are in SI-units.

LCATAD will be again set .true. if DAMPF fails to find the density `RD(P,T)`.

RDRDP

RDRDP is an abridged version of DAMPF, only to find the density corresponding to the pressure P and the temperature T in the vapor state for the code IVA3.

```
        CALL RDRDP(T,P, RD,RD1P, LCATAD)
```

For a given T and P RDRDP returns RD, the density and RD1P, the pressure derivative of the density ($\partial \rho / \partial P$) of the steam.

SAET01

```
        CALL SAET01(T,P
                ,VW,VD,HW,HD,HWD,SW,SD,TS1P,CPW,CPD,ETW,ETD,XLMBW,XLMBD,SRFTS
                ,KIND,LCATAS)
```

SAET01 calculates saturation properties of the water for IVA3 as function of the pressure P (if KIND is 1) or the temperature T (if KIND equals 0).

SAET01 returns - besides of T or P - the following thermophysical properties :

- VW and VD are the specific volume of the liquid resp. of the vapor,
- HW and HD are the corresponding specific enthalpies,
- SW and SD are the specific entropies in the liquid and vapor states,
- CPW and CPD are the corresponding heat capacities at constant pressure,
- ETW and ETD are the viscosities of the liquid resp. of the vapor,
- XLMBW and XLMBD are the corresponding thermal conductivities,
- SRFTS is the surface tension (of the liquid),
- HWD = HD - HW and

- TS1P = $1 / \partial P / \partial T$.

All the properties are again in SI-units.

LCATAS will be set .true. if SAET01 fails to find saturated properties (i. e. if TS is supercritical).

Note: There exists also an extended version of the above routine, SAETEX, created as a testing facility for the routine SAET01. Besides of the properties listed above, SAETEX calculates also the pressure derivatives dP^x/dT, $\partial P/\partial T$ and $\partial P/\partial \rho$, the constant volume heat capacities C_V and the sonic velocities.

VD1

VD1 calculates approximated specific volumes for the saturated vapor, VD in kg/m^3-s at the temperature T for the code IVA3.

```
      CALL VD1(T, VD, DUMMY)
```

For a complete list of available water property procedures see Appendix M, "TSO-procedures, calculating thermal properties of the water" on page 153.

8.0 References

/1/ V. V. Sychev, Character of the Change of Thermodynamic Quantities on Boundary Curves
High Temperature, Vol. **5**, No. 6, p. 1022, 1967.

/2/ Kh I. Amirkhanov, B. G. Alibekov, B. A. Mursalov, and G. V. Stepanov, Calculation of the Derivatives of Thermal and Caloric Quantities on a Saturation Line
High Temperature, Vol. **10**, No. 3, p. 475, 1972.

/3/ V. P. Skripov, Metatstable Liquids, English translation
John Wiley and sons, New York, 1974.

/4/ J. V. Sengers, Thermal Conductivity of Steam in the Critical Region, P. 822 in
Proceedindgs of the Seventh Symposium on Thermophysical Properties, ASME, 1977.

/5/ K.Thurnay, Thermophysical Properties of Sodium in the Liquid and Gaseous States,
KFK 2863, Februar 1981.

/6/ L. A. Weber, Measurements of the heat capacities C_V of dense gaseous and liquid nitrogen and nitrogen trifluoride.
J. Chem. Thermodynamics, Vol. **13**, p. 389-403, 1981.

/7/ A. M. Sirota, V. I. Latunin, N. E. Nikolaeva, An Experimental Investigation of the Thermal Conductivity Maxima of Water in the Critical Region
Thermal Engineering Vol. **28**, No. 4, p. 246, 1981.

/8/ K.Thurnay, Evaluation of Thermophysical Properties of Sodium as Surfaces of the Temperature and the Density,
Nucl. Sci. Eng., **82**. p. 181, 1982.

/9/ C. A. Meyer, R. B. McClintock, G. J. Silvestri, R. C. Spencer, Jr., ASME steam tables : thermodynamic and transport properties of steam comprising tables and charts for steam and water, 5. ed.
American Soc. of Mechanical Engineers, New York, 1983.

/10/ L. Haar, J. S. Gallagher, G. S. Kell, NBS/NRC steam tables : thermodynamic and transport properties and computer programs for vapor and liquid states of water in SI units
Hemisphere Publ. Co., Washington, 1984.

/11/ J. V. Sengers and B. Kamgar Parsi, Representative Equations for the Viscosity of Water Substance
J. Phys. Chem. Ref. Data, Vol. **13**, No. 1, p. 185, 1984.

/12/ H. J. White, Jr., Activities of the International Association for the Properties of Steam Between 1979 and 1984, pp. 31 - 34 in
V. V. Sytchev (ed.) The properties of steam
Proc. of the 10th internat. conf., Moscow, USSR, 3 - 7 Sept. 1984.
Moscow : Mir Publishers, London : Plenum Press, 1984.

/13/ L. Haar and J. S. Gallagher, Thermodynamic Values Near the Critical Point of Water, pp. 167 - 179 in
V. V. Sytchev (ed.) The properties of steam
Proc. of the 10th internat. conf., Moscow, USSR, 3 - 7 Sept. 1984.
Moscow : Mir Publishers, London : Plenum Press, 1984.

/14/ W. Wagner and A. Saul, Correlation Equations for the Vapor Pressure and for the Orthobaric Densities of Water Substance, pp. 199 - 209 in
V. V. Sytchev (ed.) The properties of steam
Proc. of the 10th internat. conf., Moscow, USSR, 3 - 7 Sept. 1984.
Moscow : Mir Publishers, London : Plenum Press, 1984.

/15/ V. P. Skripov, Phase-Change Behaviour of Water in Nonequilibrium Processes, pp. 227 - 251 in
V. V. Sytchev (ed.) The properties of steam
Proc. of the 10th internat. conf., Moscow, USSR, 3 - 7 Sept. 1984.
Moscow : Mir Publishers, London : Plenum Press, 1984.

/16/ J. V. Sengers, Transport Properties of Fluids Near Critical Points, pp. 363 - 389 in
V. V. Sytchev (ed.) The properties of steam
Proc. of the 10th internat. conf., Moscow, USSR, 3 - 7 Sept. 1984.
Moscow : Mir Publishers, London : Plenum Press, 1984.

/17/ V. V. Sychev, V. A. Rabinovich, and Yu. E. Sheludyak, Using Differential Thermodynamic Equations at the Critical Point of Pure Materials
High Temperature, Vol. **22**, No. 6, p. 829, 1985.

/18/ P. W. Atkins, Physical chemistry, 3. ed.
Oxford University Press, 1986.

/19/ K. Thurnay, SERVUS. Ein System von Dateien und Prozeduren zur Speicherung und Präsentation von Rechenergebnissen, die Funktionen (einer Variablen) sind. Version V, April 1989.
Kfk 4587, Juli 1989.

/20/ K. Thurnay, SODIUM. A Code for Calculating Thermophysical Properties of the Sodium in the Liquid and Gaseous States.
KfK 4609, September 1989.

/21/ N. I. Kolev, IVA3 : Computer Code For Modeling of Transient Three Dimensional Three Phase Flow in Complicated Geometry
Kfk 4950, Dezember 1991.

/22/ V. P. Skripov, Metastable States
J. Non-Equilib. Thermodyn., Vol. **17**, No. 3, p. 193 - 206, 1992.

/23/ C. A. Meyer, R. B. McClintock, G. J. Silvestri, R. C. Spencer, Jr., ASME Steam Tables : Thermodynamic and Transport Properties of Steam Comprising Tables and Charts For Steam and Water, 6. ed.
New York, NY : American Soc. of Mechanical Engineers, 1993.

Appendix A. Some thermal relations

A.1 The derivatives of the free energy

In developing the thermal properties of a substance as functions of the **temperature and density** the appropriate energy property to begin with is the free energy :

$$A \equiv U - S \cdot T \qquad [A.1]$$

also known as Helmholtz-function or total work.

Since it holds (s. e. g. /18/)

$$dU = T \cdot dS - P \cdot dV \qquad [A.2]$$

one has for A the following (ρ,T)-differential :

$$dA(\rho,T) = dU - S\,dT - T\,dS = -S\,dT + \frac{P}{\rho^2}\,d\rho \quad . \qquad [A.3]$$

The differential [A.3] gives following relations for entropy and pressure :

$$S(\rho,T) = -\frac{\partial A}{\partial T} \quad , \qquad [A.4]$$

$$P(\rho,T) = \frac{\partial A}{\partial \rho}\,\rho^2 \quad . \qquad [A.5]$$

A.2 The enthalpy and its derivatives

For the enthalpy one can also apply a (dρ,dT) - expression :

$$dH(\rho,T) = \frac{\partial H}{\partial T}\,dT + \frac{\partial H}{\partial \rho}\,d\rho \quad . \qquad [A.6]$$

The T-derivative in the Eq. [A.6] can be calculeted from the equation

$$H \equiv U + \frac{P}{\rho} \qquad [A.7]$$

as

$$\left.\frac{\partial H}{\partial T}\right|_{\rho} = C_V + \frac{1}{\rho}\,\frac{\partial P}{\partial T} \qquad [A.8]$$

and the ρ-derivative from the equation

$$H \equiv A + \frac{P}{\rho} + S \cdot T \qquad [A.9]$$

transformed with the relations [A.4] and [A.5] properly. The result is

$$\frac{\partial H}{\partial \rho}\Big|_T = \frac{1}{\rho}\left(\frac{\partial P}{\partial \rho} - \frac{T}{\rho}\,\frac{\partial P}{\partial T}\right) \quad . \qquad [A.10]$$

Setting the pressure-differential

$$dP(\rho,T) = \frac{\partial P}{\partial T}\,dT + \frac{\partial P}{\partial \rho}\,d\rho$$

to zero gives the important relation

$$\frac{\partial P/\partial T}{\partial P/\partial \rho} = -\frac{\partial \rho}{\partial T}\Big|_P \quad . \qquad [A.11]$$

Using this relation and the reduced temperature-derivative of the density

$$\rho_T^0 \equiv \frac{T}{\rho}\,\frac{\partial \rho}{\partial T}\Big|_P \qquad [A.12]$$

allows to describe the density derivative [A.10] in a more simple form

$$\frac{\partial H}{\partial \rho}\Big|_T = \frac{1}{\rho}\,\frac{\partial P}{\partial \rho}\,(1 + \rho_T^0) \quad . \qquad [A.13]$$

To develop the derivatives of the **(P,T)-description** the differential [A.6] is to be transformed accordingly. Using a (P,T)-differential for the density results in

$$\begin{aligned} dH &= \frac{\partial H}{\partial T}\,dT + \frac{\partial H}{\partial \rho}\left(\frac{\partial \rho}{\partial T}\,dT + \frac{\partial \rho}{\partial P}\,dP\right) = \\ &= \left(\frac{\partial H}{\partial T}\Big|_\rho + \frac{\partial H}{\partial \rho}\Big|_T\,\frac{\partial \rho}{\partial T}\right)dT + \frac{\partial H}{\partial \rho}\Big|_T\,\frac{\partial \rho}{\partial P}\,dP \quad . \end{aligned} \qquad [A.14]$$

Eq. [A.14] combined with Eqs. [A.8], [A.11] and [A.13] results in the following descriptions for the (P,T)-derivatives of the enthalpy :

$$\begin{aligned} \frac{\partial H}{\partial T}\Big|_P &\equiv C_P(P,T) = \frac{\partial H}{\partial T}\Big|_\rho + \frac{\partial H}{\partial \rho}\Big|_T\,\frac{\partial \rho}{\partial T}\Big|_P = \\ &= C_V(\rho,T) + \frac{T}{\partial P/\partial \rho}\left(\frac{1}{\rho}\,\frac{\partial P}{\partial T}\right)^2 \quad , \end{aligned} \qquad [A.15]$$

respectively

$$\frac{\partial H}{\partial P}\Big|_T = \frac{\partial H}{\partial \rho}\Big|_T\,\frac{\partial \rho}{\partial P}\Big|_T = \frac{1}{\rho}\,(1 + \rho_T^0) \quad . \qquad [A.16]$$

This last derivative is also known as the 'isothermal factor'

$$\frac{\partial H}{\partial P}\Big|_T = V - T\cdot\frac{\partial V}{\partial T} \quad . \qquad [A.17]$$

A.3 The sonic velocity

is defined as

$$v_s^2 = \frac{\partial P}{\partial \rho}\bigg|_S \quad . \qquad [A.18]$$

To develop the derivative in the equation above one needs the following differential

$$dS(\rho,P) = \frac{\partial S}{\partial T}\left(\frac{\partial T}{\partial \rho}\,d\rho + \frac{\partial T}{\partial P}\,dP\right) + \frac{\partial S}{\partial \rho}\,d\rho \quad . \qquad [A.19]$$

Eq. [A.19] can be transformed to

$$dS(\rho,P) = \frac{\partial S/\partial T}{\partial P/\partial T}\,dP + \left(\frac{\partial S}{\partial \rho} + \frac{\partial S/\partial T}{\partial \rho/\partial T}\right)d\rho \qquad [A.20]$$

Setting dS = 0 in the above equation and using Eq. [A.11] gives

$$\frac{\partial P}{\partial \rho}\bigg|_S = -\frac{\left(\dfrac{\partial S}{\partial \rho} + \dfrac{\partial S/\partial T}{\partial \rho/\partial T}\right)}{\dfrac{\partial S/\partial T}{\partial P/\partial T}} = \frac{\partial P}{\partial \rho} - \frac{\partial P}{\partial T}\,\frac{\partial S/\partial \rho}{\partial S/\partial T} \quad . \qquad [A.21]$$

Differentiating the equations Eq. [A.1] respectively Eq. [A.4] result in the entropy-derivatives

$$\frac{\partial S}{\partial T} = \frac{C_V}{T} \quad , \qquad \frac{\partial S}{\partial \rho} = -\frac{1}{\rho^2}\,\frac{\partial P}{\partial T} \quad . \qquad [A.22]$$

Setting the above derivatives into the Eq. [A.21] and comparing the result with the Eq. [A.15] ends in

$$\frac{\partial P}{\partial \rho}\bigg|_S = \frac{\partial P}{\partial \rho} + \frac{T}{C_V}\left(\frac{\partial P/\partial T}{\rho}\right)^2 = \frac{\partial P}{\partial \rho}\,\frac{C_P}{C_V} \quad . \qquad [A.23]$$

Related to the sonic velocity are the adiabatic resp. the isothermal compressibilities :

$$\kappa_S = \frac{1}{\rho\cdot\partial P/\partial \rho\big|_S} = \frac{1}{\rho\cdot v_s^2} \quad , \qquad [A.24]$$

$$\kappa_T = \frac{1}{\rho\cdot\partial P/\partial \rho} = \frac{C_P}{C_V}\cdot\kappa_S \quad . \qquad [A.25]$$

A.4 The Joule Thomson coefficient

This property is defined as

$$\text{Joule Thomson coefficient} \quad \equiv \quad \left.\frac{\partial T}{\partial P}\right|_H \qquad [A.26]$$

Using the restraint

$$dH = \frac{\partial H}{\partial P}\,dP + \frac{\partial H}{\partial T}\,dT = 0$$

and the result of Eq. [A.16] one has

$$\left.\frac{\partial T}{\partial P}\right|_H = -\frac{\dfrac{\partial H}{\partial P}}{\dfrac{\partial H}{\partial T}} = -\frac{\dfrac{\partial H}{\partial P}}{C_P} = -\frac{1 + \rho_T^0}{\rho \cdot C_P} \quad . \qquad [A.27]$$

Appendix B. Derivatives of some saturated properties

B.1 Derivatives of the vapor pressure

Since the derivative of w in the Eq. [5.2] can be expressed as

$$\frac{dw}{dT} = \frac{-0.5}{T_c}\left(1 - \frac{1}{T_c}\right)^{-0.5} = \frac{-0.5}{w \bullet T_c} \qquad [B.1]$$

it holds for the first T-derivative of the vapor pressure

$$Q_1 \equiv \frac{T}{P^\times}\frac{dP^\times}{dT} = T\,\frac{d\ln P^\times}{dT} =$$

$$= -\frac{T_c}{T} \bullet \sum_{j=1}^{13} A_j \bullet w^{j-1} + \left(\frac{T_c}{T} - 1\right)\sum_{j=1}^{8} T \bullet \frac{dw}{dT} \bullet (j-1) \bullet A_j \bullet w^{j-2} = \qquad [B.2]$$

$$= -\sum_{j=1}^{8} A_j \bullet w^{j-1}\left[\frac{T_c}{T} + \frac{j-1}{2}\right] \quad .$$

The derivative of Q_1 is

$$Q_2 \equiv T\,\frac{dQ_1}{dT} =$$

$$= \sum_{j=1}^{8} A_j \bullet w^{j-1}\left[\frac{T_c}{T} + \frac{j-1}{2}\left(1 + \frac{j+1}{2}\,\frac{T}{T_c - T}\right)\right] \quad . \qquad [B.3]$$

On the other hand one can develop a pressure relation for Q_2 :

$$Q_2 \equiv T\,\frac{d}{dT}\left[\frac{T}{P^\times}\,\frac{dP^\times}{dT}\right] =$$

$$\frac{T}{P^\times}\,\frac{dP^\times}{dT} + \frac{T^2}{P^\times}\,\frac{d^2P^\times}{dT^2} - \left[\frac{T}{P^\times}\,\frac{dP^\times}{dT}\right]^2 = \qquad [B.4]$$

$$= \frac{T^2}{P^\times}\,\frac{d^2P^\times}{dT^2} + Q_1\,(1 - Q_1) \quad .$$

Transforming Eq. [B.2] and [B.4] result in the following equations for the temperature derivatives of the vapor pressure :

$$\frac{dP^\times}{dT} = \frac{P^\times}{T}\,Q_1 \qquad \text{and} \qquad [B.5]$$

$$\frac{d^2P^\times}{dT^2} = \frac{P^\times}{T^2}\,[\,Q_2 - Q_1\,(1 - Q_1)\,] \quad . \qquad [B.6]$$

To ease the descriptions of the Q-s the coefficients

$$B_j \equiv \frac{j-1}{2} A_j \qquad \text{and} \tag{B.7}$$

$$C_j \equiv \frac{j+1}{2} B_j \tag{B.8}$$

will be introduced, resulting in the Q-equations

$$Q_1 = -\frac{T_c}{T} \sum_{j=1}^{8} A_j \cdot w^{j-1} - \sum_{j=1}^{8} B_j \cdot w^{j-1} \tag{B.9}$$

and

$$Q_2 = \frac{T_c}{T} \sum_{j=1}^{8} A_j \cdot w^{j-1} + \sum_{j=1}^{8} B_j \cdot w^{j-1} + \frac{T}{T_c - T} \sum_{j=1}^{8} C_j \cdot w^{j-1} \quad . \tag{B.10}$$

B.2 Derivatives of the saturated densities

For

the cold water one can develop from Eq. [5.9] with Eq. [B.1] the derivative

$$\frac{d\rho_L}{dT} = -\frac{0.5 \cdot \rho_c}{T_c} \sum_{j=2}^{12} (j-1) \cdot A_j \cdot w^{j-3} \quad . \tag{B.11}$$

Correspondingly the second derivative is here

$$\frac{d^2\rho_L}{dT^2} = \left[\frac{0.5}{T_c} \right]^2 \cdot \rho_c \sum_{j=2}^{12} (j-1) \cdot (j-3) \cdot A_j \cdot w^{j-5} \quad . \tag{B.12}$$

To develop the density derivatives for

the cold steam is a bit more complicated, since here is not the density, but the factor of reality given by a polynomial (s. Eq. [5.10]).
From the gas law

$$P^{x} = \rho_V \cdot R_{gas} \cdot T \cdot Z_V$$

one has the following relations between the first and second derivatives :

$$\frac{T}{P^{x}} \frac{dP^{x}}{dT} = 1 + \frac{T}{\rho_V} \frac{d\rho_V}{dT} + \frac{T}{Z_V} \frac{dZ_V}{dT} \tag{B.13}$$

and

$$\frac{T^2}{P^\times}\frac{d^2P^\times}{dT^2} - 2\cdot\frac{T}{P^\times}\frac{dP^\times}{dT} + 2 =$$
$$= \frac{T^2}{\rho_V}\frac{d^2\rho_V}{dT^2} + 2\cdot\frac{T}{\rho_V}\frac{d\rho_V}{dT}\cdot\frac{T}{Z_V}\frac{dZ_V}{dT} + \frac{T^2}{Z_V}\frac{d^2Z_V}{dT^2} \quad . \qquad [B.14]$$

Since the Z-derivatives are

$$\frac{dZ_V}{dT} = -\frac{0.5}{T_c}\sum_{j=2}^{12}(j-1)\cdot Z_j\cdot w^{j-3} \quad \text{and} \qquad [B.15]$$

$$\frac{d^2Z_V}{dT^2} = \left[\frac{0.5}{T_c}\right]^2\sum_{j=2}^{12}(j-1)\cdot(j-3)\cdot Z_j\cdot w^{j-5} \qquad [B.16]$$

so the density derivatives in the cold steam can be described as

$$\frac{d\rho_V}{dT} = \frac{\rho_V}{T}\left[Q_1 - 1 + \frac{0.5\cdot T}{Z_V\cdot T_c}\sum_{j=2}^{12}(j-1)\cdot Z_j\cdot w^{j-3}\right] \quad , \qquad [B.17]$$

$$\frac{d^2\rho_V}{dT^2} = \frac{\rho_V}{T^2}\left[Q_1^{\,2} - 3\,Q_1 + Q_2 + 2\right] +$$
$$+ \frac{1}{Z_V\cdot T_c}\left[\frac{d\rho_V}{dT}\sum_{j=2}^{12}(j-1)\cdot Z_j\cdot w^{j-3} + \right. \qquad [B.18]$$
$$\left. - 0.25\,\frac{\rho_V}{T_c}\sum_{j=2}^{12}(j-1)\cdot(j-3)\cdot Z_j\cdot w^{j-5}\right] \quad .$$

In the vicinity of the critical point AQUA describes the densities as

$$\rho(T) = \rho_c\cdot\omega(\tau)$$
$$\omega(\tau) \equiv 1 + B\tau^\beta + \tau\left[U_1 + \tau\left(U_2 + \tau^2\,U_4\right)\right] \qquad [B.19]$$

with $B = 2.03$ for the liquid and $B = -2.03$ for the vapor. The τ-derivative is

$$\frac{d\tau}{dT} = -\frac{\rho_c}{T_c} \qquad [B.20]$$

so the first density derivatives will be

$$\frac{d\rho}{dT} = -\frac{\rho_c}{T_c}\frac{d\omega}{d\tau} = -\frac{\rho_c\cdot g}{T_c - T} \qquad [B.21]$$

the function g being

$$g \equiv \tau \frac{d\omega}{d\tau} = B\tau^{\beta}\beta + \tau\left[\, U_1 + \tau\left(2\,U_2 + \tau^2\, 4\,U_4\right)\right] \quad . \qquad [B.22]$$

The second density-derivatives one can develop as

$$\frac{d^2\rho}{dT^2} = \frac{\rho_c}{T_c^{\,2}}\,\frac{d^2\omega}{d\tau^2} = \frac{\rho_c}{(\tau \cdot T_c)^2}\left[\tau \frac{dg}{d\tau} - g\right] = \frac{\rho_c \cdot h}{(T_c - T)^2} \qquad [B.23]$$

with

$$h \equiv \tau \frac{dg}{d\tau} - g = B\tau^{\beta}\beta(\beta - 1) + \tau^2\left(2\,U_2 + \tau^2\, 12\,U_4\right) \quad . \qquad [B.24]$$

Requesting smooth transitions at the switching point for the density and its first and second derivatives leads to the following system of equations for the coefficients U :

$$\begin{aligned} U_4 &= \frac{2(o - g) + h - (1 - \beta)(2 - \beta)b}{6\tau^4} \\ U_2 &= -\,\frac{4(o - g) + h - (1 - \beta)(4 - \beta)b}{2\tau^2} \\ U_1 &= \frac{8o - 5g + h - (2 - \beta)(4 - \beta)b}{3\tau} \end{aligned} \qquad [B.25]$$

with the abbreviations

$$b = B\tau^{\beta} \qquad \text{and} \qquad o = \frac{\rho - \rho_c}{\rho_c} \quad . \qquad [B.26]$$

Appendix C. Adjusting the saturated states

To be in thermal equilibrium the pressures and the Gibbs-energies of the saturated liquid and the saturated vapor must be in accordance with each other :

$$G_L = G_V \qquad \text{and} \qquad P_L = P_V \ . \tag{C.1}$$

For the Gibbs-function

$$G \equiv A + \frac{P}{\rho} \tag{C.2}$$

along an isotherm it holds :

$$dG \,|_T = d\rho \cdot \frac{\partial G}{\partial \rho} \tag{C.3}$$

with

$$\frac{\partial G}{\partial \rho} = \frac{\partial A}{\partial \rho} - \frac{P}{\rho^2} + \frac{1}{\rho}\,\frac{\partial P}{\partial \rho} = \frac{1}{\rho}\,\frac{\partial P}{\partial \rho} \quad . \tag{C.4}$$

Suppose the liquid and the vapor state differ from each other by $\Delta P = P_L - P_V$ and $\Delta G = G_L - G_V$. To get rid of these differences one needs small changes in the liquid density, x and in the vapor density, y.

The density changes, x and y can be calculated as follows : The pressure difference can be transformed to

$$\Delta P = P_L - P^x + P^x - P_V = -\Delta P_L + \Delta P_V = -\,x\,\frac{\partial P}{\partial \rho}_L + y\,\frac{\partial P}{\partial \rho}_V \tag{C.5}$$

with $\Delta P_L = P^x - P_L$ and $\Delta P_V = P^x - P_V$. Correspondingly one has for the G-s

$$\Delta G = G_L - G^x + G^x - G_V = -\Delta G_L + \Delta G_V = -\,\frac{x}{\rho_L}\,\frac{\partial P}{\partial \rho}_L + \frac{y}{\rho_V}\,\frac{\partial P}{\partial \rho}_V \ . \tag{C.6}$$

The Eqs. [C.5]- [C.6] can be solved for x and y :

$$\begin{aligned} x &= \frac{\rho_L}{\rho_L - \rho_V} \cdot \frac{\Delta G \cdot \rho_V - \Delta P}{\partial P / \partial \rho_L} \ , \\ y &= \frac{\rho_V}{\rho_L - \rho_V} \cdot \frac{\Delta G \cdot \rho_L - \Delta P}{\partial P / \partial \rho_V} \ . \end{aligned} \tag{C.7}$$

Using the properties in their reduced forms leads to :

$$\begin{aligned} x &= \frac{\rho_L}{\rho_L - \rho_V} \cdot \frac{\rho_V (\Delta G^0 + Z_V) - \rho_L \cdot Z_L}{\mathrm{PR}^0{}_L} \ , \\ y &= \frac{\rho_V}{\rho_L - \rho_V} \cdot \frac{\rho_L (\Delta G^0 - Z_L) + \rho_V \cdot Z_V}{\mathrm{PR}^0{}_V} \ . \end{aligned} \tag{C.8}$$

Appendix D. Module trees

D.1 Interactive routines

WATER

```
WATER
              ONLYT(T,IECCH)
              SUBCRI(T)
              DLFIND(T,Q,DX,IECCH)
                             SUBCRI(T)
                             DILE(Y)
                             DIVE(Y)
                             DINLI(T,Q,FX,X,LT,IECCH)
                             BASE(T,DX,IECCH)
                             RESID(T,DX)
              DVFIND(T,Q,DX,IECCH)
                             SUBCRI(T)
                             DIVE(Y)
                             DINVA(T,Q,FX,X,LT,IECCH)
                             BASE(T,DX,IECCH)
                             RESID(T,DX)
              DGFIND(T,Q,DX,IECCH)
                             DINSU(T,Q,FX,X,IT,IECCH)
                             BASE(T,DX,IECCH)
                             RESID(T,DX)
              HYDOR(T,RH,IECCH)
```

Figure 37. Module tree for the procedure WATER

KAISOT

```
KAISOT
              SERDIO(IRE,NORMR,IWR,NORMW,MODUS,KPRI)
              SIGNAT(KLASSE,NUMMER,NAMF,MASF,NEIG)
                             NAPSZM(NR)
              MUSTER(IA,IE,II,NAMR,MASR,NAMG,MASG,*200)
                             SDEXG8(KLASSE,NUMMER,MP
                                   ,T,TMI,TMA,TED,NAMT,MAST
                                   ,F,FMI,FMA,FED,NAMF,MASF,LOS)
              HYDOR(T,RHI,IECCH)
              FENST8(MP,RH,XF,RMI,RMA,RED,LAX,KAXE,MASR)
              FENST8(MP,F,XF,FMI,FMA,FED,LAX,KAXE,MASF)
              SDING8(KLASSE,NUMMER,MP
                    ,RH,RMI,RMA,RED,NAMR,MASR
                    ,F,FMI,FMA,FED,NAMF,MASF,MODUS)
```

Figure 38. Module tree for the procedure KAISOT

KAISOK

```
KAISOK
              SERDIO(IRE,NORMR,IWR,NORMW,MODUS,KPRI)
              SIGNAT(KLASSE,NUMMER,NAMF,MASF,NEIG)
                             NAPSZM(NR)
              MUSTER(IA,IE,II,NAMR,MASR,NAMG,MASG,*200)
                             SDEXG8(KLASSE,NUMMER,MP
                                   ,T,TMI,TMA,TED,NAMT,MAST
                                   ,F,FMI,FMA,FED,NAMF,MASF,LOS)
              HYDOR(T,RHI,IECCH)
              FENST8(MP,RH,XF,RMI,RMA,RED,LAX,KAXE,MASR)
              FENST8(MP,F,XF,FMI,FMA,FED,LAX,KAXE,MASF)
              SDING8(KLASSE,NUMMER,MP
                    ,RH,RMI,RMA,RED,NAMR,MASR
                    ,F,FMI,FMA,FED,NAMF,MASF,MODUS)
```

Figure 39. Module tree for the procedure KAISOK

D.2 IVA3-ROUTINES

WASSER

```
WASSER(T,P,RW,HW,SW,CPW,VSW
      ,ETW,XLMBW,PRNW,HW1P,RW1T,RW1P,SRFTS, LCATAW)
              ONLYT(T,IECCH)
              DGFIND(T,Q,DX,IECCH)
                             DINSU(T,Q,FX,X,IT,IECCH)
                             BASE(T,DX,IECCH)
                             RESID(T,DX)
              DLFIND(T,Q,DX,IECCH)
                             SUBCRI(T)
                             DILE(Y)
                             DIVE(Y)
                             DINLI(T,Q,FX,X,LT,IECCH)
                             BASE(T,DX,IECCH)
                             RESID(T,DX)
              REDUK(T,DX,IECCH)
                             BASE(T,DX,IECCH)
                             RESID(T,DX)
                             TRANSP(T,RH,P1R0,P1T0,ETA,XLMB)
              SRFTS=SIGMA(T)
```

Figure 40. Module tree for the procedure WASSER

DAMPF

```
DAMPF(T,P,RGAS,WGM,RD,HD,SD,CPD,VSD
     ,ETD,XLMBD,PRND,HD1P,RD1T,RD1P, LCATAD)
              ONLYT(T,IECCH)
              DGFIND(T,Q,DX,IECCH)
                             DINSU(T,Q,FX,X,IT,IECCH)
                             BASE(T,DX,IECCH)
                             RESID(T,DX)
              DVFIND(T,Q,DX,IECCH)
                             SUBCRI(T)
                             DIVE(Y)
                             DINVA(T,Q,FX,X,LT,IECCH)
                             BASE(T,DX,IECCH)
                             RESID(T,DX)
              REDUK(T,DX,IECCH)
                             BASE(T,DX,IECCH)
                             RESID(T,DX)
                             TRANSP(T,RH,P1R0,P1T0,ETA,XLMB)
```

Figure 41. Module tree for the procedure DAMPF

RDRDP

```
RDRDP(T,P,RD,RD1P,LCATAD)
              ONLYT(T,IECCH)
              DGFIND(T,Q,DX,IECCH)
                             DINSU(T,Q,FX,X,IT,IECCH)
                             BASE(T,DX,IECCH)
                             RESID(T,DX)
              DVFIND(T,Q,DX,IECCH)
                             SUBCRI(T)
                             DINVA(T,Q,FX,X,LT,IECCH)
                             BASE(T,DX,IECCH)
                             RESID(T,DX)
              REDUK(T,DX,IECCH)
                             BASE(T,DX,IECCH)
                             RESID(T,DX)
                             TRANSP(T,RH,P1R0,P1T0,ETA,XLMB)
```

Figure 42. Module tree for the procedure RDRDP

SAET01

```
SAET01(T,P,VW,VD,HW,HD,HWD,SW,SD,TS1P,CPW,CPD
      ,ETW,ETD,XLMBW,XLMBD,SRFTS, LCATAS)
              TCORR(T,Q,DLK,DVK,IECCH)
                             TS1(P,T,TS1P,IECCH)
                             ONLYT(T,IECCH)
                             SUBCRI(T)
                             GPCORR(T,Q,DLK,DVK)
              ONLYT(T,IECCH)
              SUBCRI(T)
              GPCORR(T,Q,DLK,DVK)
              TWOPHA(T)
              REDUK(T,DX,IECCH)
                             BASE(T,DX,IECCH)
                             RESID(T,DX)
                             TRANSP(T,RH,P1R0,P1T0,ETA,XLMB)
              SRFTS=SIGMA(T)
```

Figure 43. Module tree for the procedure SAET01

D.3 SUB-ROUTINES

HYDOR

```
HYDOR(T,RH,IECCH)
              ONLYT(T,IECCH)
              SRFTS=SIGMA(T)
              SUBCRI(T)
              REDUKE(T,D,IECCH)
                              BASE(T,D,IECCH)
                              RESID(T,D)
                              TWOPHA(T)
                              TRANSP(T,RH,P1R0,P1T0,ETA,XLMB)
```

Figure 44. Module tree for the procedure HYDOR

SUBCRI

```
SUBCRI(T)
              SSP(T)
              SSRHO(T)
              SPIRHO(T)
              SPIP(T)
```

Figure 45. Module tree for the procedure SUBCRI

TWOPHA

```
TWOPHA(T)
              SSCEV(T,CVL,CVV)
              SSPER(T)
              SSPET(T,PTL,PTV)
              SSPET(T,PTL,PTV)
              SSGIS(T,SL,GL)
              SSETA(T,ETL,ETV)
              SSLAM(T,XLL,XLV)
```

Figure 46. Module tree for the procedure TWOPHA

GPCORR

```
GPCORR(T,Q,DLK,DVK)
              BASE(T,DLK,IECCH)
              RESID(T,DLK)
              BASE(T,DVK,IECCH)
              RESID(T,DVK)
```

Figure 47. Module tree for the procedure GPCORR

DINSU

```
DINSU(T,Q,FX,X,IT,IECCH)
              QUADO(FX,X,Q,DX,JECCH)
                         MONIKA(FX,X,JECCH)
              SORBET(KI,FX,X,Q,DX)
              BASE(T,DX,IECCH)
              RESID(T,DX)
```

Figure 48. Module tree for the function DINSU

DINLI

```
DINLI(T,Q,FX,X,IT,IECCH)
              QUADO(FX,X,Q,DX,JECCH)
                         MONIKA(FX,X,JECCH)
              SORBET(KI,FX,X,Q,DX)
              BASE(T,DX,IECCH)
              RESID(T,DX)
```

Figure 49. Module tree for the function DINLI

DINVA

```
DINVA(T,Q,FX,X,IT,IECCH)
              QUADO(FX,X,Q,DX,JECCH)
                         MONIKA(FX,X,JECCH)
              SORBET(KI,FX,X,Q,DX)
              BASE(T,DX,IECCH)
              RESID(T,DX)
```

Figure 50. Module tree for the function DINVA

Appendix E. Common blocks

A substantial part of the data transfer between the moduls of the AQUA code takes place via common blocks. Present appendix serves to describe the most important ones of theese common blocks in an abridged form.

/SATIF/ PS,TPS1T,TTPS2T,DL,DV,RL1T0,RV1T0,RHSL,PSL,RHSV,PSV,SL,GL, PL1R0,PL1T0,CVL,ETL,XLL,PV1R0,PV1T0,CVV,ETV,XLV
carries thermal properties in the saturated and spinodal states, namely

PS, TPS1T, TTPS2T
the vapor pressure and its first and second derivatives,
SL, GL
the entropy and the Gibbs-function of the saturated liquid,
DL, DV, RL1T0, RV1T0
the saturated densities with their reduced derivatives,
PL1R0, PV1R0, PL1T0, PV1T0
the density- resp. temperature-derivatives of the pressure in the saturated states,
CVL, CVV, ETL, ETV, XLL, XLV
the constant volume heat capacities, the viscosities and the thermal conductivities in the saturated states,
and the densities and pressures on the liquid and vapor spinodals
RHSL, RHSV, PSL, PSV .

SATIF is supplied with data - under the direction of SUBCRI and TWOPHA - by the moduls SSP, SSRHO, SPIRHO, SSPER and SPIP. SATIF supplies data to the moduls DLFIND, DVFIND, REDUKE, TS1, TCORR, SAET01 and VD1.

/REDUF/ G0,S0,Z,P1T0,P1R0,CV0,CP0,ETA,XLMB,R1T0,CPC,H0,VQ
holds thermal properties of the water, mostly in the reduced form, namely

S0, H0, G0
the entropy, the enthalpy and the Gibbs-function,
Z
the factor of reality,
CV0, CP0, CPC
the constant volume and constant pressure heat capacities and their ratio,
P1T0, P1R0, R1T0
the density- resp. temperature-derivative of the pressure
and the temperature-derivative of the density,
ETA, XLMB
the viscosity and the thermal conductivity
and the vapor quality
VQ .

REDUF is filled either by REDUK or by REDUKE, and used by the moduls HYDOR, WATER, WASSER, DAMPF and RDRDP.

/HYDUF/ S,G,H,H1P,U1R,CV,CP,P,P1T,P1R,R1T,R1P,ETA,XLMB,SRFTS,P1RS, VS,PRN,V,VQ
is needed to supply the thermal properties of the water - calculated by HYDOR to the prorams WATER, KAISOK AND KAISOT.

The following common blocks

/BASIF/ GB,SB,ZB,PB1T0,PB1R0,CVB
and

/RESEF/ GR,SR,ZR,PR1T0,PR1R0,CVR
carry contributions - in the reduced form - to the thermal properties as calculated by the modul BASE resp. RESID. The properties are :

S, G
the entropy and the Gibbs-function,
Z
the factor of reality,
P1T0, P1R0
the density- resp. temperature-derivative of the pressure, and
CV
the constant volume heat capacity.

BASIF and RESEF are neded by the moduls DGFIND, DLFIND, DVFIND, REDUK, REDUKE and GPCORR.

/TEMPO/ RT,B1,B2,B1T,B2T,B1TT,B2TT,TV,GI,SI,CVI
holds properties calculated in ONLYT, which depend only from the temperature as e.g.

SI, GI, CVI
the entropy, the Gibbs-function and the constant volume heat capacity
of the ideal gas.

RT is RGAS*T in J / g. The remainig properties are needed only in BASE.

Appendix F. Property routines

The following routines serve directly to describe some thermal property of the water. The essential ones are basically the same, as the corresponding routines in the H.G.K.-code.

F.1 INR105.AQUA.FORT(BLOCK)

```
      BLOCK DATA                                                        BLOCKD
      IMPLICIT REAL*8(A-H,O-Z)                                          BLOCKD
      COMMON /AKONST/ WM,TC,DC,GASCON,G1,G2,GF,TZ,UREF,SREF,TLOW        BLOCKD
      COMMON /ADDCON/ ATZ(4),ADZ(4),AAT(4),AAD(4)                       BLOCKD
      COMMON /NCONST/ G(40),II(40),JJ(40),NC                            BLOCKD
C     THIS BLOCKDATA SUBROUTINE SUPPLIES PARAMETERS AND COEFFICIENTS    BLOCKD
C     USED IN THE REST OF THE ROUTINES.                        93/05/04 BLOCKD
C        D IN G/CM**3, GASCON IN J/(G*K), T IN K                        BLOCKD
C     G1, G2, AND GF ARE THE ALPHA, BETA AND GAMMA   OF EQ. A-2,        BLOCKD
C     G,II,JJ ARE THE G(I),K(I) AND L(I)             OF EQ. A-5.        BLOCKD
C     VALIDITY-LIMITS : TLOW=260. K, THIGH=2500. K, PHIGH=3000. MPA     BLOCKD
C                     DMAX=1.2 G/CM**3                                  BLOCKD
      DATA ATZ/2*64.D1,641.6D0,27.D1/,ADZ/3*.319D0,1.55D0/,AAT/2*2.D4   BLOCKD
     =,4.D4,25.D0/,AAD/34.D0,4.D1,3.D1,1.05D3/                          BLOCKD
      DATA WM/18.0152D0/,GASCON/.461522D0/,TZ/647.073D0/,NC/36/         BLOCKD
     =,TLOW/260.D0/                                                     BLOCKD
      DATA UREF,SREF,TC,DC/-4328.455039D0,7.6180802D0,647.126D0,0.322D0/BLOCKD
      DATA G1,G2,GF/11.D0,44.333333333333D0,3.5D0/                      BLOCKD
      DATA G/-.53062968529023D3, .22744901424408D4, .78779333020687D3   BLOCKD
     1,-.69830527374994D2, .17863832875422D5,-.39514731563338D5         BLOCKD
     2, .33803884280753D5,-.13855050202703D5,-.25637436613260D6         BLOCKD
     3, .48212575981415D6,-.34183016969660D6, .12223156417448D6         BLOCKD
     4, .11797433655832D7,-.21734810110373D7, .10829952168620D7         BLOCKD
     5,-.25441998064049D6,-.31377774947767D7, .52911910757704D7         BLOCKD
     6,-.13802577177877D7,-.25109914369001D6, .46561826115608D7         BLOCKD
     7,-.72752773275387D7, .41774246148294D6, .14016358244614D7         BLOCKD
     8,-.31555231392127D7, .47929666384584D7, .40912664781209D6         BLOCKD
     9,-.13626369388386D7, .69625220862664D6,-.10834900096447D7         BLOCKD
     A,-.22722827401688D6, .38365486000660D6, .68833257944332D4         BLOCKD
     B, .21757245522644D5,-.26627944829770D4,-.70730418082074D5         BLOCKD
     C,-.225D0,-1.68D0,.055D0,-93.0D0/                                  BLOCKD
      DATA II/4*0,4*1,4*2,4*3,4*4,4*5,4*6,4*8,2*2,0,4,3*2,4/            BLOCKD
      DATA JJ/2,3,5,7,2,3,5,7,2,3,5,7,2,3,5,7,2,3,5,7,2,3,5,7,2,3,5,7   BLOCKD
     =,2,3,5,7,1,3*4,0,2,0,0/                                           BLOCKD
      END                                                               BLOCKD
```

F.2 INR105.AQUA.FORT(ONLYT)

```
      SUBROUTINE ONLYT(T,IECCH)                                         ONLYT
C     THIS SUBROUTINE CALCULATES THERMAL PROPERTIES                     ONLYT
C     DEPENDING ONLY FROM THE TEMPERATURE                      93/08/30 ONLYT
      IMPLICIT REAL*8(A-H,O-Z)                                          ONLYT
      COMMON /AKONST/ WM,TC,DC,GASCON,G1,G2,GF,TZ,UREF,SREF,TLOW        ONLYT
      COMMON /TEMPO/  CGT,B1,B2,B1T,B2T,B1TT,B2TT,TV,GI,SI,CVI          ONLYT
      DIMENSION BP(7),BQ(7),C(18)                                       ONLYT
      DATA BP/.7478629D0,-.3540782D0,2*0.D0,.7159876D-2,0.D0            ONLYT
     =,-.3528426D-2/, BQ/1.1278334D0,0.D0,-.5944001D0                   ONLYT
     =,-5.010996D0,0.D0,.63684256D0,0.D0/                               ONLYT
      DATA C/.19730271018D2,.209662681977D2,-.483429455355D0            ONLYT
     =,.605743189245D1,22.56023885D0,-9.87532442D0,-.43135538513D1      ONLYT
     =,.458155781D0,-.47754901883D-1,.41238460633D-2,-.27929052852D-3   ONLYT
     =,.14481695261D-4,-.56473658748D-6,.16200446D-7,-.3303822796D-9    ONLYT
     =,.451916067368D-11,-.370734122708D-13,.137546068238D-15/          ONLYT
      DATA UN/1.D0/                                                     ONLYT
C---------------------------------------------------------------------- ONLYT
      IF(T .GE. TLOW)        GO TO 10                                   ONLYT
      WRITE(6,'(16H TEMPERATURE T =,F10.4,13H K TOO LOW ! /)' ) T       ONLYT
      IECCH=-4                                                          ONLYT
      RETURN                                                            ONLYT
C---------------------------------------------------------------------- ONLYT
C     THIS PART CALCULATES THE B'S OF EQS. 3,4 USING COEFFICIENTS       ONLYT
C     FROM BLOCKDATA , CALCULATING ALSO THE FIRST AND SECOND DERIVS     ONLYT
C     W.R. TO TEMP.  THE B'S CALCULATED HERE ARE IN CM3/G.              ONLYT
   10 CGT=GASCON*T                                                      ONLYT
      T1=UN/T                                                           ONLYT
      TV=TZ*T1                                                          ONLYT
C---------------------------------------------------------------------- ONLYT
      B1=BP(7)*TV                                                       ONLYT
      B2=BQ(7)*TV                                                       ONLYT
      B1T=B1*5                                                          ONLYT
      B2T=B2*5                                                          ONLYT
      B1TT=B1T*6                                                        ONLYT
      B2TT=B2T*6                                                        ONLYT
      DO 20 I=1,4                                                       ONLYT
      JI=7-I                                                            ONLYT
      AP=BP(JI)                                                         ONLYT
      AQ=BQ(JI)                                                         ONLYT
      B1=(B1+AP)*TV                                                     ONLYT
      B2=(B2+AQ)*TV                                                     ONLYT
      AP=AP*(JI-2)                                                      ONLYT
      AQ=AQ*(JI-2)                                                      ONLYT
      B1T=(B1T+AP)*TV                                                   ONLYT
      B2T=(B2T+AQ)*TV                                                   ONLYT
      AP=AP*(JI-1)                                                      ONLYT
      AQ=AQ*(JI-1)                                                      ONLYT
      B1TT=(B1TT+AP)*TV                                                 ONLYT
   20 B2TT=(B2TT+AQ)*TV                                                 ONLYT
      B1=B1-BP(2)*DLOG(TV)+BP(1)                                        ONLYT
      B2=B2+BQ(1)                                                       ONLYT
      B1T=(BP(2)-B1T)*T1                                                ONLYT
```

```
      B2T=-B2T*T1                                                       ONLYT
      B1TT=(-BP(2)+B1TT)*T1*T1                                          ONLYT
      B2TT=B2TT*T1*T1                                                   ONLYT
C---------------------------------------------------------------------- ONLYT
C     THIS PART CALCULATES THE THERMODYNAMIC PROPERTIES FOR             ONLYT
C     WATER IN THE IDEAL GAS STATE FROM FUNCTION OF H.W. WOOLLEY        ONLYT
   50 TT=T*1.D-2                                                        ONLYT
      TT1=UN/TT                                                         ONLYT
      TL=DLOG(TT)                                                       ONLYT
      GI=-(C(2)+C(1)*TT1)*TL                                            ONLYT
      HI= C(2)+C(1)*(UN-TL)*TT1                                         ONLYT
      CPI=C(2)-C(1)*TT1                                                 ONLYT
      DO 60 J=3,18                                                      ONLYT
      CIT=C(J)*TT**(J-6)                                                ONLYT
      GI=GI-CIT                                                         ONLYT
      HI=HI+(J-6)*CIT                                                   ONLYT
   60 CPI=CPI+(J-5)*(J-6)*CIT                                           ONLYT
      SI=HI-GI                                                          ONLYT
      CVI=CPI-UN                                                        ONLYT
C---------------------------------------------------------------------- ONLYT
      RETURN                                                            ONLYT
      END                                                               ONLYT
```

F.3 INR105.AQUA.FORT(BASE)

```
      SUBROUTINE BASE(T,D,IECCH)                                        BASE
C     BASE CALCULATES REDUCED THERMAL PROPERTIES OF THE WATER           BASE
C     VIA THE BASE-FUNCTION FOR A GIVEN TEMPERATURE, T AND DENSITY, D.  BASE
C     THE PROPERTIES ARE: ZB=P/(D*RT), PB1T0=(DP/DT)/(D*R),             BASE
C     PB1R0=(DP/DRH)/(T*R), GB=G/RT, SB=S/R, CVB=CV/R.                  BASE
C     T IS IN K, D IS IN G/CM**3.                             94/12/01  BASE
      IMPLICIT REAL*8 (A-H,O-Z)                                         BASE
C  G1,G2 AND GF ARE THE ALPHA, BETA AND GAMMA OF EQ 2, WHICH ARE        BASE
C  SUPPLIED BY THE BLOCKDATA ROUTINE.  B1 AND B2 ARE THE "EXCLUDED      BASE
C  VOLUME" AND "2ND VIRIAL" (EQS 3 AND 4) SUPPLIED BY THE SUBROUTINE    BASE
C  BB(T), WHICH ALSO SUPLIES THE 1ST AND 2ND DERIVATIVES WITH           BASE
C  RESPECT TO T (B1T,B2T,B1TT,B2TT).                                    BASE
      COMMON /AKONST/ WM,TC,DC,GASCON,G1,G2,GF,TZ,UREF,SREF,TLOW        BASE
      COMMON /TEMPO/  CGT,B1,B2,B1T,B2T,B1TT,B2TT,TV,GI,SI,CVI          BASE
      COMMON /BASIF/  GB,SB,ZB,PB1T0,PB1R0,CVB                          BASE
      PARAMETER(DMAX=1.2D0)                                             BASE
      DATA UN/1.D0/                                                     BASE
      IF(D .GT. DMAX)        THEN                                       BASE
          WRITE(6,9100) D                                               BASE
          IECCH = 4                                                     BASE
          RETURN                                                        BASE
      ENDIF                                                             BASE
 9100 FORMAT(12H DENSITY D =,G15.6,20H G/CM**3 TOO HIGH ! /)            BASE
   10 YK=2.5D-1*B1*D                                                    BASE
      W=UN/(UN-YK)                                                      BASE
      W2=W*W                                                            BASE
      Z0=(UN+(G1+G2*YK)*YK)*W2*W                                        BASE
      BPB=B2/B1                                                         BASE
```

```
      Z1=4.D0*(BPB-GF)                                                  BASE
      ZB = Z0+Z1*YK                                                     BASE
      DZ0=( 3.D0*Z0 + (G1+2.D0*G2*YK)*W2 )*W                            BASE
      DBZ = DZ0+Z1                                                      BASE
      AB = 17.4560887731D0 + Z1*YK + W*(UN-G2+28.16666667D0*W)          BASE
     =+ DLOG(D*CGT*W)                                                   BASE
      GB = AB + ZB                                                      BASE
      UBB1=B1T/B1                                                       BASE
      UB = -T*(UBB1*(ZB-UN-D*B2) + D*B2T)                               BASE
      SB = UB - AB                                                      BASE
      CVB = 2.D0*UB + T*T*( (Z0-UN)*(UBB1*UBB1-B1TT/B1)                 BASE
     =- D*(B2TT-GF*B1TT) - UBB1*UBB1*YK*DZ0 )                           BASE
      PB1T0 = ZB + T*D*(B2T+(2.5D-1*DBZ-BPB)*B1T)                       BASE
      PB1R0 = ZB+DBZ*YK                                                 BASE
      RETURN                                                            BASE
      END                                                               BASE
```

F.4 INR105.AQUA.FORT(RESID)

```
      SUBROUTINE RESID(T,D)                                             RESID
C     RESID CALCULATES REDUCED THERMAL PROPERTIES OF THE WATER VIA      RESID
C     THE RESIDUAL-FUNCTION FOR A GIVEN TEMPERATURE, T AND DENSITY, D.  RESID
C     THE PROPERTIES ARE: ZR=P/(D*RT), PR1T0=(DP/DT)/(D*R),             RESID
C     PR1R0=(DP/DRH)/(T*R), GR=G/RT, SR=S/R, CVR=CV/R.                  RESID
C     T IS IN K, D IS IN G/CM**3.                             94/11/30  RESID
C   TERMS 37 THRU 39 ARE THE ADDITIONAL TERMS AFFECTING ONLY THE        RESID
C   IMMEDIATE VICINITY OF THE CRITICAL POINT, AND TERM 40 IS THE        RESID
C   ADDITIONAL TERM IMPROVING THE LOW T, HIGH P REGION.                 RESID
      IMPLICIT REAL*8(A-H,O-Z)                                          RESID
      DIMENSION QR(11),QT(10),QZR(9),QZT(9)                             RESID
      COMMON /AKONST/ WM,TC,DC,GASCON,G1,G2,GF,TZ,UREF,SREF,TLOW        RESID
      COMMON /ADDCON/ ATZ(4),ADZ(4),AAT(4),AAD(4)                       RESID
      COMMON /NCONST/ G(40),II(40),JJ(40),NC                            RESID
      COMMON /TEMPO/  CGT,B1,B2,B1T,B2T,B1TT,B2TT,TV,GI,SI,CVI          RESID
      COMMON /RESEF/  GR,SR,ZR,PR1T0,PR1R0,CVR                          RESID
      EQUIVALENCE (QR(3),QZR(1)),(QT(2),QZT(1))                         RESID
      DATA ZERO,UN/0.D0,1.D0/                                           RESID
      GASI=UN/GASCON                                                    RESID
      QR(1)=ZERO                                                        RESID
      PRR=ZERO                                                          RESID
      PR=ZERO                                                           RESID
      AR=ZERO                                                           RESID
      DADT=ZERO                                                         RESID
      CVR=ZERO                                                          RESID
      PTR=ZERO                                                          RESID
      IF(D .LT. 1.D-15)      GO TO 90                                   RESID
      W = D                                                             RESID
      TWPD=2.D0/D                                                       RESID
      IF(W .GE. 1.D-3)       THEN                                       RESID
          E=DEXP(-W)                                                    RESID
          Q20=UN-E                                                      RESID
      ELSE                                                              RESID
          Q20=W*(UN-5.D-1*W*(UN-3.3333333D-1*W))                        RESID
```

```
          E=UN-Q20                                                      RESID
      ENDIF                                                             RESID
      VER0=E/Q20                                                        RESID
      VER1=VER0*D*D                                                     RESID
      Q10=D*D*E                                                         RESID
      Q10RT=Q10*CGT                                                     RESID
      QR(2)=Q10                                                         RESID
      QT(1)=T/TZ                                                        RESID
      DO 11 I=2,10                                                      RESID
      IF(QR(I) .LE. 1.E-35)  QR(I)=ZERO                                 RESID
      QR(I+1)=QR(I)*Q20                                                 RESID
   11 QT(I)=QT(I-1)*TV                                                  RESID
      DO 21 I=1,NC                                                      RESID
      K=II(I)+1                                                         RESID
      L=JJ(I)                                                           RESID
      GPP=G(I)*QZT(L)                                                   RESID
      QP=GPP*QZR(K-1)                                                   RESID
      PR = PR + QP                                                      RESID
      PRR = PRR + QP*( TWPD - (UN-(K-1)*VER0) )                         RESID
      BUAM=K*LOG(Q20)                                                   RESID
      IF(BUAM  .LE. -35.)    AUA=ZERO                                   RESID
      IF(BUAM  .GT. -35.)    AUA=EXP(BUAM)                              RESID
      AR=AR+GPP*QZR(K)/(K*Q10RT)                                        RESID
      DFDT=AUA*(1-L)*QZT(L+1)/(TZ*K)                                    RESID
      GPDT=G(I)*DFDT                                                    RESID
      DADT=DADT+GPDT                                                    RESID
      DPT=GPDT*K*VER1                                                   RESID
      PTR=PTR+DPT                                                       RESID
      D2F=L*GPDT*GASI                                                   RESID
   21 CVR=CVR+D2F                                                       RESID
      QP=ZERO                                                           RESID
      Q2A=ZERO                                                          RESID
      DO 31 J=37,40                                                     RESID
      IF(G(J) .EQ. ZERO)     GO TO 31                                   RESID
      K=II(J)                                                           RESID
      KM=JJ(J)                                                          RESID
      DDZ = ADZ(J-36)                                                   RESID
      DDZ1 = UN/DDZ                                                     RESID
      DEL = D*DDZ1 - UN                                                 RESID
      IF(DABS(DEL) .LT. 1.D-10)                  DEL=1.D-10             RESID
      EX1 = -AAD(J-36)*DEL**K                                           RESID
      IF(EX1 .GT. -170.D0)   DEX=DEXP(EX1)*DEL**KM                      RESID
      IF(EX1 .LE. -170.D0)   DEX=ZERO                                   RESID
      ATT = AAT(J-36)                                                   RESID
      TX1=UN/ATZ(J-36)                                                  RESID
      TA=T*TX1                                                          RESID
      TAU=TA-UN                                                         RESID
      EX2 = -ATT*TAU*TAU                                                RESID
      IF(EX1+EX2 .GT. -170.D0)                   THEN                   RESID
          TEX=DEXP(EX2)                                                 RESID
          Q10 = DEX*TEX                                                 RESID
          IF(DABS(Q10) .LT. 1.D-14)              Q10=ZERO               RESID
      ELSE                                                              RESID
          Q10 = ZERO                                                    RESID
          GO TO 31                                                      RESID
```

```
      ENDIF                                                             RESID
      QM1 = KM/DEL                                                      RESID
      QM2 = K*AAD(J-36)*DEL**(K-2)                                      RESID
      QM = QM1 - QM2*DEL                                                RESID
      FCT1=D*D*Q10*DDZ1                                                 RESID
      FCT=QM*FCT1                                                       RESID
      Q5T = (FCT1*DDZ1)*(QM*(TWPD*DDZ+QM) - (QM1/DEL+(K-1)*QM2))        RESID
      PRR = PRR + Q5T*G(J)                                              RESID
      QP = QP + G(J)*FCT                                                RESID
      GATT = 2.D0*G(J)*ATT*TX1                                          RESID
      GATTA = GATT*TAU                                                  RESID
      IF(DABS(GATTA) .LT. 1.D-14)                GATTA=ZERO             RESID
      DADT = DADT - Q10*GATTA                                           RESID
      PTR = PTR - GATTA*FCT                                             RESID
      Q2A = Q2A + Q10*GATT*(2.D0*EX2+UN)*TA                             RESID
      AR = AR + Q10*G(J)/CGT                                            RESID
   31 CONTINUE                                                          RESID
      CVR = CVR + Q2A*GASI                                              RESID
      PR = PR + QP                                                      RESID
      ZR = PR/(CGT*D)                                                   RESID
      PR1T0 = PTR/(GASCON*D)                                            RESID
      PR1R0 = PRR/CGT                                                   RESID
   90 GR = AR + ZR                                                      RESID
      SR = -DADT*GASI                                                   RESID
      RETURN                                                            RESID
      END                                                               RESID
```

F.5 INR105.AQUA.FORT(SIGMA)

```
      FUNCTION SIGMA(T)                                                 SIGMA
C     CALCULATES THE SURFACE TENSION OF THE WATER IN N/M      94/12/07  SIGMA
      IMPLICIT REAL*8(A-H,O-Z)                                          SIGMA
      PARAMETER(TC=647.126D0,BG=0.2358D0,BK=-0.625D0,AMU=1.256D0)       SIGMA
      Y=(TC-T)/TC                                                       SIGMA
      SIGMA=BG*(1+BK*Y)*Y**AMU                                          SIGMA
      RETURN                                                            SIGMA
      END                                                               SIGMA
```

F.6 INR105.AQUA.FORT(TRANSP)

```
      SUBROUTINE TRANSP(T,RH,P1R0,P1T0,ETA,XLMB)                        TRANSP
C     TRANSP CALCULATES THE TRANSPORT PROPERTIES                        TRANSP
C     VISCOSITY ( ETA ) AND THERMAL CONDUCTIVITY ( XLMB )               TRANSP
C     AS FUNCTIONS OF THE TEMPERATURE ( T ) AND THE DENSITY ( RH ).     TRANSP
C     ALL THE PROPERTIES ARE IN IN S.I. UNITS .                         TRANSP
C     'TRANSP' NEEDS ALSO THE REDUCED PRESSURE DERIVATIVES              TRANSP
C     P1R0 = DP/(RGAS*T*DRHO) AND  P1T0 = DP/(RGAS*RHO*DT)              TRANSP
C     FOR CALCULATING THE NONREGULAR PART OF THE                        TRANSP
C     THERMAL CONDUCTIVITY ( DELLAM )                         94/12/01  TRANSP
      IMPLICIT REAL*8(A-H,O-Z)                                          TRANSP
      DIMENSION E0(4),X0(4),BE(6,5),BX(5,6)                             TRANSP
      PARAMETER(TC=647.126D0,RGAS=4.61522D+2)                           TRANSP
```

```
      DATA RZO,TZO,PZO/317.763D+0,647.126D+0,22.115D+6/                 TRANSP
     =,CC,OMEGA,AC,BC/3.7711D-8,0.4678D+0,18.66D+0,1.0D+0/              TRANSP
      DATA E0/1.81583D-2,1.77624D-2,1.05287D-2,-0.36744D-2/             TRANSP
      DATA X0/2.02223D+0,1.411166D+1,5.25597D+0,-2.01870D+0/            TRANSP
      DATA BE/5.01938D-1,1.62888D-1,-1.30356D-1,9.07919D-1,-5.51119D-1  TRANSP
     =,1.46543D-1,2.35622D-1,7.89393D-1,6.73665D-1,1.207552D-0          TRANSP
     =,6.70665D-2,-8.43370D-2,-2.74637D-1,-7.43539D-1,-9.59456D-1       TRANSP
     =,-6.87343D-1,-4.97089D-1,1.95286D-1,1.45831D-1,2.63129D-1         TRANSP
     =,3.47247D-1,2.13486D-1,1.00754D-1,-3.29320D-2,-2.70448D-2         TRANSP
     =,-2.53093D-2,-2.67758D-2,-8.22904D-2,6.02253D-2,-2.02595D-2/      TRANSP
      DATA BX/1.32930460D-0,1.7018363D-0,5.2246158D-0,8.7127675D-0      TRANSP
     =,-1.852599D-0,-4.0452437D-1,-2.2156845D-0,-1.0124111D+1           TRANSP
     =,-9.5000611D-0,9.340469D-1,2.440949D-1,1.6511057D-0,4.9874687D-0  TRANSP
     =,4.3786606D-0,0.D-0,1.8660751D-2,-7.6736002D-1,-2.7297694D-1      TRANSP
     =,-9.1783782D-1,0.D-0,-1.2961068D-1,3.7283344D-1,-4.3083393D-1     TRANSP
     =,0.D-0,0.D-0,4.4809953D-2,-1.120316D-1,1.3333849D-1,0.D-0,0.D-0/  TRANSP
C     THE IDEAL TEMPERATURE                                             TRANSP
      TID=PZO/(RZO*RGAS)                                                TRANSP
      TR=T/TID                                                          TRANSP
C     REDUCED VARIABLES                                                 TRANSP
      TT=T/TZO                                                          TRANSP
      T1=1.D0/TT                                                        TRANSP
      T2=SQRT(TT)                                                       TRANSP
      T3=T1-1.D-0                                                       TRANSP
      T4=TT-1.D-0                                                       TRANSP
      RR=RH/RZO                                                         TRANSP
      R2=SQRT(RR)                                                       TRANSP
      R3=RR-1.D-0                                                       TRANSP
C     VISCOSITY                                                         TRANSP
      EN=E0(1)+T1*(E0(2)+T1*(E0(3)+T1*E0(4)))                           TRANSP
      ETA0=T2*1.D-6/EN                                                  TRANSP
      ESU=0.D0                                                          TRANSP
      DO 110 J=6,1,-1                                                   TRANSP
      BEJ = BE(J,1)+R3*(BE(J,2)+R3*(BE(J,3)+R3*(BE(J,4)+ R3*BE(J,5))))  TRANSP
  110 ESU=T3*ESU+BEJ                                                    TRANSP
      ETA=ETA0*EXP(RR*ESU)                                              TRANSP
C     THERMAL CONDUCTIVITY                                              TRANSP
      XN=X0(1)+T1*(X0(2)+T1*(X0(3)+T1*X0(4)))                           TRANSP
      XLA0=T2/XN                                                        TRANSP
      EXU=0.D0                                                          TRANSP
      DO 210 I=5,1,-1                                                   TRANSP
      BXI = BX(I,1)+R3*(BX(I,2)+R3*(BX(I,3)                             TRANSP
     =+R3*(BX(I,4)+R3*(BX(I,5)+R3*BX(I,6)))))                           TRANSP
  210 EXU=T3*EXU+BXI                                                    TRANSP
      QXQ=RR*EXU                                                        TRANSP
      XLMB=XLA0*EXP(RR*EXU)                                             TRANSP
C     NONREGULAR PART OF THE THERMAL CONDUCTIVITY                       TRANSP
C     ISOTHERMAL COMPRESSIBILITY                                        TRANSP
  230 IF(P1R0 .NE. 0.D0)     CHIT = RR/(TR*P1R0)                        TRANSP
      IF(P1R0 .EQ. 0.D0)     CHIT = 1.D40                               TRANSP
      FIO = ABS(CHIT)                                                   TRANSP
      IF(FIO .EQ. 0.D0)      FIO=1.D-6                                  TRANSP
      CHITE = FIO**OMEGA                                                TRANSP
      ZW=R3*R3                                                          TRANSP
C     DAMPNG FUNCTION OF SENGERS, J. V. , 10TH INT. CONF. ON STEAM      TRANSP
```

```
      CHECKA=R2*EXP(-AC*T4*T4-BC*ZW*ZW)                                 TRANSP
      ZW=TR*P1T0                                                        TRANSP
      DELLAM =(CC/ETA)*(ZW*ZW)*CHITE*CHECKA                             TRANSP
      XLMB=XLMB+DELLAM                                                  TRANSP
  999 RETURN                                                            TRANSP
      END                                                               TRANSP
```

F.7 INR105.AQUA.FORT(SSP)

```
      SUBROUTINE SSP(T)                                                 SSP
C     SSP(T) CALCULATES APPROXIMATIONS                                  SSP
C     TO THE VAPOR PRESSURE       P(T)                   ( PS      )    SSP
C     TO ITS FIRST                T*DPS(T)/DT            ( TPS1T   )    SSP
C     AND SECOND DERIVATIVE       T*T*D2PS(T)/DT**2      ( TTPS2T  )    SSP
C     AS FUNCTIONS OF THE INPUT TEMPERATURE T IN PA-S.                  SSP
C     AFTER WAGNER ET SAUL, 1984,  IAPS MOSCOW CONFERENCE.    94/11/30  SSP
      IMPLICIT REAL*8(A-H,O-Z)                                          SSP
      COMMON /SATIF/  PS,TPS1T,TTPS2T,DL,DV,RL1T0,RV1T0,RHSL,PSL        SSP
     =,RHSV,PSV,SL,GL,PL1R0,PL1T0,CVL,ETL,XLL,PV1R0,PV1T0,CVV,ETV,XLV   SSP
      PARAMETER(TC=647.126D0,PC=22.0549D6)                              SSP
      DATA UN/1.D0/                                                     SSP
      IF(T .GT. TC)          RETURN                                     SSP
      TR=T/TC                                                           SSP
      TI=UN/TR                                                          SSP
      W=SQRT(UN-TR)                                                     SSP
      W3=W*W*W                                                          SSP
      W7=W*W3*W3                                                        SSP
      POLA = -7.85823D+0 + W*( 1.83991D+0 - W3*( 11.7811D+0             SSP
     = - W*( 22.67050D+0 - W*( 15.9393D+0 - W7*1.77516D+0 ))))          SSP
      POLB = W*( 9.19955D-1 - W3*( 23.5622D+0                           SSP
     = - W*( 56.67625D+0 - W*( 47.8179D+0 - W7*11.53854D+0 ))))         SSP
      POLC = W*( 1.3799325D-0 - W3*( 70.6866D+0                         SSP
     = - W*( 198.366875D+0 - W*( 191.2716D+0 - W7*86.53905D+0 ))))      SSP
      SSPLN=(TI-UN)*POLA                                                SSP
      PS=EXP(SSPLN)*PC                                                  SSP
      Q1=-(TI*POLA+POLB)                                                SSP
      TPS1T=PS*Q1                                                       SSP
      IF(T .EQ. TC)          RETURN                                     SSP
      Q2=(TI*POLA+POLB+POLC/(TI-UN))                                    SSP
      TTPS2T=PS*(Q2-Q1*(UN-Q1))                                         SSP
      RETURN                                                            SSP
      END                                                               SSP
```

F.8 INR105.AQUA.FORT(TS1)

```
      SUBROUTINE TS1(P,TS,TS1P,IECCH)                                   TS1
C     TS1 RETURNS - AT A GIVEN PRESSURE P IN PA  - APPROXIMATED VALUES  TS1
C     FOR THE SATURATION TEMPERATURE, TS AND                            TS1
C     FOR THE DERIVATIVE TS1P = DTS/DP                        94/12/01  TS1
      IMPLICIT REAL*8(A-H,O-Z)                                          TS1
      COMMON /SATIF/  PS,TPS1T,TTPS2T,DL,DV,RL1T0,RV1T0,RHSL,PSL        TS1
     =,RHSV,PSV,SL,GL,PL1R0,PL1T0,CVL,ETL,XLL,PV1R0,PV1T0,CVV,ETV,XLV   TS1
      PARAMETER(PMIN=6.1D+2,PCN=21.75966D6,PC=22.0549D6)                TS1
C     PARAMETER(TMIN=273.15D0,TCN=646.00,TC=647.126D0)                  TS1
      DATA FINIS/1.D-5/                                                 TS1
      IF(P .GE. PMIN .AND. P .LE. PC)           GO TO 10                TS1
      WRITE(6,9100) P                                                   TS1
      IF(P .LT. PMIN)        IECCH=-2                                   TS1
      IF(P .GT. PC)          IECCH=+2                                   TS1
      RETURN                                                            TS1
C     ANALYTIC DESCRIPTION                                              TS1
   10 IF(P .GT. PCN)         THEN                                       TS1
          TS=562.764692D0+3.82515211D-6*P                               TS1
      ELSE                                                              TS1
          PL=DLOG(P)-11.512925465D0                                     TS1
          TS=372.83D0+                                                  TS1
     =    PL*(27.7589D0+PL*(2.3819D0+PL*(0.24834D0+PL*0.193855D-1)))    TS1
      ENDIF                                                             TS1
C     CORRECTURES                                                       TS1
      DO 21 K=1,7                                                       TS1
      CALL SSP(TS)                                                      TS1
      IF(P .GE. PCN)         GO TO 90                                   TS1
      DP=P-PS                                                           TS1
      DEV=ABS(DP/P)                                                     TS1
      IF(DEV .LT. FINIS)     GO TO 90                                   TS1
   21 TS = TS*(1.D0+DP/TPS1T)                                           TS1
      IECCH=3                                                           TS1
      WRITE(6,9300)                                                     TS1
      RETURN                                                            TS1
   90 TS1P=TS/TPS1T                                                     TS1
 9100 FORMAT(27H PRESSURE RANGE EXCEED! P =,G15.6,7H PASCAL)            TS1
 9300 FORMAT(51H ITERATION FOR THE SATURATUION TEMPERATURE FAILED ! )   TS1
      RETURN                                                            TS1
      END                                                               TS1
```

F.9 INR105.AQUA.FORT(SSRHO)

```
      SUBROUTINE SSRHO(T)                                               SSRHO
C     SSRHO RETURNS APPROXIMATED VALUES OF THE DENSITIES OF             SSRHO
C     THE SATURATED LIQUID ( DL )  AND THE SATURATED VAPOR ( DV )       SSRHO
C     - IN G/CM**3 ! - AS FUNCTIONS OF THE TEMPERATURE ( T ).           SSRHO
C     SSRHO CALCULATES ALSO THE REDUCED DENSITY DERIVATIVES             SSRHO
C     RL1T0 = T/RHL * DRHL/DT AND RV1T0 = T/RHV * DRHV/DT.    94/12/01  SSRHO
      IMPLICIT REAL*8(A-H,O-Z)                                          SSRHO
      DIMENSION PUL(12),DPUL(12),PUV(12),DPUV(12)                       SSRHO
```

```
      COMMON /SATIF/  PS,TPS1T,TTPS2T,DL,DV,RL1T0,RV1T0,RHSL,PSL        SSRHO
     =,RHSV,PSV,SL,GL,PL1R0,PL1T0,CVL,ETL,XLL,PV1R0,PV1T0,CVV,ETV,XLV   SSRHO
      PARAMETER(TC=647.126D0,DC=0.322D0,RGAS=4.61522D+2)                SSRHO
      PARAMETER(BR=2.03D0,BET=0.325D0)                                  SSRHO
      DATA PUL/        6.072746418D+0,-149.9456089D+0,2.121007381D+3    SSRHO
     =,-17.16483301D+3,90.17684684D+3,-323.0456094D+3,805.4435472D+3    SSRHO
     =,-1.398515305D+6,1.658696365D+6,-1.281195416D+6,580.7414278D+3    SSRHO
     =,-117.1819097D+3/                                                 SSRHO
      DATA DPUL/ 0.0D+0,-74.97280445D+0,  2.121007381D+3                SSRHO
     =,-25.747249515D+3, 180.35369368D+3,-807.6140235D+3                SSRHO
     =, 2.4163306416D+6,-4.8948035675D+6, 6.63478546D+6                 SSRHO
     =,-5.765379372D+6,  2.903707139D+6, -644.50050335D+3/              SSRHO
      DATA PUV/-1.125999398D+0,42.15284538D+0,-538.0831003D+0           SSRHO
     =,4.150173161D+3,-20.80947751D+3,71.29507069D+3,-170.4446088D+3    SSRHO
     =,284.4989747D+3,-325.1357165D+3,242.4593090D+3,-106.2552030D+3    SSRHO
     =,20.74762670D+3/                                                  SSRHO
      DATA DPUV/0.0D+0,21.07642269D+0,-538.0831003D+0,6.225259741D+3    SSRHO
     =,-41.61895501D+3,178.2376767D+3,-511.3338264D+3,995.7464115D+3    SSRHO
     =,-1.300542866D+6,1.091066891D+6,-531.2760148D+3,114.1119469D+3/   SSRHO
      IF(T .GE. TC)          THEN                                       SSRHO
          DL = DC                                                       SSRHO
          DV = DC                                                       SSRHO
          RETURN                                                        SSRHO
      ENDIF                                                             SSRHO
      Y = (TC-T)/TC                                                     SSRHO
      CHI = T/(T-TC)                                                    SSRHO
      IF(T .LE. 620.D0)      THEN                                       SSRHO
C     LOW TEMPERATURES                                                  SSRHO
          W = SQRT(Y)                                                   SSRHO
          OML  =  PUL(12)                                               SSRHO
          TDOML  =  DPUL(12)                                            SSRHO
          ZV = PUV(12)                                                  SSRHO
          TDZV = DPUV(12)                                               SSRHO
          DO 31 K = 11,1,-1                                             SSRHO
          TDOML = TDOML*W+DPUL(K)                                       SSRHO
          TDZV  =  TDZV*W+DPUV(K)                                       SSRHO
          OML = OML*W+PUL(K)                                            SSRHO
  31      ZV  =  ZV*W+PUV(K)                                            SSRHO
          DLOML = TDOML/OML                                             SSRHO
          DV = PS*1.D-3/(RGAS*T*ZV)                                     SSRHO
          DLZV = CHI*TDZV/ZV                                            SSRHO
          RV1T0 = TPS1T/PS -1.D0 -DLZV                                  SSRHO
      ELSE                                                              SSRHO
C     NEAR CRITICAL TEMPERATURES                                        SSRHO
          BEG = BR*Y**BET                                               SSRHO
          EB = BEG*BET                                                  SSRHO
          AL = Y*(3.075668853D-0-Y*(19.58060683D-0-Y*Y*1.383086689D+3)) SSRHO
          OML = 1.D0 + BEG + AL                                         SSRHO
          DAL = AL -Y*Y*(19.58060683D-0-Y*Y*4.149260067D+3)             SSRHO
          DLOML = (EB + DAL)/OML                                        SSRHO
          AV = Y*(1.451962241D-0-Y*(5.041325559D-0-Y*Y*611.1810749D-0)) SSRHO
          OMV = 1.D0 -BEG + AV                                          SSRHO
          DAV = AV -Y*Y*(5.041325559D-0 -Y*Y*1.833543225D+3)            SSRHO
          DV = OMV*DC                                                   SSRHO
          RV1T0 = CHI*(-EB + DAV)/OMV                                   SSRHO
```

```
      ENDIF                                                              SSRHO
      DL = OML*DC                                                        SSRHO
      RL1T0 = DLOML*CHI                                                  SSRHO
      RETURN                                                             SSRHO
      END                                                                SSRHO
```

F.10 INR105.AQUA.FORT(SPIRHO)

```
      SUBROUTINE SPIRHO(T)                                               SPIRHO
C     SPIRHO RETURNS - FOR A GIVEN TEMPEREATURE T - APPROXIMATED VALUES  SPIRHO
C     OF THE DENSITY OF THE LIQUID SPINODAL ( RHSL ) AND                 SPIRHO
C     OF THE VAPOR SPINODAL ( RHSV ) IN KG/M**3.                94/12/01 SPIRHO
      IMPLICIT REAL*8(A-H,O-Z)                                           SPIRHO
      DIMENSION PUL(9),PUV(9)                                            SPIRHO
      COMMON /SATIF/  PS,TPS1T,TTPS2T,DL,DV,RL1T0,RV1T0,RHSL,PSL         SPIRHO
     =,RHSV,PSV,SL,GL,PL1R0,PL1T0,CVL,ETL,XLL,PV1R0,PV1T0,CVV,ETV,XLV    SPIRHO
      PARAMETER(TC=647.126D0,RHC=322.D0,RGAS=4.61522D+2)                 SPIRHO
      PARAMETER(BSP=1.40D0,BET=0.325D0)                                  SPIRHO
      DATA PUL/1.911888766D+0,-16.54157899D+0,179.7358194D+0             SPIRHO
     =,-940.1374463D+0,2.957964417D+3,-5.752429056D+3,6.772987553D+3     SPIRHO
     =,-4.428001759D+3,1.234441322D+3/                                   SPIRHO
      DATA PUV/0.9201508683D-0,-2.159701944D-0,-0.2932760492D-0          SPIRHO
     =,6.324940268D-0,-10.49597891D-0,8.78261245D-0,-3.119252324D-0      SPIRHO
     =,0.D0,0.D0/                                                        SPIRHO
      IF(T .GE. TC)          THEN                                        SPIRHO
          RHSL = RHC                                                     SPIRHO
          RHSV = RHC                                                     SPIRHO
          RETURN                                                         SPIRHO
      ENDIF                                                              SPIRHO
      Y = (TC-T)/TC                                                      SPIRHO
      IF(T .LE. 620.D0)      THEN                                        SPIRHO
C     LOW TEMPERATURES                                                   SPIRHO
          W = SQRT(Y)                                                    SPIRHO
          OML = PUL(9)                                                   SPIRHO
          OMV = PUV(9)                                                   SPIRHO
          DO 31 I=8,1,-1                                                 SPIRHO
          OML = OML*W+PUL(I)                                             SPIRHO
 31       OMV = OMV*W+PUV(I)                                             SPIRHO
      ELSE                                                               SPIRHO
C     NEAR CRITICAL TEMPERATURES                                         SPIRHO
          AL = Y*(487.1717941D-3+Y*(24.26899415D-0-Y*Y*2.461494028D+3))  SPIRHO
          OML = 1.D0 + BSP*Y**BET + AL                                   SPIRHO
          AV = Y*(1.132819731D-0-Y*(30.70131059D-0-Y*Y*3.324944936D+3))  SPIRHO
          OMV = 1.D0 - BSP*Y**BET + AV                                   SPIRHO
      ENDIF                                                              SPIRHO
      RHSL = OML*RHC                                                     SPIRHO
      RHSV = OMV*RHC                                                     SPIRHO
      RETURN                                                             SPIRHO
      END                                                                SPIRHO
```

F.11 INR105.AQUA.FORT(SPIP)

```
      SUBROUTINE SPIP(T)                                                SPIP
C     SPIP RETURNS - FOR A GIVEN TEMPEREATURE T - APPROXIMATED VALUES   SPIP
C     OF THE PRESSURE OF THE LIQUID SPINODAL ( PSL ) AND                SPIP
C     OF THE VAPOR SPINODAL ( PSV ) IN PASCALS.               94/12/01  SPIP
      IMPLICIT REAL*8(A-H,O-Z)                                          SPIP
      DIMENSION PUL(9),PUV(9)                                           SPIP
      COMMON /SATIF/  PS,TPS1T,TTPS2T,DL,DV,RL1T0,RV1T0,RHSL,PSL        SPIP
     =,RHSV,PSV,SL,GL,PL1R0,PL1T0,CVL,ETL,XLL,PV1R0,PV1T0,CVV,ETV,XLV   SPIP
      PARAMETER(TC=647.126D0,RHC=322.D0,RGAS=4.61522D+2)                SPIP
      PARAMETER(PSVCL=498.D6,PSVCV=200.D6,PC=22.0549D6)                 SPIP
      DATA PUL/-1.387361801D-3,151.0341901D-3,1.980521347D-0            SPIP
     =,        2.921317875D-0,-38.3240633D-0,162.2329521D-0             SPIP
     =,       -384.0027285D-0,479.3961544D-0,-264.1248035D-0/           SPIP
      DATA PUV/-749.7146775D-6,162.845372D-3,-319.0428782D-3            SPIP
     =,       -1.582269493D-0, 6.787420254D-0,-10.41333815D-0           SPIP
     =,        7.599390008D-0,-2.285884902D-0,0.D0/                     SPIP
      IF(T .GE. TC)          THEN                                       SPIP
          PSL=PC                                                        SPIP
          RETURN                                                        SPIP
      ENDIF                                                             SPIP
      Y=(TC-T)/TC                                                       SPIP
      IF(T .LE. 620.D0)      THEN                                       SPIP
C     LOW TEMPERATURES                                                  SPIP
          DPSL = PUL(9)                                                 SPIP
          DPSV = PUV(9)                                                 SPIP
          DO 31 I=8,1,-1                                                SPIP
          DPSL = DPSL*Y+PUL(I)                                          SPIP
   31     DPSV = DPSV*Y+PUV(I)                                          SPIP
      ELSE                                                              SPIP
C     NEAR CRITICAL TEMPERATURES                                        SPIP
          DPSL = Y*(52.35363429D-3+Y*(4.419437236D-0-Y*18.86727538D-0)) SPIP
          DPSV = Y*(110.3469546D-3+Y*(913.150777D-3 -Y*10.87001272D-0)) SPIP
      ENDIF                                                             SPIP
      PSL=PS-DPSL*PSVCL                                                 SPIP
      PSV=PS+DPSV*PSVCV                                                 SPIP
      RETURN                                                            SPIP
      END                                                               SPIP
```

F.12 INR105.AQUA.FORT(SSPER)

```
      SUBROUTINE SSPER(T)                                               SSPER
C     SSPER RETURNS - FOR A GIVEN TEMPERATURE T -                       SSPER
C     APPROXIMATED DENSITY DERIVATIVES OF THE PRESSURE                  SSPER
C     IN THE SATURATED STATES - IN REDUCED FORM .             94/12/01  SSPER
C     PL1R0 = D(P,L)/DRH * 1/(RGAS*T)                                   SSPER
C     PV1R0 = D(P,V)/DRH * 1/(RGAS*T)                                   SSPER
      IMPLICIT REAL*8(A-H,O-Z)                                          SSPER
      DIMENSION PUL(12),PUV(12)                                         SSPER
      COMMON /SATIF/  PS,TPS1T,TTPS2T,DL,DV,RL1T0,RV1T0,RHSL,PSL        SSPER
     =,RHSV,PSV,SL,GL,PL1R0,PL1T0,CVL,ETL,XLL,PV1R0,PV1T0,CVV,ETV,XLV   SSPER
```

```
      PARAMETER(TC=647.126D0,DC=322.D-3,PR0C=15.D0,GAM=1.24D0)          SSPER
      DATA PUL/91.68326854D-3,141.0358804D-3,293.745773D-0              SSPER
     =,-3.694718543D+3,35.3159487D+3,-224.5051913D+3,968.1021275D+3     SSPER
     =,-2.837321019D+6,5.560661663D+6,-6.973719089D+6,5.058540246D+6    SSPER
     =,-1.615244963D+6/                                                 SSPER
      DATA PUV/1.435429119D-3,3.738553917D-0,3.552096947D-0             SSPER
     =,-55.88507759D-0,217.5743793D-0,-535.2445068D-0,812.4224243D-0    SSPER
     =,-670.9285889D-0,229.0827484D-0,0.D0,0.D0,0.D0/                   SSPER
      IF(T .GE. TC)          THEN                                       SSPER
          SSPRHL = 0.D0                                                 SSPER
          RETURN                                                        SSPER
      ENDIF                                                             SSPER
      Y=(TC-T)/TC                                                       SSPER
      IF(T .LE. 620.D0)      THEN                                       SSPER
C     LOW TEMPERATURES                                                  SSPER
          PL1R0 = PUL(12)                                               SSPER
          PV1R0 = PUV(12)                                               SSPER
          DO 31 I=11,1,-1                                               SSPER
          PL1R0 = PL1R0*Y+PUL(I)                                        SSPER
   31     PV1R0 = PV1R0*Y+PUV(I)                                        SSPER
      ELSE                                                              SSPER
C     NEAR CRITICAL TEMPERATURES                                        SSPER
          YGA = Y**GAM                                                  SSPER
          FIL = PR0C                                                    SSPER
     =    -Y*(79.73057836D-0+Y*(430.1476885D-0-Y*14139.7632D+0))        SSPER
          FIV = PR0C                                                    SSPER
     =    +Y*(369.6750971D-0-Y*(5110.089071D-0-Y*46121.0908D-0))        SSPER
          PL1R0 = (DL/DC)*FIL*YGA                                       SSPER
          PV1R0 = (DV/DC)*FIV*YGA                                       SSPER
      ENDIF                                                             SSPER
      RETURN                                                            SSPER
      END                                                               SSPER
```

F.13 INR105.AQUA.FORT(SSPET)

```
      SUBROUTINE SSPET(T,PL1T0,PV1T0)                                   SSPET
C     SSPET RETURNS - FOR A GIVEN TEMPERATURE T -                       SSPET
C     APPROXIMATED TEMPERATURE DERIVATIVES OF THE PRESSURE              SSPET
C     IN THE SATURATED STATES - IN REDUCED FORM .              94/12/01 SSPET
C     PL1T0 = D(P,L)/DT * 1/(RGAS*RH,L)                                 SSPET
C     PV1T0 = D(P,V)/DT * 1/(RGAS*RH,V)                                 SSPET
      IMPLICIT REAL*8(A-H,O-Z)                                          SSPET
      DIMENSION PUL(12),PUV(12)                                         SSPET
      PARAMETER(TC=647.126D0,PT0C=1.80215732D-0)                        SSPET
      DATA PUL/1.818884544D-0,16.40737736D-0,-65.85455759D-0            SSPET
     =,52.04371914D-0,4.031238729D+3,-47.2570313D+3,277.4234268D+3      SSPET
     =,-988.3324003D+3,2.218919758D+6,-3.077677674D+6,2.414555699D+6    SSPET
     =,-821.0002244D+3/                                                 SSPET
      DATA PUV/2.029700729D-0,-125.5470649D-3,-61.54407176D-0           SSPET
     =,544.8751175D-0,-2.848188899D+3,9.708977783D+3,-21.26251929D+3    SSPET
     =,28.71096533D+3,-21.75527246D+3,7.082224609D+3,0.D0,0.D0/         SSPET
      IF(T .GE. TC)          THEN                                       SSPET
          PL1T0 = PT0C                                                  SSPET
```

```fortran
          PV1T0 = PT0C                                                  SSPET
          RETURN                                                        SSPET
      ENDIF                                                             SSPET
      Y=(TC-T)/TC                                                       SSPET
      IF(T .LE. 620.D0)      THEN                                       SSPET
C     LOW TEMPERATURES                                                  SSPET
          PL1T0 = PUL(12)                                               SSPET
          PV1T0 = PUV(12)                                               SSPET
          DO 31 I=11,1,-1                                               SSPET
          PL1T0 = PL1T0*Y+PUL(I)                                        SSPET
   31     PV1T0 = PV1T0*Y+PUV(I)                                        SSPET
      ELSE                                                              SSPET
C     NEAR CRITICAL TEMPERATURES                                        SSPET
          PL1T0 = PT0C                                                  SSPET
     =    + Y*(17.63135647D-0-Y*(97.95806903D-0-Y*451.8053522D-0))      SSPET
          PV1T0 = PT0C                                                  SSPET
     =    + Y*(16.00172011D-0-Y*(439.3063303D-0-Y*3359.729201D-0))      SSPET
      ENDIF                                                             SSPET
      RETURN                                                            SSPET
      END                                                               SSPET
```

F.14 INR105.AQUA.FORT(SSCEV)

```fortran
      SUBROUTINE SSCEV(T,CVL,CVV)                                       SSCEV
C     SSCEV RETURNS - FOR A GIVEN TEMPERATURE T - APPROXIMATED VALUES   SSCEV
C     OF THE CONSTANT VOLUME HEAT CAPACITY                              SSCEV
C     IN THE SATURATED LIQUID ( CVL ) AND                               SSCEV
C     IN THE SATURATED VAPOR  ( CVV ) IN RGAS UNITS.          94/12/01  SSCEV
      IMPLICIT REAL*8(A-H,O-Z)                                          SSCEV
      DIMENSION PUL(12),PUV(12)                                         SSCEV
      PARAMETER(TC=647.126D0,ALF=-1.0D-1)                               SSCEV
      DATA PUL/7.4305055D-0,-24.93618016D-0,195.5654567D-0              SSCEV
     =,1.986485797D+3,-53.30543411D+3,505.6971723D+3,-2.724774677D+6    SSCEV
     =,9.167737673D+6,-19.62203378D+6,25.98472533D+6,-19.41943135D+6    SSCEV
     =,6.263206554D+6/                                                  SSCEV
      DATA PUV/8.956404735D-0,-33.97230774D-0,126.5786602D-0            SSCEV
     =,-513.7462523D-0,1.521138693D+3,-2.182266721D+3,-73.57828067D-0   SSCEV
     =,4.483361889D+3,-5.407324042D+3,2.099821164D+3,0.D0,0.D0/         SSCEV
      IF(T .GE. TC)          THEN                                       SSCEV
          CVL = 0.D0                                                    SSCEV
          CVV = 0.D0                                                    SSCEV
          RETURN                                                        SSCEV
      ENDIF                                                             SSCEV
      Y = (TC-T)/TC                                                     SSCEV
      IF(T .LE. 620.D0)      THEN                                       SSCEV
C     LOW TEMPERATURES                                                  SSCEV
          CVL = PUL(12)                                                 SSCEV
          CVV = PUV(12)                                                 SSCEV
          DO 31 I=11,1,-1                                               SSCEV
          CVL = CVL*Y+PUL(I)                                            SSCEV
   31     CVV = CVV*Y+PUV(I)                                            SSCEV
      ELSE                                                              SSCEV
C     NEAR CRITICAL PART OF THE SATURATION LINE                         SSCEV
```

```
          BEG = Y**ALF                                                  SSCEV
          PSIL = 4.5D-0 +                                               SSCEV
     =         Y*(15.21452007D-0-Y*(167.2951053D-0-Y*1.042333618D+3))   SSCEV
          PSIV = 5.1D-0 +                                               SSCEV
     =         Y*(42.35451412D-0-Y*(1.002675527D+3-Y*6.911203183D+3))   SSCEV
          CVL = PSIL*BEG                                                SSCEV
          CVV = PSIV*BEG                                                SSCEV
      ENDIF                                                             SSCEV
      RETURN                                                            SSCEV
      END                                                               SSCEV
```

F.15 INR105.AQUA.FORT(SSGIS)

```
      SUBROUTINE SSGIS(T,SL,GL)                                         SSGIS
C     SSGIS RETURNS APPROXIMATED VALUES FOR THE ENTROPY AND FOR         SSGIS
C     THE GIBBS-FUNCTIOM OF THE SATURATED LIQUID                        SSGIS
C     AS FUNCTIONS OF THE INPUT TEMPERATURE T.                          SSGIS
C     SL = S.LIQUID/RGAS , GL = G.LIQUID/(RGAS*T)              94/06/10 SSGIS
      IMPLICIT REAL*8(A-H,O-Z)                                          SSGIS
      DIMENSION PSL(9),PGL(9)                                           SSGIS
      PARAMETER(TC=647.126D0,SC=9.430D0,BR=0.25D0,BET=0.325D0)          SSGIS
      DATA PSL/952.8621116D-3,272.5494325D-3,-8.457375782D-0            SSGIS
     =,44.57631748D-0,-150.7441559D-0,311.7914429D-0,-391.8528748D-0    SSGIS
     =,273.5529633D-0,-82.2848053D-0/                                   SSGIS
      DATA PGL/-2.569516966D-0,5.187876405D-0,640.8677408D-3            SSGIS
     =,1.374106295D-0,-18.01831251D-0,68.53337198D-0,-176.2024879D-0    SSGIS
     =,226.296896D-0,-130.4997179D-0/                                   SSGIS
      IF(T .GE. TC)          THEN                                       SSGIS
          SL = 0.D0                                                     SSGIS
          GL = 0.D0                                                     SSGIS
          RETURN                                                        SSGIS
      ENDIF                                                             SSGIS
      Y = (TC-T)/TC                                                     SSGIS
C     ENTROPY                                                           SSGIS
      IF(T .LE. 620.D0)      THEN                                       SSGIS
C     LOW TEMPERATURES                                                  SSGIS
          W = SQRT(Y)                                                   SSGIS
          OMV = PSL(9)                                                  SSGIS
          DO 31 K = 8,1,-1                                              SSGIS
   31     OMV = OMV*W+PSL(K)                                            SSGIS
      ELSE                                                              SSGIS
C     NEAR CRITICAL TEMPERATURES                                        SSGIS
          BEG = BR*Y**BET                                               SSGIS
          AV = -Y*(1.23383958D-0+Y*(1.656615363D-0-Y*Y*911.5375812D-0)) SSGIS
          OMV = 1.D0 - BEG + AV                                         SSGIS
      ENDIF                                                             SSGIS
      SL = OMV*SC                                                       SSGIS
C     GIBBS-FUNCTION                                                    SSGIS
      GL = PGL(9)                                                       SSGIS
      DO 71 K = 8,1,-1                                                  SSGIS
   71 GL = GL*Y+PGL(K)                                                  SSGIS
      RETURN                                                            SSGIS
      END                                                               SSGIS
```

F.16 INR105.AQUA.FORT(SSETA)

```
      SUBROUTINE SSETA(T,ETL,ETV)                                       SSETA
C     SSETA RETURNS - FOR A GIVEN TEMPERATURE T - APPROXIMATED VALUES   SSETA
C     OF THE DYNAMIC VISCOSITY                                          SSETA
C     IN THE SATURATED LIQUID ( ETL ) AND                               SSETA
C     IN THE SATURATED VAPOR  ( ETV ) IN PA*SEC.               94/12/01 SSETA
      IMPLICIT REAL*8(A-H,O-Z)                                          SSETA
      DIMENSION PUL(11),PUV(11)                                         SSETA
      PARAMETER(TC=647.126D0,ETC=3.95D-5,BET=0.325D0,BE=1.50D0)         SSETA
      DATA PUL/945.822019D-3,-3.50567042D-0,23.61506749D-0              SSETA
     =,-176.2217806D-0,960.5599634D-0,-3.533866457D+3,8.609716158D+3    SSETA
     =,-13.6377187D+3,13.43590472D+3,-7.453116048D+3,1.776272412D+3/    SSETA
      DATA PUV/973.3280423D-3,-4.061585251D-0,20.72025593D-0            SSETA
     =,-80.06616536D-0,214.64856D-0,-384.6194719D-0,433.0818384D-0      SSETA
     =,-278.1018786D-0,78.25010044D-0,0.D0,0.D0/                        SSETA
      DATA UN/1.D0/                                                     SSETA
      IF(T .GE. TC)          THEN                                       SSETA
          ETL=ETC                                                       SSETA
          ETV=ETC                                                       SSETA
          RETURN                                                        SSETA
      ENDIF                                                             SSETA
      Y = (TC-T)/TC                                                     SSETA
      CHI = T/(T-TC)                                                    SSETA
      IF(T .LE. 620.D0)      THEN                                       SSETA
C     LOW TEMPERATURES                                                  SSETA
          W = SQRT(Y)                                                   SSETA
          FL = PUL(11)                                                  SSETA
          FV = PUV(11)                                                  SSETA
          DO 31 K = 10,1,-1                                             SSETA
          FL = FL*W+PUL(K)                                              SSETA
   31     FV = FV*W+PUV(K)                                              SSETA
C     VISCOSITY OF THE LIQUID  ( IN PA*SEC )                            SSETA
          ETL=ETC/FL                                                    SSETA
C     VISCOSITY OF THE VAPOR   ( IN PA*SEC )                            SSETA
          ETV=ETC*FV                                                    SSETA
C     NEAR CRITICAL TEMPERATURES                                        SSETA
      ELSE                                                              SSETA
C     NEAR CRITICAL TEMPERATURES                                        SSETA
          BEG = BE*Y**BET                                               SSETA
          AL = Y*(4.669845392D-0-Y*(19.58930391D-0-Y*Y*1.768652432D+3)) SSETA
          ETL=ETC*( UN + BEG + AL )                                     SSETA
          AV = Y*(4.262680597D-0-Y*(34.89896734D-0-Y*Y*2.452105565D+3)) SSETA
          ETV=ETC*( UN - BEG + AV )                                     SSETA
      ENDIF                                                             SSETA
      RETURN                                                            SSETA
      END                                                               SSETA
```

F.17 INR105.AQUA.FORT(SSLAM)

```
      SUBROUTINE SSLAM(T,XLL,XLV)                                       SSLAM
C     SSLAM RETURNS - FOR A GIVEN TEMPERATURE T - APPROXIMATED VALUES   SSLAM
C     OF THE THERMAL CONDUCTIVITY                                       SSLAM
C     IN THE SATURATED LIQUID ( XLL ) AND                               SSLAM
C     IN THE SATURATED VAPOR  ( XLV ) IN W/(M*K).           94/12/01  SSLAM
      IMPLICIT REAL*8(A-H,O-Z)                                          SSLAM
      DIMENSION PUL(9),PUV(9)                                           SSLAM
      PARAMETER(TC=647.126D0,XLC=1.D+70,ALA=-0.50D0)                    SSLAM
      DATA PUL/397.252283D-3,1.361430056D-0,288.6579476D-3              SSLAM
     =,-1.137153088D-0,-75.20016424D-0,333.88123D-0,-546.0632435D-0     SSLAM
     =,311.575529D-0,0.D0/                                              SSLAM
      DATA PUV/201.2600289D-3,-2.546881317D-0,19.24073027D-0            SSLAM
     =,-79.8925815D-0,178.6322576D-0,-195.5137779D-0,62.06051011D-0     SSLAM
     =,38.86696488D-0,-15.11624233D-0/                                  SSLAM
      IF(T .GE. TC)          THEN                                       SSLAM
          XLL=XLC                                                       SSLAM
          XLV=XLC                                                       SSLAM
          RETURN                                                        SSLAM
      ENDIF                                                             SSLAM
      Y=(TC-T)/TC                                                       SSLAM
      IF(T .LE. 620.D0)      THEN                                       SSLAM
C     LOW TEMPERATURES                                                  SSLAM
          XLL = PUL(9)                                                  SSLAM
          XLV = PUV(9)                                                  SSLAM
          DO 31 I=8,1,-1                                                SSLAM
          XLL = XLL*Y+PUL(I)                                            SSLAM
   31     XLV = XLV*Y+PUV(I)                                            SSLAM
      ELSE                                                              SSLAM
C     NEAR CRITICAL TEMPERATURES                                        SSLAM
          BEG=Y**ALA                                                    SSLAM
          AL =  20.D-3 +Y*( 2.484618723D-0 -Y*( 31.16080409D-0          SSLAM
     =      -Y*( 421.0825394D-0 -Y*2390.60876D-0 )))                    SSLAM
          XLL=BEG*AL                                                    SSLAM
          AV =  20.D-3 +Y*( 722.1916099D-3 -Y*( 29.16989349D-0          SSLAM
     =      -Y*( 495.5936588D-0 -Y*3169.252794D-0 )))                   SSLAM
          XLV=BEG*AV                                                    SSLAM
      ENDIF                                                             SSLAM
      RETURN                                                            SSLAM
      END                                                               SSLAM
```

F.18 INR105.AQUA.FORT(DILE)

```
      FUNCTION DILE(Y)                                                  DILE
C     RETURNS AN ADDITIVE TERM FOR PSL(T) TO DESCRIBE THE PRESSURE      DILE
C     OF THE LIQUID SPINODAL ON A HGK-ISOTHERM.                94/07/12 DILE
C     DILE IS A FUNCTION OF THE SCALED TEMPERATURE  Y=1-T/TC            DILE
      DILE = (40.49363544D+0 -Y*(20.73773694D+3 -Y*(7.607260219D+6      DILE
     = -Y*(1.796520658D+9 -Y*(258.928784714D+9 -Y*(22.71838331D+12      DILE
     = -Y*(1.182066587D+15 -Y*(33.45912597D+15 -Y*396.4473916D+15       DILE
     = ))))))))*Y                                                       DILE
```

```
      RETURN                                                            DILE
      END                                                               DILE
```

F.19 INR105.AQUA.FORT(DIVE)

```
      FUNCTION DIVE(Y)                                                  DIVE
C     RETURNS AN ADDITIVE TERM FOR PSV(T) TO DESCRIBE THE PRESSURE      DIVE
C     OF THE VAPOR SPINODAL ON A HGK-ISOTHERM.                94/07/12  DIVE
C     DIVE IS A FUNCTION OF THE SCALED TEMPERATURE  Y=1-T/TC            DIVE
      DIVE = Y*(17.17830396D+0 -Y*(3.434969255D+3                       DIVE
     = -Y*(448.795607300D+3 -Y*(3.808457011D+7 -Y*(1.646319316D+9       DIVE
     = -Y*2.719071374D+10)))))                                          DIVE
      RETURN                                                            DIVE
      END                                                               DIVE
```

F.20 INR105.AQUA.FORT(PLMAX)

```
      FUNCTION PLMAX(T)                                                 PLMAX
C     PLMAX RETURNS THE PRESSURE IN PA-S ON THE WHOLE ISOCHORE          PLMAX
C     RHO = 1200 KG/M**3                                                PLMAX
C     AS FUNCTION OF THE INPUT TEMPERATURE T                  95/01/18  PLMAX
      IMPLICIT REAL*8(A-H,O-Z)                                          PLMAX
      IF(T .LE. 373.15D0)    THEN                                       PLMAX
          PLMAX =                  128.0317635D+9 -T*( 1.503427271D+9   PLMAX
     =    -T*( 6.608885188D+6 -T*( 12.83080180D+3 -T*9.305682808D-0 ))) PLMAX
      ELSE IF(T .GT. 373.15D0 .AND. T .LE. 647.126D0)        THEN       PLMAX
          PLMAX =                 1.121372093D+9 -T*( 6.278371391D+6    PLMAX
     =   -T*( 26.44577782D+3 -T*( 33.36474460D-0 -T*14.81761478D-3 )))  PLMAX
      ELSE IF(T .GT. 647.126D0)                  THEN                   PLMAX
          PLMAX =                  -731.6836907D+6  +T*( 6.597512847D+6 PLMAX
     =    -T*( 7.08407094D+3   -T*( 5.764846806D-0  -T*( 3.188721412D-3 PLMAX
     =    -T*( 1.193485372D-6  -T*( 2.946300976D-10 -T*( 4.575561580D-14PLMAX
     =    -T*( 4.040003535D-18 -T*1.543953233D-22 )))))))               PLMAX
      ENDIF                                                             PLMAX
      RETURN                                                            PLMAX
      END                                                               PLMAX
```

F.21 INR105.AQUA.FORT(P1200)

```
      FUNCTION P800(T)                                                  P800
C     P800 RETURNS THE PRESSURE IN PA-S ON THE SUPERCRITICAL PART       P800
C     OF THE ISOCHORE  RHO = 800 KG/M**3                                P800
C     AS FUNCTION OF THE INPUT TEMPERATURE T                  95/01/18  P800
      P800 =                    -576.9907617D+6 +T*( 229.0441320D+3     P800
     =  +T*( 3.275463692D+3 -T*( 4.331244582D-0 -T*( 3.008662103D-3     P800
     =  -T*( 1.272178994D-6 -T*( 337.938642D-12 -T*( 55.07102702D-15    P800
     =  -T*( 5.030253110D-18 -T*197.1414663D-24 ))))))))                P800
      RETURN                                                            P800
      END                                                               P800
```

F.22 INR105.AQUA.FORT(P600)

```
      FUNCTION P600(T)                                                  P600
C     P600 RETURNS THE PRESSURE IN PA-S ON THE SUPERCRITICAL PART       P600
C     OF THE ISOCHORE  RHO = 600 KG/M**3                                P600
C     AS FUNCTION OF THE INPUT TEMPERATURE T                   95/01/18 P600
      P600 =                    -193.5182531D+6  -T*( 662.0824295D+3    P600
     = -T*( 3.041300733D+3  -T*( 3.331588396D-0  -T*( 2.091973967D-3    P600
     = -T*( 822.3881849D-9  -T*( 205.8600153D-12 -T*( 31.88132588D-15   P600
     = -T*( 2.784677333D-18 -T*104.8772468D-24 ))))))))                 P600
      RETURN                                                            P600
      END                                                               P600
```

F.23 INR105.AQUA.FORT(P400)

```
      FUNCTION P400(T)                                                  P400
C     P400 RETURNS THE PRESSURE IN PA-S ON THE SUPERCRITICAL PART       P400
C     OF THE ISOCHORE  RHO = 400 KG/M**3                                P400
C     AS FUNCTION OF THE INPUT TEMPERATURE T                   95/01/18 P400
      P400 =                    -48.22391500D+6  -T*( 534.0793594D+3    P400
     = -T*( 1.893391113D+3  -T*( 2.052518308D-0  -T*( 1.304520061D-3    P400
     = -T*( 521.2344512D-9  -T*( 132.5272444D-12 -T*( 20.80959374D-15   P400
     = -T*( 1.839373352D-18 -T* 69.98460931D-24 ))))))))                P400
      RETURN                                                            P400
      END                                                               P400
```

F.24 INR105.AQUA.FORT(P322)

```
      FUNCTION P322(T)                                                  P322
C     P322 RETURNS THE PRESSURE IN PA-S ON THE SUPERCRITICAL PART       P322
C     OF THE CRITICAL ISOCHORE                                          P322
C     AS FUNCTION OF THE INPUT TEMPERATURE T                   95/01/18 P322
      P322 =                    -115.3411242D+6  -T*( 14.61390831D+3    P322
     = -T*( 713.8479434D-0  -T*( 843.0259976D-3  -T*( 558.6511529D-6    P322
     = -T*( 228.9982666D-9  -T*( 59.25205508D-12 -T*( 9.423230731D-15   P322
     = -T*( 841.0373279D-21 -T*32.24390358D-24  ))))))))                P322
      RETURN                                                            P322
      END                                                               P322
```

F.25 INR105.AQUA.FORT(P250)

```
      FUNCTION P250(T)                                                  P250
C     P250 RETURNS THE PRESSURE IN PA-S ON THE SUPERCRITICAL PART       P250
C     OF THE ISOCHORE  RHO = 250 KG/M**3                                P250
C     AS FUNCTION OF THE INPUT TEMPERATURE T                  95/01/18  P250
      P250 =  -161.604255D+6 +T*( 370.3861090D+3 -T*( 183.5936079D+0    P250
     = -T*( 93.96635527D-3 -T*( 30.07808886D-6 -T*( 5.819644183D-9      P250
     = -T*( 619.724735D-15 -T*27.81950727D-18 ))))))                    P250
      RETURN                                                            P250
      END                                                               P250
```

Appendix G. Secondary routines

The following routines either organize the calculation of the thermal properties of the water, or help to find a density to a given pressure.

G.1 *INR105.AQUA.FORT(DGFIND)*

```
      SUBROUTINE DGFIND(T,Q,DX,IECCH)                                   DGFIND
C     DGFIND SEEKS A DENSITY DX (G/CM**3)                               DGFIND
C     CORRESPONDING TO THE PRESSURE Q(MPA) AND                          DGFIND
C     TO THE - SUPERCRITICAL - TEMPERATURE T (K).             94/12/06  DGFIND
      IMPLICIT REAL*8(A-H,O-Z)                                          DGFIND
      COMMON /TEMPO/  RT,B1,B2,B1T,B2T,B1TT,B2TT,TV,GI,SI,CVI           DGFIND
      COMMON /BASIF/  GB,SB,ZB,PB1T0,PB1R0,CVB                          DGFIND
      COMMON /RESEF/  GR,SR,ZR,PR1T0,PR1R0,CVR                          DGFIND
      REAL*8 FX(3),X(3)                                                 DGFIND
      PARAMETER(DC=0.322D0,DMAX=1.2D0,FINIS=5.D-5)                      DGFIND
      DATA PF/1.D-6/,ZERO/0.D0/                                         DGFIND
C     RANGE CONTROL                                                     DGFIND
      QSMAX=PLMAX(T)*PF                                                 DGFIND
      IF(Q .GT. QSMAX)       IECCH=+2                                   DGFIND
      IF(IECCH .NE. 0)       GOTO 900                                   DGFIND
      QSMIN=P600(T)*PF                                                  DGFIND
      IF(Q .LT. QSMIN)       GO TO 200                                  DGFIND
C     REGION OF HIGH DENSITIES                                          DGFIND
  100 X(3)=DMAX                                                         DGFIND
      X(1)=0.6D0                                                        DGFIND
      X(2)=0.80D0                                                       DGFIND
      FX(2)=P800(T)*PF                                                  DGFIND
      SAM=2.D-1                                                         DGFIND
          GOTO 600                                                      DGFIND
C     REGION OF MODERATE DENSITIES                                      DGFIND
  200 X(3)=0.6D0                                                        DGFIND
      X(1)=DC                                                           DGFIND
      QSMAX=QSMIN                                                       DGFIND
      QSMIN=P322(T)*PF                                                  DGFIND
      IF(Q .LT. QSMIN)       GO TO 300                                  DGFIND
      X(2)=0.40D0                                                       DGFIND
      FX(2)=P400(T)*PF                                                  DGFIND
      SAM=1.D-1                                                         DGFIND
          GOTO 600                                                      DGFIND
C     REGION OF LOW DENSITIES                                           DGFIND
  300 X(3)=DC                                                           DGFIND
      X(1)=ZERO                                                         DGFIND
      QSMAX=QSMIN                                                       DGFIND
      QSMIN=ZERO                                                        DGFIND
      IF(Q .LT. QSMIN)       IECCH=-2                                   DGFIND
      IF(IECCH .NE. 0)       GOTO 900                                   DGFIND
      X(2)=0.25D0                                                       DGFIND
      FX(2)=P250(T)*PF                                                  DGFIND
      SAM=5.D-2                                                         DGFIND
  600 FX(1)=QSMIN                                                       DGFIND
```

```
      FX(3)=QSMAX                                                       DGFIND
C     WRITE(6,4400) QSMIN,QSMAX,SAM                                     DGFIND
C     FIRST DENSITY APPROXIMATIONS                                      DGFIND
      LI=1                                                              DGFIND
C     WRITE(6,6660)                                                     DGFIND
      IF(DX .LE. ZERO)       DX=DINSU(T,Q,FX,X,LI,IECCH)                DGFIND
      IF(IECCH .NE. 0)       RETURN                                     DGFIND
C     REFINED DENSITY APPROXIMATIONS IN THE MANNER OF MR. I. NEWTON     DGFIND
C     WRITE(6,'( )')                                                    DGFIND
C     ITERATING                                                         DGFIND
      LF=LI+19                                                          DGFIND
      DO 71 L=LI,LF                                                     DGFIND
      IF(DX .GT. DMAX)       THEN                                       DGFIND
          DX=DMAX                                                       DGFIND
          IECCH=2                                                       DGFIND
      ENDIF                                                             DGFIND
      IF(DX .LE. ZERO)       DX=Q/RT                                    DGFIND
      CALL BASE(T,DX,IECCH)                                             DGFIND
      CALL RESID(T,DX)                                                  DGFIND
      ZP = ZB + ZR                                                      DGFIND
      QAX = ZP*DX*RT                                                    DGFIND
      DELQ=Q-QAX                                                        DGFIND
      EPS=ABS(DELQ/Q)                                                   DGFIND
      IF(EPS .LT. FINIS)     THEN                                       DGFIND
          IECCH=0                                                       DGFIND
          RETURN                                                        DGFIND
      ENDIF                                                             DGFIND
      IF(IECCH .NE. 0)       GOTO 900                                   DGFIND
      P1R0 = PB1R0+PR1R0                                                DGFIND
      DQD = P1R0*RT                                                     DGFIND
      DELR = DELQ/DQD                                                   DGFIND
C     WRITE(6,6666)  L,DX,QAX,DELQ,FINIS,DELR                           DGFIND
      IF(DELR .GT. ZERO)     DELR=MIN(DELR,SAM)                         DGFIND
      IF(DELR .LT. ZERO)     DELR=MAX(DELR,-SAM)                        DGFIND
      DX = DX + DELR                                                    DGFIND
   71 CONTINUE                                                          DGFIND
      IECCH=3                                                           DGFIND
      WRITE(6,9000)                                                     DGFIND
      RETURN                                                            DGFIND
  900 IF(IECCH .GT. 0)       WRITE(6,9100) Q                            DGFIND
      IF(IECCH .LT. 0)       WRITE(6,9300) Q                            DGFIND
      RETURN                                                            DGFIND
C4400 FORMAT(8H QSMIN =,F14.9,2X,7HQSMAX =,F14.9,2X,5HSAM =,2X,F7.3)     DGFIND
C6660 FORMAT(3H  L,5X,2HDX,12X,3HQAX,13X,4HDELQ,11X,3HQGR,7X,4HDELR)     DGFIND
C6666 FORMAT(I3,F13.9,G18.9,G15.6,F9.5,F13.9)                           DGFIND
 9000 FORMAT(41H ITERATION FOR THE DENSITY VALUE FAILED ! )             DGFIND
 9100 FORMAT(13H PRESSURE P =,G15.7,16H MPA TOO HIGH !   )              DGFIND
 9300 FORMAT(13H PRESSURE P =,G15.7,16H MPA TOO LOW  !   )              DGFIND
      END                                                               DGFIND
```

G.2 INR105.AQUA.FORT(DLFIND)

```
      SUBROUTINE DLFIND(T,Q,DX,IECCH)                                   DLFIND
C     DLFIND SEEKS A LIQUID DENSITY DX (G/CM**3)                        DLFIND
C     CORRESPONDING TO THE PRESSURE Q(MPA) AND                          DLFIND
C     TO THE - SUBCRITICAL - TEMPERATURE T (K).               94/12/08  DLFIND
      IMPLICIT REAL*8(A-H,O-Z)                                          DLFIND
      COMMON /TEMPO/  RT,B1,B2,B1T,B2T,B1TT,B2TT,TV,GI,SI,CVI           DLFIND
      COMMON /BASIF/  GB,SB,ZB,PB1T0,PB1R0,CVB                          DLFIND
      COMMON /RESEF/  GR,SR,ZR,PR1T0,PR1R0,CVR                          DLFIND
      COMMON /SATIF/  PS,TPS1T,TTPS2T,DL,DV,RL1T0,RV1T0,RHSL,PSL        DLFIND
     =,RHSV,PSV,SL,GL,PL1R0,PL1T0,CVL,ETL,XLL,PV1R0,PV1T0,CVV,ETV,XLV   DLFIND
      REAL*8 FX(3),X(3)                                                 DLFIND
      PARAMETER(TC=647.126D0,TCN=646.0D0,DC=0.322D0)                    DLFIND
      PARAMETER(DMAX=1.2D0,FINIS=1.D-4)                                 DLFIND
      DATA ZERO/0.D0/,PF/1.D-6/                                         DLFIND
      CALL SUBCRI(T)                                                    DLFIND
C     RANGE CONTROL                                                     DLFIND
      QSV = PSV*PF                                                      DLFIND
      QSL = PSL*PF                                                      DLFIND
      IF(T .GE. 635.D0)       THEN                                      DLFIND
          Y=1-T/TC                                                      DLFIND
          QSL = QSL + DILE(Y)                                           DLFIND
          QSV = QSV - DIVE(Y)                                           DLFIND
          IF(QSL .GE. QSV)    QSL=QSV-FINIS                             DLFIND
      ENDIF                                                             DLFIND
      QMAX = PLMAX(T)*PF                                                DLFIND
      IF(Q .GT. QSV*1.3D0)    SAM=2.D-1                                 DLFIND
      IF(Q .LE. QSV*1.3D0)    SAM=1.D-1                                 DLFIND
      DMI=RHSL*1.D-3                                                    DLFIND
      DS0=(DMI-DC)*5.D-2                                                DLFIND
C     VICINITY OF THE CRITICAL POINT ?                                  DLFIND
      IF(T .GE. TCN .AND. Q .LE. QSV*1.005D0)    DMI=DC*0.965D0         DLFIND
C     WRITE(6,4400) QSL,QSV,DMI,SAM                                     DLFIND
      IF(Q .GT. QMAX)         IECCH=+2                                  DLFIND
      IF(Q .LT. QSL)          IECCH=-2                                  DLFIND
      IF(IECCH .NE. 0)        GOTO 900                                  DLFIND
C     FIRST DENSITY APPROXIMATIONS                                      DLFIND
C     WRITE(6,6660)                                                     DLFIND
      LI=1                                                              DLFIND
      IF(DX .LE. ZERO)        THEN                                      DLFIND
          X(3)=DMAX                                                     DLFIND
          FX(3)=QMAX                                                    DLFIND
          X(2)=DL                                                       DLFIND
          FX(2)=PS*PF                                                   DLFIND
          X(1)=DMI                                                      DLFIND
          FX(1)=QSL                                                     DLFIND
          DX=DINLI(T,Q,FX,X,LI,IECCH)                                   DLFIND
          IF(IECCH .NE. 0)    RETURN                                    DLFIND
      ENDIF                                                             DLFIND
C     REFINED DENSITY APPROXIMATIONS IN THE MANNER OF MR. I. NEWTON     DLFIND
C     WRITE(6,'( )')                                                    DLFIND
      LF=LI+19                                                          DLFIND
      DO 71 L=LI,LF                                                     DLFIND
```

```
      IF(DX .GT. DMAX)      THEN                                        DLFIND
          DX=DMAX                                                       DLFIND
          IECCH=2                                                       DLFIND
      ENDIF                                                             DLFIND
      IF(DX .LT. DMI)       DX=DMI                                      DLFIND
      CALL BASE(T,DX,IECCH)                                             DLFIND
      CALL RESID(T,DX)                                                  DLFIND
      ZP = ZB + ZR                                                      DLFIND
      QAX = ZP*DX*RT                                                    DLFIND
      DELQ=Q-QAX                                                        DLFIND
      EPS=ABS(DELQ)                                                     DLFIND
      IF(EPS .LT. FINIS)    THEN                                        DLFIND
          IECCH=0                                                       DLFIND
          RETURN                                                        DLFIND
      ENDIF                                                             DLFIND
      IF(IECCH .GT. 0)      GO TO 900                                   DLFIND
      P1R0 = PB1R0+PR1R0                                                DLFIND
      DQD = P1R0*RT                                                     DLFIND
      IF(DQD .LE. ZERO)     THEN                                        DLFIND
          DX=DX+DS0                                                     DLFIND
          GO TO 71                                                      DLFIND
      ENDIF                                                             DLFIND
      DELR = DELQ/DQD                                                   DLFIND
C     WRITE(6,6666) L,DX,QAX,DELQ,FINIS,DELR                            DLFIND
      IF(ABS(DELR) .LT. 1.D-6)        THEN                              DLFIND
          IECCH=0                                                       DLFIND
          RETURN                                                        DLFIND
      ENDIF                                                             DLFIND
      IF(DELR .GT. ZERO)    DELR=MIN(DELR,SAM)                          DLFIND
      IF(DELR .LT. ZERO)    DELR=MAX(DELR,-SAM)                         DLFIND
      DX = DX + DELR                                                    DLFIND
   71 CONTINUE                                                          DLFIND
      IECCH=3                                                           DLFIND
  800 WRITE(6,9000)                                                     DLFIND
      RETURN                                                            DLFIND
  900 IF(IECCH .GT. 0)      WRITE(6,9100) Q                             DLFIND
      IF(IECCH .LT. 0)      WRITE(6,9300) Q                             DLFIND
      RETURN                                                            DLFIND
C4400 FORMAT(7H QSPL =,F14.9,3X,6HQSPV =,F14.9,3X,5HDMI =,3X,F10.6,3X    DLFIND
C    =,5HSAM =,F7.3 )                                                   DLFIND
C6660 FORMAT(3H  L,5X,2HDX,12X,3HQAX,13X,4HDELQ,11X,3HQGR,7X,4HDELR)     DLFIND
C6666 FORMAT(I3,F13.9,G18.9,G15.6,F9.5,F13.9)                            DLFIND
 9000 FORMAT(41H ITERATION FOR THE DENSITY VALUE FAILED !)              DLFIND
 9100 FORMAT(13H PRESSURE P =,G15.7,16H MPA TOO HIGH !    )             DLFIND
 9300 FORMAT(13H PRESSURE P =,G15.7,16H MPA TOO LOW  !    )             DLFIND
      END                                                               DLFIND
```

G.3 *INR105.AQUA.FORT(DVFIND)*

```
      SUBROUTINE DVFIND(T,Q,DX,IECCH)                                   DVFIND
C     DVFIND SEEKS A VAPOR DENSITY DX (G/CM**3)                         DVFIND
C     CORRESPONDING TO THE PRESSURE Q(MPA) AND                          DVFIND
C     TO THE - SUBCRITICAL - TEMPERATURE T (K).               94/12/08  DVFIND
      IMPLICIT REAL*8(A-H,O-Z)                                          DVFIND
      COMMON /TEMPO/  RT,B1,B2,B1T,B2T,B1TT,B2TT,TV,GI,SI,CVI           DVFIND
      COMMON /BASIF/  GB,SB,ZB,PB1T0,PB1R0,CVB                          DVFIND
      COMMON /RESEF/  GR,SR,ZR,PR1T0,PR1R0,CVR                          DVFIND
      COMMON /SATIF/  PS,TPS1T,TTPS2T,DL,DV,RL1T0,RV1T0,RHSL,PSL        DVFIND
     =,RHSV,PSV,SL,GL,PL1R0,PL1T0,CVL,ETL,XLL,PV1R0,PV1T0,CVV,ETV,XLV   DVFIND
      REAL*8 FX(3),X(3)                                                 DVFIND
      PARAMETER(TC=647.126D0,TCN=646.0D0,DC=0.322D0,FINIS=5.D-5)        DVFIND
      DATA ZERO/0.D0/,PF/1.D-6/                                         DVFIND
      CALL SUBCRI(T)                                                    DVFIND
C     RANGE CONTROL                                                     DVFIND
      QSL = PSL*PF                                                      DVFIND
      QSV = PSV*PF                                                      DVFIND
      IF(Q .LT. QSL*0.95D0)  SAM=1.D-1                                  DVFIND
      IF(Q .GE. QSL*0.95D0)  SAM=5.D-2                                  DVFIND
      DMX=RHSV*1.D-3                                                    DVFIND
C     VICINITY OF THE CRITICAL POINT ?                                  DVFIND
      IF(T .GE. TCN .AND. Q .GE. QSL*0.9999D0)   SAM=5.D-2              DVFIND
      IF(T .GE. 635.D0)      THEN                                       DVFIND
          Y=1-T/TC                                                      DVFIND
          QSV = QSV - DIVE(Y)                                           DVFIND
      ENDIF                                                             DVFIND
C     WRITE(6,4400) QSL,QSV,DMX,SAM                                     DVFIND
      IF(Q .GT. QSV)         IECCH=+2                                   DVFIND
      IF(Q .LT. ZERO)        IECCH=-2                                   DVFIND
      IF(IECCH .NE. 0)       GO TO 900                                  DVFIND
C     FIRST DENSITY APPROXIMATIONS                                      DVFIND
C     WRITE(6,6660)                                                     DVFIND
      LI=1                                                              DVFIND
      IF(DX .LE. ZERO)       THEN                                       DVFIND
          X(3)=DMX                                                      DVFIND
          FX(3)=QSV                                                     DVFIND
          X(2)=DV                                                       DVFIND
          FX(2)=PS*1.D-6                                                DVFIND
          X(1)=ZERO                                                     DVFIND
          FX(1)=ZERO                                                    DVFIND
          DX=DINVA(T,Q,FX,X,LI,IECCH)                                   DVFIND
          IF(IECCH .NE. 0)   RETURN                                     DVFIND
      ENDIF                                                             DVFIND
C     REFINED DENSITY APPROXIMATIONS IN THE MANNER OF MR. I. NEWTON     DVFIND
C     WRITE(6,'(   )' )                                                 DVFIND
      LF=LI+19                                                          DVFIND
      IF(DX .GT. DMX)        DX=DMX                                     DVFIND
      DO 71 L=LI,LF                                                     DVFIND
      IF(DX .LE. ZERO)       DX=Q/RT                                    DVFIND
      CALL BASE(T,DX,IECCH)                                             DVFIND
      CALL RESID(T,DX)                                                  DVFIND
      ZP = ZB + ZR                                                      DVFIND
```

```
      QAX = ZP*DX*RT                                                    DVFIND
      DELQ = Q-QAX                                                      DVFIND
      EPS=ABS(DELQ/Q)                                                   DVFIND
      IF(EPS .LT. FINIS)     THEN                                       DVFIND
          IECCH=0                                                       DVFIND
          RETURN                                                        DVFIND
      ENDIF                                                             DVFIND
      IF(IECCH .NE. 0)       GO TO 900                                  DVFIND
      P1R0 = PB1R0+PR1R0                                                DVFIND
      DQD = P1R0*RT                                                     DVFIND
      IF(DQD .LE. ZERO)      THEN                                       DVFIND
          IF(L .EQ. LI)      THEN                                       DVFIND
              DX=DMX                                                    DVFIND
          ELSE                                                          DVFIND
              DX=DV*0.9D0                                               DVFIND
          ENDIF                                                         DVFIND
          GO TO 71                                                      DVFIND
      ENDIF                                                             DVFIND
      DELR = DELQ/DQD                                                   DVFIND
C     WRITE(6,6666) L,DX,QAX,DELQ,FINIS,DELR                            DVFIND
      IF(DELR .GT. ZERO)     DELR=MIN(DELR,SAM)                         DVFIND
      IF(DELR .LT. ZERO)     DELR=MAX(DELR,-SAM)                        DVFIND
      DX = DX + DELR                                                    DVFIND
      IF(DX .GT. DMX)        THEN                                       DVFIND
          DX=DMX                                                        DVFIND
          IECCH=+2                                                      DVFIND
      ENDIF                                                             DVFIND
   71 CONTINUE                                                          DVFIND
      IECCH=3                                                           DVFIND
      WRITE(6,9000)                                                     DVFIND
      RETURN                                                            DVFIND
  900 IF(IECCH .GT. 0)       WRITE(6,9100) Q                            DVFIND
      IF(IECCH .LT. 0)       WRITE(6,9300) Q                            DVFIND
      RETURN                                                            DVFIND
C4400 FORMAT(7H QSPL =,F14.9,3X,6HQSPV =,F14.9,3X,5HDMX =,3X,F10.6,3X   DVFIND
C    =,5HSAM =,F7.3 )                                                   DVFIND
C6660 FORMAT(3H  L,5X,2HDX,12X,3HQAX,13X,4HDELQ,11X,3HQGR,7X,4HDELR)    DVFIND
C6666 FORMAT(I3,F13.9,G18.9,G15.6,F9.5,F13.9)                            DVFIND
 9000 FORMAT(41H ITERATION FOR THE DENSITY VALUE FAILED ! )             DVFIND
 9100 FORMAT(13H PRESSURE P =,G15.7,16H MPA TOO HIGH !   )              DVFIND
 9300 FORMAT(13H PRESSURE P =,G15.7,16H MPA TOO LOW  !   )              DVFIND
      END                                                               DVFIND
```

G.4 *INR105.AQUA.FORT(DINSU)*

```
      FUNCTION DINSU(T,Q,FX,X,IT,IECCH)                                 DINSU
C     DINSU IS AN INITAL DENSITY FOR THE ROUTINE DSFIND.                DINSU
C     DINSU IS THE ROOT OF A 2. GRADE EQUATION  Q.IS(DX)-Q = 0.         DINSU
C     PRESSURES ARE IN MPA-S, DENSITIES IN G/CM**3-S.          93/09/03 DINSU
      IMPLICIT REAL*8(A-H,O-Z)                                          DINSU
      COMMON /TEMPO/  RT,B1,B2,B1T,B2T,B1TT,B2TT,TV,GI,SI,CVI           DINSU
      COMMON /BASIF/  GB,SB,ZB,PT0B,PR0B,CVB                            DINSU
      COMMON /RESEF/  GR,SR,ZR,PT0R,PR0R,CVR                            DINSU
```

```
      REAL*8 FX(3),X(3)                                                  DINSU
      PARAMETER(TC2=648.126D0,DUG=0.122D0,DOG=0.522D0,DMAX=1.2D0)        DINSU
      DATA ZERO/0.D0/                                                    DINSU
      FINIS=1.D-2                                                        DINSU
C     ITERATING                                                          DINSU
C     QUADRATIC INTERPOLATION TO THE POINTS 1 2 3                        DINSU
   50 CALL QUADO(FX,X,Q,DX,JECCH)                                        DINSU
      IF(DX .GT. DMAX)       DX=DMAX                                     DINSU
      IF(DX .LE. ZERO)       DX=1.D-6                                    DINSU
C     SELECTING THE NEW PILLARS                                          DINSU
      IF(IT .EQ. 1)          THEN                                        DINSU
          IF(T .LT. TC2 .AND.                                            DINSU
     =    (DX .GT. DUG .AND. DX .LT. DOG))        FINIS=1.D-3            DINSU
      ELSE                                                               DINSU
          IF(IT .GT. 3 .OR. EPS .LT. FINIS)       GO TO 900              DINSU
      ENDIF                                                              DINSU
      KI=0                                                               DINSU
      CALL SORBET(KI,FX,X,Q,DX)                                          DINSU
      IF(KI .EQ. 2 .OR. JECCH .NE. 0)             GO TO 900              DINSU
C     PRESSURE AT THE NEW DENSITY VALUE                                  DINSU
   52 DELR=DX-DXA                                                        DINSU
      IF(ABS(DELR) .LT. 1.D-4)                    GO TO 900              DINSU
      CALL BASE(T,DX,IECCH)                                              DINSU
      CALL RESID(T,DX)                                                   DINSU
      ZP = ZB + ZR                                                       DINSU
      FX(KI)= ZP*DX*RT                                                   DINSU
      DELQ=FX(KI)-Q                                                      DINSU
      EPS=ABS(DELQ)                                                      DINSU
      DXA=DX                                                             DINSU
C     WRITE(6,6666) IT,DX,FX(KI),DELQ,FINIS,DELR                         DINSU
      IT=IT+1                                                            DINSU
          GO TO 50                                                       DINSU
  900 DINSU=DX                                                           DINSU
 6666 FORMAT(I3,F13.9,G18.9,G15.6,F9.5,F13.9)                            DINSU
  999 RETURN                                                             DINSU
      END                                                                DINSU
```

G.5 *INR105.AQUA.FORT(DINLI)*

```
      FUNCTION DINLI(T,Q,FX,X,IT,IECCH)                                  DINLI
C     DINLI IS AN INITAL DENSITY FOR THE ROUTINE DLFIND.                 DINLI
C     DINLI IS THE ROOT OF A 2. GRADE EQUATION  Q.IS(DX)-Q = 0           DINLI
C     PRESSURES ARE IN MPA-S, DENSITIES IN G/CM**3-S.          93/09/03  DINLI
      IMPLICIT REAL*8(A-H,O-Z)                                           DINLI
      COMMON /TEMPO/  RT,B1,B2,B1T,B2T,B1TT,B2TT,TV,GI,SI,CVI            DINLI
      COMMON /BASIF/  GB,SB,ZB,PT0B,PR0B,CVB                             DINLI
      COMMON /RESEF/  GR,SR,ZR,PT0R,PR0R,CVR                             DINLI
      REAL*8 FX(3),X(3)                                                  DINLI
      PARAMETER(TC1=646.126D0,DOG=0.522D0)                               DINLI
      FINIS=1.D-2                                                        DINLI
      DMX=X(3)                                                           DINLI
C     ITERATING                                                          DINLI
C     QUADRATIC INTERPOLATION TO THE POINTS 1 2 3                        DINLI
```

```
   50 CALL QUADO(FX,X,Q,DX,JECCH)                                       DINLI
      IF(DX .GT. DMX)        DX=DMX                                     DINLI
C     SELECTING THE NEW PILLARS                                         DINLI
      IF(IT .EQ. 1)          THEN                                       DINLI
          IF(T .GT. TC1)     THEN                                       DINLI
              KI=1                                                      DINLI
              IF(DX .LT. DOG)                    FINIS=1.D-3            DINLI
          ELSE                                                          DINLI
              KI=0                                                      DINLI
          ENDIF                                                         DINLI
      ENDIF                                                             DINLI
      CALL SORBET(KI,FX,X,Q,DX)                                         DINLI
      IF(KI .EQ. 2 .OR. JECCH .NE. 0)            GO TO 900              DINLI
      IF(IT .LE. 1)          GO TO 52                                   DINLI
      IF(IT .GT. 3 .OR. EPS .LT. FINIS)          GO TO 900              DINLI
C     PRESSURE AT THE NEW DENSITY VALUE                                 DINLI
   52 DELR=DX-DXA                                                       DINLI
      IF(ABS(DELR) .LT. 1.D-4)                   GO TO 900              DINLI
      CALL BASE(T,DX,IECCH)                                             DINLI
      CALL RESID(T,DX)                                                  DINLI
      ZP = ZB + ZR                                                      DINLI
      FX(KI)= ZP*DX*RT                                                  DINLI
      DELQ=FX(KI)-Q                                                     DINLI
      EPS=ABS(DELQ)                                                     DINLI
      DXA=DX                                                            DINLI
C     WRITE(6,6666) IT,DX,FX(KI),DELQ,FINIS,DELR                        DINLI
      IT=IT+1                                                           DINLI
          GO TO 50                                                      DINLI
  900 DINLI=DX                                                          DINLI
 6666 FORMAT(I3,F13.9,G18.9,G15.6,F9.5,F13.9)                           DINLI
      RETURN                                                            DINLI
      END                                                               DINLI
```

G.6 *INR105.AQUA.FORT(DINVA)*

```
      FUNCTION DINVA(T,Q,FX,X,IT,IECCH)                                 DINVA
C     DINVA IS AN INITAL DENSITY FOR THE ROUTINE DVFIND.                DINVA
C     DINVA IS THE ROOT OF A 2. GRADE EQUATION  Q.IS(DX)-Q = 0.         DINVA
C     PRESSURES ARE IN MPA-S, DENSITIES IN G/CM**3-S.         93/09/17  DINVA
      IMPLICIT REAL*8(A-H,O-Z)                                          DINVA
      COMMON /TEMPO/  RT,B1,B2,B1T,B2T,B1TT,B2TT,TV,GI,SI,CVI           DINVA
      COMMON /BASIF/  GB,SB,ZB,PT0B,PR0B,CVB                            DINVA
      COMMON /RESEF/  GR,SR,ZR,PT0R,PR0R,CVR                            DINVA
      REAL*8 FX(3),X(3)                                                 DINVA
      PARAMETER(TC1=646.126D0,DUG=0.122D0)                              DINVA
      FINIS=1.D-2                                                       DINVA
C     ITERATING                                                         DINVA
C     QUADRATIC INTERPOLATION TO THE POINTS 1 2 3                       DINVA
   50 CALL QUADO(FX,X,Q,DX,JECCH)                                       DINVA
      IF(DX .LT. 0.D0)       DX=1.D-6                                   DINVA
C     SELECTING THE NEW PILLARS                                         DINVA
      IF(IT .EQ. 1)          THEN                                       DINVA
          IF(T .GT. TC1)     THEN                                       DINVA
```

```
              KI=3                                                      DINVA
              IF(DX .GT. DUG)                   FINIS=1.D-3             DINVA
          ELSE                                                          DINVA
              KI=0                                                      DINVA
          ENDIF                                                         DINVA
      ENDIF                                                             DINVA
      CALL SORBET(KI,FX,X,Q,DX)                                         DINVA
      IF(KI .EQ. 2 .OR. JECCH .NE. 0)           GO TO 900               DINVA
      IF(IT .LE. 1)         GO TO 52                                    DINVA
      IF(IT .GT. 3 .OR. EPS .LT. FINIS)         GO TO 900               DINVA
C     PRESSURE AT THE NEW DENSITY VALUE                                 DINVA
   52 DELR=DX-DXA                                                       DINVA
      IF(ABS(DELR) .LT. 1.D-4)                  GO TO 900               DINVA
      CALL BASE(T,DX,IECCH)                                             DINVA
      CALL RESID(T,DX)                                                  DINVA
      ZP = ZB + ZR                                                      DINVA
      FX(KI)= ZP*DX*RT                                                  DINVA
      DELQ=FX(KI)-Q                                                     DINVA
      EPS=ABS(DELQ/Q)                                                   DINVA
      DXA=DX                                                            DINVA
C     WRITE(6,6666) IT,DX,FX(KI),DELQ,FINIS,DELR                        DINVA
      IT=IT+1                                                           DINVA
          GO TO 50                                                      DINVA
  900 DINVA=DX                                                          DINVA
 6666 FORMAT(I3,F13.9,G18.9,G15.6,F9.5,F13.9)                           DINVA
  999 RETURN                                                            DINVA
      END                                                               DINVA
```

G.7 *INR105.AQUA.FORT(QUADO)*

```
      SUBROUTINE QUADO(FX,X,FXS,XS,JECCH)                               QUADO
C     SEARCHES AN XS WITH THE PROPERTY FX(XS) = FXS           93/09/07  QUADO
C     BY FITTING A QUADRATIC POLYNOMIAL TO THE POINTS FX(1) - FX(3)     QUADO
      IMPLICIT REAL*8(A-H,O-Z)                                          QUADO
      REAL*8 FX(3),X(3),FY(3)                                           QUADO
      DATA ZERO,UN/0.D0,1.D0/                                           QUADO
      CALL MONIKA(FX,X,JECCH)                                           QUADO
      IF(JECCH .NE. 0)      RETURN                                      QUADO
      A=ZERO                                                            QUADO
      B=ZERO                                                            QUADO
      C=ZERO                                                            QUADO
      IECCH=0                                                           QUADO
C     THE COEFFICIENTS OF THE POLYNOMIAL                                QUADO
      DO 21 K=1,3                                                       QUADO
   21 FY(K)=FX(K)-FXS                                                   QUADO
      DO 31 I=1,3                                                       QUADO
      J=I+1                                                             QUADO
      IF(J .GT. 3)          J=J-3                                       QUADO
      XD=X(J)-X(I)                                                      QUADO
      K=J+1                                                             QUADO
      IF(K .GT. 3)          K=K-3                                       QUADO
      YXD= FY(K)*XD                                                     QUADO
      YX2=-YXD*(X(J)+X(I))                                              QUADO
```

```
      YX3= YXD*X(J)*X(I)                                                QUADO
      A=A+YXD                                                           QUADO
      B=B+YX2                                                           QUADO
      C=C+YX3                                                           QUADO
   31 CONTINUE                                                          QUADO
C     THE ROOTS OF THE POLYNOMIAL                                       QUADO
      XG=-B/(A+A)                                                       QUADO
      US=UN-C/(A*XG*XG)                                                 QUADO
      IF(US .LE. ZERO)       THEN                                       QUADO
          XS=XG                                                         QUADO
          RETURN                                                        QUADO
      ENDIF                                                             QUADO
      W=SQRT(US)                                                        QUADO
      XA=XG*(UN+W)                                                      QUADO
      XB=XG*(UN-W)                                                      QUADO
C     THE APPROPRIATE ROOT                                              QUADO
   80 XM=(X(1)+X(2)+X(3))/3.D0                                          QUADO
      XS=XA                                                             QUADO
      IF(ABS(XB-XM) .LT. ABS(XA-XM))             XS=XB                  QUADO
      RETURN                                                            QUADO
      END                                                               QUADO
```

G.8 INR105.AQUA.FORT(MONIKA)

```
      SUBROUTINE MONIKA(FX,X,JECCH)                                     MONIKA
C     ARRANGES THE POINTS FX,X TO A SET WITH                            MONIKA
C     MONOTONOUS ABSCISSAS                                    93/09/07  MONIKA
      IMPLICIT REAL*8(A-H,O-Z)                                          MONIKA
      DIMENSION X(3),FX(3),W(3),FW(3)                                   MONIKA
      IF((X(1) .EQ. X(2)) .OR. (X(2) .EQ. X(3)) .OR.                    MONIKA
     = (X(1) .EQ. X(3)))     THEN                                       MONIKA
          JECCH = 1                                                     MONIKA
          RETURN                                                        MONIKA
      ENDIF                                                             MONIKA
      U=MIN(X(1),X(2),X(3))                                             MONIKA
      IF(U .EQ. X(1))        I=1                                        MONIKA
      IF(U .EQ. X(2))        I=2                                        MONIKA
      IF(U .EQ. X(3))        I=3                                        MONIKA
      J=I+1                                                             MONIKA
      IF(J .GT. 3)           J=J-3                                      MONIKA
      K=J+1                                                             MONIKA
      IF(K .GT. 3)           K=K-3                                      MONIKA
      V=MIN(X(J),X(K))                                                  MONIKA
      IF(V .EQ. X(J))        GO TO 90                                   MONIKA
      JF=J                                                              MONIKA
      J=K                                                               MONIKA
      K=JF                                                              MONIKA
      GO TO 92                                                          MONIKA
   90 IF(I .EQ. 1)           RETURN                                     MONIKA
   92 W(1)=X(I)                                                         MONIKA
      FW(1)=FX(I)                                                       MONIKA
      W(2)=X(J)                                                         MONIKA
      FW(2)=FX(J)                                                       MONIKA
```

```
      W(3)=X(K)                                                         MONIKA
      FW(3)=FX(K)                                                       MONIKA
      DO 91 N=1,3                                                       MONIKA
      X(N)=W(N)                                                         MONIKA
   91 FX(N)=FW(N)                                                       MONIKA
      END                                                               MONIKA
```

G.9 INR105.AQUA.FORT(SORBET)

```
      SUBROUTINE SORBET(KI,FX,X,FXS,XS)                                 SORBET
C     REPLACES ONE OF THE BOUNDARY POINTS X(1) OR X(3) WITH XS.         SORBET
C     NORMALLY THE POINT WITH THE LARGER (FXS-FX) WILL BE REPLACED.     SORBET
C     IF 2 OF THE 3 POINTS ARE VERY NEAR TO EACH OTHER                  SORBET
C     ( X(K)/X(L) < 1.05 ), THEN AT FIRST ONE OF THESE POINTS WELL BE   SORBET
C     REPLACED.                                               93/07/09  SORBET
      IMPLICIT REAL*8(A-H,O-Z)                                          SORBET
      DIMENSION X(3),FX(3)                                              SORBET
C     ------------------------------------------------------------------SORBET
      IF(KI .NE. 0)          GO TO 10                                   SORBET
      KI=1                                                              SORBET
      D3=ABS(FX(3)-FXS)                                                 SORBET
      D1=ABS(FX(1)-FXS)                                                 SORBET
      IF(D3 .GT. D1)         KI=3                                       SORBET
C     SORTING                                                           SORBET
C     FIRST POINT DISCARDED                                             SORBET
   10 IF(KI .EQ. 3)          GO TO 50                                   SORBET
      IF(XS .GT. X(2))       GO TO 20                                   SORBET
      X(KI) = XS                                                        SORBET
      FX(KI) = FXS                                                      SORBET
          GO TO 90                                                      SORBET
   20 X(1) = X(2)                                                       SORBET
      FX(1) = FX(2)                                                     SORBET
      IF(XS .GT. X(3))       GO TO 30                                   SORBET
      KI=2                                                              SORBET
      X(KI) = XS                                                        SORBET
      FX(KI) = FXS                                                      SORBET
          GO TO 90                                                      SORBET
   30 X(2) = X(3)                                                       SORBET
      FX(2) = FX(3)                                                     SORBET
      KI=3                                                              SORBET
      X(KI) = XS                                                        SORBET
      FX(KI) = FXS                                                      SORBET
          GO TO 90                                                      SORBET
C     LAST POINT DISCARDED                                              SORBET
   50 IF(XS .LT. X(2))        GO TO 60                                  SORBET
      X(KI) = XS                                                        SORBET
      FX(KI) = FXS                                                      SORBET
          GO TO 90                                                      SORBET
   60 X(3) = X(2)                                                       SORBET
      FX(3) = FX(2)                                                     SORBET
      IF(XS .LT. X(1))        GO TO 70                                  SORBET
      KI=2                                                              SORBET
      X(KI) = XS                                                        SORBET
```

```
      FX(KI) = FXS                                                      SORBET
         GO TO 90                                                       SORBET
   70 X(2) = X(1)                                                       SORBET
      FX(2) = FX(1)                                                     SORBET
      KI=1                                                              SORBET
      X(KI) = XS                                                        SORBET
      FX(KI) = FXS                                                      SORBET
   90 RETURN                                                            SORBET
      END
```

G.10 INR105.AQUA.FORT(TCORR)

```
      SUBROUTINE TCORR(TS,Q,DLK,DVK,IECCH)                              TCORR
C     RETURNS FOR A GIVEN Q ( IN MPA ) THE CORRESPONDING SATURATION     TCORR
C     TEMPERATURE TS AND THE SATURATED DENSITIES                        TCORR
C     DLK, DVK ( IN G/CM**3 ).                                 94/11/30 TCORR
      IMPLICIT REAL*8 (A-H,O-Z)                                         TCORR
      COMMON /SATIF / PS,TPS1T,TTPS2T,DL,DV,RL1T0,RV1T0,RHSL,PSL        TCORR
     =,RHSV,PSV,SL,GL,PL1R0,PL1T0,CVL,ETL,XLL,PV1R0,PV1T0,CVV,ETV,XLV   TCORR
      COMMON /TCOFF / DQ,DTS,QC                                         TCORR
      PARAMETER(TCN=646.0D0,QCN=21.83815D0,FINIS=1.D-5)                 TCORR
C     FIRST APPROXIMATE VALUES OF TS AND DTS/DQ                         TCORR
      P=Q*1.D6                                                          TCORR
      CALL TS1(P,TS,TS1P,IECCH)                                         TCORR
      TS1Q=TS1P*1.D6                                                    TCORR
      IF(IECCH .NE. 0)       RETURN                                     TCORR
C     ITERATING FOR CORRECTED TEMPERATURES                              TCORR
      DO 22 I=1,5                                                       TCORR
      CALL ONLYT(TS,IECCH)                                              TCORR
      IF(IECCH .NE. 0)       RETURN                                     TCORR
      CALL SUBCRI(TS)                                                   TCORR
      DLK=DL                                                            TCORR
      DVK=DV                                                            TCORR
      QC=Q                                                              TCORR
      IF(Q .GT. QCN .OR. TS .GE. TCN)             RETURN                TCORR
      CALL GPCORR(TS,QC,DLK,DVK,IECCH)                                  TCORR
      IF(IECCH .NE. 0)       RETURN                                     TCORR
      DQ=Q-QC                                                           TCORR
      DTS=DQ*TS1Q                                                       TCORR
      IF(ABS(DTS) .LT. FINIS)                     RETURN                TCORR
   22 TS = TS+DTS                                                       TCORR
      RETURN                                                            TCORR
      END                                                               TCORR
```

G.11 INR105.AQUA.FORT(GPCORR)

```
      SUBROUTINE GPCORR(T,Q,DLK,DVK,IECCH)                              GPCORR
C     RETURNS FOR A GIVEN T THE SATURATED PRESSURE Q ( IN MPA )         GPCORR
C     AND THE SATURATED DENSITIES DLK, DVK ( IN G/CM**3 )               GPCORR
C     CORRECTED FOR G,L = G,V AND Q,L = Q,V                  94/11/30   GPCORR
      IMPLICIT REAL*8(A-H,O-Z)                                          GPCORR
      COMMON /TEMPO/  RT,B1,B2,B1T,B2T,B1TT,B2TT,TV,GI,SI,CVI           GPCORR
```

```
      COMMON /BASIF/  GB,SB,ZB,PB1T0,PB1R0,CVB                          GPCORR
      COMMON /RESEF/  GR,SR,ZR,PR1T0,PR1R0,CVR                          GPCORR
      COMMON /GPCOF/  ZL,ZV,GL,GV                                       GPCORR
      PARAMETER(TCN=646.0D0,FINIS=1.D-7)                                GPCORR
      DO 55 K=1,10                                                      GPCORR
C     SATURATED LIQUID STATE                                            GPCORR
      CALL BASE(T,DLK,IECCH)                                            GPCORR
      CALL RESID(T,DLK)                                                 GPCORR
      ZL = ZB + ZR                                                      GPCORR
      GL = GB + GR                                                      GPCORR
      PR0L = PB1R0+PR1R0                                                GPCORR
C     SATURATED VAPOR STATE                                             GPCORR
      CALL BASE(T,DVK,IECCH)                                            GPCORR
      CALL RESID(T,DVK)                                                 GPCORR
      ZV = ZB + ZR                                                      GPCORR
      GV = GB + GR                                                      GPCORR
      PR0V = PB1R0+PR1R0                                                GPCORR
      Q=RT*(ZV*DVK+ZL*DLK)*5.D-1                                        GPCORR
      IF(T .GE. TCN)        RETURN                                      GPCORR
C     DEVIATIONS AND CORRECTIONS                                        GPCORR
      DD=DLK-DVK                                                        GPCORR
      DELG=GL-GV                                                        GPCORR
      CL=DVK*(DELG+ZV)-DLK*ZL                                           GPCORR
      CV=DLK*(DELG-ZL)+DVK*ZV                                           GPCORR
      CXL=CL*DLK/(DD*PR0L)                                              GPCORR
      CXV=CV*DVK/(DD*PR0V)                                              GPCORR
      IF(ABS(CXL) .LT. FINIS)                   RETURN                  GPCORR
      DLK=DLK+CXL                                                       GPCORR
   55 DVK=DVK+CXV                                                       GPCORR
      RETURN                                                            GPCORR
      END                                                               GPCORR
```

G.12 INR105.AQUA.FORT(SUBCRI)

```
      SUBROUTINE SUBCRI(T)                                              SUBCRI
C     SUBCRI CALCULATES DENSITIES, PRESSURES AND DERIVATIVES            SUBCRI
C     OF THE PRESSURES IN THE SATURATED STATES                          SUBCRI
C       DL     DV                           G/CM**3                     SUBCRI
C       PS     TPS1T    TTPS2T              PASCAL                      SUBCRI
C     AS WELL AS THE DENSITIES AND PRESSURES OF THE SPINODALS           SUBCRI
C       RHSL  RHSV                          KG/M**3                     SUBCRI
C       PSL    PSV                          PASCAL                      SUBCRI
C     AS FUNCTIONS OF THE TEMPERATURE T.                  94/12/01      SUBCRI
      IMPLICIT REAL*8(A-H,O-Z)                                          SUBCRI
      COMMON /SATIF/  PS,TPS1T,TTPS2T,DL,DV,RL1T0,RV1T0,RHSL,PSL        SUBCRI
     =,RHSV,PSV,SL,GL,PL1R0,PL1T0,CVL,ETL,XLL,PV1R0,PV1T0,CVV,ETV,XLV   SUBCRI
C-----APPROXIMATION TO THE VAPOR PRESSURE ----------------------------- SUBCRI
      CALL SSP(T)                                                       SUBCRI
C-----APPROXIMATIONS TO THE SATURATED DENSITIES------------------------ SUBCRI
      CALL SSRHO(T)                                                     SUBCRI
C-----APPROXIMATIONS TO THE------------------------------------------- SUBCRI
C-----DENSITIES AND PRESSURES ON THE LIQUID AND VAPOR SPINODAL--------- SUBCRI
      CALL SPIRHO(T)                                                    SUBCRI
```

```
      CALL SPIP(T)                                                      SUBCRI
C---------------------------------------------------------------------- SUBCRI
      RETURN                                                            SUBCRI
      END                                                               SUBCRI
```

G.13 INR105.AQUA.FORT(TWOPHA)

```
      SUBROUTINE TWOPHA(T)                                              TWOPHA
C     THIS SUBROUTINE CALCULATES DIFFERENT SATURATION PROPERTIES        TWOPHA
C     IN S.I. UNITS                                           94/11/30  TWOPHA
      IMPLICIT REAL*8(A-H,O-Z)                                          TWOPHA
      COMMON /SATIF/  PS,TPS1T,TTPS2T,DL,DV,RL1T0,RV1T0,RHSL,PSL        TWOPHA
     =,RHSV,PSV,SL,GL,PL1R0,PL1T0,CVL,ETL,XLL,PV1R0,PV1T0,CVV,ETV,XLV   TWOPHA
      PARAMETER(TC=647.126D0)                                           TWOPHA
C---------------------------------------------------------------------  TWOPHA
      IF(T .LE. TC)          GO TO 10                                   TWOPHA
      WRITE(6,1000) T                                                   TWOPHA
      T=-1.D0                                                           TWOPHA
      RETURN                                                            TWOPHA
C-----HEAT CAPACITY OF THE SATURATED STATES  CVL CVV ------------------ TWOPHA
   10 CALL SSCEV(T,CVL,CVV)                                             TWOPHA
C-----DENSITY DERIVATIVE OF THE SATURATED STATES  PLR10 PVR10 --------- TWOPHA
      CALL SSPER(T)                                                     TWOPHA
C-----TEMPERATURE DERIVATIVE OF THE SATURATED STATES  PL1T0 PV1T0 ------TWOPHA
      CALL SSPET(T,PL1T0,PV1T0)                                         TWOPHA
C-----ENTROPY AND GIBBS FUNCION OF THE SATURATED LIQUID  SL  GL ------- TWOPHA
      CALL SSGIS(T,SL,GL)                                               TWOPHA
C-----VISCOSITY OF THE SATURATED STATES  ETL ETV --------------------- TWOPHA
      CALL SSETA(T,ETL,ETV)                                             TWOPHA
C-----THERMAL CONDUCTIVITY OF THE SATURATED STATES  XLL XLV ----------- TWOPHA
      CALL SSLAM(T,XLL,XLV)                                             TWOPHA
C---------------------------------------------------------------------- TWOPHA
 1000 FORMAT(16H TEMPERATURE T =,F10.4,22H K IS SUPERCRITICAL ! /)      TWOPHA
      RETURN                                                            TWOPHA
      END                                                               TWOPHA
```

G.14 INR105.AQUA.FORT(REDUK)

```
      SUBROUTINE REDUK(T,D,IECCH)                                       REDUK
C     CALCULATES THERMAL PROPERTIES OF THE WATER                        REDUK
C      IN REDUCED ( DIMENSIONLESS ) UNITS                               REDUK
C     IN THE LIQUID, VAPOR AND GAS STATES                               REDUK
C     D : DENSITY IN G/CM**3 , RH : IN KG/M**3                94/12/08  REDUK
C     R1T0 : REDUCED DENSITY DERIVATIVE = (T/RH)*(DRH/DT)               REDUK
C     P1T0, P1R0 : REDUCED DERIVATIVES OF THE PRESSURE                  REDUK
C     H0  , S0   : REDUCED ENTHALPY RESP. ENTROPY ,    CPC : CP/CV      REDUK
C     ETA , XLMB : VISCOSITY RESP. THERMAL CONDUCTIVITY                 REDUK
      IMPLICIT REAL*8(A-H,O-Z)                                          REDUK
      COMMON /AKONST/ WM,TC,DC,GASCON,G1,G2,GF,TZ,UREF,SREF,TLOW        REDUK
      COMMON /TEMPO/  RT,B1,B2,B1T,B2T,B1TT,B2TT,TV,GI,SI,CVI           REDUK
      COMMON /BASIF/  GB,SB,ZB,PB1T0,PB1R0,CVB                          REDUK
      COMMON /RESEF/  GR,SR,ZR,PR1T0,PR1R0,CVR                          REDUK
```

```
      COMMON /REDUF/  G0,S0,Z,P1T0,P1R0,CV0,CP0,ETA,XLMB,R1T0,CPC,H0    REDUK
      PARAMETER(RGAS=4.61522D+2)                                        REDUK
      DATA FNI/1.D+70/                                                  REDUK
C---------------------------------------------------------------------- REDUK
      CALL BASE(T,D,IECCH)                                              REDUK
      IF(IECCH .NE. 0)       RETURN                                     REDUK
      CALL RESID(T,D)                                                   REDUK
  200 Z = ZB + ZR                                                       REDUK
      P1T0 = PR1T0 + PB1T0                                              REDUK
      P1R0 = PR1R0 + PB1R0                                              REDUK
      S0 = SB + SR + SI - SREF                                          REDUK
      G0 = GB + GR + GI - 1.D0 - UREF/T + SREF                          REDUK
      CV0 = CVB + CVR + CVI                                             REDUK
      IF(P1R0 .EQ. 0.D0)     THEN                                       REDUK
          R1T0 = -FNI                                                   REDUK
          CP0 = FNI                                                     REDUK
          CPC = FNI                                                     REDUK
      ELSE                                                              REDUK
          R1T0 = -P1T0/P1R0                                             REDUK
          CP0 = CV0-P1T0*R1T0                                           REDUK
          CPC = CP0/CV0                                                 REDUK
      ENDIF                                                             REDUK
  900 H0 = S0 + G0                                                      REDUK
      RH=D*1.D3                                                         REDUK
      CALL TRANSP(T,RH,P1R0,P1T0,ETA,XLMB)                              REDUK
C---------------------------------------------------------------------- REDUK
      RETURN                                                            REDUK
      END                                                               REDUK
```

G.15 INR105.AQUA.FORT(REDUKE)

```
      SUBROUTINE REDUKE(T,D,IECCH)                                      REDUKE
C     CALCULATES THERMAL PROPERTIES OF THE WATER                        REDUKE
C      IN REDUCED ( DIMENSIONLESS ) UNITS                               REDUKE
C     MIXED STATE INCLUDED                                              REDUKE
C     D : DENSITY IN G/CM**3 , RH : IN KG/M**3                 94/12/08 REDUKE
C     R1T0 : REDUCED DENSITY DERIVATIVE = (T/RHO)*(DRO/DT)              REDUKE
C     P1T0, P1R0 : REDUCED DERIVATIVES OF THE PRESSURE                  REDUKE
C     H0  ,  S0  : REDUCED ENTHALPY RESP. ENTROPY,   CPC : CP/CV        REDUKE
C     ETA , XLMB : VISCOSITY RESP. THERMAL CONDUCTIVITY                 REDUKE
C     VQ   : VAPOR QUALITY                                              REDUKE
      IMPLICIT REAL*8(A-H,O-Z)                                          REDUKE
      COMMON /AKONST/ WM,TC,DC,GASCON,G1,G2,GF,TZ,UREF,SREF,TLOW        REDUKE
      COMMON /TEMPO/  RT,B1,B2,B1T,B2T,B1TT,B2TT,TV,GI,SI,CVI           REDUKE
      COMMON /BASIF/  GB,SB,ZB,PB1T0,PB1R0,CVB                          REDUKE
      COMMON /RESEF/  GR,SR,ZR,PR1T0,PR1R0,CVR                          REDUKE
      COMMON /REDUF/  G0,S0,Z,P1T0,P1R0,CV0,CP0,ETA,XLMB,R1T0,CPC,H0,VQ REDUKE
      COMMON /SATIF/  PS,TPS1T,TTPS2T,DL,DV,RL1T0,RV1T0,RHSL,PSL        REDUKE
     =,RHSV,PSV,SL,GL,PL1R0,PL1T0,CVL,ETL,XLL,PV1R0,PV1T0,CVV,ETV,XLV   REDUKE
      PARAMETER(RGAS=4.61522D+2)                                        REDUKE
      DATA ZERO/0.D0/,FNI/1.D+70/,UN/1.D0/                              REDUKE
C---------------------------------------------------------------------- REDUKE
      CALL BASE(T,D,IECCH)                                              REDUKE
```

```
      IF(IECCH .NE. 0)       RETURN                                     REDUKE
      CALL RESID(T,D)                                                   REDUKE
      RH=D*1.D3                                                         REDUKE
      IF(T .GE. TC)          GO TO 200                                  REDUKE
      RDY = (D-DV)/(DL-DV)                                              REDUKE
      IF(D .GT. DL)     THEN                                            REDUKE
          VQ = ZERO                                                     REDUKE
      ELSEIF(D .LT. DV) THEN                                            REDUKE
          VQ = UN                                                       REDUKE
      ELSE                                                              REDUKE
          VQ = (UN-RDY)*DV/D                                            REDUKE
      ENDIF                                                             REDUKE
      IF(RH .LT. RHSV .OR. RH .GT. RHSL)        GO TO 200               REDUKE
C-----TWO PHASED MIXTURE------------------------------------------------ REDUKE
  100 CALL TWOPHA(T)                                                    REDUKE
      RGAT=RGAS*T                                                       REDUKE
      ROT=RGAT*D*1.D3                                                   REDUKE
      ROTL=RGAT*DL*1.D+3                                                REDUKE
      DIFF= UN - D/DL                                                   REDUKE
      X = RDY                                                           REDUKE
      IF(X .GT. ZERO .AND. X .LT. UN)                                   REDUKE
     =   X = DEXP(DLOG(RDY)/3.D0)                                       REDUKE
      P1R0 = ZERO                                                       REDUKE
      Z = PS/ROT                                                        REDUKE
      P1T0 = TPS1T/ROT                                                  REDUKE
      S0 = SL+DIFF*P1T0                                                 REDUKE
      G0 = GL                                                           REDUKE
      DCVS = RL1T0*RL1T0*PL1R0                                          REDUKE
      AU = TTPS2T/ROT                                                   REDUKE
      CV0 = CVL + DCVS + DIFF*AU                                        REDUKE
      R1T0 = -FNI                                                       REDUKE
      CP0 = FNI                                                         REDUKE
      CPC = FNI                                                         REDUKE
      ETAR=(UN-X)/ETV + X/(ETV + X*X*(ETL-ETV))                         REDUKE
      XLAR=(UN-X)/XLV + X/(XLV + X*X*(XLL-XLV))                         REDUKE
      ETA = UN/ETAR                                                     REDUKE
      XLMB = UN/XLAR                                                    REDUKE
          GO TO 900                                                     REDUKE
C-----LIQID, VAPOR OR SUPERCRITICAL REGIONS----------------------------- REDUKE
  200 P1R0 = PR1R0 + PB1R0                                              REDUKE
      Z = ZB + ZR                                                       REDUKE
      P1T0 = PR1T0 + PB1T0                                              REDUKE
      S0 = SB + SR + SI - SREF                                          REDUKE
      G0 = GB + GR + GI - UN - UREF/T + SREF                            REDUKE
      CV0 = CVB + CVR + CVI                                             REDUKE
      IF(P1R0 .EQ. 0.D0)     THEN                                       REDUKE
          R1T0 = -FNI                                                   REDUKE
          CP0 = FNI                                                     REDUKE
          CPC = FNI                                                     REDUKE
      ELSE                                                              REDUKE
          R1T0 = -P1T0/P1R0                                             REDUKE
          CP0 = CV0-P1T0*R1T0                                           REDUKE
          CPC = CP0/CV0                                                 REDUKE
      ENDIF                                                             REDUKE
```

```
      CALL TRANSP(T,RH,P1R0,P1T0,ETA,XLMB)                              REDUKE
C                                                                       REDUKE
  900 H0 = S0 + G0                                                      REDUKE
C---------------------------------------------------------------------- REDUKE
      RETURN                                                            REDUKE
      END                                                               REDUKE
```

G.16 INR105.AQUA.FORT(HYDOR)

```
      SUBROUTINE HYDOR(T,RH,IECCH)                                      HYDOR
C     CALCULATES THERMAL PROPERTIES OF WASSER                           HYDOR
C       S  G  H  H1P  U1R  CV  CP  P  P1T  P1R  R1T  R1P                HYDOR
C       ETA  XLMB  SRFTS   P1RS  VS   PRN  VQ                           HYDOR
C     AS FUNCTIONS OF THE TEMPERATURE, T AND THE DENSITY, RH            HYDOR
C     ALL PROPERTIES ARE IN S.I. UNITS                        94/12/08  HYDOR
      IMPLICIT REAL*8(A-H,O-Z)                                          HYDOR
      COMMON /REDUF/  G0,S0,Z,P1T0,P1R0,CV0,CP0,ET,XLB,R1T0,CPC,H0,VQ1  HYDOR
      COMMON /HYDUF/  S,G,H,H1P,U1R,CV,CP,P,P1T,P1R,R1T,R1P             HYDOR
     =,ETA,XLMB,SRFTS,P1RS,VS,PRN,V,VQ                                  HYDOR
      PARAMETER(TC=647.126D0,RGAS=4.61522D+2)                           HYDOR
      DATA TA/0.D0/,ZERO/0.D0/,UN/1.D0/                                 HYDOR
      IF(RH .LE. ZERO)       THEN                                       HYDOR
          WRITE(6,9100) RH                                              HYDOR
          IECCH = 5                                                     HYDOR
          RETURN                                                        HYDOR
      ENDIF                                                             HYDOR
  100 D=RH*1D-3                                                         HYDOR
      V=UN/RH                                                           HYDOR
C     VAPOR, LIQUID OR GAS STATE                                        HYDOR
  200 IF(T .EQ. TA)          GO TO 210                                  HYDOR
      ROT=T*RGAS                                                        HYDOR
      CALL ONLYT(T,IECCH)                                               HYDOR
      IF(IECCH .NE. 0)       RETURN                                     HYDOR
      SRFTS=ZERO                                                        HYDOR
      IF(T .GE. TC)          GO TO 210                                  HYDOR
      SRFTS=SIGMA(T)                                                    HYDOR
      CALL SUBCRI(T)                                                    HYDOR
  210 CALL REDUKE(T,D,IECCH)                                            HYDOR
      IF(IECCH .NE. 0)       RETURN                                     HYDOR
      VQ=VQ1                                                            HYDOR
      G=G0*ROT                                                          HYDOR
      H=H0*ROT                                                          HYDOR
      S=S0*RGAS                                                         HYDOR
      CP=CP0*RGAS                                                       HYDOR
      CV=CV0*RGAS                                                       HYDOR
      P1R=P1R0*ROT                                                      HYDOR
      P1T=P1T0*RGAS*RH                                                  HYDOR
      IF(P1R0 .GT. ZERO)     THEN                                       HYDOR
          P1RS=CPC*P1R                                                  HYDOR
          R1P=UN/P1R                                                    HYDOR
      ELSE                                                              HYDOR
          P1RS=(P1T0*P1T0/CV0)*ROT                                      HYDOR
          R1P=-R1T0/ROT                                                 HYDOR
```

```
      ENDIF                                                                     HYDOR
      U1R=(Z-P1T0)*V*ROT                                                        HYDOR
      H1P=(UN+R1T0)*V                                                           HYDOR
      R1T=(RH/T)*R1T0                                                           HYDOR
      VS=SQRT(P1RS)                                                             HYDOR
      P=Z*ROT*RH                                                                HYDOR
      ETA=ET                                                                    HYDOR
      XLMB=XLB                                                                  HYDOR
      PRN=CP*ETA/XLMB                                                           HYDOR
  999 TA=T                                                                      HYDOR
      RETURN                                                                    HYDOR
 9100 FORMAT(13H DENSITY RH =,G15.6,20H KG/M**3 TOO LOW !  /)                   HYDOR
      END                                                                       HYDOR
```

Appendix H. User-Interfaces

The following pages list the programs, which are needed directly by the users to calculate the water properties.

H.1 INR105.AQUA.FORT(WATER)

```
      PROGRAM WATER                                                     WATER
C     CALCULATES THERMAL PROPERTIES OF WATER AS FUNCTIONS               WATER
C     OF TWO VARIABLES : (T,RHO) OR (T,P)                     94/12/09  WATER
      IMPLICIT REAL*8(A-H,O-Z)                                          WATER
      COMMON /SATIF/  PS,TPS1T,TTPS2T,DL,DV,RL1T0,RV1T0,RHSL,PSL        WATER
     =,RHSV,PSV,SL,GL,PL1R0,PL1T0,CVL,ETL,XLL,PV1R0,PV1T0,CVV,ETV,XLV   WATER
      COMMON /HYDUF/  S,G,H,H1P,U1R,CV,CP,P,P1T,P1R,R1T,R1P             WATER
     =,ETA,XLMB,SRFTS,P1RS,VS,PRN,V,XI                                  WATER
      PARAMETER(TC=647.126D0,FR=1.D+3,FK=1.D-3,FM=1.D-6)                WATER
      DATA ZERO/0.D0/,IMO/0/,QIS,T,RHI/0.101325D0,373.15D0,958.4467D0/  WATER
      WRITE(6,1001)                                                     WATER
      WRITE(6,1002)                                                     WATER
      WRITE(6,1003)                                                     WATER
  100 WRITE(6,1050) T                                                   WATER
      READ(5,*) T                                                       WATER
      IF(T .LE. ZERO)        GO TO 999                                  WATER
  200 WRITE(6,1100)                                                     WATER
      READ(5,*) IMO                                                     WATER
      IF(IMO .LE. 0)         GO TO 100                                  WATER
  400 CALL PG3277                                                       WATER
      IF(IMO .EQ. 1)         GO TO 600                                  WATER
  500 WRITE(6,1200) QIS                                                 WATER
      READ(5,*) QIS                                                     WATER
      IF(QIS .EQ. ZERO)      GO TO 200                                  WATER
      DX=ZERO                                                           WATER
      IECCH=0                                                           WATER
      CALL ONLYT(T,IECCH)                                               WATER
      IF(IECCH .NE. 0)       GO TO 500                                  WATER
      IF(T .LT. TC)          THEN                                       WATER
          CALL SUBCRI(T)                                                WATER
          WRITE(6,1300)                                                 WATER
          READ(5,*)          ISTA                                       WATER
          IF(ISTA .EQ. 1)    THEN                                       WATER
              CALL DLFIND(T,QIS,DX,IECCH)                               WATER
              IF(IECCH .EQ. 0)         GO TO 700                        WATER
              WRITE(6,9300)                                             WATER
          ELSEIF(ISTA .EQ. 2)          THEN                             WATER
              CALL DVFIND(T,QIS,DX,IECCH)                               WATER
              IF(IECCH .EQ. 0 .AND. DX .GT. ZERO)          GO TO 700    WATER
              WRITE(6,9400)                                             WATER
          ENDIF                                                         WATER
      ELSE                                                              WATER
          CALL DGFIND(T,QIS,DX,IECCH)                                   WATER
          IF(IECCH .EQ. 0 .AND. DX .GT. ZERO)              GO TO 700    WATER
          WRITE(6,9500)                                                 WATER
```

```
      ENDIF                                                             WATER
          GO TO 500                                                     WATER
  600 WRITE(6,1400) RHI                                                 WATER
      READ(5,*) RHI                                                     WATER
      IF(RHI .LE. ZERO)      GO TO 200                                  WATER
      IF(RHI .GT. ZERO)      GO TO 800                                  WATER
  700 RHI=DX*FR                                                         WATER
  800 IECCH=0                                                           WATER
      CALL HYDOR(T,RHI,IECCH)                                           WATER
      IF(IECCH .NE. 0)       GO TO 300                                  WATER
      A=G-P/RHI                                                         WATER
      U=H-P/RHI                                                         WATER
      WRITE(6,1610) T,RHI,P*FM                                          WATER
      IF(P1R .NE. ZERO) THEN                                            WATER
          WRITE(6,1620) P1R*FK,P1T*FM,R1T                               WATER
          WRITE(6,1630) CV*FK,CP*FK,S*FK                                WATER
          WRITE(6,1640) U*FK,H*FK,G*FK                                  WATER
          WRITE(6,1650) U1R*FK,H1P,VS                                   WATER
      ELSE                                                              WATER
          WRITE(6,1622) P1R*FK,P1T*FM                                   WATER
          WRITE(6,1632) CV*FK,S*FK                                      WATER
          WRITE(6,1640) U*FK,H*FK,G*FK                                  WATER
          WRITE(6,1652) U1R*FK,VS                                       WATER
      ENDIF                                                             WATER
      IF(T .LE. TC)          THEN                                       WATER
          IF(XI .LT. 1.D0)   WRITE(6,1660) ETA,XLMB,SRFTS               WATER
          IF(XI .GE. 1.D0)   WRITE(6,1662) ETA,XLMB                     WATER
          IF(P1R .NE. ZERO)  WRITE(6,1670) PRN,XI,PS*FM                 WATER
          IF(P1R .EQ. ZERO)  WRITE(6,1674) XI,PS*FM                     WATER
      ELSE                                                              WATER
          WRITE(6,1662) ETA,XLMB                                        WATER
          WRITE(6,1672) PRN                                             WATER
      ENDIF                                                             WATER
      IF(T .LE. TC)          THEN                                       WATER
          WRITE(6,1680) DV*FR,RHSV,PSV*FM                               WATER
          WRITE(6,1682) DL*FR,RHSL,PSL*FM                               WATER
      ENDIF                                                             WATER
  300 WRITE(6,2000)                                                     WATER
          GO TO 400                                                     WATER
 1001 FORMAT(/2H +,39(2H++)/2H +,77X,1H+/2H +,5X,1HW,5X,1HW,5X,1HA,5X   WATER
     =,7(1HT),2X,7(1HE),2X,6(1HR),9X,17HSTATUS : 94/12/09,4X,1H+/2H +   WATER
     =,5X,1HW,5X,1HW,4X,3(1HA),4X,7(1HT),2X,7(1HE),2X,7(1HR),29X,1H+    WATER
     =/2H +,5X,1HW,5X,1HW,3X,5HAA AA,6X,1HT,5X,1HE,6X,2X,1HR,5X,1HR,29X WATER
     =,1H+/2H +,5X,7HW  W  W,2X,7HAA   AA,5X,1HT,5X,5HE EEE,4X,1HR,1X   WATER
     =,5(1HR),29X,1H+/2H +,5X,7HW WWW W,2X,1HA,5X,1HA,5X,1HT,5X,5HE EEE WATER
     =,4X,1HR,1X,4(1HR),30X,1H+/2H +,5X,7HWWW WWW,2X,7HA AAA A,5X,1HT   WATER
     =,5X,1HE,8X,5HR  RR,31X,1H+/2H +,5X,7HWW   WW,2X,7HA AAA A,5X,1HT  WATER
     =,5X,7(1HE),2X,6HR   RR,30X,1H+/2H +,5X,1HW,5X,1HW,2X,1HA,5X       WATER
     =,1HA,5X,1HT,5X,7(1HE),2X,1HR,4X,2HRR,29X,1H+ )                    WATER
 1002 FORMAT(2H +,77X,1H+/2H +,14X                                      WATER
     =,48HWATER CALCULATES THERMAL PROPERTIES OF THE WATER,15X,1H+      WATER
     =/2H +,19X,39HIN THE LIQUID, VAPOR AND GASEOUS STATES,19X,1H+      WATER
     =/2H +,15X,46HEITHER AS FUNCTIONS OF TEMPERATURE AND DENSITY,16X   WATER
     =,1H+/2H +,16X,44HOR AS FUNCTIONS OF TEMPERATURE AND PRESSURE.     WATER
     =,17X,1H+)                                                         WATER
```

```
1003 FORMAT(2H +,77X,1H+/2H +,14X                                       WATER
    =,49HTHE PROPERTIES CALCULATED ARE THE FOLLOWING ONES:              WATER
    =,14X,1H+/2H +,77X,1H+/2H +,5X,3HT =,12H TEMPERATURE,11X,3HP =      WATER
    =,9H PRESSURE,15X,5HRHO =,8H DENSITY,6X,1H+/2H +,5X                 WATER
    =,46HCV / CP = ISOBARIC / ISOCHORIC HEAT CAPACITIES,7X              WATER
    =,11HS = ENTROPY,8X,1H+/2H +,5X,18HG = GIBBS FUNCTION,8X            WATER
    =,10HU = ENERGY,17X,12HH = ENTHALPY,7X,1H+/2H +,5X                  WATER
    =,22HVEL.S = SONIC VELOCITY,4X,23HSIGMA = SURFACE TENSION,4X        WATER
    =,15HETA = VISCOSITY,4X,1H+/2H +,5X                                 WATER
    =,29HLAMBDA = THERMAL CONDUCTIVITY,12X,22HPR.NR = PRANDTL NUMBER    WATER
    =,9X,1H+/2H +,5X,20HVAP.QU = M.VAP/M.TOT,52X,1H+                    WATER
    =/2H +,77X,1H+/2H +,28X,21HFOR MORE INFORMATION ,5H =>              WATER
    =,19HK. THURNAY, T: 2477,4X,1H+/2H +,39(2H++) ///)                  WATER
1050 FORMAT(//29H ENTER TEMPERATURE ( IN K ) /,F10.3                    WATER
    =,31H / =:       ( STOP WTIH T = 0 )  )                             WATER
1100 FORMAT(/30H DO YOU WANT THE PROPERTIES AS                          WATER
    =/ 5X,35H 1 = TEMPERATURE-DENSITY FUNCTIONS                         WATER
    =/ 5X,35H 2 = TEMPERATURE-PRESSURE FUNCTIONS                        WATER
    =,15X,15H( STOP WITH 0 )  )                                         WATER
1200 FORMAT(//33H ENTER PRESSURE ( IN  MPASCAL ) /,G15.7                WATER
    =,28H / =:    ( STOP WTIH P = 0 )  )                                WATER
1300 FORMAT(/36H LIQUID ( 1 ) OR VAPOR ( 2 ) STATE ?                    WATER
    =,17H ( 0 = STOP ) =: )                                             WATER
1400 FORMAT(//31H ENTER DENSITY ( IN KG/M**3 ) /,G15.7                  WATER
    =,28H / =:  ( STOP WTIH RHO = 0 )  )                                WATER
1610 FORMAT(//4H T =,F10.3,13X,5HRHO =,G14.7,8X,3HP =,G15.7            WATER
    =/2X,2H K,25X,7HKG/M**3,20X,7HMPASCAL)                              WATER
1620 FORMAT(10H DP/DRHO =,G13.6,4X,7HDP/DT =,G13.6,7X,9HDRHO/DT =       WATER
    =,G14.6/2X,6H KJ/KG,21X,9HMPASCAL/K,18X,10HKG/(K*M**3 )             WATER
1622 FORMAT(10H DP/DRHO =,G13.6,4X,7HDP/DT =,G13.6,7X,9HDRHO/DT =       WATER
    =,6H -INF./2X,6H KJ/KG,21X,9HMPASCAL/K,17X,9HKG/M**3/K )            WATER
1630 FORMAT(5H CV =,G13.6,9X,4HCP =,G13.6,10X,3HS =,G14.6              WATER
    =/3X,9HKJ/(KG*K),17X,9HKJ/(KG*K),18X,9HKJ/(KG*K) )                 WATER
1632 FORMAT(5H CV =,G13.6,9X,4HCP =,5H INF.,18X,3HS =,G14.6             WATER
    =/3X,9HKJ/(KG*K),17X,9HKJ/(KG*K),18X,9HKJ/(KG*K) )                 WATER
1640 FORMAT(4H U =,G14.6,9X,3HH =,G14.6,10X,3HG =,G14.6                WATER
    =/2X,6H KJ/KG,21X,5HKJ/KG,22X,5HKJ/KG )                             WATER
1650 FORMAT(10H DU/DRHO =,G14.6,3X,7HDH/DP =,G14.6,6X                   WATER
    =,7HVEL.S =,G13.6/2X,14H KJ*M**3/KG**2,13X,7HM**3/KG,20X,5HM/SEC)   WATER
1652 FORMAT(10H DU/DRHO =,G14.6,3X,7HDH/DP =,6H -INF.,14X               WATER
    =,7HVEL.S =,G13.6/2X,14H KJ*M**3/KG**2,13X,7HM**3/KG,20X,5HM/SEC)   WATER
1660 FORMAT(6H ETA =,G13.6,8X,8HLAMBDA =,G13.6,6X                       WATER
    =,7HSIGMA =,G13.6/2X,9H KG/M/SEC,18X,5HW/M/K,22X,9HKG/SEC**2)       WATER
1662 FORMAT(6H ETA =,G13.6,8X,8HLAMBDA =,G13.6                          WATER
    =/2X,9H KG/M/SEC,18X,5HW/M/K)                                       WATER
1670 FORMAT(8H PR.NR =,G13.6,6X,8HVAP.QU =,G13.6,6X,4HPS =,G14.7        WATER
    =/56X,7HMPASCAL )                                                   WATER
1672 FORMAT(8H PR.NR =,G13.6/)                                          WATER
1674 FORMAT(8H PR.NR =,6H -INF.,13X,8HVAP.QU =,G13.6,6X,4HPS =,G14.7    WATER
    =/56X,7HMPASCAL )                                                   WATER
1680 FORMAT(7H RH.V =,G14.7,6X,7HRH.SV =,G14.7,6X,6HP.SV =              WATER
    =,G13.6/2X,8H KG/M**3,19X,7HKG/M**3,20X,7HMPASCAL )                 WATER
1682 FORMAT(7H RH.L =,G14.7,6X,7HRH.SL =,G14.7,6X,6HP.SL =              WATER
    =,G13.6/2X,8H KG/M**3,19X,7HKG/M**3,20X,7HMPASCAL )                 WATER
2000 FORMAT(//////////////////////////////)                             WATER
```

```
 9300 FORMAT(/35H NO LIQUID STATE AT THIS PRESSURE ! )                  WATER
 9400 FORMAT(/35H NO VAPOR STATE AT THIS PRESSURE !  )                  WATER
 9500 FORMAT(/42H NO SUPERCRITICAL STATE AT THIS PRESSURE !  )          WATER
  999 STOP                                                              WATER
      END                                                               WATER
```

H.2 INR105.AQUA.FORT(KAISOT)

```
      PROGRAM KAISOT                                                    KAISOT
C     CALCULATES THERMAL PROPERTIES OF WATER                            KAISOT
C          ALONG DIFFERENT ISOTHERMS                          94/11/30  KAISOT
      IMPLICIT REAL*8(A-H,O-Z)                                          KAISOT
      REAL*8 FRH(1000),RH(1000),F(1000),XF(1000)                        KAISOT
      INTEGER NEIG(50)                                                  KAISOT
      CHARACTER*8 NAMR,MASR,NAMF,MASF,NORMR,NORMW,MODUS                 KAISOT
     =,NAMG,MASG                                                        KAISOT
      COMMON /MUFUN/  RH,RMI,RMA,RED,FRH,FMI,FMA,FED                    KAISOT
      COMMON /HYDUF/  S,G,H,H1P,U1R,CV,CP,P,P1T,P1R,R1T,R1P             KAISOT
     =,ETA,XLMB,SRFTS,P1RS,VS,PRN,V,XI                                  KAISOT
      DATA IRE/15/,IWR/16/,MODUS/'**OLD***'/,KPRI/2/                    KAISOT
      DATA IA,IE,II/1,1000,1/                                           KAISOT
      DATA NEIG/11,15,17,21,22,23,24,26,27,31,32,34,35,36,41,42,43      KAISOT
     =,44,45,46,47,29*0/                                                KAISOT
      WRITE(6,1000)                                                     KAISOT
      CALL SERDIO(IRE,NORMR,IWR,NORMW,MODUS,KPRI)                       KAISOT
      IF(NORMR .NE. 'GRAPHIC8')                   GO TO 999             KAISOT
      IF(NORMW .NE. 'GRAPHIC8')                   GO TO 999             KAISOT
  120 WRITE(6,1100)                                                     KAISOT
      READ(5,*) T                                                       KAISOT
      IF(T .LE. 0)            GO TO 999                                 KAISOT
  200 CALL SIGNAT(KLASSE,NUMMER,NAMF,MASF,NEIG)                         KAISOT
      IF(KLASSE .LE. 0)       GO TO 120                                 KAISOT
      CALL MUSTER(IA,IE,II,NAMR,MASR,NAMG,MASG,*200)                    KAISOT
      CALL PG3277                                                       KAISOT
      NAMF(5:8)='.ITH'                                                  KAISOT
      WRITE(6,1200)                                                     KAISOT
      READ(5,*) ISCRI                                                   KAISOT
      MP=0                                                              KAISOT
      DO 111 I=IA,IE,II                                                 KAISOT
      RHI=RH(I)                                                         KAISOT
      CALL HYDOR(T,RHI,IECCH)                                           KAISOT
      IF(IECCH .NE. 0)        GO TO 111                                 KAISOT
      MP=MP+1                                                           KAISOT
      IF(ISCRI .EQ. 0 .AND.  MOD(MP,25) .EQ. 1)                         KAISOT
     =    WRITE(6,1500) NAMF,NAMG                                       KAISOT
      IF(KLASSE .EQ. 11)      F(I)=RH(I)                                KAISOT
      IF(KLASSE .EQ. 15)      F(I)=P                                    KAISOT
      IF(KLASSE .EQ. 17)      F(I)=V                                    KAISOT
      IF(KLASSE .EQ. 21)      F(I)=P1R                                  KAISOT
      IF(KLASSE .EQ. 22)      F(I)=P1T                                  KAISOT
      IF(KLASSE .EQ. 23)      F(I)=R1P                                  KAISOT
      IF(KLASSE .EQ. 24)      F(I)=R1T                                  KAISOT
      IF(KLASSE .EQ. 26)      F(I)=CV                                   KAISOT
```

```
      IF(KLASSE .EQ. 27)        F(I)=CP                                         KAISOT
      IF(KLASSE .EQ. 31)        F(I)=VS                                         KAISOT
      IF(KLASSE .EQ. 32)        F(I)=SRFTS                                      KAISOT
      IF(KLASSE .EQ. 34)        F(I)=ETA                                        KAISOT
      IF(KLASSE .EQ. 35)        F(I)=XLMB                                       KAISOT
      IF(KLASSE .EQ. 36)        F(I)=PRN                                        KAISOT
      IF(KLASSE .EQ. 41)        F(I)=G-P*V                                      KAISOT
      IF(KLASSE .EQ. 42)        F(I)=G                                          KAISOT
      IF(KLASSE .EQ. 43)        F(I)=H-P*V                                      KAISOT
      IF(KLASSE .EQ. 44)        F(I)=H                                          KAISOT
      IF(KLASSE .EQ. 45)        F(I)=S                                          KAISOT
      IF(KLASSE .EQ. 46)        F(I)=U1R                                        KAISOT
      IF(KLASSE .EQ. 47)        F(I)=H1P                                        KAISOT
      IF(ISCRI .EQ. 0)  WRITE(6,1600) I,RH(I),F(I),FRH(I)                       KAISOT
      IF(ISCRI .EQ. 0 .AND.  MOD(MP,25) .EQ. 0)  WRITE(6,1400)                  KAISOT
  111 CONTINUE                                                                  KAISOT
      IF(ISCRI .EQ. 0)          GO TO 199                                       KAISOT
      LAX=-1                                                                    KAISOT
      RMI=RMA                                                                   KAISOT
      CALL FENST8(MP,RH,XF,RMI,RMA,RED,LAX,KAXE,MASR)                           KAISOT
      FMI=FMA                                                                   KAISOT
      CALL FENST8(MP,F,XF,FMI,FMA,FED,LAX,KAXE,MASF)                            KAISOT
      NUMMER=T*10.D0                                                            KAISOT
      CALL SDING8(KLASSE,NUMMER,MP                                              KAISOT
     =,RH,RMI,RMA,RED,NAMR,MASR,F,FMI,FMA,FED,NAMF,MASF,MODUS)                  KAISOT
  199 WRITE(6,2000)                                                             KAISOT
          GO TO 200                                                             KAISOT
 1000 FORMAT(2H +,39(2H++)/2H +,77X,1H+                                         KAISOT
     =/2H +,6X,14HPROGRAM KAISOT,34X,17HSTATUS : 94/11/30,6X,1H+                KAISOT
     =/2H +,77X,1H+/2H +,6X,30HCALCULATES THERMAL PROPERTIES                    KAISOT
     =,30HOF THE WATER ALONG AN ISOTHERM,11X,1H+                                KAISOT
     =/2H +,24X,27HAS FUNCTIONS OF THE DENSITY,26X,1H+                          KAISOT
     =/2H +,12X,35HUSING A SAMPLE (RH,FRH)-FUNCTION AS                          KAISOT
     =,18H A DENSITY-PATTERN,12X,1H+/2H +,77X,1H+/2H +,39(2H++) )               KAISOT
 1100 FORMAT(49H ENTER TEMPERATURE OF THE ISOTHERM =: ( 0 = END )  )            KAISOT
 1200 FORMAT(37H FUNCTION TO BE SAVED ? ( 0 = NO ) =: )                         KAISOT
 1400 FORMAT(///)                                                               KAISOT
 1500 FORMAT(/1H0,3X,1HI,9X,2HRH,8X,5HCALC.,A8,6X,5HMAST.,A8)                   KAISOT
 1600 FORMAT(I5,2G16.6,G19.6)                                                   KAISOT
 2000 FORMAT(///////////////////////////////////)                               KAISOT
  999 STOP                                                                      KAISOT
      END                                                                       KAISOT
```

H.3 *INR105.AQUA.FORT(KAISOK)*

```
      PROGRAM KAISOK                                                            KAISOK
C     CALCULATES THERMAL PROPERTIES OF WATER                                    KAISOK
C           ALONG DIFFERENT ISOCHORES                               94/11/30   KAISOK
      IMPLICIT REAL*8(A-H,O-Z)                                                  KAISOK
      REAL*8 FX(1000),X(1000),F(1000),XF(1000)                                  KAISOK
      INTEGER NEIG(50)                                                          KAISOK
      CHARACTER*8 NAMX,MASX,NAMF,MASF,NORMR,NORMW,MODUS                         KAISOK
     =,NAMG                                                                     KAISOK
```

```
      COMMON /MUFUN/  X,XMI,XMA,XED,FX,FMI,FMA,FED                      KAISOK
      COMMON /HYDUF/  S,G,H,H1P,U1R,CV,CP,P,P1T,P1R,R1T,R1P             KAISOT
     =,ETA,XLMB,SRFTS,P1RS,VS,PRN,V,XI                                  KAISOT
      DATA IRE/15/,IWR/16/,MODUS/'**OLD***'/,KPRI/2/                    KAISOK
      DATA IA,IE,II/1,1000,1/                                           KAISOK
      DATA NEIG/11,15,17,21,22,23,24,26,27,31,32,34,35,36,41,42,43      KAISOT
     =,44,45,46,47,29*0/                                                KAISOT
      WRITE(6,1000)                                                     KAISOK
      CALL SERDIO(IRE,NORMR,IWR,NORMW,MODUS,KPRI)                       KAISOK
      IF(NORMR .NE. 'GRAPHIC8')                  GO TO 999              KAISOK
      IF(NORMW .NE. 'GRAPHIC8')                  GO TO 999              KAISOK
  120 WRITE(6,1100)                                                     KAISOK
      READ(5,*) RHI                                                     KAISOK
      IF(RHI .LE. 0)         GO TO 999                                  KAISOK
  200 CALL SIGNAT(KLASSE,NUMMY,NAMF,MASF,NEIG)                          KAISOK
      IF(KLASSE .LE. 0)      GO TO 120                                  KAISOK
      CALL MUSTER(IA,IE,II,NAMX,MASX,NAMG,MASF,*200)                    KAISOK
      CALL PG3277                                                       KAISOK
      NAMF(5:8)='.ICH'                                                  KAISOK
      WRITE(6,1200)                                                     KAISOK
      READ(5,*) ISCRI                                                   KAISOK
      MP=0                                                              KAISOK
      DO 111 I=IA,IE,II                                                 KAISOK
      T=X(I)                                                            KAISOK
      CALL HYDOR(T,RHI,IECCH)                                           KAISOK
      IF(IECCH .NE. 0)       GO TO 111                                  KAISOK
      MP=MP+1                                                           KAISOK
      IF(ISCRI .EQ. 0 .AND.  MOD(MP,25) .EQ. 1)                         KAISOK
     =    WRITE(6,1500) NAMF,NAMG                                       KAISOK
      IF(KLASSE .EQ. 15)     F(I)=P                                     KAISOK
      IF(KLASSE .EQ. 17)     F(I)=V                                     KAISOK
      IF(KLASSE .EQ. 21)     F(I)=P1R                                   KAISOK
      IF(KLASSE .EQ. 22)     F(I)=P1T                                   KAISOK
      IF(KLASSE .EQ. 23)     F(I)=R1P                                   KAISOK
      IF(KLASSE .EQ. 24)     F(I)=R1T                                   KAISOK
      IF(KLASSE .EQ. 26)     F(I)=CV                                    KAISOK
      IF(KLASSE .EQ. 27)     F(I)=CP                                    KAISOK
      IF(KLASSE .EQ. 31)     F(I)=VS                                    KAISOK
      IF(KLASSE .EQ. 32)     F(I)=SRFTS                                 KAISOK
      IF(KLASSE .EQ. 34)     F(I)=ETA                                   KAISOK
      IF(KLASSE .EQ. 35)     F(I)=XLMB                                  KAISOK
      IF(KLASSE .EQ. 36)     F(I)=PRN                                   KAISOK
      IF(KLASSE .EQ. 41)     F(I)=G-P*V                                 KAISOT
      IF(KLASSE .EQ. 42)     F(I)=G                                     KAISOK
      IF(KLASSE .EQ. 43)     F(I)=H-P*V                                 KAISOT
      IF(KLASSE .EQ. 44)     F(I)=H                                     KAISOK
      IF(KLASSE .EQ. 45)     F(I)=S                                     KAISOK
      IF(KLASSE .EQ. 46)     F(I)=U1R                                   KAISOT
      IF(KLASSE .EQ. 47)     F(I)=H1P                                   KAISOK
      IF(ISCRI .EQ. 0)  WRITE(6,1600) I,T,F(I),FX(I)                    KAISOK
      IF(ISCRI .EQ. 0 .AND.  MOD(MP,25) .EQ. 0)  WRITE(6,1400)          KAISOK
  111 CONTINUE                                                          KAISOK
      IF(ISCRI .EQ. 0)       GO TO 199                                  KAISOK
      LAX=-1                                                            KAISOK
      XMI=XMA                                                           KAISOK
```

```
     CALL FENST8(MP,X,XF,XMI,XMA,XED,LAX,KAXE,MASR)                       KAISOK
     FMI=FMA                                                              KAISOK
     CALL FENST8(MP,F,XF,FMI,FMA,FED,LAX,KAXE,MASF)                       KAISOK
     NUMMER=RHI                                                           KAISOK
     CALL SDING8(KLASSE,NUMMER,MP                                         KAISOK
    =,X,XMI,XMA,XED,NAMX,MASX,F,FMI,FMA,FED,NAMF,MASF,MODUS)              KAISOK
 199 WRITE(6,2000)                                                        KAISOK
         GO TO 200                                                        KAISOK
1000 FORMAT(2H +,39(2H++)/2H +,77X,1H+                                    KAISOK
    =/2H +,6X,14HPROGRAM KAISOK,34X,17HSTATUS : 94/11/30,6X,1H+           KAISOK
    =/2H +,77X,1H+/2H +,6X,30HCALCULATES THERMAL PROPERTIES               KAISOK
    =,30HOF THE WATER ALONG AN ISOCHORE,11X,1H+                           KAISOK
    =/2H +,23X,31HAS FUNCTIONS OF THE TEMPERATURE,23X,1H+                 KAISOK
    =/2H +,12X,33HUSING A SAMPLE (T,FT)-FUNCTION AS                       KAISOK
    =,22H A TEMPERATURE-PATTERN,10X,1H+/2H +,77X,1H+/2H +,39(2H++) )      KAISOK
1100 FORMAT(1H1/45H ENTER DENSITY OF THE ISOCHORE =: ( 0 = END )  )       KAISOK
1200 FORMAT(37H FUNCTION TO BE SAVED ? ( 0 = NO ) =: )                    KAISOK
1400 FORMAT(///)                                                          KAISOK
1500 FORMAT(/1H0,3X,1HI,5X,1HT,8X,5HCALC.,A8,6X,5HMAST.,A8)               KAISOK
1600 FORMAT(I5,F10.3,G16.6,G19.6)                                         KAISOK
2000 FORMAT(///////////////////////////////////)                          KAISOK
 999 STOP                                                                 KAISOK
     END                                                                  KAISOK
```

Note: The routine PG3277 in the above programs serves only to clear the screen for a new batch of output and can be omitted, or replaced with an alternative routine.

Appendix I. H.G.K.-users

The following programs serve to calculate different water properties using the H.G.K.-routines and the IAPS-82 equations (3.0, "Transport properties of the water." on page 9).

I.1 INR105.AQUA.FORT(HGKSPN)

```
      PROGRAM HGKSPN                                                    HGKSPN
C     CALCULATES THE SPINODALS OF THE WATER                             HGKSPN
C     USING THE HGK-CODE                                      94/12/13  HGKSPN
      IMPLICIT REAL*8(A-H,O-Z)                                          HGKSPN
      REAL*8 P(1000),F(1000),R(1000),W(1000)                            HGKSPN
      REAL*8 Y(3),FY(3)                                                 HGKSPN
      CHARACTER*8 NORMR,NORMW,MODUS                                     HGKSPN
     =,NAMW,MASW,NAMR,MASR,NAMP,MASP                                    HGKSPN
      COMMON /MUFUN/ W,WMI,WMA,WED ,F,FMI,FMA,FED                       HGKSPN
      COMMON /REDUF/  G0,S0,Z,P1T0,P1R0,CV0,CP0,ETA,XLMB,R1T0,CPC,H0    HGKSPN
      COMMON /SATIF/  PS,TPS1T,TTPS2T,DL,DV,RL1T0,RV1T0,RHSL,PSL        HGKSPN
     =,RHSV,PSV,SL,GL,PL1R0,PL1T0,CVL,ETL,XLL,PV1R0,PV1T0,CVV,EV,XLV    HGKSPN
      PARAMETER(TC=647.126D0,RHC=0.322D3,RGAS=4.61522D+2,TCN=646.3D0)   HGKSPN
      DATA IRE/15/,IWR/16/,MODUS/'**OLD***'/,KPRI/2/                    HGKSPN
      DATA IA,IE,II/1,1000,1/,IP/0/                                     HGKSPN
      DATA ITE/10/,ZERO/0.D0/,PRM0/1.D-10/,RHMAX/1.2D3/                 HGKSPN
      DATA NAMW/'    T   '/                                             HGKSPN
      DATA MASW,MASR,MASP/'    K   ',' KG/M**3','  PASCAL'/             HGKSPN
C     ----------------------------------------------------------------  HGKSPN
      WRITE(6,1000)                                                     HGKSPN
      CALL SERDIO(IRE,NORMR,IWR,NORMW,MODUS,KPRI)                       HGKSPN
      IF(NORMR .NE. 'GRAPHIC8' .OR. NORMW .NE. 'GRAPHIC8') GO TO 999    HGKSPN
  100 WRITE(6,1100) IP                                                  HGKSPN
      READ(5,*) IP                                                      HGKSPN
      IPP=MIN(IPP,2)                                                    HGKSPN
      IF(IP .LE. 0)          GO TO 999                                  HGKSPN
      CALL MUSTER(IA,IE,II,NAMT,MAST,NAMP,MASP,*100)                    HGKSPN
      CALL PG3277                                                       HGKSPN
      WRITE(6,1200)                                                     HGKSPN
      READ(5,*) ISCRI                                                   HGKSPN
C     DENSITY ITERATION FOR THE "DP/DRH = 0" POINT                      HGKSPN
      MP=0                                                              HGKSPN
      DO 211 J=IA,IE,II                                                 HGKSPN
      T=W(J)                                                            HGKSPN
C     IF(T .GT. TCN)         GO TO 490                                  HGKSPN
      CALL ONLYT(T,IECCH)                                               HGKSPN
      CALL SUBCRI(T)                                                    HGKSPN
      RED=2.D-2*RHC                                                     HGKSPN
      IF(IP .EQ. 1)          THEN                                       HGKSPN
          NAMR='RHSL.HGK'                                               HGKSPN
          NAMP='PSPL.HGK'                                               HGKSPN
          RHL=DL*1.D3                                                   HGKSPN
          RHI=MIN(1.25D0*RHL,RHMAX)                                     HGKSPN
          RED=ABS(RED)                                                  HGKSPN
```

```
      ELSEIF(IP .EQ. 2)      THEN                                       HGKSPN
          NAMR='RHSV.HGK'                                               HGKSPN
          NAMP='PSPV.HGK'                                               HGKSPN
          RHV=DV*1.D3                                                   HGKSPN
          RHI=7.5D-1*RHV                                                HGKSPN
          RED=-ABS(RED)                                                 HGKSPN
      ENDIF                                                             HGKSPN
      MP=MP+1                                                           HGKSPN
      IF(ISCRI .EQ. 0 .AND. MOD(MP,25) .EQ. 1)   WRITE(6,1600)          HGKSPN
      RH=RHI                                                            HGKSPN
      IT=0                                                              HGKSPN
      DO 311 L=1,1000                                                   HGKSPN
      ROF=RH-RHC                                                        HGKSPN
      IF(IP .EQ. 1 .AND. ROF .LT. ZERO)       THEN                      HGKSPN
          WRITE(6,9000)  T                                              HGKSPN
          GO TO 490                                                     HGKSPN
      ELSEIF(IP .EQ. 2 .AND. ROF .GT. ZERO)     THEN                    HGKSPN
          WRITE(6,9001)  T                                              HGKSPN
          GO TO 490                                                     HGKSPN
      ENDIF                                                             HGKSPN
      DX=RH*1.D-3                                                       HGKSPN
      CALL REDUK(T,DX,IECCH)                                            HGKSPN
      IF(P1R0 .EQ. ZERO)     THEN                                       HGKSPN
          R(J)=RH                                                       HGKSPN
          GO TO 450                                                     HGKSPN
      ENDIF                                                             HGKSPN
      IF(L .LE. 2)           GO TO 320                                  HGKSPN
      IF(IP .EQ. 1)          THEN                                       HGKSPN
          Y(3)=Y(2)                                                     HGKSPN
          FY(3)=FY(2)                                                   HGKSPN
      ELSE                                                              HGKSPN
          Y(1)=Y(2)                                                     HGKSPN
          FY(1)=FY(2)                                                   HGKSPN
      ENDIF                                                             HGKSPN
  320 IF(L .LE. 1)           GO TO 330                                  HGKSPN
      IF(IP .EQ. 1)          THEN                                       HGKSPN
          Y(2)=Y(1)                                                     HGKSPN
          FY(2)=FY(1)                                                   HGKSPN
      ELSE                                                              HGKSPN
          Y(2)=Y(3)                                                     HGKSPN
          FY(2)=FY(3)                                                   HGKSPN
      ENDIF                                                             HGKSPN
  330 IF(IP .EQ. 1)          THEN                                       HGKSPN
          Y(1)=DX                                                       HGKSPN
          FY(1)=P1R0                                                    HGKSPN
      ELSE                                                              HGKSPN
          Y(3)=DX                                                       HGKSPN
          FY(3)=P1R0                                                    HGKSPN
      ENDIF                                                             HGKSPN
      IF(P1R0 .GT. ZERO)     GO TO 311                                  HGKSPN
      IF(P1R0 .EQ. ZERO)     THEN                                       HGKSPN
          R(J)=RH                                                       HGKSPN
          GO TO 450                                                     HGKSPN
      ENDIF                                                             HGKSPN
      IF(L .GT. 2)           GO TO 410                                  HGKSPN
```

```
  311 RH=RH-RED                                                         HGKSPN
  410 IT=IT+1                                                           HGKSPN
  420 CALL QUADO(FY,Y,ZERO,DX,ID)                                       HGKSPN
      IF(ID .NE. 0)          THEN                                       HGKSPN
          WRITE(6,1500)      ID,(Y(K),K=1,3),(FY(K),K=1,3)              HGKSPN
      ENDIF                                                             HGKSPN
      KI=0                                                              HGKSPN
      CALL SORBET(KI,FY,Y,ZERO,DX)                                      HGKSPN
      CALL REDUK(T,DX,IECCH)                                            HGKSPN
      R(J)=DX*1.D3                                                      HGKSPN
      IF(ABS(P1R0) .LT. PRM0)                   GO TO 450               HGKSPN
      FY(KI)=P1R0                                                       HGKSPN
  440 IF(IT .LT. ITE)        GO TO 410                                  HGKSPN
  450 P(J)=R(J)*T*RGAS*Z                                                HGKSPN
      IF(ISCRI .EQ. 0)       WRITE(6,1700) T,L,IT,R(J),P(J),P1R0        HGKSPN
      IF(ISCRI .EQ. 0 .AND.  MOD(MP,25) .EQ. 0)  WRITE(6,1400)          HGKSPN
  211 CONTINUE                                                          HGKSPN
  490 IF(ISCRI .EQ. 0)       GO TO  99                                  HGKSPN
C     SAVING THE FUNCTIONS                                              HGKSPN
  500 LAX=-1                                                            HGKSPN
      WMI=WMA                                                           HGKSPN
      CALL FENST8(MP,W,F,WMI,WMA,WED,LAX,KAXE,MASW)                     HGKSPN
      RMI=RMA                                                           HGKSPN
      CALL FENST8(MP,R,F,RMI,RMA,RED,LAX,KAXE,MASR)                     HGKSPN
      PMI=PMA                                                           HGKSPN
      CALL FENST8(MP,P,F,PMI,PMA,PED,LAX,KAXE,MASP)                     HGKSPN
      MMM=0                                                             HGKSPN
      NUMMER=NAPSZM(MMM)                                                HGKSPN
C     DENSITY ON THE SPINODALE                                          HGKSPN
      KLASSE=1013 + (IP-1)*40                                           HGKSPN
      CALL SDING8(KLASSE,NUMMER,MP                                      HGKSPN
     =,W,WMI,WMA,WED,NAMW,MASW,R,RMI,RMA,RED,NAMR,MASR,MODUS)           HGKSPN
C     PRESSURE ON THE SPINODALE                                         HGKSPN
      KLASSE=1016 + (IP-1)*40                                           HGKSPN
      CALL SDING8(KLASSE,NUMMER,MP                                      HGKSPN
     =,W,WMI,WMA,WED,NAMW,MASW,P,PMI,PMA,PED,NAMP,MASP,MODUS)           HGKSPN
   99 WRITE(6,2000)                                                     HGKSPN
          GO TO 100                                                     HGKSPN
C     ----------------------------------------------------------------- HGKSPN
 1000 FORMAT(2H +,39(2H++)/ 2H +,77X,1H+                                HGKSPN
     =/2H +,6X,15HPROGRAM HGKSPN ,33X,17HSTATUS : 94/12/09,6X,1H+       HGKSPN
     =/2H +,77X,1H+/2H +,17X                                            HGKSPN
     =,42HCALCULATES THE LIQUID AND VAPOR SPINODALS ,18X,1H+            HGKSPN
     =/ 2H +,77X,1H+/2H +,39(2H++) )                                    HGKSPN
 1100 FORMAT(/11H PROPERTY /,I3,5H / =:                                 HGKSPN
     =/20H "  0 "   STOP :                                              HGKSPN
     =/20H "  1 "   LIQID :                                             HGKSPN
     =/20H "  2 "   VAPOR :                  )                          HGKSPN
 1200 FORMAT(37H FUNCTION TO BE SAVED ? ( 0 = NO ) =: )                 HGKSPN
 1400 FORMAT(////)                                                      HGKSPN
 1500 FORMAT(' KI =',I3,2X,'Y  =',3G17.8/10X,'FY =',3G17.8)             HGKSPN
 1600 FORMAT(4X,'T',6X,'L',2X,'IT',7X,'RHSP',12X,'PSP',13X,'P1R0')      HGKSPN
 1700 FORMAT(F8.3,2I4,3G16.6)                                           HGKSPN
 2000 FORMAT(//////////////////////////////)                            HGKSPN
 9000 FORMAT(22H NO P.MIN FOUND AT T =,F10.3,4H !!! )                   HGKSPN
```

```
 9001 FORMAT(22H NO P.MAX FOUND AT T =,F10.3,4H !!! )                   HGKSPN
C     ----------------------------------------------------------------  HGKSPN
  999 STOP                                                              HGKSPN
      END                                                               HGKSPN
```

I.2 INR105.AQUA.FORT(HGKSAT)

```
      PROGRAM HGKSAT                                                    HGKSAT
C     CALCULATES SATURATED PROPERTIES OF THE WATER                      HGKSAT
C     USING THE HGK-CODE                                      94/07/29  HGKSAT
      IMPLICIT REAL*8(A-H,O-Z)                                          HGKSAT
      INTEGER NEIG(50)                                                  HGKSAT
      REAL*8 X(1000),TS(1000),FL(1000),FV(1000),XF(1000),FX(1000)       HGKSAT
      CHARACTER*8 NAMT,MAST,NAMF,MASF,NORMR,NORMW,MODUS,NAMP,MASP,NAMV  HGKSAT
      COMMON /MUFUN/ X,XMI,XMA,XED,FX,FMI,FMA,FED                       HGKSAT
      COMMON /ACONST/ WM,GASCON,TZ,AA,ZB,DZB,YB,UREF,SREF               HGKSAT
      COMMON /FCTS/ AD,GD,SD,UD,HD,CVD,CPD,DPDT,DVDT,CJTT,CJTH,Z,DPDD   HGKSAT
      DATA IRE/15/,IWR/16/,MODUS/'**OLD***'/,KPRI/2/,IA,IE,II/1,1000,1/ HGKSAT
     =,TAUS/1.D3/                                                       HGKSAT
      DATA NEIG/1,11,15,21,22,26,27,42,44,45,40*0/                      HGKSAT
      WRITE(6,1000)                                                     HGKSAT
      CALL SERDIO(IRE,NORMR,IWR,NORMW,MODUS,KPRI)                       HGKSAT
      IF(NORMR .NE. 'GRAPHIC8')                  GO TO 999              HGKSAT
      IF(NORMW .NE. 'GRAPHIC8')                  GO TO 999              HGKSAT
  100 WRITE(6,1100)                                                     HGKSAT
      READ(5,*) NEP                                                     HGKSAT
      IF(NEP .LE. 0)         GO TO 999                                  HGKSAT
  200 CALL SIGNAT(KLASSE,NUMMER,NAMF,MASF,NEIG)                         HGKSAT
      IF(KLASSE .LE. 0)      GO TO 100                                  HGKSAT
      CALL MUSTER(IA,IE,II,NAMT,MAST,NAMP,MASP,*200)                    HGKSAT
  220 CALL PG3277                                                       HGKSAT
      WRITE(6,1200)                                                     HGKSAT
      READ(5,*) ISCRI                                                   HGKSAT
      NAMF(5:8)='-HGK'                                                  HGKSAT
      NAMV=NAMF                                                         HGKSAT
      MP=0                                                              HGKSAT
      J=0                                                               HGKSAT
      DL=1.D0                                                           HGKSAT
      DV=5.D-3                                                          HGKSAT
      RGAS=GASCON*TAUS                                                  HGKSAT
      DO 111 I=IA,IE,II                                                 HGKSAT
      J=J+1                                                             HGKSAT
      IF(ISCRI .EQ. 0 .AND.  MOD(J,25) .EQ. 1)                          HGKSAT
     =    WRITE(6,1500) NAMF,NAMV                                       HGKSAT
      T=X(I)                                                            HGKSAT
      TS(J)=T                                                           HGKSAT
      CALL PCORR(T,P,DL,DV)                                             HGKSAT
      IF(KLASSE .EQ. 1)      THEN                                       HGKSAT
          FL(J)=P*1.D+6                                                 HGKSAT
          FV(J)=PS(T)*1.D6                                              HGKSAT
          GO TO 110                                                     HGKSAT
      ENDIF                                                             HGKSAT
      IF(KLASSE .EQ. 11)     THEN                                       HGKSAT
```

```
          FL(J)=DL*TAUS                                                 HGKSAT
          FV(J)=DV*TAUS                                                 HGKSAT
          GO TO 110                                                     HGKSAT
      ENDIF                                                             HGKSAT
      ROT = RGAS*T                                                      HGKSAT
      ZDUM = BASE(DL,T)                                                 HGKSAT
      CALL QQ(T,DL)                                                     HGKSAT
      CALL THERM(DL,T)                                                  HGKSAT
      IF(KLASSE .EQ. 15)     FL(J)=Z*ROT*DL*TAUS                        HGKSAT
      IF(KLASSE .EQ. 21)     FL(J)=DPDD*TAUS                            HGKSAT
      IF(KLASSE .EQ. 22)     FL(J)=DPDT*1.D6                            HGKSAT
      IF(KLASSE .EQ. 26)     FL(J)=CVD*RGAS                             HGKSAT
      IF(KLASSE .EQ. 27)     FL(J)=CPD*RGAS                             HGKSAT
      IF(KLASSE .EQ. 42)     FL(J)=GD*ROT                               HGKSAT
      IF(KLASSE .EQ. 44)     FL(J)=HD*ROT                               HGKSAT
      IF(KLASSE .EQ. 45)     FL(J)=SD*RGAS                              HGKSAT
      ZDUM = BASE(DV,T)                                                 HGKSAT
      CALL QQ(T,DV)                                                     HGKSAT
      CALL THERM(DV,T)                                                  HGKSAT
      IF(KLASSE .EQ. 15)     FV(J)=Z*ROT*DV*TAUS                        HGKSAT
      IF(KLASSE .EQ. 21)     FV(J)=DPDD*TAUS                            HGKSAT
      IF(KLASSE .EQ. 22)     FV(J)=DPDT*1.D6                            HGKSAT
      IF(KLASSE .EQ. 26)     FV(J)=CVD*RGAS                             HGKSAT
      IF(KLASSE .EQ. 27)     FV(J)=CPD*RGAS                             HGKSAT
      IF(KLASSE .EQ. 42)     FV(J)=GD*ROT                               HGKSAT
      IF(KLASSE .EQ. 44)     FV(J)=HD*ROT                               HGKSAT
      IF(KLASSE .EQ. 45)     FV(J)=SD*RGAS                              HGKSAT
  110 IF(ISCRI .EQ. 0)   WRITE(6,1700) J,T,FX(I),FL(J),FV(J)            HGKSAT
      IF(ISCRI .EQ. 0 .AND.  MOD(J,25) .EQ. 0)  WRITE(6,1400)           HGKSAT
  111 CONTINUE                                                          HGKSAT
      MP=J                                                              HGKSAT
      IF(ISCRI .EQ. 0)       GO TO 199                                  HGKSAT
      LAX=-1                                                            HGKSAT
      XMI=XMA                                                           HGKSAT
      CALL FENST8(MP,X,XF,XMI,XMA,XED,LAX,KAXE,MASX)                    HGKSAT
      KLASSE=KLASSE+1000                                                HGKSAT
      FMI=FMA                                                           HGKSAT
      CALL FENST8(MP,FL,XF,FMI,FMA,FED,LAX,KAXE,MASF)                   HGKSAT
      CALL SDING8(KLASSE,NUMMER,MP                                      HGKSAT
     =,X,XMI,XMA,XED,NAMT,MAST,FL,FMI,FMA,FED,NAMF,MASF,MODUS)          HGKSAT
      IF(KLASSE .EQ. 1001)   THEN                                       HGKSAT
          KLASSV=1002                                                   HGKSAT
      ELSE                                                              HGKSAT
          KLASSV=KLASSE+40                                              HGKSAT
      ENDIF                                                             HGKSAT
      FMI=FMA                                                           HGKSAT
      CALL FENST8(MP,FV,XF,FMI,FMA,FED,LAX,KAXE,MASF)                   HGKSAT
      CALL SDING8(KLASSV,NUMMER,MP                                      HGKSAT
     =,X,XMI,XMA,XED,NAMT,MAST,FV,FMI,FMA,FED,NAMV,MASF,MODUS)          HGKSAT
  199 WRITE(6,2000)                                                     HGKSAT
          GO TO 200                                                     HGKSAT
1000  FORMAT(2H +,39(2H++)/2H +,77X,1H+/2H +,6X,14HPROGRAM HGKSAT       HGKSAT
     =,34X,17HSTATUS : 94/07/29,6X,1H+/2H +,77X,1H+/2H +,14X            HGKSAT
     =,48HCALCULATES THE SATURATED PROPERTIES OF THE WATER,15X,1H+      HGKSAT
     =/2H + ,13X,51HAS FUNCTIONS OF THE TEMPERATURE WITH THE H.G.K-CODE HGKSAT
```

```
     =,13X,1H+/2H +,7X,30HUSING A SAMPLE (T,P)-FUNCTION                 HGKSAT
     =,33HAS A TEMPERATURE/PRESSURE-PATTERN,7X,1H+                      HGKSAT
     =/2H +,77X,1H+/2H +,39(2H++) )                                     HGKSAT
 1100 FORMAT(31H NEW PROPERTY ? ( 0 = STOP ) =: )                       HGKSAT
 1200 FORMAT(37H FUNCTION TO BE SAVED ? ( 0 = NO ) =: )                 HGKSAT
 1400 FORMAT(////)                                                      HGKSAT
 1500 FORMAT(1H0,2X,1HI,8X,2HT ,8X,2HP ,11X,A8,8X,A8)                   HGKSAT
 1700 FORMAT(I4,F12.3,3G16.8)                                           HGKSAT
 2000 FORMAT(//////////////////////////////////)                        HGKSAT
  999 STOP                                                              HGKSAT
      END                                                               HGKSAT
```

I.3 INR105.AQUA.FORT(TRNSAT)

```
      PROGRAM TRNSAT                                                    TRNSAT
C     CALCULATES TRANSPORT PROPERTIES OF THE WATER                      TRNSAT
C     ON THE SATURATTION LINE                                 94/12/07  TRNSAT
      IMPLICIT REAL*8(A-H,O-Z)                                          TRNSAT
      REAL*8 X(1000),F(1000),FL(1000),FV(1000)                          TRNSAT
      CHARACTER*8 NAMT,MAST,NAMF,MASF,NORMR,NORMW,MODUS                 TRNSAT
      COMMON /MUFUN/ X,TMI,TMA,TED ,F,FMI,FMA,FED                       TRNSAT
      COMMON /REDUF/  G0,S0,Z,P1T0,P1R0,CV0,CP0,ETA,XLMB,R1T0,CPC,H0    TRNSAT
      COMMON /SATIF / PS,TPS1T,TTPS2T,DL,DV,RL1T0,RV1T0,RHSL,PSL        TRNSAT
     =,RHSV,PSV,SL,GL,PL1R0,PL1T0,CVL,ETL,XLL,PV1R0,PV1T0,CVV,ETV,XLV   TRNSAT
      PARAMETER(TC=647.126D0,RGAS=4.61522D+2,DC=0.322)                  TRNSAT
      DATA IRE/15/,IWR/16/,MODUS/'**OLD***'/,KPRI/2/                    TRNSAT
      DATA NR/0/,IA,IE,II/1,1000,1/                                     TRNSAT
      WRITE(6,1000)                                                     TRNSAT
      CALL SERDIO(IRE,NORMR,IWR,NORMW,MODUS,KPRI)                       TRNSAT
      IF(NORMR .NE. 'GRAPHIC8')                  GO TO 999              TRNSAT
      IF(NORMW .NE. 'GRAPHIC8')                  GO TO 999              TRNSAT
  100 WRITE(6,1100) IPP                                                 TRNSAT
      READ(5,*) IPP                                                     TRNSAT
      IF(IPP .LE. 0)         GO TO 999                                  TRNSAT
      CALL MUSTER(IA,IE,II,NAMT,MAST,NAMF,MASF,*100)                    TRNSAT
      CALL PG3277                                                       TRNSAT
      IF(IPP .LE. 12)        THEN                                       TRNSAT
          KLASSE=34                                                     TRNSAT
          NAMF='   ETA.L'                                               TRNSAT
      ELSEIF(IPP .GE. 13)    THEN                                       TRNSAT
          KLASSE=35                                                     TRNSAT
          NAMF='LAMBDA.L'                                               TRNSAT
      ENDIF                                                             TRNSAT
      WRITE(6,1200)                                                     TRNSAT
      READ(5,*) ISCRI                                                   TRNSAT
      MP=0                                                              TRNSAT
      DO 111 I=IA,IE,II                                                 TRNSAT
      MP=MP+1                                                           TRNSAT
      IF(ISCRI .EQ. 0 .AND.  MOD(MP,25) .EQ. 1)                         TRNSAT
     =    WRITE(6,1500)                                                 TRNSAT
      OR=F(I)                                                           TRNSAT
      T=X(I)                                                            TRNSAT
C     SATURATION PRESSURE AND DENSITIES                                 TRNSAT
```

```
      CALL ONLYT(T,IECCH)                                               TRNSAT
      CALL SUBCRI(T)                                                    TRNSAT
C     CORRECTED VALUES FOR PS,RHOL,RHOV                                 TRNSAT
      DLK=DL                                                            TRNSAT
      DVK=DV                                                            TRNSAT
      CALL GPCORR(T,Q,DLK,DVK,IECCH)                                    TRNSAT
C     HEAT CONDUCTIVITIES                                               TRNSAT
      IF(T .LE. 645.0D0)     THEN                                       TRNSAT
C         SATURATED LIQUID                                              TRNSAT
          CALL REDUK(T,DLK,IECCH)                                       TRNSAT
          ETAL=ETA                                                      TRNSAT
          XLAL=XLMB                                                     TRNSAT
C         SATURATED VAPOR                                               TRNSAT
          CALL REDUK(T,DVK,IECCH)                                       TRNSAT
          ETAV=ETA                                                      TRNSAT
          XLAV=XLMB                                                     TRNSAT
      ELSE                                                              TRNSAT
      CALL TWOPHA(T)                                                    TRNSAT
          RHL=DL*1.D3                                                   TRNSAT
          CALL TRANSP(T,RHL,PL1R0,PL1T0,ETAL,XLAL)                      TRNSAT
          RHV=DV*1.D3                                                   TRNSAT
          CALL TRANSP(T,RHV,PV1R0,PV1T0,ETAV,XLAV)                      TRNSAT
      ENDIF                                                             TRNSAT
      IF(IPP .LE. 12)        THEN                                       TRNSAT
          FL(I)=ETAL                                                    TRNSAT
          FV(I)=ETAV                                                    TRNSAT
      ELSEIF(IPP .GE. 13)    THEN                                       TRNSAT
          FL(I)=XLAL                                                    TRNSAT
          FV(I)=XLAV                                                    TRNSAT
      ENDIF                                                             TRNSAT
      FF=FL(I)                                                          TRNSAT
      IF(ISCRI .EQ. 0)       THEN                                       TRNSAT
          RR=1.D2*(OR-FF)/OR                                            TRNSAT
          WRITE(6,1700) I,MP,T,OR,FF,RR                                 TRNSAT
          IF(MOD(MP,25) .EQ. 0)        WRITE(6,1400)                    TRNSAT
      ENDIF                                                             TRNSAT
  111 CONTINUE                                                          TRNSAT
      IF(ISCRI .EQ. 0)       GO TO  99                                  TRNSAT
      NUMMER=NAPSZM(NR)                                                 TRNSAT
      KLASSE=1034                                                       TRNSAT
      LAX=-1                                                            TRNSAT
      TMI=TMA                                                           TRNSAT
      CALL FENST8(MP,X,XF,TMI,TMA,TED,LAX,KAXE,MASX)                    TRNSAT
      FMI=FMA                                                           TRNSAT
      CALL FENST8(MP,FL,XF,FMI,FMA,FED,LAX,KAXE,MASF)                   TRNSAT
      CALL SDING8(KLASSE,NUMMER,MP                                      TRNSAT
     =,X,TMI,TMA,TED,NAMT,MAST,FL,FMI,FMA,FED,NAMF,MASF,MODUS)          TRNSAT
      KLASSE=KLASSE+40                                                  TRNSAT
      NAMF(8:8)='V'                                                     TRNSAT
      FMI=FMA                                                           TRNSAT
      CALL FENST8(MP,FV,XF,FMI,FMA,FED,LAX,KAXE,MASF)                   TRNSAT
      CALL SDING8(KLASSE,NUMMER,MP                                      TRNSAT
     =,X,TMI,TMA,TED,NAMT,MAST,FV,FMI,FMA,FED,NAMF,MASF,MODUS)          TRNSAT
   99 WRITE(6,2000)                                                     TRNSAT
          GO TO 100                                                     TRNSAT
```

```
1000 FORMAT(2H +,39(2H++)/ 2H +,77X,1H+/2H +,6X,14HPROGRAM TRNSAT      TRNSAT
    =,34X,17HSTATUS : 94/12/07,6X,1H+                                 TRNSAT
    =/2H +,77X,1H+/2H +,22X,32HCALCULATING A TRANSPORT PROPERTY       TRNSAT
    =,23X,1H+/ 2H +,77X,1H+/2H +,39(2H++) )                           TRNSAT
1100 FORMAT(/11H PROPERTY /,I3,5H / =:                                 TRNSAT
    =/50H "  0 "   STOP :                                             TRNSAT
    =/52H " 12 "   VISCOSITY OF THE SATURATED STATES :                TRNSAT
    =/52H " 13 "   THERMAL CONDUCTIVITY OF THE SAT. STATES :   )      TRNSAT
1200 FORMAT(37H FUNCTION TO BE SAVED ? ( 0 = NO ) =: )                 TRNSAT
1400 FORMAT(///)                                                       TRNSAT
1500 FORMAT(/1H0,3X,1HI,3X,2HMP,6X,8HTS ( K ),5X,10HOR ( S.I ),2X      TRNSAT
    =,8HFU.APPR.,3X,13HR. DEV. ( % ) )                                TRNSAT
1700 FORMAT(2I5,F12.4,2G15.6,F14.8)                                    TRNSAT
2000 FORMAT(/////////////////////////////////)                         TRNSAT
 999 STOP                                                              TRNSAT
     END                                                               TRNSAT
```

Appendix J. The IVA Routines

The following routines are specially fitted for the codesystem IVA3, /21/. The main variables are the temperature and the pressure, the corresponding density must be gained by inverting the isotherm T. All the properties used are strictly in SI-units.

J.1 INR105.AQUA.FORT(SAET01)

```
      SUBROUTINE SAET01(T,P                                             SAET01
     =,VW,VD,HW,HD,HWD,SW,SD,TS1P,CPW,CPD,ETW,ETD,XLMBW,XLMBD,SRFTS     SAET01
     =,KIND,LCATAS)                                                     SAET01
C     CALCULATES SATURATION PROPERTIES OF THE WATER                     SAET01
C     AS FUNCTIONS OF THE TEMPERATURE T   ( IF KIND = 0 )      OR       SAET01
C     AS FUNCTIONS OF THE PRESSURE P      ( IF KIND = 1 )               SAET01
C          FOR THE CODESYSTEM  IVA3                           94/12/09  SAET01
C     Q = PRESSURE IN MEGAPASCALS, D = DENSITY IN G/CM**3               SAET01
      IMPLICIT REAL*8(A-H,O-Z)                                          SAET01
      LOGICAL LCATAS                                                    SAET01
      COMMON /SATIF/  PS,TPS1T,TTPS2T,DL,DV,RL1T0,RV1T0,RHSL,PSL        SAET01
     =,RHSV,PSV,SL,GL,PL1R0,PL1T0,CVL,ETL,XLL,PV1R0,PV1T0,CVV,ETV,XLV   SAET01
      PARAMETER(TC=647.126D0,TCN=646.0D0,RGAS=4.61522D+2)               SAET01
      DATA UN/1.D0/,RAF/1.D3/                                           SAET01
      LCATAS=.FALSE.                                                    SAET01
      IECCH = 0                                                         SAET01
      Q=P*1.D-6                                                         SAET01
      IF(KIND .EQ. 0)       GOTO 100                                    SAET01
C     SATURATION TEMPERATURE AND DENSITIES                              SAET01
      CALL TCORR(T,Q,DLK,DVK,IECCH)                                     SAET01
      IF(IECCH .NE. 0)      THEN                                        SAET01
          LCATAS=.TRUE.                                                 SAET01
          WRITE(6,9001)                                                 SAET01
          RETURN                                                        SAET01
      ENDIF                                                             SAET01
          GO TO 200                                                     SAET01
  100 IF(T .GT. TC)         THEN                                        SAET01
          LCATAS=.TRUE.                                                 SAET01
          WRITE(6,9004) T                                               SAET01
          RETURN                                                        SAET01
      ENDIF                                                             SAET01
C     SATURATION PRESSURE AND DENSITIES                                 SAET01
      CALL ONLYT(T,IECCH)                                               SAET01
      IF(IECCH .NE. 0)      THEN                                        SAET01
          LCATAS=.TRUE.                                                 SAET01
          WRITE(6,9005)                                                 SAET01
          RETURN                                                        SAET01
      ENDIF                                                             SAET01
      CALL SUBCRI(T)                                                    SAET01
C     CORRECTED VALUES FOR PS,RHOL,RHOV                                 SAET01
      DLK=DL                                                            SAET01
      DVK=DV                                                            SAET01
      IF(T .GE. TCN)        GOTO 200                                    SAET01
```

```
      CALL GPCORR(T,Q,DLK,DVK,IECCH)                                    SAET01
C     OTHER PROPERTIES OF THE SATURATED STATES                          SAET01
C     VAPOR PRESSURE DERIVATIVE T*DPS/DT                                SAET01
  200 ROT=T*RGAS                                                        SAET01
      TS1P=T/TPS1T                                                      SAET01
      CALL TWOPHA(T)                                                    SAET01
C     LIQUID PROPERTIES                                                 SAET01
      RHL=DLK*RAF                                                       SAET01
      VW=UN/RHL                                                         SAET01
      GW=GL*ROT                                                         SAET01
      SW=SL*RGAS                                                        SAET01
      HW=GW+SW*T                                                        SAET01
      CPL=CVL+PL1T0*PL1T0/PL1R0                                         SAET01
      CPW=CPL*RGAS                                                      SAET01
      ETW=ETL                                                           SAET01
      XLMBW=XLL                                                         SAET01
      SRFTS=SIGMA(T)                                                    SAET01
C     VAPOR PROPERTIES                                                  SAET01
      RHV=DVK*RAF                                                       SAET01
      VD=UN/RHV                                                         SAET01
      GD=GW                                                             SAET01
      HWD=TPS1T*(VD-VW)                                                 SAET01
      SD=SW+HWD/T                                                       SAET01
      HD=HW+HWD                                                         SAET01
      CPV=CVV+PV1T0*PV1T0/PV1R0                                         SAET01
      CPD=CPV*RGAS                                                      SAET01
      ETD=ETV                                                           SAET01
      XLMBD=XLV                                                         SAET01
 9001 FORMAT(40H SAET01-ERROR. PRESSURE RANGE EXCEEDED.  )              SAET01
 9004 FORMAT(' T  =',F12.2,' IS A SUPERCRITICAL TEMPERATURE !'  )       SAET01
 9005 FORMAT(42H SAET01-ERROR. TEMPERATURE BELOW FREEZING. )            SAET01
 9008 FORMAT(39H SAET01-ERROR. DENSITY RANGE EXCEEDED.  )               SAET01
      RETURN                                                            SAET01
      END                                                               SAET01
```

J.2 INR105.AQUA.FORT(WASSER)

```
      SUBROUTINE WASSER(T,P                                             WASSER
     =,RW,HW,SW,CPW,VSW,ETW,XLMBW,PRNW,HW1P,RW1T,RW1P,SRFTS, LCATAW)    WASSER
C     CALCULATES THERMOPHYSICAL PROPERTIES OF THE WATER                 WASSER
C       AS FUNCTIONS OT THE TEMPERATURE AND PRESSURE                    WASSER
C           FOR THE CODESYSTEM IVA3                            94/12/07 WASSER
C     Q = PRESSURE IN MEGAPASCALS, D = DENSITY IN G/CM**3               WASSER
      IMPLICIT REAL*8(A-H,O-Z)                                          WASSER
      LOGICAL LCATAW                                                    WASSER
      COMMON /REDUF/  G0,S0,Z,P1T0,P1R0,CV0,CP0,ETA,XLMB,R1T0,CPC,H0    WASSER
      PARAMETER(TC=647.126D0,RGAS=4.61522D+2)                           WASSER
      DATA ZERO/0.D0/,UN/1.D0/                                          WASSER
      LCATAW=.FALSE.                                                    WASSER
      Q=P*1.D-6                                                         WASSER
      IECCH=0                                                           WASSER
      CALL ONLYT(T,IECCH)                                               WASSER
      IF(IECCH .NE. 0)       THEN                                       WASSER
```

```
          WRITE(6,9001)                                                 WASSER
          LCATAW=.TRUE.                                                 WASSER
          GO TO 99                                                      WASSER
      ENDIF                                                             WASSER
C     CALCULATING THE DENSITY                                           WASSER
      IF(T .GE. TC)          CALL DGFIND(T,Q,DX,IECCH)                  WASSER
      IF(T .LT. TC)          CALL DLFIND(T,Q,DX,IECCH)                  WASSER
      IF(IECCH .NE. 0)       THEN                                       WASSER
          IF(IECCH .LT. 0)   WRITE(6,9005)                              WASSER
          IF(IECCH .GT. 0)   WRITE(6,9006)                              WASSER
          LCATAW=.TRUE.                                                 WASSER
          GO TO 99                                                      WASSER
      ENDIF                                                             WASSER
      RW=DX*1.D3                                                        WASSER
      V=UN/RW                                                           WASSER
C     PROPERTIES AS TEMPERATURE-DENSITY-FUNCTIONS                       WASSER
      ROT=T*RGAS                                                        WASSER
      CALL REDUK(T,DX,IECCH)                                            WASSER
C     WRITE(6,'(16H  PREDUK IECCH =,I6)' ) IECCH                        WASSER
      IF(P1R0 .LE. ZERO)     THEN                                       WASSER
          WRITE(6,9003)                                                 WASSER
          LCATAW=.TRUE.                                                 WASSER
          GO TO 99                                                      WASSER
      ENDIF                                                             WASSER
      HW=H0*ROT                                                         WASSER
      SW=S0*RGAS                                                        WASSER
      CPW=CP0*RGAS                                                      WASSER
      P1R=P1R0*ROT                                                      WASSER
      ETW=ETA                                                           WASSER
      XLMBW=XLMB                                                        WASSER
C     VELOCITY OF SOUND                                                 WASSER
      P1RS=CPC*P1R                                                      WASSER
      VSW=SQRT(P1RS)                                                    WASSER
C     DH/DP , DRHO/DT , DRHO/DP                                         WASSER
      HW1P=(UN+R1T0)*V                                                  WASSER
      RW1T=(RW/T)*R1T0                                                  WASSER
      RW1P=UN/P1R                                                       WASSER
C     PRANDTL # , SURFACE TENSION                                       WASSER
      PRNW=CPW*ETW/XLMB                                                 WASSER
      SRFTS=0.D0                                                        WASSER
      IF(T .LT. TC)          SRFTS=SIGMA(T)                             WASSER
 9001 FORMAT(34H WASSER-ERROR. TEMPERATURE TO LOW.  )                   WASSER
 9002 FORMAT(55H WASSER-ERROR. NO WATER DENSITY FOUND AT THIS PRESSURE.)WASSER
 9003 FORMAT(45H WASSER-ERROR. DP/DRHO OF THE WATER VANISHES. )         WASSER
 9005 FORMAT(51H WASSER-ERROR. PRESSURE TO LOW FOR A WATER DENSITY. )   WASSER
 9006 FORMAT(52H WASSER-ERROR. PRESSURE TO HIGH FOR A WATER DENSITY. )  WASSER
   99 RETURN                                                            WASSER
      END                                                               WASSER
```

J.3 INR105.AQUA.FORT(DAMPF)

```
      SUBROUTINE DAMPF(T,P,RGAS,WGM                                     DAMPF
     =,RD,HD,SD,CPD,VSD,ETD,XLMBD,PRND,HD1P,RD1T,RD1P, LCATAD)          DAMPF
C     CALCULATES THERMOPHYSIVCAL PROPERTIES OF THE STEAM                DAMPF
C       AS FUNCTIONS OT THE TEMPERATURE AND PRESSURE  ( IF RD = 0 )     DAMPF
C                           OR                                          DAMPF
C       AS FUNCTIONS OT THE TEMPERATURE AND DENSITY   ( IF RD > 0 )     DAMPF
C           FOR THE CODESYSTEM IVA3                           94/12/07  DAMPF
C     Q = PRESSURE IN MEGAPASCALS, D = DENSITY IN G/CM**3               DAMPF
      IMPLICIT REAL*8(A-H,O-Z)                                          DAMPF
      LOGICAL LCATAD                                                    DAMPF
      COMMON /AKONST/ WM,TC,DC,GASCON,G1,G2,GF,TZ,UREF,SREF,TLOW        DAMPF
      COMMON /REDUF / G0,S0,Z,P1T0,P1R0,CV0,CP0,ETA,XLMB,R1T0,CPC,H0    DAMPF
      DATA ZERO/0.D0/,UN/1.D0/                                          DAMPF
      LCATAD=.FALSE.                                                    DAMPF
      RGAS=GASCON*1.D3                                                  DAMPF
      WGM=WM                                                            DAMPF
      Q=P*1.D-6                                                         DAMPF
      IECCH=0                                                           DAMPF
      CALL ONLYT(T,IECCH)                                               DAMPF
      IF(IECCH .NE. 0)       THEN                                       DAMPF
          WRITE(6,9001)                                                 DAMPF
          LCATAD=.TRUE.                                                 DAMPF
          GO TO 90                                                      DAMPF
      ENDIF                                                             DAMPF
      DX=RD*1.D-3                                                       DAMPF
      IF(RD .GT. ZERO)       GO TO 50                                   DAMPF
C     CALCULATING THE DENSITY                                           DAMPF
      IF(T .GE. TC)          CALL DGFIND(T,Q,DX,IECCH)                  DAMPF
      IF(T .LT. TC)          CALL DVFIND(T,Q,DX,IECCH)                  DAMPF
      IF(IECCH .NE. 0)       THEN                                       DAMPF
          IF(IECCH .LT. 0)   WRITE(6,9005)                              DAMPF
          IF(IECCH .GT. 0)   WRITE(6,9006)                              DAMPF
          LCATAD=.TRUE.                                                 DAMPF
          GO TO 90                                                      DAMPF
      ENDIF                                                             DAMPF
      RD=DX*1.D3                                                        DAMPF
   50 V=UN/RD                                                           DAMPF
C     PROPERTIES AS TEMPERATURE-DENSITY-FUNCTIONS                       DAMPF
      ROT=T*RGAS                                                        DAMPF
      CALL REDUK(T,DX,IECCH)                                            DAMPF
      IF(P1R0 .EQ. ZERO)     THEN                                       DAMPF
          WRITE(6,9003)                                                 DAMPF
          LCATAD=.TRUE.                                                 DAMPF
          GO TO 90                                                      DAMPF
      ENDIF                                                             DAMPF
      IF(P .EQ. ZERO)        P=Z*ROT*RD                                 DAMPF
      HD=H0*ROT                                                         DAMPF
      SD=S0*RGAS                                                        DAMPF
      CPD=CP0*RGAS                                                      DAMPF
      P1R=P1R0*ROT                                                      DAMPF
      ETD=ETA                                                           DAMPF
```

```
      XLMBD=XLMB                                                        DAMPF
C     VELOCITY OF SOUND                                                 DAMPF
      P1RS=CPC*P1R                                                      DAMPF
      VSD=SQRT(P1RS)                                                    DAMPF
C     DH/DP , DRHO/DT , DRHO/DP                                         DAMPF
      HD1P=(UN+R1T0)*V                                                  DAMPF
      RD1T=(RD/T)*R1T0                                                  DAMPF
      RD1P=UN/P1R                                                       DAMPF
C     PRANDTL #                                                         DAMPF
      PRND=CPD*ETD/XLMB                                                 DAMPF
 9001 FORMAT(33H DAMPF-ERROR. TEMPERATURE TO LOW.  )                    DAMPF
 9002 FORMAT(54H DAMPF-ERROR. NO STEAM DENSITY FOUND AT THIS PRESSURE. )DAMPF
 9003 FORMAT(44H DAMPF-ERROR. DP/DRHO OF THE STEAM VANISHES. )          DAMPF
 9005 FORMAT(50H DAMPF-ERROR. PRESSURE TO LOW FOR A STEAM DENSITY. )    DAMPF
 9006 FORMAT(51H DAMPF-ERROR. PRESSURE TO HIGH FOR A STEAM DENSITY. )   DAMPF
   90 RETURN                                                            DAMPF
      END                                                               DAMPF
```

J.4 INR105.AQUA.FORT(RDRDP)

```
      SUBROUTINE RDRDP(T,P,RD,RD1P,LCATAD)                              RDRDP
C     RETURNS FOR A GIVEN TEMPERATURE TX AND PRESSURE P                 RDRDP
C     THE DENSITY, RD AND THE DENSITY DERIVATIVE, RD1P = DRD/DP         RDRDP
C     OF THE STEAM FOR THE CODESYSTEM IVA3.                   94/12/07  RDRDP
C     Q = PRESSURE IN MEGAPASCALS, D = DENSITY IN G/CM**3               RDRDP
      IMPLICIT REAL*8(A-H,O-Z)                                          RDRDP
      LOGICAL LCATAD                                                    RDRDP
      COMMON /AKONST/ WM,TC,DC,GASCON,G1,G2,GF,TZ,UREF,SREF,TLOW        RDRDP
      COMMON /REDUF/  G0,S0,Z,P1T0,P1R0,CV0,CP0,ETA,XLAMB,R1T0,CPC,H0   RDRDP
      DATA ZERO/0.D0/,UN/1.D0/                                          RDRDP
      LCATAD=.FALSE.                                                    RDRDP
      RGAS=GASCON*1.D3                                                  RDRDP
      Q=P*1.D-6                                                         RDRDP
      IECCH=0                                                           RDRDP
      CALL ONLYT(T,IECCH)                                               RDRDP
      IF(IECCH .NE. 0)       THEN                                       RDRDP
          WRITE(6,9001)                                                 RDRDP
          LCATAD=.TRUE.                                                 RDRDP
          GO TO 90                                                      RDRDP
      ENDIF                                                             RDRDP
C     CALCULATING THE DENSITY                                           RDRDP
      IF(T .GE. TC)          CALL DGFIND(T,Q,DX,IECCH)                  RDRDP
      IF(T .LT. TC)          CALL DVFIND(T,Q,DX,IECCH)                  RDRDP
      IF(IECCH .NE. 0)       THEN                                       RDRDP
          IF(IECCH .LT. 0)   WRITE(6,9005)                              RDRDP
          IF(IECCH .GT. 0)   WRITE(6,9006)                              RDRDP
          LCATAD=.TRUE.                                                 RDRDP
          GO TO 90                                                      RDRDP
      ENDIF                                                             RDRDP
      RD=DX*1.D3                                                        RDRDP
      IF(DX .LT. ZERO)       THEN                                       RDRDP
          WRITE(6,9002)                                                 RDRDP
          LCATAD=.TRUE.                                                 RDRDP
```

```
            GO TO 90                                                    RDRDP
         ENDIF                                                          RDRDP
C        PROPERTIES AS TEMPERATURE-DENSITY-FUNCTIONS                    RDRDP
      50 CALL REDUK(T,DX,IECCH)                                         RDRDP
         IF(P1R0 .EQ. ZERO)     THEN                                    RDRDP
            WRITE(6,9003)                                               RDRDP
            LCATAD=.TRUE.                                               RDRDP
            GO TO 90                                                    RDRDP
         ENDIF                                                          RDRDP
         P1R=P1R0*T*RGAS                                                RDRDP
C        DRHO/DP                                                        RDRDP
         RD1P=UN/P1R                                                    RDRDP
    9001 FORMAT(33H RDRDP-ERROR. TEMPERATURE TO LOW.  )                 RDRDP
    9002 FORMAT(54H RDRDP-ERROR. NO STEAM DENSITY FOUND AT THIS PRESSURE. )RDRDP
    9003 FORMAT(44H RDRDP-ERROR. DP/DRHO OF THE STEAM VANISHES. )       RDRDP
    9005 FORMAT(50H RDRDP-ERROR. PRESSURE TO LOW FOR A STEAM DENSITY. ) RDRDP
    9006 FORMAT(51H RDRDP-ERROR. PRESSURE TO HIGH FOR A STEAM DENSITY. )RDRDP
      90 RETURN                                                         RDRDP
         END                                                            RDRDP
```

J.5 INR105.AQUA.FORT(VD1)

```
         SUBROUTINE VD1(T,VD,DUMMY)                                     VD1
C        VD1 CALCULATES - FOR A GIVEN TEMPOERATURE ( T ) IN K -         VD1
C        AN APPROXIMATED SPECIFIC VOLUME OF THE SATURATED STEAM ( VD )  VD1
C        IN M**3/KG.  VD1 IS A SUBROUTINE OF THE IVA3-CODE.    94/12/01 VD1
         IMPLICIT REAL*8(A-H,O-Z)                                       VD1
         DIMENSION PU(12)                                               VD1
         COMMON /SATIF/  PS,TPS1T,TTPS2T,DL,DV,RL1T0,RV1T0,RHSL,PSL     VD1
        =,RHSV,PSV,SL,GL,PL1R0,PL1T0,CVL,ETL,XLL,PV1R0,PV1T0,CVV,ETV,XLV VD1
         PARAMETER(TC=647.126D0,RHC=0.322D3,RGAS=4.61522D+2,TSW=620.D0) VD1
         PARAMETER(BR=2.03D0,BET=0.325D0)                               VD1
         DATA PU/-1.125999398D+0,42.15284538D+0,-538.0831003D+0         VD1
        =,4.150173161D+3,-20.80947751D+3,71.29507069D+3,-170.4446088D+3 VD1
        =,284.4989747D+3,-325.1357165D+3,242.4593090D+3,-106.2552030D+3 VD1
        =,20.74762670D+3/, UN/1.D0/                                     VD1
         DUMMY = 0.D0                                                   VD1
         IF(T .GE. TC)          THEN                                    VD1
            DUMMY = -1.D0                                               VD1
            VD = UN/RHC                                                 VD1
            RETURN                                                      VD1
         ENDIF                                                          VD1
         Y = (TC-T)/TC                                                  VD1
         IF(T .LE. TSW)         THEN                                    VD1
C        LOW TEMPERATURES                                               VD1
C        FACTOR OF REALITY OF THE VAPOR                                 VD1
            W = SQRT(Y)                                                 VD1
            CALL SSP(T)                                                 VD1
            ZV = PU(12)                                                 VD1
            DO 31 I=11,1,-1                                             VD1
      31    ZV = ZV*W+PU(I)                                             VD1
            VD = RGAS*T*ZV/PS                                           VD1
C        NEAR CRITICAL TEMPERATURES                                     VD1
```

```
      ELSE                                                              VD1
          AV = Y*(1.451962241D-0-Y*(5.041325559D-0-Y*Y*611.1810749D-0)) VD1
          OMV = UN-BR*Y**BET+AV                                         VD1
          VD = UN/(RHC*OMV)                                             VD1
      ENDIF                                                             VD1
      RETURN                                                            VD1
      END                                                               VD1
```

Appendix K. Service routines

The following routines are needed only by the 'string'-routines KAISOT and KAISOK

K.1 INR105.AQUA.FORT(SIGNAT)

SIGNAT creates the labels and distinguishing names for the properties calculated.
The first label of the property, KLASSE will be set according to the Table 38 on page 156, the second label, NUMMER will be equated with the date of the calculation.

```
      SUBROUTINE SIGNAT(KLASSE,NUMMER,NAMF,MASF,NEIG)                   SIGNAT
C     SIGNAT SETS THE CHARACTERISTICA OF A SERVUS-FUNCTION              SIGNAT
C     THERMAL PROPERTIES                                      94/12/09  SIGNAT
      IMPLICIT REAL*8(A-H,O-Z)                                          SIGNAT
      DIMENSION NEIG(50)                                                SIGNAT
      CHARACTER*8 NAMF,MASF                                             SIGNAT
      DATA NR/0/                                                        SIGNAT
      NUMMER=NAPSZM(NR)                                                 SIGNAT
      MASF='SI-UNIT'                                                    SIGNAT
      CALL PG3277                                                       SIGNAT
      WRITE(6,1000)                                                     SIGNAT
      KLASSE=MOD(KLASSE,100)                                            SIGNAT
  100 WRITE(6,1100) KLASSE                                              SIGNAT
      READ(5,*) KLASSE                                                  SIGNAT
      IF(KLASSE .EQ. 0)      GO TO 999                                  SIGNAT
      MP=0                                                              SIGNAT
      DO 111 I=1,50                                                     SIGNAT
      IF(NEIG(I) .EQ. 0)     GO TO 190                                  SIGNAT
      IF(NEIG(I) .EQ. KLASSE)          GO TO 200                        SIGNAT
  111 MP=MP+1                                                           SIGNAT
  190 KAPRO = 0                                                         SIGNAT
      WRITE(6,1300) (NEIG(I),I=1,MP)                                    SIGNAT
          GO TO 100                                                     SIGNAT
C...................................................................... SIGNAT
  200 IF(KLASSE .EQ. 1 .OR. KLASSE .EQ. 2 .OR.                          SIGNAT
     =   KLASSE .EQ. 15 .OR. KLASSE .EQ. 16)     MASF='PASCAL  '        SIGNAT
      IF(KLASSE .EQ. 3 .OR. KLASSE .EQ. 22)      MASF='PASCAL/K'        SIGNAT
      IF(KLASSE .EQ. 4)      MASF='K/PASCAL'                            SIGNAT
      IF(KLASSE .EQ. 5)      MASF='K       '                            SIGNAT
      IF(KLASSE .GE. 11 .AND. KLASSE .LE. 13)    MASF='KG/M**3 '        SIGNAT
      IF(KLASSE .EQ. 17)     MASF='M**3/KG '                            SIGNAT
      IF(KLASSE .EQ. 18 .OR. KLASSE .EQ. 36)     MASF='- 1 -   '        SIGNAT
      IF(KLASSE .EQ. 26 .OR. KLASSE .EQ. 27 .OR.                        SIGNAT
     =   KLASSE .EQ. 45)     MASF='J/KG/K  '                            SIGNAT
      IF(KLASSE .EQ. 31)     MASF='M/SEC   '                            SIGNAT
      IF(KLASSE .EQ. 32)     MASF='N/M     '                            SIGNAT
      IF(KLASSE .EQ. 34)     MASF='PA SEC  '                            SIGNAT
      IF(KLASSE .EQ. 35)     MASF='W/(K*M) '                            SIGNAT
      IF(KLASSE .GE. 41 .AND. KLASSE .LE. 44)    MASF='J/KG    '        SIGNAT
C...................................................................... SIGNAT
      IF(KLASSE .EQ. 1)      NAMF='  P*    '                            SIGNAT
      IF(KLASSE .EQ. 2)      NAMF='P*AX    '                            SIGNAT
```

```
      IF(KLASSE .EQ. 3)      NAMF='DP*/DT  '                           SIGNAT
      IF(KLASSE .EQ. 4)      NAMF='DT*/DP  '                           SIGNAT
      IF(KLASSE .EQ. 5)      NAMF='  T*    '                           SIGNAT
C....................................................................... SIGNAT
      IF(KLASSE .EQ. 11)     NAMF='  RH????'                           SIGNAT
      IF(KLASSE .EQ. 12)     NAMF='RHAX????'                           SIGNAT
      IF(KLASSE .EQ. 13)     NAMF=' RHS????'                           SIGNAT
      IF(KLASSE .EQ. 15)     NAMF='   P????'                           SIGNAT
      IF(KLASSE .EQ. 16)     NAMF='  PS????'                           SIGNAT
      IF(KLASSE .EQ. 17)     NAMF='   V????'                           SIGNAT
      IF(KLASSE .EQ. 18)     NAMF='   Z????'                           SIGNAT
C....................................................................... SIGNAT
      IF(KLASSE .EQ. 21)     NAMF='  PR????'                           SIGNAT
      IF(KLASSE .EQ. 22)     NAMF='  PT????'                           SIGNAT
      IF(KLASSE .EQ. 23)     NAMF=' RHP????'                           SIGNAT
      IF(KLASSE .EQ. 24)     NAMF=' RHT????'                           SIGNAT
      IF(KLASSE .EQ. 26)     NAMF='  CV????'                           SIGNAT
      IF(KLASSE .EQ. 27)     NAMF='  CP????'                           SIGNAT
C....................................................................... SIGNAT
      IF(KLASSE .EQ. 31)     NAMF='  VS????'                           SIGNAT
      IF(KLASSE .EQ. 32)     NAMF='SIGMA   '                           SIGNAT
      IF(KLASSE .EQ. 34)     NAMF=' ETA????'                           SIGNAT
      IF(KLASSE .EQ. 35)     NAMF=' LAM????'                           SIGNAT
      IF(KLASSE .EQ. 36)     NAMF=' PRN????'                           SIGNAT
C....................................................................... SIGNAT
      IF(KLASSE .EQ. 41)     NAMF='   A????'                           SIGNAT
      IF(KLASSE .EQ. 42)     NAMF='   G????'                           SIGNAT
      IF(KLASSE .EQ. 43)     NAMF='   U????'                           SIGNAT
      IF(KLASSE .EQ. 44)     NAMF='   H????'                           SIGNAT
      IF(KLASSE .EQ. 45)     NAMF='   S????'                           SIGNAT
      IF(KLASSE .EQ. 46)     NAMF='  UR????'                           SIGNAT
      IF(KLASSE .EQ. 47)     NAMF='  HP????'                           SIGNAT
 1000 FORMAT(//44H TABLE OF THE PROPERTY-NUMBERS : ( END = 0 )           SIGNAT
     =/'     |  1      2      3      4      5      6      7      8   '  SIGNAT
     =/'_____________________________________________________________'  SIGNAT
     =/'  0  |  P*     P*~  DP/DT  DT/DP    T*                       '  SIGNAT
     =/' 10  |  RH     RH~  RH.SP           P     P.SP    V      Z   '  SIGNAT
     =/' 20  | P1R    P1T    R1P    R1T           CV     CP          '  SIGNAT
     =/' 30  | V.S    SIG           ETA  LAMBDA   PRN                '  SIGNAT
     =/' 40  |  A      G      U      H      S     U1R    H1P         ') SIGNAT
 1100 FORMAT(31H ENTER NUMBER OF THE PROPERTY /,I4,5H / =: )           SIGNAT
 1300 FORMAT(50H ONLY THE FOLLOWING PROPERTIES CAN BE CALCULATED :      SIGNAT
     =/(5X,18(I3,:,1H,)))                                              SIGNAT
  999 RETURN                                                            SIGNAT
      END                                                               SIGNAT
```

K.2 INR105.AQUA.FORT(MUSTER)

MUSTER helps to select a sample function from a SERVUS dataset.

```
      SUBROUTINE MUSTER(IA,IE,II,NAMT,MAST,NAMF,MASF,*)                 MUSTER
C     SELECTS A SAMPLE FUNCTION F ( T )                       94/07/29  MUSTER
      IMPLICIT REAL*8(A-H,O-Z)                                          MUSTER
      REAL*8 T(1000),F(1000)                                            MUSTER
      CHARACTER*8 NAMT,MAST,NAMF,MASF                                   MUSTER
      COMMON /MUFUN/ T,TMI,TMA,TED,F,FMI,FMA,FED                        MUSTER
      DATA LOK,LOS/0,0/                                                 MUSTER
  100 WRITE(6,1000)     LOS                                             MUSTER
      READ(5,*)         LOS                                             MUSTER
      IF(LOS .LE. 0)         RETURN 1                                   MUSTER
      CALL SDEXG8(KLASSE,NUMMER,MP                                      MUSTER
     =,T,TMI,TMA,TED,NAMT,MAST,F,FMI,FMA,FED,NAMF,MASF,LOS)             MUSTER
      WRITE(6,1100)                                                     MUSTER
      READ(5,*) LOK                                                     MUSTER
      IF(LOK .NE. 0)         GO TO 100                                  MUSTER
      IE=MAX(IE,MP)                                                     MUSTER
  120 WRITE(6,1300) IA,IE,II                                            MUSTER
      READ(5,*)     IA,IE,II                                            MUSTER
      IF(IA*IE*II .EQ. 0)    RETURN 1                                   MUSTER
      IE=MIN(IE,MP)                                                     MUSTER
      IA=MAX(IA,1)                                                      MUSTER
C                                                                       MUSTER
 1000 FORMAT(52H ENTER SERIAL NUMBER OF THE (X,FX)-MASTER-FUNCTION /    MUSTER
     =,I4,19H / =:   ( 0 = END )  )                                     MUSTER
 1100 FORMAT(36H FUNCTION OK =: ( 0 = YES | 9 = NO ) )                  MUSTER
 1300 FORMAT(45H ENTER BEGIN, END & INCREMENTUM ( 0 = END ) /           MUSTER
     =,I3,2H ,,I4,2H ,,I3,16H / =: , =: , =:  )                         MUSTER
      RETURN                                                            MUSTER
      END                                                               MUSTER
```

K.3 INR105.AQUA.FORT(NAPSZM)

NAPSZM turns the actual date into a number, the number of the day in the year.

```
      INTEGER FUNCTION NAPSZM(K)                                        NAPSZM
C     REURNS THE NUMBER OF THE DAY                            92/01/13  NAPSZM
      CHARACTER JAN*64                                                  NAPSZM
      CALL JOBINF(JAN)                                                  NAPSZM
      NAPSZM=K-K                                                        NAPSZM
      JUF=1                                                             NAPSZM
      DO 11 J=32,30,-1                                                  NAPSZM
      I=ICHAR(JAN(J:J))                                                 NAPSZM
      IF(I .GT. 240 .AND. I .LT. 250)       NAPSZM=NAPSZM+(I-240)*JUF   NAPSZM
   11 JUF=JUF*10                                                        NAPSZM
      RETURN                                                            NAPSZM
      END                                                               NAPSZM
```

Appendix L. Return codes

Table 36. Table of return codes, IECCH

IECCH	Reason
0	density found
-1,-2	pressure to low for a density
1, 2	pressure to high for a density
3	iteration failed
-4	temperature to low for any density
4	density too high for any temperature

Appendix M. TSO-procedures, calculating thermal properties of the water

Table 37. TSO-procedures in the dataset INR105.AQUA.CLIST

Member	Routine	Procedure	Dataset needed
WATER	WATER	Water properties point by point / results => dataset	-
KAISOT	KAISOT	Explores water properties along isochores	AQUAT.DATA
KAISOK	KAISOK	Explores water properties along isotherms	AQUA.DATA
HGK	NBSNRC	Water properties point by point / H.G.K. code	-
HGKIST	HGKIST	Explores water properties along isotherms / H.G.K. code	AQUA.DATA
HGKSAT	HGKSAT	Calculates properties in the saturated states / H.G.K. code	AQUAT.DATA
HGKSPN	HGKSPN	Calculates the spinodals / H.G.K. code	AQUAT.DATA
TRNSAT	TRNSAT	Calculates transport properties in the saturated states	AQUAT.DATA
TCORR	$TCORR	Checks the routines TCORR, GPCORR	-
TDFIND	$TDFND	Checks the routines DGFIND, DLFIND, DVFIND	-
TAXSL	$TAXSL	Checks properties in the saturated states	AQUAT.DATA
TWASSER	$TWSSR	Checks the routine WASSER	AQUA.DATA
TDAMPF	$TDMPF	Checks the routine DAMPF	AQUA.DATA
TST01	$TST01	Checks the routine SAET01	AQUAT.DATA
TSTEX	$TSTEX	Checks the routine SAETEX	AQUAT.DATA

Appendix N. Thermal property labels

Table 38. Thermal property labels "KLASSE" used in AQUA.DATA

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
0	$P^{\times}$	$P^{\times}_{\approx}$	dP*/dT	dT/dP*	T*					
10	ρ_L	$\rho_{L\approx}$	ρ_{sL}		P_L	P_{sL}	v_L	Z_L		
20	$\partial P/\partial \rho_L$	$\partial P/\partial T_L$	$\partial \rho/\partial P_L$	$\partial \rho/\partial T_L$		C_{VL}	C_{PL}			
30	v_{sL}	σ		η_L	λ_L	PrN_L				
40	A.L	G.L	U.L	H.L	S.L	$\partial U/\partial \rho_L$	$\partial H/\partial P_L$			
50	ρ_V	$\rho_{V\approx}$	ρ_{sV}		P_V	P_{sV}	v_V	Z_V		
60	$\partial P/\partial \rho_V$	$\partial P/\partial T_V$	$\partial \rho/\partial P_V$	$\partial \rho/\partial T_V$		C_{VV}	C_{PV}			
70	v_{sV}			η_V	λ_V	PrN_V				
80	A.V	G.V	U.V	H.V	S.V	$\partial U/\partial \rho_V$	$\partial H/\partial P_V$			