FILE

Institute of
Oceanographic Sciences
Deacon Laboratory

INTERNAL DOCUMENT No. 313

The Equation of state algorithms used by the FRAM model

D J Webb

1992

Natural Environment Research Council

INSTITUTE OF OCEANOGRAPHIC SCIENCES
DEACON LABORATORY

INTERNAL DOCUMENT No. 313

The Equation of state algorithms used by the FRAM model

D J Webb

1992

Wormley
Godalming
Surrey GU8 5UB
Tel +44-(0)428 684141
Telex 858833 OCEANS G
Telefax +44-(0)428 683066

DOCUMENT DATA SHEET

AUTHOR

WEBB, D J

PUBLICATION DATE 1992

TITLE

The Equation of state algorithms used by the FRAM model.

REFERENCE

Institute of Oceanographic Sciences Deacon Laboratory, Internal Document, No. 313, 34pp.
(Unpublished manuscript)

ABSTRACT

FRAM, the Fine Resolution Antarctic Model, used two equations of state. During the main run, the Eckart equation of state was used. During the sea ice run, a scheme based on the international equation of state for seawater (EOS80) was used. This report documents the two algorithms, discusses their errors and compares them with other equation of state algorithms.

The report recommends that future numerical models should use a polynomial equation of state (e.g. Bryan and Cox (1972), Levitus and Isayev (1992)), with polynomial coefficients calculated from an accurate reference equation of state, such as the one described in Appendix I.

KEYWORDS

EQUATION OF STATE
NUMERICAL MODELLING
PROJECT - FRAM

ISSUING ORGANISATION

Institute of Oceanographic Sciences
Deacon Laboratory
Wormley, Godalming
Surrey GU8 5UB. UK.

Director: Colin Summerhayes DSc

Telephone Wormley (0428) 684141
Telex 858833 OCEANS G.
Facsimile (0428) 683066

Copies of this report are available from: **The Library,** *PRICE* *£0.00*

CONTENTS Page

1.0 INTRODUCTION

FRAM, the UK Fine Resolution Antarctic Model, (Webb et al 1991) was based on the Cox (1984) version of the Bryan-Cox-Semtner ocean general circulation model (Bryan 1969, Semtner 1974, Cox 1984).

This is usually classified as a primitive equation model because it uses the full advection-diffusion equation for potential temperature and salinity and a full horizontal momentum equation. During each timestep of the model these equations are solved once for each of the model grid points.

The model assumes hydrostatic balance in the vertical and places a rigid lid on the model ocean in order to eliminate surface gravity waves. Although this introduces a stream function for which an extra equation must be solved, it allows the use of a long time step, making the model computationally efficient.

Calculating the density

The density of sea water is calculated at two places within the model. The density is first needed, while timestepping the momentum equation, to calculate the horizontal pressure gradient. It is again needed while checking for convection at the end of each timestep. During the test for convection at least two calculations of density are needed, once for comparison with the grid point above and once for comparison with the grid point below. If convection does occur, then depending on the algorithm used, further calculations of density may be required (Killworth, 1989).

Thus, during each timestep of the model, the density is calculated at least three times at each grid point. This can be computationally expensive and is made more so because the model uses potential temperature, salinity and depth, whereas the equation of state defining the density of sea water (EOS80) defines it as a function of in situ temperature, salinity and pressure. Model depth therefore has to be converted to pressure and potential temperature converted to in situ temperature.

This is the scheme used by the 'reference' equation of state algorithm, described in Appendix I, which was developed to check the precision of the other algorithms described in this note. Although it is very precise, it requires many hundreds of floating point calculations for each grid point, and for this reason is impractical to use in large ocean general circulation models.

To make the computational problem more manageable, a simpler equation of state is usually used. Thus Semtner (1974) used the simplifed Eckart (1958) equation of state. Ignoring calculations whose overhead can be spread over a number of grid points, this requires 20 floating point operations per model grid point.

Cox (1984) included subroutines for both the Eckart equation of state and the Bryan and Cox (1972) polynomial approximation to the Knudsen-Eckman equation of state. The latter requires 19 floating point operations per grid point.

2.0 THE FRAM MODEL RUN

Tests carried out at the beginning of the FRAM project indicated that the Eckart equation was more accurate than the Bryan and Cox polynomial in the surface waters surrounding Antarctica. For this reason the Eckart equation of state was used during the main sixteen year run of the FRAM model[1] .

Initially the main run was planned to include a sea ice model. Such a model needs to know the air temperature, humidity and cloud cover but unfortunately when the run was started suitable data sets were not available for the sea ice covered regions around Antarctica. The main run therefore uses a simplified boundary condition in which it represents the surface flux of salt and heat by relaxing the surface temperature and salinity fields towards the values of the Levitus (1982) observational dataset.

Towards the end of the FRAM project, suitable surface forcing fields became available, so part of the main run (from year 10.0 onwards) was repeated with a full sea ice model. This produced deep convection in all the regions where sea ice was forming. Some convection was to be expected because of the increased salinity that arises when sea ice is formed. However, the amount of convection was far too large and the warm water brought from below to the surface was enough to melt the sea ice that had been formed.

The problem was investigated and it was found that the fault lay with the Eckart equation of state. In the surface waters around Antarctica, the difference in density it predicts between the cold, fresh surface waters and the warmer, saline underlying waters, is far too small. As a result, when sea ice forms, the slight increase in salinity of the surface layers makes the whole system unstable and triggers deep convection.

To overcome the problem a new equation of state algorithm was quickly developed making use of algorithms used by the sea-going community at IOSDL. In this algorithm, the depth to pressure conversion is carried out using a polynomial inverse of the Fofonoff and Millard (1983) version of the Saunders and Fofonoff (1976) algorithm.

The standard method for transforming temperature adiabatically from one pressure to another (Fofonoff and Millard, 1983) is computationally expensive, so a faster perturbation based method is used. Finally the density is calculated using the Fofonoff and Millard algorithm for the international equation of state seawater (EOS80) (Fofonoff and Millard 1983, UNESCO 1981a, b, Gill 1982).

---

[1] The test results were wrong, so the Eckart equation should not have been used. The reason for the mistake is unknown and cannot now be traced.

The resulting 'FRAM equation of state' was used for a repeat run of the FRAM model from years 10.0 to 12.0. It appeared to give excellent results. The remainder of this note discusses the Eckart and FRAM equations of state in detail and compares their results with those from a reference equation of state. The latter is documented more fully in Appendix I.

3.0 THE ECKART EQUATION OF STATE

Eckart (1958) developed an equation of state for sea water using the Tumlirz (1909) equation for relating pressure and specific volume

$$(p+p_0)*(V-V_0) = \lambda, \qquad (1$$

where p is the pressure, V the specific volume and p_0, V_0 and λ are functions of temperature and salinity. Rearranging the equation we obtain

$$V = V_0 + \lambda/(p+p_0),$$

or if ρ is the density,

$$\rho = 1/[V_0 + \lambda/(p+p_0)] \qquad (2$$

In trying to derive expressions for p_0, V_0 and λ, Eckart found a number of inconsistencies between different sets of measurements then available. He used the data sets which were most self consistent, and found that they were best fitted by the equations

$$p_0 = 5890 + 38*T - 0.375*T^2 + 3*s, \qquad (3$$

$$V_0 = (1 - 10^{-4}\beta), \text{ where } \beta = 3020, \qquad (4$$

$$\lambda = 1779.5 + 11.25*T - 0.0745*T^2 - (3.8 + 0.01*T)*s. \qquad (5$$

In the above equations the *in-situ* temperature T is measured in degrees Centigrade, salinity s is measured in parts per thousand, and the pressure p is measured in atmospheres.

Equations 2-5 define the Eckart equation of state as used by Semtner (1973) and Cox (1984). The algorithm they used is included in the first subroutine of Appendix II. Because the model measures salinity in parts per part (and not the more conventional parts per thousand), all the coefficient terms linear in s are multiplied by 1000 in the subroutine. Also the depth z, in centimetres, is converted to pressure, in atmospheres, using the relation

$$p = 1 + z/1013. \qquad (6$$

The density calculated using the Eckart equation of state is compared with the reference equation of state (Appendix I), in figures 1 and 2. Figure 1 is similar to figure 2a of Bryan and Cox (1972), but their calculation was based on the Knudsen equation of state (Knudsen 1901, Fofonoff 1962) instead of the UNESCO equation of state.

4.0 THE FRAM ALGORITHM

As discussed above, the model variables are potential temperature, salinity and depth. These need to be converted to in situ temperature, salinity and pressure before the EOS80 algorithm can be used.

4.1 DEPTH TO PRESSURE

Depth is transformed to pressure by inverting the Fofonoff and Millard (1983) version of the Saunders and Fofonoff (1976) algorithm. They first calculate gravity g as a function of latitude and depth, using the equations:

```
x = sin(latitude)**2
g = 9.780318*(1.0+(5.2788E-3+2.36E-5*x)*x + 1.092e-6*p,        (7
```

where p is the pressure measured in decibars.

Saunders and Fofonoff assume a standard temperature and salinity profile for the ocean and derive a fourth order polynomial to convert pressure to depth. It gives a pressure correction at 3,000 m of about 50 decibars. In addition there is a term (ranging from -1.5 to 2.5 decibars at 1000 m and 0 to 4 decibars at 5000 m), which corresponds to the local dynamic height anomaly. Because this has a much smaller effect on the in situ pressure, it is neglected.

In the FRAM algorithm, g is calculated using equation 7 but with the assumption that the pressure in decibars is numerically equal to the depth in metres. The inverse of the fourth order pressure to depth polynomial was approximated by a fifth order polynomial with coefficients calculated using the equations 3.6.25 of Abramowitz and Stegun (1964). These coefficients and other constant quantities are calculated during the first entry into the subroutine.

The error of the FRAM inverse (shown in figure 3a) is less than 0.04 decibar at 5,000 m and less than 0.25 decibar at 10,000 m. This is less than the dynamic height term and for most purposes can be neglected. A fourth order version of the inverse polynomial gives an error of 0.04 decibar at 5,000 m and 0.4 decibar at 10,000 m.

Saunders (1981) independently revised the pressure to depth algorithm to allow for the new EOS80 international equation of state. Figure 3b shows the differences between the polynomial results and the Saunders (1981) revised algorithm.

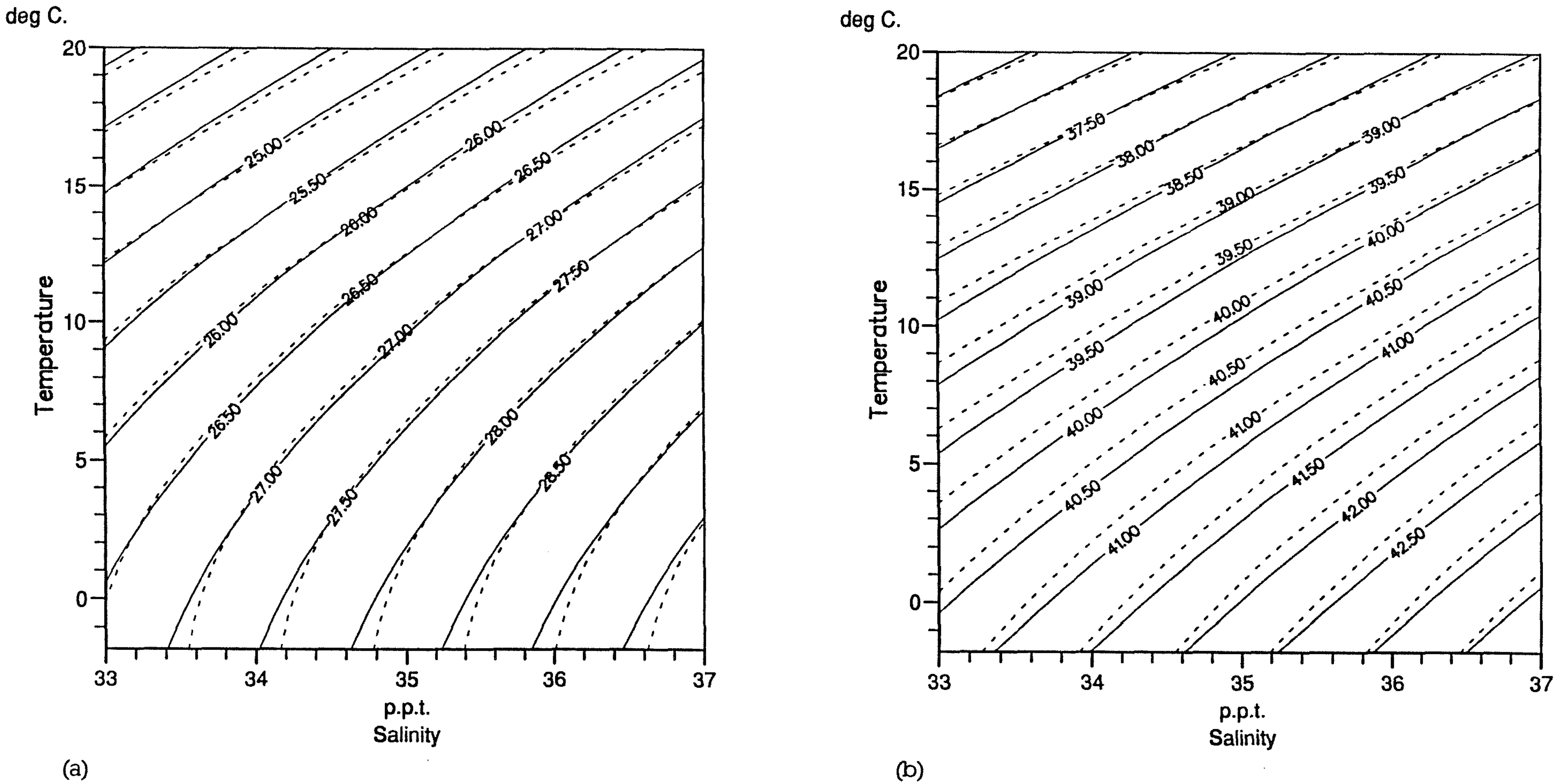


Figure 1. Density contours calculated using the Eckart equation of state (solid lines) compared with the reference calculation (dotted lines) at depths of (a) 0 m and (b) 3000 m. The density contours are in sigma units (i.e. kg/m^3-1000.0).

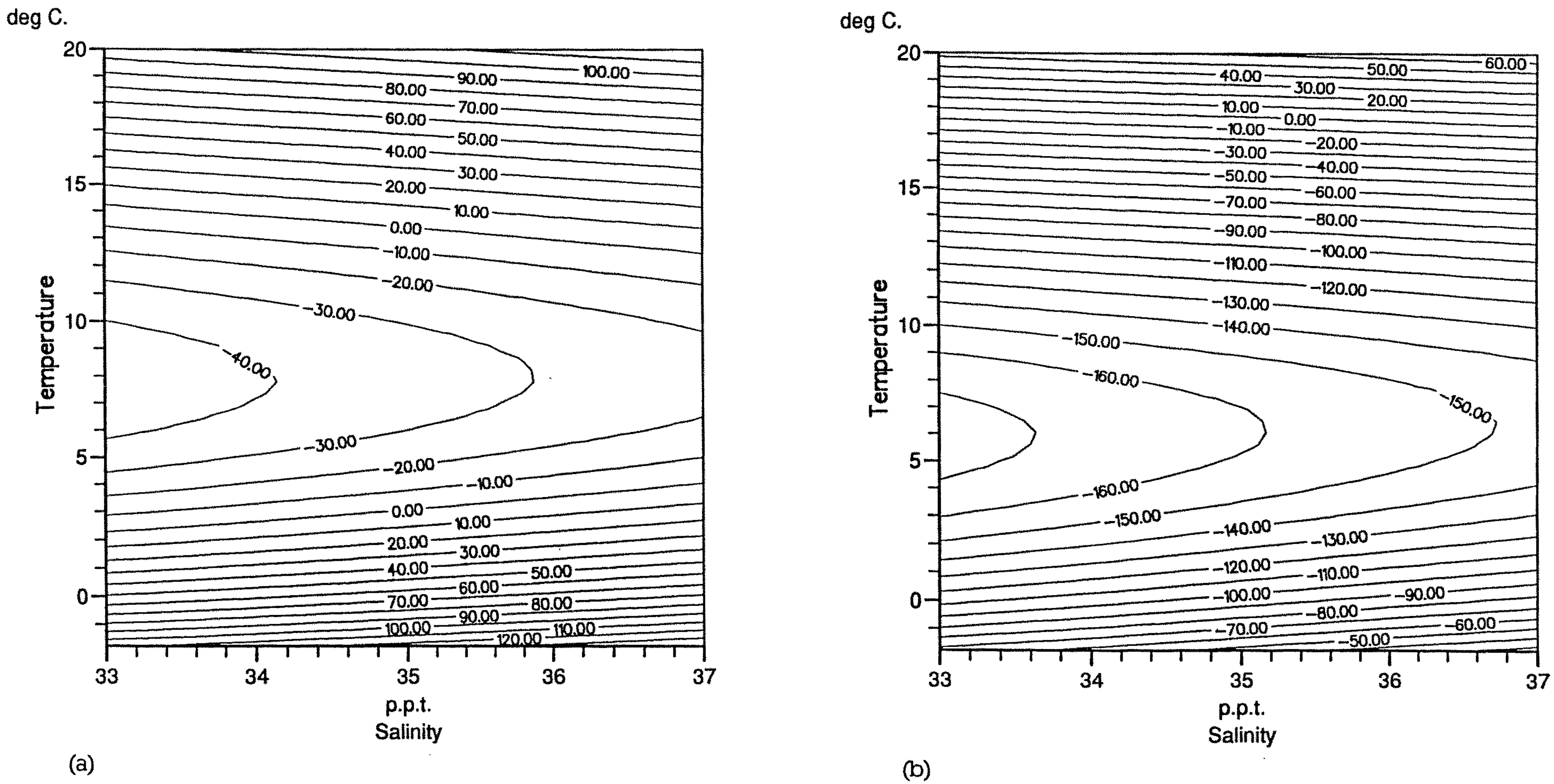


Figure 2. The difference between the Eckart equation of state and the reference calculation at depths of (a) 0 m and (b) 3000 m. The contour units are 10^{-3} sigma units.

4.2 POTENTIAL TEMPERATURE AND IN SITU TEMPERATURE

Potential temperature, as defined in the model, is the temperature a parcel of water will have if moved adiabatically from its in situ pressure to the surface. If the initial temperature, salinity and pressure are T_1, s and p_1 and the final temperature and pressure are T_2 and p_2, then

$$T_2 = T_1 + \int_{p_1}^{p_2} \Gamma(T,s,p)\, dp, \qquad (8$$

where Γ is the adiabatic lapse rate and T is the temperature of the parcel of water when moved adiabatically to pressure p. Note that in the EOS80 equation of state and other oceanographic formulae, zero pressure refers to the ocean surface where the absolute pressure is one atmosphere.

Fofonoff (1977) presents an algorithm for carrying out the integral using a fourth order Runge-Kutta scheme. For the adiabatic lapse rate he uses Bryden's (1973) polynomial equation, which contains 15 terms and has to be called four times by the Runge-Kutta scheme. As a result the temperature conversion takes about 130 floating point operations. This represents a considerable fraction of the computational requirement of the FRAM model and a modified scheme was developed to speed up the calculation.

From equation 8, the transformation needed in the model to convert potential temperature to in situ temperature is

$$T_2 = T_1 + \int_{o}^{p} \Gamma(T,s,p)\, dp \qquad (9$$

The modified scheme makes use of a number of useful properties of the adiabatic temperature gradient Γ. The first of these is that, during the transformation, the effect of the change in temperature on the adiabatic temperature gradient is small enough to be treated as a perturbation.

For example, when water with potential temperature of 0°C and salinity of 35 NSU moves adiabatically from the surface to 3000 m, the change in temperature is 0.19°C. This corresponds to a mean value of Γ of 6×10^{-5} °C/decibar. If we then calculate the change in Γ due to this small change in temperature, we find that it is only 0.1×10^{-5}°C/decibar. Thus, although the effect of the temperature change during the adiabatic transform is not so small that it can be neglected, it is small enough that higher order temperature effects can be neglected.

On this basis, equation 9 is approximated by

$$T_2 = T_1 + \int_{o}^{p} \Gamma(T(p/2),s,p)\, dp \qquad (10$$

where the adiabatic temperature gradient is only calculated at the temperature corresponding to half the final pressure.

Also, because T(p/2) only needs to be known approximately, it can be approximated by

$$T(p/2) = T_1 + \Gamma\,(T_1,s,0)*p/2 \qquad (11$$

The polynomial for Γ simplifies when the pressure is zero, so this only needs a few floating point operations. Finally, as Γ is a polynomial in pressure, equation 11 can be integrated analytically to give a polynomial which again only needs a few floating point operations to evaluate.

The resulting algorithm for the temperature transformation needs 46 floating point operations. Its precision can be estimated by comparing the in situ temperature with that calculated by the reference equation of state. This is done in figure 4. It shows that at 3000 m the error is fairly constant and has a magnitude of about 10^{-3}°C. Comparisons at other depths give comparable results.

4.3 THE FINAL CALCULATION OF DENSITY

The final algorithm used to calculate the density from the in situ temperature, salinity and pressure is a vectorised version of Fofonoff and Millard (1983). Their algorithm was designed to achieve the required accuracy on the 32-bit computers in regular use by the sea-going community.

With hindsight, it would have been more efficient to have used the original 1980 equation of state polynomial on the 64-bit Cray, rather than the slightly longer version required for 32-bit machines.

5.0 COMPARISONS

Comparisons of the different algorithms have been carried out on Cray X-MP and SUN 4 computers, using 64-bit arithmetic. The SUN 4 computers use IEEE floating point arithmetic which differs slightly[2] from that used by the Cray X-MP. For this reason, check results from both schemes are included in the listings given in Appendices II and III.

The listings are for the Cray versions of the algorithms. The SUN Fortran versions are similar except that to get the same precision, all constants (e.g. 1.234E-56) should be written as double precision (1.234D-56) constants.

---

[2] Both systems use a binary fraction and a power of two exponent but the IEEE scheme is more precise because it uses a larger fraction. The Cray allocates 48 bits to the fraction, IEEE allocates 52 bits and gains an extra bit by always normalising the fraction so that the most significant bit is one. This is then discarded and only the lower order bits are stored.

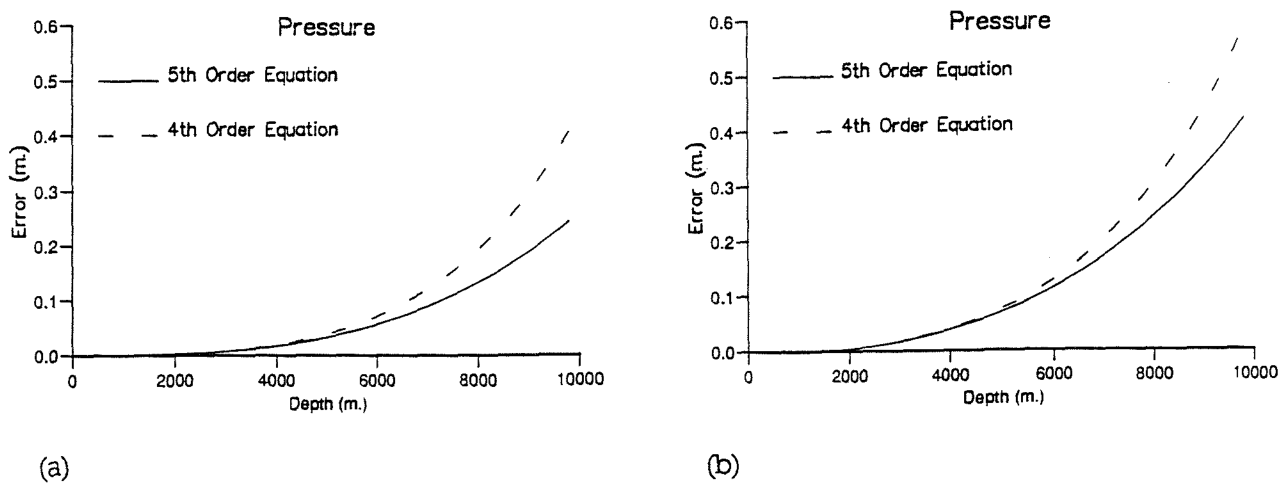


Figure 3. Errors of the fifth and fourth order depth to pressure algorithms for the depth range 0 to 10,000 m compared with (a) the Saunders and Fofonoff (1976) equation and (b) the Saunders (1981) equation

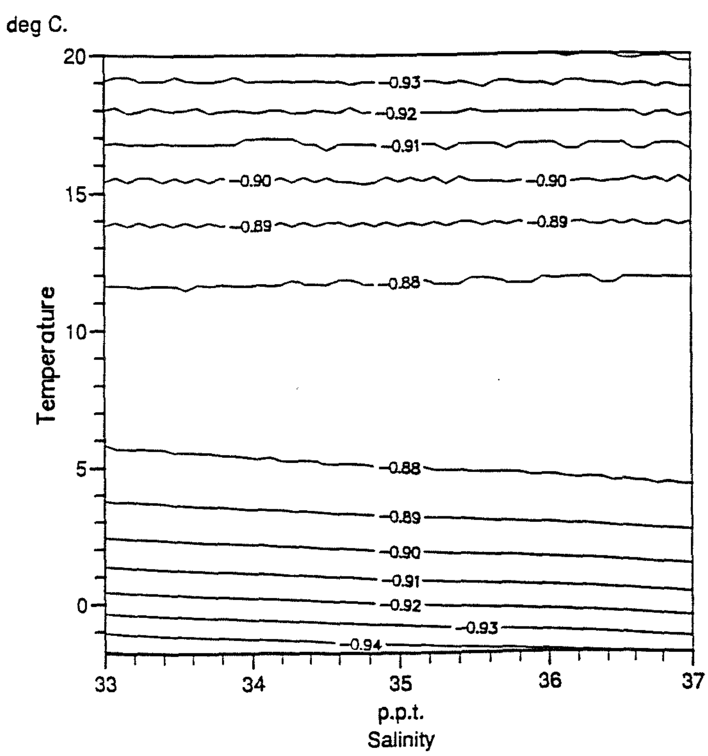


Figure 4. Errors in the FRAM potential temperature transformation algorithm, at a depth of 3000 m. The contour units are 10^{-3} degrees Celsius.

5.1 PRECISION

The performance of the Eckart equation of state at the surface and at 3000 m is compared in figures 1 and 2 with the results from the reference calculation. In the surface layer the errors in the slope of the temperature-salinity curve show up clearly. At temperatures below 0°C, the slope is out by a factor of two. This means that temperature gradients in the near surface layers have an unphysically large influence on stability.

At 3000 m the errors in the slope of the temperature-salinity curves are smaller, but there is a marked mean density offset. Such a constant offset does not affect either the stability or horizontal pressure gradient calculations, so for most modelling applications it can be neglected. (For this reason the Bryan and Cox density polynomial (Cox 1983) ignores the constant term completely.)

The FRAM modification of the UNESCO equation of state is compared with the reference calculation in figures 5 and 6. At the surface the agreement with the reference calculation should be almost perfect as no approximation is involved in calculating either pressure or temperature. However differences of order 10^{-6} occur because the FRAM scheme is formally only correct to single precision whereas the reference state is correct to double precision.

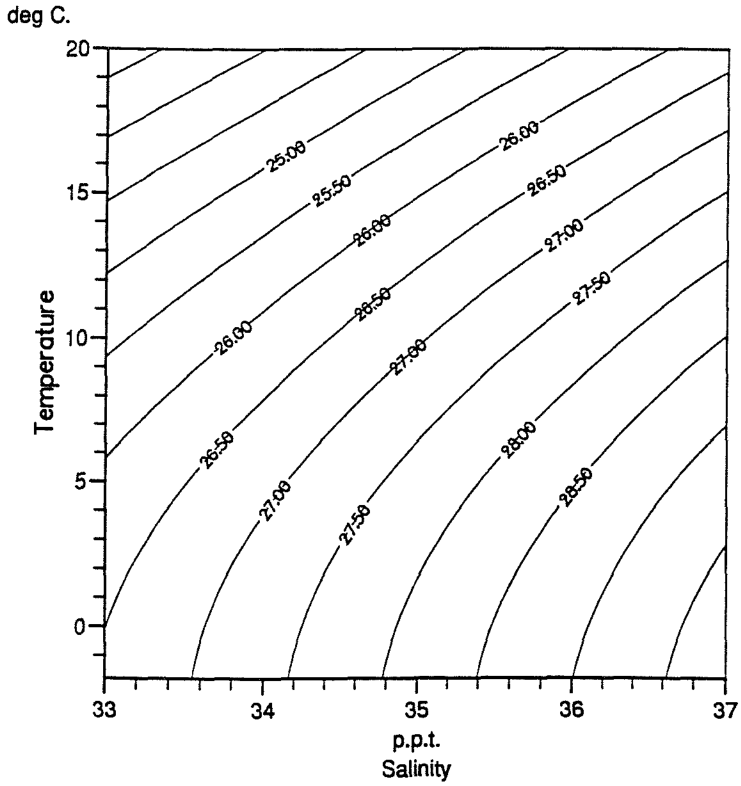
At 3000 m there are additional small errors. These are primarily the effect of the approximations made in the scheme for transforming potential temperature to temperature.

5.2 TIMING TESTS

Tests of the speed of the different equation of state algorithms have been carried out on a Cray X-MP computer and a SUN Sparcstation 1. The times required to carry out one set of density calculations for all the grid points in the FRAM model are shown in table 1. As well as the times taken by the Eckart and FRAM equations of state, the table also shows the times taken when the FRAM algorithm is modified to use either the Fofonoff (1977) algorithm for converting potential temperature to in situ temperature or the full UNESCO equation of state. Also included for comparison are tests with the Bryan and Cox (1972) polynomial equation. The Friedrich and Levitus (1972) polynomial equation of state should be slightly faster because fewer polynomial terms are used.

The tests show that the FRAM equation of state takes about 6 to 7 times longer than the Eckart equation. The Fofonoff temperature algorithm adds an overhead of about 33% to the FRAM timings. The use of the full UNESCO equation of state produces a saving of about 10%. The Bryan and Cox (1972) polynomial is slightly faster than the Eckart formula on the SUN, but is 50% slower on the Cray.

(a)



(b)

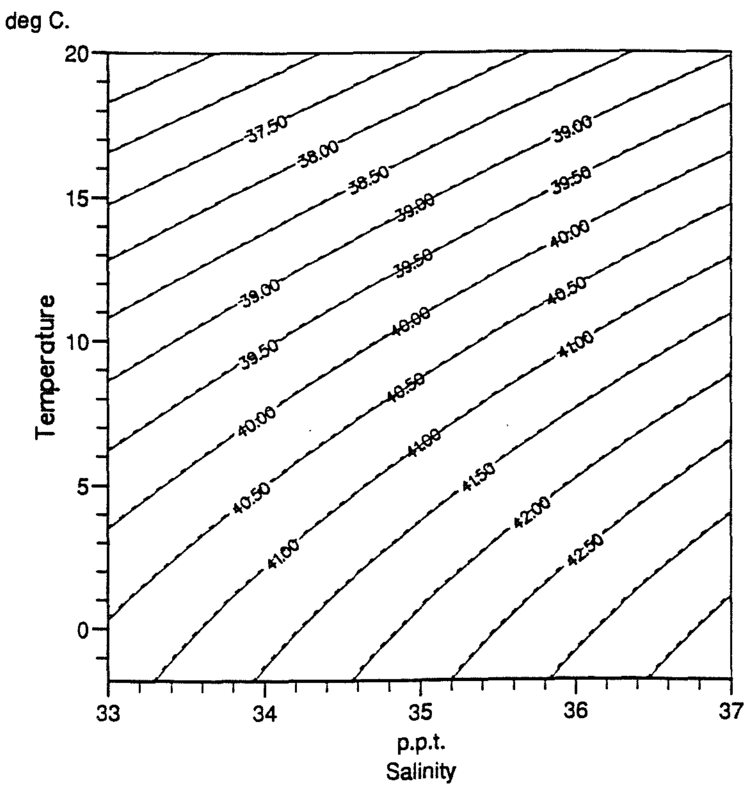
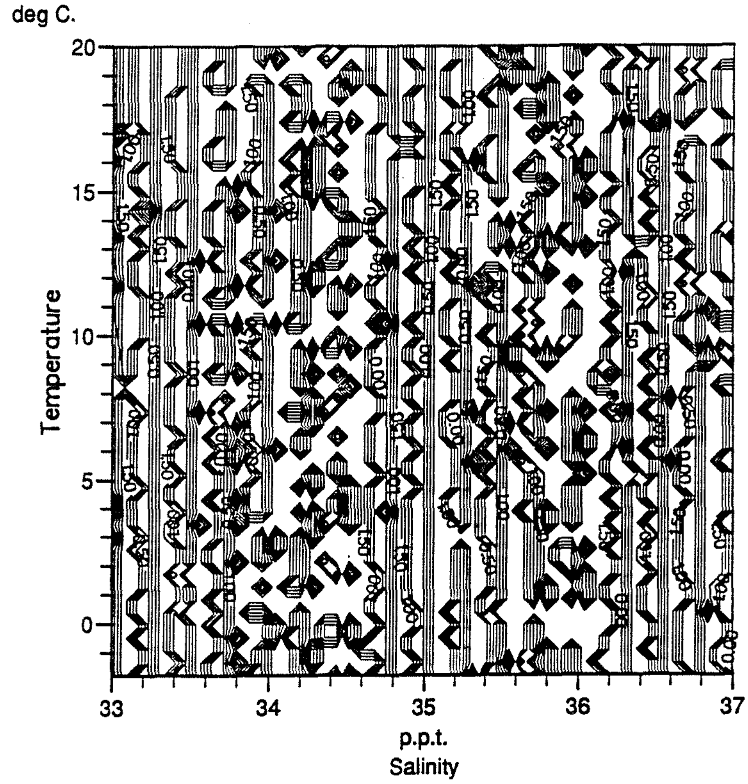


Figure 5. Density contours calculated using the FRAM modified EOS80 equation of state (solid lines) compared with the reference calculation (dotted lines) at depths of (a) 0 m and (b) 3000 m. The density contours are in sigma units (i.e. kg/m^3-1000.0).

(a)



(b)

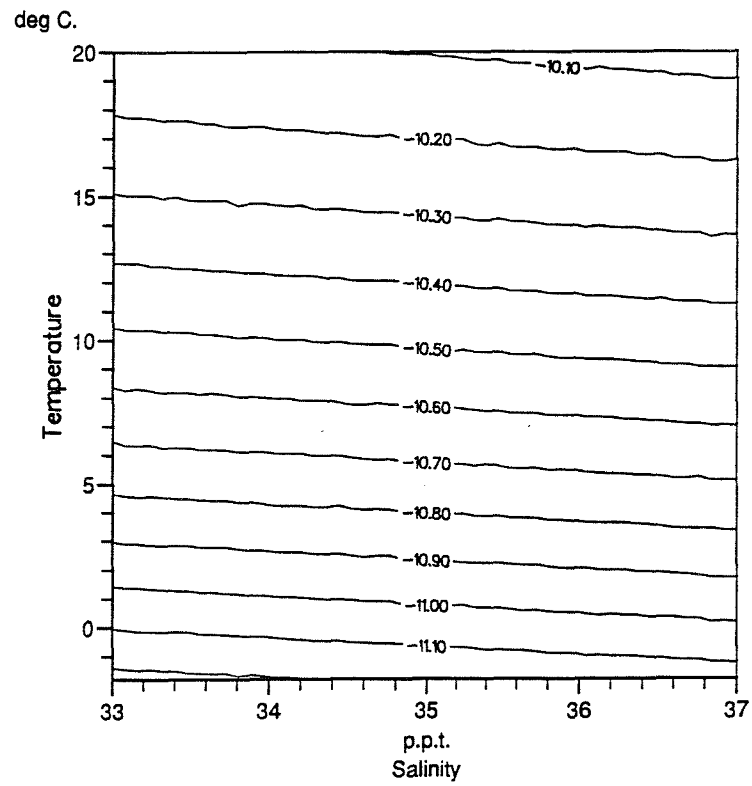


Figure 6. The difference between the FRAM modified EOS80 equation of state and the reference calculation at depths of (a) 0 m and (b) 3000 m. In figure 4a the contour units are 10^{-6} sigma units. In figure 4b they are 10^{-3} sigma units.

6.0 CONCLUSIONS

The comparisons have shown that the Eckart equation of state is not accurate enough for use in modelling polar oceans. It is recommended that in future it should not be used at all for ocean modelling. The FRAM scheme should be suitable for all oceans but it has an important drawback in that it is computationally expensive. On 64-bit machines, the computational cost can be reduced slightly by using the un-modified EOS80 formula. On 32-bit machines this would reduce the accuracy of the result.

For ocean models like FRAM, for which density is required at fixed depths within the ocean, the Bryan and Cox (1972), Friedrich and Levitus (1972) and Levitus and Isayev (1992) polynomial-based algorithms are recommended as being efficient alternatives. However, they should be initialised using the EOS80 equation of state and accurate transformations of depth to pressure and potential to in situ temperature, as are used in the reference equation of state described in Appendix 1. The precision of this scheme will be discussed in a later note.

TABLE 1 TIMING TESTS

The figures give the times in seconds, to calculate the equation of state once at each of the grid points in the FRAM model (721 x 221 x 32 grid points). All calculations were carried out one slab at a time as in the FRAM model, using 64-bit arithmetic. The Cray X-MP tests were carried out using a single processor.

ALGORTIHM	CRAY X-MP	SUN SPARCSTATION 1
Eckart	0.61	132.7
FRAM	4.55	705.2
FRAM + Fofonoff[1]	6.79	950.0
FRAM + UNESCO[2]	4.25	
Bryan and Cox[3]	0.95	121.8

Notes:

1. The FRAM algorithm with the section transforming potential temperature to in situ temperature replaced by the Fofonoff 1977 algorithm (expanded in line).

2. The FRAM algorithm with the final density calculation replaced by the full EOS80 polynomial.

3. The Bryan and Cox (1972) polynomial.

REFERENCES

ABRAMOWITZ, M. and STEGUN I.A. 1965 Handbook of Mathematical Functions. Dover, New York; Dover, 1046pp.

ARAKAWA, A. 1966 Computational design for long-term numerical integration of the equations of fluid motion: two dimensional incom-pressible flow. Part 1. Journal of Computational Physics, 1, 119-143.

BRYAN, K. 1969 A numerical method for the study of the circulation of the world ocean. Journal of Computational Physics, 4, 347-376.

BRYAN, K. & COX M.D. 1972 An approximate equation of state for numerical models of ocean circulation. Journal of Physical Oceanography, 2, 510-514.

BRYDEN, H. L. 1973 New polynomials for thermal expansion, adiabatic temperature gradient and potential temperature of sea water. Deep-Sea Research, 20, 401-408.

COX, M.D. 1984 A primitive equation three-dimensional model of the ocean. GFDL/NOAA, Princeton University; Ocean Group Technical Report No.1, 250pp.

ECKART, C. 1958 Properties of sea water, Part II. American Journal of Science, 256, 225-240.

FRIEDRICH, H. & LEVITUS S. 1972 An approximation to the equation of state for sea water suitable for numerical ocean models. Journal of Physical Oceanography, 2, 514-517.

FOFONOFF, N.P. 1962 The physical properties of sea water. The Sea, Vol 1. Interscience, New York. 864pp.

FOFONOFF, N.P. 1977 Computation of potential temperature of seawater for an arbitrary reference pressure. Deep-Sea Research, 24, 489-491.

FOFONOFF N.P. & MILLARD R.C. Jnr. 1983 Algorithms for computation of fundamental properties of seawater. UNESCO Technical Papers in Marine Science, 44, 53pp.

GILL, A.E. 1982 Atmosphere-Ocean dynamics. Academic Press, New York, 662pp.

KILLWORTH, P. 1989 On the parameterisation of deep convection in ocean models. In: 'Aha Huliko', a Winter Workshop on Parameterisation of small-scale processes. Ed. P.Muller, University of Hawaii Press, p 59-74.

KNUDSTON, M. 1901 Hydrographical Tables.
G.E.C. Gad, Copenhagen; Williams and Norgate, London.

LEVITUS, S. 1982 Climatological Atlas of the World Ocean.
Geophysical Fluid Dynamics Laboratory, Princeton University; NOAA Prof. Paper, 13.

LEVITUS, S. & ISAYEV, G. 1992 Polynomial Approximation to the International Equation of State for Seawater.
Journal of Atmospheric and Oceanic Technology, 9(5), 705-708.

MESINGER, F. & ARAKAWA A. 1976 Numerical methods used in atmospheric models.
World Meteorological Organisation; GARP Publication series, No. 17, 64pp.

SAUNDERS, P.M. & FOFONOFF N.P. 1976 Conversion of pressure to depth in the ocean.
Deep Sea Research, 23, 109-111.

SAUNDERS, P.M. 1981 Practical Conversion of Pressure to Depth.
Journal of Physical Oceanography, 11 (4), 573-574.

SAUNDERS, P.M. 1990 The international temperature scale of 1990.
WOCE Newsletter, 10, p10.

SEMTNER, A.J. 1974 An oceanic general circulation model with bottom topography.
Department of Meteorology, University of California, Los Angeles; Technical Report No. 9, 99pp.

TAKANO, K. 1974 A general circulation model for the world ocean.
Department of Meteorology, University of California, Los Angeles; Numerical Simulation of Weather and Climate. Technical Report No. 8, 47pp.

TUMLIRZ, O. 1909 Die Zustandsgleichung der Flussigkeiten bei hohem Druke.
Wein Akad. Wissensch., Sitzungsberichte, Math.-Naturwiss. Kl., 118A, 203.

UNESCO 1981a Tenth report of the joint panel on oceanographic tables and standards.
UNESCO Technical Papers in Marine Science, 36, 25pp.

UNESCO 1981b Background papers and supporting data on the International Equation of State of seawater 1980.
UNESCO Technical Papers in Marine Science, 38, 192pp.

THE FRAM GROUP (Webb, D.J. et al) 1991 An Eddy-Resolving Model of the Southern Ocean.
EOS Transactions, American Geophysical Union, 72(15), 169-174.

APPENDIX 1 : THE REFERENCE EQUATION OF STATE

This is based on the full international equation of state for seawater (EOS80) (UNESCO 1981a,b, Gill 1982), the Saunders (1981) scheme for converting pressure to depth and the Bryden (1973) equation for the adiabatic lapse rate. These are the equations used in the best experimental work. The EOS80 equation of state has not been used much by ocean modellers because of its computational cost.

Good experimental practice

Oceanographic instruments, such as the CTD, measure in situ temperature, salinity and pressure and these are the variables used in the international equation of state (EOS80). The temperature scale has recently been redefined (Saunders 1990), and for compatibility with earlier observations, if temperature measurements have been made using the later scale they should be transformed for analysis to the pre-1990 scale (International Practical Temperature Scale, IPTS-68).

Experimentalists usually calculate density using the Fofonoff and Millard (1983) version of the international equation of state for seawater algorithm (EOS80). The Fofonoff and Millard algorithm is designed to calculate density and specific volume anomaly with high accuracy on the 32-bit (single precision) computers used at sea. With computers of other precision, for example on 64-bit computers like the Cray, the algorithm needs to be modified or the unmodified equation of state used. This is because some of the constants derived by Fofonoff from the EOS80 algorithm are only correct for 32-bit word computers.

Pressure measurements made at sea are converted to depth using the algorithm of Saunders (1981) or the Fofonoff and Millard (1983) update of the Saunders and Fofonoff (1976) equation. The Saunders (1981) version is based on an analytic integration of the EOS80 and so is formally the most accurate. For this reason, it was chosen for the reference algorithm but the differences between the two schemes at depths of less than 6000 m are less that 0.2 m. The effect of the dynamic height anomaly on pressure is normally neglected. It generally increases with pressure, having values which range regionally from -0.5 to 2.5 m at 1000 db and from 0 m to 4 m at 5000 db (Saunders 1981).

Temperature measured at sea is converted to potential temperature by integrating the Bryden (1973) equation for the adiabatic lapse rate, using a fourth order Runge-Kutta algorithm (Fofonoff 1976).

The reference algorithm

The reference algorithm was designed to be as precise as possible so that it could be used to check the precision of the Eckart and FRAM equations of state. Like the latter algorithms, it calculates density as a function of temperature, salinity and pressure, but it uses more accurate algorithms and so is computationally more expensive. The Fortran code for running on a Cray is given in Appendix III.

The conversion of depth to pressure is carried out by inverting the Saunders (1981) algorithm using an iterative scheme. Usually it takes about 15 iterations to converge, either converging absolutely or to a small limit cycle with values accurate to the last few bits in the machine word.

The conversion of potential temperature to temperature is performed by integrating the Bryden (1973) equation using a leapfrog scheme with a step size of 1 db. Tests with different step sizes show that any errors introduced by the algorithm, as opposed to Bryden's formula, should be less than 10^{-8} degrees Celsius. This scheme is much more accurate than using a fourth order Runge-Kutta algorithm but the computational cost is much larger.

The reference density is then calculated from the in situ temperature, salinity and pressure using the unmodified EOS80 formula (UNESCO 1981a,b).

APPENDIX II.

FORTRAN CODE LISTINGS FOR THE ECKART AND FRAM EQUATIONS OF STATE

This appendix contains the listings of the two equations of state used by the FRAM model. The Cox (1984) model code requires the equation of state routine to have two entry points. The first calculates the density at all points in an east-west slab at a fixed latitude. The second is used for vertical stability calculations. Only the code corresponding to the first of the two entry points is given.

A.1 The Eckart equation of state

```
      SUBROUTINE STATE(TX,SX,RHO,TQ,SQ)
      IMPLICIT REAL*8(A-H,O-Z)
C
C===========================================================================
C
C  STATE computes one row of normalised densities using the
C  Eckart equation of state.
C
C  Input and output values are in 'model units':
C
C     TX(IMT,KM) is in-situ temperature in degrees C. (IPTS-68)
C     SX(IMT,KM) is salinity in parts per part minus 0.035
C     RHO(IMT,KM) is density in grams/cubic centimetre minus 1.02
C     Variables TQ and SQ are required parameters for the standard
C     Cox model STATE routine but are unused in this version.
C
C===========================================================================
*CALL PARAM
*CALL SCALAR
C===========================================================================
C
C  These two lines insert the standard parameter definitions
C  and the scalar common blocks.  The routine uses the following
C  parameters and common block variables:
C     IMT is the number of grid points in an east-west direction.
C     KM  is the number of levels.
C     ZDZZ(KM) is an array containing the depth in centimetres at the
C         centre of each model level.
C
C===========================================================================
```

```
C
C  This is the Cray version of the routine.  The SUN (or IEEE floating
C  point arithmetic) version is obtained by converting all single
C  precision constants to double precision.
C
C  CHECK VALUE: RHO = 4.0111867176869E-2       - CRAY 64-bit
C                   = 4.0111876126875D-02      - IEEE 64-bit.
C  FOR: TX=40.0, SX=0.005, ZDZZ=1000000.0.
C  These values of RHO correspond to 60.1118   - Sigma units.
C
C==========================================================================
C
      DIMENSION TX(IMT,KM),SX(IMT,KM)
      DO 100 K=1,KM
      FACTOR = 5891.0E0 + ZDZZ(K)/1013.0E0
      DO 100 I=1,IMT
      TEMP = FACTOR + 3000.0E0*(SX(I,K) + 0.035E0)
     &            +(38.0E0 - 0.375E0*TX(I,K)) * TX(I,K)
      TEMP = (1779.5E0 + (11.25E0 -0.0745E0*TX(I,K)) * TX(I,K)
     1       -(3800.0E0 + 10.0E0 * TX(I,K))*(SX(I,K) + 0.035E0))/TEMP
 100  RHO(I,K) = 1.0E0/(0.698E0 + TEMP) - 1.02E0
      RETURN
      END
```

A.2 The FRAM equation of state

```
      SUBROUTINE STATE(TX,SX,RHO,TQ,SQ,SINLAT)
      IMPLICIT REAL*8(A-H,O-Z)
C
C  STATE computes normalised density using the full UNESCO
C  equation of state and new schemes for the conversion of depth
C  to pressure and for converting potential temperature to in situ
C  temperature.
C
C  Input and output values are in 'model units':
C
C     TX(IMT,KM) is potential temperature in degrees C. (IPTS-68)
C     SX(IMT,KM) is salinity in parts per part minus 0.035.
C     RHO(IMT,KM) is density in grams/cubic centimetre minus 1.02.
C     Variables TQ and SQ are required parameters for the standard
C     Cox model STATE routine but are unused in this version.
C     SINLAT is the sine of the latitude. (Note routines STATE and
C         CLINIC need to be modified to set this parameter and to
C         include it in their calls to STATE).
C
C=======================================================================
*CALL PARAM
*CALL SCALAR
C=======================================================================
C
C  These two lines insert standard parameter definitions and
C  the scalar common blocks.  The routine uses the following
C  parameters and common block variables:
C     IMT is the number of grid points in an east-west direction.
C     KM  is the number of levels.
C     ZDZZ(KM) is an array containing the depth in centimetres at
C         the centre of each model level.
C
C=======================================================================
C
C  REFERENCES:
C      Millero et al 1980, Deep Sea Res.,27A,255-264
C      Jpots Ninth Report 1978,UNESCO Tech.Pap.Mar.Sci.No 30.
c      Jpots Tenth Report 1980,UNESCO Tech.Pap.Mar.Sci.No 36.
C      Bryden H,1973,Deep Sea Res.,20,401-408
C      Fofonoff N,1977,Deep Sea Res.,24,489-491
C      Saunders P and Fofonoff N,1976,Deep Sea Res.,23,109-111
C      Fofonoff N and Millard R,1983,UNESCO Tech.Pap.Mar.Sci.No 44.
C  Formulae given by Abramowitz and Stegan are used to convert
C  Saunders and Fonanoff's equation for convertine pressure to
C  depth into one to convert depth to pressure.
C
```

```
C  Potential temperature is calculated from Bryden's equation for
C  the gradient with pressure using an analytic method which
C  assumes that the temperature change is small.  This is instead
C  of the computationally more expensive Runge-Kutta scheme used
C  by Fofonoff 1977.
C
C =========================================================================
C
C  This is the Cray version of the routine.  The SUN (or IEEE floating
C  point arithmetic) version is obtained by converting all single
C  precision constants to double precision.
C
C  CHECK VALUE: RHO = 3.9485438761937E-2       - CRAY 64-bit
C                   = 3.9485438761930D-02      - IEEE 64-bit.
C  FOR: TX=40.0, SX=0.005, ZDZZ=1000000.0, SINLAT=SIN(30.0).
C
C  These values of RHO correspond to 59.485    - in sigma units.
C
C =========================================================================
C
      DIMENSION TX(IMT,KM),SX(IMT,KM),RHO(IMT,KM)
      SAVE IN,B1,B2,B3,B4,B5,THIRD,V350P
      DATA R3500,R4/1028.1063E0,4.8314E-4/
      DATA DR350/28.106331E0/
      DATA IN/0/
C
C  1.  Calculate constants for use later.
C
      IF(IN.EQ.0)THEN
C  Constants for presure polynomial
        A1 = 97.2659E0
        A2 = -2.2512E-3
        A3 =  2.279E-7
        A4 = -1.82E-11
        B1 =  1.0E0/A1
        B2 = -A2/A1**3
        B3 = (2.0E0*A2**2 - A1*A3)/A1**5
        B4 = (5.0E0*A1*A2*A3 - A1**2*A4 - 5E0*A2**3)/A1**7
        B5 = (6.0E0*A1**2*A2*A4 + 3.0E0*A1**2*A3**2 + 14.0E0*A2**4
     &       -21.0E0*A1*A2**2*A3)/A1**9
C  Constants for temperature calculation
        THIRD = 1.0E0/3.0E0
C  Constants for density calculation
        V350P = 1.0E0/R3500
        IN = 1
      ENDIF
C
C  2.  Variables calculated only once each entry
```

```
C
      X = SINLAT**2
      GR1 = 9.780318E0*(1.0E0 + (5.2788E-3 + 2.36E-5*X)*X)
C
C  3.  Calculations made once at each depth
C
      DO 100 K=1,KM
C  Convert model units (cm) to metres
      Z   = ZDZZ(K)*0.01E0
      Z1  = Z*(GR1 + (1.092E-5 * GR1/97.2659E0)*Z)
      P   = ((((B5*Z1 + B4)*Z1 + B3)*Z1 + B2)*Z1 + B1)*Z1
      RK35 = (5.03217E-5*P + 3.359406E0)*P + 21582.27E0
      GAM = P/RK35
      PK = 1.0E0-GAM
      V350Q = V350P*PK
      DR35P = GAM/V350Q
      DEN1 = DR350+DR35P
      RK35I = 1.0E0/RK35
      V350QI = 1.0E0/V350Q
C
C  4.  Calculations made at each grid point
C
      DO 100 I=1,IMT
      T   = TX(I,K)
C  Convert model units to NSU
      S   = SX(I,K)*1000.0E0 + 35.0E0
      SR = SQRT(S)
C
C     Calculate absolute temperature TA at pressure P.
C
      DS = S-35.0E0
C  First estimate temperature at mid depth.
      ATGR83 = (-4.2393E-7*T + 1.8932E-5)*DS
     &      +((6.6228E-9*T - 6.836E-7)*T + 8.5258E-5)*T + 3.5803E-4
      T1 = T + 0.5E0*P*ATGR83
C  Integrate adiabatic gradient polynomial analytically.
      R1 = (-2.1687E-13*T1 + 1.8676E-11)*T1 - 4.6206E-10
      R2 = ( 2.7759E-10*T1 - 1.1351E-8)*DS
     &   +((-5.4481E-12*T1 + 8.733E-10)*T1 - 6.7795E-8)*T1 + 1.8741E-6
      R3 = (-4.2393E-7*T1 + 1.8932E-5)*DS
     &   +(( 6.6228E-9*T1 - 6.836E-7)*T1 + 8.5258E-5)*T1 + 3.5803E-4
      R1 = R1*THIRD
      R2 = R2*0.5E0
      TA =  T + ((R1*P + R2)*P + R3)*P
C
C 3.  Calculate density
C
C
```

```
C  Calculate specific volume anomaly at the surface
C
      R1 = ((((6.536332E-9 *TA - 1.120083E-6)*TA + 1.001685E-4)*TA
     &        -9.095290E-3)*TA + 6.793952E-2)*TA - 28.263737E0
      R2 = (((5.3875E-9 *TA - 8.2467E-7)*TA + 7.6438E-5)*TA
     &      - 4.0899E-3)*TA + 8.24493E-1
      R3 = (-1.6546E-6*TA + 1.0227E-4)*TA - 5.72466E-3
      SIG = (R4*S + R3*SR + R2)*S + R1
      SVA = 1.0E0/(R3500 + SIG) - V350P
C
C  Now correct sva for pressure and convert to density
C
      R1 = (9.1697E-10*TA + 2.0816E-8)*TA - 9.9348E-7
      R2 = (5.2787E-8*TA - 6.12293E-6)*TA + 3.47718E-5
      A  = R1*S + R2
      R1 = 1.91075E-4
      R2 = (-1.6078E-6*TA - 1.0981E-5)*TA + 2.2838E-3
      R3 = ((-5.77905E-7*TA + 1.16092E-4)*TA + 1.43713E-3)*TA
     &      - 0.1194975E0
      B  = (R1*SR + R2)*S + R3
      R1 = (-5.3009E-4*TA + 1.6483E-2)*TA + 7.944E-2
      R2 = ((-6.1670E-5*TA + 1.09987E-2)*TA - 0.603459E0)*TA + 54.6746E0
      R3 = (((-5.155288E-5*TA + 1.360477E-2)*TA - 2.327105E0)*TA
     &     + 148.4206E0)*TA - 1930.06E0
      C  = (R1*SR + R2)*S + R3
      DK = (A*P + B)*P + C
      SVA  = SVA*PK + (V350P + SVA)*P*(RK35I - 1.0E0/(RK35 + DK))
      DVAN = V350QI - 1E0/(V350Q + SVA)
C
C  Convert density to model units i.e. gm/cc-1.02
C
      RHO(I,K) = (DEN1 - DVAN)*0.001E0 - 0.02E0
  100 CONTINUE
      RETURN
      END
```

APPENDIX III :

FORTRAN CODE FOR THE REFERENCE EQUATION OF STATE

```
      SUBROUTINE REFDEN(T,S,Z,RHO,XLAT)
      IMPLICIT REAL*8(A-H,O-Z)
C
C  SUBROUTINE TO CALCULATE THE DENSITY AT TEMPERATURE T, SALINITY S
C  AND DEPTH Z, AS ACCURATELY AS POSSIBLE.  IT USES AN INVERSE OF
C  THE SAUNDERS (1981) ROUTINE TO TRANSFORM FROM DEPTH TO PRESSURE,
C  AN ACCURATE METHOD OF CALCULATING THE IN SITU TEMPERATURE AND
C  THE UNESCO 80 EQUATION OF STATE.
C
C     INPUT UNITS:
C       IN-SITU TEMPERATURE (T): DEGREES C. (IPTS-68)
C       SALINITY (S): PRACTICAL SALINITY UNITS
C       DEPTH    (Z): METRES
C       LATITUDE (XLAT): DEGREES
C     OUTPUT UNITS:
C       DENSITY(RHO): KILOGRAMS PER CUBIC METER
C
C======================================================================
C
C  This and the following routines are the Cray versions of the
C  routines.  The SUN (or IEEE floating point arithmetic) versions
C  are obtained by converting all single precision constants to double
C  precision.
C
C  CHECK VALUE: RHO = 1059.355556531       - CRAY 64-bit
C                   = 1059.3555565304      - IEEE 64-bit.
C  FOR: T=40.0, S=40.0, Z=10000.0, XLAT=30.0.
C
C======================================================================
C
      P = FNPZ(Z,XLAT)
      T1 = POTTEM(T,S,0.0E0,P,1.0E0)
      CALL UNESCO(T1,S,P,RHO)
      RETURN
      END

      FUNCTION FNPZ(Z,XLAT)
      IMPLICIT REAL*8(A-H,O-Z)
      PARAMETER (MLOOP=30,MCONV=5,EPS=1E-6)
C
C  FUNCTION TO CALCULATE PRESSURE IN DECIBARS FROM DEPTH IN METRES
```

```
C  USING AN ITERATIVE INVERSE OF SAUNDERS ALGORITHM (FUNCTION
C  FNPZ).  ITERATES UNTIL THE ERROR IS ZERO, A LIMIT CYCLE IS
C  DETECTED OR 'MLOOP' ITERATIONS REACHED.
C  ERROR EXIT IF ERROR > EPS.
C  ARRAY PA USED TO DETECT A LIMIT CYCLE.
C
C  CHECK VALUE FNPZ = 10302.423165          - CRAY 64-bit
C                   = 10302.4231650052      - IEEE 64-bit.
C  FOR: Z = 10000m, XLAT = 30.0
C
      DIMENSION PA(MCONV)
C
      P  = Z
      IA = 0
      DO 20 I=1,30
      ZZ = FNZP(P,XLAT)
C ZERO ERROR
      IF(Z.EQ.ZZ)GOTO 50
      EE = Z - ZZ
      EA = ABS(EE)
C SAVE NEW BEST VALUE
      IF(IA.EQ.0.OR.EA.LT.EP)THEN
        IA = 1
        EP = EA
        PA(IA) = P
C LOOK FOR LIMIT CYCLE
      ELSEIF(EA.EQ.EP)THEN
        DO 40 J=1,IA
        IF(P.EQ.PA(J))GOTO 50
   40   CONTINUE
        IF(IA.LT.MCONV)THEN
          IA = IA + 1
          PA(IA) = P
        ENDIF
      ENDIF
C CORRECT P AND LOOP
      P  = P + EE
   20 CONTINUE
C
      IF(EA.GT.EPS)THEN
      PRINT *,'Subroutine Z2PB.  Iteration has not converged after',
     &         ' 30 iterations'
      PRINT *,'Object depth =',Z
      PRINT *,'Latest P = ',P,'.  Corresponding Z = ',ZZ
      PRINT *,'Minimum error = ',EA
      PRINT *,'Number of corresponding Ps =',IA
      PRINT *,'PA array',(PA(K),K=1,IA)
      STOP
```

```
      ENDIF
C
      P = PA(IA)
   50 FNPZ = P
      RETURN
      END

      FUNCTION FNZP(PIN,XLAT)
      IMPLICIT REAL*8(A-H,O-Z)
C
C  FUNCTION TO TRANSFORM PRESSURE TO DEPTH USING THE METHOD OF
C  P.M.SAUNDERS, 1981.  JOURNAL OF PHYSICAL OCEANOGRAPHY,
C  11, 573-574.
C
C  INPUT:  PIN = PRESSURE IN DECIBARS ("oceanographic" pressure
C                equals absolute pressure minus one atmosphere).
C          XLAT= LATITUDE IN DEGREES.
C
C  OUTPUT: FNZP = DEPTH IN METRES.
C
C  CHECK VALUE: FNZP = 9712.478325455          - CRAY 64-bit
C                    = 9712.4783254538,        - IEEE 64-bit.
C  FOR: PIN=10000.0, XLAT=30.0.
C
      DATA IN/0/
      SAVE IN
C
C 1.  CALCULATE CONSTANTS
C
      IF(IN.EQ.0)THEN
        IN = 1
        PI = 3.141592654E0
        RADIAN = PI/180E0
        G1 =  9.780318E0
        G2 =  9.780318E0*(5.3024E-3 - 5.9E-6*4.0E0)
        G3 = -9.780318E0*5.9E-6 * 4.0E0
C
C  AL = SPECIFIC VOLUME AT (T=0,S=35,P=0) TIMES 10**5
C  RK = CONSTANT COEFICIENT
C  RA = TERM PROPORTIONAL TO P
C  RB = TERM PROPORTIONAL TO P**2
C
        S = 35.0E0
        C1P5 = 1.5E0
        AL = 1E5/(9.99842594E2 + 8.24493E-1*S
     &          - 5.72466E-3*S**C1P5 + 4.8314E-4*S**2)
        RK = 1.965221E4 + 5.46746E1*S + 7.944E-2*S**C1P5
        RA = 3.239908E0 + 2.2838E-3*S + 1.91075E-4*S**C1P5
```

```
        RB = 8.50935E-5 - 9.9348E-7*S
        DD = SQRT(RA*RA - 4.0E0*RK*RB)
        C1 = 0.5E0/RB
        C2 = RA/RK
        C3 = RB/RK
        C4 = RA/(2.0E0*RB*DD)
        C5 = 2.0E0*RB/(RA - DD)
        C6 = 2.0E0*RB/(RA + DD)
        C7 = 0.5E0*2.226E-6
      ENDIF
C
C 2.  CALCULATE GRAVITY
C
      X  = SIN(RADIAN*XLAT)**2
      GS = (G3*X + G2)*X + G1
C  CONVERT FROM PRESSURE IN DECIBARS TO BARS
      P  = PIN*1.0E-1
C
C 3.  INTEGRATE SPECIFIC VOLUME
C
      R1 = AL*(P - C1*LOG((C3*P + C2)*P+1.0E0) + C4*LOG((1.0E0 + C5*P)
     &       /(1.0E0 + C6*P)))
      FNZP = R1/(GS + C7*PIN)
C
      RETURN
      END


      FUNCTION POTTEM(TT,SS,P0,P1,DPP)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C  SUBROUTINE TO CALCULATE THE FINAL TEMPERATURE OF WATER MOVED
C  ADIABATICALLY FROM AN INITIAL TEMPERATURE TT, SALINITY SS AND
C  PRESSURE P0, TO A FINAL PRESSURE P1.
C
C  THE INTEGRAL EQUATION IS SOLVED BY DIRECT INTEGRATION WITH A
C  PRESSURE INCREMENT DPP - USING THE BRYDEN EQUATION FOR THE
C  ADIABATIC LAPSE RATE (SUBROUITNE ATG).
C
C    T = INITIAL TEMPERATURE IN DEGREES C. (IPTS-68)
C    S = SALINITY IN NSU.
C    P0= INITIAL PRESSURE IN DECIBARS.
C    P1= FINAL PRESSURE IN DECIBARS.
C    DPP=PRESSURE STEP.
C    POTTEM = FINAL TEMPERATURE IN DEGREES CENTIGRADE.
C
C  PRESSURES ARE "OCEANOGRAPHIC" PRESSURES, EQUAL TO ABSOLUTE
C  PRESSURES MINUS ONE ATMOSPHERE.
```

```
C  TESTS WITH DPP VALUES RANGING FROM 1 TO 128 DECIBARS SHOWED
C  THE MOST ACCURATE RESULTS WERE OBTAINED WITH DPP EQUAL TO 1.
C
C  CHECK VALUE: POTTEM = 43.26663196648         - CRAY 64-bit
C                      = 43.266631967051,       - IEEE 64-bit.
C  FOR: T=40.0, S=40.0, P0=0.0, P1=10000.0, DPP=1.0.
C
      IF(P0.LT.0.0E0 .OR. P0.GT.20000.0E0
     &    .OR.P1.LT.0.0E0 .OR. P1.GT.20000.0E0)THEN
        PRINT *,' SUBROUTINE POTTEM STOPPING - PRESSURES OUT OF RANGE'
        PRINT *,' PRESSURES P0 AND P1 = ',P0,P1
        PRINT *,' ALLOWED RANGE HAS MIN OF 0.0, MAX OF 20,000'
        STOP
      ENDIF
C
      DP = SIGN(DPP,P1-P0)
      P  = P0
      T  = TT
      TB = T  - ATG(P0,T,SS)*DP
C
 10   TA = TB + 2.0E0*ATG(P,T,SS)*DP
      P  = P  + DP
      TB = T
      T  = TA
      TEST = (P - P1)*(P - DP - P1)
      IF(TEST.GT.0E0)GOTO 10
      POTTEM = ((P1 - P + DP)*T + (P - P1)*TB)/DP
      RETURN
      END

      FUNCTION ATG(P,T,S)
      IMPLICIT REAL*8(A-H,O-Z)
C
C  ADIABATIC TEMPERATURE GRADIENT DEG C PER DECIBAR
C  REF: BRYDEN,H.,1973,DEEP-SEA RES.,20,401-408
C  UNITS:
C        PRESSURE        P         DECIBARS
C        TEMPERATURE     T         DEGREES C. (IPTS-68)
C        SALINITY        S         (PSS-78)
C        ADIABATIC       ATG       DEGREES CELCIUS PER DECIBAR
C
C  PRESSURE IS "OCEANOGRAPHIC" PRESSURE EQUAL TO ABSOLUTE
C  PRESSURE MINUS ONE ATMOSPHERE.
C
C  CHECK VALUE: ATG = 3.2559758                        - CRAY 64-bit
C                   = 3.2559758000000D-04 DEG C/DBAR - IEEE 64-bit.
C  FOR: P=10000.0, T=40.0, S=40.0.
C
```

```
      DS = S-35E0
      ATG = (((-2.1687E-16*T + 1.8676E-14)*T - 4.6206E-13)*P
     &    + ((2.7759E-12*T - 1.1351E-10)*DS + ((-5.4481E-14*T
     &    + 8.733E-12)*T - 6.7795E-10)*T + 1.8741E-8))*P
     &    + (-4.2393E-8*T + 1.8932E-6)*DS
     &    + ((6.6228E-10*T - 6.836E-8)*T + 8.5258E-6)*T + 3.5803E-5
      RETURN
      END


      SUBROUTINE UNESCO (T, S, PIN, RHO)
      IMPLICIT REAL*8(A-H,O-Z)
C
C======================================================================
C     THIS SUBROUTINE CALCULATES THE DENSITY OF SEAWATER USING THE
C     STANDARD EQUATION OF STATE RECOMMENDED BY UNESCO(1981).
C
C     INPUT UNITS:
C       IN-SITU TEMPERATURE (T): DEGREES C. (IPTS-68)
C       SALINITY (S): PRACTICAL SALINITY UNITS
C       PRESSURE (PIN): DECIBARS
C     OUTPUT UNITS:
C       DENSITY(RHO): KILOGRAMS PER CUBIC METER
C
C  PRESSURE IS "OCEANOGRAPHIC" PRESSURE EQUAL TO ABSOLUTE
C  PRESSURE MINUS ONE ATMOSPHERE.
C
C     REFERENCES:
C        GILL, A., ATMOSPHERE-OCEAN DYNAMICS: INTERNATIONAL GEOPHYSICAL
C         SERIES NO. 30. ACADEMIC PRESS, LONDON, 1982, PP 599-600.
C        UNESCO, 10TH REPORT OF THE JOINT PANEL ON OCEANOGRAPHIC TABLES
C          AND STANDARDS. UNESCO TECH. PAPERS IN MARINE SCI. NO. 36,
C          PARIS, 1981.
C
C======================================================================
C
C  CHECK VALUE: RHO = 1059.82037676         - CRAY 64-bit
C                   = 1059.8203767598       - IEEE 64-bit.
C  FOR: T=40.0, S=40.0, PIN=10000.0.
C
      C1P5 = 1.5E0
C  CONVERT DEPTH (IN DECIBARS) TO BARS
      P  = PIN * 1.0E-1
      RW =     9.99842594E2 + 6.793952E-2*T - 9.095290E-3*T**2
     &       + 1.001685E-4*T**3 - 1.120083E-6*T**4 + 6.536332E-9*T**5
      RSTO = RW + (8.24493E-1 - 4.0899E-3*T + 7.6438E-5*T**2
     &          - 8.2467E-7*T**3 + 5.3875E-9*T**4) * S
     &        + (-5.72466E-3 + 1.0227E-4*T - 1.6546E-6*T**2) * S**C1P5
```

```
&            + 4.8314E-4 * S**2
 XKW    =    1.965221E4 + 1.484206E2*T - 2.327105E0*T**2
&          + 1.360477E-2*T**3 - 5.155288E-5*T**4
 XKSTO = XKW + (5.46746E1 - 6.03459E-1*T + 1.09987E-2*T**2
&          - 6.1670E-5*T**3) * S
&          + (7.944E-2 + 1.6483E-2*T - 5.3009E-4*T**2) * S**C1P5
 XKSTP = XKSTO + (3.239908E0 + 1.43713E-3*T + 1.16092E-4*T**2
&                    - 5.77905E-7*T**3) * P
&         + (2.2838E-3 - 1.0981E-5*T - 1.6078E-6*T**2) * P * S
&         + 1.91075E-4 * P * S**C1P5
&         + (8.50935E-5 - 6.12293E-6*T + 5.2787E-8*T**2) * P**2
&         + (-9.9348E-7 + 2.0816E-8*T + 9.1697E-10*T**2) * P**2 * S
 RHO =     RSTO / (1.0E0 - P/XKSTP)
C
 RETURN
 END
```