

Notes on the function,  
**gsw\_t\_from\_entropy(SA, entropy, p),**  
 which evaluates *in situ* temperature from specific entropy

This function, **gsw\_t\_from\_entropy**, finds *in situ* temperature  $t$  for given values of Absolute Salinity, specific entropy and pressure. This function is essentially simply the following two lines of code.

```
pt = gsw_pt_from_entropy(SA,entropy);
t = gsw_pt_from_t(SA,pt,0,p);
```

This first line finds the potential temperature  $\theta$  with respect to the reference pressure  $p_r = 0$  dbar from the given values of Absolute Salinity and specific entropy, using the GSW function **gsw\_pt\_from\_entropy**. The second line calculates the *in situ* temperature at (reference) pressure  $p$  for the seawater “parcel”  $(S_A, \theta, 0)$  using the GSW function **gsw\_pt\_from\_t**.

The Notes describing the function **gsw\_pt\_from\_entropy**(SA, entropy) follow below.

## Notes on the function, `gsw_pt_from_entropy`( $S_A$ , entropy), which evaluates potential temperature with reference pressure of 0 dbar from specific entropy

This function, `gsw_pt_from_entropy`, finds  $\theta = \theta(S_A, \eta)$ , the potential temperature with respect to the reference pressure  $p_r = 0$  dbar for given values of Absolute Salinity and specific entropy. A modified Newton-Raphson iterative solution technique is employed to find the zero of the function

$$f(\theta) \equiv \tilde{\eta}(S_A, \theta) - \eta = 0, \quad (1)$$

which equates the specific entropy of the “bottle”,  $\tilde{\eta}(S_A, \theta)$ , to the input specific entropy  $\eta$ . This function, `gsw_pt_from_entropy` is very similar in its operation to the function `gsw_pt0_from_t`, which calculates the potential temperature referenced to  $p_r = 0$  dbar from the inputs  $(S_A, t, p)$ . The main difference between these functions relates to the way the initial value of potential temperature,  $\theta_0$ , is evaluated. How this is done for the present function is now described.

The plot of  $(\theta - \Theta)$  on the  $S_A - \Theta$  diagram shown as Figure A.17.1 of IOC *et al.* (2010) (this figure is reproduced below) suggests the following approximation

$$(\theta - \Theta) \approx -0.05(1 - S_A/S_{SO})\Theta, \quad (2)$$

from which we find that  $\tilde{\Theta}_\theta$  can be written as

$$\tilde{\Theta}_\theta \approx [1 - 0.05(1 - S_A/S_{SO})]^{-1}. \quad (3)$$

The derivative of  $f(\theta)$  with respect to potential temperature is (from Eqn. (1) and from Eqn. (P.14a) of IOC *et al.* (2010))

$$f'(\theta) = \tilde{\eta}_\theta = \frac{c_p^0}{(T_0 + \theta)} \tilde{\Theta}_\theta. \quad (4)$$

Combining Eqns. (3) and (4) we find the following approximate expression for the derivative  $f'(\theta)$ ,

$$f'(\theta) = \tilde{\eta}_\theta \approx \frac{c_p^0}{(T_0 + \theta)} [1 - 0.05(1 - S_A/S_{SO})]^{-1}. \quad (5)$$

Integrating this expression with respect to  $\theta$  we find the following approximate expression,

$$\tilde{\eta}(S_A, \theta) \approx c_p^0 \ln \left( 1 + \frac{\theta}{T_0} \right) \left[ 1 - 0.05 \left( 1 - \frac{S_A}{S_{SO}} \right) \right]^{-1} + \frac{c_p^0}{T_0} \left( 1 - \frac{S_A}{S_{SO}} \right) \left[ 1 - 1.01 \left( 1 - \frac{S_A}{S_{SO}} \right) \right]. \quad (6)$$

The second part of this expression is a function of only  $S_A$  and has been found as a simple fit to entropy at zero potential temperature, that is, a simple fit to  $\tilde{\eta}(S_A, \theta = 0^\circ\text{C})$ .

The initial value of potential temperature  $\theta_0$  is found by equating the right-hand side of Eqn. (6) to the input value of entropy,  $\eta$ , and solving this equation for  $\theta$ . This initial value of potential temperature,  $\theta_0$ , is then used in Eqn. (5) to find the initial estimate of the derivative  $f'(\theta_0)$ .

### The modified Newton-Raphson iteration technique

The normal Newton-Raphson technique converges iteratively towards a root of the function  $f(\theta) \equiv \tilde{\eta}(S_A, \theta) - \eta = 0$  with each successive iteration being found from the previous one according to

$$\theta_{n+1} = \theta_n - \frac{f(\theta_n)}{f'(\theta_n)}. \quad (7)$$

Notice that in Eqn. (7) the function value and its derivative are evaluated at the same value of  $\theta$ . For the thermodynamic cases we consider, the derivative  $f'(\theta)$  is a slowly varying function of  $\theta$ , so that we adopt the numerical technique of McDougall *et al.* (2003) (see pages 731-732 therein) and evaluate the function and its derivative at different values of  $\theta$ , thereby improving convergence.

Starting from  $\theta_0$  and the crude estimate Eqn. (5) of  $f'$ , an intermediate value of potential temperature,  $\theta_1$ , is found as  $\theta_1 = \theta_0 - f(\theta_0)/f'$ . The reason for calculating this intermediate value is so that the derivative can be evaluated at a potential temperature which is close to the mid point between the initial value and the final solution. The derivative is evaluated at  $0.5[\theta_0 + \theta_1]$  and the next estimate of potential temperature,  $\theta_2$ , is found from

$$\theta_2 = \theta_0 - \frac{f(\theta_0)}{f'(0.5[\theta_0 + \theta_1])}. \quad (8)$$

This marks the end of the first iteration of the modified Newton-Raphson method. At this stage we have performed just one evaluation of both  $f$  and  $f'$ ; the same number of such evaluations as are involved in one full iteration of the standard Newton-Raphson procedure. The next two-step modified Newton-Raphson iteration proceeds as follows

$$\theta_3 = \theta_2 - \frac{f(\theta_2)}{f'(0.5[\theta_0 + \theta_1])}, \quad \text{then} \quad \theta_4 = \theta_2 - \frac{f(\theta_2)}{f'(0.5[\theta_2 + \theta_3])}. \quad (9a,b)$$

In this whole process leading to the value  $\theta_4$ , the function  $f$  is evaluated just twice (at  $\theta_0$  and  $\theta_2$ ) and its derivative  $f'$  is also evaluated just twice (at  $0.5[\theta_0 + \theta_1]$  and  $0.5[\theta_2 + \theta_3]$ ).

In the application of this modified Newton-Raphson procedure to the function **gsw\_pt\_from\_entropy** we find that the solution converges to machine precision after two iterations, and so the code returns the value  $\theta_4$  of Eqn. (9b) above. Hence the function **gsw\_pt\_from\_entropy** evaluates  $f$  twice (at  $\theta_0$  and  $\theta_2$ ) and its derivative  $f'$  twice (at  $0.5[\theta_0 + \theta_1]$  and  $0.5[\theta_2 + \theta_3]$ ).

The derivative  $f'(\theta)$  is given by Eqn. (4) above and is evaluated as the second derivative of the Gibbs function,

$$f'(\theta) = \tilde{\eta}_\theta = -g_{TT}(S_A, \theta, 0), \quad (10)$$

this being called twice, once at  $0.5[\theta_0 + \theta_1]$  and once at  $0.5[\theta_2 + \theta_3]$ . Because this second derivative of the Gibbs function is called at zero pressure, a special library function **gsw\_gibbs\_pt0\_pt0** has been written to gain computational efficiency by explicitly recognizing that the pressure is zero.

## References

- Jackett, D. R., T. J. McDougall, R. Feistel, D. G. Wright and S. M. Griffies, 2006: Algorithms for density, potential temperature, conservative temperature and freezing temperature of seawater. *Journal of Atmospheric and Oceanic Technology*, **23**, 1709-1728.
- McDougall, T. J., D. R. Jackett, D. G. Wright and R. Feistel, 2003: Accurate and computationally efficient algorithms for potential temperature and density of seawater. *Journal of Atmospheric and Oceanic Technology*, **20**, 730-741.
- IOC, SCOR and IAPSO, 2010: *The international thermodynamic equation of seawater – 2010: Calculation and use of thermodynamic properties*. Intergovernmental Oceanographic Commission, Manuals and Guides No. 56, UNESCO (English), 196 pp. Available from <http://www.TEOS-10.org>

Here follows appendix A.10 of the TEOS-10 Manual (IOC *et al.* (2010)).

### A.10 Proof that $\theta = \theta(S_A, \eta)$ and $\Theta = \Theta(S_A, \theta)$

Consider changes occurring at the sea surface, (specifically at  $p = 0$  dbar) where the temperature is the same as the potential temperature referenced to 0 dbar and the increment of pressure  $dp$  is zero. Regarding specific enthalpy  $h$  and chemical potential  $\mu$  to be functions of entropy  $\eta$  (in place of temperature  $t$ ), that is, considering the functional form of  $h$  and  $\mu$  to be  $h = \hat{h}(S_A, \eta, p)$  and  $\mu = \hat{\mu}(S_A, \eta, p)$ , it follows from the fundamental thermodynamic relation (Eqn. (A.7.1)) that

$$\hat{h}_\eta(S_A, \eta, 0) d\eta + \hat{h}_{S_A}(S_A, \eta, 0) dS_A = (T_0 + \theta) d\eta + \mu(S_A, \eta, 0) dS_A, \quad (\text{A.10.1})$$

which shows that specific entropy  $\eta$  is simply a function of Absolute Salinity  $S_A$  and potential temperature  $\theta$ , that is  $\eta = \eta(S_A, \theta)$ , with no separate dependence on pressure. It follows that  $\theta = \theta(S_A, \eta)$ .

Similarly, from the definition of potential enthalpy and Conservative Temperature in Eqns. (3.2.1) and (3.3.1), at  $p = 0$  dbar it can be seen that the fundamental thermodynamic relation (A.7.1) implies

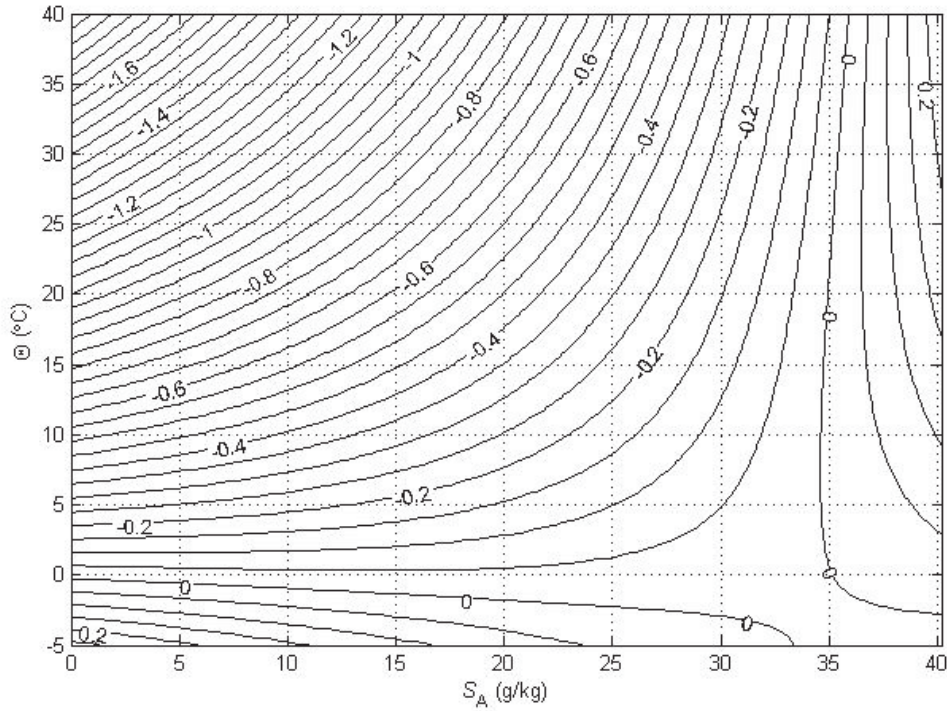
$$c_p^0 d\Theta = (T_0 + \theta) d\eta + \tilde{\mu}(S_A, \theta, 0) dS_A. \quad (\text{A.10.2})$$

This shows that Conservative Temperature is also simply a function of Absolute Salinity and potential temperature,  $\Theta = \Theta(S_A, \theta)$ , with no separate dependence on pressure. It then follows that  $\Theta$  may also be expressed as a function of only  $S_A$  and  $\eta$ . It follows that  $\Theta$  has the “potential” property.

Also, note Figure A.17.1 below (from IOC *et al.* (2010)) showing the difference between potential temperature  $\theta$  and Conservative Temperature  $\Theta$ . In the `gsw_pt_from_entropy` function we have approximated the figure below as

$$(\theta - \Theta) \approx -0.05(1 - S_A/S_{SO})\Theta,$$

in order to obtain an initial estimate for  $\tilde{\Theta}_\theta$  in the iterative modified Newton-Raphson procedure.



**Figure A.17.1.** Contours (in °C) of the difference between potential temperature and Conservative Temperature  $\theta - \Theta$ . This plot illustrates the non-conservative production of potential temperature  $\theta$  in the ocean.