

Interactive comment on “Numerical implementation and oceanographic application of the Gibbs thermodynamic potential of seawater” by R. Feistel

J. Reissmann

jan.reissmann@io-warnemuende.de

Received and published: 26 December 2004

MATLAB implementation of the thermodynamic functions based on the Gibbs potential presented by R. Feistel

I have passed a set of MATLAB functions derived from his Gibbs potential to R. Feistel. He will add them to the supplements of this paper or they will be available as download on his homepage at least. This may (hopefully) be of interest because MATLAB is widely used for data analysis and presentation in the oceanographic community. I for one have used these functions (based on the 1995 Gibbs thermodynamic potential by R. Feistel) for the analysis of large three-dimensional CTD data fields with the order of tenthousand data points for each quantity.

Full Screen / Esc

Print Version

Interactive Discussion

Discussion Paper

For the presented new Gibbs potential from 2003 I have completely updated and debugged the set of functions. The programming of the functions is fully according to the MATLAB philosophy from my point of view. That means the calculations are array oriented and all combinations of equally sized matrices and scalars are allowed as input arguments. A help header is given in each function to describe how to use it. The case of salinity $S=0$ is considered explicitly where needed. This concerns the converging term $x^2 \ln(x)$ which leads to a bug from the diverging $\ln(x)$ term if programmed as it is. The diverging terms resulting from negative powers of x and from the solely standing $\ln(x)$ term in derivatives of the Gibbs potential with respect to salinity S are kept as $+\text{Inf}$ and $-\text{Inf}$ respectively which may lead to NaNs in their summation. For the respective functions these cases are mentioned in the help headers given in each function.

MATLAB functions are given for all thermodynamic quantities derived solely from the seawater Gibbs potential presented by R. Feistel in this article which include the Gibbs potential itself and its first derivatives (apart from their sign) in particular and all second derivatives. Only the specific volume as the reverse of the density which is needed more frequently is omitted. The input arguments of all of these functions (symbols refer to the paper of R. Feistel) are the salinity S in psu, the temperature t in $^{\circ}\text{C}$ (not the absolute temperature T in K), and the pressure p in dbar (not the absolute pressure P , but P reduced by normal atmospheric pressure P_0 , and in dbar instead of Mpa) as used in the main representation of the Gibbs potential in equation (1), because oceanographic data is commonly given using these variables. So, conversions of the variables are not necessary in the most oceanographic applications of the functions given here.

Each of the functions is hardcoded as a separate routine and given an (hopefully) intuitive name. This should make their use more comfortable and the code calling them clearer and easier to understand. So, bugs can be avoided better or at least reduced. Another reason for having a separate function for each quantity is to make their calculation faster by avoiding many logical statements needed within an all-in-

[Full Screen / Esc](#)[Print Version](#)[Interactive Discussion](#)[Discussion Paper](#)

one function which are very time consuming, particularly, if an interpreter language like MATLAB is employed. The seawater properties calculated as combinations of the basic functions are implemented straight forward as these combinations using the hardcoded basic functions.

The potential temperature is calculated by solving the entropy conservation equation (which gives an implicit definition of the potential temperature) by means of Newton iteration as proposed by R. Feistel. This function has two additional optional arguments. The first is the reference pressure, the second is the minimum accuracy of the potential temperature, i.e. the maximum absolute value of the difference between two sequential temperatures at which the iteration stops. Their default values are 0 (surface pressure in the used variables) and 0.000001 °C respectively. While for the reference pressure a scalar or an appropriately sized matrix are allowed, for the accuracy a scalar is expected.

Other potential quantities can be calculated straight forward from the potential temperature and the reference pressure as proposed by R. Feistel. But otherwise they can also be calculated within an iteration of the potential temperature using the absolute value of the difference between their sequential values as criterion to stop the iteration to have control over their accuracy. This is done in the given function for the potential density as an example. Again, this function has the two optional arguments for the reference pressure (scalar or appropriately sized matrix) and the accuracy (scalar expected) like for the potential temperature. Their default values are 0 for reference pressure again and 0.000001 kg/m³ for accuracy, because it is for density now. The advantages/disadvantages of this method with respect to the straight forward calculation of potential quantities from potential temperature and reference pressure are discussed briefly in the help header of the potential density function.

One routine in the set is just for the writing of the coefficient matrix of the Gibbs potential to a binary file, because the calling of a variable from a binary file is faster than filling it element by element each time one of the functions is called. The set of functions is

[Full Screen / Esc](#)[Print Version](#)[Interactive Discussion](#)[Discussion Paper](#)

Interactive
Comment

ready to use without any call of this coefficient writing routine, because the binary file with the original coefficient matrix is passed along with the set of functions. In fact, this routine is needed only if one intends to change coefficients in the matrix and included in the set of functions for reasons of completeness only.

It is planned to implement such a set of functions also in IDL programming language and it is likely that I decide to add analogous sets of functions for the Gibbs potential of sea ice mentioned by R. Feistel both in MATLAB and IDL. I hope to finish these sets soon and would like to make them available for the community as a supplement of a letter in this journal in spring 2005 if I feel the demand for that.

Interactive comment on Ocean Science Discussions, 1, 1, 2004.

[Full Screen / Esc](#)[Print Version](#)[Interactive Discussion](#)[Discussion Paper](#)