Int. J. Engng Ed. Vol. 25, No. 1, pp. 3–10, 2009
Printed in Great Britain.

0949-149X/91 $3.00+0.00
© 2009 TEMPUS Publications.

WATER95—A MATLAB® Implementation of the IAPWS-95 Standard for Use in Thermodynamics Lectures*

P. JUNGLAS
Private University of Applied Sciences, Vechta/Diepholz, Germany. E-mail: peter@peter-junglas.de

WATER95 is an open source implementation of IAPWS95, the current standard formulation of the properties of water and steam for scientific use. It is written in MATLAB and is aimed primarily at educational use. Its value for teaching standard processes with water as a working medium is shown on the basis of worked exercises. They provide examples of isobaric and adiabatic processes and the calculation of the efficiency of the standard Clausius–Rankine cycle for steam turbines, and show how to extend the package for the computation of processes of constant enthalpy. Finally the advantages and drawbacks of using the package in a thermodynamics course are discussed.

Keywords: IAPWS95; water and steam properties; teaching thermodynamics

INTRODUCTION

ONE OF THE CENTRAL SUBJECTS of a thermodynamics course for engineers is the computation of cyclic processes. The students have to learn how to determine the values of the thermodynamic state variables at the corner points using the properties of the state changes that make up the cyclic process. The basic tools are the state equations of the considered medium:

- the thermal state equation relating pressure p, specific volume v and temperature T;
- the caloric state equation giving the specific enthalpy h as function of the thermal variables;
- the entropic state equation giving the specific entropy s as function of the thermal variables.

Generally one starts with processes of the ideal gas using the additional simplification of constant heat capacity—for didactical reasons and because of its practical importance. Here all equations are given explicitly and all computations can be done analytically, which leads to simple formulas (e.g. for the efficiency) that can be analysed afterwards.

The next important medium considered is water (liquid and vapour phases), especially in the technically important Clausius–Rankine cycle. Owing to the complex behaviour of water no simple equations are at hand, so one has to use numerical schemes instead. Most textbooks (e.g. [1, 2]) provide tables that give values for all interesting state variables in the pure vapour and the pure fluid phases as well as along the phase boundary. Together with simple linear interpolation these tables work as a substitute for formal state equations. Though they allow one to do all computations, they nevertheless have some drawbacks:

- Qualitative behaviour (like monotonicity or continuity) is often difficult to recognise.
- All relations between the state variables are hidden; they seem to be totally independent.
- Two-dimensional interpolation with a standard pocket calculator is a tedious task. Therefore only very simple problems can be used in examinations due to time restrictions.

In industrial applications the water tables are replaced by carefully designed equations, which reproduce the measured properties of water within the known accuracy. Two such formulations are now in wide use:

- IAPWS-95 [3, 4], the current standard for general and scientific use, as proposed by the IAPWS [5];
- IAPWS-IF97 [6, 7], designed for the special requirements of the steam power industry, which is easier to implement and generally faster than IAPWS-95.

IF97 has been implemented for many environments, e.g. in C++ [8], in JavaScript [9], in MS-Excel [10] and for several pocket calculators [11]. Some of these programs are open source. For IAPWS95 two commercial software packages exist ([12, 13]), the last one is free for university use, but not open source. The freesteam project [8] contains some basic IAPWS95 routines, but is still very rudimentary.

WATER95, the package presented here, is an open source MATLAB implementation of

* Accepted 21 April 2008.

IAPWS95. Utilising it in a thermodynamics course has several advantages over using one of the IF97 programs:

- IAPWS95 is the current scientific standard providing the most accurate and self-consistent values.
- It is based on the computation of the free energy. This shows a concrete application of the abstract formalism of thermodynamical potentials.
- MATLAB is generally well known to the students, so they can study and enhance the code.

After a short summary of the basic ideas of the IAPWS95 standard we will describe the main routines of WATER95 and show, with some examples, how they can be used in a thermodynamics course.

THE IAPWS95 STANDARD

The basic ingredient of the IAPWS95 formulation is an explicit formula for the specific free energy $f(\rho, T)$ of water as a function of the density $\rho = 1/v$ and the temperature T, which reproduces all measurements with a high degree of accuracy. The structure of this formula is based on mathematical considerations and physical insight, its many coefficients are fitted to the experimental data using a complicated iteration scheme. The standard is described in [4]; all details can be found in [3] and references therein.

From the free energy the three state equations can be easily computed as:

$$\begin{aligned} p(\rho, T) &= \rho^2 \frac{\partial f}{\partial \rho} \\ h(\rho, T) &= f - T\frac{\partial f}{\partial T} + \rho\frac{\partial f}{\partial \rho} \\ s(\rho, T) &= -\frac{\partial f}{\partial T} \end{aligned} \tag{1}$$

For values of ρ and T inside the two-phase region (wet steam) the formula for the free energy gives unphysical results. The behaviour can be best seen in a p–v diagram, where p should decrease with v, but instead oscillates (Fig. 1).

Here one has to use the well-known Maxwell construction to get the saturation pressure p_S and the specific volumina v' and v'' of pure water and pure vapour: p_S has to be chosen such that the areas of the graph of $p(v)$ below and above the line $p = p_S$ are equal, i.e.

$$p_S(T)(v'' - v') = \int p(v, T)dv \tag{2}$$

In terms of the densities this amounts to solving the following equations simultaneously for $p_S(T)$, ρ' and ρ'':

$$\begin{aligned} p_S(T)\left(\frac{1}{\rho''} - \frac{1}{\rho'}\right) &= f(\rho', T) - f(\rho'', T) \\ p(\rho', T) &= p_S(T) \\ p(\rho'', T) &= p_S(T) \end{aligned} \tag{3}$$

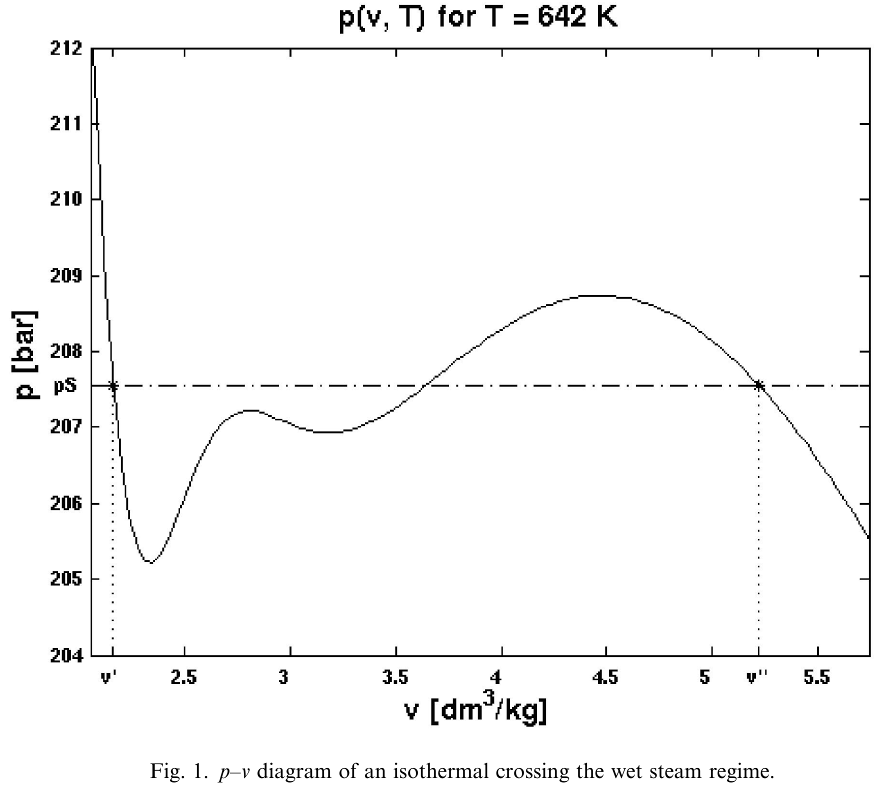

Fig. 1. p–v diagram of an isothermal crossing the wet steam regime.

With this supplemental construction the IAPWS95 formulation reproduces the wet steam properties as well.

A MATLAB IMPLEMENTATION OF IAPWS95

The WATER95 package consists of a set of low level routines implementing the IAPWS95 equations, and user level routines for the standard tasks that are needed in a thermodynamics course. After adding the package directory to the MATLAB path, the public routines can be easily used, whereas the low level routines cannot be accessed directly.

The basic private routine is freeEnergy, which employs several subroutines to reproduce the complicated formula for $f(\rho, T)$. Three raw routines implement Equation (1) for the pressure, enthalpy and entropy without using the Maxwell construction.

The fundamental user routine is `saturationPressure`, which computes $p_S(T)$, ρ' and ρ'' for a given temperature. It contains an iteration scheme that implements (3) and carefully copes with numerical problems near the critical point. To accelerate the iteration it resorts to `auxSaturationPressure` and `auxSaturationDensities` for start values. These auxiliary routines implement the direct formulas given in [14], which are already very accurate, but cannot be used here by themselves because they are not consistent with the IAPWS95 formulation.

The state equations are now implemented in `pressure`, `enthalpy` and `entropy`. These functions return the physically correct values for $p(\rho, T)$, $h(\rho, T)$ and $s(\rho, T)$, using `saturationPressure` to identify the two-phase region. In addition they return the dryness fraction

$$x = \frac{m_{vapor}}{m_{vapor} + m_{water}} \tag{4}$$

Additional functions supply the inverse relations, $\rho(p, T)$, $T(p, \rho)$, $T(p_S)$, as well as some other combinations of state variables that are needed in the applications such as $T(p, s)$ or $\rho(T, x)$ (on the saturation curve). In some cases the inversion with MATLAB's `fzero` function needs carefully chosen starting values due to the complicated behaviour of the function to invert.

The function `clausiusRankineCornerPoints` shows how to apply the basic routines in a concrete example: Given the low and high pressure and the highest temperature of a Clausius–Rankine cycle, it computes the values of all relevant state variables at the corner points. The function `efficiencyCR` directly returns the (thermodynamical and mechanical) efficiencies of the cycle.

To simplify the plotting of state changes in a *p–v* or *T–s* diagram, routines for all standard processes are provided. They return a number of points (including additional corner points at phase boundaries) that can be plotted directly with

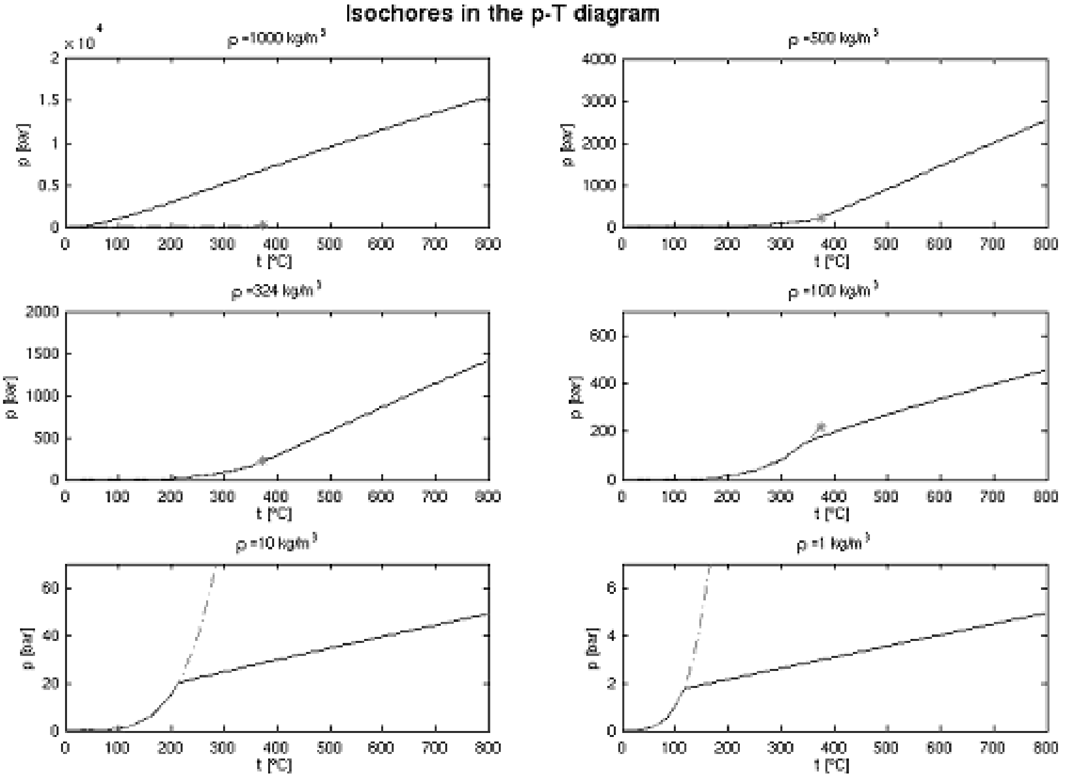

Fig. 2. *p–T* diagrams of several isochors.

MATLAB's `plot` command. Increment values can be chosen to be linear or logarithmic according to the type of plot. The routine `clausiusRankine` combines four state changes to return values for plotting the complete cycle.

Two additional groups of high level functions provide further examples of how to use the package:

- A few routines create standard tables for water, wet and hot steam, as they are used in many textbooks. They could, for example, be useful for an instructor preparing an examination: By providing (amongst others) the values needed in an exercise, the user can reduce the time otherwise wasted for interpolation by hand.
- Some additional plotting routines create the figures one usually shows in the lecture, ready for inclusion in the handout. Among them are plots of isochores such as in Fig. 2, which are rarely used for water, but are very instructive nevertheless.

Since all sources are provided, one can easily adapt these functions to personal needs.

WATER95 has been tested against the values that are provided online at the NIST Chemistry WebBook [15], and the results of the IAPWS95 program from [13]. The package and a complete list of all functions can be found on the WATER95 homepage [16]. It is provided under standard open source conditions.

COMPUTING ISOBARIC AND ADIABATIC PROCESSES

The following exercises are taken from a thermodynamics lecture for mechanical engineers, where the students previously solved them using standard water tables and pocket calculators. Here they additionally provide examples of how to use WATER95.

Example 1: 25 000 kg/h of vapour leave the high pressure turbine of a power plant at a pressure of 4 bar and a temperature of 150°C. Before they enter the low pressure turbine, they are reheated isobarically to 450°C. What is the heat transfer rate to the vapour?

The answer can be easily obtained once one knows the values of the enthalpy before and after the heating. The corresponding WATER95 routine `enthalpy` needs the density and the temperature, therefore the basic idea is to compute the densities using the routine for $\rho(p, T)$. A corresponding MATLAB code is

```
% define the given values, then:
rho1 = density(p1, T1)
h1 = enthalpy(rho1, T1)
rho2 = density(p2, T2)
h2 = enthalpy(rho2, T2)
Qdot12 = mdot*(h2-h1)
```

Example 2: In a turbine 100 kg of vapour at 450°C and 10 bar are expanded adiabatically and internally reversibly to 1.5 bar. How much shaft work is delivered by the turbine in this process?

Again one needs the enthalpies h_1 and h_2 at the start and end point of the process. Using T_1 and p_1 the enthalpy h_1 can be computed as in the last exercise. But for h_2 only p_2 is given explicitly. Since the process is adiabatic, one additionally has $s_2 = s_1$, so the first step of the MATLAB solution is

```
rho1 = density(p1, T1)
h1 = enthalpy(rho1, T1)
s1 = entropy(rho1, T1)
```

Now the problem boils down to computing $h(\rho, T)$ given p and s. At this point many students are stuck. They see that the solution should be possible, since two values are generally sufficient to define the thermodynamical state, but don't understand how. After the hint to explicitly write down the functions and their variables that are included in WATER95, and collecting the results in front of the class, the following scheme emerges:

a) Use $s(\rho, T)$ and $p(\rho, T)$ to 'somehow' find ρ and T, then compute $h(\rho, T)$.
b) Refinement: Invert $p(\rho, T)$ to get $T(p, \rho)$ (or: use the corresponding function `temperature`) and insert it in $s(\rho, T)$.
c) Solve $s = s(\rho, T(p, \rho))$ for ρ, get $T = T(p, \rho)$, then h.

Having found a strategy, the next question is how to implement it in MATLAB. After a recapitulation of how to invert a function using `fzero` (and/or a glance at the source code of `temperature`), a MATLAB solution is found such as:

```
f1 = @(rho) entropy(rho,temperature
(p2,rho))-s1;
rho2 = fzero(f1, rho0);
```

The remaining problem is to find a good starting value `rho0`. Here a physical intuition is needed: Since we are looking at vapour, a value in the order of 1 kg/m^3 should (and indeed does) work. Presenting the effect of using 10 kg/m^3 instead—the result is `NaN` since `fzero` extends its search to negative values of ρ—gives the students a first clue about the nontrivial numerical problem that is hiding here.

EFFICIENCY OF THE CLAUSIUS–RANKINE CYCLE

The main objective of introducing water as a working fluid in a basic thermodynamics lecture is of course the study of the steam turbine. Its operation is described by the Clausius–Rankine cycle, which consists of two isobaric and two adiabatic processes.

The next example combines and extends the methods of the two previous exercises to compute the efficiency of a Clausius–Rankine cycle:

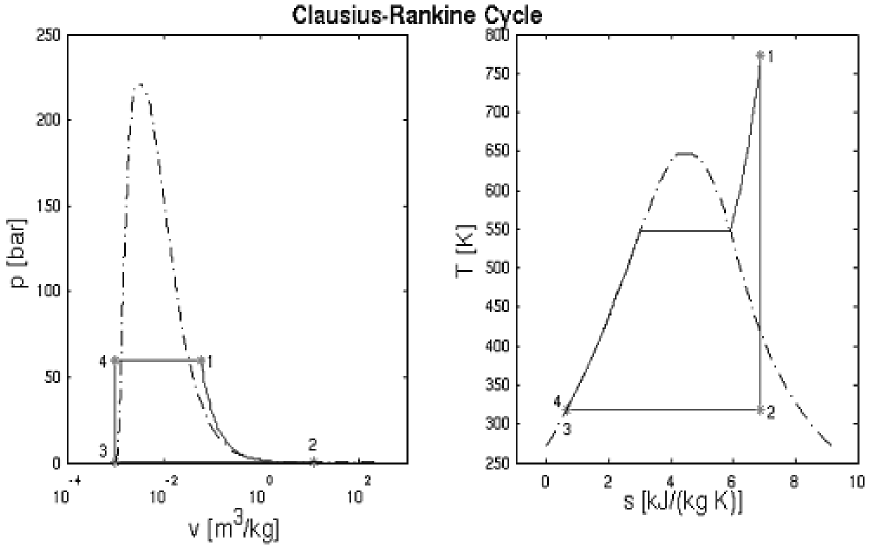

Fig. 3. *p–v* and *t–s* diagrams of the Clausius–Rankine cycle.

Example 3: The vapour leaves the boiler of a steam turbine with 60 bar and 500°C. The pressure in the condenser is 0.1 bar. Compute the thermal efficiency of the cycle.

Again the problem is to get the enthalpies at the four corner points of the cycle. One starts at the point after the boiler, where the pressure and the temperature are known, so that h_1 can be computed as in Example 1. Computation of h_2 proceeds as in Example 2, but here the problem of finding a good starting value is acute: If one uses $\rho = 1$ kg/m^3 as above, `fzero` fails! A standard way to cope with this situation is to plot the function that has to be inverted over a reasonable range of values, e.g. with

```
f1 = @(rho) entropy(rho,temperature
(p2,rho))-s1;
rr = 0.02:0.02:2;
ss = vectorize1d(f1, rr);
plot(rr, ss, ‘k-’, rr, 0*ss, ‘k:’);
```

A good start value can now be read off from the plot.

Finding h_3 needs a new kind of argument: The point lies on the saturation curve at the pure water side (i.e. $x = 0$). Given the (saturation) pressure $p_3 = p_2$ one computes the temperature and the density of the water phase to get h_3:

```
T3 = saturationTemperature(p2)
[pS, rho3, rhopp] = saturation
Pressure(T3)
h3 = enthalpy(rho3, T3)
```

The last point again follows the lines of Example 2. Since one deals with (liquid) water here, one can

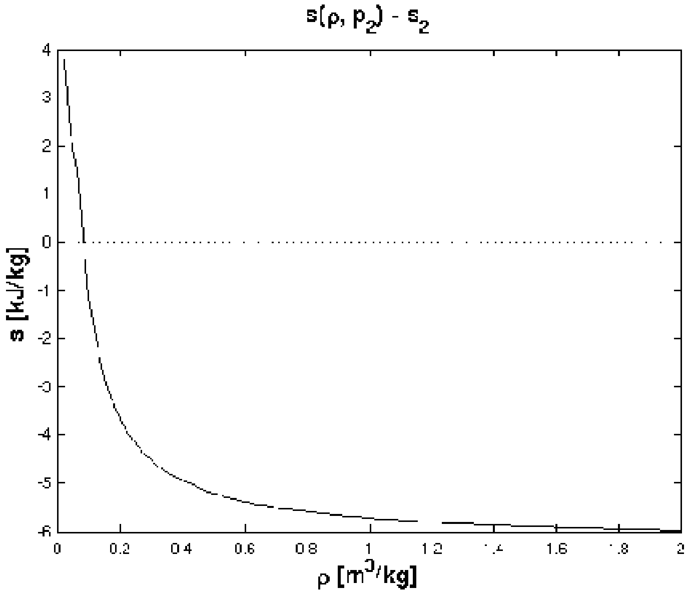

Fig. 4. Auxiliary plot of *s* over *ρ*.

successfully use a standard density of ρ = 1000 kg/m^3 for a starting value.

Ploughing one's way around the circle one thus collects all the data needed. Admittedly the presented method has a pedestrian touch and doesn't lead itself immediately to a quick generalisation, for example, for writing a routine like `efficiencyCR`. If one has enough time or a focus on numerical methods, one can discuss the implementation of the routine `temperaturePS`, which reveals how to find correct starting values automatically using a rough classification of the input and simple interpolation functions.

EXTENDING WATER95

The final example deals with isenthalpic processes (where h is constant). Since there are no corresponding routines included in WATER95, one has to extend the package. This requires a deeper familiarity with MATLAB programming, as achieved, for example, in a programming or a numerical mathematics course.

Example 4: Write a routine `isenthalpic` that computes values of the state variables along a line of constant enthalpy. Use it to plot the isenthalpic line h = 2600 kJ/kg, for specific volumes from 0.005 m^3/kg to 100 m^3/kg. What happens for h = 1200 kJ/kg (using the same volume range)?

The first question to consider is how to describe a given isenthalpic line, i.e. which parameters the routine should have. An important aspect is consistency with the rest of the package, which leads one to use start pressure, start and end volume and the following function declaration

```
function [p,v,T,s,x] = isenthalpic
(p1,v1,v2,N,flag)
```

Starting with a copy of `isentropic` the implementation consists mainly in inserting the calculation of the enthalpy (instead of the entropy) from the initial conditions. Additionally one has to replace the auxiliary function `temperatureRS`, which computes the temperature given ρ and s, by a new function `temperatureRH`, which inverts $h(\rho, T)$ instead of $s(\rho,T)$. The interesting point here is to find good starting values of the temperature for `fzero`. Whether the simple approach of `temperatureRS`, which uses a constant value, works here, is not evident beforehand, but has to be checked carefully.

After these preparations the plotting of the first isenthalpic line proceeds in a straightforward way by converting the given set of state variables to the needed ones:

```
rho1 = 1/v1;
T1 = temperatureRH(rho1, h);
p1 = pressure(rho1, T1);
[pi,vi,Ti,si,xi] = isenthalpic
(p1,v1,v2,100,1);
plotPVandTS(pi,vi,Ti,si,N, 350e5).
```

See Fig. 5.

Repeating the procedure with the lower enthalpy value, one gets a strange result: as shown in Fig. 6.

Since the students have little experience with the interpretation of isenthalpic processes, they tend to believe the result and explain the sharp bend in the T–s diagram as an artefact of a large volume step. One could easily check this by enlarging the number of points by a factor of 10, but this takes some time: the computation needs about half an hour on current standard PCs! Instead a look at the numbers is more illuminative: The values at the endpoint are:

$$v = 100 \text{ m}^3/\text{kg}$$
$$p = 0 \text{ bar}$$
$$s = 8.819 \text{ kJ/(kg K)}$$
$$T = 273.15 \text{ K}$$

The unphysical pressure value clearly indicates a problem, and the temperature gives the clue to its

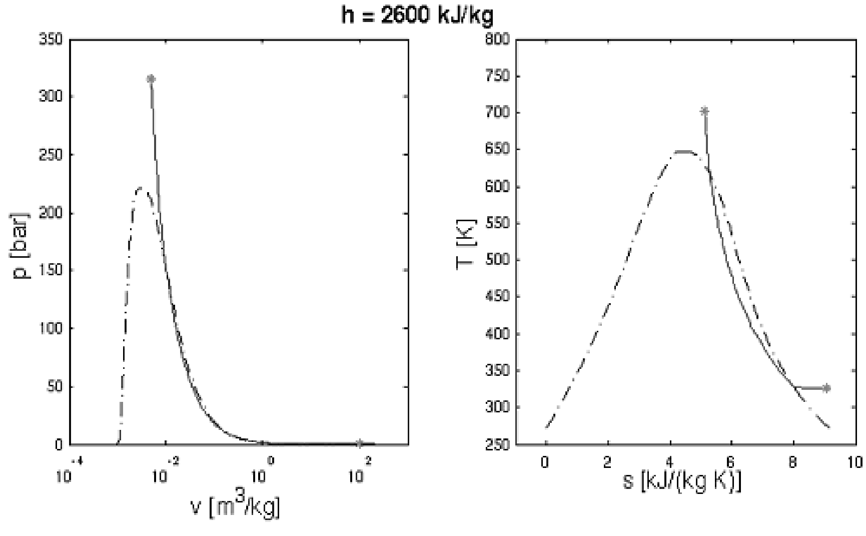

Fig. 5. *p*–*v* and *t*–*s* diagrams of an isenthalpic line for high *h* value.

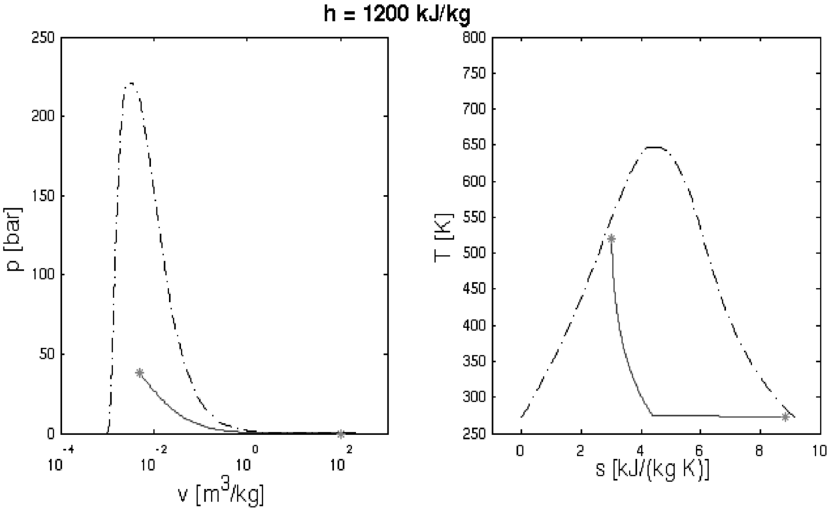

Fig. 6. *p–v* and *t–s* diagrams of an isenthalpic line for low *h* value.

cause. It is below the triple point of water! A look at the WATER95 homepage reveals:

> Missing are: (. . .) behaviour below the temperature of the triple point, . . .

Opening a discussion on how the isenthalpic line continues for low temperatures is a good check of the students' general understanding. What actually happens is that the line reaches the sublimation region, which is outside the scope of WATER95 (so far, at least).

CONCLUSIONS

Introducing the WATER95 package in a thermodynamics course has several benefits:

- More complex systems can be analysed in the lecture or in private study.
- The students are confronted with a software tool for thermodynamic property calculations. This comes closer to many real life situations than the usual table-based approach.
- They learn to cope with numerical problems and software limitations, which helps to build up a—so often lacking—critical understanding of computer results.

Unfortunately these advantages do not come free. Many students already struggle with the conceptual problems of thermodynamics and are not inclined to face up to the additional difficulty of learning a new software package. But if MATLAB itself is already well known, for example as a tool in a previous numerical mathematics course, then using the WATER95 package should be quite easy. Furthermore the numerical methods used in the implementation can be compared with the methods that have been learned before.

Though WATER95 is ready for use in educational applications, it is far from being a complete and robust IAPWS95 implementation. The main points lacking are:

- functions for second-order derivatives like heat capacities or the speed of sound;
- correct values below the triple point temperature;
- thorough testing of all routines in all regions of phase space.

In addition some of the routines are rather slow. The overall performance, especially of inversion functions, should be enhanced, for example by providing better estimates for start values.

Since WATER95 is an open source project [16], everybody who is interested in MATLAB based IAPWS95 computations is invited to contribute.

Acknowledgement—It is a pleasure for me to thank Thomas Schramm and the other members of the Department of Geomatics at the Harbour City University Hamburg for many helpful discussions and the hospitality extended to me.

REFERENCES

1. M. Moran and H. Shapiro, *Fundamentals of Engineering Thermodynamics,* John Wiley & Sons, Chichester (2006).
2. W. Cerbe, *Technische Thermodynamik*, Hanser, München (2004).
3. W. Wagner and A. Pruss, The IAPWS formulation 1995 for the thermodynamic properties of ordinary water substance for general and scientific use, *J. Phys. Chem. Ref. Data,* **31**, 2002, pp. 387–535.

4. International Association for the Properties of Water and Steam, IAPWS formulation 1995 for the thermodynamic properties of ordinary water substance for general and scientific use, IAPWS Release, IAPWS Secretariat, (1996).
5. International Association for the Properties of Water and Steam, 2007. *IAPWS homepage* [online]. Available from: http://www.iapws.org/ [accessed: 21 April 2008].
6. W., Wagner, J. R., Cooper *et al.*, The IAPWS industrial formulation 1997 for the thermodynamic properties of water and steam, *J. Eng. Gas Turbines Power,* **122**, 2000, pp. 150–182.
7. W. Wagner and H.-J. Kretzschmar, *International Steam Tables—Properties of Water and Steam Based on the Industrial Formulation IAPWS-IF97*, Springer, Berlin (2007).
8. freesteam project, 2007. *freesteam homepage* [online]. Available from: http://freesteam.sourceforge.net/ [Accessed: 21 April 2008].
9. L., Kos, *IAPWS-IF97 implementation in JavaScript* [online]. Laboratory for Engineering Design, University of Ljubljana, (2005). Available from: http://www.lecad.uni-lj.si/~leon/research/iapws-if97/IAPWS97.html [accessed: 21 April 2008].
10. H.-J., Kretzschmar, I. Stöcker *et al. Stoffwertprogrammbibliothek für die Industrie-Formulation IAPWS-IF97 von Wasser und Wasserdampf* [online], Hochschule Zittau/Görlitz, (2007). Available from: http://thermodynamik.hs-zigr.de/cmsfg/Stoffwertbibliothek/index.php [accessed: 21 April 2008].
11. H.-J., Kretzschmar, I. Stöcker *et al. Property Library for the Industrial Formulation IAPWS-IF97 of Water and Steam* [online]. Hochschule Zittau/Görlitz (2007). Available from: http://www.steam-tables-pocket-calculators.com/ [accessed: 21 April 2008].
12. National Institute of Standards and Technology, *NIST/ASME Steam Properties Database* [online], (2007). Available from: http://www.nist.gov/data/nist10.htm [accessed: 21 April 2008].
13. W. Wagner,. *Software for the IAPWS-95 Formulation* [online]. Ruhr-Universität Bochum, (2007). Available from: http://www.ruhr-uni-bochum.de/thermo/Software/Seiten/IAPWS-95-eng.htm [accessed: 21 April 2008].
14. W. Wagner and A. Pruss, International equations for the saturation properties of ordinary water substance. revised according to the international temperature scale of 1990, *J. Phys. Chem. Ref. Data,* **22**, 1993, pp. 783–787.
15. National Institute of Standards and Technology, *Thermophysical Properties of Fluid Systems* [online], (2005). Available from: http://webbook.nist.gov/chemistry/fluid/ [accessed: 21 April 2008].
16. P. Junglas, *WATER95 homepage* [online]. FHWT Vechta/Diepholz, (2007). Available from: http://www.peter-junglas.de/fh/water95/index.html [accessed: 21 April 2008].

Peter Junglas is professor of physics and computer science at the Department of Mechanical Engineering of the University of Applied Sciences Vechta/Diepholz (FHWT). He received his PhD at the University of Hamburg with a topic that covered general quantum field theory. After stays at the University of Goettingen and the MPI for Aeronomy in Katlenburg/Lindau, he worked at the Computing Centre of the TU Hamburg-Harburg. His present interests cover the development of multimedia techniques for teaching, as well as the broad use of simulation techniques.