Reply:

In his letter, Dr. Salim reports that he implemented the new algorithm for solving cubic equations, but failed to obtain the CPU time reduction over the Cardano method that had been mentioned in our paper.

First of all it has to be stated that we fully agree with the principles of efficient programming laid down in his letter. His analysis of the Cardano algorithm is fine, and his lines of C code practically match those in our program. Therefore the discrepancies of CPU time reductions are probably not the result of different implementations of the Cardano algorithm.

It may be worthwhile mentioning that using the C library cbrt(3M) instead of the generic power function pwr(3M) for computing cube roots simplifies the Cardano program, but that—at least on the computers available to us—the construct

```
((x > 0.0)   ?   pow(x, 1.0/3.0) :
                        −pow(−x, 1.0/3.0))
```

appears to run faster [pow(3M) is officially not defined for negative arguments, although some compilers do not seem to mind].

Concerning the efficiency of the new algorithm one has to keep in mind that its second step is a search of the interval $[-w; +w]$ with $w = 1 + \max(|b_i|)$, $i = 0, 1, 2$, where the b_i are the coefficients of the reduced cubic polynomial

$$x^3 + b_2x^2 + b_1x + b_0 = 0 \qquad (1)$$

According to Vieta's rules, the coefficient b_0 is the product of the three real or complex roots, x_1 to x_3

$$b_0 = -x_1x_2x_3 \qquad (2)$$

If the x_i represent volumes given in cm^3/mol, b_0 can be a rather large number, and consequently the search interval can be very wide, which slows down the algorithm.

It had been pointed out in our paper that it is advantageous to formulate the calculation of roots of an equation of state in terms of densities rather then volumes. For example, we implemented the equation of state of van der Waals in the *ThermoC* program package[1,2] in the following way[1]

$$p = \frac{RT}{V_m - b} - \frac{2\pi RT^*}{V_m^2} \qquad (3)$$

By introducing reduced variables

$$\tilde{\rho} = b/V_m \qquad \tilde{T} = T/(2\pi T^*) \qquad \tilde{p} = pb/(2\pi RT^*) \qquad (4)$$

this equation of state is transformed into

$$\tilde{p} = \frac{\tilde{T}\tilde{\rho}}{1 - \tilde{\rho}} - \tilde{\rho}^2 \qquad (5)$$

From this the following cubic equation for the reduced density is obtained

$$\tilde{\rho}^3 - \tilde{\rho}^2 + (\tilde{p} + \tilde{T})\tilde{\rho} - \tilde{p} = 0 \qquad (6)$$

Note that the physically meaningful solutions for $\tilde{\rho}$ are now in the range [0; 1] and that the reduced temperature and pressure are small positive numbers; thus the problem of a search range that is too wide is avoided.

Our recent implementation of the new algorithm has CPU time consumptions of 1.7–1.9 μs (one real root) or 2.1–2.2 μs (three real roots) on a Hewlett–Packard B1000 workstation (HP-PA2 processor, 300 MHz); the corresponding CPU times for the Cardano method are 2.1 and 3.5 μs, respectively. On an Intel Centrino processor with 1.4 GHz the results are 0.5–0.9 μs (one root) and 0.9–1.1 μs (three roots); the times for the Cardano algorithm are 1.0–1.3 and 1.3–1.7 μs, respectively.

These CPU time results are valid for cubic polynomials in which all coefficients are of similar size, and we must stress the point that the problem of the density calculation for an equation of state can always be programmed in such a way that this condition is fulfilled.

If the condition is not fulfilled, that is, in the case of cubic polynomials with arbitrary coefficients, the CPU time required by the new algorithm can increase by a factor of about 2, and then it is indeed no longer faster than Cardano's algorithm. In this case, Dr. Salim's observation is confirmed.

For the sake of completeness we note that the iterative algorithm can be improved to cope with coefficients of arbitrary size by introducing a scaling factor, that is, by transforming Eq. 1 into

$$z^3 + \frac{b_2}{\lambda}z^2 + \frac{b_1}{\lambda^2}z + \frac{b_0}{\lambda^3} = 0 \qquad \text{with} \qquad z = \lambda x \qquad (7)$$

The scaling factor λ could in principle be estimated from

$$\lambda = \max(\sqrt[3]{|b_0|}, \sqrt{|b_1|}, |b_2|) \qquad (8)$$

but that would require time-consuming evaluations of root functions. For practical purposes, however, rough estimates for the roots are sufficient; they can be obtained by extracting the exponents of the floating-point representations of the b_i with the C library function frexp(3M), dividing them by 2 or 3, and converting the largest result back to a floating-point representation with ldexp(3M). With the λ value obtained this way, Eq. 7 is solved for z, and from this x is recovered. This algorithm is stable for arbitrary values of the polynomial coefficients, but it is no longer faster than Cardano's method.

Finally it should be noted that a further CPU time reduction can be achieved for the van der Waals equation by omitting the reduction step (the division of the polynomial coefficients by a_3 to obtain the form of Eq. 1), if reduced properties are used as in Eq. 6; this saves one floating point division and three multiplications. The same is true for the Redlich–Kwong equation of state and its variations,[3] but not for the Peng–Robinson equation.[4] This way of saving CPU time applies to the Cardano method and the iterative algorithm alike.

Literature Cited

1. Deiters UK. A modular program system for the calculation of thermodynamic properties of fluids. *Chem Eng Technol.* 2000;23:581-584.
2. Deiters UK. ThermoC project homepage. http://thermoc.uni-koeln.de/index.html.
3. Redlich O, Kwong JNS. On the thermodynamics of solutions. V. An equation of state-fugacities of gaseous solutions. *Chem Rev.* 1949;44:233-244.
4. Peng DY, Robinson DB. A new two-constant equation of state. *Ind Eng Chem Fundam.* 1976;15:59-64.

Ulrich K. Deiters
Institute of Physical Chemistry
University at Cologne
Luxemburger Str. 116
D-50939 Köln, Germany
e-mail: ulrich.deiters@uni-koeln.de

[1] The factor 2π is here not only for theoretical reasons; it also ensures that T^* is comparable to the critical temperature, which is numerically convenient.

AIChE Journal, Vol. 51, 3310 (2005)
© 2005 American Institute of Chemical Engineers
DOI 10.1002/aic.10660
Published online August 23, 2005 in Wiley InterScience (www.interscience.wiley.com).