LETTERS TO THE EDITOR

To the Editor:

In "Calculation of Densities from Cubic Equations of State," (April 2002), Deiters[1] reported that Cardano's method was not the fastest algorithm for finding the roots of a cubic polynomial, and he proposed an algorithm based on iteration that was found to be faster than Cardano's method. My finding was the contrary, as shown later in this letter.

A cubic polynomial to be solved is Eq. 5, the normalized form, in Deiters,[1] and the computer program for Cardano's method in C/C++ language is partially shown below:

```
p = b1 − b2*b2/3.;                                  (1)

q = b0 − b1*b2/3. + 2.*b2*b2*b2/27.;                (2)

a2 = b2/3.;                                         (3)

d = p*p*p/27. + q*q/4.;                             (4)
```

If d > 0, there is one real root (x0) as follows:

```
arg = pow(−q/2. + sqrt(d), (1./3.));                (5)

x0 = arg − (p/3.)/arg − a2;                         (6)
```

Otherwise, there are three real roots (x0, x1, and x2):

```
theta
    = acos(−q/2.*sqrt(−27./(p*p*p)));               (7)

rootq = 2.*sqrt(−p/3.);                             (8)

x0 = rootq*cos(theta/3.) − a2;                      (9)

x1 = rootq*cos((theta + 2.*M_PI)/3.)
                                       − a2;        (10)

x2 = rootq*cos((theta + 4.*M_PI)/3.)
                                       − a2;        (11)
```

The previous program was made as efficient as possible by taking into consideration of

(1) using "slow" functions (such as pow, sqrt, and trigonometric functions) sparingly or only when they were necessary

(2) as much as possible replacing the "slow" functions with mathematical operations that were very quick in execution

(3) avoiding repeats of the same calculation which especially used "slow" functions

The above-mentioned consideration was also used for Deiters' iterative method.

An example in Walas,[2] for solving the compressibility factors of the Redlich–Kwong EOS, was used to compare the execution speed of Cardano's method and Deiters' iterative method. As suggested by Deiters,[1] Cardano's method was always followed with a Newton iteration step for one root of interest to guarantee the root's precision of 10^{-15} or better. The same precision was also guaranteed for Deiters' iterative method.

Test runs on a personal computer [Dell Series GX280 (processor Intel Pentium 4, 3.6 GHz)] showed that, when efficient computer codes were written, Cardano's method was still faster than Deiters' iterative method, as seen in Table 1.

Table 1. CPU Time Required to Determine All Roots of a Cubic Polynomial

Temperature (K)	Pressure (atm)	Time (μs)	
		Cardano's Method	Deiters' Iterative Method
300*	11.81	0.821	1.262
320*	18.51	0.821	1.261
340*	27.39	0.821	1.422
350*	32.72	0.821	1.422
360**	38.68	0.751	1.302
370**	45.27	0.751	1.302

*Three real roots.
**One real root.

Literature Cited

1. Deiters UK. Calculation of densities from cubic equations of state, *AIChE J.* 2002;48: 882-886.
2. Walas SM. *Phase Equilibria in Chemical Engineering.* Stoneham, MA: Butterworth Publishers; 1985 (see example 1.13, p. 49).

Paul H. Salim
Dept. of Coiled Tubing Research & Engineering
BJ Services Company
6620 36th Street S.E.
Calgary, Alberta, Canada T2C 2G4
e-mail: psalim@bjservices.ca

AIChE Journal, Vol. 51, 000–000 (2005)

DOI 10.1002/aic.10659
Published online August 19, 2005 in Wiley InterScience (www.interscience.wiley.com).