Finding Real Roots of Quartics

Don Herbison-Evans (donherbisonevans@yahoo.com)
Department of Software Engineering, School of Information Science, University of Technology, Sydney, Australia
&
Michel Daoud Yacoub & Gustavo Fraidenraich
University of Campinas, Brazil

ABSTRACT

Methods are presented for controlling round-off error and overflow in the analytic solution of quartic equations. The solution of a quartic requires the solution of a subsidiary cubic equation. The cubics derived in the algorithms by Descartes, by Neumark, by Ferrari, by Brown, Yacoub & Fraidenraich, and by Christianson are examined.

An operation count of the resulting algorithms is presented, and statistics presented after performing a wide ranging comparison of their rounding-error behaviours.

The article updates a previous article that looked only at the three older algorithms.

INTRODUCTION

This article addresses the problems of solving quartic equations. These equations are interesting as they can be solved by analytic algorithms, and in principle need no iterative techniques. So far, five distinctly different algorithms have been published for this. This article updates a previous article (Herbison-Evans, 1995) that looked only at the three older algorithms.

Quartic equations need to be solved when ray tracing 4th degree surfaces e.g., a torus. Quartics also need to be solved in a number of problems involving quadric surfaces. Quadric surfaces (i.e. ellipsoids, paraboloids, hyperboloids, cones) are useful in computer graphics for generating objects with curved surfaces (Badler, 1979). Fewer primitives are required than with planar surfaces to approximate a curved surface to a given accuracy (Herbison-Evans, 1982).

Bicubic surfaces may also be used for the composition of curved objects. They have the advantage of being able to incorporate recurves: lines of inflection. There is a problem, however, when drawing the outlines of bicubics in the calculation of hidden arcs. The visibility of an outline can change where its projection intersects that of another outline. The intersection can be found as the simultaneous solution of the two projected outlines. For bicubic surfaces, these outlines are cubics, and the simultaneous solution of two of these is a sextic which can only be solved by iterative techniques. For quadric surfaces, the projected outlines are quadratic. The simultaneous solution of two of these leads to a quartic equation which can be solved analytically.

ITERATIVE TECHNIQUES

The roots of quartic equations can be obtained by iterative techniques. These techniques can be useful in animation where scenes change little from one frame to the next. Then the roots for the equations in one frame are good starting points for the solution of the equations in the next frame. There are two problems with this approach.

One is storage. For a scene composed of 'n' quadric surfaces, storage may be required for 4n(n - 1) roots between frames. A compromise is to store pointers to those pairs of quadrics which give no roots. This trivial idea can be used to halve the number of computations within a given frame, for if quadric 'Q1' has no intersection with quadric 'Q2', then 'Q2' will not intersect 'Q1'.

The other problem is more serious: it is the problem of deciding when the number of roots changes. There appears to be no simple way to find the number of roots of a quartic. The most well-known algorithm for finding the number of real roots, the Sturm sequence, involves approximately as much computation as solving the equations directly by radicals (Ralston, 1965). Without information about the number of roots, iteration where a root has disappeared can waste a lot of computer time, and searching for new roots that may have appeared becomes difficult.

Even when a root has been found, deflation of the polynomial to the next lower degree is prone to severe round-off exaggeration (Conte and de Boor, 1980).

Thus there may be an advantage in examining the techniques available for obtaining the real roots of quartics analytically.

QUARTIC EQUATIONS

Quartics are the highest degree polynomials which can be solved analytically in general by the method of radicals i.e.: operating on the coefficients with a sequence of operators from the set: sum, difference, product, quotient, and the extraction of an integral order root. An algorithm for doing this was first published by Cardano in the 16th century (Cardano, 1545). A number of other algorithms have subsequently been published. The question arises: which algorithm is best for use on a computer for finding the real roots in terms of speed and stability.

Very little attention appears to have been given to a comparison of the algorithms. They have differing properties with regard to overflow and the exaggeration of round-off errors. Where a picture results from the computation, any errors may be rather obvious. Figures 1, 2, and 3 show a computer bug composed of ellipsoids with full outlines, incorrect hidden outlines, and correct hidden outlines, respectively. In computer animation, the flashing of incorrectly calculated hidden arcs is most disturbing.

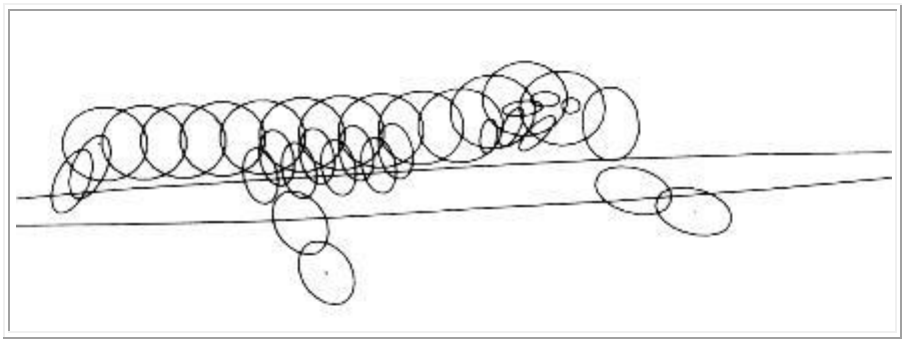

Figure 1: full ellipsoid outlines

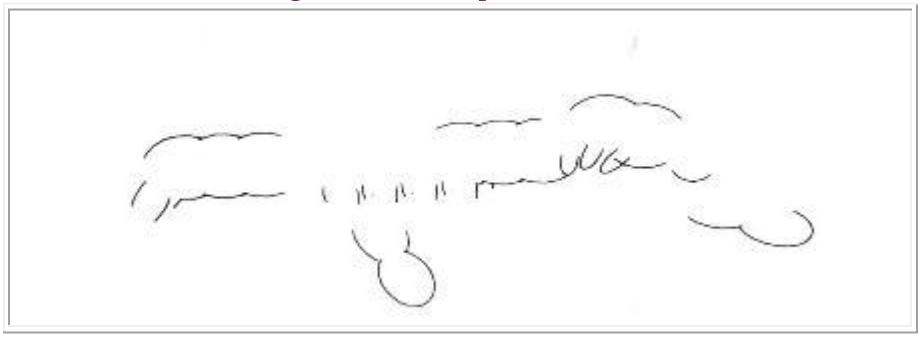

Figure 2: hidden outlines computed using simplistic version of Ferrari's algorithm

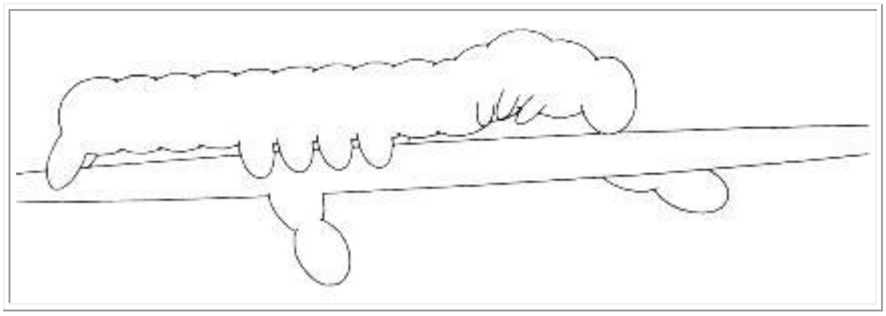

Figure 3: hidden outlines computed using the combined algorithm described in this article

The five algorithms examined here all use the idea of first solving a particular cubic equation, the coefficients of which are derived from those of the quartic. A root of the cubic is then used to factorize the quartic into quadratics, which may then be solved. These five algorithms are particular cases of general transformations discovered by Schmakov (Schmakov, 2011). The five algorithms may be classified according to the way the coefficients of the quartic are combined to form the coefficients of the subsidiary cubic equation. For a general quartic equation of the form:

$$x^4 + ax^3 + bx^2 + cx + d = 0$$

the subsidiary cubic can be one of the forms:

Christianson-Brown (Christianson, 1991)

$$y^3 + (4a^2b - 4b^2 - 4ac + 16d - 3a^4/4)y^2/(a^3 - 4ab + 8c) + (3a^2/16 - b/2)y + (ab/16 - c/8 + a^3/64) = 0$$

Descartes-Euler-Cardano (Strong, 1859)

$$y^3 + (2b - 3a^2/4)y^2 + (3a^4/16 - a^2b + ac + b^2 - 4d)y + (abc - a^6/64 + a^4b/8 - a^3c/4 - a^2b^2/4 - c^2) = 0$$

Ferrari-Lagrange (Turnbull, 1947)

$$y^3 + by^2 + (ac - 4d)y + (a^2d + c^2 - 4bd) = 0$$

Neumark (Neumark, 1965)

$$y^3 - 2by^2 + (ac + b^2 - 4d)y + (a^2d - abc + c^2) = 0$$

Yacoub-Fraidenraich-Brown (Yacoub & Fraidenraich, 2012)

$$(a^3 - 4ab + 8c)y^3 + (a^2b - 4b^2 + 2ac + 16d)y^2 + (a^2c - 4bc + 8ad)y + (a^2d - c^2) = 0$$

The casual user of the literature may be confused by variations in the presentation of quartic and cubic equations. Sometimes, the highest degree term has a non-unit coefficient. Sometimes the coefficients are labelled from the lowest degree term to the highest. Sometimes numerical factors of 3, 4 and 6 are included.

CHRISTIANSON - BROWN

Brown (Brown, 1967) originally published a method of solving palindromic quartics. Christianson (Christianson, 1991) showed how to apply this to a general quartic equation, using the subsidiary cubic equation coefficients:

$$p = (4a^2b - 4b^2 - 4ac + 16d - 3a^4/4)/(a^3 - 4ab + 8c)$$
$$q = 3a^2/16 - b/2$$
$$r = ab/16 - c/8 + a^3/64$$

		a+		a-	
		b+	b-	b+	b-
c+	d+		q	r	q
c+	d-		p q	r	q
c-	d+	r	q		q
c-	d-	r	q	p	p q

Table 1

The coefficients of the subsidiary cubic for

```
Christianson and Brown's algorithm that can be stably
    computed from the coefficients of the quartic.
```

Unfortunately, no combinations of signs of the quartic coefficients give a stable computation of the cubic coefficients.

Having solved the cubic, any solution 'y' may be used to calculate 'k' using either

$$k^4 = y^4 + y^2e_2 + ye_1 + e_0$$

or

$$k^2 = y^2 + e_2/2 + e_1/4y$$

where

$$e_0 = d - ac/4 + a^2b/16 - 3a^4/256$$
$$e_1 = c - ab/2 + a^3/8$$
$$e_2 = b - 3a^2/8$$

The use of the equation for k^4 is only of benefit if either 'y=0' or else 'y' and 'e_1' are negative and 'e_0' and 'e_2' are positive.

The value of 'k' is used to calculate the quantities :

$$g = 4y\,k^3$$
$$h = (6y^2 + e_2)k^2$$

These are used to form the quadratic in 'Z' :

$$Z^2 + gZ/k^4 + (h/k^4 - 2) = 0$$

The roots of this, 'Z_1' and 'Z_2', are then use form the quadratics in 'z' :

$$z^2 - Z_1{*}z + 1 = 0$$

and

$$z^2 - Z_2{*}z + 1 = 0$$

the roots of which are each used to form the corresponding root of the original quartic :

$$x = y - kz - a/4$$

DESCARTES - EULER - CARDANO

This algorithm uses a subsidiary cubic with coefficients :

$$p = 2b - 3a^2/4$$
$$q = 3a^4/16 - a^2b + ac + b^2 - 4d$$
$$r = abc - a^6/64 + a^4b/8 - a^3c/4 - a^2b^2/4 - c^2$$

There is only one combination of quartic coefficients for which the evaluation of the subsidiary cubic coefficients is stable:

		a+		a-	
		b+	b-	b+	b-
c+	d+		p r		p
	d-		p q r		p
c-	d+		p		p
	d-		p		p q

Table 2

For Descartes' algorithm, the coefficients of the subsidiary cubic that can be stably computed from the coefficients of the quartic.

However, if 'a' and 'c' are negated, the combination (a-, b-, c-, d-) becomes stable at the trivial cost of having to negate the resulting quartic solutions.

In this algorithm, the calculation of the cubic coefficients involves significantly more operations than Ferrari's or Neumark's algorithms. Also, the high power of 'a' in the coefficients makes this algorithm prone to loss of precision and also overflow.

In this algorithm, if the greatest root of the cubic, 'y', is negative, the quartic has no real roots. Otherwise, the solutions 'x' of the quartic are obtained from the quadratics:

$x^2 + kx + g = 0$

and

$x^2 - kx + h = 0$

where

$k = y^{(1/2)}$

$g = (y + e_2 + e_1/k)$

$h = (y + e_2 - e_1/k)$

and

$e_1 = c + a^3/8 - ab/2$

$e_2 = b - 3a^2d/8$

There appears to be no way of making the evaluation of 'e_1', 'e_2', 'g' and 'h' stable. Some quantities are bound to be subtracted, leading to possible loss of precision.

FERRARI

Ferrari's algorithm has the following coefficients for the subsidiary cubic :

$p = b$
$q = ac - 4d$
$r = a^2d + c^2 - 4bd$

These have two combinations of signs of 'a', 'b', 'c' and 'd' for which the derivation of all of the coefficients of the cubic, 'p', 'q', and 'r' are stable:

		a+		a-	
		b+	b-	b+	b-
c+	d+	p	p r	p q	p q r
	d-	p q	p q	p	p
c-	d+	p q	p q r	p	p r
	d-	p	p	p q	p q

Table 3

For Ferrari's algorithm, the coefficients
of the subsidiary cubic that can be stably
computed from the coefficients of the quartic.

The coefficients of the subsequent quadratics depend on two intermediate quantities, 'e' and 'f', where

$e^2 = a^2/4 - b - y$

$f^2 = y^2/4 - d$

$ef = (ay/4 + c/2)$

Using Ferrari's method, the quadratic equations giving the solutions to the quartic are :

$x^2 + Gx + H = 0$
and
$x^2 + gx + h = 0$

where

$G = a/2 + e$
$g = a/2 - e$
$H = -y/2 + f$
$h = -y/2 - f$

The signs of each of the quartic coefficients 'a', 'b', 'c', 'd' and the root of the cubic 'y', may be positive or negative, giving 32 possible combinations of signs. Of these, only 12 can be clearly solved in a stable fashion for 'e' and 'f' by the choice of 2 out of the 3 equations involving them. In the remaining 20 cases, the most stable choices are unclear. This is shown in tables 2 and 3. Of the 12 stable cases, 2 are from the stable cases for the calculation of 'p', 'q' and 'r'. In the other 10 cases, the value of y may be unreliable.

y+		a+		a-	
		b+	b-	b+	b-
c+	d+	ef	ef		
	d-	ef f^2	ef f^2	f^2	f^2
c-	d+			ef	ef
	d-	f^2	f^2	ef f^2	ef f^2

Table 4

For Ferrari's algorithm, the intermediate quantities that can be stably computed from the coefficients of the quartic and a positive root of the cubic.

y-		a+		a-	
		b+	b-	b+	b-
c+	d+		e^2	ef	ef e^2
	d-	f^2	e^2 f^2	ef f^2	ef e^2 f^2
c-	d+	ef	ef e^2		e^2
	d-	ef f^2	ef e^2 f^2	f^2	e^2 f^2

Table 5

For Ferrari's algorithm, the intermediate quantities that can be stably computed from the coefficients of the quartic and a negative root of the cubic.

If 'a' and 'e' are the same sign, and 'b' and 'y' are the same sign, then 'g' may be more accurately computed using:

$$g = (b + y)/G$$

If 'a' and 'e' are opposite signs then 'G' can be more accurately computed from 'g' in a similar fashion.

If 'y' and 'f' are the same sign, then 'H' may be more accurately computed using:

$$H = d/h$$

If 'y' and 'f' are opposite in sign, then 'h' can be computed similarly from H more accurately.

The solution of the quadratic equations requires the evaluation of the discriminants:

$$g^2 - 4h \quad \text{and} \quad G^2 - 4H$$

Unless 'h' and 'H' are negative, one or both of these evaluations will be unstable. Unfortunately, positive 'h' and 'H' values are incompatible with the 2 stable cases for the evaluation of 'p', 'q', 'r', 'e' and 'f', so there is no combination of coefficients for which Ferrari's algorithm can be made entirely stable.

NEUMARK

The algorithm of Neumark unfortunately has no combinations of signs of the quartic coefficients for which there is a stable computation of all of the cubic coefficients :

$$p = 2b$$
$$q = ac + b^2 - 4d$$
$$r = a^2d - abc + c^2$$

		a+		a-	
		b+	b-	b+	b-
c+	d+	p	p r	p r	p
	d-	p q	p q	p	p
c-	d+	p r	p	p	p r
	d-	p	p	p q	p q

Table 6

For Neumark's algorithm, the coefficients of the subsidiary cubic that can be stably computed from the coefficients of the quartic.

Again, the solutions 'x' of the quartic are obtained from the quadratics:

$$x^2 + Gx + H = 0$$
and
$$x^2 + gx + h = 0$$

where:

$$G = (a + (a^2 - 4y)^{(1/2)})/2$$
$$g = (a - (a^2 - 4y)^{(1/2)})/2 \text{ and}$$
$$H = (b-y)/2 + (a(b-y) - 2c)/(2(a^2 - 4y)^{(1/2)})$$
$$h = (b-y)/2 - (a(b-y) - 2c)/(2(a^2 - 4y)^{(1/2)})$$

In Nemark's algorithm, the coefficients of the quadratic equations can be computed in a stable fashion from the solution of the cubic. Any cancellation due to the additions and subtractions can be eliminated by writing :

$$G = g_1 + g_2$$
$$g = g_1 - g_2$$
$$H = h_1 + h_2$$
$$h = h_1 - h_2$$

where

$$g_1 = a/2$$
$$g_2 = ((a^2 - 4y)^{(1/2)})/2$$
$$h_1 = (b - y)/2$$
$$h_2 = (a(b-y)/2 - c)/(a^2 - 4y)^{(1/2)}$$

and using the identities

$$g.G = y$$
$$h.H = d$$

Thus if 'g_1' and 'g_2' are the same sign, 'G' will be accurate but 'g' will lose significant digits by cancellation. Then the value of 'g' can be better obtained using:

$$g = y/G$$

If 'g_1' and 'g_2' are of opposite signs, then 'g' will be accurate, and 'G' better obtained using:

$$G = y/g$$

Similarly, 'h' and 'H' can be obtained without cancellation from 'h_1', 'h_2' and 'd'.

The computation of 'g_2' and 'h_2' can be made more stable under some circumstances using the alternative formulation:

$$h_2 = (((b - y)^2 - 4d)^{(1/2)})/2$$

Furthermore

$$g_2 = (a - c)/((b - y)^2 - 4d)^{(1/2)}$$

Thus 'g_2' and 'h_2' can both be computed either using

$$m = (b - y)^2 - 4d$$

or using

$$n = a^2 - 4y$$

If 'y' is negative, 'n' should be used. If 'y' is positive and 'b' and 'd' are negative, 'm' should be used. Thus 7 of the 32 sign combinations give stable results with this algorithm. These are shown in tables 7 and 8.

y+		a+		a-	
		b+	b-	b+	b-
c+	d+	g1	g1 h1	g1	g1 h1
c+	d-	g1	g1 g2 h1 h2	g1	g1 h1 h2
c-	d+	g1	g1 h1	g1	g1 h1
c-	d-	g1	g1 h1 h2	g1	g1 h1 h2

Table 7

For Neumark's algorithm, the intermediate quantities that can be stably computed from the coefficients of the quartic and a positive root of the cubic.

y-		a+		a-	
		b+	b-	b+	b-
c+	d+	g1 g2 h1	g1 g2	g1 g2 h1 h2	g1 g2
c+	d-	g1 g2 h1 h2	g1 g2	g1 g2 h1 h2	g1 g2
c-	d+	g1 g2 h1 h2	g1 g2	g1 g2 h1	g1 g2
c-	d-	g1 g2 h1 h2	g1 g2	g1 g2 h1 h2	g1 g2

Table 8

For Neumark's algorithm, the intermediate quantities that can be stably computed from the coefficients of the quartic and a negative root of the cubic.

For other cases, a rough guide to which expression to use can be found by assessing the errors of each of these expressions by summing the moduli of the addends:

$$e(m) = b^2 + 2|by| + y^2 + 4|d|$$
$$e(n) = a^2 + 4|y|$$

Thus, if

$$|m|.e(n) > |n|.e(m)$$

then 'm' should be used, otherwise 'n' is more accurate.

YACOUB - FRAIDENRAICH - BROWN

Brown (Brown, 1967) originally showed how to solve palindromic quartics, and Yacoub and Fraidenraich (Yacoub & Fraidenraich, 2012) showed a different way of applying this to a general quartic equation using, except for special cases, the subsidiary cubic equation coefficients:

$p = P/U$, $q = Q/U$ and $r = R/U$ with

$$P = a^2b - 4b^2 + 2ac + 16d$$
$$Q = a^2c - 4bc + 8ad$$
$$R = a^2d - c^2$$
$$U = a^3 - 4ab + 8c$$

		a+		a-	
		b+	b-	b+	b-
c+	d+		Q U		
c+	d-	R	R	R	P Q R
c-	d+			U	Q
c-	d-	R	P Q R	R U	R

Table 9

The coefficients of the subsidiary cubic for Yacoub, Fraidenraich and Brown's algorithm that can be stably computed from the coefficients of the quartic.

Unfortunately, no combinations of signs of the quartic coefficients give a stable computation of the cubic coefficients.

Having solved the cubic, any solution 'y' may be used to calculate 'k' using

$$k = a + 4y$$

and this is used to calculate the quantities :

$$e = (a^3 - 4c - 2ab + 6a^2y - 16by)/k$$
$$f^2 = (a^3 + 8c - 4ab)/k$$

and then :

$$g^2 = 2(e + f*k)$$
$$h^2 = 2(e - f*k)$$

where the positive square root may be taken for each of 'f','g', and 'h'. The quartic roots are calculated from these using :

$$x_1 = (-a - f - g)/4$$
$$x_2 = (-a - f + g)/4$$
$$x_3 = (-a + f - h)/4$$
$$x_4 = (-a + f + h)/4$$

If there are three real roots of the cubic, then choosing one with the same sign as 'a' will make the calculation of 'k' more accurate. However there is no guarantee that this alone produces the best results, and all three cubic roots may need to be tried to find the most accurate answers.

OVERFLOW

The different algorithms have differing overflow behaviours for the cubic. Following Tartaglia, let the cubic equation be

$$y^3 + py^2 + qy + r = 0$$

The solution has been published elsewhere in many texts, but here is expressed (Littlewood, 1950) using:

$$u = q - p^2/3$$
$$v = r - pq/3 + 2p^3/27$$

and the discriminant:

$$j = 4(u/3)^3 + v^2$$

If 'j' is positive then there is one root 'y' to the cubic, which may be found using:

$$y = ((w - v)/2)^{(1/3)} - (u/3)(2/(w - v))^{(1/3)} - p/3$$

where

$$w = j^{(1/2)}$$

In terms of overflow, the most obviously affected quantity is discriminant 'j'. In terms of the individual coefficients :

Christianson: $O(j) = O(a^6) + O(b^6) + O(c^2) + O(c^{-6}) + O(d^6)$

Descartes: $O(j) = O(a^{12}) + O(b^6) + O(c^4) + O(d^3)$

Ferrari: $O(j) = O(a^3) + O(b^6) + O(c^4) + O(d^3)$

Neumark: $O(j) = O(a^6) + O(b^6) + O(c^4) + O(d^3)$

Yacoub & Fraidenraich: $O(j) = O(a^6) + O(a^{-6}) + O(b^6) + O(c^4) + O(d^6)$

Before evaluating the terms of 'w', it is wise to test 'p', 'q' and 'r' against the appropriate root of the maximum number represented on the machine ('M'). The values of 'u' and 'v' should similarly be tested. In the event that some value is too large, various approximations may be employed: e.g.

if $|p| > 27M^{(1/3)}$, then $y \sim= -p$

if $|v| > M^{(1/2)}$, then $y \sim= v^{(1/3)}$

if $|u| > 3M^{(1/3)}/4$, then $y \sim= 4^{(1/3)}u/3$

If the discriminant 'j' is negative, then there are 3 real roots to the cubic. These real roots of the cubic may then be obtained via parameters 's', 't' and 'k':

$s = (-u/3)^{(1/2)}$
$t = -v/(2s^3)$
$k = \arccos(t)/3$

giving

$y_1 = 2s.\cos(k) - p/3$
$y_2 = s(-\cos(k) + 3^{(1/2)}\sin(k)) - p/3$
$y_3 = s(-\cos(k) - 3^{(1/2)}\sin(k)) - p/3$

Note that if the discriminant 'j' is negative, then 'u' must also be negative, guaranteeing a real value for 's'. This value may be taken as positive without loss of generality. Also, 'k' will lie in the range 0 to 60 degrees, so that cos(k) and sin(k) are both positive. Thus we have:

$y_1 >= y_2 >= y_3$.

When the cubic is a subsidiary of a quartic, then either 'y_1' or 'y_3' may be the most useful root. For example, in Neumark's algorithm b = 2p , so although 'y_1' may be the largest root, it may not be positive. Then if 'b' and 'd' are both negative, it would be advantageous to use the most negative root 'y_3'.

NO REAL ROOTS

The absence of real roots to a quartic becomes apparent in some of the algorithms examined here after the subsidiary cubic has been solved, when the arguements of square roots of intermediate quantities are found to be negative. Alternatively, the extrema of the quartic can be found by solving the cubic:

$4x^3 + 3ax^2 + 2bx + c = 0$

and substituting each root back into the quartic to find its values at these extrema. If all these values are positive, then the quartic has no real roots.

More economical methods for discovering that there are no real roots may be available.

CONCLUSIONS

There have been many algorithms proposed for solving quartic equations, but most have been proposed with aims of elegance, generality or simplicity rather than error minimisation or overflow avoidance.

The operation counts of the best combination of stabilized algorithms for non-special cases are summarized in table 10:

	additions and subtractions	multiplications and divisions	functions e.g. root, sine	tests
cubic	best 8 worst 13	best 10 worst 15	best 2 worst 3	best 18 worst 19
rest of quartic	best 12 worst 16	best 20 worst 34	best 1 worst 2	best 27 worst 39
2 x quadratic	3	5	2	3
totals	best 26 worst 35	best 40 worst 59	best 7 worst 9	best 51 worst 64

Table 10

Operation counts for a best combination of stabilized algorithms.

A computer program was written to perform a comparison of the stabilities of the five algorithms for the solution of quartic equations. Quartics were examined which had all combinations and permutations of coefficients from the set:

$10^8, 10^4, 1, 10^{-4}, 10^{-8}, -10^8, -10^4, -1, -10^{-4}, -10^{-8}$

Of the 10,000 equations, all five algorithms agreed on the number of real roots in only 5,947 cases. For these quartics, all five agreed that 726 had no real roots. Of the remaining equations, Neumark's algorithm had the least worst error in 2,080 quartics, Ferrari's in 863, Yacoub & Fraidenraich's 544, Descartes' 42, and Christianson's 13. For the other quartics, there were problems in trying to compare the algorithms.

These occurred when the subsidiary cubic produced 3 roots, and the use of these different roots produced a different number of roots for the quartic. This problem was apparent for all the algorithms in a significant number of equations. It is unclear how to compare their accuracies in these situations. The statistics of these findings are presented in Table 11.

algorithm	number out of 10,000 quartics that have 3 roots to susidiary cubic	number where different cubic roots produce different numbers of quartic roots
Christianson	4014	1927
Descartes	2430	288
Ferrari	2886	224
Neumark	2862	302
Yacoub	3884	1142

Table 11

Comparison of cases where different cubic roots produced different numbers of quartic roots in a benchmark test of 10,000 quartics.

A check on the accuracy of the roots can be done at the cost of more computation. Each root may be substituted back into the original equation and the residual calculated. This can then be substituted into the derivative to give an estimate of the error of the root or used for a Reguli-Falsi or better still a Newton-Raphson correction.

A further comment may be useful here concerning the language used to implement these algorithm. Compilers for the language C often perform double precision operations on single precision variables ('float'), converting back to single precision for storage. Thus there might be little speed advantage in using 'float' variables compared with using 'double' for these algorithms. Fortran compilers may not do this. For example, using a VAX8600, the time taken to solve the 10,000 different quartics was 6 seconds, for Fortran single precision (using f77), 15 seconds for C single precision (using cc), and 16 seconds for C using double precision.

It may have been observed that the 2 stable cases for the computation of the cubic coefficients for Ferrari's algorithm are different from those of Descartes' algorithm, so that the cubic coefficients may be computed in a stable fashion for 4 of the possible 16 sign combinations of the quartic coefficients. Further work may be able to improve this situation. It would be good to find more algorithms or variations on the five listed here which allowed other combinations of coefficient signs to be handled in a stable fashion. At present it is unclear how Schmakov's transformations can be used to find a stable algorithm for the solution of a quartic involving a particular set of coefficients.

ACKNOWLEDGEMENTS

Thanks are due to the staff of the School of Information Technologies at the University of Sydney (Australia), the Department of Computer Science at the University of Waterloo (Canada), and the Department of Software Engineering at the University of Technology, Sydney (Australia), where much of this work was done. Thanks are also due to Charles Prineas on whose initial investigations this work was based. Thanks are also due to the late Alan Tritter for discussions, Zoe Kaszas for the initial preparation of this paper, Alan Paeth for initially formatting the paper for publication, and to Professor John Bennett for his continual encouragement.

REFERENCES

Badler, N.I. & Smoliar, S.W., **Digital Representations of Human Movement**, *ACM Computing Surveys*, Vol. 11, No. 1, pp. 24-27 (1979)

Brown, K.S., **Reducing Quartics to Cubics**, http://mathpages.com/home/kmath296.htm (1967)

Cardano, G., **Ars Magna**, Nurmberg (1545)

Christianson, B., **Solving Quartics Using Palindromes**, *Mathematical Gazette*, Vol. 75, pp. 327-328 (1991)

Conte, S.D. & de Boor, C., **Elementary Numerical Analysis**, McGraw-Hill, p. 117 (1980)

Herbison-Evans, D., **Real Time Animation of Human Figure Drawings with Hidden Lines Omitted**, *I.E.E. Computer Graphics and Applications*, Vol. 2, No. 9, pp. 27-33 (1982)

Herbison-Evans, D., **Caterpillars and the Inaccurate Solution of Cubic and Quartic Equations**, *Australian Computer Science Communications*, Vol. 5, No. 1, pp. 80-9-(1983)

Herbison-Evans, D., **Solving Quartics and Cubics for Graphics**, *Graphics Gems V* (ed.: A. Paeth) Academic Press, Chesnut Hill, pp. 3-15 (1995).

Littlewood, D.E., **A University Algebra**, Heineman, London, p. 173 (1950)

Nonweiler, T.R.F., **Roots of Low Order Polynomial Equations**, *Collected Algorithms of the ACM*, (C2), Algorithm 326 (1967)

Neumark, S., **Solution of Cubic and Quartic Equations**, Pergamon Press, Oxford (1965)

Ralston, A., **A First Course in Numerical Analysis**, McGraw-Hill, p. 351 (1965)

Schmakov, S. I., **A Universal Method of Solving Quartic Equations**, *International Journal of Pure and Applied Mathematics*, Vol. 71, No. 2, pp. 251-260 (2011)

Strong, T., **Elementary and Higher Algebra**, Pratt and Oakley, p. 469 (1859)

Turnbull, H.W., **Theory of Equations**, Oliver and Boyd, London, Fourth Edition, p. 130 (1947)

Wilkinson, J.J., **Rounding Errors in Algebraic Processes**, Prentice-Hall, p. 17 (1963)

Yacoub, M. D. & Fraidenraich G., **A Solution to the Quartic Equation**, Mathematical Gazette (2012).

(updated 25 July 2012)