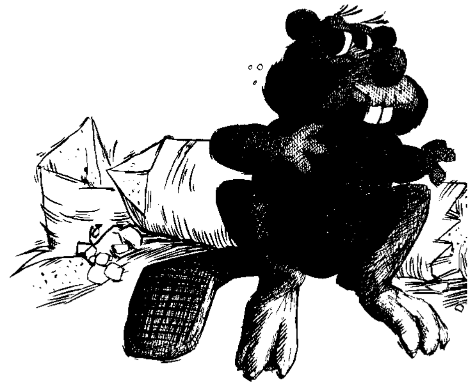
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT



Solving Quartics and Cubics for Graphics

Research Report

Don Herbison-Evans

CS-86-56

November, 1986

Solving Quartics and Cubics for Graphics

Don Herbison-Evans

Department of Computer Science
University of Waterloo

and

Basser Department of Computer Science
University of Sydney

ABSTRACT

A number of problems in computer graphics reduce to finding approximate real roots of quartic and cubic equations in one unknown. Various solution techniques are discussed briefly. The algorithms for analytic solution are discussed at length.

Methods are presented for controlling round-off error and overflow in the analytic solution of such equations. The solution of the quartic requires the solution of a subsidiary cubic equation. The use of the cubic derived by Neumark is shown to be the most stable of the techniques published for solving quartics. An algorithm based on this is derived which gives maximum stability.

An operation count of resulting algorithm is presented. Films have been made of animation computed in single precision floating point on a PDP/11/34 demonstrating the effects of before and after the use of the techniques described in the paper.

Introduction

This article addresses the problems of solving quartic and cubic equations in computer graphics.

Quartic equations need to be solved when ray tracing 4th degree surfaces e.g., a torus. A number of problems in computer graphics which involve the use of cubic splines require the solution of cubic equations in computer graphics.

Quartics also need to be solved in a number of problems involving quadric surfaces. Quadric surfaces (e.g. ellipsoids, paraboloids, hyperboloids, cones) are useful in computer graphics for generating objects wih 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.

One simplifying feature of the computer graphics problem is that only the real roots (if there are any) are required. The full solution of the quartic in the complex domain (Nonweiler, 1967) is an unnecessary use of computing resources. Another simplication in the graphics problem is that displays have only a limited resolution, so that only a limited number of accurate digits in the solution to the quartic are required. A resolution of 1 in 1,000,000 should be achievable using single precision floating point (32 bit) arithmetic, which is more than adequate for most displays.

Iterative Techniques

Roots can be obtained by iterative techniques. These 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, $4n(n-1)$ roots may need to be stored between frames. A compromise is to store pointers to those pairs of quadrics which give no roots. This, incidently, can be used to halve the computation for these quadrics within a given frame, for if quadric a has no intersection with quadric b, then b will not intersect a.

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 cubic or quartic. The most well-known algorithm for finding the number of real roots, the Stürm sequence, involves approximately as much computation as solving the equations directly by radicals (11). 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 and cubics directly.

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 in the 16th centry (Cardano, 1545). A number of other algorithms have subsequently been published. Many use the idea of first solving a particular cubic equation, the coefficients of which are derived from those of the quartic. The root of the cubic is then used to factorize the quartic into quadratics, which may then be solved. The question arises: which algorithm is for best to use on a computer to finding the real roots to the limited accuracy needed in the computer graphics?

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 can be seen. 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.

The 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:

(a) Ferrari-Lagrange solution (Turnbull, 1947)

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

(b) Descartes-Euler Solution (Strong, 1859)

$$y^3 + \left(b - \frac{3}{8}a^2\right) y^2 + \left(\frac{a^2}{8} - \frac{ab}{2} + c\right) y + \left(a^2 \frac{b}{16} - \frac{3a^4}{256} - \frac{ac}{4} + d\right) = 0$$

(c) (Neumark, 1965)

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

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. There are also a number of trivial changes to the cubic caused by the following:

if	$y^3 + py^2 + qy + r = 0$		
then	$z^3 - pz^2 + qz - r = 0$	for	$z = -y$
and	$z^3 + 2p\,z^2 + 4q\,z + 8r = 0$	for	$z = 2y$

None of these changes affect the stability of the algorithms.

Of the three subsidiary cubics, that from Ferrari's algorithm has the least computation in the derivation of the coefficients of the cubic. This is important not only for speed, but because every addition or subtraction can cause round-off exaggeration by cancellation. For this reason, attempts were made initially to use Ferrari's method for finding quadric outline intersections (Herbison-Evans, 1983). Unfortunately, the coefficients of the subsequent quadratics depend on two intermediate quantities, e and f, where

$$e^2 = \frac{a^2}{4} - b - 4y$$

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

$$ef = 2ay + \frac{c}{2}$$

The signs of each of the quartic coefficients a,b,c,d and the cubic signs. Of these, only 14 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 18 cases, the most stable choices are unclear.

More success has been obtained in stablizing the algorithm of Neumark. In this, the coefficients of the quadratic equations are obtained via parameters g, G, h and H, where:

$$G,g = \frac{a \pm \sqrt{a^2-4y}}{2}$$
$$H,h = \frac{b-y}{2} \pm \frac{a(b-y)-2c}{2\sqrt{a^2-4y}}$$

Any cancellation due to the $\pm$ signs can be eliminated by writing:

$$G,g = g_1 \pm g_2; \quad \text{where} \quad g_1 = \frac{a}{2}; \quad g_2 = \frac{\sqrt{a^2-4y}}{2}$$

$$H,h = h_1 \pm h_2 \quad \text{where} \quad h_1 = \frac{b-y}{2}; \; h_2 = \frac{a\left(\frac{b-y}{2}\right)-c}{\sqrt{a^2-4y}}$$

and using the identities

$$G \cdot g = y$$
$$H \cdot h = d$$

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 as

$$g = y/G$$

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

$$G = y/g$$

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 alterative formulation:

$$h_2 = \frac{\sqrt{(b-y)^2-4d}}{2}$$

Furthermore

$$g_2 = \frac{a h_1 - c}{\sqrt{(b-y)^2 - 4d}}$$

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 and d are positive and b is negative, m should be used. Thus 18 of the 32 sign combinations give stable results with this algorithm. For other cases, the errors of each of these expressions can be assessed by summing the moduli of the addends:

$$e_m = b^2 + 2|by| + y^2 + 4|d|$$

$$e_n = a^2 + |4y|$$

Thus, if

$$|m \cdot e_n| > |n \cdot e_m|$$

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

The Cubic

Let the cubic equation be

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

The solution may be expressed (Littlewood, 1950) using:

$$u = q - p$$
$$v = r - \frac{pq}{3} + \frac{2p^3}{27}$$

and the discriminant:

$$w = 4\left(\frac{u}{3}\right)^3 + v^2$$

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

$$y = \sqrt[3]{\frac{w-v}{2}} - \frac{u}{3}\sqrt[3]{\frac{2}{w-v}} - \frac{p}{3}$$

In this case, the quartic has two real roots.

This formulation is suitable if v is negative. The calculation in this form can lose accuracy if v is positive. This problem can be overcome by the rationalization:

$$\frac{w-v}{2} = \frac{w^2 - v^2}{2(w+v)} = \left(\frac{u}{3}\right)^3 \cdot \frac{2}{w+v}$$

giving the alternative formulation of the root:

$$y = \sqrt[3]{\frac{w+v}{2}} - \frac{u}{3}\sqrt[3]{\frac{2}{w+v}} \frac{-p}{3}$$

A computational problem with this algorithm is overflow while calculating w , for

$$0(w) = 0(p^6) + 0(q^3) + 0(r^2)$$

$$= 0(a^{12}) + 0(b^6) + 0(c^6) + 0\,(d^3)$$

Before evaluating the terms of w , it is useful to test the coefficients $a,b,c,d,p,q,s,$ against the appropriate root of the maximum number represented on the machine ('max'). The values of u and v should similarly be tested. In the event that some value is too large, various approximations can be employed: e.g.

if $\quad |p| > 27 \cdot \frac{\sqrt[3]{\max}}{2} \ : \ y \sim= -p,$

if $\quad |v| > \sqrt{\max} \ : \ y \sim= \sqrt[3]{v};$

if $\quad |u| > 3 \cdot \frac{\sqrt[3]{\max}}{4} \ : \ y \sim= u \cdot \frac{\sqrt[3]{4}}{3}.$

If the discriminant w is negative, then there are 3 real roots to the cubic, and either 0 or 4 real roots to the quartic equation. The roots to the cubic may then be obtained via parameters s, t and k :

$$s = \sqrt{-u/3}$$

$$t = -v/2s^3$$

$$k = \frac{1}{3}\, arccos\,(t)$$

giving

$$y_1 = s.cos\,(k) - p/3$$

$$y_2 = s(-\cos(k) + \sqrt{3}.\ \sin(k))/2 - p/3$$

$$y_3 = s\,(-\cos(k) - \sqrt{3}.\ \sin(k))/2 - p/3$$

Note that if the discriminant 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 $^\circ$, so that $\cos(k)$ and $\sin(k)$ are both positive.

The largest root of the cubic gives the most stable solution to the quartic if it and b are both positive.

Unfortunately, $b = p$ in Neumark's algorithm, so although y_1 is the largest root, it may not be positive. If b and d are both negative, it may be advantageous to use the most negative root: y_3 .

The functions *sine* and *cosine* of $arccos\,(t)/3$ may be tabulated to speed the calculation (Herbison-Evans, 1983) . Sufficient accuracy (1 *in* 10^{-7}) can be obtained with a table of 200 entries with linear interpolation, requiring 4 multiplications, 8 additions and 2 tests for each function. When t is near its extremes, the asymtotic forms may be useful:

$$\text{if} \quad t \sim= 1, \quad cos(k) \sim= (8 + t)/9$$

$$sin(k) \sim= \sqrt{2(1-t)/9}$$

$$\text{if} \quad t \sim= -1, \quad cos(k) \sim= ½ + \frac{\sqrt{t+1}}{6}$$

$$sin(k) \sim= \frac{\sqrt{3}}{2} - \frac{\sqrt{2(t+1)}}{6}$$

If the discriminant, w , is expanded in terms of the coefficients of the cubic, it has 10 terms. Two pairs of terms cancel and another pair coalesce, leaving 5 independent terms. In principle, any pair of subsets of these may cancel catastrophically, leaving not only an incorrect value but even an incorrect sign for the discriminant. This problem can be alleviated by calculating the 5 terms separately, and then combining them in increasing order of magnitude (Wilkinson, 1963). Better still, when solving quartics, the discriminant should be expanded in terms of the quartic coefficients directly. This gives fifteen terms, which can be sorted by modulus, and combined in increasing order.

Conclusion

There have been many algorithms proposed for solving quartic and cubic equations, but most have been proposed with aims of generality or simplicity rather than error minimisation or overflow avoidance. The work described here gives a low rate of error using single precision floating point arithmetic on a PDP11/34 for the computer animation of quadric surfaces.

The work may be summarized in the following table of operation counts:

	additions and subtractions	multiplications and divisions	square and cube roots	tests
cubic best	8	11	2	8
worst	24	30	2	14
quartic	18	27	1	12
quadratics (× 2)	6	8	2	8
total best	38	54	7	36
worst	54	73	7	42

A further comment may be useful here concerning the language used to implement these algorithms . Compilers for the language *C* often perform operations on single precision variables ('float') in double precision, converting back to single for storage. Thus there might be little speed advantage in using 'float' compared with using 'double' for these algorithms. Fortran compilers may not do this. Using a VAX8600, the time taken to solve 10,000 different quartics was 6.2 seconds, for Fortran single precision (using *f77*), 15.5 seconds for *C* single precision (using *cc*), and 16.1 seconds for *C* using double precision.

A further check on the accuracy 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 as a Newton-Raphson correction.

Acknowledgements

Thanks are due to the Departments of Computer Science at the Universities of Sydney (Australia) and Waterloo (Canada) where much of this work was done. Initial investigations on which the work was based were made by Charles Prineas. Thanks are also due to Alan Tritter for discussions and Zoë Kaszas for the preparation of this paper.

References

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

[2] Cardano, G., *Ars Magna*, Nurmberg (1545).

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

[4] 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).

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

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

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

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

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

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

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

[12] Conte,S.D., and de Boor, C.,
Elementary Numerical Analysis
McGraw-Hill, p.117 (1980)

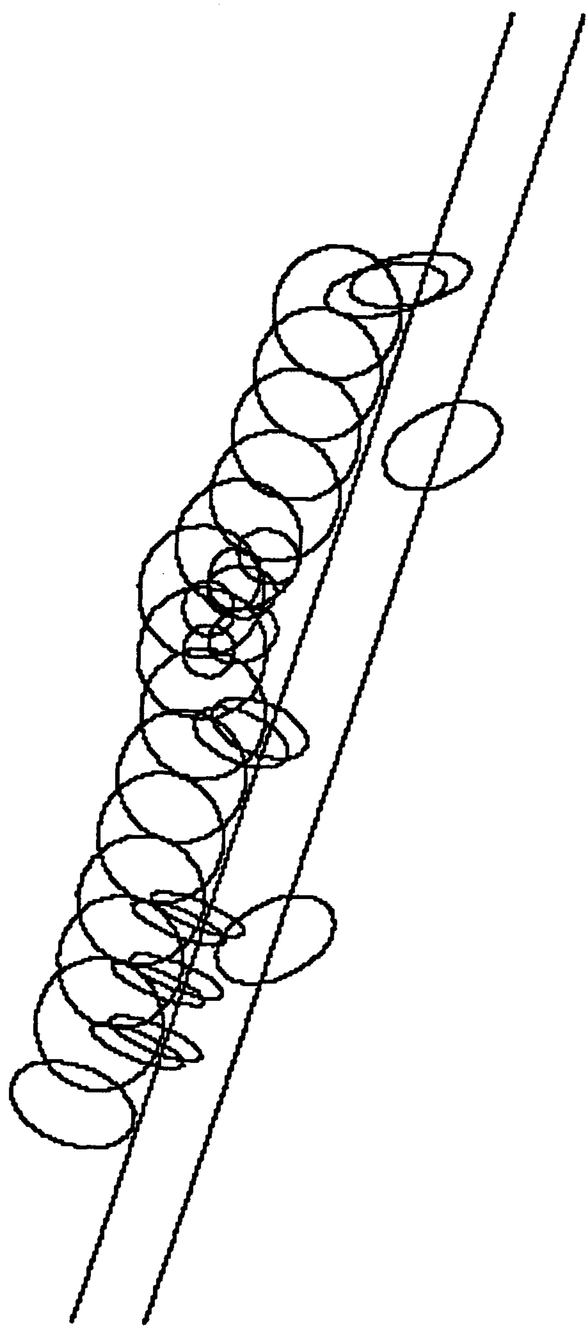


figure 1 A polyellipsoid figure

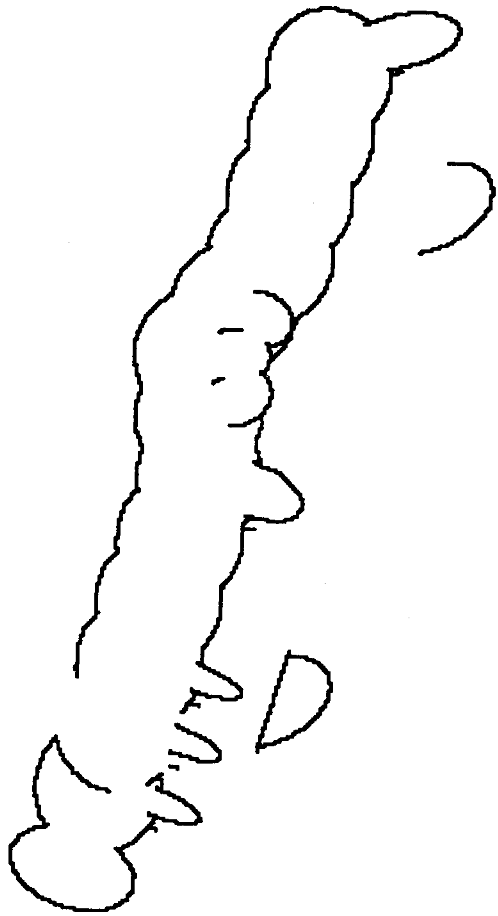


figure 2

A polyellipsoid figure with hidden arcs calculated using a simple quartic solver

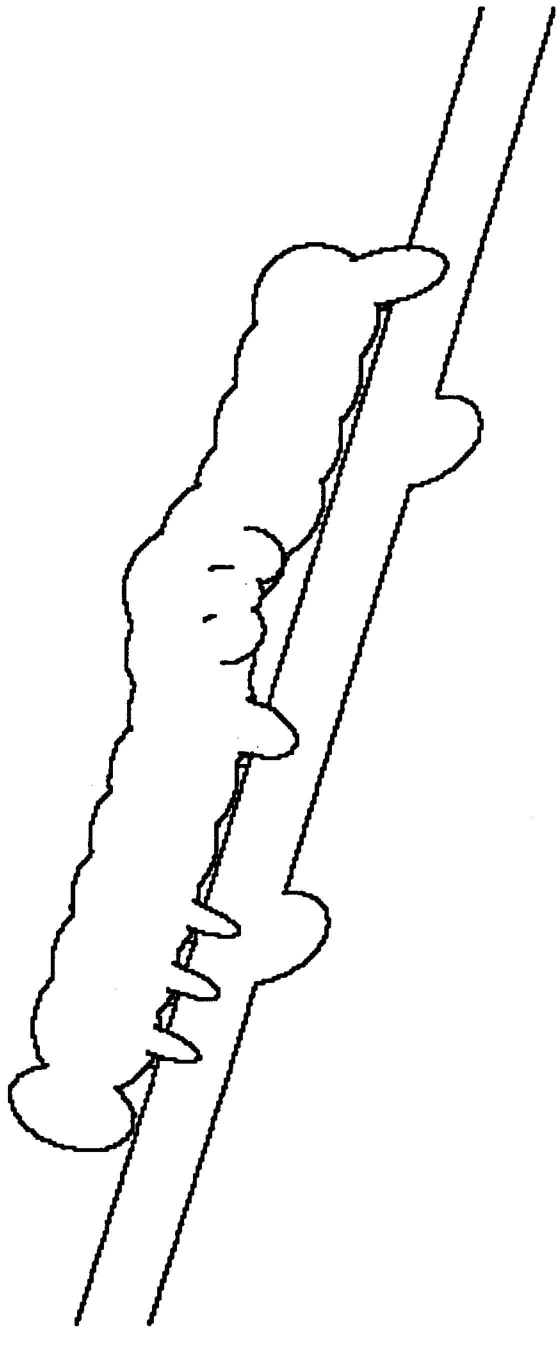


figure 3

A polyellipsoid figure with hidden arcs calculated using the quartic solver described in this article