CS 74

WHAT IS A SATISFACTORY QUADRATIC EQUATION SOLVER?

BY

GEORGE E. FORSYTHE

TECHNICAL REPORT NO. CS 74
AUGUST 7, 1967

COMPUTER SCIENCE DEPARTMENT
School of Humanities and Sciences
STANFORD UNIVERSITY

WHAT IS A SATISFACTORY QUADRATIC EQUATION SOLVER?

George E. Forsythe

Computer Science Department
Stanford University
Stanford, California 94305
USA

WHAT IS A SATISFACTORY QUADRATIC EQUATION SOLVER?

George E. Forsythe[1]

This Symposium has dealt with provable algorithms for finding zeros of general polynomials, with the tacit assumption that the processes would be implemented on an ideal computer system capable of exact arithmetic operations. In contrast, I should like to point out the near absence of algorithms to solve even a quadratic equation in a satisfactory way on actually used digital computer systems. The difficulties are partly caused by round-off error in floating-point arithmetic, but much more by the ever-present possibility of overflow or underflow (defined below).

This note presents specifications for a satisfactory quadratic equation solver suggested by Professor W. Kahan of the University of Toronto in lectures at Stanford University in 1966. The general level of performance is Kahan's, but the details are mine.

Consider the following set $F = F(\beta, t, m, M)$ of normalized floating-point numbers. This uses a number base β (bases 2, 8, 10, and 16 are in use) and a prescribed number t of significant digits. There are two limiting integer exponents m and M. The set F contains precisely

---

[1]This is a brief written contribution to Symposium on Constructive Aspects of the Fundamental Theorem of Algebra, held at the IBM Research Laboratories, Rueschlikon, Switzerland, 5-7 June 1967. The preparation of this note and the author's attendance at the Symposium were sponsored by the U. S. Office of Naval Research under project NR 044 211.

(1) $$1 + 2(\beta - 1)(M - m + 1)\beta^{t-1}$$

numbers. One of these is 0. Each other number in F has a unique representation

(2) $$\pm N \times \beta^{e},$$

where the sign is + or -, where the integer exponent e satisfies the inequality

(3) $$m \leq e \leq M,$$

and where the integer significand N satisfies the normalization condition

(4) $$\beta^{t-1} \leq N \leq \beta^{t} - 1 .$$

Frequently 0 is given the unique computer representation

(5) $$+ 0 \times \beta^{m} .$$

See [1] for a discussion of this floating-point number system in a slightly different notation.

We choose F^* to be a certain subset of F consisting of numbers not too close to overflow or underflow.

Definition. To be definite (and somewhat arbitrary), let $F^* = F^*(\beta, t, m, M)$ be the set consisting of 0 and all numbers (2)

subject to (4) and also to

(3*) $$m + 1 \leq e \leq M - 1 .$$

Definitions. A real number x is said to be in the range of F^* if either $x = 0$ or

(6) $$\beta^{t-1}\beta^{m+1} \leq |x| \leq (\beta^t - 1)\beta^{M-1} .$$

A complex number z is in the range of F^* if both $Re(z)$ and $Im(z)$ are real numbers in the range of F^*.

One similarly defines the expression in the range of F for real and complex numbers.

The statement (in computer jargon) that a real number y suffers from over- or underflow is equivalent to saying that y is not in the range of F.

In terms of these concepts I now give specifications in the form of a commented heading in Algol 60 of what I consider to be a satisfactory quadratic-equation solver for use with a processor of floating-point numbers in $F(\beta, t, m, M)$.

```
procedure QUADRATIC (a, b, c, x1, y1, x2, y2, error);
    value a, b, c; real a, b, c, x1, y1, x2, y2; switch error;
comment We are solving the equation az^2 + bz + c = 0, for arbitrary
    input parameters a, b, c in F*. Where values of the output
    parameters are not specified, they are irrelevant.
```

If $a = b = c = 0$, exit to error[1], since all complex numbers z satisfy the equation.

If $a = b = 0$ and $c \neq 0$, exit to error[2], since no z satisfies the equation.

Otherwise, let z_1 and z_2 be the exact roots of the equation, numbered so that $|z_1| \leq |z_2|$. (If $a = 0$, let $z_2 = \infty$.)

Whenever z_1 is in the range of F^*, set x1 and y1 to numbers in F close (defined below) to the real resp. imaginary part of z_1.

Whenever z_2 is in the range of F^*, set x2 and y2 to numbers in F close to the real resp. imaginary part of z_2.

Let $\zeta_1 = x1 + i \times (y1)$. We require that $\zeta_1 = 0$ (if $z_1 = 0$), and otherwise (again being somewhat arbitrary) that $|\zeta_1 - z_1| \leq \beta + 1$ units in the least-significant digit of the significand of the floating-point representation (2) of $\max(|x1|, |y1|)$. To repeat this requirement in symbols, if entier[x] denotes the greatest integer $\leq x$, we demand that

$$|\zeta_1 - z_1| \leq (\beta + 1) \times \beta^e, \tag{7}$$

where $e = \text{entier}[\log_\beta \max(|x1|, |y1|) - t]$.

We make a corresponding requirement of $\zeta_2 = x2 + i \times (y2)$.

If z_1 is not in the range of F, but z_2 is in the range of F^*, set x2, y2 as above and exit to error[3].

If z_2 is not in the range of F (including the case $z_2 = \infty$), but z_1 is in the range of F^*, set x1, y1 as above and exit to error[4].

If neither z_1 nor z_2 is in the range of F, exit to error[5] .

If a root z_i is in the range of F but not in the range of F^*, we permit either an indication of over- or underflow via the appropriate exit to error, or a determination of z_i with an accuracy satisfying (7) above.

The procedure QUADRATIC should make no unnecessary use of multiple-precision computation, but computation with 2t significant digits is essential at one part of the procedure, to achieve the accuracy (7). End of comment;

The main source of practical difficulty in writing the procedure QUADRATIC is the possibility of over- or underflow in many places. What is actually programmed depends crucially on what the computing system does in case of over- or underflow. An ideal system permits the user's program to regain control of the algorithm, if the user wishes, with the ability to interrogate Boolean variables to learn whether there has been overflow, underflow, or neither. Though such systems are rare, they have been implemented at Toronto [3] and at Stanford [2], and these systems make programming such an algorithm as QUADRATIC far more satisfactory.

Some systems merely dismiss a user's program in case of overflow. If a result underflows, many systems merely set the result to 0 and return to the user's program without any indication. Faced with systems like these, the programmer must take great pains to insure that over- or underflow can never occur. These precautions make a satisfactory algorithm tedious to write, lengthy to store, and slow to execute. One necessary subroutine must determine the exponents of a, b, c, and other real

numbers local to the procedure. This is probably most gracefully written in machine code.

The ability to achieve the prescribed accuracy (7) of $\beta + 1$ units in the last place depends on the detailed properties of the floating-point arithmetic processor. If the prescribed accuracy is not achievable, condition (7) must be relaxed as necessary. If necessary, one could also make the set F^* smaller by changing (3*).

As a simple illustration suppose that $\beta = 10$, $t = 4$, $m = -54$, $M = 45$. Then the smallest and largest positive numbers in F are

$$1000 \times 10^{-54} = 10^{-51}$$

and

$$9999 \times 10^{45} = .9999 \times 10^{49} = (1 - 10^{-4}) \times 10^{49} .$$

Here are some equations that may give trouble for this system:

(a) $10^{-40}z^2 - 5 \times 10^{-40}z + 6 \times 10^{-40} = 0$:

The roots are 2 and 3, and the only danger is that undetected underflows will introduce an error.

(b) $z^2 + 10^{10}z - 1 = 0$:

Use of the standard quadratic formula will yield 0 for the positive root, instead of the correctly rounded value 1.000×10^{-10}.

(c) $10^{-30}z^2 - 10^{40} + 1 = 0$:

The roots are near 10^{70} and 10^{-40}; use of the quadratic formula can easily cause overflow or underflow.

(d) $2.864\ z^2 - 2.864\ z + 0.7160 = 0$:

Here 0.5000 is a double root. Evaluating the quadratic formula

$$\frac{2.864 \pm \sqrt{(2.864)^2 - (4 \times 2.864) \times 0.7160}}{2 \times 2.864}$$

in single-precision rounded arithmetic yields $0.5000 \pm 0.05477\ i$, with an error of over 547 units in the last place of 0.5000.

It is not purely academic to make strict demands of the procedure QUADRATIC. Quadratic equations arising in the course of solving determinantal equations by Muller's method [6] or Laguerre's method [5], particularly in connection with large matrices, sometimes have one of the roots out of the range of F^*, and yet make essential use of the root in the range of F^*. The accuracy requirement (7) is perhaps overstrict for equations with nearly double roots.

I believe the specifications to be very reasonable for a basic process like solving a quadratic equation. Nevertheless, I venture to guess that not more than five quadratic solvers exist anywhere that meet the general level of the specifications. Kahan [4] has prepared an algorithm (in Fortran IV for the 7094-II under the Toronto version of the IBSYS operating system) which achieves the specifications for $\beta = 2$, $t = 27$, $m = -155$, $M = 100$. The error never exceeds $9/4$ $(= \beta + 1/4)$ units in the least-significant place of the root. The only multiple-precision operations occur in the computation of $\Delta = b_1^2 - 4a_1c_1$, followed by storage of a single-precision value of Δ. (Here a_1, b_1, c_1

are proportional to a, b, c .) The excess time for the double-precision computation is negligible in comparison with the time required to deal with over- and underflow.

It is obviously relevant to ask to what extent the various computer algorithms presented at this Symposium for general polynomials make provision for over- and underflow, and also what accuracy they achieve.

It is noteworthy that the programming of QUADRATIC depends crucially on the arithmetic properties of the computing system, especially on its behavior with over- and underflow. The practical numerical analyst with high standards is thus inextricably involved with the arithmetic behavior of his digital computer hardware and accompanying operating systems. Unfortunately, few numerical analysts have formulated their systems requirements explicitly, not to mention communicating them effectively to the persons who design hardware and software systems. With existing computing systems a numerical analyst faces a most disagreeable dilemma--either he writes less than satisfactory algorithms, or he undertakes the never-ending chore of writing basic software systems (and perhaps even rebuilds the arithmetic unit). Professor Kahan has generally taken the second alternative.

References

1. George E. Forsythe and Cleve B. Moler, COMPUTER SOLUTION OF LINEAR ALGEBRAIC SYSTEMS, Prentice-Hall, Englewood Cliffs, N. J., 1967; see Sec. 20.

2. Michael D. Green, "Coping with over/underflow on the B5500," multilithed typescript, Computer Science Dept., Stanford University, Stanford Calif. 94305, USA, 15 July 1966, c. 50 pp.

3. W. Kahan, "7094-II system support for numerical analysis," draft, Department of Computer Science, University of Toronto, Toronto, Canada, Aug. 1966.

4. W. Kahan, The FORTRAN IV subroutine QDRTC, computer library of McLennan Laboratories, University of Toronto, Toronto, Canada, c. 1966.

5. H. J. Maehly, "Zur iterativen Auflösung algebraischer Gleichungen," Z. Angew. Math. Phys., 5, 260-263 (1954).

6. David E. Muller, "A method for solving algebraic equations using an automatic computer." Math. Tables Other Aids Comput., 10, 208-215 (1956).

Computer Science Department
Stanford University
Stanford, California 94305
USA

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R&D
(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Computer Science Department Stanford University Stanford, California 94305	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP ----------

3. REPORT TITLE

WHAT IS A SATISFACTORY QUADRATIC EQUATION SOLVER?

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Manuscript for Publication (Technical Report)

5. AUTHOR(S) (Last name, first name, initial)
George E. Forsythe

6. REPORT DATE August 7, 1967	7a. TOTAL NO. OF PAGES nine	7b. NO. OF REFS six
8a. CONTRACT OR GRANT NO. Nonr-225(37) b. PROJECT NO. NR-044-211 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) CS 74 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) none	

10. AVAILABILITY/LIMITATION NOTICES
Releasable without limitations on dissemination.

11. SUPPLEMENTARY NOTES ---	12. SPONSORING MILITARY ACTIVITY Office of Naval Research Code 432 Washington, D. C. 20360

13. ABSTRACT

Following the suggestions of Professor W. Kahan, the author details precise requirements for a satisfactory computer program to solve a quadratic equation with floating-point coefficients. The principal practical roblem is coping with overflow and underflow.

DD FORM 1 JAN 64 1473

Unclassified
Security Classification

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
quadratic equation overflow underflow floating-point computation						

INSTRUCTIONS

1. **ORIGINATING ACTIVITY:** Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. **REPORT SECURITY CLASSIFICATION:** Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. **GROUP:** Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. **REPORT TITLE:** Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. **DESCRIPTIVE NOTES:** If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. **AUTHOR(S):** Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. **REPORT DATE:** Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7a. **TOTAL NUMBER OF PAGES:** The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. **NUMBER OF REFERENCES:** Enter the total number of references cited in the report.

8a. **CONTRACT OR GRANT NUMBER:** If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. **PROJECT NUMBER:** Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. **ORIGINATOR'S REPORT NUMBER(S):** Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. **OTHER REPORT NUMBER(S):** If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. **AVAILABILITY/LIMITATION NOTICES:** Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through _____."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through _____."

(5) "All distribution of this report is c.. .ed. Qualified DDC users shall request through _____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. **SUPPLEMENTARY NOTES:** Use for additional explanatory notes.

12. **SPONSORING MILITARY ACTIVITY:** Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. **ABSTRACT:** Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS), (S), (C), or (U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. **KEY WORDS:** Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, roles, and weights is optional.

DD FORM 1 JAN 64 1473 (BACK)

Unclassified

Security Classification