Remark on Algorithm 1010: Boosting efficiency in solving quartic equations with no compromise in accuracy.

CRISTIANO DE MICHELE, Dipartimento di Fisica, "Sapienza" Università di Roma, Italy

We present a correction and an improvement to Algorithm 1010 [A. Orellana and C. De Michele 2020]

CCS Concepts: • **Mathematics of computing → Solvers**; **Nonlinear equations**; **Computations on polynomials**.

Additional Key Words and Phrases: quartic equation, factorization into quadratics, Newton-Raphson scheme, numerical solver design, performance

This remark details a number of source changes that are required to the software package associated with [3]

(1) to fix bad estimates of roots for some rare polynomials
(2) to improve overall accuracy of roots of polynomials with random coefficients, which were used for the statistical analysis in [3, Section 3.2]

In the routine `oqs_quartic_solver` if d_2, as defined in [3, (65)], is close to 0, the LDL^t decomposition on which the present algorithm is based cannot be achieved and a different strategy is required as discussed in [3]. Assessing whether d_2 can be considered zero or not is a delicate point and it turned out that, for some polynomials, the original strategy wrongly identified $d_2 \neq 0$, which led to large errors in the estimated roots. For example, correct estimates of the roots of the following polynomial

$$x^4 + 2.2459773428819827\,x^3 - 14.480985471938862\,x^2 - 17.678187643398402\,x + 1.0 = 0.0 \tag{1}$$

are

$$-1.177171432473146, \quad 0.05418276103215475$$
$$-4.560692070110519, \quad 3.437703398669528$$

however the original codes provides the following bad estimates

$$2.299680318701711 - 1.98613863196097 \times 10^{-8} i$$
$$2.299680318701711 + 1.98613863196097 \times 10^{-8} i$$
$$-3.422668990142702 - 2.956017430098731 \times 10^{-8} i$$
$$-3.422668990142702 + 2.956017430098731 \times 10^{-8} i$$

This issue arises, since the condition for $d_2 \approx 0$ used in the original code (see [3, (65)]) is too stringent and the set of coefficients $\alpha_1, \beta_1, \alpha_2, \beta_2$ is erroneously calculated as if $d_2 \neq 0$. This condition was adapted from a termination

Author's address: Cristiano De Michele, Dipartimento di Fisica, "Sapienza" Università di Roma, P.le A. Moro, 2, Rome, I-00185, Italy, cristiano.demichele@uniroma1.it.

0098-3500/2020/1-ART1 $15.00
https://doi.org/10.1145/3386241

criterion for finding the zeros of a polynomials discussed in [1], where a less stringent criterion is also discussed. For our problem this amounts to using the condition

$$d_2 \leq \epsilon_m \left(|2b/3| + |\phi_0| + l_1^2\right) \tag{2}$$

instead of [3, (65)]. Note that, since this condition is less stringent, it is more likely that the code will check for the best set of coefficients α_1, β_1, α_2, β_2 (see the discussion on [3, page 10]). As a result of this fix, we have added the polynomial given by (1) to our accuracy tests as case 25.

In addition, an improvement has been implemented in the Newton-Raphson (NR) algorithm discussed in [3, Section 2.3] and implemented in the routine `oqs_NRabcd`. This improvement relies on the following observations:

(1) An improper termination criterion in NR may lead to large errors in the roots, if NR were to terminate too early without sufficiently improving the initial guess. The original termination criterion is based on the total relative error ϵ_t, of coefficients α_1, β_1, α_2, β_2, as defined in [3, (105)], i.e. if ϵ_t increases, then NR terminates. If this criterion is replaced with a one based on the increase of the absolute error

$$\epsilon_{abs} = \sum_i |F_i| \tag{3}$$

where F_i is the i-th component of vector $\mathbf{F}$ defined in [3, (104)], very large errors in the roots can be obtained, as in case 19 of accuracy tests discussed in [3, Section 3.1]. With this termination criterion large errors occur in the quartic roots, since NR terminates during initial iterations when the set of coefficients α_1, β_1, α_2, β_2 are far from the solution and the total absolute error is still large. Note also that if NR is able to overcome these initial steps, both absolute and relative total errors converge to values close to zero.

(2) On searching for the best set of coefficients α_1, β_1, α_2, β_2, if we attempt to minimize the total absolute error ϵ_{abs} (instead of the total relative error), more accurate roots of the quartic equations are found (see Fig. 1 below and the related discussion).

Hence, in the new NR algorithm, we decided to keep the original termination criterion based on the total relative error, but to choose the set of coefficients α_1, β_1, α_2, β_2 associated to the minimum absolute error ϵ_{abs} obtained over all NR iterations.

To implement this new algorithm, the one described in [3, Section 2.3] must be replaced with the following (where changes are underlined):

(1) Being $\mathbf{z} = (\alpha_1, \beta_1, \alpha_2, \beta_2)$
(2) Calculate the vector $\mathbf{F}$.
(3) Calculate the absolute error e_{abs}, as in Eq. (3), and total relative error e_t as in [3, (105)].
(4) set $e_{min} = e_{abs}$ and $\mathbf{z}_{min} = \mathbf{z}$.
(5) If $e_{abs} = 0$ terminate
(6) If $\mathbf{J} = \frac{\partial \mathbf{F}(\alpha_1,\beta_1,\alpha_2,\beta_2)}{\partial(\alpha_1,\beta_1,\alpha_2,\beta_2)}$, i.e. $\mathbf{J}$ is the Jacobian matrix of $\mathbf{F}$, calculate its inverse $\mathbf{J}^{-1}$ (see [3, (106)])
(7) If $\det(\mathbf{J}) = 0$ terminate.
(8) Store the actual value of e_t, i.e. set $e_o = e_t$
(9) Update $\mathbf{z}$ as follows

$$\mathbf{z} \leftarrow \mathbf{z} - \mathbf{J}^{-1}\mathbf{F} \tag{4}$$

(10) Calculate the new values of $\mathbf{F}$, e_t and e_{abs}.
(11) if $e_{abs} < e_{min}$, set $e_{min} = e_{abs}$ and $\mathbf{z}_{min} = \mathbf{z}$.
(12) If $e_{abs} = 0$, terminate.
(13) If $e_t > e_o$, terminate.
(14) Go to step 6.

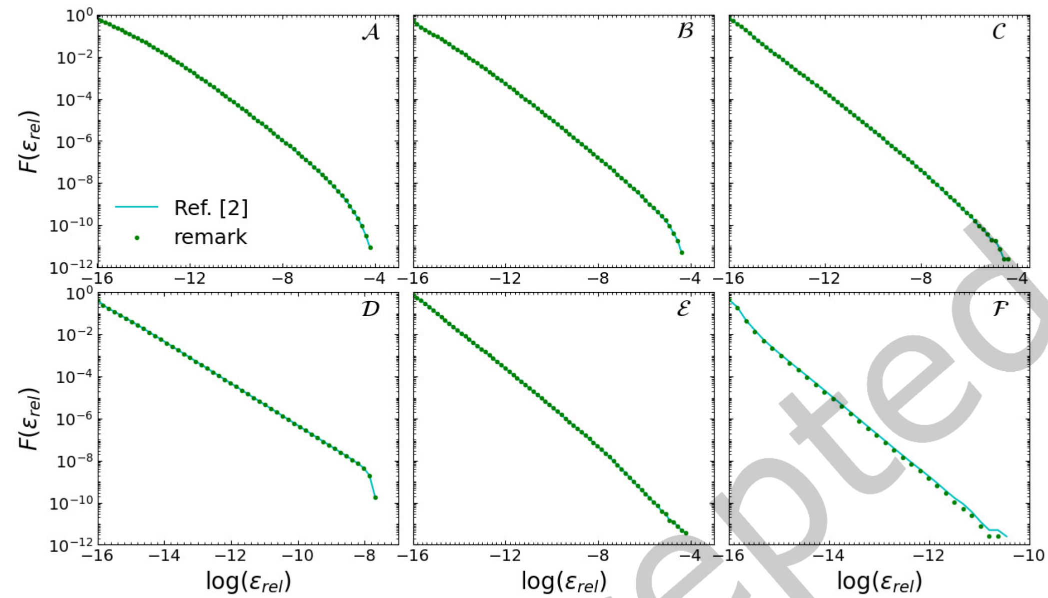

Fig. 1. Plot of the cumulative distribution function $F(\varepsilon_{rel})$ for samples $\mathcal{A} - \mathcal{F}$ where straight lines are results from original algorithm and symbols are ones from codes with changes discussed in the present remark paper.

We also increased the maximum number of possible iterations from 8 to 20. Accuracy tests as reported in [3, Table 1] are unaffected to all digits by the above changes and roots are identical in all cases. Similarly, the above changes do not impact on algorithm efficiency at all. In Fig. 1 we show the curves of the statistical analysis, as discussed in the original paper (see [3, Section 3.2]), obtained with both the original code and the new code. The present statistical analysis was carried out by generating a set of 2×10^{11} quartic polynomials for all samples. It can be seen that the cumulative probabilities in samples $\mathcal{A}$-$\mathcal{E}$ are almost identical to original ones, while for sample $\mathcal{F}$ the change implemented in the routine `oqs_abcdNR` provides a significant better accuracy, since in this case the old cumulative probability is up to twice as large than the new one. We finally note that identical changes have been applied to the version of the quartic solver for polynomials with complex coefficients.

In addition to sources and drivers which accompany the present paper, a C++ implementation of the algorithm is freely available (see [2]).

REFERENCES

[1] M. Igarashi. 1984. A Termination Criterion for Iterative Methods Used to Find the Zeros of Polynomials. *Math. Comp.* 42, 165 (1984), 165–171.

[2] C. De Michele. 2020. *quarticpp*. http://github.com/cridemichel/quarticpp

[3] Alberto Giacomo Orellana and Cristiano De Michele. 2020. Algorithm 1010: Boosting Efficiency in Solving Quartic Equations with No Compromise in Accuracy. *ACM Trans. Math. Softw.* 46, 2, Article 20 (May 2020), 28 pages. https://doi.org/10.1145/3386241