# NAG Fortran Library Chapter Introduction

# C02 – Zeros of Polynomials

## Contents

# 1 Scope of the Chapter

This chapter is concerned with computing the zeros of a polynomial with real or complex coefficients.

# 2 Background to the Problems

Let $f(z)$ be a polynomial of degree $n$ with complex coefficients $a_i$:

$$f(z) \equiv a_0 z^n + a_1 z^{n-1} + a_2 z^{n-2} + \ldots + a_{n-1} z + a_n, \quad a_0 \neq 0.$$

A complex number $z_1$ is called a **zero** of $f(z)$ (or equivalently a **root** of the **equation** $f(z) = 0$), if

$$f(z_1) = 0.$$

If $z_1$ is a zero, then $f(z)$ can be divided by a factor $(z - z_1)$:

$$f(z) = (z - z_1) f_1(z) \tag{1}$$

where $f_1(z)$ is a polynomial of degree $n - 1$. By the Fundamental Theorem of Algebra, a polynomial $f(z)$ always has a zero, and so the process of dividing out factors $(z - z_i)$ can be continued until we have a complete **factorization** of $f(z)$:

$$f(z) \equiv a_0 (z - z_1)(z - z_2) \ldots (z - z_n).$$

Here the complex numbers $z_1, z_2, \ldots, z_n$ are the zeros of $f(z)$; they may not all be distinct, so it is sometimes more convenient to write

$$f(z) \equiv a_0 (z - z_1)^{m_1} (z - z_2)^{m_2} \ldots (z - z_k)^{m_k}, \quad k \leq n,$$

with distinct zeros $z_1, z_2, \ldots, z_k$ and multiplicities $m_i \geq 1$. If $m_i = 1$, $z_i$ is called a **simple** or **isolated** zero; if $m_i > 1$, $z_i$ is called a **multiple** or **repeated** zero; a multiple zero is also a zero of the derivative of $f(z)$.

If the coefficients of $f(z)$ are all real, then the zeros of $f(z)$ are either real or else occur as pairs of conjugate complex numbers $x + iy$ and $x - iy$. A pair of complex conjugate zeros are the zeros of a quadratic factor of $f(z)$, $(z^2 + rz + s)$, with real coefficients $r$ and $s$.

Mathematicians are accustomed to thinking of polynomials as pleasantly simple functions to work with. However, the problem of numerically **computing** the zeros of an arbitrary polynomial is far from simple. A great variety of algorithms have been proposed, of which a number have been widely used in practice; for a fairly comprehensive survey, see Householder (1970). All general algorithms are iterative. Most converge to one zero at a time; the corresponding factor can then be divided out as in equation ;(1) of above – this process is called **deflation** or, loosely, dividing out the zero – and the algorithm can be applied again to the polynomial $f_1(z)$. A pair of complex conjugate zeros can be divided out together – this corresponds to dividing $f(z)$ by a quadratic factor.

Whatever the theoretical basis of the algorithm, a number of practical problems arise; for a thorough discussion of some of them see Peters and Wilkinson (1971) and Chapter 2 of Wilkinson (1963). The most elementary point is that, even if $z_1$ is mathematically an exact zero of $f(z)$, because of the fundamental limitations of computer arithmetic the **computed** value of $f(z_1)$ will not necessarily be exactly 0.0. In practice there is usually a small region of values of $z$ about the exact zero at which the computed value of $f(z)$ becomes swamped by rounding errors. Moreover, in many algorithms this inaccuracy in the computed value of $f(z)$ results in a similar inaccuracy in the computed step from one iterate to the next. This limits the precision with which any zero can be computed. Deflation is another potential cause of trouble, since, in the notation of equation ;(1) of , the computed coefficients of $f_1(z)$ will not be completely accurate, especially if $z_1$ is not an exact zero of $f(z)$; so the zeros of the computed $f_1(z)$ will deviate from the zeros of $f(z)$.

A zero is called **ill-conditioned** if it is sensitive to small changes in the coefficients of the polynomial. An ill-conditioned zero is likewise sensitive to the computational inaccuracies just mentioned. Conversely a zero is called **well-conditioned** if it is comparatively insensitive to such perturbations. Roughly speaking a zero which is well separated from other zeros is well-conditioned, while zeros which are close together are ill-conditioned, but in talking about ‘closeness’ the decisive factor is not the absolute distance between neighbouring zeros but their **ratio**: if the ratio is close to one the zeros are ill-conditioned. In particular, multiple zeros are ill-conditioned. A multiple zero is usually split into a cluster of zeros by perturbations in the polynomial or computational inaccuracies.

## 3 Recommendations on Choice and Use of Available Routines

**Note:** refer to the Users' Note for your implementation to check that a routine is available.

### 3.1 Discussion

Eight routines are available:

C02AFF for polynomials with complex coefficients,

C02AGF for polynomials with real coefficients,

C02AHF for quadratic ($n = 2$) equations with complex coefficients,

C02AJF for quadratic equations with real coefficients,

C02AKF for cubic ($n = 3$) equations with real coefficients,

C02ALF for cubic equations with complex coefficients,

C02AMF for quartic ($n = 4$) equations with real coefficients and

C02ANF for quartic equations with complex coefficients.

C02AFF and C02AGF both use a variant of Laguerre's Method to calculate each zero until the degree of the deflated polynomial is less than three, whereupon the remaining zeros are obtained by carefully evaluating the 'standard' closed formulae for a quadratic or linear ($n = 1$) equation.

For the solution of quadratic equations, C02AHF and C02AJF are simplified versions of the above routines.

C02AKF, C02ALF, C02AMF and C02ANF attempt to locate the zeros by finding the eigenvalues of the associated companion matrix using routines from LAPACK in Chapter F08.

The accuracy of the roots will depend on how ill-conditioned they are. Peters and Wilkinson (1971) and Thompson (1991) describe techniques for estimating the errors in the zeros after they have been computed.

## 4 Routines Withdrawn or Scheduled for Withdrawal

The following routines have been withdrawn. Advice on replacing calls to those withdrawn since Mark 13 is given in the document 'Advice on Replacement Calls for Withdrawn/Superseded Routines'.

| Withdrawn Routine | Mark of Withdrawal | Replacement Routine(s) |
|---|---|---|
| C02ADF | 15 | C02AFF |
| C02AEF | 16 | C02AGF |

## 5 References

Householder A S (1970) *The Numerical Treatment of a Single Nonlinear Equation* McGraw-Hill

Peters G and Wilkinson J H (1971) Practical problems arising in the solution of polynomial equations *J. Inst. Maths. Applics.* **8** 16–35

Thompson K W (1991) Error analysis for polynomial solvers *Fortran Journal (Volume 3)* **3** 10–13

Wilkinson J H (1963) *Rounding Errors in Algebraic Processes* HMSO