# Nonlinear differential algorithm to compute all the zeros of a generic polynomial

Francesco Calogero

Citation: Journal of Mathematical Physics **57**, 083508 (2016); doi: 10.1063/1.4960821

View online: http://dx.doi.org/10.1063/1.4960821

View Table of Contents: http://aip.scitation.org/toc/jmp/57/8

Published by the American Institute of Physics

JOURNAL OF MATHEMATICAL PHYSICS **57**, 083508 (2016)

# Nonlinear differential algorithm to compute all the zeros of a generic polynomial

Francesco Calogero
*Physics Department, University of Rome "La Sapienza," 00185 Rome, Italy and Istituto Nazionale di Fisica Nucleare, Sezione di Roma*

(Received 11 March 2016; accepted 1 August 2016; published online 16 August 2016)

A simple algorithm to compute all the zeros of a generic polynomial is proposed. *Published by AIP Publishing.* [http://dx.doi.org/10.1063/1.4960821]

## I. INTRODUCTION

*Notation.* Hereafter, for definiteness, we always refer to *monic* polynomials of arbitrary order $N$ $(N \geq 2)$,

$$P_N\left(z;\vec{c},\underline{x}\right) = z^N + \sum_{m=1}^{N}\left(c_m\, z^{N-m}\right) = \prod_{n=1}^{N}\left(z - x_n\right) \ , \tag{1}$$

the *complex* variable $z$ is the argument of the polynomial, indices such as $n$, $m$, $\ell$ run from 1 to $N$ (unless otherwise indicated, see below), the $N$-vector $\vec{c}$ has the $N$ *coefficients* $c_m$ of the polynomial (1) as its $N$ components, $\underline{x}$ is the *unordered* set of the $N$ *zeros* $x_n$ of the polynomial (1), and we assume all variables to be *complex* (unless otherwise explicitly indicated, see below). We call *generic* any polynomial the *coefficients* and *zeros* of which are *generic complex* numbers, and in particular feature *zeros* which are *all different among themselves,* $x_n \neq x_m$ if $n \neq m$. Note that the notation $P_N\left(z;\vec{c},\underline{x}\right)$ is somewhat redundant, since this monic polynomial can be identified by assigning *either* its $N$ coefficients *or* its $N$ zeros; indeed the $N$ coefficients $c_m$ can be expressed in terms of the $N$ zeros $x_n$ via the following standard formula:

$$c_m = (-1)^m \sum_{n_1>n_2>\cdots>n_m=1}^{N}\left(x_{n_1}\, x_{n_2}\cdots x_{n_m}\right) \ , \tag{2a}$$

so that

$$c_1 = -\left(x_1 + x_2 + \cdots + x_N\right), \tag{2b}$$

$$\begin{aligned} c_2 = {} & \left(x_1\, x_2 + x_1\, x_3 + \cdots + x_1\, x_N\right) \\ & + \left(x_2\, x_3 + x_2\, x_4 + \cdots + x_2\, x_N\right) + \cdots \\ & + \left(x_{N-2}\, x_{N-1} + x_{N-2}\, x_N\right) + x_{N-1}\, x_N \end{aligned} \tag{2c}$$

and so on. On the other hand, while the assignment of the $N$ *coefficients* $c_m$ determines the $N$ *zeros* $x_n$—uniquely, up to permutations—of course *explicit* formulas to accomplish generally this task *only* exist for $N \leq 4$. ■

The investigation of the properties—and of techniques for the numerical computation—of the $N$ *zeros* $x_n$ of a polynomial of degree $N$ defined via the assignment of its $N$ *coefficients* $c_m$ (see (1)) is a problem that has engaged mathematicians since time immemorial. In this paper, a simple nonlinear differential algorithm suitable to compute numerically *all* the $N$ zeros of a *generic* polynomial of arbitrary degree $N$ is described; I was unable to find a previous description of this algorithm in the literature, but I am aware that my search has not been—indeed, it could not have been—quite complete. This algorithm is described in Section II and proven in Section III.

## II. RESULTS

It is now convenient to introduce an additional independent variable $t$, which is hereafter assumed to be *real* and might be interpreted as *time*. Hence the above notation is now extended by writing, in addition to (1), the analogous formula,

$$p_N\left(z;\vec{\gamma}(t),\underline{y}(t)\right)=z^N+\sum_{m=1}^{N}\left[\gamma_m(t)\ z^{N-m}\right]=\prod_{n=1}^{N}\left[z-y_n(t)\right]\ , \tag{3}$$

to which notational comments quite analogous to those reported above apply.

There holds then the following.

*Proposition.* Consider the following system of $N$ nonlinear first-order differential equations satisfied by the $N$ *zeros* $y_n(t)$ of the polynomial (3),

$$\dot{y}_n(t)=-g(t)\ \left\{\prod_{\ell=1,\ \ell\neq n}^{N}\left[y_n(t)-y_\ell(t)\right]^{-1}\right\}\ \sum_{m=1}^{N}\left\{\left[c_m-\gamma_m(0)\right]\ \left[y_n(t)\right]^{N-m}\right\}\ , \tag{4a}$$

$$g(t)=\frac{\dot{f}(t)}{f(T)-f(0)}\ \ \text{implying}\int_0^T dt\ g(t)=1\ . \tag{4b}$$

Here and below a superimposed dot denote a $t$-differentiation, while the *coefficients* $c_m$ are those of the polynomial $P_N\left(z;\vec{c},\underline{x}\right)$, see (1), the *zeros* $x_n$ of which we seek, and $\gamma_m(0)$ are the $N$ *coefficients* of the polynomial $p_N(z;\vec{\gamma}(t),y(t))$, see (3), at $t=0$, hence they are related to the "initial" values $y_n(0)$ of the *zeros* of this polynomial by the formula (analogous to (2)),

$$\gamma_m(0)=(-1)^m\sum_{n_1>n_2>\cdots>n_m=1}^{N}\left[y_{n_1}(0)\ y_{n_2}(0)\cdots y_{n_m}(0)\right]\ . \tag{4c}$$

As for the function $f(t)$, and the *positive* parameter $T$, see (4b), they can both be assigned essentially *arbitrarily*; but of course so that the function $g(t)$ be finite for $0\leq t\leq T$ hence feature the property displayed by the second equality (4b).

Then

$$x_n=y_n(T)\ . \tag{4d}$$

■

It is thus seen that the *zeros* $x_n$ of the polynomial $P_N\left(z;\vec{c},\underline{x}\right)$, see (1), can be computed—once the $N$ coefficients $c_m$ of this polynomial have been assigned—via the following procedure. *Step one*: choose (*arbitrarily*!) $N$ *complex* numbers $y_n(0)$. *Step two*: compute, via the formulas (4c), the $N$ quantities $\gamma_m(0)$. *Step three*: integrate (numerically) the system of differential equations (4a) from $t=0$ to $t=T$, starting from the $N$ initial data $y_n(0)$, getting thereby the $N$ values $y_n(T)$, which give the sought result, see (4d).

Will this procedure always work? The only possible snag is that the solution $\vec{y}(t)$ of the "dynamical system" (4a) runs into a singularity during its evolution from $t=0$ to $t=T$. The only mechanism whereby this might occur is because during this evolution two different coordinates $y_n(t)$ might coincide, $y_\ell(t)=y_n(t)$ for $\ell\neq n$, at some value of the *real* variable $t$ in the interval $0<t<1$ causing the right-hand side of (4a) to blow up. This "collision" might indeed happen, but it is *not* a *generic* phenomenon; hence it will be enough to change the assignment of the (arbitrary!) initial data $y_n(0)$ to avoid this difficulty; note however that this suggests that to apply this method it will be advisable to always start with *complex* initial data $y_n(0)$, even in the case of *real* polynomials with *real* zeros. And note moreover that the numerical integration of the differential equations (4a) with different initial data $\vec{y}(0)$ and different assignments of the function $f(t)$ and of the parameter $T$—for instance

$$f(t)=t\ ,\quad T=1\ ,\quad \text{implying}\ g(t)=1\ , \tag{5a}$$

or

$$f(t)=a\,t^\lambda\ ,\ \lambda>0\ ,\quad T=1\ ,\quad \text{implying}\ g(t)=\lambda\,t^{\lambda-1}\ , \tag{5b}$$

or

$$f(t) = \frac{\exp(\lambda\, t) - 1}{\exp(\lambda\, T) - 1}\,, \ \lambda > 0\,, \quad \text{implying } g(t) = \frac{\lambda\, \exp(\lambda\, t)}{\exp(\lambda\, T) - 1} \tag{5c}$$

—allows to assess the *accuracy* of the computation, by comparing the results obtained starting from different assignments of these input data.

*Remark.* It is plain that this procedure will work more efficiently the closer the, arbitrarily chosen, initial values $y_n(0)$ are to the $N$ *zeros* $x_n$ the values of which one is trying to compute; indeed if the $N$ initial values $y_n(0)$ happened to *coincide* with the $N$ zeros $x_n$, $y_n(0) = x_n$, this would imply $\gamma_m(0) = c_m$ (compare (2) with (4c)) hence the right-hand side of the differential equations (4a) would vanish identically, entailing $\dot{y}_n = 0$ hence $y_n(1) = y_n(0) = x_n$, consistently with (4d). This suggests that it might be convenient to employ this technique *iteratively*, using the output $y(T)$ of each application of it as input $y(0)$ for a subsequent application.

Let us also emphasize that the dependence (via (4c)) of the right-hand sides of the differential equations (4a) upon the initial values $y_n(0)$ of the dependent variables $y_n(t)$ implies that these differential equations are rather differential functional equations than ordinary differential equations; but this fact has hardly any relevance on *step three* of the procedure, see above. ■

Preliminary computations performed by younger colleagues confirm these findings, but a comparison of the actual effectiveness of this technique with that of other methods to compute *all* the $N$ zeros of a *generic* polynomial of arbitrary degree $N$ is beyond the scope of this short communication, and in any case it is a task to be rather pursued by specialists in numerical analysis if they consider it worthy of their attention.

## III. PROOF

The proof of the above *Proposition* is actually quite easy (raising thereby some doubts on the novelty of this finding). The starting point is the *identity*,

$$\dot{y}_n(t) = -\left\{ \prod_{\ell=1,\ \ell\neq n}^{N} \left[y_n(t) - y_\ell(t)\right]^{-1} \right\} \sum_{m=1}^{N} \left\{ \dot{\gamma}_m(t)\ \left[y_n(t)\right]^{N-m} \right\}\,, \tag{6}$$

valid for any $t$-dependent polynomial with *zeros* $y_n(t)$ and *coefficients* $\gamma_m(t)$, see (3); for a proof of this formula see (if need be) Ref. 1. Now make the assignment

$$\gamma_m(t) = \gamma_m(0) + \left[\frac{f(t) - f(0)}{f(T) - f(0)}\right]\ \left[c_m - \gamma_m(0)\right] \tag{7a}$$

consistent with the initial assignment at $t = 0$ and clearly implying (see (4b))

$$\dot{\gamma}_m(t) = g(t)\ \left[c_m - \gamma_m(0)\right] \tag{7b}$$

and

$$\gamma_m(T) = c_m\,. \tag{7c}$$

The insertion of the first of these two formulas, (7b), in (6) yields (4a); while the second, (7c), implies that, at $t = T$, the polynomial $p_N\left(z; \vec{\gamma}(t), \underline{y}(t)\right)$, see (3), coincides with the polynomial $P_N\left(z; \vec{c}, \underline{x}\right)$, see (1), hence the validity of (4d). Q. E. D.

[1] F. Calogero, "New solvable variants of the goldfish many-body problem," Stud. Appl. Math. **137**, 123 (2015).