On computing zeros and poles of meromorphic functions[1]

P. Kravanja
M. Van Barel
A. Haegemans

Abstract. Given a meromorphic function f, we present an accurate numerical method that computes all the zeros and poles of f that lie inside a Jordan curve γ, together with their respective multiplicities and orders. An upper bound for the total number of poles of f that lie inside γ is assumed to be known. Our algorithm is based on numerical integration along γ and formal orthogonal polynomials. It uses the logarithmic derivative f'/f. Initial approximations for the zeros and poles are not needed. Numerical examples illustrate the effectiveness of our approach.

§1 Introduction

Let $\mathcal{P}$ be the linear space of polynomials. Let $\langle \cdot, \cdot \rangle : \mathcal{P} \times \mathcal{P} \to \mathbb{C}$ be any symmetric and bilinear form such that

$$\langle \phi, \psi \rangle = \sum_{k=1}^{m} \lambda_k \phi(x_k) \psi(x_k) \tag{1.1}$$

for any $\phi, \psi \in \mathcal{P}$, where $m \in \mathbb{N}_0$, $\lambda_1, \ldots, \lambda_m \in \mathbb{C}_0$ and $x_1, \ldots, x_m \in \mathbb{C}$ are unknown. Suppose that the points $x_1, \ldots, x_m$ are known to be mutually distinct. Assume that we have an 'oracle' at our disposal that provides us with the value of $\langle \phi, \psi \rangle$ for any $\phi, \psi \in \mathcal{P}$ upon simple request. Let an upper bound M for m be given. We will present an accurate numerical algorithm for computing the unknowns in (1.1). It is based on formal orthogonal polynomials and proceeds by evaluating $\langle \cdot, \cdot \rangle$ and by solving certain generalized eigenvalue problems.

Our interest in this problem is motivated by the following application.

Let W be a simply connected region in $\mathbb{C}$, $f : W \to \mathbb{C}$ meromorphic in W, and γ a positively oriented Jordan curve in W. Suppose that f has

[1]This research was partially supported by the Fund for Scientific Research-Flanders (FWO-V), project "Orthogonal Systems and their Applications," grant #G.0278.97.

neither zeros nor poles on γ. We consider the problem of computing *all* the zeros and poles of f that lie in the interior of γ, together with their respective multiplicities and orders. Let N denote the total number of zeros of f that lie inside γ (i.e., the number of zeros where each zero is counted according to its multiplicity), and let P denote the total number of poles of f that lie inside γ (i.e., the number of poles where each pole is counted according to its order). Suppose from now on that $N + P > 0$. Let n denote the number of mutually distinct zeros of f that lie inside γ. Let $z_1, \ldots, z_n$ be these zeros and $\nu_1, \ldots, \nu_n$ their respective multiplicities. Let p denote the number of mutually distinct poles of f that lie inside γ. Let $y_1, \ldots, y_p$ be these poles and $\mu_1, \ldots, \mu_p$ their respective orders. We define a symmetric and bilinear form $\langle \cdot, \cdot \rangle : \mathcal{P} \times \mathcal{P} \to \mathbb{C}$ by setting

$$\langle \phi, \psi \rangle := \frac{1}{2\pi i} \int_\gamma \phi(z)\psi(z)\frac{f'(z)}{f(z)}\, dz \tag{1.2}$$

for any two polynomials $\phi, \psi \in \mathcal{P}$. An easy calculation shows that $f'(z)/f(z)$ has a simple pole at z_k with residue ν_k for $k = 1, \ldots, n$ and a simple pole at y_l with residue $-\mu_l$ for $l = 1, \ldots, p$. It follows that

$$\langle \phi, \psi \rangle = \sum_{k=1}^{n} \nu_k \phi(z_k)\psi(z_k) - \sum_{l=1}^{p} \mu_l \phi(y_l)\psi(y_l).$$

The form $\langle \cdot, \cdot \rangle$ can be evaluated via numerical integration along γ. It is clearly of type (1.1). Our algorithm requires an upper bound M for $m = n + p$. Assume that an upper bound $\hat{P}$ for P is known, and define $s_r := \langle 1, z^r \rangle$ for $r = 0, 1, 2, \ldots$. Then $s_r = \nu_1 z_1^r + \cdots + \nu_n z_n^r - \mu_1 y_1^r - \cdots - \mu_p y_p^r$. In particular, $s_0 = N - P$ and we may assume that the value of s_0 has been computed. As $n+p \leq N+P = s_0+2P \leq s_0+2\hat{P}$, it follows that we may take $M = s_0+2\hat{P}$. In case γ is the unit circle, an upper bound for P can be obtained by using the heuristic approach of Gleyse and Kaliaguine [7].

For previous attempts to tackle this problem, we refer to Abd-Elall, Delves and Reid [1] and also Ioakimidis [9, 10].

In case $P = 0$, our problem reduces to that of computing *all* the zeros, together with their respective multiplicities, of an analytic function that lie inside a Jordan curve. Methods for the determination of zeros of analytic functions that are based on the numerical evaluation of integrals are called *quadrature methods*. A review of such methods is given in [11]. The integrals s_r are equal to the Newton sums of the unknown zeros, $s_r = \sum_{k=1}^{n} \nu_k z_k^r$, and thus the problem is transformed into that of solving a system of polynomial equations. This approach was taken by Delves and Lyness [5] and was reconsidered by Li [12]. What is wrong with these approaches, in our opinion, is that they are based on integrals involving polynomials from the standard

monomial basis (ordinary moments). The mapping between the data and the unknowns is usually ill-conditioned, especially in the case of clustered zeros. This leads to algorithms that may produce very inaccurate results. We will present an algorithm that gives very accurate results. Instead of the standard monomial basis, it uses the formal orthogonal polynomials associated with $\langle \cdot, \cdot \rangle$. The key to our success is the fact that we consider the zeros (and poles) and the multiplicities (and orders) as *separate* unknowns, an idea that was overlooked in the past. Note that our algorithm does not require initial approximations for the zeros and poles.

§2 Theoretical solution

Our unknowns m, $x_1, \ldots, x_m$ and $\lambda_1, \ldots, \lambda_m$ in (1.1) can be calculated by applying the form $\langle \cdot, \cdot \rangle$ to polynomials from an arbitrary basis for $\mathcal{P}$. Indeed, let ψ_k be a monic polynomial of degree k for $k = 0, 1, 2, \ldots$. Define the $k \times k$ matrices F_k and $F_k^{(1)}$ as

$$F_k := \left[\ \langle \psi_r, \psi_s \rangle \ \right]_{r,s=0}^{k-1} \qquad \text{and} \qquad F_k^{(1)} := \left[\ \langle \psi_r, \psi_1 \psi_s \rangle \ \right]_{r,s=0}^{k-1}$$

for $k = 1, 2, \ldots$.

Theorem 1. *$m = \operatorname{rank} F_{m+r}$ for every nonnegative integer r. In particular, $m = \operatorname{rank} F_M$.*

Proof: Let r be a nonnegative integer. The matrix F_{m+r} can be written as

$$\begin{aligned} F_{m+r} &= \sum_{k=1}^{m} \lambda_k \begin{bmatrix} \psi_0(x_k)\psi_0(x_k) & \cdots & \psi_0(x_k)\psi_{m+r-1}(x_k) \\ \vdots & & \vdots \\ \psi_{m+r-1}(x_k)\psi_0(x_k) & \cdots & \psi_{m+r-1}(x_k)\psi_{m+r-1}(x_k) \end{bmatrix} \\ &= \sum_{k=1}^{m} \lambda_k \begin{bmatrix} \psi_0(x_k) \\ \vdots \\ \psi_{m+r-1}(x_k) \end{bmatrix} \begin{bmatrix} \psi_0(x_k) & \cdots & \psi_{m+r-1}(x_k) \end{bmatrix}. \end{aligned}$$

This implies that $\operatorname{rank} F_{m+r} \leq m$. However, F_m is regular. Indeed, one can easily verify that F_m can be factorized as $F_m = V_m D_m V_m^T$ where V_m is the Vandermonde-like matrix $V_m := [\, \psi_r(x_s) \,]_{r=0,s=1}^{m-1,m}$ and D_m is the diagonal matrix $D_m := \operatorname{diag}(\lambda_1, \ldots, \lambda_m)$. Therefore $\operatorname{rank} F_{m+r} \geq m$. It follows that $\operatorname{rank} F_{m+r} = m$. ■

Theorem 2. *The eigenvalues of the pencil $F_m^{(1)} - \lambda F_m$ are $\psi_1(x_1), \ldots, \psi_1(x_m)$.*

Proof: Define V_m as the Vandermonde-like matrix $V_m := [\,\psi_r(x_s)\,]_{r=0,s=1}^{m-1,m}$ and let D_m and $D_m^{(1)}$ be the diagonal matrices $D_m := \mathrm{diag}(\lambda_1, \ldots, \lambda_m)$ and $D_m^{(1)} := \mathrm{diag}(\lambda_1\psi_1(x_1), \ldots, \lambda_m\psi_1(x_m))$. Then F_m and $F_m^{(1)}$ can be factorized as $F_m = V_m D_m V_m^T$ and $F_m^{(1)} = V_m D_m^{(1)} V_m^T$. Let $\lambda^\star$ be an eigenvalue of the pencil $F_m^{(1)} - \lambda F_m$ and x a corresponding eigenvector. Then

$$\begin{aligned} & F_m^{(1)} x = \lambda^\star F_m x \\ \Leftrightarrow \quad & (V_m D_m^{(1)} V_m^T)x = \lambda^\star (V_m D_m V_m^T)x \\ \Leftrightarrow \quad & D_m^{(1)} y = \lambda^\star D_m y \quad \text{if } y := V_m^T x \\ \Leftrightarrow \quad & \mathrm{diag}(\psi_1(x_1), \ldots, \psi_1(x_m))y = \lambda^\star y. \end{aligned}$$

This proves the theorem. ■

As $\psi_1(z)$ is a polynomial of degree one, $x_1, \ldots, x_m$ can be calculated by applying the previous theorem. Once m and $x_1, \ldots, x_m$ have been found, the problem becomes linear and $\lambda_1, \ldots, \lambda_m$ can be computed by solving a Vandermonde-like system of linear equations.

The question remains, of course, which polynomials $\psi_k(z)$ to choose. We will obtain very accurate numerical results by using the formal orthogonal polynomials associated with $\langle \cdot, \cdot \rangle$.

§3 Formal orthogonal polynomials

A monic polynomial φ_t of degree $t \geq 0$ that satisfies

$$\langle z^k, \varphi_t(z) \rangle = 0, \qquad k = 0, 1, \ldots, t-1, \tag{3.1}$$

is called a *formal orthogonal polynomial* (FOP) of degree t. (Observe that condition (3.1) is void for $t = 0$.) The adjective *formal* emphasizes the fact that, in general, the form $\langle \cdot, \cdot \rangle$ does not define a true inner product. An important consequence of this fact is that, in contrast to polynomials that are orthogonal with respect to a true inner product, FOPs φ_t need not exist or need not be unique for every degree t. (For details, see for example [8] and the references cited therein.) If (3.1) is satisfied and φ_t is unique, then φ_t is called a *regular* FOP and t a *regular index*. If we set

$$\varphi_t(z) =: u_{0,t} + u_{1,t} z + \cdots + u_{t-1,t} z^{t-1} + z^t$$

then condition (3.1) translates into the Yule-Walker system

$$\begin{bmatrix} s_0 & s_1 & \cdots & s_{t-1} \\ s_1 & & \iddots & \vdots \\ \vdots & \iddots & & \vdots \\ s_{t-1} & \cdots & \cdots & s_{2t-2} \end{bmatrix} \begin{bmatrix} u_{0,t} \\ u_{1,t} \\ \vdots \\ u_{t-1,t} \end{bmatrix} = - \begin{bmatrix} s_t \\ s_{t+1} \\ \vdots \\ s_{2t-1} \end{bmatrix} \tag{3.2}$$

where $s_r := \langle 1, z^r \rangle$ for $r = 0, 1, 2, \ldots$. Hence, the regular FOP of degree $t \geq 1$ exists if and only if the matrix $H_t := [s_{k+l}]_{k,l=0}^{t-1}$ is regular. Thus, the rank profile of $H := [s_{k+l}]_{k,l \geq 0}$ determines which regular FOPs exist.

If $s_0 = \langle 1, 1 \rangle = \sum_{k=1}^{m} \lambda_k \neq 0$, then the regular FOP of degree 1 exists and is given by $\varphi_1(z) = z - \mu$ where $\mu := s_1/s_0$. Theorem 1 implies that the regular FOP φ_m of degree m exists and tells us also that regular FOPs of degree larger than m do not exist. The polynomial φ_m is easily seen to be

$$\varphi_m(z) = (z - x_1) \cdot \cdots \cdot (z - x_m).$$

It is the monic polynomial of degree m that has $x_1, \ldots, x_m$ as simple zeros.

If H_m is strongly regular, i.e., if all its leading principal submatrices are regular, then we have a full set $\{\varphi_0, \varphi_1, \ldots, \varphi_m\}$ of regular FOPs.

What happens if H_m is not strongly regular? By filling up the gaps in the sequence of existing regular FOPs it is possible to define a sequence $\{\varphi_t\}_{t=0}^{\infty}$, with φ_t a monic polynomial of degree t, such that if these polynomials are grouped into blocks according to the sequence of regular indices, then polynomials belonging to different blocks are orthogonal with respect to $\langle \cdot, \cdot \rangle$. More precisely, define $\{\varphi_t\}_{t=0}^{\infty}$ as follows. If t is a regular index, then let φ_t be the regular FOP of degree t. Else define φ_t as $\varphi_r \psi_{t,r}$ where r is the largest regular index less than t and $\psi_{t,r}$ is an arbitrary monic polynomial of degree $t - r$. In the latter case φ_t is called an *inner polynomial*. If $\psi_{t,r}(z) = z^{t-r}$ then we say that φ_t is defined *by using the standard monomial basis*. These polynomials $\{\varphi_t\}_{t=0}^{\infty}$ can be grouped into blocks. Each block starts with a regular FOP and the remaining polynomials are inner polynomials. Note that the last block has infinite length.

The block orthogonality property is expressed by the fact that the Gram matrix $G_m := [\langle \varphi_r, \varphi_s \rangle]_{r,s=0}^{m-1}$ is block diagonal. The diagonal blocks are regular, symmetric and zero above the main antidiagonal. If all the inner polynomials in a certain block are defined by using the standard monomial basis, then the corresponding diagonal block has Hankel structure. The matrix $G_m^{(1)} := [\langle \varphi_r, \varphi_1 \varphi_s \rangle]_{r,s=0}^{m-1}$ is block tridiagonal. The diagonal blocks are symmetric and lower anti-Hessenberg (i.e., its entries are equal to zero along all the antidiagonals that lie above the main antidiagonal, except for the antidiagonal that precedes the main antidiagonal). Again, if all the inner polynomials in a certain block are defined by using the standard monomial basis, then the corresponding diagonal block is a Hankel matrix. The entries of the off-diagonal blocks are all equal to zero, except for the entry in the south-east corner. For proofs and further details, we refer to [3].

Corollary 1. *The eigenvalues of the pencil* $G_m^{(1)} - \lambda G_m$ *are* $\varphi_1(x_1), \ldots, \varphi_1(x_m)$. *In particular, if* $s_0 = \sum_{k=1}^{m} \lambda_k \neq 0$, *then they are given by* $x_1 - \mu, \ldots, x_m - \mu$ *where* $\mu := s_1/s_0$.

§4 The algorithm

If m and $\varphi_0, \varphi_1, \ldots, \varphi_{m-1}$ are known, then we can apply the previous theorem to obtain $x_1, \ldots, x_m$. By Theorem 1 the value of m can be computed as the rank of G_M, but this is not a very practical approach. Instead, we will start to compute the polynomials $\{\varphi_t\}_{t=0}^{\infty}$ one by one and determine the value of m as the degree of the last existing regular FOP. The upper bound M will play a crucial role in the computations.

Corollary 1 can be interpreted as follows: the zeros of the mth degree regular FOP can be calculated by solving a generalized eigenvalue problem. This property holds for *all* regular FOPs. This will enable us to compute regular FOPs in their product form, which is numerically very stable. Define $G_k := [\langle \varphi_r, \varphi_s \rangle]_{r,s=0}^{k-1}$ and $G_k^{(1)} := [\langle \varphi_r, \varphi_1 \varphi_s \rangle]_{r,s=0}^{k-1}$ for $k = 1, 2, \ldots$.

Theorem 3. *Let $t \geq 1$ be a regular index and let $z_{t,1}, \ldots, z_{t,t}$ be the zeros of the regular FOP φ_t. Then the eigenvalues of the pencil $G_t^{(1)} - \lambda G_t$ are given by $\varphi_1(z_{t,1}), \ldots, \varphi_1(z_{t,t})$. In particular, if $s_0 \neq 0$, then they are given by $z_{t,1} - \mu, \ldots, z_{t,t} - \mu$ where $\mu := s_1/s_0$.*

Proof: Define the Hankel matrix $H_t^<$ as $H_t^< := [s_{1+k+l}]_{k,l=0}^{t-1}$. We will first show that the zeros of φ_t are given by the eigenvalues of the pencil $H_t^< - \lambda H_t$. The zeros of φ_t are given by the eigenvalues of its companion matrix C_t. Let $\lambda^\star$ be an eigenvalue of C_t and x a corresponding eigenvector. As H_t is regular, we may conclude that $C_t x = \lambda^\star x \Leftrightarrow H_t C_t x = \lambda^\star H_t x$. Using (3.2) one can easily verify that $H_t C_t = H_t^<$.

Let A_t be the unit upper triangular matrix that contains the coefficients of $\varphi_0, \varphi_1, \ldots, \varphi_{t-1}$. Then G_t can be factorized as $G_t = A_t^T H_t A_t$. Suppose that $\varphi_1(z) =: z - \beta$. (If 1 is a regular index, then $\beta = s_1/s_0$.) Then $G_t^{(1)}$ is given by $[\langle \varphi_r, z\varphi_s \rangle]_{r,s=0}^{t-1} - \beta G_t$. The matrix $[\langle \varphi_r, z\varphi_s \rangle]_{r,s=0}^{t-1}$ can be written as $A_t^T H_t^< A_t$ and thus $G_t^{(1)} = A_t^T (H_t^< - \beta H_t) A_t$. Now let $\lambda^\star$ be an eigenvalue of the pencil $H_t^< - \lambda H_t$ and x a corresponding eigenvector. Then

$$
\begin{aligned}
& H_t^< x = \lambda^\star H_t x \\
\Leftrightarrow \quad & (H_t^< - \beta H_t) x = (\lambda^\star - \beta) H_t x \\
\Leftrightarrow \quad & A_t^T (H_t^< - \beta H_t) A_t y = \varphi_1(\lambda^\star) A_t^T H_t A_t y \quad \text{if } y := A_t^{-1} x \\
\Leftrightarrow \quad & G_t^{(1)} y = \varphi_1(\lambda^\star) G_t y.
\end{aligned}
$$

This proves the theorem. ■

Regular FOPs are characterized by the fact that the determinant of a Hankel matrix is different from zero, while inner polynomials correspond to

singular Hankel matrices. By using an explicit determinant expression for regular FOPs, one can show that $\langle \varphi_t, \varphi_t \rangle = \det H_{t+1} / \det H_t$. Therefore, if t is a regular index, then $t+1$ is a regular index if and only if $\langle \varphi_t, \varphi_t \rangle \neq 0$. However, from a numerical point of view a test "is equal to zero" does not make sense. Because of rounding errors in the evaluation of $\langle \cdot, \cdot \rangle$ we would encounter only regular FOPs. Strictly speaking one could say that inner polynomials are not needed in numerical calculations. However, the opposite is true! Let us call a regular FOP *well-conditioned* if its corresponding Yule-Walker system (3.2) is well-conditioned, and *ill-conditioned* otherwise. To obtain a numerically stable algorithm, it is crucial to generate only well-conditioned regular FOPs and to replace ill-conditioned regular FOPs by inner polynomials. Stable look-ahead solvers for linear systems of equations that have Hankel structure are based on this principle [6, 4, 2]. In this approach the diagonal blocks in G_m are taken (slightly) larger than strictly necessary to avoid ill-conditioned blocks. A disadvantage is that part of the structure of G_m and $G_m^{(1)}$ gets lost and that there will be some additional fill-in.

We will ask the user for two thresholds, ϵ_{stop} and ϵ_{cond} with $\epsilon_{\text{stop}} < \epsilon_{\text{cond}}$, to decide whether the algorithm may stop or not, and to determine the size of a block. Suppose that the algorithm has just generated a well-conditioned regular FOP φ_r. If $r = M$, then we may stop. Else we proceed to calculate $\langle \varphi_r, \varphi_r \rangle$. If $|\langle \varphi_r, \varphi_r \rangle| \geq \epsilon_{\text{cond}}$ then we generate φ_{r+1} as a regular FOP. Else we scan the sequence $\left(|\langle z^t \varphi_r, \varphi_r \rangle| \right)_{t=0}^{M-1-r}$. If $|\langle z^t \varphi_r, \varphi_r \rangle| < \epsilon_{\text{stop}}$ for $t = 0, 1, \ldots, M-1-r$ then we conclude that $m = r$ and stop. Else we search for the first element that is larger than ϵ_{cond}. The corresponding value of t then determines the size of the block of polynomials. If all the elements are less than ϵ_{cond} then we use the value of t that corresponds to the maximum to determine the block size and warn the user that we could not obtain the level of well-conditioning that he or she requested.

§5 Numerical examples

We implemented our algorithm in Matlab. The m-files are available from the first author. In the following examples the computations were done with floating point relative accuracy $\approx 2.2204\,10^{-16}$.

Example 1: Suppose that $m = 10$ and $x_k = k$, $\lambda_k = 1$ for $k = 1, \ldots, m$. This corresponds to the problem of computing the zeros of the Wilkinson-like polynomial $\prod_{k=1}^{m}(z-k)$. We evaluate the form (1.1) explicitely. By calculating the eigenvalues of the pencil $H_m^{<} - \lambda H_m$ (which corresponds to the classical approaches of [5] and [12]) we obtain the approximations

$$0.99997610636668, \quad 1.99812692712904, \quad 2.96964756470803,$$

3.85164221027260, 4.76929486601830, 5.83891586633208,
6.93895160469148, 7.98903115901926, 8.99931401487053, 9.99999170411583.

(The imaginary parts are not shown—they are $\mathcal{O}(10^{-9})$.) Our algorithm (with $M = 15$, $\epsilon_{\text{stop}} = 10^{-12}$ and $\epsilon_{\text{cond}} = 1$) generates only regular FOPs, concludes that $m = 10$ and obtains the (very accurate!) approximations

1.00000000000001, 2.00000000000000, 3.00000000000000,
4.00000000000000, 4.99999999999996, 5.99999999999998,
7.00000000000001, 8.00000000000000, 9.00000000000000, 9.99999999999999.

(The imaginary parts are $\mathcal{O}(10^{-15})$.) The mapping between the ordinary moments and the zeros is particularly ill-conditioned in the case of clustered zeros. However, also in this case our algorithm obtains comparatively very good results. Of course, the problem of computing zeros of polynomials is currently receiving a lot of attention and there exist specific algorithms that are better suited than our more general approach. See, e.g., the paper by Pan [13] for a recent review. ■

In the following examples the form (1.2) was evaluated via numerical integration. If γ is a circle, then (1.2) is the integral of a periodic function over a complete period, and therefore the trapezoidal rule is an appropriate quadrature rule.

To increase the accuracy of our algorithm, the approximations for the zeros or poles were refined iteratively via Newton's method,

$$z_k^{(\alpha+1)} = z_k^{(\alpha)} - \nu_k \frac{f(z_k^{(\alpha)})}{f'(z_k^{(\alpha)})}, \qquad y_l^{(\alpha+1)} = y_l^{(\alpha)} + \mu_l \frac{f(y_l^{(\alpha)})}{f'(y_l^{(\alpha)})}, \qquad \alpha \geq 0.$$

Example 2: Let $f(z) := e^{3z} + 2z\cos z - 1$ and let γ be the circle $\{z \in \mathbb{C} : |z| = 2\}$. Our algorithm (with $\hat{P} = 0$ and thus $M = s_0 = \langle 1, 1\rangle = 4$, $\epsilon_{\text{stop}} = 10^{-12}$ and $\epsilon_{\text{cond}} = 1$) finds that $N = 4$. It generates φ_1 as a regular FOP, defines φ_2 as an inner polynomial, generates φ_3 and φ_4 as regular FOPs and concludes that $m = n = 4$. The approximations for the zeros and the corresponding multiplicities are shown in the following table.

$-\underline{1.84423395326221} + i\,\underline{0.00000000000000}$	1
$\underline{0.53089493029293} + i\,\underline{1.33179187675112}$	1
$\underline{0.53089493029293} - i\,\underline{1.33179187675112}$	1
$\underline{0.00000000000000} + i\,\underline{0.00000000000000}$	1

The correct digits are underlined. The calculated multiplicities were at a distance of $\mathcal{O}(10^{-15})$ to integers. If we put $\epsilon_{\text{cond}} = 0.1$ then our algorithm

generates only regular FOPs. Now the approximations for the zeros are less accurate.

$-1.84423395326194 - i\ 0.00000000000499$	1
$0.53089493028783 + i\ 1.33179187675442$	1
$0.53089493029794 - i\ 1.33179187674729$	1
$-0.00000000000071 - i\ 0.00000000000211$	1

This shows the importance of generating only well-conditioned regular FOPs. The calculated multiplicities were at a distance of $\mathcal{O}(10^{-11})$ to integers. ■

Example 3: Let $f(z) := \dfrac{1}{z^2(z-1)(z^2+9)} + z\sin z + e^{-3z} + 4$ and let γ be the circle $\{z \in \mathbb{C} : |z| = 2\}$. We set the upper bound $\hat{P} := 5$. It turns out that $\langle 1, 1\rangle = 0$ (in other words, $N = P$) and thus $M = s_0 + 2\hat{P} = 10$. The approximations for the zeros and poles obtained by our algorithm (again with $\epsilon_{\text{stop}} = 10^{-12}$ and $\epsilon_{\text{cond}} = 1$) had about 12 correct significant digits. After one step of iterative refinement, we obtain the following results:

x_k	λ_k
0.97843635600921	1
0.16974891913248	1
−0.13327146070751	1
1.00000000000000	−1
0.00000000000000	−2

The computed values of the λ_k's were at a distance of $\mathcal{O}(10^{-12})$ to integers. Note that zeros and poles can be distinguished easily by checking the signs of the λ_k's. ■

References

[1] L. F. Abd-Elall, L. M. Delves, and J. D. Reid, *A numerical method for locating the zeros and poles of a meromorphic function*, Numerical Methods for Nonlinear Algebraic Equations (P. Rabinowitz, ed.), Gordon and Breach, 1970, pp. 47–59.

[2] A. W. Bojanczyk and G. Heinig, *A multi-step algorithm for Hankel matrices*, J. Complexity **10** (1994), no. 1, 142–164.

[3] A. Bultheel and M. Van Barel, *Linear algebra, rational approximation and orthogonal polynomials*, Studies in Computational Mathematics, vol. 6, North-Holland, 1997.

[4] S. Cabay and R. Meleshko, *A weakly stable algorithm for Padé approximants and the inversion of Hankel matrices*, SIAM J. Matrix Anal. Appl. **14** (1993), no. 3, 735–765.

[5] L. M. Delves and J. N. Lyness, *A numerical method for locating the zeros of an analytic function*, Math. Comput. **21** (1967), 543–560.

[6] R. W. Freund and H. Zha, *A look-ahead algorithm for the solution of general Hankel systems*, Numer. Math. **64** (1993), 295–321.

[7] B. Gleyse and V. Kaliaguine, *On algebraic computation of number of poles of meromorphic functions in the unit disk*, Nonlinear Numerical Methods and Rational Approximation II (A. Cuyt, ed.), Kluwer Academic Publishers, 1994, pp. 241–246.

[8] W. B. Gragg and M. H. Gutknecht, *Stable look-ahead versions of the Euclidean and Chebyshev algorithms*, Approximation and Computation: A Festschrift in Honor of Walter Gautschi (R. V. M. Zahar, ed.), Birkhäuser, 1994, pp. 231–260.

[9] N. I. Ioakimidis, *Application of the generalized Siewert-Burniston method to locating zeros and poles of meromorphic functions*, J. Appl. Math. Phys. **36** (1985), 733–742.

[10] ———, *Determination of poles of sectionally meromorphic functions*, J. Comput. Appl. Math. **15** (1986), 323–327.

[11] ———, *Quadrature methods for the determination of zeros of transcendental functions—a review*, Numerical Integration: Recent Developments, Software and Applications (P. Keast and G. Fairweather, eds.), Reidel, Dordrecht, The Netherlands, 1987, pp. 61–82.

[12] T.-Y. Li, *On locating all zeros of an analytic function within a bounded domain by a revised Delves/Lyness method*, SIAM J. Numer. Anal. **20** (1983), no. 4, 865–871.

[13] V. Y. Pan, *Solving a polynomial equation: some history and recent progress*, SIAM Review **39** (1997), no. 2, 187–220.

P. Kravanja
Katholieke Universiteit Leuven, Department of Computer Science
Celestijnenlaan 200 A, B-3001 Heverlee, Belgium
Peter.Kravanja@na-net.ornl.gov

M. Van Barel
Katholieke Universiteit Leuven, Department of Computer Science
Celestijnenlaan 200 A, B-3001 Heverlee, Belgium
Marc.VanBarel@cs.kuleuven.ac.be

A. Haegemans
Katholieke Universiteit Leuven, Department of Computer Science
Celestijnenlaan 200 A, B-3001 Heverlee, Belgium
Ann.Haegemans@cs.kuleuven.ac.be