Circuits Syst Signal Process
DOI 10.1007/s00034-014-9824-3

Fraction-Free Unit Circle Stability Tests

Yuval Bistritz

Received: 6 October 2013 / Revised: 15 May 2014 / Accepted: 16 May 2014

Abstract The paper considers five fraction-free (FF) tests to determine whether a complex or a real polynomial has all its zeros inside the unit-circle. A FF test for complex polynomial is applicable to both complex and real polynomials where the FF property means that when it is applied to Gaussian or real integer coefficients, it can be completed over the respective ring of integers, i.e., it is Gaussian integers preserving (GIP) for Gaussian integer coefficient polynomials and integer preserving (IP) for integer coefficient polynomials. Three of the FF tests are for both complex and real polynomials. Two of the FF tests are specific for real polynomials and are IP for a polynomial with integer coefficients. Two GIP tests and two corresponding IP tests are immittance-type (stem from the Bistritz test). The third GIP test, a modified test proposed by Jury, is scattering-type (stems from the Schur–Cohn test). The two real immittance-type IP tests are significantly more efficient as integer algorithms than using any of the GIP tests for a real integer polynomial. The focus of the paper is on stability conditions. However, assuming normal conditions, stability conditions are always embedded in rules for counting also zeros outside the unit-circle.

Keywords Unit-circle stability · Testing stability of discrete systems · The Schur–Cohn problem · Zero location of polynomials · Fraction free · Integer algorithms · Immittance algorithms

1 Introduction

Algebraic algorithms to test whether all the zeros of a polynomial reside inside the unit-circle are well known as a means to test the stability of a linear discrete-time

Y. Bistritz
School of Electrical Engineering, Tel Aviv University, 69978 Tel Aviv, Israel
e-mail: bistritz@eng.tau.ac.il

Published online: 10 June 2014 Birkhäuser system. They also serve as a tool in filters design and systems control and bear intimate relationships with some important problems in linear algebra and signal processing. The current paper considers the testing of stability by algorithms that are fraction free (FF). The FF property means that when the algorithm is applied to a polynomial with Gaussian (i.e., complex) or real integer polynomials, it can be completed over the same ring of integers. FF tests are not restricted to integer polynomials and we shall discuss some good reasons to prefer them also for not integer polynomials. The term integers preserving is often used as a synonym for FF in the literature. Our preference is to use FF as a common term and use Gaussian integer preserving (GIP) and integers preserving (IP) to distinguish complex and real FF algorithms. The FF property makes stability tests more suitable for symbolic computation, engineering design problems, and to test stability of higher dimensional systems, as will become apparent through citations and comments during their presentation. We shall return to discuss these and other advantages of the FF stability tests in the last section of the paper.

The first direct solution to the unit-circle stability problem was obtained by Schur [31] and Cohn [19]. The Schur–Cohn method was subsequently studied by many researchers with notable early contributions by Marden [29] and Jury [22,25]. Another and more efficient stability test was proposed by Bistritz in [3,4]. It introduced a different approach that has become called the *immittance* formulation (also "split" in related signal processing contexts) where, for distinction, the classical tests of Schur Cohn Marden and Jury are called the *scattering* formulation (these two terms follow from their possible interpretation as two different ways to describe wave propagation). Immittance-type tests associate the tested polynomial with a sequence of symmetric polynomials of descending degrees that are created by a three-term polynomial recursion. In difference, scattering-type tests use a two-term polynomial recursion and create a sequence of polynomials with no specific structure. The Bistritz test (BT) was first noted for its improved computational efficiency. It involves a count of (standard) arithmetic operations that is lower by a factor of two or more (depending on type) than the four possible types of scattering-type stability tests [6]. Subsequently, it became appreciated also for its close similarity to the Routh test, the well-known stability test for continuous-time linear systems. Jury and Mansour declared that the BT is the discrete equivalent of the Routh test [26]. It also has an inherent ability to accommodate smoothly abnormal cases that were previously treated as singularities [12]. The current paper will support with further evidence the affinity between the BT and the Routh test. The two tests share in common the situation where there is a Gaussian integer version and a simpler real integer version as shown here for the BT and was shown recently in [18] for the Routh test, a situation that has no parallel within the scattering-type tests.

A polynomial $D(z) = d_n \prod_{k=1}^{n}(z - z_k)$ will be called in this paper "stable" if all its zeros reside inside the unit-circle (IUC) , $|z_k| < 1$. This is the familiar requirement that the characteristic polynomial of a stable discrete-time linear system has to fulfill. A typical stability test consists of an algorithm and a stability theorem. The algorithm builds for $D(z)$, in a finite number (of $o(n^2)$) of arithmetic operations, a sequence of descending degree polynomials. The theorem provides necessary and sufficient conditions for stability posed typically on the sign of a set of n parameters taken from the built polynomial sequence. A (unit-circle) zero location (ZL) test generalizes the stability test into counting the numbers of outside the unit-circle (OUC) zeros, $|z_k| > 1$, unit-circle (UC) zeros, $|z_k| = 1$, and IUC zeros of $D(z)$. In this paper, we focus on FF stability tests. However, ZL rule for each FF test will also be provided, assuming normal conditions (that roughly means, as long as the algorithm used to test stability can still be completed "as is").

The paper considers five FF stability tests, outlined for the sake of orientation in Table 1. Three of them are applicable to both complex and real polynomials such that they are GIP for Gaussian integer polynomial and IP for real integer polynomials. One GIP test is the modified Jury test (MJT), a scattering-type test that Jury devised, first for real polynomials and later, in [27], for complex polynomials, in order to get directly the sequence of Schur–Cohn stability determinants. An early real version was shown to be IP already in [1]. The two other GIP tests are FF versions of the BT. One is the FFG ("FF for Gaussian integers") test that follows the original initiation of the BT in [3,4,12]. The second is the FFGM, test that follows the BT with the modified initiation [7] (the ending M stands for "modified"). The remaining two tests are the FFR ("FF for real integers") and the FFRM tests. They are IP tests usable only for real polynomials. The FFR and FFRM tests are more efficient integer algorithms than using the FFG or the FFGM (or the MJT) test for a real integer polynomial.

The FFG and the FFGM tests were conceived in the context of developing stability tests for two-dimensional (2D) discrete systems. The FFG test underlies the 2D tabular test in [9] and the FFGM underlies the one in [8]. The MJT was too applied successfully to test 2D stability [11,20]. However, only now they are presented and characterized as GIP stability tests. As it will be discussed later, the application of these tests to testing 2D stability demonstrates a generalized use that relies on the fact that the algorithms remain FF also for polynomials with coefficients over more general integral domains.

The paper states for each test the algorithm that produces a sequence of polynomials of descending degrees, characterizes its integer properties, and presents for it the ZL rule (for normal condition). The implied stability conditions are simplified, when possible, into conditions with shorter integers. Section 2 brings some terminology and introductory notes. Section 3 presents the FFG and the FFR tests. Section 4 brings the corresponding FFGM and FFRM tests. Section 5 brings the MJT. Section 6 compares the binary complexity of all the integer tests. The last section discusses the five FF test from several perspectives and comments on applications and some planned further work.

Table 1 A chart of the FF tests in this paper

Polynomial $D(z)$	Immittance tests	Immittance modified tests	Scattering test
$D(z) \in \mathbb{C}[z]$	FFG	FFGM	MJT
GIP for $D(z) \in \mathbb{Z}_j[z]$ IP for $D(z) \in \mathbb{Z}[z]$	$\hat{G}$-sequence Sect. 3.1 (also FFGr [a], Sect. 3.2.1)	$\tilde{G}$-sequence Sect. 4.1	C-sequence Sect. 5
$D(z) \in \mathbb{R}[z]$	FFR	FFRM	No simpler
IP for $D(z) \in \mathbb{Z}[z]$	R-sequence Sect. 3.22	$\tilde{R}$-sequence Sect. 4.2	IP specific test

[a] The FFGr algorithm (G-sequence) follows the FFG recursion but admits a minor reduction for $D(z) \in \mathbb{Z}_j[z]$ if $D(1) \in \mathbb{R}$

2 Preliminaries

Let $\mathbb{C}$ and $\mathbb{R}$ present the complex and the real numbers, respectively. We shall consider the testing of the following complex polynomial

$$D(z) = \sum_{i=0}^{n} d_i z^i \in \mathbb{C}[z],\ D(1) \neq 0, d_n \neq 0, \tag{1}$$

or, when dealing specifically with a real polynomial, we shall assume

$$D(z) = \sum_{k=0}^{n} d_k z^k \in \mathbb{R}[z]\ ,\ D(1) > 0,\ d_n \neq 0. \tag{2}$$

Note that the assumptions are broad enough for stability testing. If $D(1) = 0$, then the polynomial is not stable (has a UC zero at $z = 1$). Similarly, a polynomials of degree n that is degree deficient ($d_n = 0$) represents too a not stable case (vanishing leading coefficients mean OUC zeros at infinity). Fixing $D(1) \neq 0$ into $D(1) > 0$ in the real case (2) often admits "nicer" expressions for the stability conditions. The same "nice stability expression" would be reached also if the pair $D(1) > 0$ and $d_n \neq 0$ is interchanged with the pair $d_n > 0$ and $D(1) \neq 0$ (as can be verified by setting $z = 1$ into a real polynomial $D(z) = d_n \prod_{k=1}^{n}(z - z_k)$ with $|z_k| < 1$ for all its zeros). However, the choice $D(1) > 0$ is more coherent with the central role of $z = 1$ in the immittance-type algorithms.

Let $\mathbb{Z}$ denote the ring of real integers, and $\mathbb{Z}_\mathrm{j}$ the ring of Gaussian integers defined by $\mathbb{Z}_\mathrm{j} = \{a + \mathrm{j}b \mid a, b \in \mathbb{Z}\}$, $\mathrm{j} = \sqrt{-1}$. We recall that we refer to an algorithm that remains FF for $D(z) \in \mathbb{Z}_\mathrm{j}[z]$ or for $D(z) \in \mathbb{Z}[z]$, Gaussian integer preserving (GIP) and integer preserving (IP), respectively. All the forthcoming GIP algorithms are arranged such that they become IP for real integer polynomials. However, we shall have two IP stability tests that can be used only for real polynomials.

A naive way to turn a stability test into an integer algorithm is to avoid divisions. For example, the Marden test [29] and related tests listed in the "type D" category in the classification of the scattering-type tests in [6] are actually integer tests because they are division free. The problem with this approach is that it produces integers whose size increases at an exponential rate from step to step making them very inefficient integer tests (the terms "size of integers" and "efficiency of integers" will be explained in a moment). The term FF algorithm infers efficiency that is manifested by a restrained (linear rather than exponential) growth of the size of integers. Division-free tests often provide an instrumental tool toward developing an efficient integer test. We used a division-free intermediate stage in the derivation of the FFR IP test in [15] and beforehand in the derivation of the 2D stability tests in [9] and [8] (that underlie the current FFG and FFGM tests). The FF integer tests considered in this paper are not division-free. They use division to remove recursively a common integer factors from the coefficients of the produced sequence of polynomials and in this way restrain the growth of the size of integers.

By the size of an integer b ($\in \mathbb{Z}$ or $\in \mathbb{Z}_{\mathrm{j}}$) we mean a measure, denoted by $\ell(b)$, that is postulated to have two properties (i) it is additive when two integers are multiplied, $\ell(b_1b_2) = l(b_1)+l(b_2)$ and (ii) $\ell(b_1+b_2) = max\{\ell(b_1), \ell(b_2)\}$. It is possible to think of $\ell(b)$ as presenting the number of bits (or $Log_2(|b|)$) or the number of decimal digits (or $Log_{10}(|b|)$) at least approximately (note that these interpretations do not obey property (ii) exactly). We shall denote by $\ell(D(z))$ the *integer-size* of a polynomial $D(z)$ that can be defined as (say) the maximal size of its integer coefficients. The efficiency of an integer algorithm takes into account not just the structure of the algorithm (that affects the conventional measure of efficiency based on counting the number of standard arithmetic operations) but also the size of the involved integers (multiplying *big* integers is more expensive), using a measure that is called *computing time* [1] or *binary complexity* [2]. We shall state the integer size as part of the presentation of each algorithm and their implied respective binary complexity in Sect. 6.

As mentioned already, the immittance-type GIP tests in this paper were conceived during the development of efficient 2D stability tests and the MJT was too applied successfully to 2D stability testing. A brief digression on these applications provides revealing insight on the power of FF tests. Stability of 2D systems poses a “no zeros outside the unit bi-circle” requirement on a bivariate polynomial. Due to some helpful auxiliary theorems, the latter requirement can be tested by treating the bivariate polynomial as a univariate polynomial in one of the variables with coefficients that are polynomials in an “auxiliary” variable. Assume, for the simplicity of the following description, that the bivariate polynomial has degree (N, N). Early 2D stability tests relied on division-free 1D stability and consequently ended with 2D tests whose complexity grows at an exponential rate with N making them quite messy and even impractical (due to numerical inaccuracy) for already moderate values of N. The second generation of 2D tests in [8,9,20], called “tabular” or “polynomial array” tests, reduced the complexity to $o(N^6)$ by restraining the growth of the so-called “balanced polynomial” coefficients (that involve both positive and negative powers of the auxiliary variable) by extracting from the coefficients common divisor polynomials. The balanced polynomial coefficients behave in this scheme in a manner that is analogous to Gaussian integer coefficients. In this analogy, polynomial degrees of the coefficients correspond to the size of integers, staying with polynomial coefficients corresponds to staying over integers, and the overall count of arithmetic operations corresponds to the binary complexity of the GIP algorithm. It is also relevant to mention here that a third generation of 2D stability tests that attain $o(N^4)$ of complexity, attain this complexity using the MJT, FFG, and FFGM tests as “companion 1D stability tests” in [10,11,13], respectively, to “telepolate” (interpolate) corresponding second generation tabular tests. The foregoing notes will be used to deduce the FF properties from proofs in the 2D stability context. They are also helpful to follow the count of binary operations by regarding them as standard arithmetical operations in recursions with polynomial coefficients.

3 Immittance-Type FF Tests

Define for a polynomial $D(z) = \sum_{i=0}^{n} d_i z^i$ the reciprocal polynomial $D^\sharp(z) = \sum_{i=0}^{n} d_{n-i}^\star z^i$, where * denotes the operation of complex conjugation. The immittance-type algorithms associate to the tested polynomial a sequence of symmetric polynomials of descending degrees, where a polynomial $D(z)$ is called symmetric if $D^\sharp(z) = D(z)$. It is noted that the coefficients of a complex or real symmetric polynomial exhibit conjugate-symmetry, $d_k = d^\star_{n-k}$, or real symmetry, $d_k = d_{n-k}$, $k = 0, \ldots, n$, with respect to the center. Therefore, a symmetric polynomial is well defined by about half of its coefficients.

3.1 Complex Polynomials (FFG)

We begin with a stability test for complex polynomials that is GIP and will be called the FFG (Fraction-Free Gaussian) test.

The FFG algorithm Pre-scale $D(z) \in \mathbb{C}[z]$ as in (1) to form $\hat{D}(z) = D(1)^\star D(z)$ and construct for it a sequence of symmetric complex polynomials $\hat{G}_m(z) = \sum_{i=0}^{n-m} \hat{g}_{m,i} z^i$, $m = 0, 1, \ldots, n$, as follows.

$$\hat{G}_0(z) = \hat{D}(z) + \hat{D}^\sharp(z)\ ,\ \hat{G}_1(z) = \frac{\hat{D}(z) - \hat{D}^\sharp(z)}{z-1}\ ,\ \hat{q}_0 = \hat{G}_0(1) \tag{3a}$$

For $m = 1, \ldots, n-1$ do:

$$\hat{h}_m = \hat{g}_{m-1,0}\hat{g}^\star_{m,0}, \quad \hat{q}_m = |\hat{g}_{m,0}|^2$$
$$z\hat{G}_{m+1}(z) = \frac{(\hat{h}_m + \hat{h}^\star_m z)\hat{G}_m(z) - \hat{q}_m \hat{G}_{m-1}(z)}{\hat{q}_{m-1}}. \tag{3b}$$

Normal conditions are defined, here and in all the forthcoming algorithms, as the case where all the polynomials in the sequence have nonzero leading coefficients. Normal conditions provide a broad enough framework for testing stability because they always form necessary conditions for stability. (Consequently, a polynomial can be declared "not stable" as soon as a degree deficient polynomial is encountered.) Since the immittance-type stability tests involve symmetric polynomials, normal conditions also mean non zero free terms. Specifically, the FFG sequence is said to be normal if

$$\hat{G}_m(0) \neq 0, \quad m = 0, \ldots, n. \tag{4}$$

Note that for $\hat{G}_1(z)$ to be a polynomial, the numerator must vanish at $z = 1$. This requirement is taken care by forming $\hat{D}(z)$ that is real at $z = 1$. We use the hat upper script $\hat{}$ as an indicator that the recursion is actually applied to $\hat{D}(z) = D(1)^\star D(z)$. We reserve the no $\hat{}$ notation to a corresponding G-sequence that applies directly to $D(z)$. The preliminary scaling is avoidable when $D(z) \in C[z]$ happens to have $D(1) \in \mathbb{R}$ and of course when $D(z) \in \mathbb{R}[z]$. This admits a somewhat simpler integer algorithm, see the FFGr algorithm in the next section.

Theorem 1 *Assume that the FFG algorithm is applied to* $D(z) \in \mathbb{Z}_\mathrm{j}[z]$ *and obeys normal conditions. Then all* $\hat{G}_m(z) \in \mathbb{Z}_\mathrm{j}[z]$ *and the algorithm can be completed over*

$\mathbb{Z}_\mathrm{j}$. *Let $B = \ell(D(z))$, then the integer size of $\hat{G}_m(z)$ is*

$$\ell_{\hat{G}}(m) = 2mB\ , \quad m = 1, \ldots, n. \tag{5}$$

Proof The fact that each divisor $\hat{q}_m$, $m \geq 1$ cancels out a common factor from all the coefficients of the polynomial in the numerator (3b) was shown in [8, Lemma 2]. The origin of the factor $\hat{q}_0$ is different. It is created by the specific way that the recursion is initiated. The fact that it is an exact divisor in the creation of $\hat{G}_2$ follows by direct substitution as was detailed in [9, Lemma 2] (and it is similar to one we will carry out here for the forthcoming FFGr algorithm). Assume $\ell(D(z)) = B$. Then the integer-size for $\hat{D}(z) = D(1)^\star D(z)$ is $2B$. Therefore, $\ell_{\hat{G}}(0) = \ell_{\hat{G}}(1) = 2B$ (note that the measure neglects by definition possible increase by one bit or digit during the additive operations). Next $\ell_{\hat{G}}(2) = 4B$ is obtained by adding up $\ell(\hat{g}_{1,0})$, $\ell(\hat{g}_{0,0})$, and $\ell_{\hat{G}}(1)$ (or, equivalently, $\ell(\hat{g}_{1,0})$, $\ell(\hat{g}_{1,0})$, and $\ell_{\hat{G}}(0)$). For $m \geq 2$, the recursion takes a steady form that obeys the difference equation

$$\ell(m+1) = 2\ell(m) - \ell(m-1). \tag{6}$$

The solution of this equation for the initial conditions $\ell_{\hat{G}}(1) = 2B$, $\ell_{\hat{G}}(2) = 4B$ is (5). □

It was shown in [13], where the FFG test serves as a "companion 1D stability tests" for 2D stability testing, that the polynomials of the FFG sequence are proportional to respective degree polynomials of the original sequence in [4], say $T_m(z) = \sum_{k=0}^{n-m} t_{m,k} z^k$ (where the indexing of the polynomials here is in reversed order to that in [4]), viz.

$$\hat{G}_m(z) = \hat{\beta}_m T_m(z), \tag{7}$$

where $\hat{\beta}_m$ are given by $\hat{\beta}_0 = \hat{\beta}_1 = 1$ and for $m = 2, \ldots, n$, by products of $|t_{m,0}|^2$ as follows:

$$\hat{\beta}_m = \prod_{i=1}^{m-1} \frac{|t_{i,0}|^2}{\hat{q}_0}\ , \tag{8}$$

or by products of $|\widetilde{g}_{m,0}|^2$,

$$\hat{\beta}_m = \begin{cases} \displaystyle\prod_{i=1}^{\ell} \frac{|\hat{g}_{2i,0}|^2}{|\hat{g}_{2i-1,0}|^2} & m = 2\ell + 1 \\[2ex] \displaystyle\frac{|\hat{g}_{1,0}|^2}{\hat{q}_0} \prod_{i=2}^{\ell} \frac{|\hat{g}_{2i-1,0}|^2}{|\hat{g}_{2i-2,0}|^2} & m = 2\ell \end{cases} \tag{9}$$

Since all $\hat{\beta}_m > 0$, the zero location rule stated in the next theorem follows from [4] or [12] assuming normal conditions there.

Theorem 2 *Assume the FFG algorithm produces for $D(z)$ (2) a normal sequence. Then, $D(z)$ has ν IUC and $n-\nu$ OUC zeros, where ν presents the number of sign variations in the sequence*

$$\nu = Var\{\hat{G}_0(1), \hat{G}_1(1), \ldots, \hat{G}_n(1)\} \tag{10}$$

The sign rule is well defined because the conjugation in the symmetry of $\hat{G}_m(z)$ implies $\hat{G}_m(1) \in \mathbb{R}$ and $D(1) \neq 0$ plus normal conditions imply that $\hat{G}_m(1) \neq 0$ for all m (set $z = 1$ into the polynomial recursion to realize that). We also remind that normal conditions imply no UC zeros (because, as was shown for the original test, UC zeros imply a premature termination with an identically zero polynomial).

Stability conditions for the FFG algorithm correspond to the special case of $\nu = 0$. Namely, all entries in (10) must have a common sign. This common sign must be positive because $G_0(1) = 2|D(1)|^2 > 0$. So, $D(z)$ is stable if, and only if,

$$\hat{G}_m(1) > 0, \quad m = 0, 1, \ldots, n. \tag{11}$$

Some further necessary conditions for stability are

$$\mathcal{R}\text{e}\{\hat{g}_{m-1,0}\hat{g}^{\star}_{m,0}\} > 0, \quad m = 1, \ldots, n \tag{12}$$

This can be seen by setting $z = 1$ into the recursion (3b) to obtain

$$\hat{q}_{m-1}\hat{G}_{m+1}(1) + \hat{q}_m\hat{G}_{m-1}(1) = 2\mathcal{R}\text{e}\{\hat{h}_m\}\hat{G}_m(1) \tag{13}$$

Since all $\hat{q}_m > 0$ and stability implies (11), it follows that all $\mathcal{R}\text{e}\{\hat{h}_m\} > 0$, i.e., (12), holds for all m. It is seen from (12) that indeed the normal conditions (4) are necessary for stability. By a similar reasoning, it will follow that normal conditions for all the forthcoming immittance-type tests are necessary for stability. So that the normal conditions always provide a broad enough framework for testing stability.

Example 1 Consider the next Gaussian integer polynomial.

$$D^{(c)}(z) = \mathrm{j} + 3z + 2z^2 + 4z^3 + 8z^4 + 7z^5 + 5z^6 + 8z^7 \tag{14}$$

Application of the FFG algorithm to (14) produces the following sequence of polynomials.

$$\begin{aligned}
\hat{G}_0(z) &= 297 + 45\mathrm{j} + (296 + 2\mathrm{j})z + (333 + 5\mathrm{j})z^2 + (444 + 4\mathrm{j})z^3 \\
&\quad + (444 - 4\mathrm{j})z^4 + (333 - 5\mathrm{j})z^5 + (296 - 2\mathrm{j})z^6 + (297 - 45\mathrm{j})z^7 \\
\hat{G}_1(z) &= 295 - 29\mathrm{j} + (369 - 21\mathrm{j})z + (554 - 12\mathrm{j})z^2 + 702z^3 + (554 + 12\mathrm{j})z^4 \\
&\quad + (369 + 21\mathrm{j})z^5 + (295 + 29\mathrm{j})z^6 \\
\hat{G}_2(z) &= 11360 - 1048\mathrm{j} + (18324 + 278\mathrm{j})z + (25230 + 676\mathrm{j})z^2 \\
&\quad + (25230 - 676\mathrm{j})z^3 + (18324 - 278\mathrm{j})z^4 + (11360 + 1048\mathrm{j})z^5 \\
\hat{G}_3(z) &= 596152 - 136\mathrm{j} + (855712 + 52896\mathrm{j})z + 901872z^2
\end{aligned}$$

$$
\begin{aligned}
&\quad + (855712 - 52896\mathrm{j})z^3 + (596152 + 136\mathrm{j})z^4\\
\hat{G}_4(z) &= 25766056 + 743380\mathrm{j} + (22309044 + 685512\mathrm{j})z\\
&\quad + (22309044 - 685512\mathrm{j})z^2 + (25766056 - 743380\mathrm{j})z^3\\
\hat{G}_5(z) &= 477929932 - 32791076\mathrm{j} + 240566688z + (477929932 + 32791076\mathrm{j})z^2\\
\hat{G}_6(z) &= 5525250784 - 1272045056\mathrm{j} + (5525250784 + 1272045056\mathrm{j})z\\
\hat{G}_7(z) &= 90733722368
\end{aligned}
$$

All $\hat{G}_m(1) > 0$. Therefore, by Theorem 2 or (11), $D^{(c)}(z)$ is stable.

3.2 Real Polynomials

Consider now the testing of a real polynomial (2). Clearly, the FFG test can still be used and it is readily realized that it remains over the real integers when $D(z) \in \mathbb{Z}[z]$. However, the real case admits simpler integer algorithms. We begin with a minor simplification of the initiation only and then proceed to a more substantial simplification that modifies the whole form of the recursion.

3.2.1 A Minor Simplification (FFGr)

When $D(z) \in \mathbb{R}[z]$ then necessarily $D(1) \in \mathbb{R}$. It is then possible to apply the FFG recursion directly to $D(z)$ without first forming $\hat{D}(z) = D(1)D(z)$. Actually, the same is true also for $D(z) \in \mathbb{C}[z]$ when $D(1) \in \mathbb{R}$. We refer to this situation as the FFGr (reduced FFG) algorithm and assign to it a G-sequence with all ˆ's removed. We spell out here the FFGr case for the real case (it will be used later for further simplification) and will add comments on its complex version at the end of this subsection.

The FFGr algorithm Construct for $D(z)$ in (2) a sequence $G_m(z) = \sum_{i=0}^{m} g_{m,i} z^i$, $m = 0, \ldots, n$ of real symmetric polynomials as follows.

$$G_0(z) = D(z) + D^{\sharp}(z)\ ,\ G_1(z) = \frac{D(z) - D^{\sharp}(z)}{z-1} \tag{15a}$$

Set $q_0 = 2$ and $m = 1, \ldots, n-1$ do:

$$
\begin{aligned}
h_m &= g_{m-1,0}\, g_{m,0}, \quad q_m = g_{m,0}^2\\
zG_{m+1}(z) &= \frac{h_m(z+1)G_m(z) - q_m G_{m-1}(z)}{q_{m-1}}.
\end{aligned} \tag{15b}
$$

The fact that the divisions by the q_m's for $m \geq 1$ leave the recursion FF is inherited from the FFG algorithm. To realize that the factor $q_0 = 2$ leaves $G_2(z)$ over the integers, substitute into $q_0(z-1)zG_2(z)$, $g_{0,0} = d_0 + d_n$ and $g_{1,0} = -d_0 + d_n$ and obtain after some evaluation

$$q_0(z-1)zG_2(z) = 2(d_0 d_n - d_o^2)(zD(z) - D(z)^{\sharp}) + 2(d_0 d_n - d_n^2)(zD(z)^{\sharp} - D(z)).$$

Notice that overlooking the division by $q_0 = 2$ at the beginning (i.e., using $q_0 = 1$) produces a sequence of polynomials such that $G_{m+1}(z) \to 2^m G_{m+1}(z)$, $m = 1, \ldots, n-1$.

Lemma 1 *The polynomials of the FFG sequence for a real $D(z)$ and the polynomials of the FFGr sequence are related by*

$$\hat{G}_m(z) = D(1)G_m(z) \,, \quad m = 0, \ldots, n \tag{16}$$

Proof The relations evidently hold for $m = 0, 1$ from where $\hat{h}_1 = D(1)^2 h_1$ and $\hat{q}_1 = D(1)^2 q_1$ follow. Also, by their definition, $\hat{q}_0 = 2D(1)^2 = D(1)^2 q_0$. Next,

$$z\hat{G}_2(z) = \frac{\hat{h}_1(z+1)\hat{G}_1(z) - \hat{q}_1\hat{G}_0(z)}{\hat{q}_0} = \frac{D(1)^3}{D(1)^2} zG_2(z)$$

verifies (16) for $m = 2$. Assume (induction step) that (16) holds for $k \leq m$. Use $\hat{h}_k = D(1)^2 h_k$ for $k = m-1, m$ and $\hat{q}_{m-1} = D(1)^2 q_{m-1}$ to evaluate

$$z\hat{G}_{m+1}(z) = \frac{\hat{h}_m(z+1)\hat{G}_m(z) - \hat{q}_m\hat{G}_{m-1}(z)}{\hat{q}_{m-1}} = D(1)zG_{m+1}(z),$$

shows that (16) holds also for $m+1$ and proves the induction step. □

It follows that the integer-size of the polynomials in the FFGr sequence is lower by B than respective polynomials of the FFG sequence (5), viz.

$$\ell_G(m) = (2m-1)B \,, \quad m = 1, \ldots, n. \tag{17}$$

It also follows that the zero location rule for the FFGr algorithm is still given by Theorem 2 this time with

$$\nu = Var\{G_0(1), G_1(1), \ldots, G_n(1)\} \tag{18}$$

The necessary and sufficient conditions for stability are represented by $\nu = 0$ in (18). Since $D(1) > 0$ is a part of the assumption in (2), $G_0(1) = 2D(1) > 0$, and it can be stated that $D(z)$ is stable if, and only if,

$$G_m(1) > 0, \quad m = 0, \ldots, n. \tag{19}$$

However, notice that in difference from the stability rule for the FFG algorithm that is given by (11) irrespective of the sign of $D(1)$, here if $D(1) < 0$ then $D(z)$ is stable if and only if all $G_m(1) < 0$. A further set of necessary conditions for stability of $D(z)$ in (2) is given by

$$g_{m,0} > 0, \quad m = 0, \ldots, n. \tag{20}$$

To prove (20), take (12) and (16) and run (13) for $m = n, n-1, \ldots$ starting with $g_{n,0} = G_n(1) > 0$.

Example 2 Consider the next real polynomial

$$D^{(r)}(z) = 1 + 3z + 2z^2 + 4z^3 + 8z^4 + 7z^5 + 5z^6 + 8z^7 \tag{21}$$

Testing (21) with the FFGr algorithm produces the next sequence of polynomials.

$$\begin{aligned}
G_0(z) &= 9 + 8z + 9z^2 + 12z^3 + 12z^4 + 9z^5 + 8z^6 + 9z^7 \\
G_1(z) &= 7 + 9z + 14z^2 + 18z^3 + 14z^4 + 9z^5 + 7z^6 \\
G_2(z) &= 308 + 504z + 714z^2 + 714z^3 + 504z^4 + 308z^5 \\
G_3(z) &= 18304 + 26488z + 27984z^2 + 26488z^3 + 18304z^4 \\
G_4(z) &= 881920 + 715520z + 715520z^2 + 881920z^3 \\
G_5(z) &= 15476000 + 3985600z + 15476000z^2 \\
G_6(z) &= 121180000 + 121180000z \\
G_7(z) &= 1653360000
\end{aligned}$$

It is obvious that the conditions (19) hold. Therefore, the polynomial $D^{(r)}(z)$ is stable.

FFGr for a complex polynomial A reduced FFG algorithm is possible also for $D(z) \in \mathbb{C}[z]$ if $D(1) \in \mathbb{R}$. The realness of $D(1)$ allows the application of the FFG recursion directly to $D(z)$. The algorithm begins with (15a) and $q_0 = 2$ instead of (3a). It proceeds with a recursion similar to (3b) except that all the ˆ's are removed. Again, each $G_m(z)$ saves, compared to $\hat{G}_m(z)$, a factor $D(1)$ as in (16). The number of OUC zeros is given by the rule (18). The stability conditions are given by (19) (if $D(1) > 0$), and (12) with $\hat{}s$ removed provides a set of further necessary conditions for stability.

3.2.2 A Major Simplification (FFR)

It turns out that it is possible to test stability of a real polynomial by a significantly more efficient IP test than the FFGr or the FFG tests. We call it here the FFR (fraction-free real) test and show its relation to the real FFGr test from where its ZL rule can also be deduced. The test was presented before in a previous paper in this journal [15]. Its ZL rule in Theorem 4 is derived here from the ZL location of the FFGr algorithm through relations between the two sequences of polynomials. During the preparation of this paper, we realized that a simplified set of necessary and sufficient conditions in [15, Theorem 7] is not correct. We derive in this section two sets of necessary and sufficient conditions for stability (Theorem 6) where each of them attain the intended strength of simplification, in offering about n conditions of shorter integers (shorter than using the "no change of signs" condition in the ZL rule).

The FFR algorithm Construct for $D(z) \in \mathbb{R}[z]$ in (2) a sequence of real symmetric polynomials $R_m(z) = \sum_{k=0}^{n-m} r_{m,k} z^k$ $m = 0, \dots, n$, as follows:

$$R_0(z) = D(z) + D^{\sharp}(z), \quad R_1(z) = \frac{D(z) - D^{\sharp}(z)}{z - 1} \tag{22a}$$

Set $\eta_0 = 2$, $\eta_1 = 1$ and for $m = 1, 2, \dots, n-1$ do:

$$\begin{aligned} \eta_{m+1} &= r_{m,0} \\ zR_{m+1}(z) &= \frac{r_{m-1,0}(1+z)R_m(z) - r_{m,0}R_{m-1}(z)}{\eta_{m-1}}. \end{aligned} \tag{22b}$$

The normal conditions for the FFR algorithm are

$$R_m(0) \neq 0 \quad m = 0, \dots, n\,. \tag{23}$$

The FFR algorithm is FF over real integers and produces integers that grow at a rate lower by a factor of about 2 than using the FFG or FFGr algorithms for a real integer polynomial. Both parts of the next theorem were proved in [15].

Theorem 3 *Assume the FFR algorithm is applied to $D(z) \in \mathbb{Z}[z]$ and obeys normal conditions. Then, all $R_m(z) \in \mathbb{Z}[z]$ and the algorithm can be completed over $\mathbb{Z}$. Let $B = \ell(D(z))$, then the integer-size of $\tilde{R}_m(z)$, $\ell_R(m) = \ell(\tilde{R}_m(z))$, is*

$$\ell_R(m) = mB, \quad m = 1, \dots, n \tag{24}$$

Lemma 2 *The relationship between the polynomials produced by the FFR and (real) FFGr algorithms is as follows. $G_0(z) = R_0(z)$, $G_1(z) = R_1(z)$ and*

$$G_m(z) = r_{m-1,0}R_m(z), \quad m = 2, \dots, n \tag{25}$$

Proof by induction.

$$\begin{aligned} zG_2(z) &= g_{1,0}[g_{0,0}(z+1)G_1(z) - g_{1,0}G_2(z)]/2 \\ &= r_{1,0}[r_{0,0}(z+1)G_1(z) - r_{1,0}R_2(z)]/2 = r_{1,0}zR_2(z) \end{aligned}$$

shows that (25) holds for $m = 2$. Assume (25) holds for $k = 2, 3, \dots, m$. Then,

$$\begin{aligned} zG_{m+1}(z) &= \frac{g_{m,0}g_{m-1,0}(1+z)G_m(z) - g_{m,0}G_{m-1}(z)]}{g_{m-1,0}^2} \\ &= \frac{r_{m-1,0}r_{m,0}[r_{m-2,0}r_{m-1,0}^2(z+1)R_m(z) - r_{m-1,0}r_{m,0}r_{m-2,0}R_{m-1}(z)]}{r_{m-2,0}^2 r_{m-1,0}^2} \\ &= r_{m,0}\left[\frac{r_{m-1,0}(z+1)R_m(z) - r_{m,0}R_{m-1}(z)}{r_{m-2,0}}\right] = r_{m,0}zR_{m+1}(z) \end{aligned}$$

implies that (25) holds also for $k = m+1$. □

The next zero location rule was stated in [15, Theorem 9]

Theorem 4 *Assume the FFR algorithm produces for $D(z)$ in (2) a normal sequence. Then $D(z)$ has ν IUC and $n-\nu$ OUC zeros, where ν presents the number of sign variations in the sequence*

$$\nu = Var\{R_0(1), R_1(1), r_{1,0}R_2(1), \ldots, r_{n-1,0}R_n(1)\} \tag{26}$$

It can now be deduced from Theorem 2. It amounts to setting (25) into the FFGr ZL conditions in (18). Stability is represented by $\nu = 0$ in (26). Since $R_0(1) = 2D(1) > 0$ by the assumption in (2), it follows that $D(z)$ is stable if, and only if,

$$R_0(1) > 0,\ R_1(1) > 0 \text{ and } r_{k-1,0}R_k(1) > 0\ ,\ \ k = 2, \ldots, n \tag{27}$$

Next, we prove that in this case each term in the product $r_{k-1,0}R_k(1) > 0$ must be positive.

Theorem 5 *Assume the FFR algorithm is applied to the real polynomial $D(z)$ in (2) and produces a normal sequence. Then $D(z)$ is stable if, and only if,*

$$R_m(1) > 0,\ \ r_{m,0} > 0\ ,\quad m = 0, 1, \ldots, n \tag{28}$$

Proof If all the conditions in (28) hold then all the entries in (27) are positive and stability follows. For the converse, assume that $D(z)$ in (2) is stable. Then $D(1) > 0$ implies $d_n > 0$. Stability also implies $|d_n| > |d_0|$. Therefore, $r_{0,0} = d_n + d_0 > 0$, $r_{1,0} = d_n - d_0 > 0$. Use the necessary conditions for stability of the FFGr algorithm in (20), $g_{m,0} > 0$, and (25) to get $g_{m,0} = r_{m-1,0}r_{m,0} > 0$ for $m \geq 2$. The latter implies $r_{m,0} > 0$ for also $m = 2, \ldots, n$. The positivity of all $r_{m,0}$ in combination with the positivity of all terms in (27) imply that positivity of all $R_m(1)$. □

Beyond ruling out the possibility that two negatives multiplicands make a positive product, the set (28) is also more attractive in that it involves shorter integers than the product terms $r_{m-1,0}R_m(1)$ in (27). This is a desirable simplification when the testing of stability involves symbols. For example, suppose one of the coefficients is not numerical, but letteral, say $d_k = K$, and the goal is to find the range of values of K for which the polynomial is stable. In such a situation, it can be argued that the coefficients evolve into polynomial in K of degree up to the bound found for the integers size in (24). So, Theorem 5 admits separate consideration of two polynomials of degrees up to $m-1$ and m instead of the examination of $r_{m-1,0}R_m(1)$, a polynomial in K of degree up to $2m-1$.

Theorem 5 halves the integer size at the expense of doubling the number of the integers that have to be examined. At least two obvious duplicities, $R_n(1) = r_{n,0}$ and $R_{n-1}(1) = 2r_{n-1,0}$ can be readily dropped from the set (28). Actually, about half of the conditions in the set (28) are redundant. The next theorem provides two reduced sets of necessary and sufficient condition for stability each uses about half of the conditions in (28).

Theorem 6 *Assume the FFR algorithm produces for the real polynomial $D(z)$ (2) a normal sequence. Then, each of the next two sets provides a set of necessary and* *sufficient conditions for the stability of* $D(z)$.

Set I:

$$\begin{aligned} r_{2i,0} > 0\,, \quad i = 0, \dots, \left\lfloor \tfrac{n}{2} \right\rfloor \\ R_{2i-1}(1) > 0\,,\ \ i = 1, \dots, \left\lfloor \tfrac{n+1}{2} \right\rfloor \end{aligned} \tag{29}$$

Set II:

$$\begin{aligned} r_{0,0} > 0,\ r_{2i-1,0} > 0\,,\ \ i = 1, \dots, \left\lfloor \tfrac{n+1}{2} \right\rfloor \\ R_{2i}(1) > 0\,, \quad i = 1, \dots, \left\lfloor \tfrac{n}{2} \right\rfloor \end{aligned} \tag{30}$$

where $\lfloor x \rfloor$ *denotes the integer floor of* x.

Proof All the participating entries are positive if the polynomial is stable because they are members of the set of conditions in Theorem 5 that provide necessary conditions for stability. To prove the converse, we need to show that each set is enough to make all the entries in (26) positive. Let us write out, step by step, the equations that result by setting $z = 1$ into (22b)

$$\begin{aligned} m = 1: &\quad 2r_{0,0}R_1(1) = 2R_2(1) + r_{1,0}R_0(1) \\ m = 2: &\quad 2r_{1,0}R_2(1) = R_3(1) + r_{2,0}R_1(1) \\ m = 3: &\quad 2r_{2,0}R_3(1) = r_{1,0}R_4(1) + r_{3,0}R_2(1) \\ m = 4: &\quad 2r_{3,0}R_4(1) = r_{2,0}R_5(1) + r_{4,0}R_3(1) \\ &\qquad\qquad \vdots \qquad \vdots \\ m \geq 3: &\quad 2r_{m-1,0}R_m(1) = r_{m-2,0}R_{m+1}(1) + r_{m,0}R_{m-1}(1) \end{aligned}$$

Set I ensures that each real number that appears in the right-hand side of the equations labeled by $m = 2, 4, 6, \ldots$ are positive. Notice then that this implies not just the positivity of the respective left-hand sides, but also the positivity of the left-hand sides of the respective next equation. Altogether, the requested positivity of all the entries in (26) is ensured. Set II, is designed to cover the positivity of each real number in the right-hand side of the lines labeled $m = 1, 3, 5, \ldots$. Then, it follows similarly that all the left hand sides are also positive and that altogether, all the terms in (26) are positive. □

Note that both sets contain $n + 1$ terms (not counting $R_0(1) = 2D(1) > 0$ that is part of the assumption in (2)).

Example 3 Testing the polynomial $D^{(r)}(z)$ in (21) with the FFR algorithm produces the next sequence of polynomials.

$$\begin{aligned} R_0(z) &= 9 + 8z + 9z^2 + 12z^3 + 12z^4 + 9z^5 + 8z^6 + 9z^7 \\ R_1(z) &= 7 + 9z + 14z^2 + 18z^3 + 14z^4 + 9z^5 + 7z^6 \end{aligned}$$

$$
\begin{aligned}
R_2(z) &= 44 + 72z + 102z^2 + 102z^3 + 72z^4 + 44z^5\\
R_3(z) &= 416 + 602z + 636z^2 + 602z^3 + 416z^4\\
R_4(z) &= 2120 + 1720z + 1720z^2 + 2120z^3\\
R_5(z) &= 7300 + 1880z + 7300z^2\\
R_6(z) &= 16600 + 16600z\\
R_7(z) &= 99600
\end{aligned}
$$

It is apparent that the sequence meets the stability requirement in Theorem 4 (or 5). According to set I (say) of Theorem 6, it is possible to conclude stability from the positivity of just $r_{0,0}, r_{2,0}, r_{4,0}, r_{6,0}$ and $R_1(1), R_3(1), R_5(1), R_7(1)$. The smaller size of integers here in comparison to Example 2 illustrates well the about half size of integers in (24) compared to (17).

4 FF Immittance-Type Tests with Modified Initiation

In this section, we bring two FF tests, the FFGM and the FFRM tests that form the modified BT test [7] counterparts of the FFG and the FFR tests ("M" is added for "modified"). The modified test in [7] differs from the original BT in the way the recursion is initiated. In the complex case, it circumvents the pre-scaling of $D(z)$ by $D(1)^\star$. In the real case, it produce a sequence of polynomials that are proportional to symmetric parts of scattering-type sequence of polynomials [5]. The modified initiation also affects the way that the FFGM and FFRM relate to stability determinants (an aspect that is left out from the scope of the current presentation). It was shown in [7] that the modified test assigns to a polynomial $D(z)$ a same sequence of polynomials that the original test form would assign to the polynomial $(z-1)D(z)$. Similar relationships exist between the FFGM and the FFG tests and between the FFRM and the FFR tests. This implies readily the FF properties of the FFGM and FFRM leaving the main remaining required attention to ZL and stability rules. For brevity, the presentation will be without numerical illustrations.

4.1 Complex Polynomials (FFGM)

Although the following FFGM test has not pointed out never before as a GIP test, the 2D stability test in [8] can be regarded as its application to testing 2D stability and it appeared as the "companion 1D stability test" for its interpolation in [10].

The FFGM algorithm Construct for the complex polynomial $D(z)$ in (1) a sequence of complex symmetric polynomials, $\tilde{G}_m(z) = \sum_{k=0}^{n-m} \tilde{g}_{m,k} z^k$, $m = -1, 0, 1, \dots, n$, as follows.

$$\tilde{G}_{-1}(z) = (z-1)(D(z) - D^\sharp(z))\,,\ \tilde{G}_0(z) = D(z) + D^\sharp(z)\,,\ \tilde{q}_{-1} = 2 \qquad (31a)$$

For $m = 0, \dots, n-1$ do:

$$\tilde{h}_m = \tilde{g}_{m-1,0}\tilde{g}^{\star}_{m,0}, \quad \tilde{q}_m = |\tilde{g}_{m,0}|^2$$
$$z\tilde{G}_{m+1}(z) = \frac{(\tilde{h}_m + \tilde{h}^{\star}_m z)\tilde{G}_m(z) - \tilde{q}_m \tilde{G}_{m-1}(z)}{\tilde{q}_{m-1}} \tag{31b}$$

The normal conditions for the FFGM algorithm are

$$\tilde{G}_m(0) \neq 0 \quad m = -1, 0, \ldots, n\,. \tag{32}$$

Theorem 7 *Assume the FFGM algorithm is applied to $D(z) \in \mathbb{Z}_{\mathrm{j}}[z]$ and obeys normal conditions. Then all $\tilde{G}_m(z) \in \mathbb{Z}_{\mathrm{j}}[z]$ and the algorithm can be completed over $\mathbb{Z}_{\mathrm{j}}$. Let $B = \ell(D(z))$ then the integer-size of $\tilde{G}_m(z)$ is*

$$\ell_{\tilde{G}}(m) = (2m+1)B\,, \quad m = 1, \ldots, n \tag{33}$$

Proof The algorithm produces the same sequence that the FFG algorithm without pre-scaling would produce for the complex polynomial $(z-1)D(z)$. (We mean the FFGr algorithm for a complex polynomial with $q_0 = 2$ mentioned at the end of the Sect. 3.2.1). Therefore the FFGM algorithm is GIP for $D(z) \in \mathbb{Z}_{\mathrm{j}}[z]$. The integer-size of the polynomials begins with $\ell_{\tilde{G}}(-1) = B$, $\ell_{\tilde{G}}(0) = B$, $\ell_{\tilde{G}}(1) = 3B$. Next, for $m \geq 1$, the integer sizes obey the difference Eq. (6) whose solution for the current $\ell_{\tilde{G}}(0) = B$ and $\ell_{\tilde{G}}(1) = 3B$ is (33). □

Zero location rule for the FFGM test can be obtained from the ZL rule for the modified BT in a similar way that Theorem 2 was obtained from the original BT. Denote the polynomials in [7] by $\{\tilde{T}_m(z) = \sum_{k=0}^{n-m} \tilde{t}_{m,k} z^k, m = -1, 0, 1, \ldots, n\}$. The polynomials $\{\widetilde{G}_m(z)\}$ of the FFGM sequence are proportional to polynomials of corresponding degree $\{\tilde{T}_m(z)\}$ of the modified BT

$$\widetilde{G}_m(z) = \widetilde{\beta}_m \widetilde{T}_m(z), \quad m = -1, 0, 1, \ldots, n \tag{34}$$

where $\widetilde{\beta}_{-1} = 1$ $\widetilde{\beta}_0 = 1$ and afterwards for $m = 1, \ldots, n$, it can be shown by induction that $\widetilde{\beta}_m$ are given by products of and $|\widetilde{t}_{i,0}|^2$ like in (8) or products of $|\widetilde{g}_{m,0}|^2$ like in (9). Therefore, all $\widetilde{\beta}_m > 0$. It follows that, assuming normal conditions, $D(z)$ has ν OUC and $n-\nu$ IUC zeros, where ν is given by (if $G_0(1) \neq 0$, else by (38) below!)

$$\nu = Var\{\widetilde{G}_0(1), \widetilde{G}_1(1), \ldots, \widetilde{G}_{n-1}(1), \widetilde{G}_n(1)\}. \tag{35}$$

The expression (35) fails to give the number of OUC zeros when $\mathcal{R}\mathrm{e}\{D(1)\} = 0$. We have $G_{-1}(1) = 0$ by definition. If $\mathcal{R}\mathrm{e}\{D(1)\} = 0$, that is not excluded by the assumption in (1), then $G_0(1) = 2\mathcal{R}\mathrm{e}\{D(1)\} = 0$. It then follows that $\widetilde{G}_m(1) = 0$ for all m (verifiable by setting $z = 1$ into the recursion). In [7] this situation was avoided by setting the assumption $\mathcal{R}\mathrm{e}\{D(1)\} \neq 0$. This is a not restrictive assumption for a stand alone test because otherwise the test can be applied, e.g., to $jD(z)$. However, for certain generalized applications occasional change of the actually tested polynomial is not adequate. We encountered such a situation in the 2D testing context [8,10] and showed there that it is possible to adhere to the assumption (1) by creating a sequence of auxiliary parameters $\tilde{\gamma}_m$ by the next recursion. Set $\tilde{\gamma}_{-1} = 0$, $\tilde{\gamma}_0 = 1$ then for $m \geq 0$ do:

$$\tilde{\gamma}_{m+1} = \frac{(\tilde{h}_m + \tilde{h}_m^\star)\tilde{\gamma}_m - \tilde{q}_m \tilde{\gamma}_{m-1}}{\tilde{q}_{m-1}}. \tag{36}$$

The $\tilde{\gamma}_m$'s are related to the values $G_m(1)$ by

$$\tilde{\gamma}_m = \frac{\widetilde{G}_m(1)}{\widetilde{G}_0(1)}\,, \quad m = 1, \ldots, n \tag{37}$$

However, they remain nonzero (if normal conditions hold) also when $G_0(1) = 2\mathcal{R}\text{e}\{D(1)\} = 0$. So, to avoid the requirement $\mathcal{R}\text{e}\{D(1)\} \neq 0$, the ZL rule for the FFGM algorithm can be stated as follows.

Theorem 8 *Assume the FFGM algorithm produces for $D(z)$ in (1) a normal $\widetilde{G}$-sequence with parameters $\tilde{\gamma}_m$ (36). Then $D(z)$ has ν IUC and $n - \nu$ OUC zeros, where ν is given by*

$$\nu = Var\{1, \tilde{\gamma}_1, \ldots, \tilde{\gamma}_{n-1}, \tilde{\gamma}_n\}. \tag{38}$$

When $D(z) \in \mathbb{R}[z]$ or when $D(z) \in \mathbb{C}[z]$ and $\widetilde{G}_0(1) \neq 0$, ν is also given by (35).

In particular, $D(z)$ is stable if, and only if,

$$\tilde{\gamma}_m > 0\,, \quad m = 1, \ldots, n \tag{39}$$

Note that if $D(z) \in \mathbb{Z}_{\mathrm{j}}[z]$ then all $\tilde{\gamma}_m$ are real integers (because the recursion (36) used to create them is IP) of size $\ell(\tilde{\gamma}_m) = 2mB$, in accordance with (33) and (37).

Setting $z = 1$ into (31) and using (35) shows that next set of condition are necessary for stability

$$\mathcal{R}\text{e}\{\widetilde{g}_{m-1,0}\widetilde{g}^\star_{m,0}\} > 0, m = 0, \ldots, n. \tag{40}$$

When $D(z)$ is real (2), the conditions (40) simplify into the next set of necessary conditions for stability

$$\widetilde{g}_{m,0} > 0, \quad m = -1, 0, \ldots, n. \tag{41}$$

We shall use (41) later in the proof for Theorem 11. It can be realized as follows. For a real and stable $D(z)$, $D(1) > 0$ implies $d_n > 0$, and since $|d_n| > |d_0|$ is a necessary condition for stability, it follows that $\widetilde{g}_{-1,0} = d_n - d_0 > 0$. It then follows via (40) that all subsequent $\widetilde{g}_{m,0}$ are too positive.

4.2 Real Polynomials (FFRM)

The forthcoming FFRM test is the only FF stability test in this paper that was not presented neither as stability test (without attention to its FF property) nor utilized before implicitly in some generalized way. It complements the FFR test with simple dual relations to stability inner determinants whose relation to the BT was studied [30] and offers a particularly simple relations to the polynomials of the MJT sequence. It relates to the FFGM test like the FFR test relates to the FFG test. However, here there is no intermediate stage in moving from FFGM to FFRM, like the FFGr test that stands between the FFG and the FFR tests. The reason for this is that the FFGM involves no pre-scaling of the tested polynomial that the FFGr test drops.

The FFRM Algorithm Construct for the real polynomial $D(z)$ (2) a sequence of real symmetric polynomials $\widetilde{R}_m(z) = \sum_{k=0}^{n-m} \widetilde{r}_{m,k} z^k$, $m = 0, \ldots, n$, as follows.

$$\widetilde{R}_{-1}(z) = (z-1)(D(z) - D^\sharp(z)) \quad , \quad \widetilde{R}_0(z) = D(z) + D^\sharp(z) \tag{42a}$$

Set $\tilde{\eta}_{-1} = 2$, $\tilde{\eta}_0 = 1$ and for $m = 0, 1, \ldots, n$ do:

$$\begin{aligned} \widetilde{\eta}_{m+1} &= \widetilde{r}_{m,0} \\ z\widetilde{R}_{m+1}(z) &= \frac{\widetilde{r}_{m-1,0}(1+z)\widetilde{R}_m(z) - \widetilde{r}_{m,0}\widetilde{R}_{m-1}(z)}{\widetilde{\eta}_{m-1}} \end{aligned} \tag{42b}$$

Theorem 9 *Assume the FFRM algorithm is applied to $D(z) \in \mathbb{Z}[z]$ and obeys normal conditions. Then all $\widetilde{R}_m(z) \in \mathbb{Z}[z]$ and the algorithm can be completed over $\mathbb{Z}$. Assume $B = \ell(D(z))$ then the integer-size of $\widetilde{R}_m(z)$ is*

$$\ell_{\widetilde{R}}(m) = (m+1)B \ , \quad m = 0, \ldots, n \tag{43}$$

Proof The FFRM is FF because it can be regarded as application of the FFR test to $(z-1)D(z)$ that is FF by Theorem 3. The recursion obeys for $m \geq 2$, the integer-size difference equation $\ell(m+1) = \ell(m) + \ell(m-1) - \ell(m-2)$ (same as for the FFR algorithm). Solving it for the initial conditions $\ell(0) = B$, $\ell(1) = 2B$ and $\ell(2) = 3B$ gives (43). □

Lemma 3 *The relationship between the polynomials produced by the FFRM and the FFGM algorithm, applied to a real $D(z)$, is $\widetilde{G}_{-1}(z) = \widetilde{R}_{-1}(z)$, $\widetilde{G}_0(z) = \widetilde{R}_0(z)$ and*

$$\widetilde{G}_m(z) = \widetilde{r}_{m-1,0}\widetilde{R}_m(z) \quad m = 1, \ldots, n \tag{44}$$

Proof Similar to the proof for Lemma 2. □

Theorem 10 *Assume the FFRM algorithm produces for the real polynomial $D(z)$ (2) a normal sequence. Then $D(z)$ has ν IUC and $n - \nu$ OUC zeros, where ν is given by*

$$\nu = Var\{\widetilde{R}_0(1), \widetilde{r}_{0,0}\widetilde{R}_1(1), \ldots, \widetilde{r}_{n-2,0}\widetilde{R}_{n-1}(1), \widetilde{r}_{n-1,0}\widetilde{R}_n(1)\} \tag{45}$$

Proof For the current case of real polynomial, the assumption that $D(1) > 0$ allows the use of Theorem 8 with ν given by (35). The expression (45) follows from substitution of the relations in Lemma 3 into (35). □

Let us look more intensively into the stability conditions. The common sign required by $\nu = 0$ in (45) must be positive because $\widetilde{R}_0(1) = 2D(1) > 0$ by assumption (2). The next theorem states that in the product terms $\widetilde{r}_{m-1,0}\widetilde{R}_m(1)$ both factors must be positive.

Theorem 11 *Assume the FFRM algorithm produces for $D(z)$ in (2) a normal sequence. Then $D(z)$ is stable if and only if*

$$\widetilde{R}_m(1) > 0\ ,\ \widetilde{r}_{m,0} > 0 \quad , \quad m = 0, \ldots, n \tag{46}$$

Proof Use the ZL rule (45), the FFGM necessary condition for stability in the real case (41), and (44) to write $\widetilde{g}_{m,0} = \widetilde{r}_{m-1,0}\widetilde{r}_{m,0} > 0$. Then follow the proof of Theorem 5. □

Similarly to the FFR test case, about half of the conditions in (46) are redundant.

Theorem 12 *Each of the next two sets provides necessary and sufficient conditions for stability of a real polynomial* (2) *treated by the FFRM algorithm.*
Set I

$$\begin{aligned} \widetilde{r}_{-1,0} > 0\ ,\ \widetilde{r}_{2i,0} > 0\ , \quad i = 0, \ldots, \left\lfloor \tfrac{n}{2} \right\rfloor \\ \widetilde{R}_{2i-1}(1) > 0\ ,\ \ i = 1, \ldots, \left\lfloor \tfrac{n+1}{2} \right\rfloor \end{aligned} \tag{47}$$

Set II:

$$\begin{aligned} \widetilde{r}_{2i-1,0} > 0\ ,\ \ i = 1, \ldots, \left\lfloor \tfrac{n+1}{2} \right\rfloor \\ \widetilde{R}_{2i}(1) > 0\ , \quad i = 1, \ldots, \left\lfloor \tfrac{n}{2} \right\rfloor \end{aligned} \tag{48}$$

Proof The conditions are subsets of the conditions in Theorem 11 therefore are necessary for stability. To prove that each set is sufficient for stability, set $z = 1$ into (42b) to write

$$\begin{aligned}
m = 0: &\quad \widetilde{r}_{-1,0}\widetilde{R}_0(1) &&= \widetilde{R}_1(1) \\
m = 1: &\quad 2\widetilde{r}_{0,0}\widetilde{R}_1(1) &&= \widetilde{R}_2(1) + r_{1,0}\widetilde{R}_0(1) \\
m = 2: &\quad 2\widetilde{r}_{1,0}\widetilde{R}_2(1) &&= \widetilde{r}_{0,0}\widetilde{R}_3(1) + \widetilde{r}_{2,0}\widetilde{R}_1(1) \\
m = 3: &\quad 2\widetilde{r}_{2,0}\widetilde{R}_3(1) &&= \widetilde{r}_{1,0}\widetilde{R}_4(1) + \widetilde{r}_{3,0}\widetilde{R}_2(1) \\
&\qquad\quad \vdots && \vdots \\
m \geq 2: &\quad 2\widetilde{r}_{m-1,0}\widetilde{R}_m(1) &&= \widetilde{r}_{m-2,0}\widetilde{R}_{m+1}(1) + \widetilde{r}_{m,0}\widetilde{R}_{m-1}(1)
\end{aligned}$$

The rest of the proof proceeds in a manner similar to the proof of Theorem 6. Set I covers positivity of all terms in (45) by making the right of equation lines labeled by $m = 0, 2, 4, \ldots$ positive, while set II achieves the same by ensuring that right sides of the equation lines labeled by $m = 1, 3, 5, \ldots$ are positive. □

5 FF Scattering-Type Test

The FF test within the scattering-type class of stability tests is the modified Jury test. Jury proposed this test first for a real polynomial in several variations [21,22, p.

104, 23,24] and then for complex polynomials in [27]. His original intention was to devise a stability test that produces explicitly the Schur–Cohn sequence of stability determinants. In all the mentioned references Jury presents the test in a table form (a long standing tradition for stability tests). In [6] they were collected into class C type of tests and converted into polynomial recursions. The next algorithm follows the so-called "prototype for C-type tests" there (which is an arrangement that does not conform exactly with no one of variations in the above cited references).

The MJT algorithm Construct for the complex polynomial $D(z)$ (1) a sequence of polynomials $C_m(z) = \sum_{i=0}^{n-m} c_{m,i} z^i, \quad m = 1, \ldots, n$, as follows:

$$zC_1(z) = d_n^\star D(z) - d_0 D^\sharp(z)\ , \quad q_0 = 1 \tag{49a}$$

For $m = 1, \ldots, n-1$ do:

$$zC_{m+1}(z) = \frac{c_{m,n-m} C_m(z) - c_{m,0} C_m^\sharp(z)}{q_{m-1}}\ , \quad q_m = c_{m,n-m} \tag{49b}$$

The normal condition, for the MJT algorithm are

$$c_{m,n-m} \neq 0 \quad m = 1, \ldots, n \tag{50}$$

Theorem 13 *Assume the MJT algorithm obeys for D(z) in (1) normal condition. Then, $D(z)$ has ν OUC and $n - \nu$ IUC zeros, where ν is given by*

$$\nu = Var\{1, c_{1,n-1}, c_{2,n-2}\ , \ \ldots\ , \ c_{n,0}\} \tag{51}$$

A formal proof for the above theorem was given in [6]. We also proved there (apparently for the first time) that the leading coefficients $c_{m,n-m}$ of this sequence are indeed the principal minors of the Schur–Cohn–Fujiwara matrix (a matrix also known as the unit-circle Bezoutian matrix). Since the Schur–Cohn–Fujiwara matrix was designed to provide a matrix whose inertia corresponds to the distribution of the zeros of $D(z)$ with respect to the unit-circle, once the $c_{m,n-m}$ are recognized to be its principal minors, (51) also follows from a familiar rule to get the inertia of a Hermitian matrix. The relation of the MJT to the Schur–Cohn principal minors led Jury to suggest, already in [27], to use this test to test stability of 2D discrete systems. This application was implemented by Hu and Jury in [20]. This was the first tabular (or "polynomial array") 2D stability test that attained $o(N^6)$ complexity as shown in [11], where it was further simplified into an $o(N^4)$ test carried out by a collection of 1D MJT's that interpolate it.

Theorem 14 *Assume the MJT algorithm is applied to $D(z) \in \mathbb{Z}_j[z]$ then all $C_m(z) \in \mathbb{Z}_j[z]$ and the algorithm can be computed over $\mathbb{Z}_j$. Let $B = \ell(D(z))$ then the integer-size for $C_m(z)$ is*

$$\ell_C(m) = 2mB\ , \quad m = 1, \ldots, n \tag{52}$$

Proof Anderson et al proved in [1] the IP property of the real MJT in [23]. Their proof, that involves manipulation of a unit-circle Sylvester matrix, can be extended to the complex case. An alterative possible proof could be successive substitution till the claimed common factors are identified. The proof that Hu and Jury used in [20] to show that the all entries of the 2D array remain polynomials can be used to deduce this proof from the analogy, mentioned in Sect. 2, between the GIP property of a polynomial recursion and a corresponding recursion of polynomial whose coefficients are the "balanced polynomial." The rate of growth (52) was stated already in [1]. The proof goes like this. Clearly, $\ell_c(1) = 2B$, $\ell_c(2) = 2B$ (no division yet). Then for $m \geq 2$, $\ell_c(m+1) = 2\ell_c(m) - \ell_c(m-1)$ whose solution for the given initial conditions is (52). □

The size of integers in the MJT is similar to that of the FFG or the FFGM tests. The scattering-type algorithm framework does not seem to admit an IP test specific for real polynomials with half-sized integers like that FFR or FFRM tests.

Example 4 We illustrate the algorithm with only the real polynomial $D^{(r)}(z)$ (21). The MJT produces for this polynomial the next sequence of polynomials.

$$
\begin{aligned}
C_1(z) &= 19 + 9z + 24z^2 + 60z^3 + 54z^4 + 37z^5 + 63z^6 \\
C_2(z) &= -136 + 486z + 2640z^2 + 2946z^3 + 2160z^4 + 3608z^5 \\
C_3(z) &= 32496 + 157552z + 174416z^2 + 124752z^3 + 206336z^4 \\
C_4(z) &= 7886560 + 8403680z + 5715360z^2 + 11507360z^3 \\
C_5(z) &= 250221200 - 2459200z + 340326000z^2 \\
C_6(z) &= -19256000 + 4624096000z \\
C_7(z) &= 62827680000
\end{aligned}
$$

$D^{(r)}(z)$ is stable according to Theorem 14 because the leading coefficients of all the polynomials are positive. Notice that this time, the polynomial of the sequence is not symmetric, a feature that characterizes the immittance-type algorithm (and saves computation). It is also possible to compare the integers here with Examples 2 and 3. The integers in Example 2 are slightly shorter and in Example 3 significantly shorter, in accordance with (52) (17) and (24).

6 Binary Complexity

As mentioned earlier, the efficiency of an integer algorithm is measured by its so-called *computing time* [1] or *binary complexity* [2]. The binary complexity regards multiplication of integers like multiplications between power-of-two-based polynomials. Therefore, the task of measuring binary complexity is conceptually similar to counting of the number of standard arithmetic operations in the context of estimating the complexity of tabular 2D stability tests. Namely, at step m of the recursion, it is possible to treat the coefficients of length $\ell(m)$ as polynomials of degree $\ell(m)$ and proceed with count of standard operations. It can readily be realized that the binary complexity of all the FF stability tests is $o(B^2n^4)$. The exact count of binary operations is a bivariate polynomial in the variable B and n. For better resolution, we use

$O\left(\alpha B^2 n^4\right)$ to mean that the leading coefficient of this bivariate polynomial is $\alpha B^2 n^4$ and can reach with relative ease by ignoring at each step of the calculation terms of lower than the leading degree. The next reported counts use the structure of each recursion plus the length values at each step and carry out the implied count of binary operations, taking the binary cost of multiplication of two integers whose size are b_1 and b_2 bits to be $b_1 b_2$ and the cost of exact division of an integer of size $b_1 + b_2$ by an integer of size $b_2 (< b_1)$ as $b_1^2/2$. We also ignore difference between the binary cost of multiplication of two Gaussian or two real integers. With these assumptions, the binary complexities of the MJT algorithm are $O\left(\frac{5}{6} B^2 n^4\right)$. The binary complexity of the FFG and FFGM algorithms is $O\left(\frac{2}{3} B^2 n^4\right)$, and the binary complexity of FFR or FFRM algorithm is $O\left(\frac{5}{48} B^2 n^4\right)$.

7 Concluding Remarks

The paper presented five fraction-free unit-circle stability tests as summarized in Table 1. Three of them, the FFG, FFGM and the MJT tests are applicable to both complex and real polynomials such that they are GIP for Gaussian integer polynomials and IP for real integer polynomials. The FFG and FFGM tests are somewhat more efficient than the scattering-type MJT test. The FFR and FFRM tests are restricted to real polynomials but produce integers of half size by comparison with using a GIP test for a real integer polynomial and are significantly more efficient integer algorithms.

The MJT was applied to test stability of 2D discrete systems and the immittance-type FFG and FFGM tests were conceived in this context. The 2D stability tests appeared first as "tabular tests" (i.e., they look like 1D stability "tables" whose entries are univariate polynomials) in [8,9,27]. These tabular tests have $o(N^6)$ complexity (for a bivariate polynomial of degree (N, N)), a significant reduction of complexity compared to previous tests of exponential order of complexity. These 2D stability tests present an important generalized application for the underlying 1D FF test in which the coefficients are assumed to be polynomials instead of over integers. Roughly, their $o(N^6)$ complexities correspond to the $o(B^2 n^4)$ binary complexity here with B and n replaced by N and the integers sizes here presenting the degree of the polynomial coefficients. In a third generation of 2D stability, called the "telepolation" approach, the FFG, FFGM, and MJT tests were used directly (though not us integer tests) to reduce the overall complexity of 2D stability tests to $o(N^4)$ by interpolations [10,13] and [11]. The use of FFR test to test 2D stability by a tabular test was studied in [14] and by interpolation in [16]. It turns out that 2D stability tests based on real integer tests are less efficient than the use of GIP tests, except that [16] circumvents complex number arithmetics that occurs in more efficient telepolation tests. It follows that 2D stability testing illustrates a situation that is handled better by complex FF stability. The telepolation $o(N^4)$ 2D tests do not compete with the tabular $o(N^6)$ 2D tests in tasks that involve symbolic computation. For example, a tabular 2D stability test can handle better determination of stability range for a bivariate polynomial with one or more designable letteral (i.e., symbolic) parameter.

The role of the FF tests in 2D stability tests illustrates well the usability of the FF tests for polynomials whose coefficients are not integers but belong to an integral domain that behaves like the Gaussian or the real integers.

When it comes to testing 1D systems, especially with symbols, the advantage apparently moves from GIP tests to the real immittance-type FF tests. Determining the stability range for a designable free parameter is handled better with the FFR test (or the FFRM test) [15,17] because shorter integers (compared to GIP tests) mean polynomials (of the symbols) of lower degrees and using Theorems 6 (or 12), the number of constraints remain about n.

An often quoted advantage of integer algorithms is their numerical exactness because integer arithmetic is exact. Each of the presented FF test has a predeterminable bound on the integers size that it may reach. If this bound challenges a size limitation for a computational platform, it can be alleviated by modular arithmetics [1]. It can be argued that a zero location test is not fully reliable because the parameters whose signs are inspected are produced at the end of number crunching that may accumulate numerical inaccuracy. Integer arithmetics can eliminate this concern. It also worths attention that the numerical accuracy of the integer algorithms lends itself to also not integer polynomial because a polynomial with decimal coefficients can be scaled into an integer polynomial).

The FFG test and its further simplification in the real case into the FFR test (and similarly the FFGM and FFRM tests) share this property with the Routh test (used to determine whether all zeros are in the left half of the complex plane). It was shown recently in [18] that there exists a FF Routh test for complex polynomials and a FF test that is specific for real polynomial that runs integers whose size is lower by a factor of two than running the real integer polynomial with the complex FF test (even a parallel to the intermediate FFGr test here is noticed also for the Routh test). This analogy between the immittance-type tests and the lack of a corresponding simplification for real integer polynomials within the scattering-type of tests, strengthens the view stated in [26] that BT forms the discrete equivalent of the Routh test. Actually, the trigger for the work reported in [18] was searching after Routh test equivalents to the immittance-type FF tests in this paper.

Stability tests are closely related to stability criteria that express stability as conditions posed on the signs of a sequence of determinants of sub-matrices of a matrix built from the polynomial coefficients. Best known for the unit-circle stability problem is the Schur–Cohn Fujwara matrix, that was mentioned in the context of Theorem 13, which is a Bezout-type matrix of size n whose positive definiteness is necessary and sufficient for stability. In this case, the relevant determinants are its principal minors. Schur originally considered sub-matrices of a Sylvester-type square matrix of size $2n$. This matrix was afterward arranged in several other forms and some simpler determinants for real polynomials were also studied [25,28]. Premaratne and Jury studied in [30] the relationships between these determinant and the original form of the BT. The current immittance-type FF tests bear some very direct relationships with these determinants as will be explored in some separate article (because their derivation is rather long and involves tools not used in the current presentation). It will be shown that the FFR and FFRM tests for real polynomials and the FFG and FFGM tests for complex polynomials relate in two complementary ways to the unit-circle stability criteria determinants. These relationships provide an additional important perspective to the FF tests in this paper that, like their stability testing capacity, is pertinent for both integer and non-integer polynomials.

References

1. P.G. Anderson, M.R. Garey, L.E. Heindel, Combinational aspects of deciding if all roots of a polynomial lie within the unit circle. Computing **16**, 293–304 (1976)
2. S. Basu, R. Pollack, M.F. Roy, *Algorithms in Real Algebraic Geometry*, 2nd edn. (Springer, New York, 2008)
3. Y. Bistritz, Zero location with respect to the unit circle of discrete-time linear system polynomials. Proc. IEEE **72**(9), 1131–1142 (1984)
4. Y. Bistritz, A circular stability test for general polynomials. Syst. Control Lett. **7**(2), 89–97 (1986)
5. Y. Bistritz, H. Lev-Ari, T. Kailath, Immittance-type three-term Levinson and Schur recursions for quasi-Toeplitz complex Hermitian Matrices. SIAM J. Matrix Anal. Appl. **12**(3), 497–520 (1991)
6. Y. Bistritz, Reflection on Schur–Cohn matrices and Jury–Marden tables and classification of related unit circle zero location criteria. Circuits. Syst. Sig. Process. **15**(1), 111–136 (1996)
7. Y. Bistritz, A modified unit-circle zero location test. IEEE Trans. Circuits. Syst. I **43**(6), 472–475 (1996)
8. Y. Bistritz, Stability testing of two-dimensional discrete linear system polynomials by a two-dimensional tabular form. IEEE Trans. Circuits Syst.I **46**(6), 666–676 (1999)
9. Y. Bistritz, Immittance-type tabular stability test for 2D LSI systems based on a zero location test for 1D complex polynomials. Circuits Syst. Sig. Process. **19**(3), 245–265 (2000)
10. Y. Bistritz, Stability testing of 2D discrete linear systems by telepolation of an immittance-type tabular test. IEEE Trans. Circuits Syst. I **48**(7), 840–846 (2001)
11. Y. Bistritz, Stability testing of two-dimensional discrete time systems by a scattering-type stability table and its telepolation. Multidimens. Syst. Sig. Process. **13**, 55–77 (2002)
12. Y. Bistritz, Zero location of polynomials with respect to the unit-circle unhampered by nonessential singularities. IEEE Trans. Circuits Syst. I **49**(3), 305–314 (2002)
13. Y. Bistritz, On testing stability of 2D discrete systems by a finite collection of 1D stability tests. IEEE Trans. Circuits Syst. I **49**(11), 1634–1638 (2002)
14. Y. Bistritz, Real polynomial based immittance-type tabular stability test for two-dimensional discrete systems. Circuits Syst. Signal Process. **22**(3), 255–276 (2003)
15. Y. Bistritz, An efficient integer-preserving stability test for discrete-time systems. Circuits Syst. Sig. Proces. **23**(3), 195–213 (2004)
16. Y. Bistritz, Testing stability of 2D discrete systems by a set of real 1D stability tests. IEEE Trans. Circuits Syst. I **51**(7), 1312–1320 (2004)
17. Y. Bistritz, Critical stability constraints for linear discrete systems and their efficient evaluation. IEEE Trans. Circuits Syst. II **53**(2), 95–99 (2006)
18. Y. Bistritz, Optimal fraction-free Routh tests for complex and real integer polynomials. IEEE Trans. Circuits Syst. I **60**(9), 2453–2464 (2013)
19. A. Cohn, Über die Anzahl der Wurzeln einer algebraischen Gleichung in einem Kreise. Math. Zeit. **14**, 110–148 (1922)
20. X. Hu, E.I. Jury, On two-dimensional filter stability test. IEEE Trans. Circuits Syst. II **41**(7), 457–462 (1994)
21. E.I. Jury, Further remarks on a paper 'Über die Wurzelvelteilung von linearen Abtastsystemen', by M. Thoma [8], Regelnungstednik, **2**, 75–79 (1964).
22. E.I. Jury, *Theory and Application of the Z-transform Method* (Wiley, New York, 1964)
23. E.I. Jury, A modified stability table for linear discrete systems. Proc. IEEE **53**, 184–185 (1965)
24. E.I. Jury, Inners approach to some problems of system theory. IEEE Trans. Autom. Control **16**, 233–240 (1971)
25. E.I. Jury, *Inners and the Stability of Linear Systems* (Wiley, New York, 1982)
26. E.I. Jury, M. Mansour, On the terminology relationship between continuous and discrete systems criteria. Proc. IEEE **73**(4), 884 (1985)
27. E.I. Jury, Modified stability table for 2D digital filter. IEEE Trans. Circuits Syst. **35**, 116–119 (1988)

28. E.I. Jury, A note on the modified stability table for linear discrete time system. IEEE Trans. Circuits Syst. **38**(2), 221–223 (1991)
29. M. Marden, *The Geometry of the Zeros of a Polynomial in the Complex Plane*, 2nd edn. (American Mathematical Society, New York, 1966)
30. K. Premaratne, E.I. Jury, On the Bistritz tabular form and its relationship with the Schur–Cohn minors and inner determinants. J. Frankl. Inst. **330**(1), 165–182 (1993)
31. I. Schur, Über Potenzreihen, die in Innern des Einheitskreises Beschränkt Sind, Journal für die Reine und Angewandte Mathematik. **147**, 205–232, (1917), and **148** 122–145, (1918). [English translation in I Schur Methods in Operator Theory and Signal Processing, Operator Theory: Advances and Applications, I. Gohberg (ed.) 18 31–88, (Birkhaüer Verlag, 1986).].