Self-Adaptive Mesh Generator for Global Complex Roots and Poles Finding Algorithm

Sebastian Dziedziewicz, Malgorzata Warecka, *Graduate Student Member, IEEE*, Rafal Lech, *Senior Member, IEEE*, and Piotr Kowalczyk, *Member, IEEE*

***Abstract*— In any global method of searching for roots and poles, increasing the number of samples increases the chances of finding them precisely in a given area. However, the global complex roots and poles finding algorithm (GRPF) (as one of the few) has direct control over the accuracy of the results. In addition, this algorithm has a simple condition for finding all roots and poles in a given area: it only requires a sufficiently dense initial grid. However, in practice, this requirement may turn out to be very difficult to implement. For a complex and sophisticated function, the use of a regular high-density mesh may be ineffective or even impossible due to limited computational resources. In this article, a method for creating a self-adaptive initial mesh for this algorithm is presented. The proposed solution uses gradient calculation to identify areas that require mesh refinement, including areas where a zero and a pole are in close proximity. The adaptive mesh allows for faster and more accurate analysis of functions with a much smaller number of samples. As shown in the numerical examples, this approach reduces the number of function calls by several orders of magnitude, and thus significantly reduces the computational time.**

***Index Terms*— Complex zeros and poles, graphene, numerical computations, root finding algorithm, roots of complex equation, self-adaptive mesh, zero/pole analysis of microwave filters.**

I. Introduction

MOST of the electromagnetic problems regarding guiding and resonant structures or radiation and propagation phenomena are formulated in a complex domain. Such notation simplifies the analysis and the interpretation of the involved parameters: propagation coefficient [1], [2], [3], [4], [5], [6], resonance frequencies [7], [8], antenna input impedance [9], material characterization [10], and so on. The evaluation of these parameters often boils down to finding a complex root of a specific function, which evaluation may require various computer resources. The problem of complex root calculation can also be found in many other fields of science such as elementary particles physics [11], solving nonlinear Schrodinger equation [12], spectroscopy [13], acoustics [14], optics [15], [16], and automatic stability analysis [17].

There are many algorithms that search for the roots (zeros) and poles (singularities) of complex functions: local algorithms such as Newton's [18], Muller's method [19], tracking techniques [20], [21], [22], [23], and a lot of global methods [16], [24], [25], [26], [27]. They differ in application, efficiency, and effectiveness, and each of them has its own limitations. Local algorithms, as well as tracking algorithms, require preliminary knowledge of the root location. Most of the global algorithms are based on the Cauchy argument principle (CAP), i.e., an integral over the boundary of the area in which the zeros are searched for. Since the 1960s, a lot of different algorithms involving CAP were developed [28]. However, a problem arises when there are the same number of zeros and poles inside the considered domain and, as a result, it is indistinguishable from the absence of zeros and poles. On the other hand, one of the most effective tools for decades, the ratdisk algorithm [26], and its modification the adaptive Antoulas–Anderson (AAA) algorithm [29], use rational interpolation. Nevertheless, these methods and the CAP methods do not control the accuracy of the results: increasing the number of samples does not have a clear effect on the accuracy of the results, as shown in [30]. The algorithm also provides artificial zero-pole solutions (Froissart doublets) which require extra verification.

Recently, the global complex roots and poles finding algorithm (GRPF) [30] with its modification [31] based on function probing over the considered domain [32] has gained great popularity [1], [2], [10], [11], [13], [14], [33], [34], [35]. Unlike other algorithms based on CAP and rational interpolation, GRPF (despite its numerical complexity) is very universal and allows for having full control over the precision of the results obtained. The main limitation of this algorithm is the initial mesh, which is uniform with a given number of nodes where the values of the function are determined. However, the correct selection of nodes in the initial mesh requires preliminary analyses, and, as with all algorithms, it is difficult to guarantee that all roots and poles are found.

In 2018, Michalski [36] proposed a variation of algorithms [21], [22], [30], [32], based on tracking changes in the sign of the real and imaginary parts of a function. However,

Manuscript received 24 August 2022; revised 10 January 2023; accepted 15 January 2023. This work was supported in part under Ministry Subsidy for Research for Gdansk University of Technology. *(Corresponding author: Rafal Lech.)*

The authors are with the Department of Microwave and Antenna Engineering, Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, 80-233 Gdansk, Poland (e-mail: sebastian.dziedziewicz@pg.edu.pl; malgorzata.warecka@pg.edu.pl; rafal.lech@pg.edu.pl; pio.kow@gmail.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMTT.2023.3238014.

Digital Object Identifier 10.1109/TMTT.2023.3238014

the problem remains in the areas where the zeros and poles are close to each other (the so-called "islands"), which the algorithm is unable to find. These "islands" are difficult to trace and to locate with CAP-based integral methods.

In this article, we propose a procedure for creating a self-adaptive initial mesh that is the starting point for the GRPF algorithm. As in the previous GRPF algorithm, the user specifies the search domain $\Omega \subset \mathbb{C}$, the expected precision of the results δ, and the number of function discretization points N in the initial mesh. Generally, a higher number of points reduces the risk of missing zeros or poles. We demonstrate that the proposed procedure makes it possible to determine the zeros and poles with the number of nodes N smaller by several orders of magnitude than in the case of the uniform mesh applied in the regular GRPF [30]. The reduction of the number of function calls has a direct impact on the speed and efficiency of the algorithm, especially with more complex functions (e.g., numerically determined). This allows the user to perform a faster, more accurate, and more in-depth analysis of the problem under investigation.

II. Formulation

Before discussing the method of creating a self-adaptive mesh, it is worth briefly recalling the principle of the GRPF algorithm and its limitations.

A. GRPF and Its Limitation

The starting point for the GRPF algorithm [30] and its improved version [31] is the discretization of the tested function $f(z)$ in the user-defined domain $\Omega \subset \mathbb{C}$ using an even mesh consisting of N nodes $z_n \in \Omega$. At this point, only the phase of the function ϕ at these vertices is analyzed: $f(z_n) = f_n = |f_n|e^{j\phi_n}$ (the function can be normalized with respect to the modulus). The division of the domain into triangular elements and the definition of edges are based on Delaunay triangulation [32].

Subsequently, the function phase is digitized to four levels

$$Q_n = \begin{cases} 1, & 0 \le \phi_n < \pi/2 \\ 2, & \pi/2 \le \phi_n < \pi \\ 3, & \pi \le \phi_n < 3\pi/2 \\ 4, & 3\pi/2 \le \phi_n < 2\pi \end{cases} \quad (1)$$

and the phase shift along all edges is analyzed. A new node is added in the middle of the candidate edges, i.e., those along which the phase change exceeds two quadrants. At this point, the procedure requires the calculation of the function values and the modification of the triangulation. In the next steps, the mesh consequently becomes denser around the areas where the zeros or poles are located up to accuracy δ (as assumed by the user). In the last stage, the candidate regions are verified – based on the discretized version of the CAP [37], [38], the nature of the area is determined, whether it is a zero or a pole, and of which order. Also, during the mesh refinement process, mesh conditioning is verified: if a triangle is formed with a disproportionately large side ("skinny"), an extra point is added on the longest edge of the triangle to improve mesh quality [39], [40].

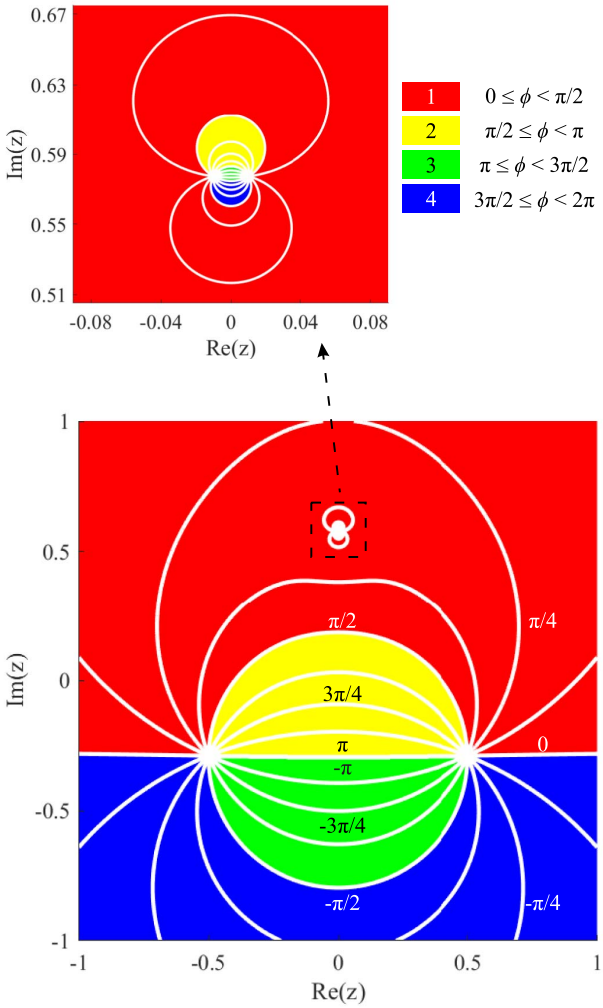

Fig. 1. Phase portrait of the function (2) for $\epsilon = 0.01$. The white curves represent isophases.

The condition for GRPF to find all zeros and poles is, in theory, simple and unambiguous: a sufficiently dense initial mesh must be utilized, i.e., for $N \to \infty$, all zeros and poles can be determined. However, in practice, implementing this assumption is very difficult, especially when there are zero-pole pairs ("islands") located in the area under consideration. Unfortunately, functions of this type often occur in microwave problems: filters [36], guides [30], and resonators [41]. If the initial mesh does not have candidate edges in a given area, the mesh refinement process does not start, and thus the algorithm will not detect a zero or a pole in this area.

Fig. 1 shows a phase portrait of an example function

$$f(z) = \frac{(z - z_a)(z - z_b - \epsilon)}{(z - z_c)(z - z_b + \epsilon)}$$
$$z_a = \frac{1}{2} - \frac{\sqrt{3}}{6}j, \quad z_b = \frac{\sqrt{3}}{3}j, \quad z_c = -\frac{1}{2} - \frac{\sqrt{3}}{6}j. \quad (2)$$

The function has two single zeros in z_a and in $z_b + \epsilon$, and two poles in z_c and $z_b - \epsilon$. Thus, in the vicinity of z_b, a zero-pole pair ("island") is formed. The points z_a, z_b, and z_c are the vertices of an equilateral triangle with the side length $a = 1$ and its center at the origin of the coordinate system.

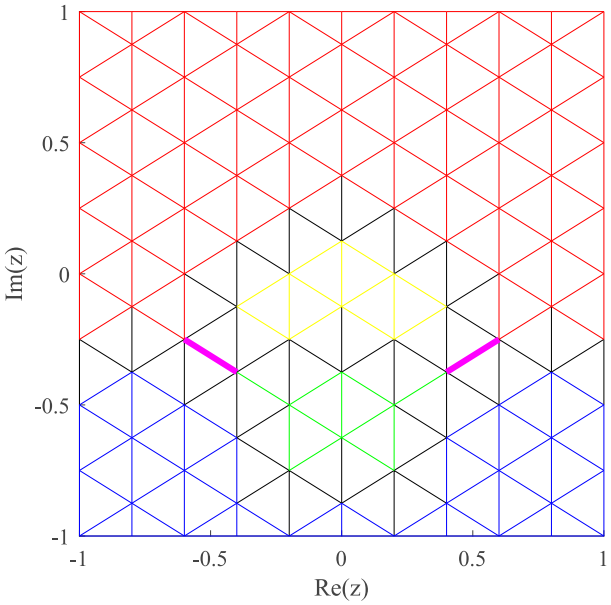

Fig. 2. Initial mesh triangulation of the function (2) with $N = 104$ points for $\epsilon = 0.01$. Two candidate regions are detected (bold magenta lines represent the candidate edges).

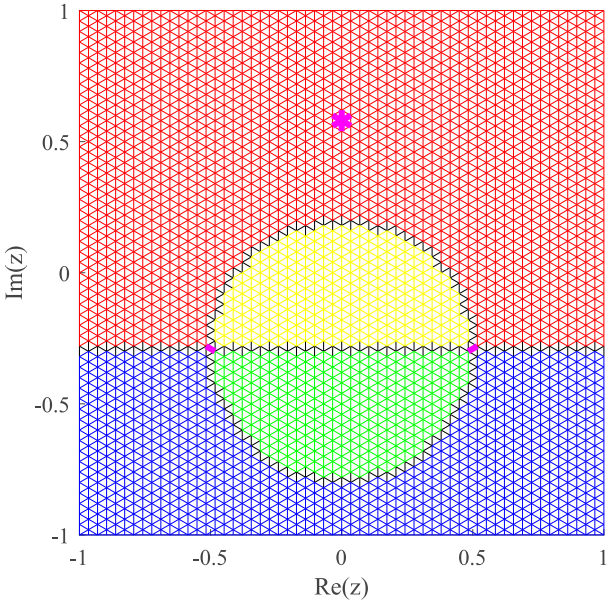

Fig. 3. Initial mesh triangulation of the function (2) with $N = 3038$ points for $\epsilon = 0.01$. Three candidate regions are detected (bold magenta lines represent the candidate edges).

Figs. 2 and 3 show images of a uniformly discretized mesh used in the standard GRPF algorithm with two different discretizations. If we apply an initial discretization with $N = 104$ nodes to function (2) (see Fig. 2), only two candidate regions will be detected: around the zero in z_a and the pole in z_c, while the "island" around z_b will not be located. The mesh refinement process will focus only on z_a and z_c. Ultimately, the GRPF algorithm will locate only these two points with the given accuracy δ, while the zero-pole pair will be omitted in the analysis. GRPF can find this pair, but it requires discretization of the assumed domain using a much larger number of samples: $N = 3038$ (see Fig. 3).

In general, it seems impossible to establish the sampling density a priori for an arbitrary function. For example, if ϵ was reduced to 0.001, the number of samples needed to find the "island" would be in the order of one million, and for $\epsilon = 0.00001$, the number of samples would have to increase to 10 billion. In special cases, the algorithm is able to detect the "island" even with an extremely small number of nodes, but it is a matter of chance related to the fortunate arrangement of the mesh nodes (sometimes it is enough to slightly change the size of the investigated domain or slightly move it).

In conclusion, observing only the value of the too poorly sampled function, as in Fig. 2, does not provide an indication that there may be a zero-pole pair in the region around z_b. Standard CAP-based integral methods also fail because the phase around the "island" varies slightly and the CAP integral is set to zero [37], [38].

B. Generation of a Self-Adaptive Mesh

In this article, we propose a self-adaptive generator for creating a triangular mesh that automatically refines around suspicious regions. The basis of operation for the presented algorithm is the analysis of the behavior of isophases (curves representing a constant level of the function phase; see Fig. 1) in the domain under consideration – in particular, the study of the density and curvature of the isophases.

Let us consider any mesh for which the phase value of the investigated function in its nodes is known. On each triangle ABC, the phase increments along each side are analyzed: $\Delta\phi_{AB} = \phi_B - \phi_A$, $\Delta\phi_{BC} = \phi_C - \phi_B$, and $\Delta\phi_{CA} = \phi_A - \phi_C$. According to CAP, if there are no zeros or poles in the triangle, the sum of the phase increments along the circumference is zero [30], [37], [38]. When there is a zero within the triangle, the phase increment is 2π (or its multiple in the case of multiple zeros). Similarly, for the poles and their multiplicity, it is -2π and a multiple, respectively. However, when the zero-pole pair is found in the triangular region, the sum of the phases is also zero. Therefore, if the mesh is too sparse, finding the "islands" becomes unlikely. Important conclusions can be drawn, however, by examining not only the phase changes along the edges but also the phase gradient over the entire triangular element. Fig. 4 shows the function (2) sampled using the same number of points $N = 104$ as in Fig. 2. As noted above, such a discretization is sufficient to find the zero in z_a and the pole in z_b but too sparse to find the zero-pole pair. However, if we determine an approximate phase gradient representing the direction of the fastest phase growth $\vec{g}_n$ (see Appendix A) for each of the elements, it can be easily noticed that the area around the z_b also distinguishes itself – the black arrows in Fig. 4 show normalized phase gradients of the function. The change of the direction of a gradient in adjacent elements is directly related to a change in the density or curvature of isophases.

It should be noted here that phase gradients can be uniquely determined only for those triangles in which the phases sum to zero on the perimeter. When this condition is not met (e.g., when there is a candidate edge on the perimeter), the gradient cannot be determined unambiguously, and we assign to it the

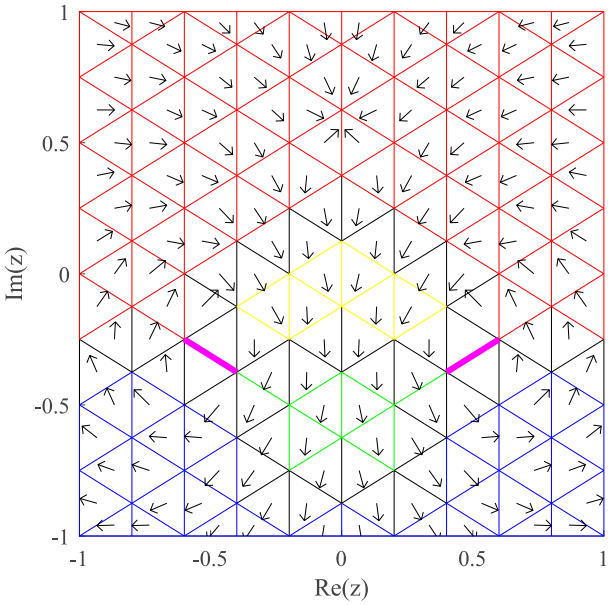

Fig. 4. Initial mesh triangulation of the function (2) with $N = 104$ points for $\epsilon = 0.01$, and arrows representing the average gradients on each mesh element.

zero vector. The angle between the phase gradients for two adjacent triangles can be determined for each inner edge in the mesh (while this angle is assumed to be zero for the edge from the domain boundary)

$$\alpha_m = \measuredangle\{\vec{g}_{n_1}, \vec{g}_{n_2}\}. \tag{3}$$

This angle goes to zero in areas where the phase changes slightly, suggesting the absence of zeros and poles. The closer an edge is to a zero or a pole, the greater the discrepancy between the gradients and the greater the angle between them. One might assume that the edges with the greatest angle should be divided in the next steps. However, such an approach would result in an intensive refinement only in the area found first around the zero or the pole, since these angles become larger the closer we are to zero/pole. The algorithm might not start exploring other areas of the domain.

In the presented approach, we propose that, when refining the mesh, in addition to the angle at each edge, we also take into account its length. Thus, let us define an indicator that determines the risk of a zero or pole occurring in the vicinity of a given edge, as the product of the mentioned angle and the logarithm of the length normalized to the current mesh resolution

$$I_m = \alpha_m \cdot \log\left(\frac{d_m}{D}\right) \tag{4}$$

where length D represents the minimum length of an edge in the current mesh (ultimately $D \to \delta$). This indicator tends to zero when the angle between the gradients tends to zero or when the edge length is reduced to D.

The proposed process of creating a self-adaptive mesh consists of iteratively searching for edges with the highest indicators and dividing them in half – adding new nodes to the mesh. In this way, in subsequent iterations, we will create a mesh adapted to the specific variability of the function. We can start from any mesh (including a regular one) and refine it adaptively in the subsequent steps. It is worth noting that a few points should be enough to start the procedure, e.g., defining the boundary of the domain (e.g., four points for a square or rectangular domain).

TABLE I

MINIMUM NUMBER OF NODES REQUIRED BY GRPF WITH REGULAR INITIAL MESH [30] AND ADAPTIVE INITIAL MESH TO FIND ALL ZEROS AND POLES (INCLUDING "ISLANDS" OF ϵ SIZE) WITH THE GIVEN PRECISION δ. (THE NUMBER OF NODES OF THE INITIAL MESH IS GIVEN IN BRACKETS). (a) GRPF WITH REGULAR INITIAL MESH. (b) GRPF WITH ADAPTIVE INITIAL MESH

(a)

	$\epsilon = 10^{-2}$	$\epsilon = 10^{-3}$	$\epsilon = 10^{-4}$
initial mesh	3,038	1,157,734	28,883,662
$\delta = 10^{-3}$	3,137	1,157,748	-
$\delta = 10^{-6}$	3,323	1,157,931	28,883,805
$\delta = 10^{-9}$	3,561	1,158 077	28,883,971

(b)

	$\epsilon = 10^{-2}$	$\epsilon = 10^{-3}$	$\epsilon = 10^{-4}$
$\delta = 10^{-3}$	704 (673)	922 (919)	-
$\delta = 10^{-6}$	862 (673)	1,054 (935)	3,287 (3,248)
$\delta = 10^{-9}$	1,036 (673)	1,210 (935)	3,817 (3,715)

The curvature of the isophases along the divided edges is also analyzed in each step of the process. If a new node z_M is added to the mesh in the center of the edge based on the nodes z_A and z_B, the value of the function $f(z_M) = f_M = |f_M|e^{j\phi_M}$ is evaluated at this point. One can easily check whether the phase of the function in the new node is within the range of the phases determined by the phases at the ends of the edge. If the condition

$$\min\{\phi_A, \phi_B\} \leq \phi_M \leq \max\{\phi_A, \phi_B\} \tag{5}$$

is not met, the isophases along the edge intersect the edge at least twice – see Fig. 5(b). In this case, it is necessary to further refine this area. For this reason, in the next iteration, all edges attached to node z_M (for which the (5) condition is not met) will be split in half by adding extra nodes. Moreover, in the process, as in standard GRPF, the edges with a phase change exceeding two quarters (candidate edges) and the longest edges in triangles with highly disproportional sides (skinny) are also split in half in each iteration to improve the mesh conditioning.

In summary, the adaptive mesh generation algorithm can be presented in the following steps (see also Fig. 6).

1) At the beginning of the process, the user:
   a) sets the analyzed domain Ω (points delineating the boundary of the domain – standard four points for a rectangular domain);
   b) defines the accuracy with which the results δ are to be determined (similar to the standard GRPF algorithm);
   c) gives the number of nodes (number of function calls) N that will be used to create the initial mesh.
2) The values of the function at the new mesh nodes are calculated (in particular, the phase of the function) and Delaunay triangulation is performed.

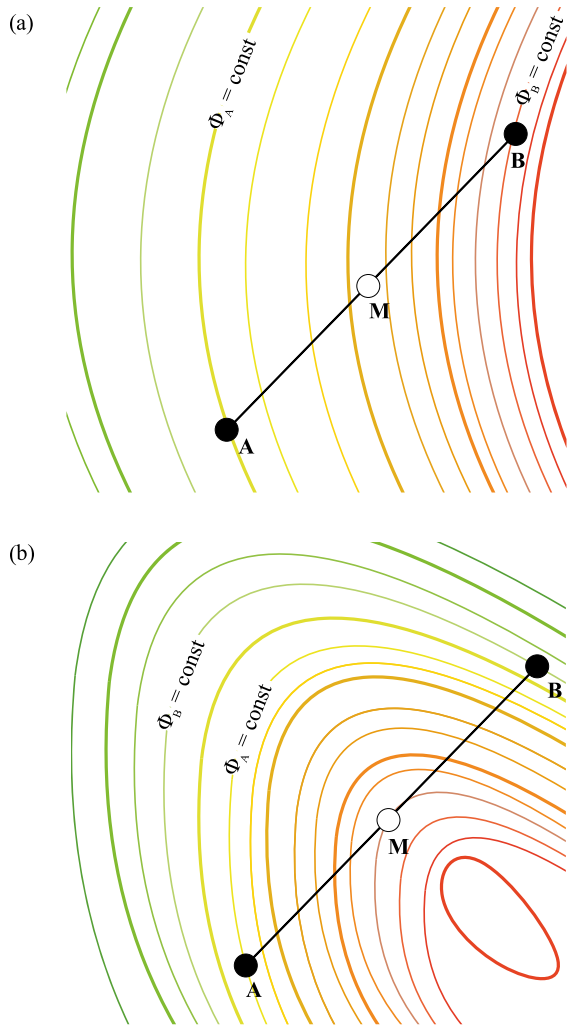

Fig. 5. Isophases along the divided edge: (a) condition (5) is met and (b) condition (5) is not met.

3) The candidate edges (with a double quadrant change) and the edges on which the (5) condition is not met (it does not occur in the first iteration) are determined.
4) For each new nth element (triangle), the gradient of the phase $\vec{g}_n$ is determined, and based on this, the I_m indicator is determined for each mth edge.
5) Edges that will be halved to densify the adaptive mesh are selected. This set includes all edges from point 3 and extra edges from point 4. The edges chosen from point 4 are the edges with the highest indicator in an amount equal to the number of edges from point 3. In the case of a zero indicator, the decisive factor is the length, and if there are no edges in point 3, at least one from point 4 is chosen.
6) The mesh is analyzed in search of skinny triangles, and the longest edges in a given element are split in half.
7) If the total number of nodes in the mesh does not exceed the assumed N, then new nodes are added to the mesh in the middle of the edges determined in points 5 and 6 and the algorithm returns to point 2.

It is worth noting here that the imposed value of N does not have to represent the termination condition of the adaptation process. This article uses this option as a reference to the even distribution of mesh nodes used in [30]. Alternatively, in subsequent iterations, the level of indicator I_n and the number of new nodes not satisfying condition (5) can be analyzed. The adaptation process can therefore be completed (regardless of the imposed number of N nodes) when no more points appear in successive iterations that do not meet condition (5) and the indicator values in the analyzed areas drop below the assumed level. The whole process can also be carried out interactively, where the user observes the mesh refinement on an ongoing basis and allows or rejects the next iteration. The proposed approach seems to be much more convenient than the one described in [30], as it does not require multiple repetitions of the algorithm for increasingly denser meshes.

Fig. 6 shows the steps of creating an adaptive mesh for the discussed function (2) for $\epsilon = 0.01$. Table I(a) and (b) summarize the number of nodes N sufficient to find all zeros and poles in the function (2) depending on the ϵ parameter and the final precision δ of GRPF algorithm. As can be seen, for large "islands", the benefit of using an adaptive mesh is small, but for a smaller ϵ, the number of nodes in the adaptive mesh may be even 10 000 times smaller than for a regular mesh.

III. Numerical Results

This section provides some practical examples of using the self-adaptive mesh generator. It has been shown that, with closely located zeros/poles, the benefits of using the proposed approach can be enormous. When the zeros/poles are separated and evenly distributed within the domain, the use of an adaptive grid is unnecessary and its effectiveness is similar to the use of a uniform grid. However, it is difficult to a priori determine the distribution of zeros/poles for an arbitrary function, so regardless of the variant, it is more advantageous to use the presented adaptive algorithm.

A. Dual-Band Cross-Coupled Eighth-Order Microwave Filter

As the first numerical example, the transfer function of the microwave filter defined by the coupling matrix in [42] was investigated. Such an analysis was also carried out in [36], as a troublesome case, where an "island" appeared, which was a limitation for that algorithm.

The research involved the function describing the scattering parameter of the filter

$$S_{11}(z) = 1 + 2\ j[\mathbf{M} - j\mathbf{R} + z\mathbf{I}_N]^{-1}_{11} \tag{6}$$

where all the parameters are defined in [36] and $z = \sigma + j\omega$. The area of $\Omega = \{z \in \mathbb{C} : -0.4 < \mathrm{Re}(z) < 0.2 \wedge 0 < \mathrm{Im}(z) < 1.2\}$, with the "island", was analyzed (the function is symmetrical to ω, so only $\omega \geq 0$ was considered). The zeros and poles founded are listed below:

1) roots

$$z^{(1)} = 0.000000000 + 0.720326852\,j$$
$$z^{(2)} = 0.000000000 + 0.961957568\,j$$
$$z^{(3)} = 0.000000000 + 0.491541700\,j$$
$$z^{(4)} = -0.000000000 + 0.407949595\,j$$

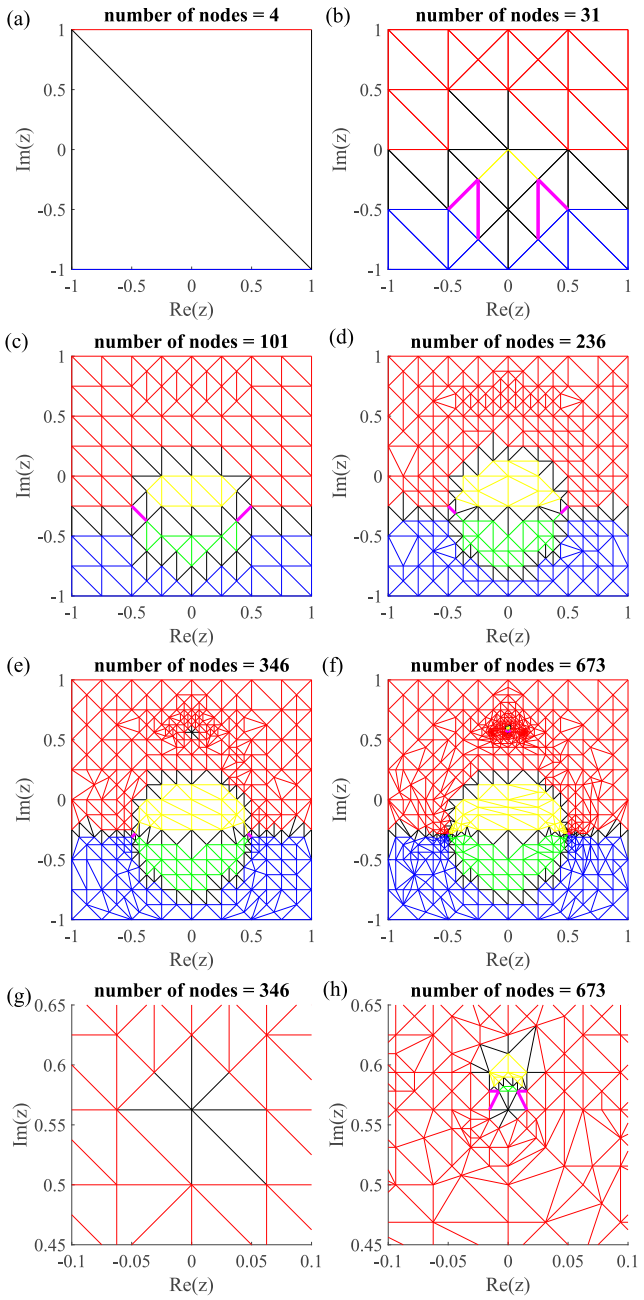

Fig. 6. Iterative process of self-adaptive mesh generation for function (2) with $\delta = 0.001$, $\epsilon = 0.01$. (a)–(f) chosen steps of the process, (g) and (h) magnifications of the "island" areas in steps (e) and (f), respectively.

2) and poles

$$z^{(5)} = -0.128689969 + 0.408634135\,j$$
$$z^{(6)} = -0.324725457 + 0.702182823\,j$$
$$z^{(7)} = -0.167916222 + 1.095438413\,j$$
$$z^{(8)} = -0.022514110 + 0.378089723\,j.$$

As can be seen in Table II, the number of points required for analysis is similar for both methods – the adaptive mesh required 1262 calls, while the regular mesh required 1435 calls. This is mainly due to the fact that the "island" size is not small in relation to the size of the computational domain Ω. However, it is worth pointing out that finding all

TABLE II

MINIMUM NUMBER OF NODES REQUIRED BY GRPF WITH A REGULAR INITIAL MESH [30] AND ADAPTIVE INITIAL MESH TO FIND ALL ZEROS AND POLES WITH THE PRECISION δ FOR A DUAL-BAND CROSS-COUPLED EIGHTH-ORDER MICROWAVE FILTER. THE NUMBER OF NODES OF THE INITIAL MESH IS GIVEN IN BRACKETS

accuracy	GRPF + regular mesh	GRPF + adaptive mesh
$\delta = 10^{-3}$	1,567 (1,435)	1,344 (1,262)
$\delta = 10^{-6}$	1,880 (1,435)	1,656 (1,262)
$\delta = 10^{-9}$	2,253 (1,435)	1,918 (1,262)

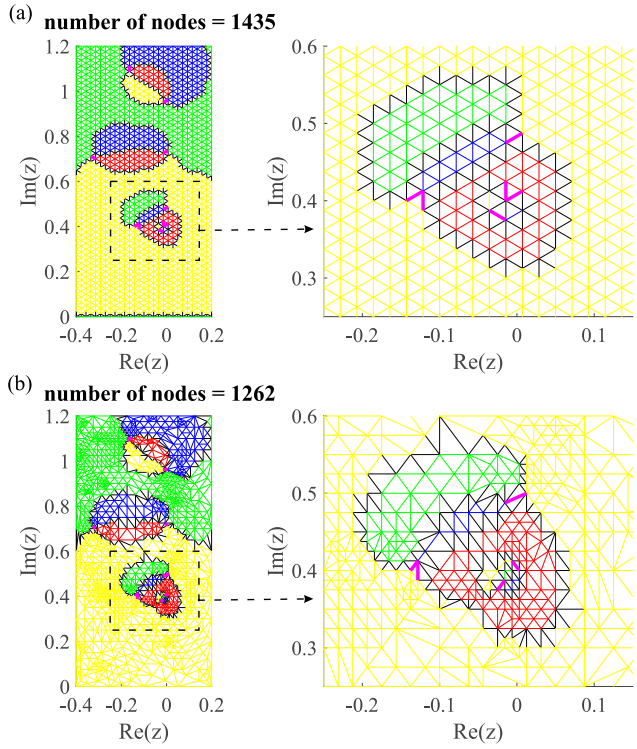

Fig. 7. Initial mesh ($\delta = 0.001$) for dual-band cross-coupled eighth-order microwave filter example: (a) regular and (b) adaptive.

zeros and poles with the use of the minimal regular mesh [30] required a series of tests with different densities. The use of a sparser mesh, than the one shown in Fig. 7(a), will not ensure that all zeros and poles in this region are found. In the adaptive approach, the user can observe the mesh in subsequent steps and reduce or increase N if necessary. Additionally, as can be seen in Fig. 7(b), there are shorter edges in the crucial areas than in the areas with smaller function changes, which may speed up the execution of the next steps of the GRPF algorithm.

B. Fabry–Perot Open Resonator

The second example is a Fabry–Perot open resonator (FPOR) that is widely used to determine the parameters of dielectric materials [41]. The resonator comprised two mirrors – one planar and one spherical – and a dielectric sample that was located between them. The structure was divided into three parts described by individual scattering matrices, which were calculated with the use of a hybrid approach involving mode-matching and the finite element method.

The problem boils down to solving the following equation:

$$\det(\mathbf{I} - \mathbf{S}_P(z)\mathbf{S}_R(z)) = 0 \tag{7}$$

where z represents a resonant frequency, $\mathbf{S}_P$ is a scattering matrix of the planar mirror, $\mathbf{S}_R$ is a cascade of the first and the second sections' scattering matrices, and $\mathbf{I}$ is an identity matrix [41]. By incorporating a lossy material as a dielectric sample into the analysis, the resonant frequency becomes a complex number.

The investigated structure had the following dimensions: radius of the spherical mirror $R = 150$ mm, distance between the planar mirror and the farthest point of the spherical mirror $D = 100$ mm, and aperture diameter of the spherical mirror $D_{\text{ap}} = 200$ mm. The center point of the sample was located at a distance of $ds = 9.6$ mm from the planar mirror, and the sample parameters were as follows: relative permittivity $\varepsilon_r = 5 - 10j$ and the width of the sample was 2 mm. The length of the mirror section analyzed with the finite element method was 40 mm and the number of elements was 4365.

Due to the assumed losses, the search region for the resonant frequency must have involved a positive imaginary part: $\mathbf{\Omega} = \{z \in \mathbb{C} : 20 \cdot 10^9 < \text{Re}(z) < 20.5 \cdot 10^9 \wedge 0.05 \cdot 10^9 < \text{Im}(z) < 0.55 \cdot 10^9\}$. The following roots and poles were found in the considered area:

1) eleven single roots

$$\begin{aligned}
z^{(1)} &= 20.007761298 \cdot 10^9 + 0.204952543\,j \cdot 10^9 \\
z^{(2)} &= 20.021725063 \cdot 10^9 + 0.082217697\,j \cdot 10^9 \\
z^{(3)} &= 20.053612145 \cdot 10^9 + 0.255719944\,j \cdot 10^9 \\
z^{(4)} &= 20.107273419 \cdot 10^9 + 0.112345937\,j \cdot 10^9 \\
z^{(5)} &= 20.146603314 \cdot 10^9 + 0.226088719\,j \cdot 10^9 \\
z^{(6)} &= 20.214322143 \cdot 10^9 + 0.253186318\,j \cdot 10^9 \\
z^{(7)} &= 20.241271881 \cdot 10^9 + 0.161733798\,j \cdot 10^9 \\
z^{(8)} &= 20.349966603 \cdot 10^9 + 0.111563764\,j \cdot 10^9 \\
z^{(9)} &= 20.437490478 \cdot 10^9 + 0.092512809\,j \cdot 10^9 \\
z^{(10)} &= 20.477526173 \cdot 10^9 + 0.251887051\,j \cdot 10^9 \\
z^{(11)} &= 20.499250678 \cdot 10^9 + 0.220886005\,j \cdot 10^9
\end{aligned}$$

2) an additional single root (see the right-hand side of Fig. 8)

$$z^{(12)} = 20.200693325 \cdot 10^9 + 0.480861643\,j \cdot 10^9$$

3) and an additional pole (see the right-hand side of Fig. 8)

$$z^{(13)} = 20.203665927 \cdot 10^9 + 0.479927477\,j \cdot 10^9.$$

The analysis was carried out for three different accuracies, and the efficiency of the processes is presented in Table III and in Fig. 8. The self-adaptive approach requires almost 20 times fewer nodes to find all zeros and poles of the function (7) in the considered region $\mathbf{\Omega}$.

C. Graphene Transmission Line

For the last example, a graphene transmission line was considered [43] (the same example was analyzed in [30] using a regular mesh). The structure comprised a thin graphene layer deposited and a silicone substrate. The normalized propagation coefficient z, for the TM modes, can be found using the following equation:

$$f(z) = \frac{\varepsilon_{r1}}{\eta_0\sqrt{\varepsilon_{r1} + z^2}} + \frac{\varepsilon_{r2}}{\eta_0\sqrt{\varepsilon_{r2} + z^2}} + \left[\sigma_{\text{lo}} - z^2 k_0^2(\alpha_{\text{sd}} + \beta_{\text{sd}})\right] \tag{8}$$

where all parameters are defined in [30]. The same region $\mathbf{\Omega} = \{z \in \mathbb{C} : -100 < \text{Re}(z) < 400 \wedge -100 < \text{Im}(z) < 400\}$ was considered and the pointwise product of four Riemann sheets of the function (8) was analyzed [44].

The problem was solved for various accuracies δ and the results are presented in Table IV and in Fig. 9. The roots and poles obtained for the function in $\mathbf{\Omega}$ are:

TABLE III

NUMBER OF NODES REQUIRED BY GRPF WITH REGULAR INITIAL MESH [30] AND ADAPTIVE INITIAL MESH TO FIND ZEROS AND POLES WITH THE GIVEN PRECISION δ FOR FPOR. THE NUMBER OF NODES OF THE INITIAL MESH IS GIVEN IN BRACKETS

accuracy	GRPF + regular mesh	GRPF + adaptive mesh
$\delta = 10^{-3}$	73,071 (72,935)	4,569 (4,534)
$\delta = 10^{-6}$	73,703 (72,935)	5,104 (4,605)
$\delta = 10^{-9}$	74,099 (72,935)	5,616 (4,605)

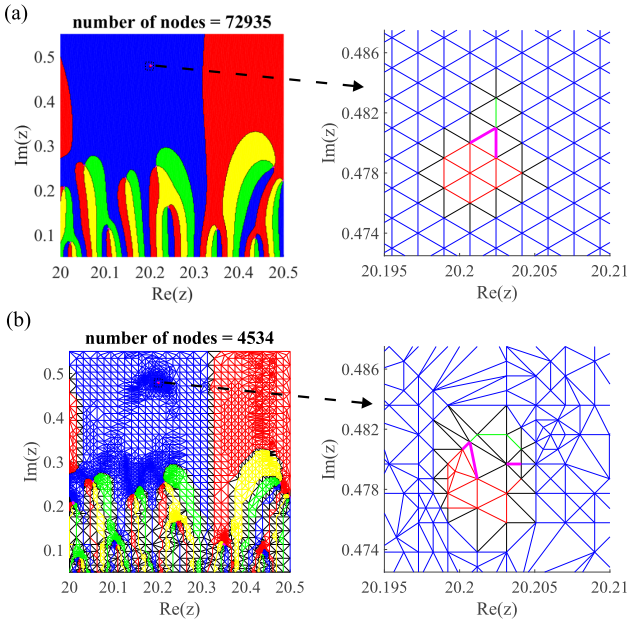

Fig. 8. Initial mesh ($\delta = 0.001$) for FPOR example: (a) regular and (b) adaptive.

TABLE IV

NUMBER OF NODES REQUIRED BY GRPF WITH REGULAR INITIAL MESH [30] AND ADAPTIVE INITIAL MESH TO FIND ZEROS AND POLES WITH THE GIVEN PRECISION δ FOR GRAPHENE TRANSMISSION LINE. THE NUMBER OF NODES OF THE INITIAL MESH IS GIVEN IN BRACKETS

accuracy	GRPF + regular mesh	GRPF + adaptive mesh
$\delta = 10^{-3}$	4,510,733,668 (4,510,733,255)	17,744 (17,608)
$\delta = 10^{-6}$	4,510,734,734 (4,510,733,255)	18,546 (17,728)
$\delta = 10^{-9}$	4,510,735,232 (4,510,733,255)	19,267 (17,728)

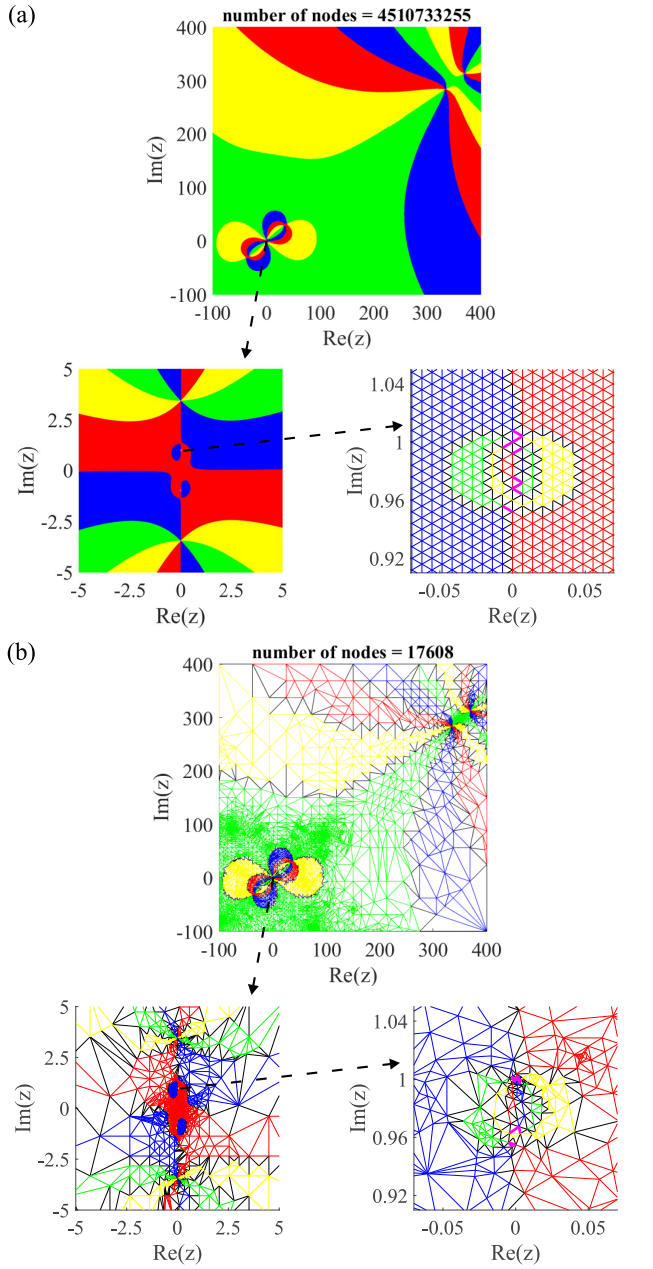

Fig. 9. Initial mesh ($\delta = 0.001$) for graphene transmission line example: (a) regular and (b) adaptive.

1) eight single roots also found in [30]

$$z^{(1)} = -32.101962251 - 27.430861936\,j$$
$$z^{(2)} = 32.101962251 + 27.430861936\,j$$
$$z^{(3)} = -38.177725314 - 32.529521045\,j$$
$$z^{(4)} = 38.177725314 - 32.529521045\,j$$
$$z^{(5)} = 332.744888929 + 282.243079954\,j$$
$$z^{(6)} = 336.220287339 + 285.191091014\,j$$
$$z^{(7)} = 368.439467216 + 312.522078059\,j$$
$$z^{(8)} = 371.007570834 + 314.700407677\,j$$

2) and two poles of second order also found in [30]

$$z^{(9)} = 0.000000000 - 3.449637662\,j$$
$$z^{(10)} = -0.000000000 + 3.449637662\,j$$

3) additional single roots (see Fig. 9)

$$z^{(11)} = -0.003206780 - 0.964810358\,j$$
$$z^{(12)} = 0.003206780 + 0.964810358\,j$$
$$z^{(13)} = -0.004526720 + 0.955901829\,j$$
$$z^{(14)} = 0.004526720 - 0.955901829\,j$$

4) and additional poles (see Fig. 9)

$$z^{(15)} = -0.000000000 - 1.000000000\,j$$
$$z^{(16)} = -0.000000000 + 1.000000000\,j.$$

Due to the use of an adaptive mesh, the algorithm detected two additional zeros and two additional poles in the area under consideration, which were omitted in [30] using a regular mesh. This is not surprising, as it would require the use of a regular mesh with more than four billion nodes. The adaptive mesh needed hundreds of thousands of times fewer nodes.

In some cases, alternative root finding methods provide results faster and with less computer resources. For instance, in the first numerical example, accurate results can be obtained using less than 100 points with the use of AAA [29], while the proposed approach requires more than 1000 points. In this case, the function under consideration is rational and well suited for rational approximation (used in AAA). However, in other cases such as the second and third example, the AAA algorithm does not produce the reliable results. Despite the use of a large number of points on the domain boundary and the decomposition of the domain into smaller subregions, only some of the correct zeros/poles were found and many artificial results were produced.

IV. Conclusion

In the article, we presented a method for creating a self-adaptive initial mesh for the GRPF algorithm. The proposed technique has low numerical complexity and, as a result, the total computation time depends on the analyzed function, and thus consists mainly of the number of function calls. As can be seen from the presented numerical examples, such an approach can significantly reduce the number of nodes (function calls) in the initial mesh and thus the analysis time. In simple examples, where the zeros/poles are well separated, the benefits of utilizing the adaptive mesh are insignificant, but in the case of zeros/poles located close to each other, the efficiency of the algorithm may increase by several orders of magnitude. In summary, the adaptive mesh allows for a faster and more accurate analysis of functions, a good example of which is the graphene transmission line, where the application of the proposed method reduced the necessary number of samples by 250 000 times.

Appendix A
Gradient Calculations

Let us consider a single (nth) triangle of the discretization mesh where function values are calculated in its three vertices $z_A = x_A + jy_A$, $z_B = x_B + jy_B$, and $z_C = x_C + jy_C$, as shown in Fig. 10. Differences between phases of the function values along the triangle edges are denoted as $\Delta\phi_\zeta$, where ζ belong to $\{AB, BC, CA\}$. Thus $\Delta\phi_{AB}$ is the difference between the

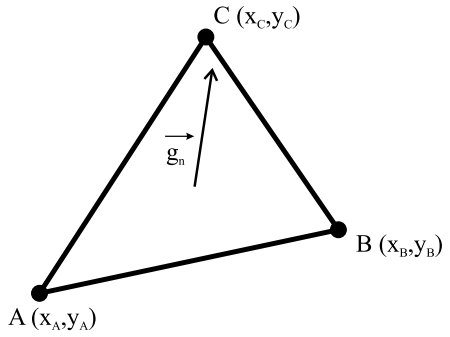

Fig. 10. Average gradient vector $\vec{g}_n$ defined for nth single mesh element.

phase of $f(z_B)$ and $f(z_A)$, and so on. The phase in the triangle can be approximated by a plane function

$$\phi = ax + by + c \tag{9}$$

where a, b, and c are unknown coefficients. The phase gradient (in 2-D) can be expressed by

$$\vec{g}_n = [a, b] \tag{10}$$

and the constant c is not involved – only the increase in phase is meaningful (its exact value is actually ambiguous). Let us assume that the A is the reference point and the phases in points B and C are as follows: $\phi_B = \phi_A + \Delta\phi_{AB}$ and $\phi_C = \phi_A + \Delta\phi_{ac} = \phi_A - \Delta\phi_{CA}$. Substituting $\Delta\phi_\zeta$ and the coordinates of the vertices, we obtain the following system of equations:

$$\begin{cases} \phi_A = ax_A + by_A + c \\ \phi_A + \Delta\phi_{AB} = ax_B + by_B + c \\ \phi_A + \Delta\phi_{ac} = ax_C + by_C + c. \end{cases} \tag{11}$$

If we determine c from the first equation, we can obtain the gradient from a matrix equation

$$\begin{bmatrix} x_B - x_A & y_B - y_A \\ x_C - x_A & y_C - y_A \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \Delta\phi_{AB} \\ \Delta\phi_{ac} \end{bmatrix}. \tag{12}$$

It should be emphasized here that the determination of the gradient is possible only when the phase of the function changes along the successive sides add up to zero

$$\Delta\phi_{AB} + \Delta\phi_{BC} + \Delta\phi_{CA} = 0. \tag{13}$$

Otherwise, for example, when there is a zero or a pole inside the triangle, the sum is $+2\pi$ or -2π, and the gradient cannot be uniquely determined (e.g., the candidate regions in Fig. 4).

APPENDIX B
SOURCE CODE

The source code for the self-adaptive mesh generator can be found at: https://github.com/PioKow/SAGRPF, and it is licensed under the MIT License.

REFERENCES

[1] F. Mesa, G. Valerio, R. Rodriguez-Berral, and O. Quevedo-Teruel, "Simulation-assisted efficient computation of the dispersion diagram of periodic structures: A comprehensive overview with applications to filters, leaky-wave antennas and metasurfaces," *IEEE Antennas Propag. Mag.*, vol. 63, no. 5, pp. 33–45, Oct. 2021.

[2] F. Gasdia and R. A. Marshall, "A new longwave mode propagator for the earth–ionosphere waveguide," *IEEE Trans. Antennas Propag.*, vol. 69, no. 12, pp. 8675–8688, Dec. 2021.

[3] M. Warecka, R. Lech, and P. Kowalczyk, "Hybrid method analysis of unshielded guiding structures," in *Proc. 23rd Int. Microw. Radar Conf. (MIKON)*, Oct. 2020, pp. 398–401.

[4] F. Mesa, R. Rodríguez-Berral, and F. Medina, "On the computation of the dispersion diagram of symmetric one-dimensionally periodic structures," *Symmetry*, vol. 10, no. 8, p. 307, Aug. 2018.

[5] M. Warecka, R. Lech, and P. Kowalczyk, "Propagation in the open cylindrical guide of arbitrary cross section with the use of field matching method," *IEEE Trans. Antennas Propag.*, vol. 66, no. 6, pp. 3237–3240, Jun. 2018.

[6] E. L. Tan and D. Y. Heh, "Application of belevitch theorem for pole-zero analysis of microwave filters with transmission lines and lumped elements," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 11, pp. 4669–4676, Nov. 2018.

[7] R. Lech, "Calculation of resonance in planar and cylindrical microstrip structures using a hybrid technique," *IEEE Trans. Antennas Propag.*, vol. 66, no. 1, pp. 497–500, Jan. 2018.

[8] R. Lech, W. Marynowski, and A. Kusiek, "An analysis of elliptical-rectangular multipatch structure on dielectric-coated confocal and non-confocal elliptic cylinders," *IEEE Trans. Antennas Propag.*, vol. 63, no. 1, pp. 97–105, Jan. 2015.

[9] R. Lech, W. Marynowski, A. Kusiek, and J. Mazur, "An analysis of probe-fed rectangular patch antennas with multilayer and multipatch configurations on cylindrical surfaces," *IEEE Trans. Antennas Propag.*, vol. 62, no. 6, pp. 2935–2945, Jun. 2014.

[10] G. Tytko, "Locating defects in conductive materials using the eddy current model of the filamentary coil," *J. Nondestruct. Eval.*, vol. 40, no. 3, Sep. 2021.

[11] R Aaij, "Study of the lineshape of the $\chi_{c1}(3872)$ state," *Phys. Rev. D.*, vol. 102, Nov. 2020, Art. no. 092005.

[12] I. Chekhovskoy, S. B. Medvedev, I. A. Vaseva, E. V. Sedov, and M. P. Fedoruk, "Introducing phase jump tracking—A fast method for eigenvalue evaluation of the direct zakharov-shabat problem," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 96, May 2021, Art. no. 105718.

[13] L. Y. T. Nguyen, Y. H. Lee, Y. F. Chang, C. C. Hsu, J. Y. Lin, and H. C. Kan, "Subwavelength-resolution imaging of surface plasmon polaritons with up-conversion fluorescence microscopy," *Opt. Exp.*, vol. 30, no. 2, pp. 3113–3124, 2022.

[14] M. K. Schmidt, M. C. O'Brien, M. J. Steel, and C. G. Poulton, "ARRAW: Anti-resonant reflecting acoustic waveguides," *New J. Phys.*, vol. 22, no. 5, May 2020, Art. no. 053011.

[15] H. Wang, C. Wu, T. Jia, and D. Li, "A phase unwrapping method based on sparse depth for depth perception," *Opt. Lasers Eng.*, vol. 109, pp. 60–67, Oct. 2018.

[16] P. Y. Chen and Y. Sivan, "Robust location of optical fiber modes via the argument principle method," *Comput. Phys. Commun.*, vol. 214, pp. 105–116, May 2017.

[17] D. Trofimowicz and T. P. Stefański, "Testing stability of digital filters using optimization methods with phase analysis," *Energies*, vol. 14, no. 5, p. 1488, Mar. 2021.

[18] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables*. New York, NY, USA: US Government Printing Office, 1972.

[19] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in Fortran 77: The Art of Scientific Computing*. Cambridge, U.K.: Cambridge Univ. Press, 1992.

[20] K. S. Gritton, J. D. Seader, and W.-J. Lin, "Global homotopy continuation procedures for seeking all roots of a nonlinear equation," *Comput. Chem. Eng.*, vol. 25, nos. 7–8, pp. 1003–1019, Aug. 2001.

[21] J. J. Michalski and P. Kowalczyk, "Efficient and systematic solution of real and complex eigenvalue problems employing simplex chain vertices searching procedure," *IEEE Trans. Microw. Theory Techn.*, vol. 59, no. 9, pp. 2197–2205, Sep. 2011.

[22] P. Kowalczyk and W. Marynowski, "Efficient complex root tracing algorithm for propagation and radiation problems," *IEEE Trans. Antennas Propag.*, vol. 65, no. 5, pp. 2540–2546, May 2017.

[23] S. Dziedziewicz, R. Lech, and P. Kowalczyk, "A self-adaptive complex root tracing algorithm for the analysis of propagation and radiation problem," *IEEE Trans. Antennas Propag.*, vol. 69, no. 8, pp. 5171–5174, Aug. 2021.

[24] T. Yu-Bo, "Solving complex transcendental equations based on swarm intelligence," *IEEJ Trans. Elect. Electron. Eng.*, vol. 4, no. 6, pp. 755–762, 2009.

[25] P. Gonnet, R. Pachón, and L. N. Trefethen, "Robust rational interpolation and least-squares," *Electron. Trans. Numer. Anal.*, vol. 38, pp. 146–167, Jan. 2011.

[26] A. P. Austin, P. Kravanja, and L. N. Trefethen, "Numerical algorithms based on analytic function values at roots of unity," *SIAM J. Numer. Anal.*, vol. 52, no. 4, pp. 1795–1821, 2014.

[27] M. K. A. Ariyaratne, T. G. I. Fernando, and S. Weerakoon, "A self-tuning modified firefly algorithm to solve univariate nonlinear equations with complex roots," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2016, pp. 1477–1484.

[28] L. M. Delves and J. N. Lyness, "A numerical method for locating the zeros of an analytic function," *Math. Comput.*, vol. 21, no. 100, pp. 543–560, Oct. 1967.

[29] Y. Nakatsukasa, O. Sète, and L. N. Trefethen, "The AAA algorithm for rational approximation," *SIAM J. Sci. Comput.*, vol. 40, no. 3, pp. A1494–A1522, Jan. 2018.

[30] P. Kowalczyk, "Global complex roots and poles finding algorithm based on phase analysis for propagation and radiation problems," *IEEE Trans. Antennas Propag.*, vol. 66, no. 12, pp. 7198–7205, Dec. 2018.

[31] M. Jasinski, S. Dziedziewicz, M. Jozwicka, and P. Kowalczyk, "An improvement of global complex roots and poles finding algorithm for propagation and radiation problems," in *Proc. 13th Eur. Conf. Antennas Propag. (EuCAP)*, Mar. 2019, pp. 1–5.

[32] P. Kowalczyk, "Complex root finding algorithm based on Delaunay triangulation," *ACM Trans. Math. Softw.*, vol. 41, no. 3, pp. 1–13, Jun. 2015.

[33] M. Mul, A. Lamecki, R. Gomez-Garcia, and M. Mrozowski, "Inverse nonlinear eigenvalue problem framework for the synthesis of coupled-resonator filters with nonresonant nodes and arbitrary frequency-variant reactive couplings," *IEEE Trans. Microw. Theory Techn.*, vol. 69, no. 12, pp. 5203–5216, Dec. 2021.

[34] A. H. Orta, M. Kersemans, and K. Van Den Abeele, "A comparative study for calculating dispersion curves in viscoelastic multi-layered plates," *Composite Struct.*, vol. 294, Aug. 2022, Art. no. 115779.

[35] L. Grzymkowski and T. P. Stefanski, "Numerical test for stability evaluation of discrete-time systems," in *Proc. 23rd Int. Conf. Methods Models Autom. Robot. (MMAR)*, Aug. 2018, pp. 803–808.

[36] J. J. Michalski, "Complex border tracking algorithm for determining of complex zeros and poles and its applications," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 12, pp. 5383–5390, Dec. 2018.

[37] P. Henrici, *Applied and Computational Complex Analysis: Power Series Integration Conformal Mapping Location of Zero*, vol. 1. New York, NY, USA: Wiley, 1988.

[38] X. Ying and I. N. Katz, "A reliable argument principle algorithm to find the number of zeros of an analytic function in a bounded domain," *Numerische Mathematik*, vol. 53, nos. 1–2, pp. 143–163, 1988.

[39] M.-C. Rivara, "Mesh refinement processes based on the generalized bisection of simplices," *SIAM J. Numer. Anal.*, vol. 21, no. 3, pp. 604–613, Jun. 1984.

[40] K. Noboru, K. Y. Chung, T. Toshikazu, and J. E. Taylor, "Adaptive finite element methods for shape optimization of linearly elastic structures," *Comput. Methods Appl. Mech. Eng.*, vol. 57, no. 1, pp. 67–89, Aug. 1986.

[41] B. Salski, T. Karpisz, M. Warecka, P. Kowalczyk, P. Czekala, and P. Kopyt, "Microwave characterization of dielectric sheets in a plano-concave Fabry–Perot open resonator," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 5, pp. 2732–2742, May 2022.

[42] R. J. Cameron, C. M. Kudsia, and R. R. Mansour, *Microwave Filters for Communication Systems: Fundamentals, Design and Applications*, 1st ed. Hoboken, NJ, USA: Wiley, 2007.

[43] J. S. Gomez-Diaz, J. R. Mosig, and J. Perruisseau-Carrier, "Effect of spatial dispersion on surface waves propagating along graphene sheets," *IEEE Trans. Antennas Propag.*, vol. 61, no. 7, pp. 3589–3596, Jul. 2013.

[44] P. Kowalczyk, "On root finding algorithms for complex functions with branch cuts," *J. Comput. Appl. Math.*, vol. 314, pp. 1–9, Apr. 2017.

Sebastian Dziedziewicz was born in Elblag, Poland, in 1995. He received the M.Sc.E.E. degree in microwave engineering from the Gdansk University of Technology, Gdansk, Poland, in 2019, where he is currently pursuing the Ph.D. degree.

He is currently with the Faculty of Electronics, Telecommunications and Informatics, Department of Microwave and Antenna Engineering, Gdansk University of Technology. His current research interests include scattering and propagation of electromagnetic wave problems, algorithms, and numerical methods.

Malgorzata Warecka (Graduate Student Member, IEEE) received the M.Sc.E.E. degree from the Gdansk University of Technology, Gdansk, Poland, in 2018, where she is currently pursuing the Ph.D. degree.

She is currently with the Faculty of Electronics, Telecommunications and Informatics, Department of Microwave and Antenna Engineering, Gdansk University of Technology. Her current research interests include scattering and propagation of electromagnetic wave problems, algorithms, and numerical methods.

Rafal Lech (Senior Member, IEEE) was born in Elblag, Poland, in 1977. He received the M.Sc.E.E., Ph.D. (Hons.), and D.Sc. degrees from the Gdansk University of Technology, Gdansk, Poland, in 2001, 2007, and 2018, respectively.

He is currently with the Department of Microwave and Antenna Engineering, Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology. His main research interests are electromagnetic wave scattering, hybrid methods, filter design, complex materials, metamaterial applications at microwave frequencies, electromagnetic analysis of periodic structures, and antenna design.

Piotr Kowalczyk (Member, IEEE) was born in Wejherowo, Poland, in 1977. He received the M.Sc.E.E. degree in applied physics and mathematics, the Ph.D. (Hons.), and D.Sc. degrees in electrical engineering from the Gdansk University of Technology, Gdansk, Poland, in 2001, 2008, and 2018, respectively.

He is currently with the Faculty of Electronics, Telecommunications and Informatics, Department of Microwave and Antenna Engineering, Gdansk University of Technology. His current research interests include scattering and propagation of electromagnetic wave problems, algorithms, and numerical methods.