GDAŃSK UNIVERSITY OF TECHNOLOGY

The author of the doctoral dissertation: Sebastian Dziedziewicz
Scientific discipline: Automatics, Electronics, Electrical Engineering and Space Technologies

DOCTORAL DISSERTATION

Title of doctoral dissertation: Analysis of nonlinear eigenvalue problems for guides and resonators in microwave and terahertz technology

Title of doctoral dissertation (in Polish): Analiza nieliniowych problemów własnych dla prowadnic i struktur rezonansowych w optyce i technice mikrofalowej

Supervisor *signature*
dr hab. inż. Rafał Lech, prof. PG

Gdańsk, year 2023

GDAŃSK UNIVERSITY OF TECHNOLOGY

STATEMENT

The author of the doctoral dissertation: Sebastian Dziedziewicz

I, the undersigned, declare that I am aware that in accordance with the provisions of Art. 27 (1) and (2) of the Act of 4th February 1994 on Copyright and Related Rights (Journal of Laws of 2021, item 1062), the university may use my doctoral dissertation entitled:
Analysis of nonlinear eigenvalue problems for guides and resonators in microwave and terahertz technology
for scientific or didactic purposes.[1]

Gdańsk,..... .....

signature of the PhD student

Aware of criminal liability for violations of the Act of 4th February 1994 on Copyright and Related Rights and disciplinary actions set out in the Law on Higher Education and Science(Journal of Laws 2021, item 478),as well as civil liability, I declare, that the submitted doctoral dissertation is my own work.

I declare, that the submitted doctoral dissertation is my own work performed under and in cooperation with the supervision of dr hab. inż. Rafał Lech, prof. PG.

This submitted doctoral dissertation has never before been the basis of an official procedure associated with the awarding of a PhD degree.

All the information contained in the above thesis which is derived from written and electronic sources is documented in a list of relevant literature in accordance with Art. 34 of the Copyright and Related Rights Act.

I confirm that this doctoral dissertation is identical to the attached electronic version.

Gdańsk,..... .....

signature of the PhD student

I, the undersigned, agree to include an electronic version of the above doctoral dissertation in the open, institutional, digital repository of Gdańsk University of Technology.

Gdańsk,..... .....

signature of the PhD student

[2] [1] Art 27. 1. Educational institutions and entities referred to in art. 7 sec. 1 points 1, 2 and 4–8 of the Act of 20 July 2018 – Law on Higher Education and Science, may use the disseminated works in the original and in translation for the purposes of illustrating the content provided for didactic purposes or in order to conduct research activities, and to reproduce for this purpose disseminated minor works or fragments of larger works.

2. If the works are made available to the public in such a way that everyone can have access to them at the place and time selected by them, as referred to in para. 1, is allowed only for a limited group of people learning, teaching or conducting research, identified by the entities listed in paragraph 1.

GDAŃSK UNIVERSITY OF TECHNOLOGY

DESCRIPTION OF DOCTORAL DISSERTATION

The Author of the doctoral dissertation: Sebastian Dziedziewicz

Title of doctoral dissertation:

Analysis of nonlinear eigenvalue problems for guides and resonators in microwave and terahertz technology

Title of doctoral dissertation in Polish:

Analiza nieliniowych problemów własnych dla prowadnic i struktur rezonansowych w optyce i technice mikrofalowej

Language of doctoral dissertation: English

Supervisor: dr hab. inż. Rafał Lech, prof. PG

Date of doctoral defense:

Keywords of doctoral dissertation in Polish:

nieliniowe problemy własne, zera i bieguny funkcji zespolonych, algorytm znajdowania pierwiastków zespolonych, śledzenia pierwiastków zespolonych, analiza elektromagnetyczna, obliczenia numeryczne, problemy propagacyjne i rezonasowe

Keywords of doctoral dissertation in English:

nonlinear eigenvalue problems, zeros and poles of complex function, complex root finding, complex root tracing, electromagnetics analysis, numerical computations, propagation and radiation problems

Summary of doctoral dissertation in Polish:

Niniejsza rozprawa doktorska przedstawia opracowane narzędzia numeryczne do badania właściwości falowodów i rezonatorów w technologii mikrofalowej i terahercowej. Analiza elektromagnetyczna wymaga rozwiązywania zespolonych problemów własnych, reprezentujących różne parametry fizyczne, np. częstotliwość rezonansową, współczynnik propagacji. Rozwiązywanie równań z wartościami własnymi sprowadza się do znalezienia pierwiastków wyznacznika macierzy. Na początku przedstawiono przykładowe problemy elektromagnetyczne. Następnie zbadano skuteczność i ograniczenia obecnie dostępnych algorytmów szukania pierwiastków zespolonych oraz przedstawiono ulepszenia algorytmu GRPF (Global Complex Roots and Poles Finding). Zaproponowano samoadaptacyjny generator siatki początkowej dla GRPF, który umożliwia szybsze i dokładniejsze znajdowanie zer/biegunów funkcji zespolonych. Ponadto przedstawiono optymalizację metody śledzenia,

która opiera się na samoadaptacyjnej dyskretyzacji argumentu Cauchy'ego. Omówiono również ograniczenia śledzenia i rozwidlenie się charakterystyk. Na koniec przedstawiono nowe podejście do wyznaczania krzywych reprezentujących pierwiastki w funkcji dodatkowego parametru. Zaproponowane metody skracają czas analizy i zwiększają dokładność otrzymywanych wyników. Opracowane narzędzia do rozwiązywania nieliniowych problemów własnych mogą być stosowane w innych dziedzinach inżynierii.

Summary of doctoral dissertation in English:

This dissertation presents developed numerical tools for investigating waveguides and resonators' properties for microwave and terahertz technology. The electromagnetics analysis requires solving complex eigenvalue problems, representing various parameters such as resonant frequency or propagation coefficient. Solving equations with eigenvalue boils down to finding the roots of the determinant of the matrix. At the beginning, one presents examples of electromagnetic problems. The following chapter investigates the effectiveness and limitations of currently available global root-finding algorithms and presents improvements to the Global Complex Roots and Poles Finding (GRPF). The proposed self-adaptive initial mesh generator for the GRPF algorithm enables the faster and more accurate zero/pole finding of complex functions. The optimization of the tracing method is proposed based on the self-adaptive discretization of Cauchy's argument principle. The limitations of tracing and the multipath problem are also discussed. Finally, a new approach to determining the curves representing roots as a function of an extra parameter is presented. The techniques proposed in each chapter reduce the analysis time and enhance the accuracy of the results. The developed tools for solving nonlinear eigenvalue problems can also be applied in other engineering fields.

GDAŃSK UNIVERSITY OF TECHNOLOGY

FACULTY OF ELECTRONICS, TELECOMMUNICATIONS AND INFORMATICS

Analysis of nonlinear eigenvalue problems for guides and resonators in microwave and terahertz technology

mgr inż. Sebastian Dziedziewicz

Supervisor: dr hab. inż. Rafał Lech, prof. PG

Gdańsk, 30.11.2023 r.

„To co budujesz latami, może runąć w ciągu jednej nocy - buduj mimo wszystko."

dr. Kent M. Keith

Podziękowania

Na początku bardzo serdecznie chciałbym podziękować mojemu promotorowi dr. hab. inż. Rafałowi Lechowi, za: super współpracę, poświęcony czas, wsparcie naukowe poparte ogromną wiedzą, pomocną dłoń i cenne wskazówki w czasie badań oraz pisania niniejszej rozprawy.

Serdeczne podziękowania chciałbym również złożyć dr. hab. inż. Piotrowi Kowalczykowi. Prowadzone przez niego w czasie studiów zajęcia z elektrodynamiki oraz koło naukowe Maxwell odmieniły moje życie, kształtując moje życie zawodowe. Dziękuję za przekazaną mi wiedzę matematyczno-fizyczną, która stała się fundamentem tej rozprawy.

Dziękuje!, ale to bardzo bardzo dziękuję moim kochanym rodzicom, babci oraz wszystkim świętym i patronom :) przede wszystkim za: nieocenione wsparcie, wiarę we mnie, dobre słowo, pomoc w sprawach najdrobniejszych jak i w tych beznajdzienych.

W tym miejscu chciałbym też wspomnieć o moich przyjaciołach z pokoju 206 którzy mnie motywowali do pracy; Bartkowi oraz w szczególności mojemu imiennikowi Sebastianowi za bardzo cenne dywagacje i inspirujące pomysły, niekończące się rozmowy oraz trafne wskazówki. Dziękuje również Tomkowi Madejkowi oraz Angelice Golubek za wieloletnie wsparcie.

Dziękuję również szanownym doktorantom z pokoju 725: Maciejowi, Michałowi, za: humor, ciekawe dyskusje, wzajemną pomoc, życząc im obfitości publikacyjnej oraz dalszego rozwoju naukowego. Dziękuje dr. inż Małgorzacie Wareckiej za współpracę i wkład włożony w rozwój algorytmów powstałych w ramach tej pracy. Chcę podziękować także Ani Domagalskiej za dobre słowo, opiekuńczość oraz współpracę w projekcie CMI, który mnie bardzo rozwinął i przygotował do pracy dydaktycznej.

Na koniec chciałbym podziękować wszystkim niewymienionym tutaj osobom, które spotkałem na swojej drodze, gdyż z pewnością także mieli bardzo cenny wpływ na powstanie niniejszej rozprawy.

Streszczenie

W dzisiejszym szybko rozwijający się świecie, ciągła ewolucja technologii i stale rosnąca częstotliwość pracy urządzeń elektronicznych wymagają zaawansowanych narzędzi projektowych, które są zarówno wyspecjalizowanie jak złożone. Analiza elektromagnetyczna wymaga rozwiązywania zespolonych problemów własnych, reprezentujących różne parametry fizyczne, takie jak częstotliwość rezonansową, współczynnik propagacji lub właściwości materiałowe. Rozwiązywanie równań z wartościami własnymi sprowadza się do znalezienia pierwiastków wyznacznika macierzy, co jest podstawowym elementem algebry liniowej. Niemniej jednak, szukanie pierwiastków funkcji zespolonych i rozwiązywanie nieliniowych problemów własnych jest nadal ogromny wyzwaniem.

Niniejsza rozprawa doktorska ma na celu opracowanie narzędzi numerycznych do badania właściwości falowodów i rezonatorów w technologii mikrofalowej i terahercowej. W pierwszym rozdziale przedstawiono przykładowe problemy oraz opisano funkcje analityczne i metody numeryczne do analizy falowodów, struktur rezonansowych i innych systemów. W kolejnym rozdziale zbadano skuteczność i ograniczenia obecnie dostępnych globalnych algorytmów szukania pierwiastków funkcji zespolonych, w szczególności algorytmu GRPF (Global Complex Roots and Poles Finding). Przedstawiono dalsze ulepszenia algorytmu GRPF, które wynikają z przeprowadzonych badań. Zaproponowano samoadaptacyjny generator siatki początkowej dla algorytmu GRPF, który umożliwia szybsze i dokładniejsze znajdowanie zer i biegunów funkcji zespolonych. Ponadto przedstawiono optymalizację metody śledzenia, która opiera się na zastosowaniu samoadaptacyjnej dyskretyzacji zasady argumentu Cauchy'ego. Omówiono również ograniczenia śledzenia i proponowane rozwiązanie problemu rozwidlenia charakterystyk. Na koniec przedstawiono nowe podejście do wyznaczania krzywych reprezentujących zera i bieguny w funkcji dodatkowego parametru. Proponowany algorytm opiera się na uogólnieniu zasady argumentu Cauchy'ego dla przestrzeni zespolonej i rzeczywistej.

Ostatni rozdział jest podsumowaniem rozprawy, przedstawia wnioski i sugeruje możliwe dalsze badania. Zaproponowane w każdym rozdziale metody zmniejszają liczbę wywołań funkcji, skracają czas analizy i zwiększają dokładność otrzymywanych wyników. Opracowane narzędzia do rozwiązywania nieliniowych problemów wartości własnych mają praktyczne zastosowania wykraczające poza problemy elektromagnetyczne i mogą być stosowane w innych dziedzinach inżynierii. Na przykład, podobne równania występują w akustyce, teorii sterowania i optyce. Wszystkie opracowane algorytmy zostały zaimplementowane w środowisku Matlab i są publicznie dostępne na platformie GitHub na licencji oprogramowania typu Open Source.

Abstract

In today's fast-paced world, the continuous evolution of technology and the ever-increasing operating frequency of electronic devices demand advanced design tools that are both sophisticated and complex. The electromagnetics analysis requires solving complex eigenvalue problems, representing various parameters such as resonant frequency, propagation coefficient, or material parameters. Solving equations with eigenvalue boils down to finding the roots of the determinant of the matrix, which is a fundamental element of linear algebra. Nevertheless, finding the roots of complex functions and solving nonlinear eigenproblems is still a significant challenge.

This dissertation aims to develop numerical tools for investigating waveguides and resonators' properties for microwave and terahertz technology. The first chapter presents example problems and describes analytic functions and numerical methods for analyzing waveguides, resonant structures, and other systems. The next chapter investigates the effectiveness and limitations of currently available global algorithms for finding roots of complex functions, notably the Global Complex Roots and Poles Finding (GRPF) algorithm. Further improvements to the GRPF algorithm, which result from the conducted research, are presented. The proposed self-adaptive initial mesh generator for the GRPF algorithm enables the faster and more accurate finding of zero and poles of complex functions. Furthermore, the optimization of the tracing method is presented, which is achieved by applying the self-adaptive discretization of Cauchy's Argument Principle. The limitations of tracing and the proposed solution to the multipath problem are also discussed. Finally, a new approach to determining the curves representing zeros and poles as a function of an extra parameter is presented. This new algorithm is based on generalizing Cauchy's Argument Principle in the analyzed complex and real space. It takes advantage of the features of global algorithms while eliminating problems that occur in tracing.

The last chapter summarizes the dissertation, presents conclusions, and suggests further research. The proposed techniques in each chapter reduce the number of function calls, shorten the analysis time, and enhance the accuracy of the results. The developed tools for solving nonlinear eigenvalue problems have practical applications beyond electromagnetic problems and can be applied in other engineering fields. For instance, similar equations occur in acoustics, control theory, and optics. All the developed algorithms are implemented in a Matlab environment and are publicly available on the GitHub platform under an Open Source Software License.

Contents

List of abbreviations and symbols

Fundamental physical quantities and mathematical expressions

$\mathbf{A}$,$\mathbf{B}$,$\mathbf{C}$, ...	–	matrices, labeled with bold capital letter
$F(\cdot)$	–	function
$\arg(\cdot)$, $\phi(\cdot)$	–	argument, phase of the complex function
$\mathrm{Re}(\cdot)$, $\mathrm{Im}(\cdot)$	–	real, imaginary part of the complex number or function
$\mathbb{C}$, $\mathbb{R}$, $\mathbb{Z}$, $\mathbb{N}$	–	set of complex, real, integer, natural numbers
j	–	imaginary unit
e	–	Euler's number
$\det(\cdot)$	–	determinant of a matrix
λ	–	eigenvalue and also wavelength
Ω	–	range of the domain
Q	–	quadrant
f	–	frequency $[\mathrm{Hz}]$
γ	–	propagation coefficient
ε,ε_e	–	permittivity, relative permittivity
ξ	–	the curve in space $\mathbb{C} \times \mathbb{R}$
δ	–	the accuracy of results, algorithms

The abbreviations used in the thesis

CAP	–	Cauchy's Argument Principle
DCAP	–	Discrete Cauchy's Argument Principle
GRPF	–	Global Complex Root and Poles Finding
SACRTA	–	Self-Adaptive Complex Root Tracing Algorithm
AAA	–	Adaptive Antoulas–Anderson
FEM	–	Finite Element Method
FDM	–	Finite Difference Method
MoM	–	Method of Moments
BEM	–	Boundary Element Method
FPOR	–	Fabry-Pérot Open Resonator
ARRAW	–	Anti-Resonant Reflecting Acoustic Waveguides
CPU	–	Central Processing Unit

Chapter 1

Introduction

The ongoing evolution of technology and increasingly higher frequency bands in wireless communication require even more complex and efficient tools for designing electronic devices. In the low-frequency range, various system modeling methods are used [1]. They are mainly based on circuit models and lumped elements described by Kirchoff's or Ohm's laws [2], [3]. In many cases, especially in high-frequency systems, the approximation by classical circuit theory becomes insufficient [4]. In order to understand the behavior of modern microwave systems, it is necessary to consider electromagnetic phenomena at the macroscopic level as well as quantum phenomena occurring inside the materials used. Therefore, the design of microwave devices and even optical systems enforces electromagnetic analysis involving the direct solution of a system of Maxwell's equations [5]–[10]. It boils down to solving more or less complex differential-integral equations [11], [12] depending on the assumptions constructed. In order to simplify the analysis, some physical quantities are described using phasor algebra and complex numbers [13], [14].

Electromagnetic analysis is an essential step in the design of passive high-frequency devices such as waveguides [15], antennas [16], resonators [17], couplers [18], filters [19], and the like. Electromagnetic simulations create mathematical models describing the physical phenomena in such systems. There are many methods for designing and optimizing devices operating in the microwave or optical spectra. Depending on the problem statement, electromagnetic analysis may require analytical [8], [20], quasi-analytical [21]–[23], discrete [24]–[26], or hybrid [27], [28] methods.

Analytical determination of electromagnetic field distribution is possible only in uncomplicated or exceptional cases (e.g., assuming appropriate simplifications or boundary conditions) [29]. Many times, due to the high complexity of existing microwave systems, it is necessary to use computer-aided design (CAD) techniques [30], especially if the analysis involves nonlinear problems [31]. Typical methods used to solve electromagnetic problems include the finite element method (FEM) [32]–[35], finite difference method (FDM) [25], [36]–[38], or method of moments (MoM) [26], [39]. Analyses can be done in the time or frequency domain [9]. The results may differ depending on the selection of a particular computation method. Determining which method to use could be challenging, and simulations may always be subject to some numerical error. The

implementation of numerical methods, in general, is a complex process. Engineers designing specific devices can use commercial electromagnetic simulators, e.g., ADS [40], HFSS [41], QWED [42], and InventSim [43], which allow full-wave analysis of systems operating at high frequencies. They often allow one to choose an analysis method or use several to solve a given problem. The main advantage of the simulators is their flexibility, while the disadvantage is the high numerical cost, which results in low efficiency in the design and optimization of electromagnetic structures.

In conclusion, in-depth knowledge of electrodynamics and related fields of science, especially the appropriate mathematical instruments for solving differential problems in complex space, is fundamental to the development of modern high-frequency electronics.

1.1 State-of-the-art review

Modeling electromagnetic structures using analytical as well as numerical methods very often leads to the formulation of more or less complicated eigenvalue problems:

$$\mathbf{M}(\lambda)\mathbf{u} = 0 \tag{1.1}$$

where:
$\lambda \in \mathbb{C}$ - eigenvalue,
$u \in \mathbb{C}$ - right eigenvector,
$\mathbf{M} \in \mathbb{C}^{m \times m}$ - operator matrix.

In electromagnetic issues (as shown in later chapters), the eigenvalue can represent, for example, the resonant frequency, propagation coefficient, or material parameters. The eigenvector is often related to the field distribution in the structure. The elements of the $\mathbf{M}$ matrix may linearly or nonlinearly depend on the eigenvalue. For linear eigenproblems, the matrix has the form $\mathbf{M}(\lambda) = \mathbf{A} - \lambda\mathbf{B}$, where $\mathbf{A}$ and $\mathbf{B}$ are matrices of constant entries. These problems often occur in the modal analysis [44] using discrete techniques such as FDM or FEM. In general, the solution of equation (1.1) boils down to finding the roots of the determinant of the matrix $\mathbf{M}$. For matrices of large dimensions (e.g., obtained in FEM), the calculation of the determinant is numerically inefficient or even unworkable. Classical methods to calculate the determinant require significant memory and processor resources. In that case, iterative methods for determining vectors and eigenvalues are used [45]. Unfortunately, these methods fail when the problem becomes nonlinear. There are relatively few techniques in the literature to solve large nonlinear eigenproblems, for example, Beyn's method [46]. Hence, it is necessary to use alternative approaches, including effective and efficient algorithms for finding the roots of complex functions.

For simple quasi-analytical techniques such as a mode-matching method [47], [48], nonlinear problems result from the formulated basis functions (specified field distributions). The matrix $\mathbf{M}$ is constructed by determining the appropriate boundary conditions in the analyzed system. Typically, the size of the problems (matrix dimension) is relatively small, and it is possible to obtain solutions

using appropriate algorithms for finding zeros of the determinant [49]–[57]. In some cases, one variable analysis (purely real or imaginary) is adequate. For more sophisticated approaches, such as spectral methods [58], [59], the function $\mathbf{M}(\lambda)$ is obtained using Green's function [38], [60]–[62]. Nonlinearity in the case of discrete methods [35], [63] is often due to various material models whose parameters may vary, e.g., in a function of frequency. However, as mentioned earlier, solving such problems is very complicated. In general, numerical methods lead to problems of huge size, especially when the geometry structure is inhomogeneous and complicated. In the case of hybrid methods [64], nonlinear problems are created by combining discrete and analytical domains, e.g., with radiation boundary conditions. In contrast to discrete methods, an appropriate problem formulation in hybrid approaches can lead to a relatively small nonlinear problem [17], [22].

It is worth noting that currently available commercial full-wave software does not support these problems. For instance, in simulators based on FEM (e.g., HFSS, CST [65], InventSim), the modal analysis is possible only when material parameters are independent of frequency. Thus, advanced techniques for solving nonlinear eigenproblems have become essential for electromagnetic analysis. Similarly formulated eigenproblems are also common in other fields of science and engineering, such as optics [66], [67], control theory [68], [69], spectroscopy [70], acoustics [71], [72], materials engineering [73], quantum mechanics [74]–[77], and even astrophysics [78].

1.1.1 Methods for solving eigenvalue problems

For linear eigenproblems, the algorithms based on matrix transformations and iterative techniques [79] can be distinguished. One of the oldest iterative techniques for finding the eigenvalues of a matrix is the power method [80]. In its assumptions, it is to find the largest absolute eigenvalue and the corresponding eigenvector. The method convergence strongly depends on the spectral properties of the matrix. This procedure is sufficient only for a narrow class of problems and is part of more complicated algorithms. Other methods for solving linear eigenproblems are algorithms in Krylov subspace, the Arnoldi algorithm [81], the Jacobi algorithm [82] (intended only for symmetric matrices), or the Lanczos method [83]. Among the algorithms based on matrix transformations, the most popular is the QR factorization method [84].

Theoretical analysis and numerical methods for polynomial eigenvalue problems of the form $\mathbf{M}(\lambda) = \mathbf{A}_0 + \mathbf{A}_1\lambda + \mathbf{A}_2\lambda^2 + ... + \mathbf{A}_n\lambda^n$ ($\mathbf{A_n}$ are matrices of constant entries, $n \in \mathbb{N}$) usually seek linearization, i.e., embedding the nonlinear eigenvalue problem into a larger linear problem [85]. It is not a perfect approach because the size of the problem increases, which may result in a degeneration of the matrix conditions. Moreover, depending on the available computational memory, the matrix size may be limited for dense matrices. Despite these limitations, such methods are used in practice, especially in the case of sparse matrices [86]–[89].

For problems with strong nonlinearity and a relatively small size of the matrix $\mathbf{M}$, algorithms for root finding of the determinant can be used. Nevertheless, in some cases, the size of the nonlinear problem is large. As a result, the numerical calculation of the determinant may be inefficient and

ineffective. There are alternative integral methods [90] and iterative techniques (discussed in detail in [31]), although their effectiveness and efficiency may still be improved. Hence, one of the critical aspects of modern computational engineering is creating and developing stable and efficient algorithms for finding eigenvalues for dense and large sparse matrices.

1.1.2 Root finding approaches

Frequently, the eigenproblem can be generalized by introducing a holomorphic function defined on a subset of the complex plane. Problem (1.1) boils down to determining the zero of a complex function and can be written in the following form:

$$F(z) = \det(\mathbf{M}(z)) = 0 \tag{1.2}$$

where:

$z \in \mathbb{C}$ - the zero of the function $F(z)$, representing the eigenvalue λ.

There are many methods to solve the abovementioned equation [53], [67], [91]–[93]. These algorithms find functions' roots (zeros) and poles (singularities) in an assumed search region belonging to the function's domain $F(z)$. They differ in efficiency and effectiveness depending on the class of the investigated function and the analyzed area's dimensions. There are two approaches: local methods, based on sampling functions only in a slight part of the domain, and global methods, which investigate the entire region and aim to determine all roots and poles in this area. Both kinds have their advantages but also limitations. The local ones are more efficient than global algorithms, but global ones guarantee greater control over the convergence and accuracy of the solutions found. The most popular local techniques are Newton's method [52] and Mueller's method [51]. In general, verifying the zero of a function is based on Cauchy's argument principle (CAP) [13], according to which the appropriate investigation region is integrated. When there are an equal number of zeros and poles in the domain under consideration, it may not be easy to differentiate them, especially if the roots are close to each other. Therefore, the final verification of the results is complicated, and no algorithms can fully guarantee effectiveness.

When a single root is found, knowledge of its behavior in a function of extra parameters $t \in \mathbb{R}$ is often also required. In such case, the investigated function can be written as a function of two variables:

$$F(z, t) = 0 \tag{1.3}$$

where:

$z = \xi(t)$ - the curve in space $\mathbb{C} \times \mathbb{R}$ represents the roots value.

In electromagnetic wave propagation problems, z can represent the propagation coefficient γ, for example, and an additional parameter can be the frequency f (to determine dispersion characteristics). On the other hand, in resonators, t may be a structure dimension or extra parameter describing some material properties. The primary way to solve equation (1.3) is to carry out separate analyses for different values of the additional parameter. This approach is the so-called

"slicing method," using any standard global algorithm applied independently for different values of the discretized parameter in the range from t_{min} to t_{max}. Although this method guarantees correct results, its efficiency is very low (especially for a very dense discretization of the t parameter). In addition, the disadvantage of this solution is the lack of knowledge of the direct relation between the following roots. An alternative approach is to use root tracing algorithms as a function of an additional parameter t in the real domain [94] and the complex domain [60], [61], [95]. Such methods are much more efficient than the "slicing method," but the solution is limited to only a single curve ξ.

Local methods of finding zeros or poles

There are many methods for finding the zero of a function of a real variable [96]–[101]. Most of these approaches are based on iterative approximation to the root. When the investigated function is continuous, and the zero is a single multiplicity, the root location can be easily determined by checking the change in the sign of the function value in zero close vicinity. However, this method of solution requires prior knowledge of the search range. Subsequent iterations of the algorithm provide a more accurate root approximation until the assumed accuracy is achieved. By evaluating consecutive values of the function, it is possible to determine the algorithm's convergence. Algorithms of this type are also known as local, as the investigation domain is limited. It is undoubtedly a disadvantage because it requires appropriate initial conditions to ensure the algorithm's convergence. The advantages include short procedure and low computational cost, especially when the search domain is sufficiently small.

The basic local methods include bracketing methods (bisection) [99], methods based on the derivative (Newton-Raphson method) [96], [97], or interpolation methods (regula falsi, Muller's method) [101], [102]. Generally, all these algorithms use the intermediate value theorem. It assumes that if a continuous function has values of opposite signs at the endpoints of an interval, there is at least one zero within that interval. Therefore, the value of the function must take on opposite signs at the extreme points of the assumed interval. When the investigated function is rational, other alternative methods, such as Descartes' rule of signs [102], Sturm's theorem [103], or Fourier-Budan theorem [100], yield information about the number of zeros in the given interval, ensuring finding all the roots of real polynomials with guaranteed accuracy.

Generally, in the case of local methods, determining all real zeros in the investigated domain is possible but requires dividing the search area into subdomains. The root finding is more complicated for the complex function. The imaginary part of the function's argument can be considered an additional real variable. Thus, the analysis boils down to solving a two-variable problem $F(z) = F(\mathrm{Re}(z) + j\mathrm{Im}(z)) = 0$. Although there are more or less compiled techniques for finding complex roots, they are usually only utilized for a particular class of functions or are limited to a specific analysis area (e.g., Newton's or Muller's method).

Global methods of complex roots finding

Global algorithms aim to analyze the entire area $\Omega \in \mathbb{C}$ and find all zeros in it. Since this approach requires discretization of the entire region, global methods are more reliable than local and interpolation algorithms. Thus, these methods allow for better accuracy control but at an increased numerical cost. An example of such algorithms is the Global Complex Roots and Poles Finding Algorithm [104] (GRPF), as well as its variants [55], [57], [60], based on separate analysis (tracing) of the sign of the real and imaginary parts of the function. In comparison, interpolation algorithms based on discretization of only the region's boundary are very efficient for simple polynomial functions [105], [106]. Many procedures based on rational approximation have also been proposed [107]–[109]. Although the obtained zeros may strongly depend on changes in the values of the polynomial coefficients, especially of higher orders [110], methods based on the polynomial approximation can be used successfully. However, it is not possible to set the parameter that controls the accuracy of the obtained result. There may also be artificial solutions of zeros and poles (Froissart doubles [111]), which require additional verification. It has been shown that in the general case, the roots of any considered function may have no relation with its polynomial approximation [49].

One of the more efficient algorithms using polynomial interpolation is ratdisk [53]. In the standard version of the algorithm, the computational domain must be a unit circle, which usually forces the use of some additional transformations. The AAA (adaptive Antoulas–Anderson) algorithm [54], another version of ratdisk, allows using a domain with any contour. Despite the improvement of the algorithm, AAA for more complicated functions still does not guarantee finding all roots and does not eliminate artificial solutions. The increasing number of samples does not directly factor into the accuracy of the results. The obtained roots require further validation. Therefore, the results can be used, for example, as initial values for additional local procedures.

Root tracing techniques

Most of the methods proposed above require iterative execution of global and local root finding algorithms to enhance the precision of the zero location. Another class of algorithms is called root tracing, also referred to as the tracking method in the literature. This method was initially introduced in [94] to analyze the functions of two real variables. For continuous real functions, it is well established that changing the sign of the function value is both necessary and sufficient for the existence of a root. Thus, the basic idea of tracing is to systematically evaluate new points and iteratively check the sign changes in the function value. As noted earlier, in practical electromagnetic applications (and other fields as well), the goal is to find the position of a zero or pole on the complex plane as a function of some extra parameter. Consequently, this approach was expanded to the complex domain, leading to the development of many root tracing methods [57], [60], [95]. Typically, the problem then boils down to the three-dimensional space $\mathbb{C} \times \mathbb{R}$, where the sign change of the real function-based method turns out to be ineffective, as demonstrated in

[61]. It was demonstrated that different signs of the real and imaginary parts of the function are necessary but not sufficient for the existence of roots or poles within the considered region. Therefore, verifying the existence of zeros (or poles) in a fixed region is a far more challenging task in the complex domain. There is a more effective method called Simplex Chain Vertices Searching (SCVS) presented in [60], which has a low numerical cost and an extension in [57], [95], [112] However, the region close to the root location must be known to execute these methods. It can be achieved using Complex Border Tracking (CBT), which tracks the complex border between two regions where the function changes sign for real and imaginary parts at two vertices of an equilateral triangle. The candidate regions, where the function yields values with different signs at all three triangle vertices, are considered potential locations for zeros and poles.

On the other hand, using the CAP calculation procedure, whether zeros, poles, or regular points are present in specific regions can be determined. The root tracing algorithm can also be based on CAP [61], a technique commonly used in electromagnetics and control theory. This method is generalized to the $\mathbb{C} \times \mathbb{R}$ space and is discretized with the small number of points (function evaluations) needed to confirm roots' existence. This approach allows for an efficient analysis of the entire root characteristics of an additional parameter. However, the tracing process requires initial points from the global root finding algorithm for a fixed extra parameter value.

In summary, determining the zeros of functions is one of the fundamental elements of linear algebra. Despite being a purely theoretical problem, it has practical applications in various science branches, particularly electromagnetism. Even though many efficient and effective algorithms for global root finding of complex functions have been developed in recent years, this issue is still an open research problem.

1.2 Scope and goal of the thesis

The development of efficient and stable algorithms for finding the roots of complex functions and, consequently, for solving nonlinear eigenproblems remains a considerable challenge. The existing solutions still have not reached sufficient quality and flexibility. The need for further development of current global algorithms, especially regarding their stability and efficiency, has been discerned. There is also a deficiency in implementing proper software packages similar to those available for linear eigenproblems [113]–[115]. These essential tools are inadequate for nonlinear eigenvalue problems. In response to this requirement, **this dissertation aims to develop numerical tools to study the properties of waveguides and resonators applicable in microwave and terahertz technology**. In order to complete this goal, the following theses were put forward:

- The proper generation of the initial mesh and the optimal process of its densification in the GRPF algorithm [104] will improve the efficiency of the procedure and increase the probability of finding all roots in a fixed search region;

- The use of self-adaptive discretization of the CAP condition will reduce the number of function calls in the tracing algorithm [61] without impacting the achieved accuracy of the results;
- Generalization of the GRPF [104] to $\mathbb{C} \times \mathbb{R}$ space will allow us to obtain a full picture of the distribution of zeros and poles of the function $F(z,t)$ in the entire analyzed region $\mathbb{C} \times \mathbb{R}$, eliminating the limitations existing in tracing and the "slicing methods."

The proof of each of the above theses can be found in Chapters 3, 4, 5, and JCR (Journal Citation Reports) publications by the author of this dissertation, respectively [116]–[118].

1.3 Description of the research methodology

The following is a short description of the contents of each chapter.

Chapter 1

In the first chapter, the aim and description of the research problem are presented, with a justification of the subject matter undertaken in the current state of knowledge.

Chapter 2

This theoretical chapter presents a set of example problems (mainly electrodynamic). It will be handy for numerical verification of newly developed or modified algorithms. The fundamental electromagnetic methods are presented for analysis of the waveguides, resonant structures, and other systems presented in the literature, utilizing various functions (analytical and numerical). They boil down to determining the roots of more or less complicated complex functions. It should be emphasized that the presented eigenproblems are not limited to computational electrodynamics.

Chapter 3

The next chapter examines the effectiveness and limitations of currently available global algorithms for finding roots of complex functions, particularly GRPF. Further improvements to the GRPF algorithm [116], [119], which are the result of the author's research, are presented. Numerous numerical tests are presented to confirm the effectiveness of the proposed approaches and modifications.

Chapter 4

This chapter proposes the improvement of the tracing method achieved during the implementation of this work [117], along with the numerical results. The algorithm's limitations and the proposed solution to the multi-curve $\xi(t)$ problem are also shown [120].

Chapter 5

This chapter describes a new approach to determining the $\xi(t)$ curves representing zeros and poles as a function of an extra parameter throughout the analyzed space $\mathbb{C} \times \mathbb{R}$ [118]. A new algorithm is presented along with numerical results confirming the efficiency and stability of the proposed method.

Chapter 6

The final chapter of the thesis includes a summary, conclusions, and possible directions for further research.

All developed algorithms were implemented in the Matlab environment and have open-source code. They are publicly available on the GitHub platform under an Open Source Software License (MIT):

- Self-Adaptive Mesh Generator for GRPF https://github.com/PioKow/SAGRPF
- Self-Adaptive Complex Root Tracing Algorithm https://github.com/PioKow/SACRTA
- GRPF in $\mathbb{C} \times \mathbb{R}$ Domain https://github.com/PioKow/GRPF3D

Numerical tests presented in this thesis were carried out using the Matlab R2021a environment (Windows 10 64-bit) [115] on the Intel(R) Xeon(R) X5680 3.33GHz CPU (Central Processing Unit) and 12GB RAM machine.

Chapter 2

Complex analysis in electromagnetic problems

Electromagnetic analysis is one of the critical aspects in the design of modern high-frequency electronic devices. The circuit, wave, and material parameters of the analyzed microwave and optical engineering systems are usually defined in the complex domain. This notation facilitates the analysis and interpretation of basic physical phenomena and properties of systems. Problems related to propagation, radiation, and resonance are most often represented by propagation coefficient [22], [121]–[125], resonance frequency [26], [58], material parameters or input impedance [126], and the like.

2.1 Complex parameters

The parameters describing any electromagnetic structure's propagation and radiation properties are typically defined in the complex domain. Electromagnetic analysis includes investigating parameters such as propagation coefficient, resonance frequency, antenna input impedance and permittivity. They describe the operating of systems, including materials, and their behavior in electromagnetic fields. In the main, these parameters express variation of a signal's amplitude and phase as an electromagnetic wave. The complex parameters allow a clear understanding of the effects of polarization, dispersion, and material losses, which are fundamental in many practical applications.

Propagation coefficient

The propagation coefficient $\gamma\ [1/\mathrm{m}]$ is one of the most important parameters of electromagnetic analysis. It determines the propagation properties of an electromagnetic wave in a given medium. In general, γ is a complex number representing guided, leaked, complex, or hybrid modes. This coefficient determines the variation of the complex amplitude A of the wave along the direction of propagation $z\ [\mathrm{m}]$ according to the following relation (suppressing time dependence $e^{j\omega t}$):

$$A(z) = A(z_0)e^{-\gamma z} \tag{2.1}$$

where, γ can be written by two components:

$$\gamma = \alpha + j\,\beta \tag{2.2}$$

where:

α - attenuation constant $\mathrm{Re}(\gamma)\ [\mathrm{Np/m}]$,

β - phase constant $\mathrm{Im}(\gamma)\ [\mathrm{rad/m}]$.

The propagation coefficient is a dimensionless change in magnitude or phase per unit length. Both the attenuation and phase constant are important quantities in describing the propagation of an electromagnetic wave in a given medium. The attenuation coefficient represents the rate at which the wave amplitude decreases as it propagates through the medium. It depends on a material's complex magnetic permeability μ and permittivity ε (refractive index n in optics), particularly dielectric losses defined as $\tan\delta$. In contrast, the phase constant represents a phase variation during the wave propagation and is expressed as $\beta = \frac{2\pi}{\lambda}$. Therefore, its value is related to a given medium's wavelength and phase velocity.

Resonant frequency

The complex resonant frequency $f_r\ [\mathrm{Hz}]$ is another critical parameter commonly used when describing resonant systems. For example, resonators are integral components of filters or are utilized in the investigation of material parameters. These structures involve the existence of standing waves resulting from wave reflections. The resonant frequency can be written in the following form:

$$f_r = \mathrm{Re}(f_r) + \mathrm{Im}(f_r) \tag{2.3}$$

This notation considers the actual operating frequency of the device $\mathrm{Re}(f_r)$, that is, the length of the propagated electromagnetic wave and the attenuation of its amplitude, included indirectly in $\mathrm{Im}(f_r)$. The signal attenuation results from energy loss, e.g., from utilizing lossy materials. In other words, this coefficient determines the periodic variability of electromagnetic field or radiation.

Permittivity

The scalar quantity known as permittivity $\varepsilon\ [\mathrm{F/m}]$ characterizes the electrical properties of a given medium. It is determined by dividing the electric displacement field by the electric field strength for isotropic materials. However, the frequency-dependent nature of this parameter in inhomogeneous media can create nonlinear difficulties. Typically, permittivity is a complex number and can be expressed as follows:

$$\varepsilon = \varepsilon' + j\,\varepsilon'' \tag{2.4}$$

where:

$\varepsilon' = \varepsilon_r\varepsilon_0\ [\mathrm{F/m}]$ - lossless permittivity expressed by the product of the permittivity in free space ε_0 and the relative permittivity ε_r,

ε'' $[\mathrm{F/m}]$ - the imaginary part of the permittivity that results from dielectric loss (scatter of heat energy) and the existence of free charges, which are expressed by the conductivity σ $[\mathrm{S/m}]$.

The real component of the permittivity ε', also known as the dielectric constant, characterizes the polarization properties of the medium and the material's ability to store electric charge. The relative permittivity ε_r is a dimensionless quantity that determines how many times the permittivity of a given medium is more than the permittivity of a vacuum (physical constant ε_0). The loss tangent $\tan\delta$ expresses the relation between the real and imaginary parts of the permittivity. It is the ratio of ε'' and ε' (i.e., the angle on the complex plane) that depends on the angular frequency ω and conductivity σ:

$$\tan\delta = \frac{\omega\varepsilon'' + \sigma}{\omega\varepsilon''} \tag{2.5}$$

Both parameters, $\tan\delta$ and ε_r, are fundamental in developing materials for electromagnetic applications and analyzing antennas, waveguides, transmission lines, and so on.

Antenna input impedance

The antenna input impedance Z_{in} $[\Omega]$ is a crucial factor in studying electromagnetic structures. It signifies the impedance of the antenna as viewed by the radio receiver in a high-frequency electronics system. This parameter is a complex number and can be expressed in the following manner:

$$Z_{in}(\omega) = R_{in}(\omega) + j\, X_{in}(\omega) \tag{2.6}$$

where:
R - resistance $[\Omega]$,
X - reactance $[\Omega]$,
ω - angular frequency $[\mathrm{rad/s}]$,

The real part is related to the power radiated by the antenna resulting from heat losses. The antenna input impedance's imaginary part corresponds to a series resonant circuit. It is related to the energy stored in the electric and magnetic fields in the near-field region antenna. A resonance condition must arise for the antenna to receive or emit a signal in the form of an electromagnetic wave. It is met for pulsations (related to the wavelength) at which the resultant reactance $X(\omega)$ equals zero. In such a case, the impedance has only a real part. In general, the value of an antenna's input impedance is affected by such factors as the antenna's geometry, the properties of the materials from which it is constructed, and the operating frequency. A mismatch between the antenna's input impedance and the transmitting (receiving) circuit impedance can result in reflection and signal loss. Therefore, understanding and optimizing the input impedance is essential for efficient and effective wireless communications.

2.2 Overview of electromagnetic analysis methods

Electromagnetic analysis uses various methods to investigate the behavior of electromagnetic fields and their relations with objects. The first fundamental stage is to define the problem by constructing appropriate mathematical equations. Next, depending on the complexity of the problem and the method of describing physical phenomena, analytical, quasi-analytical, and discrete methods can be distinguished. There are also hybrid methods that merge them. This chapter will discuss the most important ones used to solve propagation and radiation problems.

Analytical methods

The basic analytical method used to solve electromagnetic problems is the separation of variables. This method assumes that the solution of a differential equation (e.g., Laplace, Helmholtz equations) can be expressed as a product of functions of various variables. This reduces the differential equation to a set of simpler equations that can be solved independently. This method is valuable for solving partial differential equations where the variables can be separated. The other method is the method of integral equations. This method involves transforming the differential equation into an integral equation. The method of integral equations is practical for solving boundary value problems and can be used to derive Green's function [38], [60]. Green's function is a solution to the differential equation with a point source. Another alternative method is the series expansion approach. This method uses a series of known functions to solve a differential equation. It is handy for solving boundary value problems, where the boundary conditions are known at the boundaries of a region. By expanding the solution of the differential equation in a series, the boundary conditions can be used to determine the coefficients of the series.

There are a variety of analytical methods available, each with its own unique strengths and limitations. The choice of method depends on the problem's nature and the differential equation's complexity. However, it is important to note that these methods are not universally applicable and are typically limited to structures with simple geometries. Analytical techniques work well for analyzing objects such as rectangular prism, cylinders, ellipses, or spheres and for modeling unbounded space. Depending on geometry, the electromagnetic field is described using Bessel, Hankel, or Mathieu functions. Individual field components can be determined using the mode-matching technique [47], [48] or the field-matching method [22] based on determining appropriate boundary conditions. Despite their usefulness, these methods are inflexible and require significant effort to implement specific problems.

Extensions of analytical methods are quasi-analytical procedures, such as the method of moments [26], [39] and spectral domain approach [58], [59]. They follow the assumption of local convergence to an analytical function within a narrow domain range, which is achieved through numerical techniques. In practice, they are used when the problem can be approximated by a simpler problem that can be solved analytically. They prove to be especially handy for problems with more complicated geometries or nonlinearities. Nevertheless, they can be computationally

expensive and require significant computing resources. Moreover, they do not perform well when modeling inhomogeneous and anisotropic media.

Discrete methods

With the increase in computer resources, discrete methods have become an alternative to analytical methods. It is used in electromagnetic analysis to solve systems with complex geometries or problems with strong nonlinearity. This class of techniques is based on dividing the analyzed structure (computational domain) into small regions. In each element, the differential or integral equations derived most often directly from Maxwell's equations are solved using numerical methods. The resulting electric or magnetic fields are combined to produce the final approximate solution. Various numerical methods are available for solving electrodynamic problems. Numerous numerical methods exist for solving electrodynamic problems, including the finite difference method (FDM), finite element method (FEM), boundary element method (BEM), and method of moments (MoM). In numerical electromagnetic methods, time-domain (FDTD, TDFEM) and frequency-domain (FDFD, FEM, MoM) approaches are used to solve electromagnetic problems. This time domain approach helps analyze transient phenomena like the propagation of electromagnetic waves in a dispersive medium or the response of particular excitation. On the other hand, the frequency domain approach is useful for analyzing steady-state phenomena, such as the behavior of the systems' resonance structures, antennas, or scattering parameters.

Finite difference methods transform differential equations, which may also be nonlinear, into a system of linear equations resolved by standard linear algebra approaches. This technique is based on the approximation of derivatives through finite value differences. A specific grid discretizes the computational domain (space and time), and the field's value at these discrete points is approximated by solving linear equations (e.g., the Gauss-Jordan method) containing function values from nearby points. Although this method is very efficient, the resulting field vectors electric $\vec{E}$ and magnetic $\vec{H}$ are shifted from each other in the discretization grid, making it difficult to set appropriate boundary conditions.

The finite element method is a numerical technique used to solve differential equations. It involves approximating the field vectors electric $\vec{E}$ and magnetic $\vec{H}$ in a discrete set of points called nodes by linear interpolation. To discretize the computational domain, a mesh is created using triangular elements for two-dimensional structures and tetrahedral elements for three-dimensional structures. Higher-order curvilinear elements can also be used to improve accuracy. The problem is solved by solving a system of equations created from a linear combination of functions that approximate the differential equations. Next, the boundary conditions are defined, and the Galerkin method can be utilized to obtain a solution in terms of the mean value at the nodes. The solution's accuracy improves as the number of elements increases.

FEM is a versatile computational method that is widely used in commercial software such as HFSS. However, it requires significant computational costs for the optimal discretization of the computational domain. Therefore, it is better suited for solving small and medium-sized struc-

tures. Unlike analytical methods, modeling open spaces using FEM is problematic. While it is possible to use a perfect matching layer (PML) [127] to solve this issue, it comes with additional difficulties. PML is based on placing an absorption layer at the boundaries of the computational domain through a lossy medium that absorbs incident waves. This approach increases the size of the computational domain, and selecting anisotropic parameters is not straightforward.

Another type of discrete method is based on solving integral equations. The boundary element method reduces the analysis to a boundary problem that can be solved directly without discretizing the whole domain. It can result in computational complexity reduction, especially for large problems. The obtained electric or magnetic fields are combined to produce a final approximate solution. BEM applies to problems for which Green's functions can be computed.

The method of moments is a discrete technique that relies on quasi-analytical field representation. It involves a sum of basic functions, which are either charges or currents, and these functions are defined on the surface of the structure being analyzed. The solution to the problem is achieved by satisfying the boundary conditions for the electric and magnetic fields at each point on the surface. To determine the unknown coefficients of the basis functions, a system of linear equations is solved. This method is computationally efficient and requires less memory than other numerical methods. However, it may not be ideal for problems involving a large number of unknowns or very thin structures.

Discrete methods offer various advantages but can be computationally expensive and time-consuming. Additionally, they require a careful focus on numerical errors and convergence. The selection of a particular method depends on the geometry of an object, the type of problem, and the desired level of accuracy. For instance, FDM suits problems with simple geometries and uniform materials. In contrast, FEM is better suited for problems with more complex geometries and heterogeneous materials. BEM is useful for problems with infinite boundaries, while MoM is commonly used for problems involving metal and dielectric structures.

Hybrid methods

Hybrid techniques represent a method of modeling electromagnetic structures that capitalizes on the benefits of commonly used approaches while minimizing their drawbacks. They involve dividing the computational domain into a few sub-domains (it may also be an infinite open space) and analyzing them separately using methods adjusted to a given sub-region. Through appropriate modeling of boundary conditions, it is possible to connect sub-domains analyzed by various methods, e.g., analytically and discrete based on the impedance matrix [28], [128]–[130]. The smaller parts of the domain analyzed using discrete methods, the more efficient the procedure is.

The examples of a hybrid procedure are demonstrated in [130], [131]. It involves analyzing the scattering parameters of single or multiple objects using a multimode impedance matrix. The investigated structure, such as a resonator, a waveguide segment, or any other arbitrary scattering system, is assumed to be replaced by an n-port network, the so-called "black box." Its parameters are determined by the corresponding scattering or impedance matrix. In this approach, the

analysis is based on the definition of the electromagnetic wave relations between the input and output of the system. In other words, a multi-port is formed and expressed using an impedance matrix. It is defined in theoretical regions such as a cylinder or sphere (generalizing in any convex volume) surrounding the analyzed structure [47]. The impedance matrix is the relationship between the electric and magnetic fields inside the system without considering the external incident wave. This approach allows for the analysis of propagation in various variants, e.g., closed waveguides and resonators, but also scattering in open regions, e.g., antenna structures. This solution makes it possible to use different analysis methods for individual parts of the system [28], [132]. For example, geometries with curvilinear shapes can be analyzed by discrete techniques and other parts by analytical methods, significantly reducing analysis time and computational resource requirements. Furthermore, after calculating the impedance matrix, the structure's rotation or arbitrary shift can be performed without needing discrete analysis again [28]. In addition, the hybrid technique can be applied to multi-object analysis that considers reciprocal coupling between elements, which can be modeled using iterative procedures [133]. Another notable benefit of using impedance matrices in hybrid methods is the ability to study open structures. Unlike other solutions that require radiation boundary conditions and encounter related issues, the approach employing connecting impedance matrices (discrete and analytical parts) takes place outside of the structure.

2.3 Example problems

This section contains example structures in which the fundamental issue is wave propagation, mainly electromagnetic. Therefore, the essence is determining the propagation coefficient or its variability resulting from the analysis parameters, e.g. material properties. The following issues are analyzed:

- a lossy multilayered waveguide [20]
- a coaxially loaded cylindrical waveguide [8]
- a graphene transmission line [134]
- an anti-resonant reflecting acoustic waveguides [72]
- a triangular dielectric fiber [22]
- a cylindrical-rectangular microstrip resonant structure [58]
- a Fabry-Pérot open resonator [17]
- a dual-band cross-coupled eighth-order microwave filter [135]

The functions describing the problems are defined analytically (mode-matching technique) or numerically for the above structures, e.g. the determinant obtained from the spectral domain approach, FEM or field matching method.

2.3.1 Lossy multilayered waveguide

Multilayered lossy structures are widely used in microwave applications [56], [136]. They are components of various high-frequency systems, such as couplers or optical devices like lasers. As an example, a planar waveguide is analyzed [15]. Its structure consists of three layers: air, polystyrene, and silver, as shown in Figure 2.1.

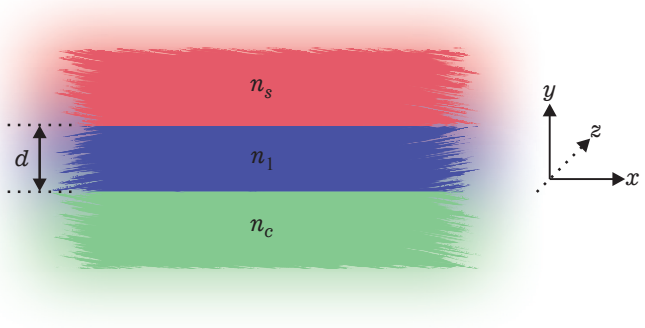

Figure 2.1: The cross section of the structure. It is assumed that electromagnetic wave propagates in the z direction.

For such a structure, the analysis boils down to the analytical solution of the wave equation assuming specific boundary conditions. It is already a well-known linear problem, and many investigations have been made to solve it [20], [137]. The objective is to determine the normalized complex propagation coefficient γ for a set propagation frequency f (related to the wavelength λ_0) associated with mods: TE and TM. Hence, it is essential to calculate the roots of the following determinant:

$$F(\gamma) = \det\left(\begin{bmatrix} 1 & -\cos(k_0 d\kappa_1) - \frac{\gamma_c \sin(k_0 d\kappa_1)}{\kappa_1} \\ j\gamma_s & -j\kappa_1 \sin(k_0 d\kappa_1) + j\gamma_c \cos(k_0 d\kappa_1) \end{bmatrix}\right) \tag{2.7}$$

where:

$k_0 = \frac{2\pi}{\lambda_0}$ - free-space wavenumber $[1/\mathrm{m}]$,

$\kappa_1 = \sqrt{n_1^2 - \gamma^2}$ - $[\cdot]$,

$\gamma_s = \sqrt{\gamma^2 - n_s^2}$ - $[\cdot]$,

$\gamma_c = \sqrt{\gamma^2 - n_c^2}$ - $[\cdot]$.

The analysis used typical material parameters according to [20]. The refractive indices $n_s = 0.065 - j4$, $n_1 = 1.5835$, $n_c = 1$, define dielectric layers. The thickness of the central layer is $d = 1.81\ \mu\mathrm{m}$. It is assumed that the length and width of the other layers go to infinity (see Figure 2.1).

2.3.2 Coaxially loaded cylindrical waveguide

A coaxial cylindrical waveguide loaded with a dielectric is analyzed in the following example. The structure is shown in Figure 2.2, and it is a well-known electromagnetic problem [8], [29] that aims to determine the normalized propagation coefficient γ. In this structure, the propagating complex modes are described by Bessel functions. The distribution of the electromagnetic field in such a line can be determined analytically using the mode-matching technique. In order to ensure the continuity of the fields at the dielectric-air boundary, the following determinant must be equal to zero:

$$F(\gamma) = \det\left(\begin{bmatrix} -J_1 & 0 & J_2 & Y_2 & 0 & 0 \\ 0 & J_1 & 0 & 0 & -J_2 & -Y_2 \\ -\frac{\gamma m J_1}{a\kappa_1^2} & -j\frac{\eta_0 J_1'}{\kappa_1} & \frac{\gamma m J_2}{a\kappa_2^2} & \frac{\gamma m Y_2}{a\kappa_2^2} & j\frac{\eta_0 J_2'}{\kappa_2} & j\frac{\eta_0 Y_2'}{\kappa_2} \\ -\frac{j\varepsilon_r J_1'}{\kappa_1\eta_0} & -\frac{\gamma m J_1}{a\kappa_1^2} & j\frac{J_2'}{\kappa_2\eta_0} & j\frac{Y_2'}{\kappa_2\eta_0} & \frac{\gamma m J_2}{a\kappa_2^2} & \frac{\gamma m Y_2}{a\kappa_2^2} \\ 0 & 0 & J_3 & Y_3 & 0 & 0 \\ 0 & 0 & \frac{\gamma m J_3}{b\kappa_2} & \frac{\gamma m Y_3}{b\kappa_2} & j\frac{\eta_0 J_3'}{\kappa_2} & j\frac{\eta_0 Y_3'}{\kappa_2} \end{bmatrix}\right) \tag{2.8}$$

where:

$J_1 = J_m(k_0\kappa_1 a)$, $J_2 = J_m(k_0\kappa_2 a)$, $J_3 = J_m(k_0\kappa_2 b)$,

$Y_1 = Y_m(k_0\kappa_1 a)$, $Y_2 = Y_m(k_0\kappa_2 a)$, $Y_3 = Y_m(k_0\kappa_2 b)$,

$J_m(\cdot)$ - Bessel functions of order m (prime denotes their derivatives),

$Y_m(\cdot)$ - Neumann functions of order m (prime denotes their derivatives),

m - angular varability of the modes $[\cdot]$,

$\kappa_1 = \sqrt{\gamma^2 + \varepsilon_r}$ - $[\cdot]$,

$\kappa_2 = \sqrt{\gamma^2 + 1}$ - $[\cdot]$,

$k_0 = \frac{2\pi f}{c}$ - free-space wavenumber $[1/\mathrm{m}]$,

$\eta_0 = 120\pi$ - impedance of free space $[\Omega]$.

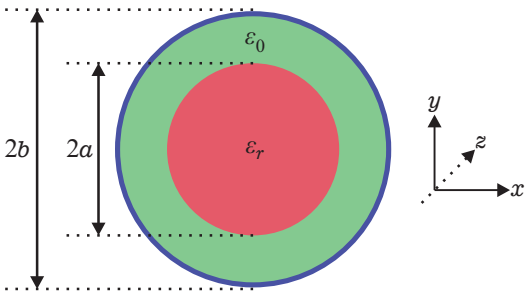

Figure 2.2: The cross section of the structure. It is assumed that electromagnetic wave propagates in the z direction.

The following parameters are used during the numerical tests: core radius $a = 6.35$ mm, outer dielectric radius $b = 10$ mm, dielectric layer permittivity $\varepsilon_r = 10$, and transverse field distributions for mode $m = 1$.

2.3.3 Graphene transmission line

Another example involves the analysis of a simple transmission line. It comprises a thin graphene layer, air, and a silicon substrate (homogeneous medium). For the structure shown in Figure 2.3, the propagation of surface waves and dispersion relations for the TM and TE modes are analyzed. A model describing the propagation properties in tensor conductivity of graphene is presented in [134]. It is derived from the semi-classical Boltzmann transport equation [138].

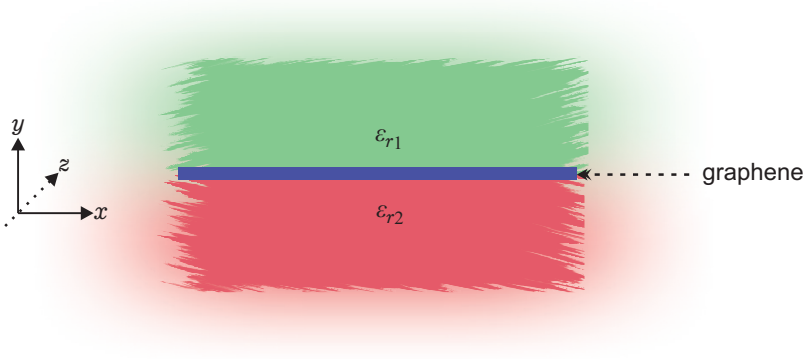

Figure 2.3: The cross section of the structure. It is assumed that electromagnetic wave propagates in the z direction.

The normalized propagation coefficient γ for the TM modes can be obtained by calculating the zero of the following function:

$$F(\gamma) = \frac{\varepsilon_{r1}}{\eta_0\sqrt{\varepsilon_{r1}+\gamma^2}} + \frac{\varepsilon_{r2}}{\eta_0\sqrt{\varepsilon_{r2}+\gamma^2}} + \left[\sigma_{lo} - \gamma^2 k_0^2(\alpha_{sd}+\beta_{sd})\right] \tag{2.9}$$

In addition, the conductivity tensors depend on the frequency as follows:

$$\begin{gathered}\sigma_{lo} = j\frac{q_e^2 k_B T}{\pi\hbar^2(2\pi f - j\tau^{-1})}\ln\left[2\left(1+\cosh\left(\frac{\mu_c}{k_B T}\right)\right)\right] \\ \alpha_{sd} = \frac{-3v_F^2\sigma_{lo}}{4(2\pi f - j\tau^{-1})^2} \qquad \beta_{sd} = \frac{\alpha_{sd}}{3}\end{gathered} \tag{2.10}$$

where:

$k_0 = \frac{2\pi f}{c}$ - free-space wavenumber $[1/\mathrm{m}]$,

$\eta_0 = 120\pi$ - impedance of free space $[\Omega]$,

q_e - elementary charge $1.602176634 \cdot 10^{19}$ $[\mathrm{C}]$,

$k_B = 1.380649 \cdot 10^{-23}$ - Boltzmann's constant $[\mathrm{J/K}]$,

$T = 300$ - temperature $[\mathrm{K}]$,

$\tau = 0.135$ - $[\mathrm{ps}]$,

$\mu_c = 0.05 q_e$ - $[\mathrm{C}]$,

$v_F = 10^6$ - $[\mathrm{m/s}]$.

This model is accurate for low propagation constant values and considers spatial dispersion and substrate effects for low THz ranges. Therefore, analysis will be performed for such frequency ranges. The dielectric layers are air $\varepsilon_{r1} = 1$ and silicone with $\varepsilon_{r2} = 11.9$.

2.3.4 Anti-resonant reflecting acoustic waveguide

Another example is an acoustic waveguide. According to [72], ARRAW (Anti-Resonant Refecting Acoustic Waveguides) inholds with core and anti-resonant outer and inner cladding. A schematic of the cylindrical structure of alternating silicon/silica/silicon layers is shown in Figure 2.4. The core of radius $a = 1\mu\mathrm{m}$ is made of the same material as the outer cladding and allows propagation of longitudinal (P) and shear (S) waves with velocities higher than the inner cladding. The silica thickness d can be changed as an additional parameter in the analysis. In this example, the purely torsional modes are investigated, dependent only on shear velocity. Propagation along the z-axis is characterized by a constant propagation coefficient throughout the waveguide structure.

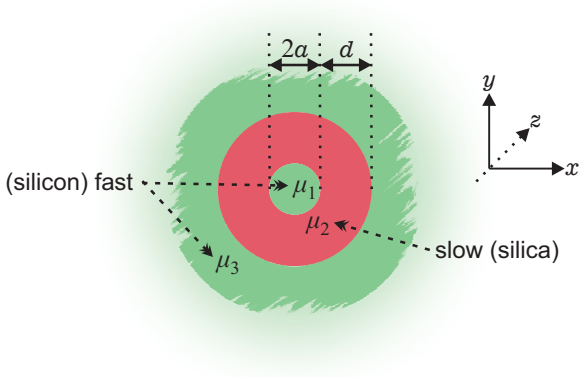

Figure 2.4: The cross section of the structure. It is assumed that accustic wave propagates in the z direction.

The elastic equation of motion describes the waveguide modes for such a structure:

$$\rho \frac{d^2}{dt^2}\vec{u} = \nabla \cdot \mathbf{T} \tag{2.11}$$

where:

ρ - density $[\mathrm{kg/m^3}]$,

$\vec{u}$ - displacement field $[\mathrm{m}]$,

$\mathbf{T}$ - stress tensor $[\mathrm{N/m^2}]$.

The problem (2.11) is solved analytically using cylindrical coordinates (r, θ) and time-harmonic dependency:

$$\vec{u}_z(r, \theta, z, t) = U_z(r, \theta)e^{j(\beta z - \Omega t)} \tag{2.12}$$

The displacement field is analyzed as a function of the longitudinal wavenumber for only shear mods:

$$k_s^2 = \frac{\Omega^2}{v_s^2} - \beta^2 \tag{2.13}$$

where:

k_s - transverse wavenumber $[1/\mathrm{m}]$,

β - longitudinal wavenumber $[1/\mathrm{m}]$,

Ω - angular frequency $[\mathrm{rad/s}]$,

v_s - shear velocity $[\mathrm{m/s}]$.

In each layer $i = \{1, 2, 3\}$, outgoing and incoming waves can be determined $\hat{u}^i = \begin{bmatrix} u_+^i & u_-^i \end{bmatrix}^{\mathrm{T}}$. The appropriate matrices of coefficients describe them as follows:

$$\mathbf{M}_r^i = \begin{bmatrix} -H_1 & -J_1 \\ \mu_i k_s H_2 & \mu_i k_s J_2 \end{bmatrix} \tag{2.14}$$

where:

μ_i - shear modulus of the material for layer i $[\mathrm{Pa}]$,

$H_1 = H_0^{'}(k_s^i r)$, $H_2 = 2H_0^{''}(k_s^i r) + 2H_0(k_s^i r)$,

$J_1 = J_0^{'}(k_s^i r)$, $J_2 = 2J_0^{''}(k_s^i r) + 2J_0(k_s^i r)$,

$J_m(\cdot)$ - Bessel functions of order m,

$H_m^{(1)}(\cdot)$ - Hankel functions of order m (prime denotes their derivatives).

By taking into account the boundary conditions of each of the layers and the symmetry of the system, the matrix equation is constructed:

$$\hat{u}^3 = (\mathbf{M}_{r_1}^1 {\mathbf{M}_{r_2}^1}^{-1} \mathbf{M}_{r_2}^2 {\mathbf{M}_{r_3}^2}^{-1}) \hat{u}^1 = \mathbf{S}^{all} \hat{u}^1 \tag{2.15}$$

For the resonance condition to be met, the fields in the first and last medium must only be outgoing: $\hat{u}_-^1 = \hat{u}_+^3 = 0$, this condition will be fulfilled when:

$$F(\beta) = \mathbf{S}_{22}^{all}(\beta) = 0 \tag{2.16}$$

This example shows that a universal and efficient algorithm for root finding of complex functions can be applied not only in electrodynamic problems, but also in various fields of modern engineering.

2.3.5 Triangular dielectric fiber

Another investigated structure is a dielectric waveguide as a triangular fiber. The cross-section of the inner dielectric layer consists of an equilateral triangle with rounded corners, as shown in Figure 2.5. For such a structure, it is possible to analyze the propagation of electromagnetic waves using the field-matching technique [22]. This quasi-analytical method is described in detail in [63]. It does not require using the Green function, discretizing the computational domain, and implementing absorbing boundary conditions. Only the continuity condition for the tangential components of the field on the surface of two areas $i = \{1, 2\}$ is necessary. The longitudinal components of the electric and magnetic fields, respectively, $F_z = \{E, H\}$ in both regions, have the following form (suppressing time dependence $e^{j\omega t}$):

$$\begin{aligned} F_z^1 &= \sum_{m=-M}^{M} A_m^F J_m(k_0 \kappa_1 \rho) e^{jm\phi} e^{-\gamma z} \\ F_z^2 &= \sum_{m=-M}^{M} B_m^F H_m^{(2)}(k_0 \kappa_2 \rho) e^{jm\phi} e^{-\gamma z} \end{aligned} \tag{2.17}$$

where:

(ρ,ϕ,z) - coordinate in the cylindrical system $\rho = \varrho(s)$ and $\phi = \varphi(s)$ $[\mathrm{m}]$,

s - curvilinear coordinate that follows the surface $[\mathrm{m}]$,

A_m^F, B_m^F - unknown fields coefficients in the first and second regions respectively $[\cdot]$,

$J_m(\cdot)$ - Bessel functions of order m,

$H_m^{(2)}(\cdot)$ - Hankel functions of order m,

$k_0 = \frac{2\pi f}{c}$ - free-space wavenumber $[1/\mathrm{m}]$,

$\kappa_i = \sqrt{\gamma^2 + \varepsilon_{ri}}$ - $[\cdot]$

The other field components, F_ϕ, and F_ρ, can be determined directly from Maxwell's equations. Through specifying the boundary conditions of these regions, the following matrix equation is obtained:

$$\begin{bmatrix} \mathbf{M}^{E,1} & -\mathbf{M}^{E,2} \\ \mathbf{M}^{H,2} & -\mathbf{M}^{H,2} \end{bmatrix} \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} = 0 \tag{2.18}$$

A detailed expansion of the individual elements of the matrix $\mathbf{M}$ can be found in [22]. In the proposed approach, the solution is to determine the roots of the determinant $F(\gamma) = \det(\mathbf{M}(\gamma))$, representing the normalized propagation coefficients for leaky and complex modes. The TE_{01}, TM_{01}, and HE_{11} modes are analyzed, for which the parameters are assumed following [22]. The dielectric materials have permittivity: $\varepsilon_{r1} = 8.41$ and $\varepsilon_{r2} = 2.4025$, respectively. The analysis is performed at frequencies $f = 10^{14}$ Hz and $M = 5$. The triangle dimensions are $a = 1.3471$ μm and $r = 0.01a$.

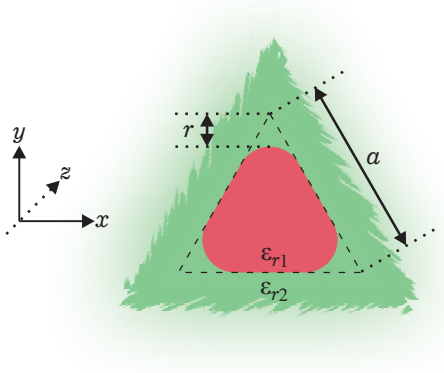

Figure 2.5: The cross section of the structure. It is assumed that electromagnetic wave propagates in the z direction.

2.3.6 Cylindrical-rectangular microstrip resonant structure

Another example is a cylindrical-rectangular microstrip resonator. It is a conformal structure [58], [126], which can find application in systems having curved surfaces, such as airplanes and building facades. The geometry of the structure is shown in Figure 2.6. It consists of a cylinder of radius $r = 200$ mm, a conductive patch, and three dielectric layers. These are an air gap of thickness $s = 5$ mm, a substrate of permittivity $\varepsilon_r = 2.3$ and thickness $h = 2.4$ mm, and open-air spaces

outside the structure. The resonant element is the patch of a perfect conductor placed on the cylindrical surface of the dielectric. The patch has length $L = 80$ mm (along the z-axis) and width $W = 168$ mm (along the ϕ-axis).

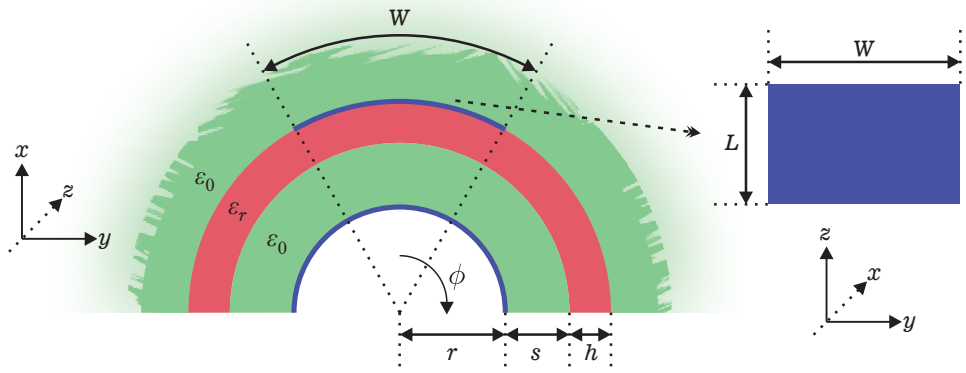

Figure 2.6: The geometry of an investigated cylindrical-rectangular microstrip antenna.

The problem is solved numerically using the spectral domain approach [139]. The analysis is carried out in a cylindrical coordinate system (ρ, ϕ, z). The longitudinal components of electric and magnetic fields, respectively $F_z = \{E, H\}$, in region $i = \{1, 2, 3\}$ and have the following form (suppressing time dependence $e^{j\omega t}$):

$$F_z(\rho,\phi,z) = \frac{1}{2\pi}\sum_{m=-M}^{M} e^{jm\phi}\int_{-\infty}^{\infty} dk_z e^{jk_z z} \times \begin{cases} (H^{\rho}_{m1} - \beta^F_1 J^{\rho}_{m1})A^F_{m1}, & r \le \rho \le r+s \\ H^{\rho}_{m2}A^F_{m2} + J^{\rho}_{m2}B^F_{m2}, & r+s \le \rho \le r+s+h \\ H^{\rho}_{m3}A^F_{m3}, & r+s+h \le \rho \end{cases} \tag{2.19}$$

where:

(ρ, ϕ, z) - coordinate in the cylindrical system $[\mathrm{m}]$,

A^F_{mi}, B^F_{mi} - unknown fields coefficients of the harmonic order m and layer i,

$\beta^E_1 = H^{r1}_{v1}/J^{r1}_{v1}$, $\beta^H_1 = H^{\prime r1}_{v1}/J^{\prime r1}_{v1}$,

$J^{\rho}_{mi} = J_m(k_0\kappa_i\rho)$,

$H^{\rho}_{mi} = H^1_m(k_0\kappa_i\rho)$,

$J_m(\cdot)$ - Bessel functions of order m,

$H^{(1)}_m(\cdot)$ - Hankel functions of order m,

$k_0 = \frac{2\pi f}{c}$ - free-space wavenumber $[1/\mathrm{m}]$,

$\kappa_i = \sqrt{\varepsilon_{ri} - k_z^2}$ - $[\cdot]$.

It is assumed that the patch is an ideal conductor. Therefore, the tangential components of the electric field on its surface are zero. By utilizing this fact, the surface currents can be determined:

$$\sum_{m=-M}^{M} e^{jm\phi}\int_{-\infty}^{\infty} dk_z e^{jk_z z}\mathbf{G}_m(k_z)\tilde{\mathbf{J}}_m(k_z) = 0 \tag{2.20}$$

where:
$\mathbf{G}_m(\cdot)$ - Green matrix of the harmonic order m,
$\tilde{\mathbf{J}}_m(\cdot)$ - vector of the patch surface current densities of the harmonic order m ,
$M = 30$ - number of harmonic order used for matrices.

As can be seen, the problem is formulated as an integral equation in the frequency domain using the vector Fourier transform. A detailed description of the method and the expansion of matrix coefficients can be found in [126]. In order to solve equation (2.20), the Galerkin method of moments can be used, according to which the surface current density on the patch is a linear combination of known basis functions [59], obtaining the following matrix equation:

$$(\mathbf{Z}_{\phi\phi} + \mathbf{Z}_{\phi z} + \mathbf{Z}_{z\phi} + \mathbf{Z}_{zz})\mathbf{I} = 0 \tag{2.21}$$

where:
$\mathbf{Z}_{(\cdot)}$ - submatrix of equations (2.20),
$\mathbf{I}$ - vector of coefficients of the basis functions.

Considering that the wave vector k_z depends on frequency, the problem concerns solving the eigenproblem $F(f_r) = \mathbf{Z}(f_r) = 0$, whose roots are the complex resonant frequencies f_r. It should be noted that due to the orthogonality of eigenfunctions $e^{jm\phi}$, this problem can be solved for any eigenvalue separately.

2.3.7 Fabry-Pérot open resonator

Another example is the Fabry-Pérot open resonator (FPOR). This structure is commonly used to determine the parameters of dielectric materials [17], [140]. The system consists of two mirrors, for which the resonant frequency f_r can be calculated, related to the resonator's dimensions. When a dielectric plate is placed between the mirrors, this resonant frequency shifts depending on the permittivity ε_r of the placed sample. The structure can be analyzed in two configurations: a plane mirror, one spherical mirror (see Figure 2.7), and two spherical mirrors (see Figure 2.8). In both versions, the structure can be divided into three parts described by separate scattering matrices $\mathbf{S}_{(\cdot)}$. From the point of view of the resonator analysis, only axial modes (propagating along the z-axis) are considered. It can be assumed that the resonator walls (along the y-axis) are far enough away and do not influence the field distribution. The scattering parameters $\mathbf{S}_{FEM}$ of the section with a spherical mirror can be calculated using the finite element method. The waveguide section $\mathbf{S}_{WG}$ between the mirror and the sample can be described analytically using the mode-matching method.

Configuration with the plane mirror

The last section can also be expressed analytically. It is a section of waveguide with a dielectric plate $t = 2$ mm thickness. The sample center is a distance of $ds = 9.6$ mm from the plane mirror. The further analysis parameters are the radius of the spherical mirror $R = 150$ mm, the distance between the planar mirror and the farthest point of the spherical mirror $D = 100$ mm, the diameter of the aperture of the spherical mirror (at the same time, the diameter of the waveguide) $D_{ap} = 200$ mm. The length of the section with the spherical mirror is $L_{FEM} = 40$ mm. For the FEM analysis, $N = 4365$ elements are used (the maximum edges length of a triangle in the mesh is 1.2 mm). The lengths of individual sections can be written as $L_P = 2ds$ and $L_{WG} = D - L_P - L_{FEM}$, respectively.

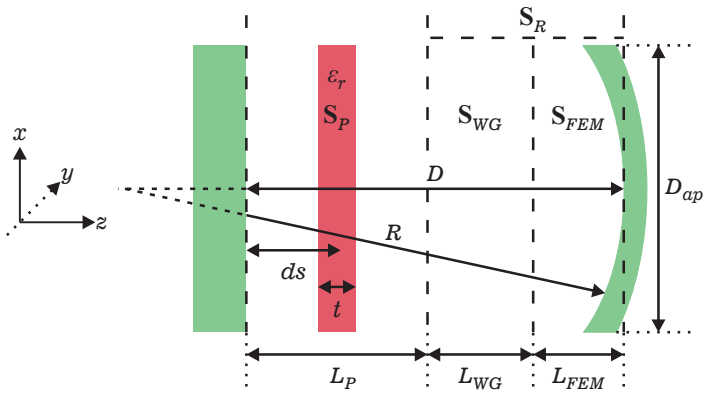

Figure 2.7: The geometry of investigated the resonator with the plane mirror and its divide into sections described by appropriate scattering matrices.

The analysis boils down to finding the roots of the following determinant, dependent on the resonance frequency f_r:

$$F(f_r) = \det(\mathbf{I} - \mathbf{S}_P(f_r)\mathbf{S}_R(f_r)) \tag{2.22}$$

where:

$\mathbf{S}_R$ - scattering matrices cascade: waveguide section $\mathbf{S}_{WG}$ and spherical mirror $\mathbf{S}_{FEM}$,

$\mathbf{S}_P$ - scattering matrix for plane mirror with plate,

$\mathbf{I}$ - identity matrix.

Configuration with two spherical mirrors

The second configuration has a symmetrical layout. The dielectric plate is set at the center of the resonator. In this case, the section expressed analytically by the scattering matrix $\mathbf{S}_S$ consists of half of the sample with a thickness $t = 1003$ μm placed in a waveguide of length $L_S = 2t$. The investigated structure has the following dimensions: radius of the spherical mirror $R = 150$ mm, distance between the center of the sample and the farthest point of the spherical mirrors $D = 100$ mm, diameter of the spherical mirror aperture (simultaneously the diameter of the waveguide)

$D_{ap} = 180$ mm. The length of the section with a spherical mirror is $L_{FEM} = 40$ mm. $N = 5887$ elements are used for FEM analysis (the maximum length of the triangle edges in the mesh is 1 mm). The lengths of the empty waveguide section can be written as $L_{WG} = D - L_S - L_{FEM}$.

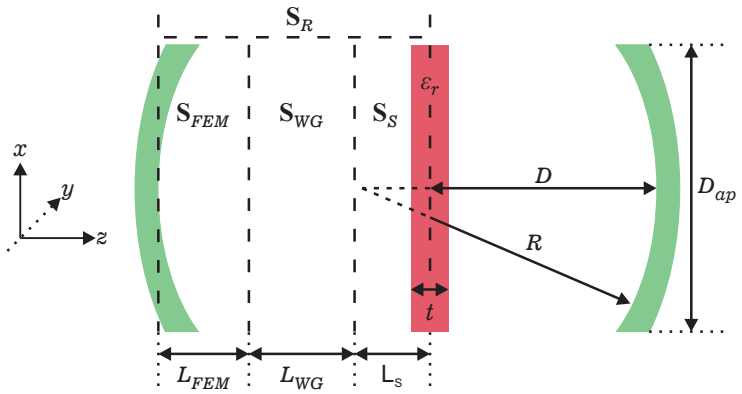

Figure 2.8: The geometry of investigated the spherical resonator and its divide into sections described by appropriate scattering matrices.

The investigation resonant frequency, f_r, can be sought by determining the roots of the determinant resulting from the following matrix equation:

$$F(f_r) = \det(\mathbf{I} - \mathbf{S}_R(f_r)\mathbf{S}_R(fr)) \tag{2.23}$$

where:

$\mathbf{S}_R$ - cascade of the scattering matrices: $\mathbf{S}_{FEM}$, $\mathbf{S}_{WG}$, $\mathbf{S}_S$,

$\mathbf{I}$ - identity matrix.

In both configurations, the relative permittivity of the dielectric ε_r can be swept during the analysis as an additional parameter. The resonant frequency is a complex number when the dielectric plate is made of a lossy material.

2.3.8 Dual-band cross-coupled eighth-order microwave filter

The last structure involves the analysis of the transfer function of an example microwave filter, defined by the following coupling matrix $\mathbf{M}$:

$$\mathbf{M} = \begin{bmatrix} 0 & 0.8024 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.8024 & 0 & 0.8467 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.8467 & 0 & 0.4142 & 0 & 0 & 0 & -0.29 & 0 & 0 \\ 0 & 0 & 0.4142 & 0 & 0.2754 & 0 & -0.4170 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.2754 & 0 & 0.2832 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.2832 & 0 & 0.2754 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.4170 & 0 & 0.2754 & 0 & 0.4142 & 0 & 0 \\ 0 & 0 & -0.29 & 0 & 0 & 0 & 0.4142 & 0 & 0.8467 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.8467 & 0 & 0.8024 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.8024 & 0 \end{bmatrix} \tag{2.24}$$

This model was initially presented in [135] (page 210), and its roots were also analyzed in [57]. The two-band characteristic is obtained by applying a symmetric frequency transformation, derived taking into account suitable boundary conditions. The matrix is created utilizing appropriate rational functions obtained by standard methods such as polynomial factorization. During the analysis, the general function describing the filter scattering parameters $\mathbf{S}_{11}(s)$ (reflection) and $\mathbf{S}_{21}(s)$ (transmission) obtained from the coupling matrix $\mathbf{M}$ is used according to the following relation:

$$\begin{aligned} \mathbf{S}_{11}(s) &= 1 + j2\left[\mathbf{M} - j\mathbf{R} + s\mathbf{I}_N\right]^{-1}_{11} \\ \mathbf{S}_{21}(s) &= -j2\left[\mathbf{M} - j\mathbf{R} + s\mathbf{I}_N\right]^{-1}_{(N+2)1} \end{aligned} \tag{2.25}$$

where:

$\mathbf{R}$ - zero matrix with non-zero elements in $\mathbf{R}_{11} = \mathbf{R}_{(N+2)(N+2)} = 1$,

$\mathbf{I}_N$ - identity matrix with zero elements in $\mathbf{I}_{N_{11}} = \mathbf{I}_{N_{(N+2),(N+2)}} = 0$.

The analysis aims to find the normalized cutoff frequency for a low-pass filter implicit in $s = \sigma + j\omega$ by finding the roots of the function $F(s) = \mathbf{S}_{11}(s)$.

Chapter 3

Global complex roots and poles finding algorithm

The Global Complex Roots and Poles Finding Algorithm [104] (abbreviation GRPF) is a technique that allows finding the zeros or poles of a complex function $F(z)$ in a region of arbitrary shape Ω. This method is capable of analyzing a wide range of functions, including those with branch cuts [141] and essential singularities. This approach is based on function phase analysis, initially proposed in [55]. Although this algorithm works efficiently and has many advantages, some imperfections in this procedure are nevertheless seen during various numerical tests. Successive versions and improvements of the algorithm [116], [119] are the result of research, which will be presented in the subsections of this chapter. Improvements have been made on several levels, including generating the initial mesh, modifying the final stages for better efficiency, and improving the accuracy of the results. The final version of the algorithm evolved the ground for other approaches. It can be used to set a starting point for local and tracing methods or as a "slicing method" for comparing results relative to the new approach in $\mathbb{C} \times \mathbb{R}$ (see Chapter 5).

The initial section of this chapter will provide a comprehensive explanation of the algorithm's fundamental principles and working procedures. The following subsections will delve into the limitations of the standard version and the general issues experienced in its application, which served as the basis for the research questions. The study's primary objective was to assess the dependability and efficacy of the latest algorithm addressing nonlinear eigenproblems.

3.1 Formulation of the problem

The destination of any global algorithm is to find all zeros and poles of the function $F(z)$ (see equation 1.2) in the considered search area $\Omega \in \mathbb{C}$. The set of points belonging to the domain $z \in \Omega$ is called the input space and can be written as a superposition of the real domain $\mathrm{Re(z)}$ and the imaginary domain $\mathrm{Im(z)}$. On the other hand, the output space is the set of complex values of the function $F(z) \in \mathbb{C}$. Utilizing Euler's formula for complex numbers, the analyzed value of the function can be expressed in the exponential form $F(z) = |F|e^{j\phi}$, taking into account separately the absolute value $|F|$ and the phase ϕ referred to as $\arg(F(z))$.

3.1.1 Visualization of the complex space

Assigning an input space to an output space is the classic definition of a function. There are many representations of mathematical functions. In addition to algebraic expression in the form of a numerical set, the function can be represented in the graph, which is the most common form. The graphical form of expressing mathematical relationships is crucial for understanding specific theories. In the case of scalar functions of one variable, the situation is relatively straightforward. However, function composition or adding another dimension to the analysis makes such visualization challenging, even using three-dimensional computer graphics. It is no different with complex functions, which, as already mentioned, have a two-dimensional input space and a two-dimensional output space, resulting in a four-dimensional space. Domain coloring [142] is the most popular and practical visualization method for complex functions. In this dissertation and related publications, a modified version of the phase portrait presented by Wegert is utilized.

In order to provide comprehensive information on the complex function, an additional graphical representation in the form of colors should be introduced to the two-dimensional graph $(\mathrm{Re}(z), \mathrm{Im}(z))$. This approach will enable the use of appropriate colors illustrating the phase of the function, and the absolute value can be represented in the third dimension. Nevertheless, from the point of view of the visualization of zeros and poles, the argument of the complex function is most important (which will be explained in detail in the following subsections); hence, the so-called phase portrait can be used. All phase portraits presented in this dissertation have a discrete form containing four colors: $\{\mathrm{red}, \mathrm{yellow}, \mathrm{green}, \mathrm{blue}\}$. It is assumed that in the output space, individual quadrants of the coordinate system $\{\mathbf{I}, \mathbf{II}, \mathbf{III}, \mathbf{IV}\}$ are marked with appropriate colors, as shown in Figure 3.1. For each point z from the input space, the so-called quadrant Q_n can be assigned according to the following rule:

$$Q_n = \begin{cases} \mathrm{I}, & 0 \leq \arg F(z_n) < \pi/2 \\ \mathrm{II}, & \pi/2 \leq \arg F(z_n) < \pi \\ \mathrm{III}, & \pi \leq \arg F(z_n) < 3\pi/2 \\ \mathrm{IV}, & 3\pi/2 \leq \arg F(z_n) < 2\pi \end{cases} \tag{3.1}$$

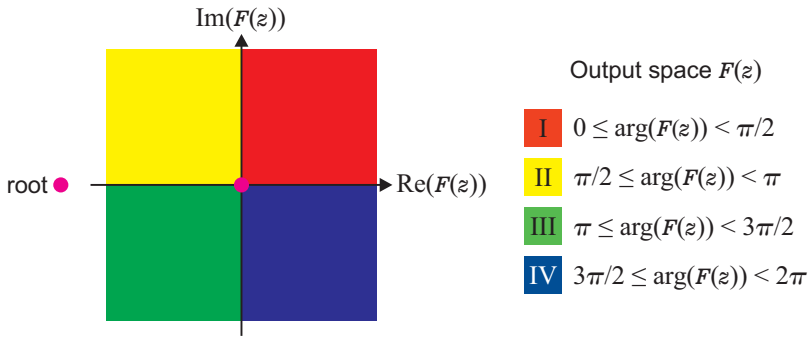

Figure 3.1: The output space of any function $F(z)$. In each quadrant of the coordinate system, the argument of the function $\arg(F(z))$ is within the given above intervals.

In the real domain, according to the geometric interpretation, the zero of the function $F(x) = 0$, $x \in \mathbb{R}$ is called the point of intersection graph of the function with the abscissa (OX in the Cartesian coordinate system) see Figure 3.2. However, things get slightly more intricate when it comes to complex numbers. The zero exists only when the real and imaginary parts of the function take the value zero. Such a scenario is only possible at the origin of the coordinate plane of the output space, see Figure 3.1. Based on the value of the function (in particular $\arg(F(z))$), the output space can be mapped onto the input space, thus obtaining a phase portrait (see Figure 3.2). As can be seen, this way of visualizing the complex function allows for an "estimated" determination of the roots of the function.

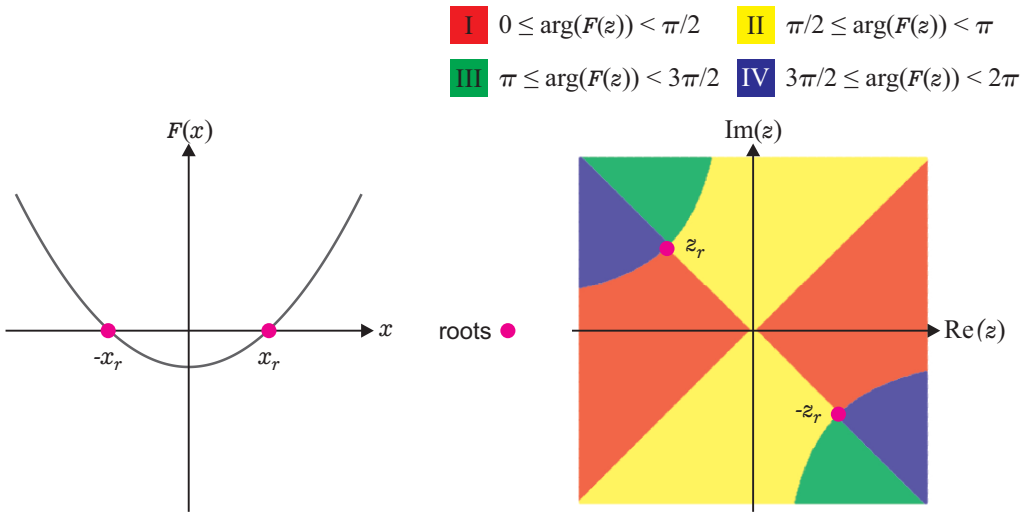

Figure 3.2: The graphical representation roots of example functions. On the left is graph of a real function $F(x) = (x + x_r)(x - x_r)$ on the right is a phase portrait of the complex function $F(z) = (z + z_r)(z - z_r)$.

3.1.2 Cauchy's argument principle

Complex functions $F(z) \in \mathbb{C}$ can be integrated along curves defined in the complex plane. In general, the value of the function changes along some curve $C \in \mathbb{C}$ from point $z = z_1$ to point $z = z_2$, see Figure 3.3. Thus, the value of the integral could depend on both the line of integration (shape of the curve) and the function itself. However, suppose $F(z)$ is a holomorphic function along the integration curve. In that case, the value of the integral is constant. It is analogous to the behavior of the potential of a field for two-dimensional real functions. In the case of an integral along a closed C_c curve, its value becomes zero if the function is holomorphic over the entire region enclosed by this curve, according to the following formula:

$$\int_{C_c} F(z)dz = 0 \tag{3.2}$$

In order to confirm the existence of a zero or a pole in the complex space, it is necessary to meet Cauchy's argument principle [13] (abbreviation as CAP), according to which the number of

all zeros and poles inside any curve C is represented by the following integral:

$$q = \frac{1}{j2\pi} \oint_C \frac{F'(z)}{F(z)} dz \tag{3.3}$$

where:

$q \in \mathbb{Z}$ - parameter that is the difference between the sum of the number of all existing zeros with their multiplicities and the sum of the number of all existing poles with their multiplicities.

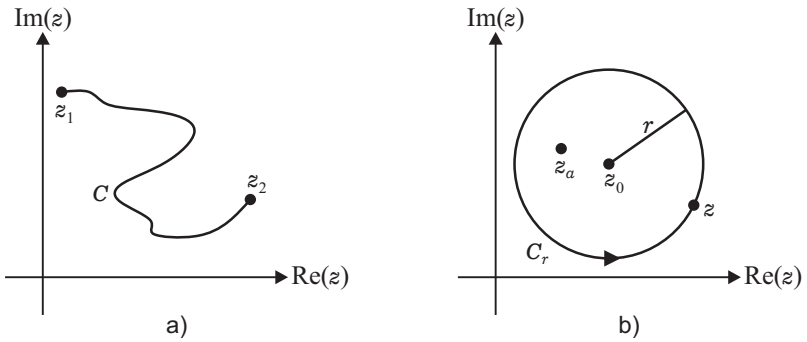

Figure 3.3: On the left is an example of a curve C in complex space. On the right is a closed curve (circle) $C_r \in \mathbb{C}$.

In Figure 3.4, an example of a CAP verification is shown. Inside curve C, there are three zeros and one pole. In this case, the parameter $q = 3 - 1 = 2$. Suppose a region contains only one essential point (zero or pole). In that case, q denotes the multiplicity (of the root or singularity), a consequence of the theorem included in the equation (3.2).

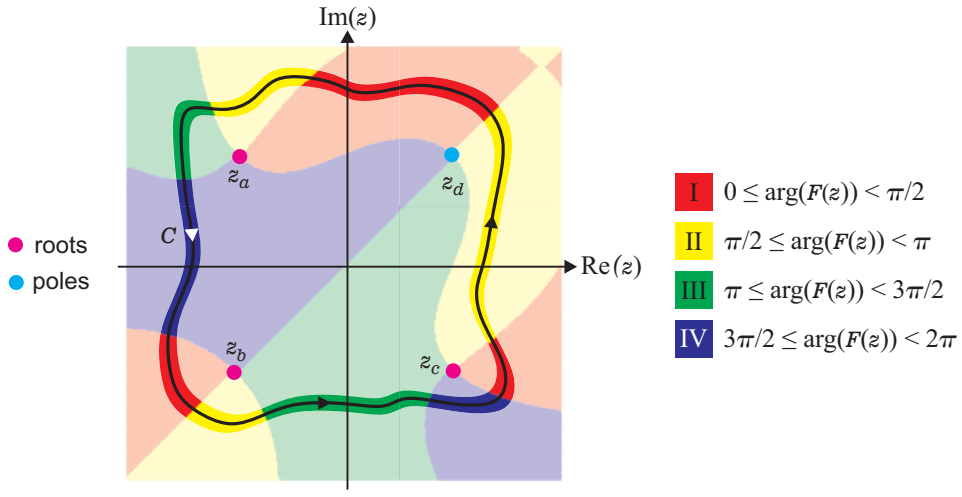

Figure 3.4: The phase portrait of an example rational function $F(z) = \frac{(z-z_a)(z-z_b)(z-z_c)}{(z-z_d)}$ with curve C plotted on it. Zeros are marked with magenta dots, and the poles are marked with cyan.

The formula (3.3) presents an integral that, at first glance, may seem trivial to solve due to its derivative of the logarithmic function. However, in complex space, the logarithm function does not have a unique form, as pointed out by reference [13] (see equation (3.4)). It complicates the

solving process of the integral and requires a more sophisticated analysis and approach.

$$\log(z) = \log|z| + j(\arg(z) + 2k\pi) \tag{3.4}$$

where: $z \in \mathbb{C}$, $k \in \mathbb{Z}$.

Therefore, the result of integration (3.3) is also not trivial. In practice, it is possible to use the split of the curve C into M smaller segments, for which the integrals can be calculated separately, obtaining:

$$\begin{gathered} q = \frac{1}{2\pi} \sum_{m=1}^{M-2} |\Theta_m| \\ \Theta_m = \begin{cases} \arg F(z_{m+1}) - \arg F(z_m), & m = 1, ..., M-1 \\ \arg F(z_1) - \arg F(z_m), & m = M \end{cases} \end{gathered} \tag{3.5}$$

However, it should be noted that the increase in the $|\Theta_m|$ phase cannot exceed π (as shown in [143], [144]). For this reason, the division into sections is not a straightforward and apparent process.

If the value of q obtained from the integral (3.3) is equal to zero, then there may be no roots inside the curve C, or there may be the same number of zeros and poles. To minimize such potential situation and the possibility of overlooking any roots, CAP can be extended to higher moments m : [145]:

$$\frac{1}{j2\pi} \oint_C \frac{F'(z)}{F(z)} z^m dz = \sum_{k \in \{roots\}} \left(z^{(k)}\right)^m - \sum_{k \in \{poles\}} \left(z^{(k)}\right)^m \tag{3.6}$$

The utilization of the moment $m = 1$ effectively resolves the issue for a singular pair of roots, as detailed in [146]. Furthermore, any higher moment can further reduce this problem of a higher order of roots. This technique is especially practical when the analytical expression of the function is known.

3.1.3 Bisection on a complex plane

The Discrete Cauchy Argument Principle (abbreviation DCAP) requires neither the computation of the function's derivative nor its integration over the contour. In practice applications, the parameter q can be evaluated by sampling the function along the contour C as outlined in [143], [147], [148].

$$q = \frac{1}{2\pi} \sum_{p=1}^{P} \arg \frac{F(z_{p+1})}{F(z_p)} \tag{3.7}$$

where:

points $z_1, z_2, ..., z_P$, and $z_{P+1} = z_1$ are obtained through discretizing curve C.

To ensure the accuracy of the discretization, it is essential that the phase change of the function $F(z)$ along curve C_p ($C = \bigcup_{p=1}^{P} C_p$), from point z_p to point z_{p+1}, satisfies the following condition:

$$\left|[\arg F(z)]_{z \in (z_p, z_{p+1})}\right| \leq \pi \tag{3.8}$$

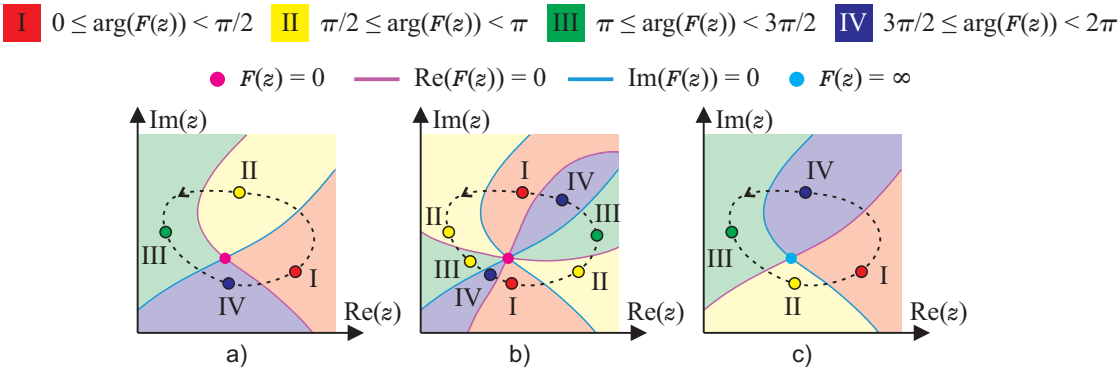

Figure 3.5: Examples of the Discrete Cauchy Argument Principle: a) single root $q = 1$ b) double root $q = 2$ c) singularity $q = -1$.

In order to confirm the existence of a single root within a small area, only four points located in different quadrants of the coordinate system are sufficient (the remaining points can be omitted), as shown in Figure 3.5 a), c). The CAP discretization allows for extending the standard bisection method on the complex plane. It involves creating a closed quadrilateral region using the four points. Each successive step of this process adds a new point at the center of the longest edge (see Figure 3.6). The value of the function at this new point must be from the same quadrant as one of the ends of the segment under consideration. Then, one of the points with the same quadrant is removed. As a result, the region is reduced. The process can be repeated for the longest edge of the new contour until the desired level of accuracy (distance between points) is reached.

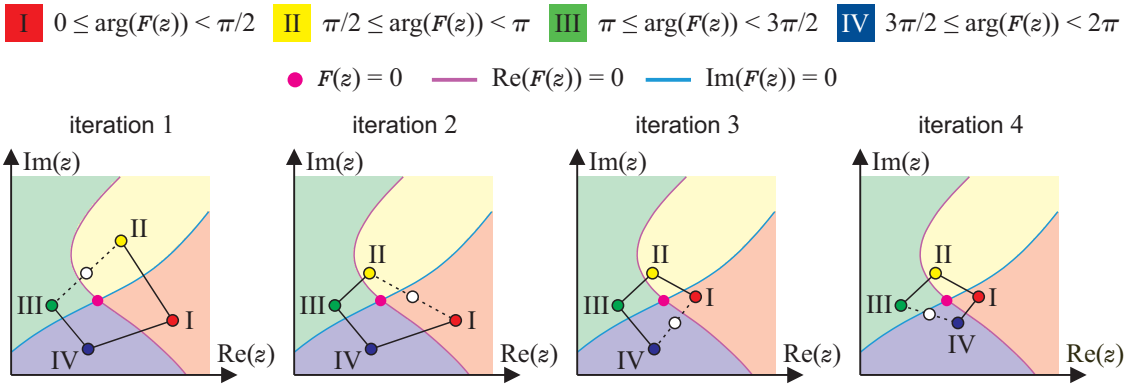

Figure 3.6: The four steps of the generalized bisection method on the complex plane for an example function with a single zero.

The rate of convergence of the process above is comparatively lower than that of the three-point technique, such as Muller's method [51]. Nevertheless, the method presented herein has one significant advantage - it determines the region where the root is located (in each procedure step).

3.2 Introduction to the algorithm

The GRPF algorithm presented in [104] can be segmented into two distinct phases (independent of each other): the *Preliminary Estimation* and *Mesh Refinement*. During the preliminary estimation process, the roots and poles are estimated by sampling the function at regular triangular mesh nodes and utilizing DCAP. The second phase of the algorithm seeks to attain the necessary level of accuracy by reducing the length of long mesh edges accordingly.

3.2.1 Preliminary estimation

This stage is further broken down into smaller steps, such as *Initial mesh*, *Function evaluation*, *Candidate edges*, and *Candidate regions*.

Initial mesh

The starting point for the GRPF algorithm is the discretization of the defined domain Ω. A regular mesh consisting of N_0 nodes ($z_n \in \Omega, n = 1, 2, ..., N_0$), and P_0 edges $\in \Omega$ is created for this purpose. The division of the domain into triangular elements and the definition of edges is accomplished by a well-established approach using Delaunay triangulation [149] in the $\mathbb{R}^2$ domain. Its origins are rooted in the multi-dimensional bisection method [150] (see subsection 3.1.3). The mesh shape based on equilateral triangles provides the highest efficiency of the algorithm. Figure 3.7 shows an example mesh constructed with $N = 14$ nodes and $P = 28$ edges. The normalized length (for the size of the domain Ω) of the longest edge is $r = 0.5$. Any other arbitrary mesh configuration is also possible; it is crucial to ensure that the length of the longest edge is smaller than the adopted resolution r. This parameter will be discussed in more detail in subsection 3.3.

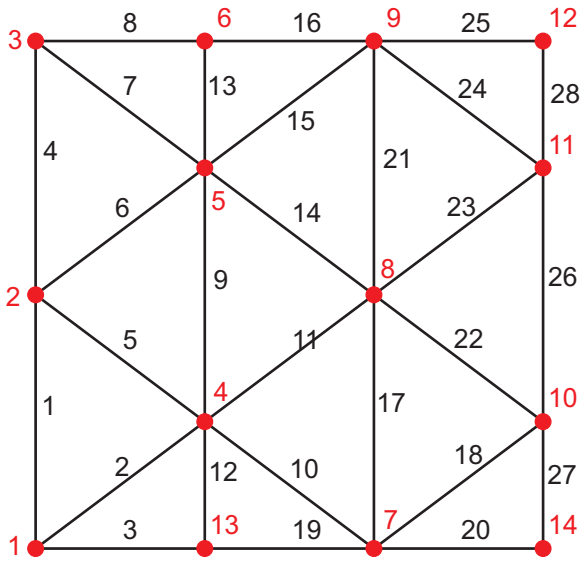

Figure 3.7: The example mesh with the numeration of nodes (labeled red) and edges (labeled black), consisting of evenly distributed equilateral triangles (or their halves on the boundary) created using Delaunay triangulation.

Function evaluation

The subsequent stage involves calculating the function values at each node of the mesh $F_n = F(z_n)$, where $n = 1, 2, 3, ..., N$. This step tends to be the most time-consuming part of the algorithm, especially for complex functions, because the number N usually reaches large values. In general, parallel computation can be applied in multithreading computer architecture, significantly improving process efficiency for extensive problems. As mentioned earlier, the argument holds more importance for the algorithm than the absolute value of the function. Thus, the phase of the function at the nodes is analyzed instead of a straightforward sign change, as in [55], for example. It is the fundamental difference between these approaches. At this stage, the argument of the function $arg(F(z))$ is discretized into four levels of values depending on the quadrants marked as Q_n according to equation (3.1).

Candidate edges

In order to perform a discrete analysis of the phase change along each edge p, the initial step involves calculating the quadrant Q_n for every vertex n on the mesh. Afterward, an additional parameter representing the difference of quadrants between the vertices defining the edge must be introduced:

$$\Delta Q_p = Q_{n_{p2}} \ominus Q_{n_{p1}} \tag{3.9}$$

where:

$\Delta Q_p \in \{-2, -1, 0, 1, 2\}$,

$\ominus$ - the difference in value may be greater than 2, but $\Delta Q_p = -3$ reduces to $\Delta Q_p = 1$ and $\Delta Q_p = 3$ to $\Delta Q_p = -1$,

n_{p1} i n_{p2} - nodes attached to the edge p.

The algorithm is based on the dependence that each zero or pole is found in a region where the phase of the function is included in four different quadrants. To illustrate this more precisely, Figure 3.8 provides the mesh with a phase portrait of an example function:

$$F(z) = \frac{(z - j)(z - 1)^3}{(z + 1)}$$

in region:

$$\Omega = \{z \in \mathbb{C} : -2 < \mathrm{Re}(z) < 2 \wedge -2 < \mathrm{Im}(z) < 2\}$$

This function contains a single zero at $z^{(1)} = j1$, a triple zero at $z^{(2)} = 1$, and single pole at $z^{(3)} = -1$. It is worth noting that each triangulation of four nodes located in four different quadrants requires at least one edge with $|\Delta Q_p| = 2$. Therefore, all edges for which the parameter $|\Delta Q_p|$ is equal to two should be considered probable zero or pole neighborhoods, and these edges will be referred to as ”candidate edges” in further analysis.

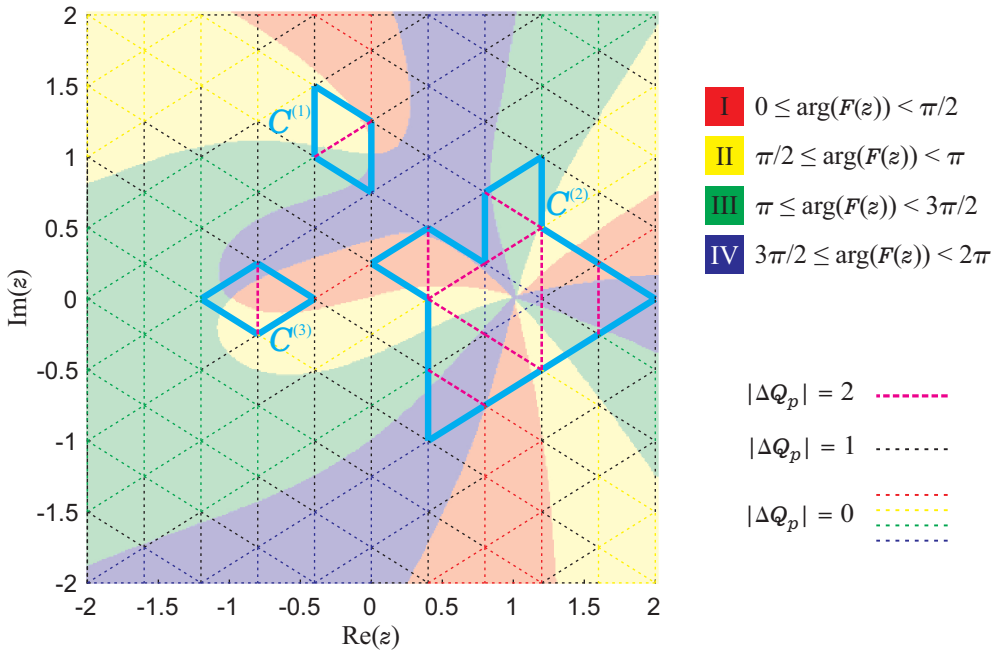

Figure 3.8: The illustration of the idea of the algorithm, with an example of a rational function with a phase portrait set in the background. The ”candidate edges” are depicted in magenta. The boundary of the ”candidate regions” is marked with a solid cyan line.

Candidate regions

All triangles joined to ”candidate edges” can be collected into one set of candidate triangles $\mathcal{E} = \{p : |\Delta Q_p| = 2\}$. This set can be broken down into subsets, i.e., the so-called ”candidate regions” containing only triangles tangential with at least one ”Candidate Edge.” The boundaries of these subsets are formed by the edges, which create curves $C^{(k)}$ that enclose the ”candidate regions”. In Figure 3.8, one can observe three ”candidate regions” surrounded by closed curves $C^{(1)}$,$C^{(2)}$, and $C^{(3)}$.

Extracting such curves is a straightforward and efficient process. It involves identifying edges tangent to the ”candidate edges” that connect only once. Collecting these edges into a ”boundary edges” set for all ”candidate regions” is simple and convenient. These edges form the boundaries of all regions because inner edges are attached to at least two ”candidate edges.” To construct the boundary of a separate region, one can start from any edge in the set of ”boundary edges” (removing it from this set). Then, it finds a connected edge and attaches it to the set that forms the $C^{(k)}$ curve (removing it from the set of ”boundary edges”). The edge should close the curve if it is not connected to a new one in the ”boundary edges” set. In this way, the region’s boundary is determined as a $C^{(k)}$ curve. If unused edges are still in the ”boundary edges” set (the set is not empty), one starts building the next edge of the region $C^{(k)}$. The procedure again begins by selecting any edge from the remaining ”boundary edges,” removing it from the set, finding the next edge connected to the previous one, and so on. The ”candidate regions” boundary must be constructed only from edges $|\Delta Q_p| \leq 1$, according to the DCAP condition (see equation (3.8)).

In some cases, regions may be defined ambiguously because their boundary may be open, e.g., the ”candidate edge” is on the domain's boundary Ω. To resolve this issue, the domain can be modified, or shorter edges of the initial mesh can be applied.

3.2.2 Verification with discretized Cauchy's argument principle

In this approach to DCAP, the minimum number of nodes at which function values are evaluated is used. Generally, a minimum of four nodes are required to verify a single root or pole. The shape of the ”candidate regions” is determined by the edges of the mesh. Let us consider the example from Figure 3.8, where there is a single root, a triple root, and a singularity. For each of these three ”candidate regions,” the function's argument changes along the contour by a integer multiplicity of $|2\pi|$, taking values from the four quadrants. Since the quadrant difference along a single edge is $|\Delta Q_p| < 2$, the discretization of the region boundary is sufficient to estimate the total phase change at the region boundary. Adding up all the changes (increments) in the quadrants along contours $C^{(1)}$, $C^{(2)}$, and $C^{(3)}$ in the counterclockwise direction, one gets values $\Delta Q_{C^{(1)}} = 4$, $\Delta Q_{C^{(2)}} = 12$, and $\Delta Q_{C^{(3)}} = -4$ for regions containing $z^{(1)}$, $z^{(2)}$, and $z^{(3)}$, respectively. Taking into account the fact that the increase in the number of quadrants along the edges represents changes in the argument of the function by a minimum of $\pi/2$, the parameter q can be calculated by transforming equation (3.7) to the following form:

$$q = \frac{1}{4}\sum_{p=1}^{P} \Delta Q_p \tag{3.10}$$

Hence, it is confirmed that the parameters $q_{C^{(1)}} = 1$, $q_{C^{(2)}} = 3$, and $q_{C^{(3)}} = -1$ correspond to a single root, a triple root, and a singularity in their respective regions. Additionally, it is worth mentioning that the change of quadrants along ”candidate edges” $|\Delta Q_p| = 2$ is not unambiguous, as it is not always possible to determine whether the phase is increasing or decreasing (only a change of two quadrants is distinguishable). Consequently, verifying condition (3.8) can be problematic in some cases, as demonstrated in [144], [147].

3.2.3 Mesh refinement

In order to improve the accuracy of the results, any local (e.g., iterative) root finding algorithm can be used. However, as mentioned earlier, such methods can be unreliable, potentially resulting in missed zeros or poles if the initial value is not precise enough. Unlike other techniques (such as Newton's or Muller's), a process that starts at a given initial point may converge to a different root, especially if the zeros or poles are near each other. The presented approach uses simple iterative mesh refinement within predefined ”candidate regions.” In order to ensure the correct convergence of the algorithm and the reliability of the obtained results, the algorithm must include all of the initially found roots or poles, even if they are not precisely in the ”candidate regions.”

In subsequent iterations, new nodes are added to the mesh at the centers of each edge in the ”candidate regions,” as illustrated in Figure 3.11. This results in a new mesh created using

Delaunay triangulation. Function values are then evaluated at these new points, and the new mesh is analyzed just as in the initial analysis. New ”candidate regions” for a locally denser mesh are determined. Of course, the new ”candidate regions” are smaller, improving the result's accuracy. This process can be repeated as often as assumed until the user's chosen accuracy δ is achieved.

Unfortunately, in subsequent iterations, mesh refinement can lead to ill-conditioned geometry and so-called ”skinny” triangles. The ”skinny” triangles are triangles with a huge disproportion in edge length. It requires creating additional zones surrounding the ”candidate regions” and introducing additional points. A detailed analysis of this issue and how to solve this problem will be arranged in the following sections of this chapter.

3.3 Effectiveness and limitations of the algorithm

Increasing the discretization level in each global method of finding zeros and poles raises the likeliness of finding all roots in the investigated region. GPRF, in its main assumptions as one of the few algorithms in this group, has direct control over the results' accuracy. It is expressed as the δ parameter, defined as the length of the ”candidate edges.” Theoretically, the condition for GRPF to accurately find all zeros and poles is simple and clear-cut. The initial mesh must be dense enough, translating into a short edge of the initial mesh marked by r. However, this can turn out challenging in practice; it can mean the number of nodes will go to infinity $N_0 \to \infty$. It is difficult to realize due to limited computational resources, especially when there are so-called ”islands” in the region under consideration. These are fragments of the domain in which zero-pole pairs are very close to each other. They are challenging to detect and locate using CAP-based integral methods, confirmed by the research presented in [112]. To further illustrate the issue of ”islands,” consider the rational function expressed by the following formula:

$$F(z) = \frac{(z - z_a)(z - z_b - \epsilon)}{(z - z_c)(z - z_b + \epsilon)} \tag{3.11}$$

$$z_a = \tfrac{1}{2} - \tfrac{\sqrt{3}}{6}j, \qquad z_b = \tfrac{\sqrt{3}}{3}j, \qquad z_c = -\tfrac{1}{2} - \tfrac{\sqrt{3}}{6}j$$

The points z_a, z_b, and z_c are the vertices of an equilateral triangle with side lengths $a = 1$ located at the origin of the coordinate system. The rational function (3.11) contains a zero at z_a, a pole at z_c, and a zero-pole pair near the point z_b. The points $z_b - \epsilon$ and $z_b + \epsilon$ are some distance apart, denoted by the parameter ϵ. In geometric terms, the distance between them is 2ϵ. For simplicity of designations, we will assume that ϵ means the size of the ”island” because it only analyzes the symmetric case concerning the separation for a specific point; in this case, it is z_b. Phase portraits of the function (3.11) with different example ”island” sizes are shown in Figure 3.9 in the following area:

$$\Omega = \{z \in \mathbb{C} : -1 < \mathrm{Re}(z) < 1 \wedge -1 < \mathrm{Im}(z) < 1\}$$

It is noticeable that a decrease in the ϵ value results in a reduction in the size of the ”island,” leading to a change in the phase of the function within a smaller region. Typically, identifying

these differences necessitates the use of more refined algorithms. Unfortunately, this type of function (with ”islands”) is frequently encountered in microwave-related problems, such as filters, waveguides, and resonators (see Chapter 2).

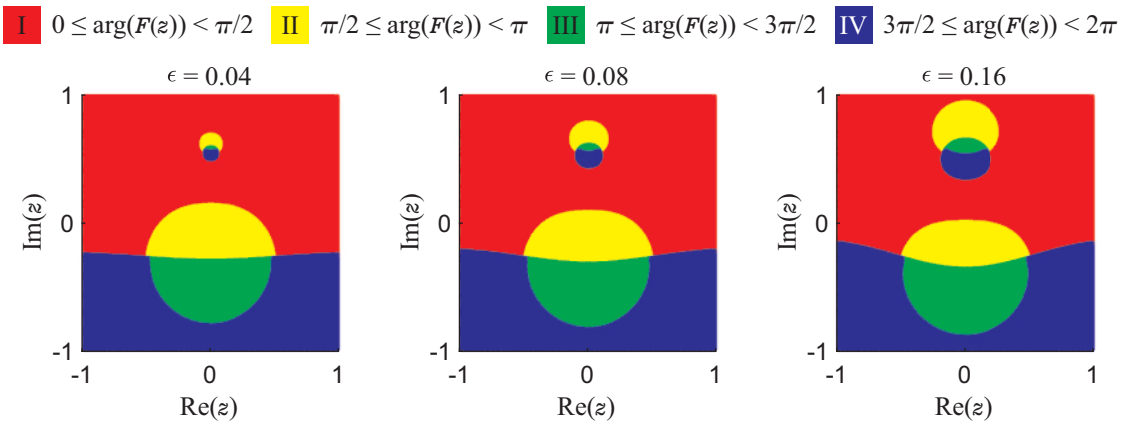

Figure 3.9: The phase portrait for function from (3.11) in assumed region Ω for three different values of the epsilon parameter ($\epsilon = 0.16$, $\epsilon = 0.08$, $\epsilon = 0.04$).

For the above function (3.11), the influence of the dimension of the triangle edges, and therefore the density of the regular initial mesh, on the detection of the ”island” of size $\epsilon = 0.01$ by GRPF will be analyzed. Table 3.1 presents the investigation results wherein the value of the r parameter is decreased, thus increasing the level of discretization and the number of nodes N_0. In addition, Figure 3.10 depicts the selected initial meshes for $N_0 = 68, 232, 795, 3038$.

Table 3.1: The comparison of the analysis results of the initial mesh for the function (3.11) with a constant parameter $\epsilon = 0.01$ for different edge lengths r, i.e., discretization parameters (c. abbreviation of candidate).

r	N_0	**no. of ”c. edges”**	**no. of ”c. regions”**	**is ”island” detected?**
0.32	68	3	2	no
0.16	232	2	2	no
0.08	795	2	2	no
0.04	3 038	8	3	yes

The outcomes shown in Figure 3.10 indicate that if the initial mesh does not have ”candidate edges” in the investigated area, the refinement process will not start. Consequently, the algorithm will not detect zero or pole in a given region, which is also confirmed by the data in Table 3.1. For instance, if for function (3.11) one uses an initial discretization with $N_0 = 104$ nodes, only two ”candidate regions” will be determined: around the zero in z_a and the pole in z_c, while the ”island” around z_b will not be located. The refinement process will concentrate only on the area around z_a and z_c. Ultimately, GRPF will find only one zero and one pole with the given δ accuracy. At the same time, the mesh around the ”island” will not be refined, and the zero-pole pair will be undetected during the analysis. Generally, the algorithm may locate this ”island” around z_b.

However, discretizing requires a significantly greater number of samples, such as $N_0 = 3\,038$ (see Figure 3.10 and Table 3.1) or fortunate distribution of edges.

In general, determining the proper sampling density appears impossible a priori for any function. As shown in example (3.11), when ϵ decreases to 0.001, over a million points are needed to locate the "island," for even smaller ϵ values, N_0 would have to increase by billions. In special cases, the algorithm can detect any "island," no matter how small, using only a few nodes. However, this outcome relies on chance, as it is related to the lucky alignment of the mesh. Sometimes, a slight shift of the domain or a change in its size may affect the results obtained (success of the algorithm).

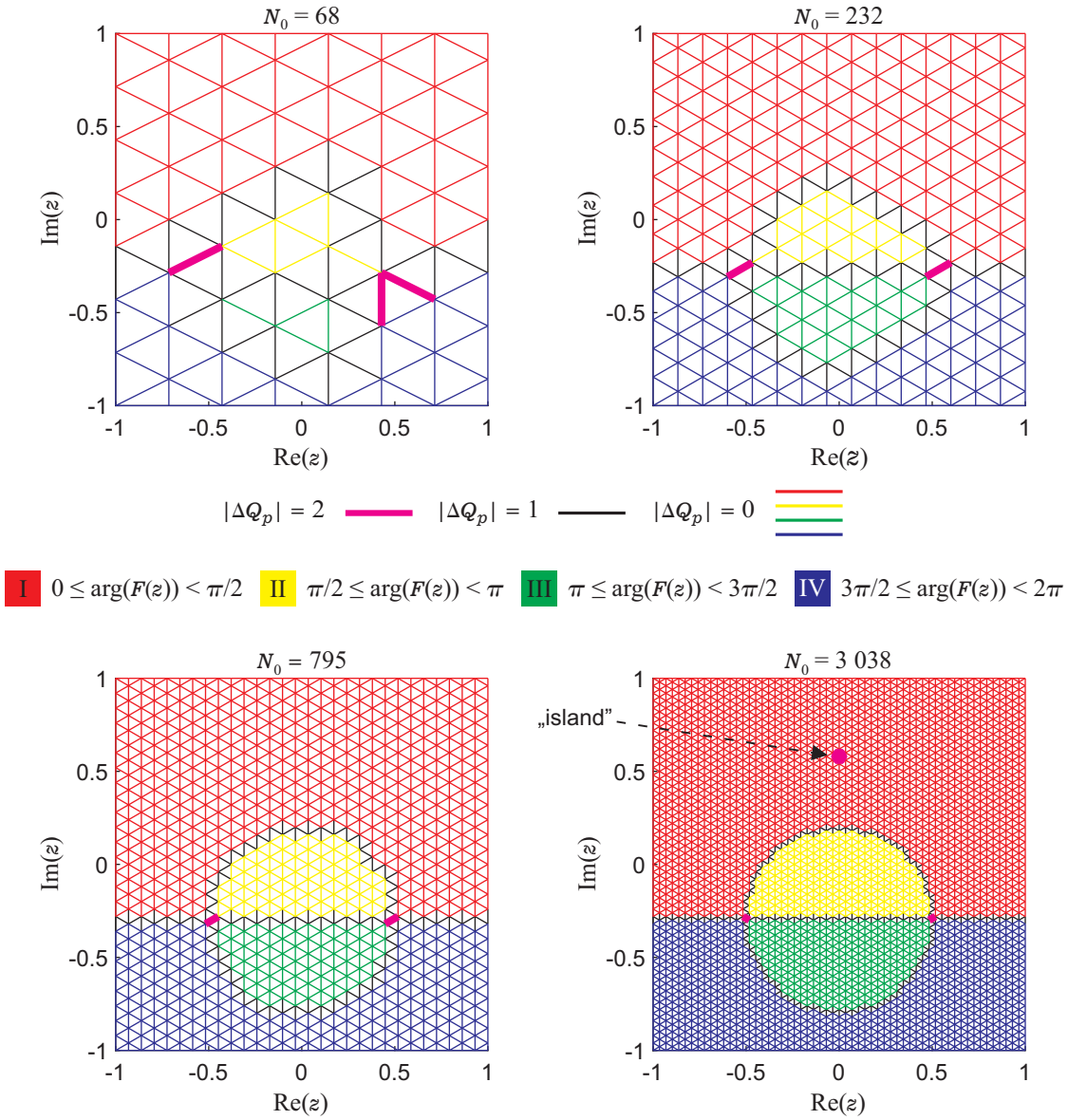

Figure 3.10: The initial triangulation and visualization of function evaluations for the rational function from (3.11) with the constant parameter $\epsilon = 0,01$ for a different number of points N_0 (i.e., different lengths of the edges see Table 3.1). Bold magenta lines represent "candidate edges."

To summarize, observation of a too sparsely sampled function may not provide indications that a zero-pole pair is present in investigated regions. Classical integral methods based on CAP also fail because the phase around the ”islands” changes slightly, and the integral in CAP equals zero. For the GRPF algorithm, a fit of the initial mesh may effectively find all zero-poles, but it is not a necessary condition. In fact, the r parameter may be larger than the distance between the roots (or poles) of the function. Unfortunately, as in the case of other well-established methods (e.g., based on interpolations [53] or discrete techniques [50], [55], [151]), there are no clear guidelines on how to select the initial discretization conditions for any function and area. As a result, users solve this issue through trial-and-error experiments. One can also conduct an initial verification for the whole boundary of region Ω using the DCAP idea. However, this still does not guarantee finding all the roots and poles.

3.4 The improvement of the algorithm

The following part of the work is a continuation and expansion of the research discussed in publication [104]. A simple modification and improvement of the mesh refinement stage is proposed. In the method presented in subsection 3.2 (hereinafter referred to as standard GRPF), new nodes are added at the center of each edge in ”candidate regions.” However, it is discovered that adding new points only at the centers of ”candidate edges” is sufficient. This modification leads to a significant reduction in computation time by minimizing function calls in successive iterations. For more sophisticated functions, calculating its value can be the most time-consuming part of the algorithm, so the improvement presented can significantly reduce the overall computation time. Despite the similarities of both approaches, regular and modified, it has been shown that reducing the number of nodes added in subsequent iterations can significantly improve the convergence of this process. Also, with the proposed solution, the mesh condition should be monitored by adding additional points in the centers of ill-conditioned (”skinny”) triangles (in both zones), but their number is insignificant. Moreover, an extra node is added in the center of the edges of the ”candidate regions” if it is located on the boundary Ω. An example of how this algorithm modification works is shown in Figure 3.11. In this case, the regular approach would require evaluating the function in 17 new nodes in the first three iterations, while the modified algorithm would only need 7.

3.4.1 Numerical examples

Both standard and modified versions of the algorithms are tried regarding the number of function calls, computation time, and the convergence and accuracy of the obtained roots. The numerical tests are carried out on selected examples from Chapter 2 by determination of the roots of functions defined analytically as well as numerically. The results are cross-verified with the reference values noted in [20], [126], [139].

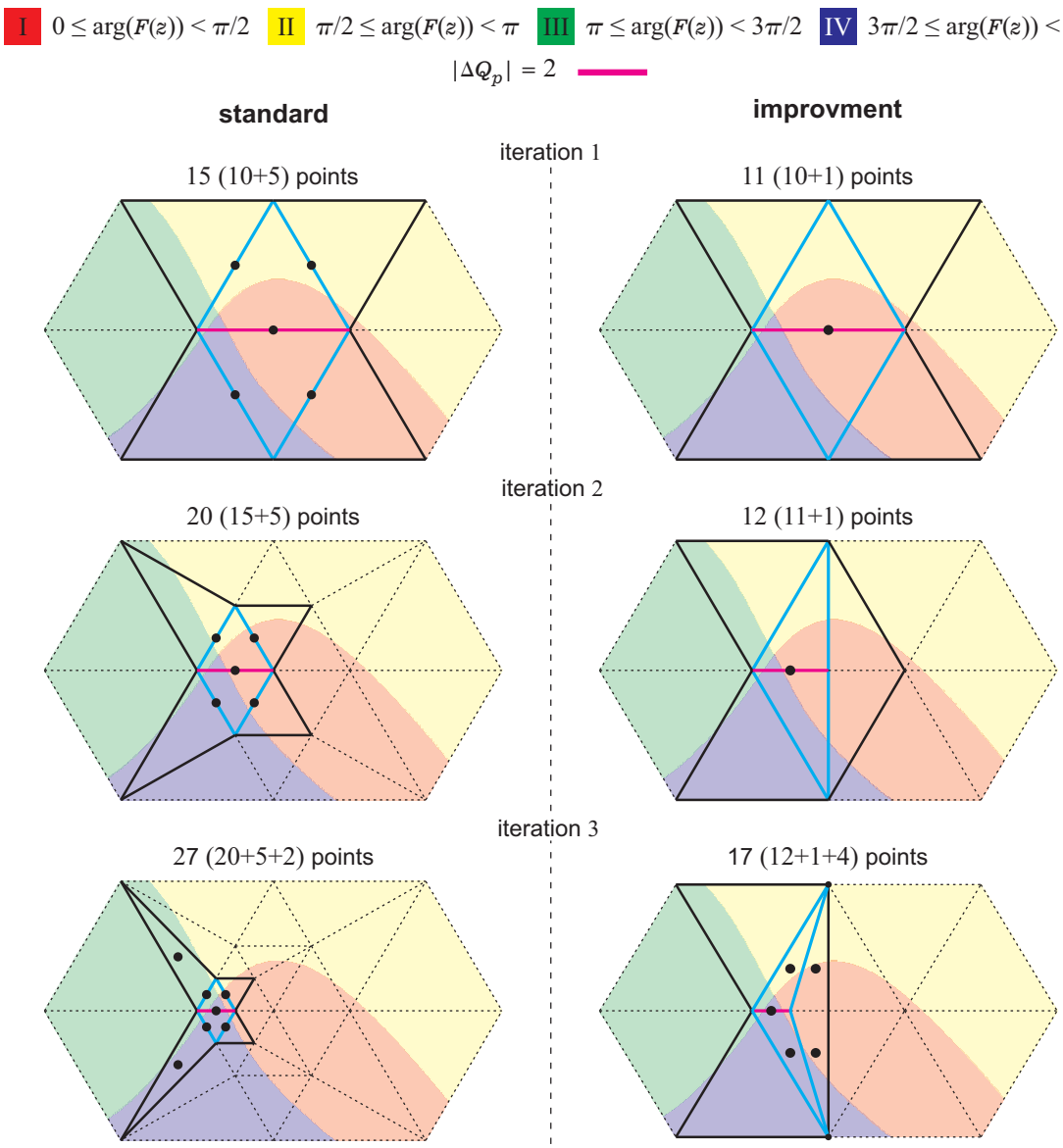

Figure 3.11: Three successive steps of an example mesh refinement process for two versions of the GRPF algorithm, regular and improved, are shown. The number of nodes in each iteration is presented with the number of new points in brackets. Bold magenta lines represent "candidate edges." The "candidate regions" boundary is marked, including a solid cyan line. The black solid bold line represents the boundary of the additional second zone. The remaining edges on the mesh are marked with a lightweight black dashed line. The new node is labeled with a black dot. In the third interaction, extra new nodes are added in the centroid of the "skinny" triangles.

Cylindrical-rectangular microstrip resonant structure

The first example includes the structure described in subsection 2.3.6. The investigated parameter z is the frequency f in the following range:

$$\Omega = \{z \in \mathbb{C} : 1.4 < \mathrm{Re}(z) < 1.5 \wedge -0.1 < \mathrm{Im}(z) < 0\}$$

The initial mesh shown in Figure 3.12 contains $N_0 = 7$ nodes at the $r = 0.1$ parameter.

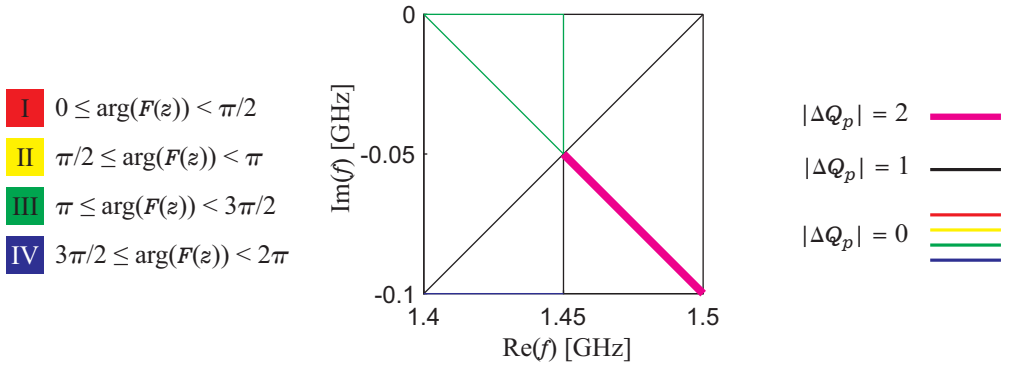

Figure 3.12: The initial mesh setup is identical for both the standard and improvement approach.

In both procedures, the determined zero (with $\delta = 10^{-12}$ precision) has the value:

$$f_r^{(1)} = 1.469084230386 - j\,0.071068929057 \text{ GHz}$$

The outcomes of both analyses align with the reference. However, Table 3.2 indicates a notable difference in the algorithms' efficacy. It is manifested in the number of nodes resulting from the final mesh form (see Figure 3.13). It is worth noting that the time to evaluate a single function is about 1.8 s; hence, the reduced number of nodes in the modified method directly impacts the total analysis time.

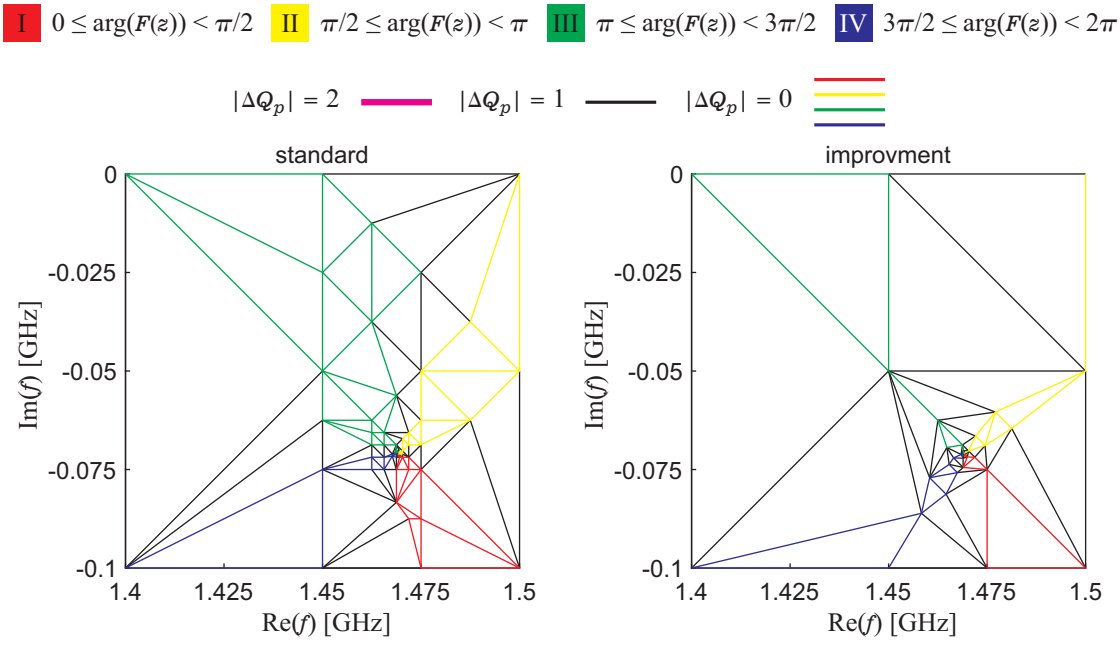

Figure 3.13: The comparison of the resulting final meshes for the standard and improvement versions of GRPF.

Table 3.2: The comparison of results obtained from analysis using the improvement and standard algorithms (standard approach in brackets [104])

accuracy δ	**time** [s]	**no. of function calls**	**no. of iterations**
10^{-3}	57.7 (90.1)	32 (50)	11 (7)
10^{-6}	122.0 (225.4)	68 (125)	27 (18)
10^{-9}	188.6 (324.5)	104 (180)	43 (27)
10^{-12}	249.0 (443.6)	138 (246)	60 (37)

Triangular dielectric fiber

Another example is a triangular dielectric fiber, the structure of which is presented in subsection 2.3.5. The domain of the analyzed function is:

$$\Omega = \{z \in \mathbb{C} : 1.25 < \mathrm{Re}(z) < 1.35 \wedge -0.4 < \mathrm{Im}(z) < -0.35\}$$

The initial mesh shown in Figure 3.14 contains $N_0 = 7$ nodes at the $r = 0.1$ parameter.

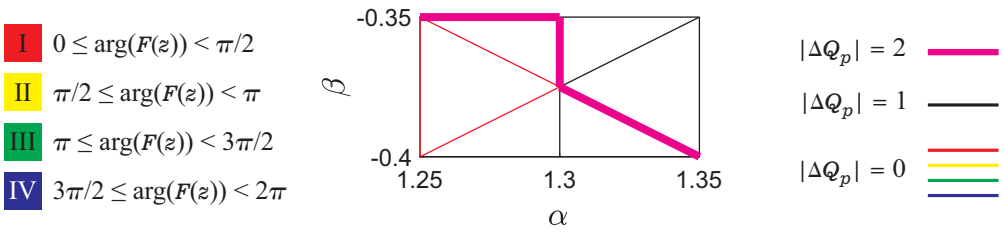

Figure 3.14: The initial mesh setup is identical for both the standard and improvement approach.

A leaky mode characterized by a normalized propagation coefficient $z = \gamma$ (with $\delta = 10^{-12}$ precision) is obtained:

$$\gamma^{(1)} = 1.260551725649 - j\,0.38483851254$$

Again, the end outcomes are the same for both procedures. As in the previous example, efficiency is enhanced. The number of nodes, analysis time, and iterations executed are listed in Table 3.3. Figure 3.15 contains the final mesh obtained from both algorithms. This example is also numerically complicated (the time to call a single function is about 0.1 s), by which the reduction in total analysis time is due to the smaller number of nodes in the modified GRPF.

Table 3.3: The comparison of results obtained from analysis using the improvement and standard algorithms (standard approach in brackets [104])

accuracy δ	**time** [s]	**no. of function calls**	**no. of iterations**
10^{-3}	16.6 (28.8)	161 (281)	17 (8)
10^{-6}	25.8 (48.0)	249 (467)	37 (18)
10^{-9}	34.8 (67.7)	336 (658)	58 (28)
10^{-12}	45.1 (85.1)	437 (829)	82 (38)

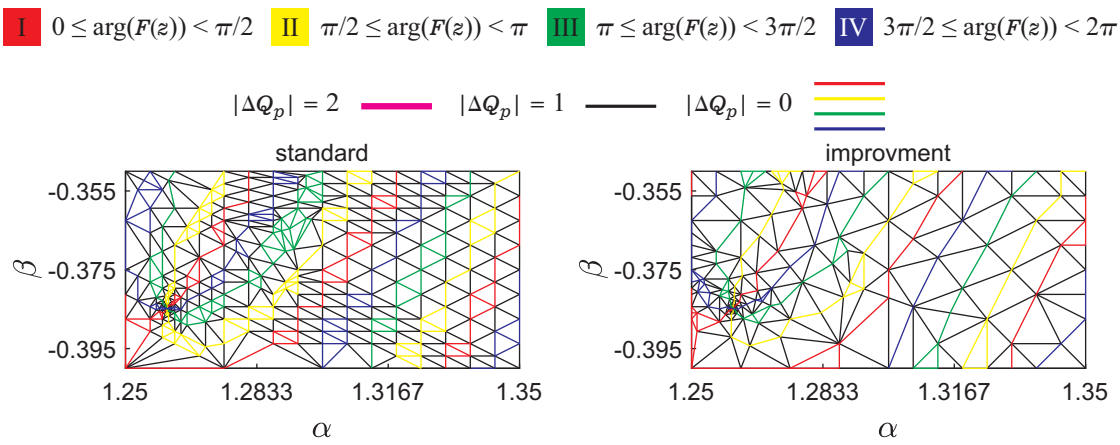

Figure 3.15: The comparison of the resulting final meshes for the standard and improvement versions of GRPF.

Lossy multilayered waveguide

As the last example to verify the improved version of the algorithm, the multilayer lossy waveguide presented in subsection 2.3.1 is considered. In this problem, the investigation for the normalized propagation coefficient $z = \gamma$ (for the wavelength $\lambda_0 = 632.8$ nm) is carried out in the range:

$$\Omega = \{z \in \mathbb{C} : 1.25 < \mathrm{Re}(z) < 1.75 \wedge -0.25 < \mathrm{Im}(z) < 0.25\}$$

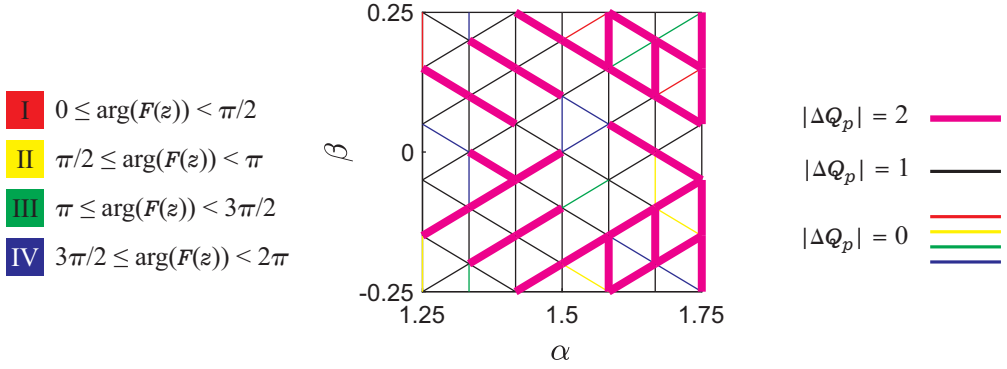

Figure 3.16: The initial mesh setup is identical for both the standard and improvement approach.

The initial mesh shown in Figure 3.16 contains $N_0 = 45$ nodes at the $r = 0.1$ parameter. The result of the analysis is the finding of the propagation coefficient for five modes (obtained with $\delta = 10^{-12}$ precision):

$$\gamma^{(1)} = 1.353140429183 - j\,0.000086139194$$
$$\gamma^{(2)} = 1.439795544245 - j\,0.000052001665$$
$$\gamma^{(3)} = 1.504169866405 - j\,0.000028029437$$
$$\gamma^{(4)} = 1.548692243882 - j\,0.000012101013$$
$$\gamma^{(5)} = 1.574863045753 - j\,0.000002974624$$

The final results are identical in both approaches. However, in contrast to previous instances, for such a relatively simple function, the efficiency of both algorithms is similar (see Table 3.4). Nevertheless, the final mesh in the modified approach is less regular than in the previous procedure, as evidenced in Figure 3.17. Since the function is not compiled (the time of evaluating a single value of the function is about 0.04 ms), the difference in total analysis time is negligible. Even if the number of function calls in the modified algorithm is much smaller. From a practical perspective, considering the total CPU use in this case, the choice between these two approaches for simple functions holds no importance.

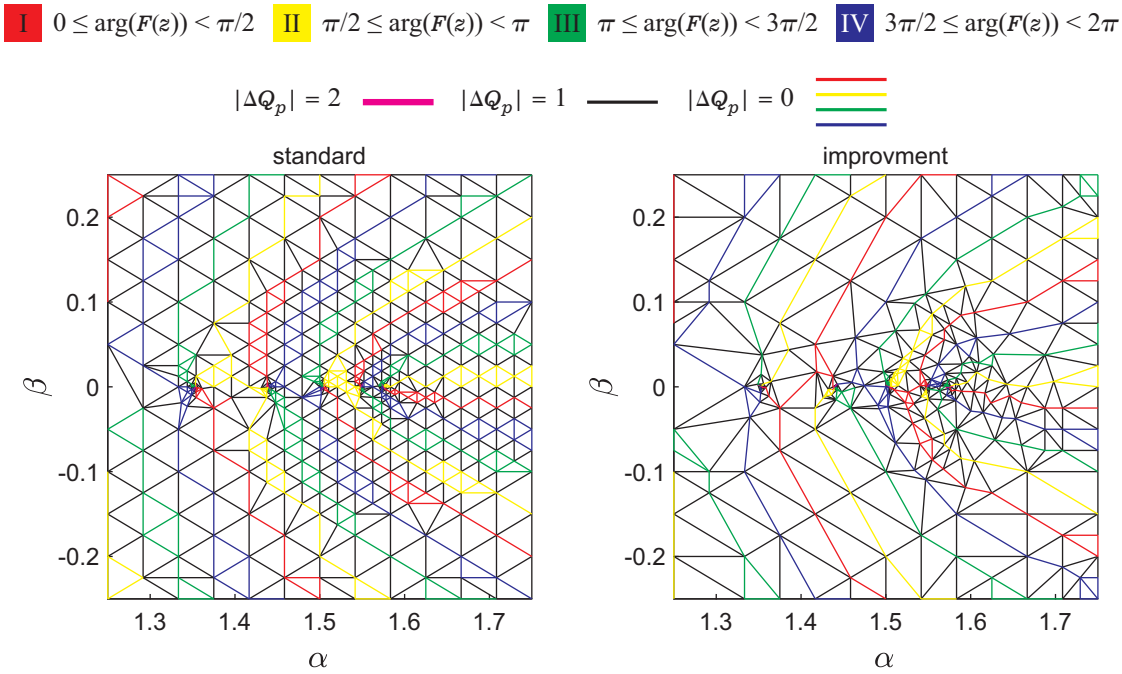

Figure 3.17: The comparison of the resulting final meshes for the standard and improvement versions of GRPF.

Table 3.4: The comparison of results obtained from analysis using the improvement and standard algorithms (standard approach in brackets [104])

accuracy δ	**time** $[\mathrm{s}]$	**no. of function calls**	**no. of iterations**
10^{-3}	$0.07\ (0.05)$	$321\ (461)$	$24\ (8)$
10^{-6}	$0.11\ (0.09)$	$528\ (769)$	$40\ (19)$
10^{-9}	$0.17\ (0.15)$	$730\ (1\,060)$	$53\ (29)$
10^{-12}	$0.24\ (0.22)$	$948\ (1\,388)$	$69\ (39)$

Comments

In summary, the modified algorithm effectively decreases the number of nodes, leading to a notably shorter analysis time. With fewer function calls required, the new approach can be up to three orders of magnitude faster. Notably, the numerical results are identical to the references, confirming the thesis: ”The optimal process of mesh densification in the GRPF algorithm [104] will improve the efficiency of the procedure and increase the probability of finding all roots in a fixed search region”.

3.4.2 Difficulties with ”skinny” triangles

The initial mesh in the regular approach assumes the division of the investigated domain Ω into equilateral triangles inside (and half triangles on the boundary) of the given region using the Delaunay triangulation algorithm, as shown in Figure 3.7. The following steps of the algorithm add new points in the centers of the suspected edges, i.e., between nodes where the phase differs by two quadrants, as outlined in subsection 3.4. This results in a new mesh being created. However, suppose subsequent iterations add new nodes only in specific regions of the domain that are very close to each other. In that case, it may lead to a significant disproportion between the lengths of the sides of the triangles. In subsequent iterations, the mesh refinement may lead to the formation of an ill-conditioned layout, i.e., ”skinny” triangles (see Figure 3.18). It is adverse from the point of view of triangulation stability and may result in the creation of duplicate nodes. Therefore, a fundamental research problem related to the meshing algorithm can be distinguished:

- How can one control the ”health” of the mesh (ideally, elements are identical), i.e., how to add new nodes to prevent deformation of the mesh (formation of ”skinny” triangles)?

The standard and improved GRPF (refer to subsection 3.4) only conducts a mesh condition analysis in the first and second zones of the ”candidate edges”. The first zone includes triangles with edges (at least two vertices) that are tangent to the ”candidate edges”. The second zone consists of triangles with only one vertex tangent to the ”candidate edges” (see Figure 3.18). Consequently, for all triangles in the first and second zones, the lengths of each side are investigated

to determine their mutual ratio. If this ratio exceeds an assumed value, typically, this value is three based on prior experience and tests; the triangle is supposed to be ”skinny.” To handle this disproportion and ensure edge length uniformity, an additional node is added to the centroid of this triangle. The solution limits the increase in disproportions between the edges to some extent but does not eliminate the formation of ”skinny” triangles.

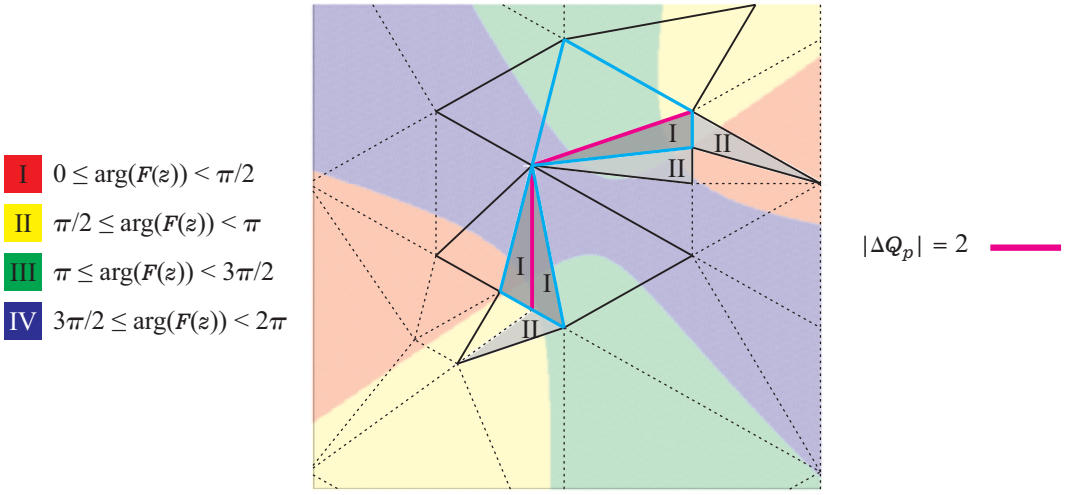

Figure 3.18: The example of ”skinny” triangles (filled in gray) occurring on the mesh for an example function. Bold magenta lines represent ”candidate edges.” The first zone, i.e., the ”candidate regions” boundary, is marked as a solid cyan line. The black solid bold line represents the additional second zone. The remaining edges on the mesh are marked with a lightweight black dashed line.

The new solution for ”skinny” triangles

After analyzing the possible difficulties resulting from occurring ”skinny” triangles, a new approach to this issue is proposed. Firstly, there is a change in determining which triangle is ”skinny.” Similar to the regular approach, only the first and second zones around the ”candidate edges” are analyzed. The difference is that the ratio of the longest edge to the lowest height is checked in all triangles from the first and second zones. The larger this ratio, the more ”skinny” the triangle is. Typically, a triangle is assumed to be ”skinny” when this parameter has a value of 5 (for the first zone) and 10 (for the second zone). One meaningful change is adopting a new technique to add new nodes in ”skinny” triangles. In order to control the stability of the mesh [152] and its correct distribution, it is assumed that new nodes can only be added by splitting existing edges. From the point of view of triangulation, adding new points in unrestricted areas is not recommended as it requires more direct control over the distances between new nodes. Therefore, a new point should be added at the midpoint of its longest edge in each ”skinny” triangle.

The idea of the formation of ”skinny” triangles, their detection, and the method of eliminating them for both the regular approach (i.e., improvement - see subsection 3.4) and the new one is presented in the first few iterations. Figure 3.19 shows noticeable differences in how new points are added and the mesh is created.

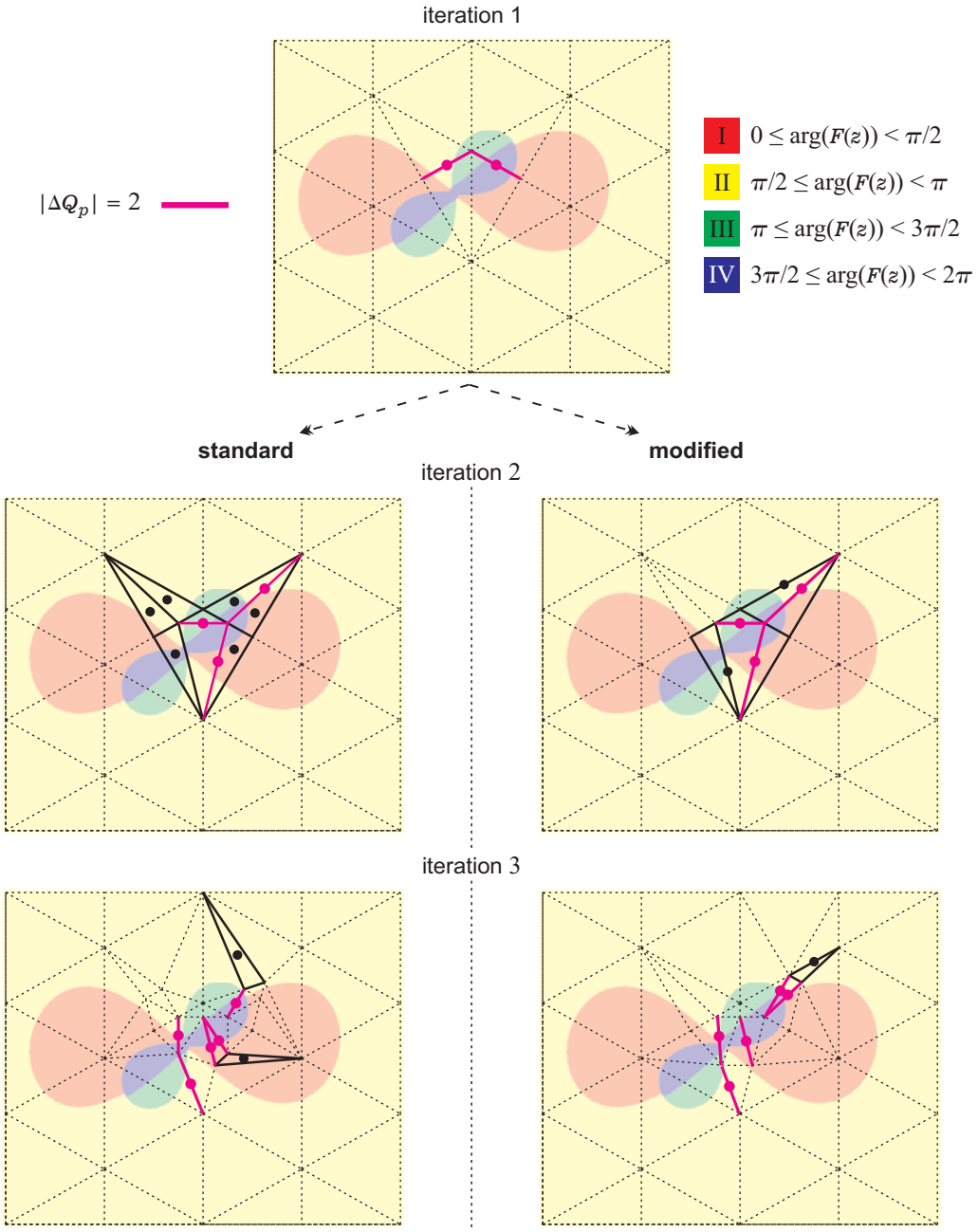

Figure 3.19: Two types of approaches (standard and newly proposed) for ”skinny” triangle problems are compared in three successive steps of an example mesh refinement process. Bold magenta lines represent ”candidate edges.” The black solid bold line represents the ”skinny” triangles. The remaining edges on the mesh are marked with a lightweight black dashed line. The new nodes are labeled with a magenta dot (by split ”candidate edges”) and a black dot by nodes added in the ”skinny” triangles.

In the case of the regular approach, adding new nodes does not eliminate the problem with the ”skinny” triangle. The presented numerical tests will show how the applied change affects the number of points, the final mesh, and the analysis results of example problems. The outcomes will be compared with the regular GRPF in the proposed improvement version.

Triangular dielectric fiber

In the first example, a triangular dielectric fiber is considered (see subsection 2.3.5), which is already tested in subsection 3.4. The investigated domain is the same:

$$\Omega = \{z \in \mathbb{C} : 1.2 < \mathrm{Re}(z) < 1.3 \wedge -0.45 < \mathrm{Im}(z) < -0.35\}$$

The initial mesh shown in Figure 3.20 contains $N_0 = 18$ nodes at the $r = 0.04$ parameter.

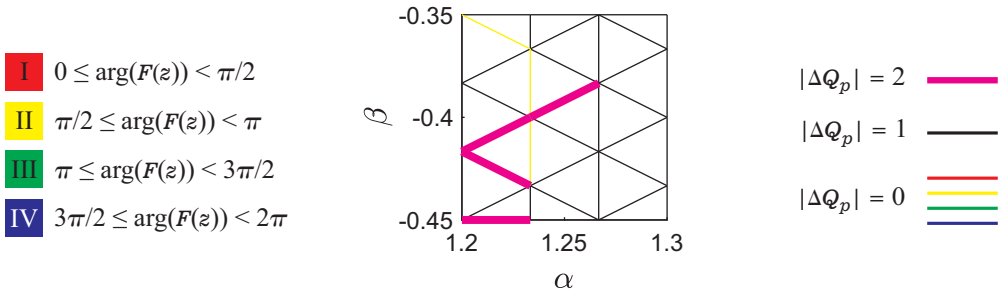

Figure 3.20: The initial mesh setup is identical for both the standard and modified approach.

In both approaches, the GRPF algorithm finds a leaky mode characterized by a normalized propagation coefficient $z = \gamma$ (with $\delta = 10^{-12}$ precision):

$$\gamma^{(1)} = 1.260551725649258 - j\,0.384838512540439$$

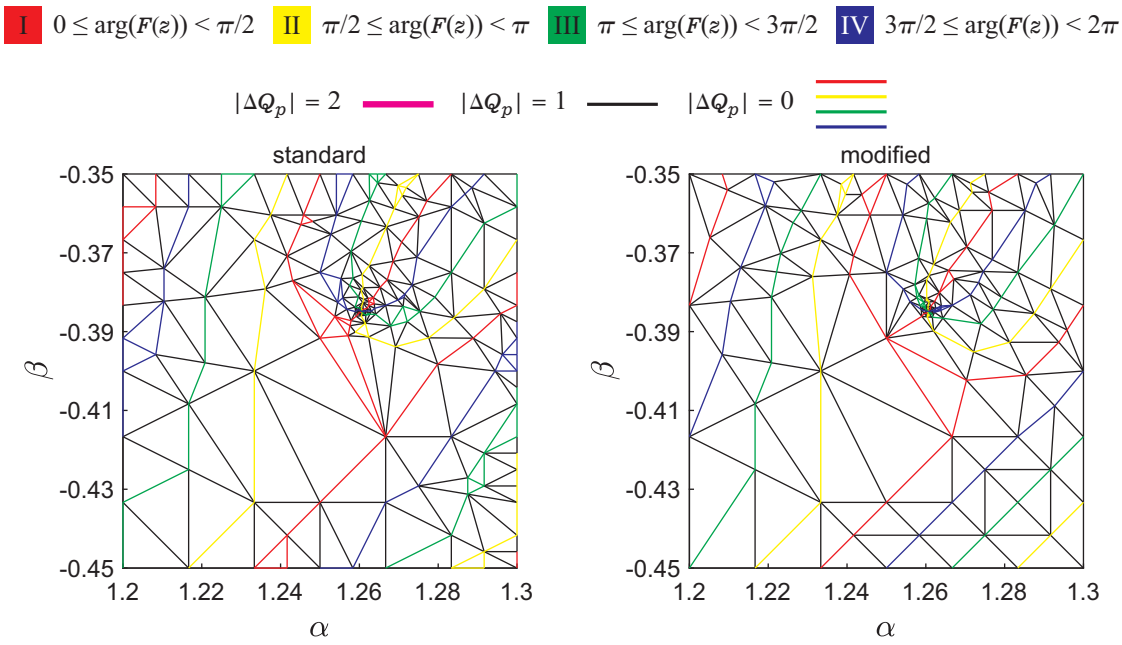

Figure 3.21: The comparison of final meshes for the standard and modified solution of the ”skinny” triangle issue.

Table 3.5: The comparison of the number of points, analysis times, and the number of iterations performed using the modified and standard algorithms (the standard approach for ”skinny” triangles in brackets).

accuracy δ	**time** [s]	**no. of function calls**	**no. of iterations**
10^{-3}	20 (24)	150 (181)	39 (31)
10^{-6}	30 (36)	232 (269)	45 (39)
10^{-9}	42 (48)	315 (363)	69 (61)
10^{-12}	52 (62)	394 (467)	90 (83)

Lossy multilayered waveguide

As another example, the lossy multilayered waveguide presented in subsection 2.3.1, as also analyzed in subsection 3.4, is considered. Again, the found roots of the function $F(z)$ represent the normalized propagation coefficients $z = \gamma$. This problem is analyzed in the region of:

$$\Omega = \{z \in \mathbb{C} : 1 < \mathrm{Re}(z) < 1.6 \wedge -0.25 < \mathrm{Im}(z) < 0.25\}$$

The initial mesh shown in Figure 3.22 contains $N_0 = 12$ nodes at the $r = 0.25$ parameter.

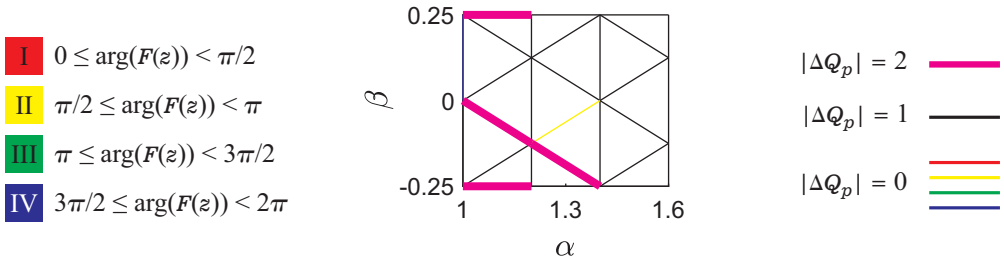

Figure 3.22: The initial mesh setup is identical for both the standard and modified approach.

Table 3.6: The comparison of the number of points, analysis times, and the number of iterations performed using the modified and standard algorithms (the standard approach for ”skinny” triangles in brackets).

accuracy δ	**time** [s]	**no. of function calls**	**no. of iterations**
10^{-3}	0.31 (0.35)	337 (368)	22 (19)
10^{-6}	0.38 (0.43)	522 (657)	39 (36)
10^{-9}	0.45 (0.53)	749 (956)	54 (51)
10^{-12}	0.54 (0.62)	973 (1 288)	72 (66)

The standard GRPF and modified ”skinny” triangle approach for this algorithm found (with an accuracy of $\delta = 10^{-12}$) five guided modes characterized by normalized propagation coefficients:

$$\gamma^{(1)} = 1.353140429034829 - j\,0.000086139366031$$
$$\gamma^{(2)} = 1.439795543998480 - j\,0.000052001476288$$
$$\gamma^{(3)} = 1.504169866194328 - j\,0.000028029233217$$
$$\gamma^{(4)} = 1.548692244042953 - j\,0.000012100934982$$
$$\gamma^{(5)} = 1.574863045538465 - j\,0.000002974495292$$

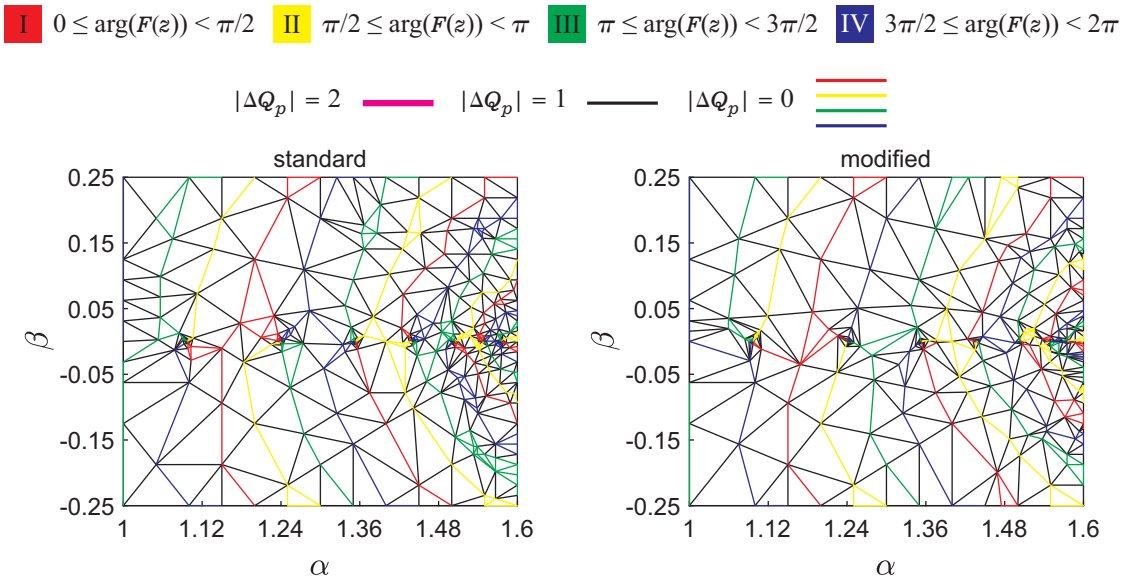

Figure 3.23: The comparison of final meshes for the standard and modified solution of the ”skinny” triangle issue.

Comments

Upon careful analysis of the mesh presented in Figure 3.21 and Figure 3.23, as well as the data outlined in Tables 3.5 and 3.6, it becomes apparent that the resulting roots are identical for both techniques. The divergence lies in the efficiency of the approaches. The decreased CPU time is attributed to the reduced number of nodes in the proposed implemented ”skinny” triangle solution. Nevertheless, these differences are minor, and the computation times are similar for relatively simple functions. The essence of the change, however, is the shape of the mesh, which is more regular and stable. It again proves the thesis: ”The optimal process of mesh densification in the GRPF algorithm [104] will improve the efficiency of the procedure and increase the probability of finding all roots in a fixed search region”. In the further part of this dissertation, the algorithm utilizing the proposed modifications, i.e. an improved method of mesh refinement and a modified approach to ”skinny” triangles, will be called ”regular” GRPF.

3.5 Initial mesh generation

As explained in the previous section, the GRPF algorithm has a main limitation in its initial mesh. The current version of the algorithm assumes a specific initial resolution r, i.e., the longest edges of the triangle. This parameter sets a particular number of initial mesh nodes N_0. As discussed earlier (see subsection 3.3), the accuracy of finding the roots of the function depends largely on this parameter. However, selecting an appropriate initial mesh requires detailed preliminary analysis, and there is still no guarantee that all zeros and poles will be located. These challenges highlight fundamental research issue related to the meshing algorithm:

- How can one choose the number of points (preferably automatically without user intervention) and the shape of the initial mesh (without knowing the analyzed function) so that the GRPF algorithm works efficiently and stably?

Properly addressing these questions is crucial to avoiding improper algorithm convergence. In some cases, in the first iterations of regular GRPF, the triangles are so large that the condition $|\Delta Q_p| = 2$ does not meet for any mesh edge. Consequently, certain regions will not be dense enough, and zeros or poles may be missed. However, every root has to be found, even if it is not exactly inside the ”candidate region.” A number of tests were performed, and several methods were developed, proposing various solutions for initial mesh generation. Simple refinement techniques have been used, interpolations, or even attempts have been made to use machine learning. The following subsections will present the research results related to creating the initial mesh.

3.5.1 Proposed methods

This section provides a short overview of the proposed solutions to the issue of initial mesh generation, which can overcome a limitation of the regular GRPF approach. The research focused on developing a stable mesh algorithm by closely examining the uneven refinement process, where some edges on the mesh are more critical than others. Accurately determining the significance of these edges is crucial to resolving the refinement issue. As part of the work below, several methods are proposed, including a simple mesh refinement algorithm, a more complex and extensive interpolation approach, and a final self-adaptive technique based on isophases and their gradients (explained in more detail in subsection 3.6).

In the beginning, a simple algorithm is proposed in which the edge importance is determined by a standard DCAP approach-based analysis of increasing the function’s argument at the mesh nodes. To evaluate an edge’s significance, the difference in quadrants and distance (edge length) between the nodes is analyzed. The algorithm is developed based on trial and error methods, and the entire procedure, along with a numerical example, is presented in Appendix A.1. The algorithm is based on seven irregular iterations. It starts with five points and divides edges unevenly from two to four parts in the first four iterations. After the sixth iteration, it refines the mesh like the stan-

dard GRPF algorithm. The method identifies ”candidate regions” and adds points at the centers of the ”skinny” triangle on the entire domain. The process involves detecting all ”candidate regions” first and focusing on the density of regions around the ”candidate edges.” Refinements are made in each iteration to create a more refined mesh. However, there is no theoretical or mathematical evidence supporting the described scheme. Although the results initially seemed promising, further attempts to develop this approach were discontinued due to too many parameters and little universality. However, the tests showed that the method based merely on the phase-quadrant approach generates insufficient information about the function's variability and character. This conclusion led to a change in technique, and subsequent algorithms used other information, such as correlations between points.

The next algorithm is based on rational interpolation, and its operating guide and a numerical example are presented in Appendix A.2. It is a combination of classical interpolation with GRPF. Generally, there are many algorithms for finding the zeros of complex functions based on interpolation methods [105], [106]. One of the most popular is the raddisk [53] and its improvement AAA [54]. Detailed numerical tests for these methods will be presented in subsection 3.6.5, where the results will be compared to the final version of the adaptive meshing algorithm.

A new adaptive approach for generating an initial mesh is introduced using local rational interpolation. The algorithm consists of three stages: *mesh nesting*, *validation of the accuracy of the interpolation*, and the *selection of a densified edge*. The initial mesh generation is an iterative process of selecting four points and evaluating function values. If no correct interpolation exists, additional points are used in successive iterations. Nodes are divided into ”training” and ”validation” points to ensure accurate interpolation. The algorithm identifies the most accurate rational type and calculates its coefficients. The rational with the smallest error is selected. If the error is significant, the domain is recursively split, and points are added based on the variability of the function.

Generally, the proposed technique is effective mainly for function classes that a rational (or polynomial) function can represent. The algorithm depends heavily on the degree of the interpolating polynomials (in the numerator and denominator), especially for investigating the large domain Ω and rapidly varying functions. In addition, because of the finite number of mesh nodes (”training” and ”validation” points), this approach does not guarantee finding all zeros and poles. Another disadvantage of the described solution is the problem with procedure convergence. It manifests itself by looping the algorithm and lowering its efficiency. The level of recursion may be very high, increasing computational complexity. This results in an increase in the total time of the analysis [1] even though the total number of added nodes is not significant (in one iteration, it is added from 5 to 9 points).

[1] It is particularly noticeable for simple functions, where the execution time of a single iteration of the interpolation procedure takes longer than evaluating a single function value.

As part of the solution to the issue of generating the initial mesh, an algorithm based on machine learning of phase portraits was also initially attempted. As a result, the interpolation algorithm presented includes a process of dividing nodes into ”training” and ”validation” points, which is inspired by machine learning concepts. Attempts were also made to extend the interpolation algorithm with the concept of the harmonic function. Such an approach is based on the assumption that if a function is harmonic, the average value of the function for boundary nodes in a given region should be equal to the function value in the center of this area. Nevertheless, testing indicated that this method may not be as precise and effective as desired. Finally, another issue regarding the interpolation algorithm is the large number of parameters. Even their minor changes can significantly affect the algorithm's operation. Some of these coefficients are selected experimentally, such as the accuracy percentage in the error matrix. Hence, the algorithm in this form is considered unsuitable for the initial mesh generation, and attempts to develop it further were discontinued. However, it is believed that interpolative methods can be applied as an add-on to standard CAP approaches (not as a stand-alone algorithm). Using a smaller number of points due to interpolation, the phase character of the function can be predicted to some extent and, for example, used to create an approximate phase portrait.

In summary, the aforementioned methods have their limitations, resulting in their discontinuation. However, the conducted tests opened the way for new research ideas, such as the isophase concept, which will be discussed in the next subsection. Overall, utilizing any adaptive algorithms provides valuable information about the investigated function, enhancing the success of root finding of complex functions.

3.6 Self-adaptive mesh generator

This subsection will present the final version of the algorithm for a self-adaptive initial mesh generator for GRPF. As outlined in [116], the result of the proposed technique is a triangular initial mesh, which becomes the starting point for the regular approach. A significant benefit of this method is its ability to automatically refine the most critical regions in the mesh from the point of view of root finding.

As described more than once before, the starting point for a regular GRPF algorithm is to discretize the investigated function using a unified mesh composed of N_0 nodes. In general, a larger number of points decreases the stake of missing zeros or poles and improves the precision of the results. In the proposed new technique, similarly to regular GRPF, the user establishes the search domain $\Omega \subset \mathbb{C}$, the expected precision of results δ, and the number of discretization points of the function in the initial mesh (in the condition of the N_{max} parameter). It should be noted that the maximum edge lengths are not defined in this case. The basis of the operation of the presented algorithm is the analysis of the behavior of curves representing a constant level of the function arguments, i.e., isophases in the considered domain (see Figure 3.24). This concept will be discussed in more detail in the next subsection. In particular, the investigation of their density

and their curvature. The proposed solution analyzes the fastest raises of phases in a specific direction (gradient) to identify areas requiring mesh refinement where zero and pole are close to each other.

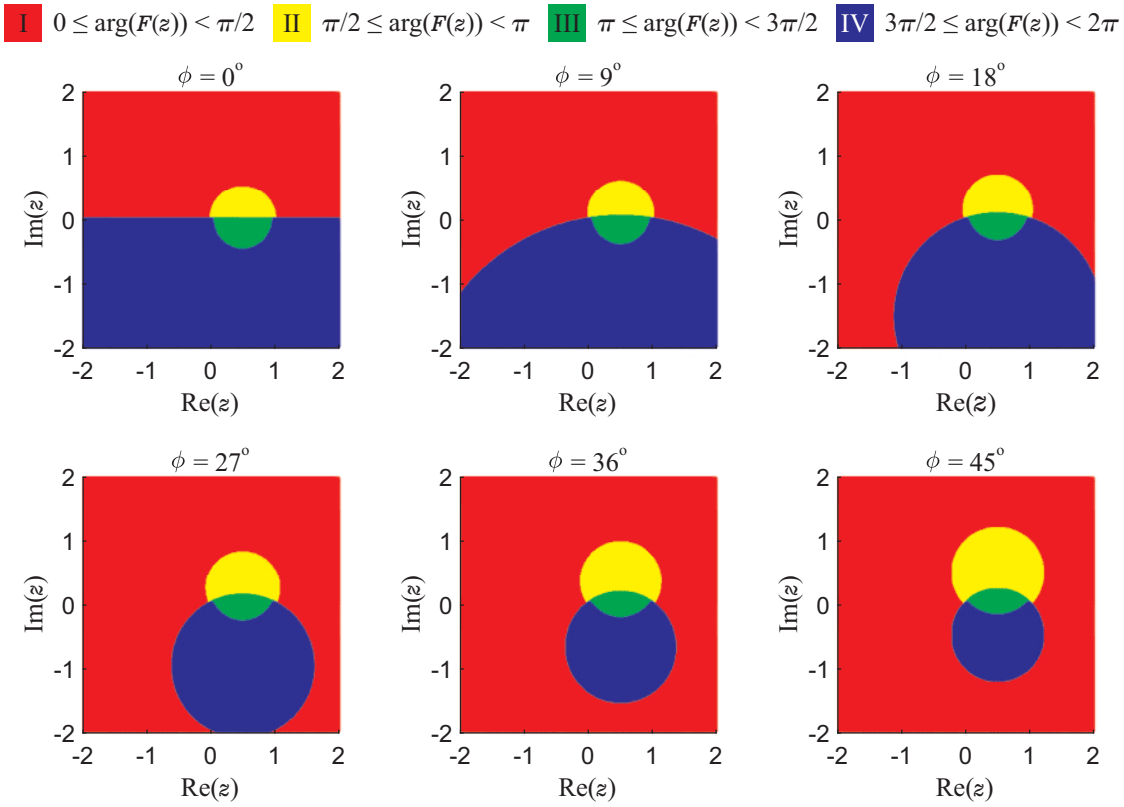

Figure 3.24: The phase portraits of the function $F(z) = \frac{z-1}{z}$ multiplied by $e^{j\phi}$ (for several example values of the angle ϕ from 0 to $\pi/4$). The change of ϕ represents a constant isophase level (shift of the argument of the function F(z)).

3.6.1 Isophases and gradient analysis

The working principle of the new initial mesh generator for the GPRF algorithm will be demonstrated using a simple, rational function as an example, see (3.12), in the following domain:

$$\Omega = \{z \in \mathbb{C} : -1 < \mathrm{Re}(z) < 1 \wedge -1 < \mathrm{Im}(z) < 1\}$$

already analyzed at the beginning of this chapter in the issue of ”islands” in subsection 3.3.

$$F(z) = \frac{(z - z_a)(z - z_b - \epsilon)}{(z - z_c)(z - z_b + \epsilon)} \tag{3.12}$$

where:

$z_a = \frac{1}{2} - \frac{\sqrt{3}}{6}j$,

$z_b = \frac{\sqrt{3}}{3}j$,

$z_c = -\frac{1}{2} - \frac{\sqrt{3}}{6}j$,

$\epsilon = 0.01$ - size of the ”island”.

In order to present the grounds of the new algorithm more precisely, it is necessary to introduce the concept of isophases. These are the curves that consist of points inside the search domain for which the function's argument (i.e., phase) takes a constant value. As shown in Figure 3.25, between zero and the pole, there is an infinite number of isophases from $-\pi$ to π. The above property is utilized in the proposed technique.

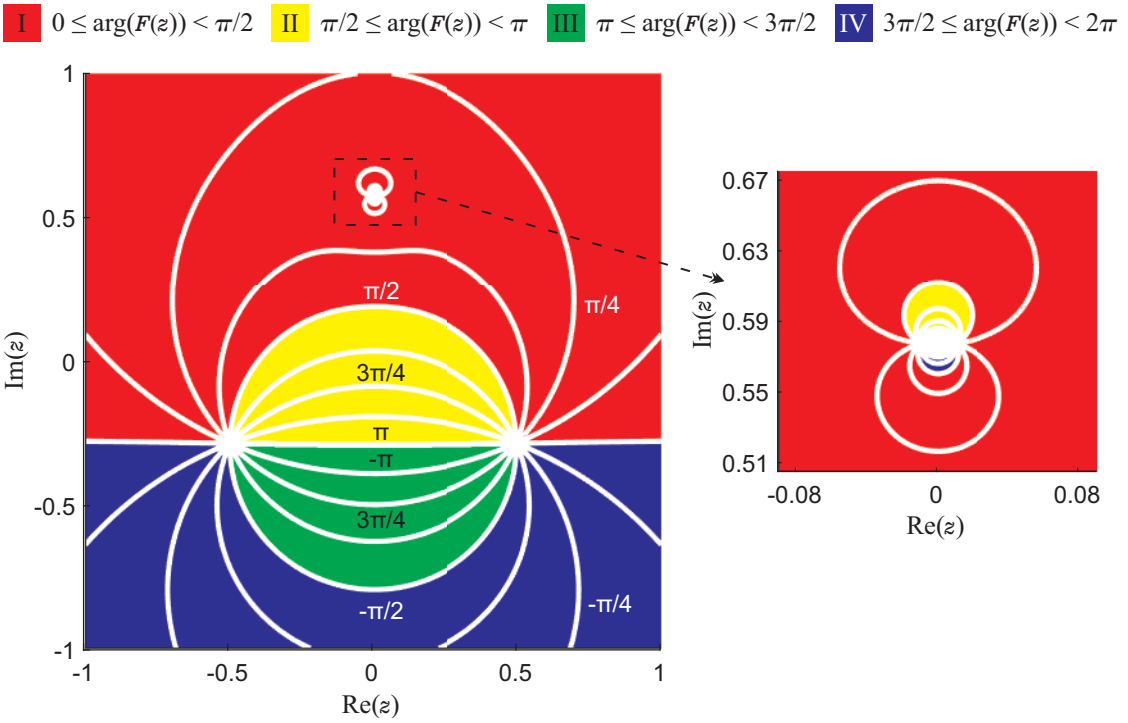

Figure 3.25: The phase portrait for the rational function from equation (3.12) and the "island" zoom ($\epsilon = 0.01$) with the example isophases plotted (white curves).

Let us consider a mesh in which the argument of the investigated function at its nodes is known, as shown in Figure 3.26. The phase increment along each side $\Delta\phi_{AB} = \phi_B - \phi_A$, $\Delta\phi_{BC} = \phi_C - \phi_B$ and $\Delta\phi_{CA} = \phi_A - \phi_C$ can be analyzed in every n-th triangle ABC in the mesh. According to CAP, if no zeros or poles are inside the triangle, the sum of the phase increments around the circumference becomes zero. When there is a zero inside, the function's argument changes about 2π or a multiple in the case of multiple zero. Similarly, poles and their multiplicities will be $-2k\pi$. However, when there is a zero-pole pair in a triangular space, the sum of the phases will also be zero. Therefore, if the triangles on the mesh are too large, finding "islands" becomes unlikely. Nevertheless, essential deductions can be drawn by investigating the phase changes along the edges and the mean phase increase over the entire element. To this end, the direction of the fastest growth of the function argument (represented by the approximate gradient $\vec{g}_n$) is evaluated for each triangle (refer to Figure 3.26).

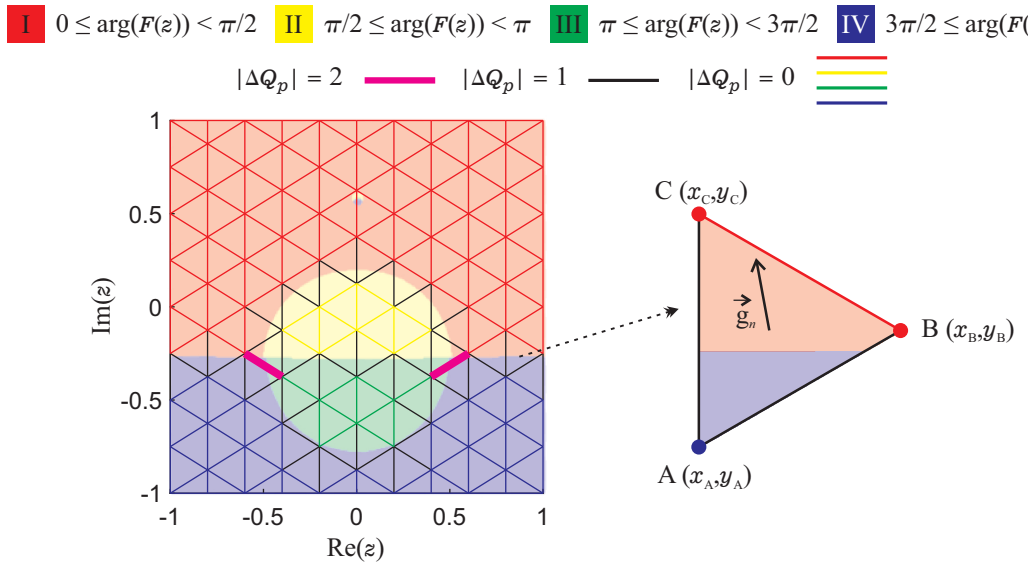

Figure 3.26: The mean gradient defined as a vector $\vec{g}_n$ for the nth individual mesh element.

The differences between the phases of the function values along the edge of the triangle are represented as $\Delta\phi_\zeta$, where $\zeta\epsilon\{AB, BC, CA\}$. Therefore, $\Delta\phi_{AB}$ is the difference between the phases of $arg(F(z_B))$ and $arg(F(z_A))$. Thus, the argument of the function in a triangle can be approximated using the equation of a plane:

$$\phi = ax + by + c \tag{3.13}$$

where:

a, b i c are unknown coefficients.

The following vector can express the gradient of the function argument (phase gradient) in two-dimensional space:

$$\vec{g}_n = [a, b] \tag{3.14}$$

Thus, the coefficient c from equation (3.13) is negligible. From the point of view of the GRPF algorithm, only the phase increase is crucial, and the exact function value is, in fact, non-essential. Let us assume that the reference point is point A, and the phases at points B and C are $\phi_B = \phi_A + \Delta\phi_{AB}$ and $\phi_C = \phi_A + \Delta\phi_{AC}$. By substituting $\Delta\phi_\zeta$ and the coordinates of the vertices, the following system of equations is obtained:

$$\begin{cases} \phi_A = ax_A + by_A + c \\ \phi_A + \Delta\phi_{AB} = ax_B + by_B + c \\ \phi_A + \Delta\phi_{AC} = ax_C + by_C + c \end{cases} \tag{3.15}$$

If c is determined from the first equation, the gradient can be obtained from the matrix equation:

$$\begin{bmatrix} x_B - x_A & y_B - y_A \\ x_C - x_A & y_C - y_A \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \Delta\phi_{AB} \\ \Delta\phi_{AC} \end{bmatrix} \tag{3.16}$$

It should be noted here that the determination of the gradient is possible only if the change of the argument of the function along successive sides adds up to zero:

$$\Delta\phi_{AB} + \Delta\phi_{BC} + \Delta\phi_{CA} = 0 \tag{3.17}$$

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 unambiguously determined.

Let us investigate the regular triangular mesh generated for the rational function from (3.12). It consists of $N_0 = 104$ nodes, as shown in Figure 3.27. It is worth noting that this discretization is sufficient to determine the zero in z_a and the pole in z_c ("candidate edges" with double quadrants change are in these locations). However, this mesh is too sparse to find the zero-pole pair around z_b Regardless, an approximate phase gradient is determined for each triangle, representing the direction of the fastest function's argument increase. In that case, it is easy to see that the region near the "island" also stands out strongly.

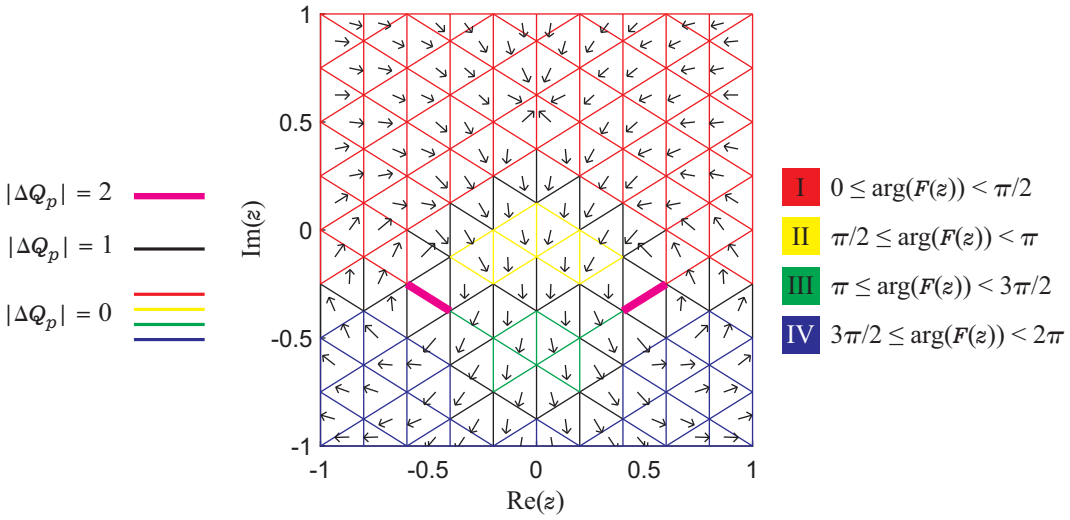

Figure 3.27: The black arrows represent normalized phase gradients, i.e., the direction of increments of the function argument on each mesh element (triangle).

3.6.2 Algorithm details

The change in the direction of the gradient in adjacent elements is directly related to the modification in the density or curvature of isophases. Taking advantage of this fact, for each inner edge in the mesh, the angle between the phase gradients for two neighboring triangles can be determined (while for edges from the border of the domain, this angle will be assumed to be zero):

$$\alpha_p = \measuredangle\{\vec{g}_{n_1}, \vec{g}_{n_2}\} \tag{3.18}$$

where:

n - element number,

p - edge number.

This angle will close in on zero value in the region where the function's argument changes slightly, which suggests no roots and poles. The nearest an edge is to zero or singularity, the greater the variation between the gradients and the greater the angle between them. The edges of the most significant angle α_p should be split in successive steps. Such an approach, nonetheless, would result in huge densification of the mesh only in the primarily found "candidate regions" around zero or the pole. Obviously, in these areas, the angles become more prominent the closer to zero or the pole one gets.

Consequently, the algorithm may not analyze the other parts of the domain. So, when refining the mesh, it is necessary to introduce the I_p indicator. This parameter determines the potential possibility of a zero or pole occurring near a particular edge as the product of the mentioned angle and the logarithm of the edge length normalized about the current resolution of the mesh:

$$I_p = \alpha_p \cdot \log\left(\frac{d_p}{D}\right) \tag{3.19}$$

where:
D - represents the minimum length of an edge in the mesh (ultimately D, should go to the δ).

This indicator goes to zero when the angle between phase gradients narrows, or the edge length decreases to the smallest of the mesh. The proposed process for generating an adaptive mesh involves iteratively determining the edges with the highest indices I_p and dividing them in half - adding new nodes to the mesh. In this way, in successive iterations, a mesh will be formed that is adjusted to the specific variability of the function. The process can be started from any mesh (including a uniform one) and adaptively refined in subsequent steps. It is worth noting that only a few points should be enough to start the procedure, e.g., defining the domain's edge (four points for a square or rectangular domain).

Another aspect the algorithm considers is analyzing the distribution and curvature of isophases. Phase gradient analysis indirectly analyzes this property, but one can additionally check whether some mesh edges intersect with the local extrema of such curves. When an edge based on the nodes z_A and z_B is split in half, the value of the function $f(z_M) = f_M = |f_M|e^{\phi_M}$ is evaluated at the midpoint point of this line segment. Thus, it can be easy to check whether the phase of the function ϕ_M at the new node z_M falls within the ambit of phase range defined by the argument of the function at the ends of the edge, as described by the following relation:

$$\min\{\phi_A, \phi_B\} \leq \phi_M \leq \max\{\phi_A, \phi_B\} \tag{3.20}$$

If the above condition is not met, the isophases along the edge cross the edge at least twice, see Figure 3.28 a). In this circumstance, additional mesh refinement in this region is necessary. Therefore, in subsequent iterations, all edges attached to nodes for which condition (3.20) is not met will be split in half by adding additional nodes.

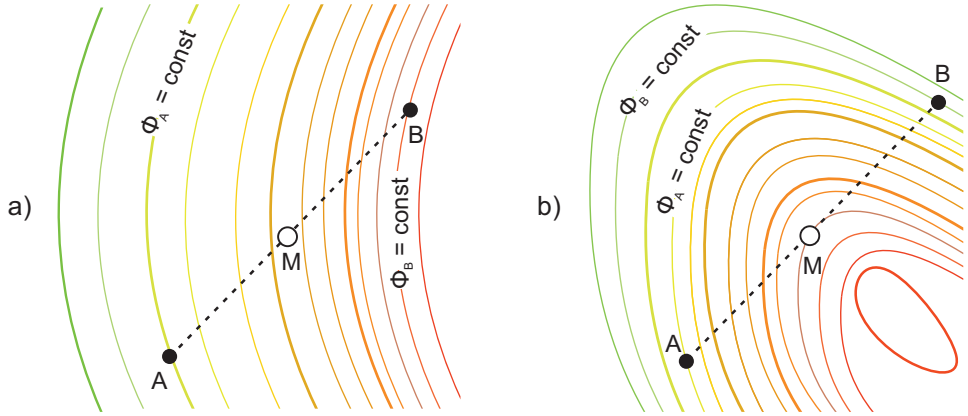

Figure 3.28: The conception of the analysis of the distribution and curvature of isophases. The isophases along the example edge: a) condition (3.20) is met, b) condition (3.20) is not met.

In summary, the proposed algorithm's procedure analyzes the gradients of the function's argument describing isophase distributions. In addition, the length of the edges in the current mesh is evaluated in each step. As in regular GRPF, edges with nodes between which the function argument differs by two quadrants ("candidate edges") are split. Also, the longest edges in elements with a significant disproportion of sides ("skinny" triangles) are reduced to improve mesh condition.

3.6.3 Algorithm steps

In order to organize the issues described in the previous section, let us illustrate the individual steps of the adaptive mesh generation algorithm. According to the block diagram shown in Figure 3.29, the algorithm consists of the following seven stages:

1. The user sets the initial conditions:
   - Defines the domain Ω - region of function analysis (N_0 points defining the boundary of the domain are defined by standard four points for a rectangular domain);
   - Imposes the number of nodes N_{max} (equivalent to the number of function calls) that will be used to create the mesh;
   - Specifies the accuracy with which the results are to be determined (in the form of the δ parameter, similar to the regular GRPF algorithm);
2. The value of the function in the new nodes is evaluated (in particular, the phase of the function, i.e., $arg(F(z_n))$), and Delunay triangulation is done;
3. The "candidate edges" are determined (i.e., all edges for which $|\Delta Q_p| = 2$) and edges for which condition (3.20) is not met (it is not possible to check in the first iteration);
4. For each new element n (triangle), the phase gradient $\vec{g}_n$ is determined and, based on it, the I_p indicator for each edge p;

5. The edges that will be divided in half to refine the adaptive mesh are selected - these are all the edges from point 3 and the same number of edges from point 4 with the highest indicator I_p (in the case of zero value of this parameter, the longest edges in the mesh are chosen). If there were no edges in point 3, then a minimum of one from point 4 is selected;

6. The shape of the mesh is analyzed. In order to eliminate ill-condition elements, the longest edges in each ”skinny” triangle are additionally split in half;

7. If the total number of nodes in the mesh does not exceed the assumed N_{max}, then new nodes are added to the mesh at the midpoint of the edges determined in points 5 and 6, and it is returned to point 2;

8. The user can change the N_{max} parameter during the process by observing (visualizing) the iterative refinement of the mesh.

Please note that point five of this algorithm involves the experimental selection of the largest I_p indicator's represented edge count. The number of edges that fail to meet the condition (3.20) reflects the current state of the mesh and indicates how many new nodes should be added to it. Adding too few points can cause slow convergence with many iterations. Although a large number of points guarantees proper mesh density, it may not be concentrated in critical regions. Therefore, manually modifying the number of added points can improve the algorithm's performance for critical issues. This adjustment provides additional opportunities to enhance the algorithm's performance for specific problems as more information is known about the function under investigation.

In summary, the proposed algorithm seems much more fitting and flexible than the regular GRPF approach, which necessitates restarting the process for a denser mesh. Therefore, it is unnecessary to perform several tests; only during one analysis can the number of iterations be freely increased. Thus, the analyzed function's complexity level can be observed in each iteration. However, the ”candidate edges” in the initial mesh may be longer and require several additional iterations to obtain higher precision results. Regardless, using the proposed initial mesh generator does not significantly impact the number of function calls when employing the standard GRPF technique to find zeros and poles in the final mesh refinement process.

The above description of the algorithm and the examples presented in the next subsection are parts of the publication [116] of the author of this dissertation. This part of the work also confirmed the thesis: ”The proper generation of the initial mesh in the GRPF algorithm [104] will improve the efficiency of the procedure and increase the probability of finding all roots in a fixed search region.” The source code of the Self-Adaptive Mesh Generator for GRPF can be found at: https://github.com/PioKow/SAGRPF.

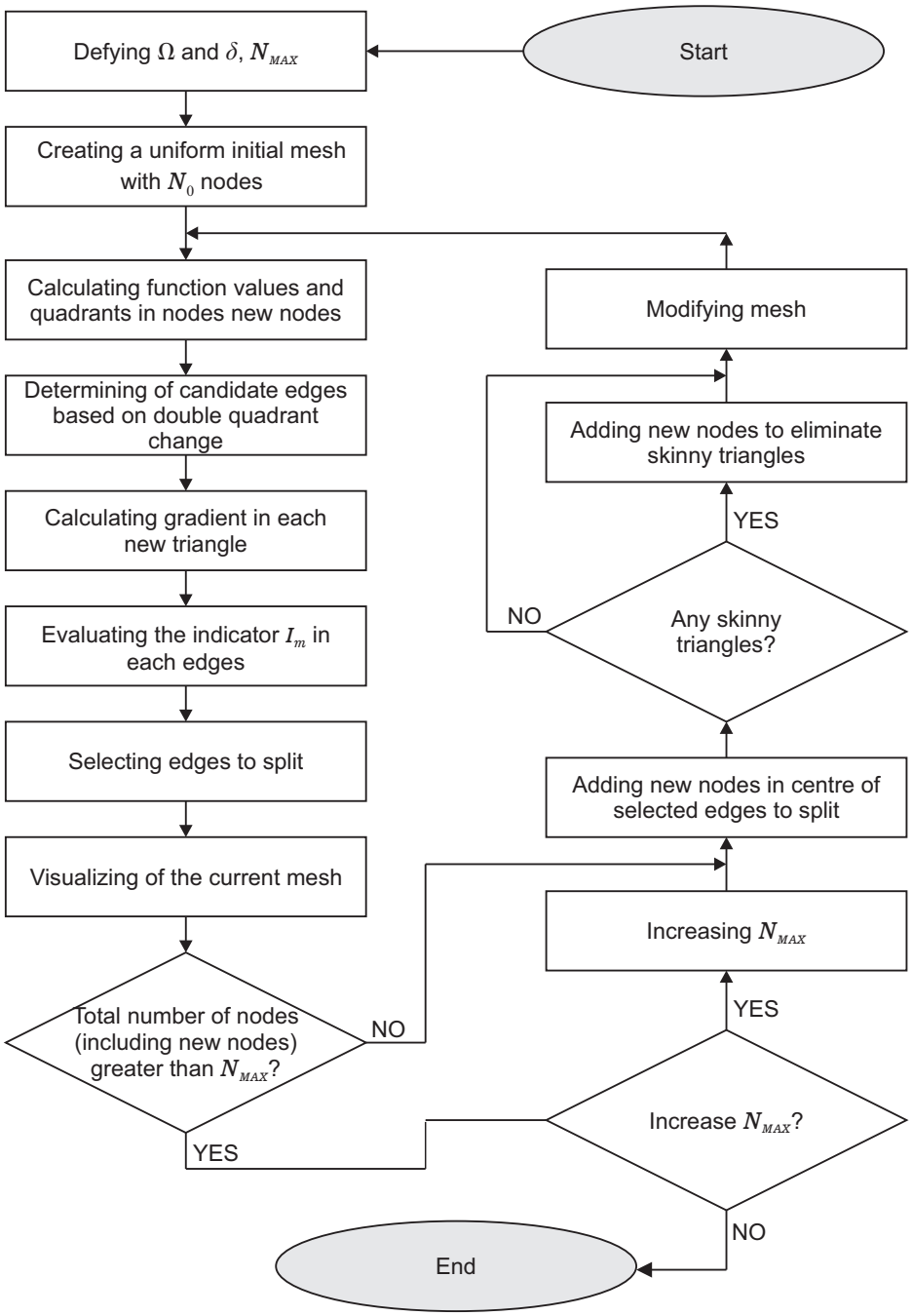

Figure 3.29: Block diagram of operating the proposed adaptive mesh generator for the GPRF algorithm.

3.6.4 Numerical examples

This subsection of the dissertation presents several practical numerical examples of operating the self-adaptive mesh generator for GRPF. Among the example problems presented in subsection 2.3, those containing ”islands” that are difficult to detect with standard CAP methods are tested. For each example, the tables detail the number of nodes sufficient to find all zeros and poles depending on the final δ accuracy. Furthermore, the adaptive and regular meshes are compared, with a particular emphasis on critical regions that impact the functionality of the standard GRPF algorithm (i.e., the ”islands”).

Simple rational function

First, numerical results are presented for the example discussed in the theoretical part (see subsections 3.3 and 3.6). The rational function from (3.12) for different values of the ϵ parameter is analyzed in the domain $\Omega = \{z \in \mathbb{C} : -1 < \mathrm{Re}(z) < 1 \wedge -1 < \mathrm{Im}(z) < 1\}$. Figure 3.30 shows the successive stages in forming an adaptive mesh for a function with an "island" of size $\epsilon = 10^{-2}$. The adaptive procedure starts with just $N_0 = 4$ points defining the vertices of the square domain.

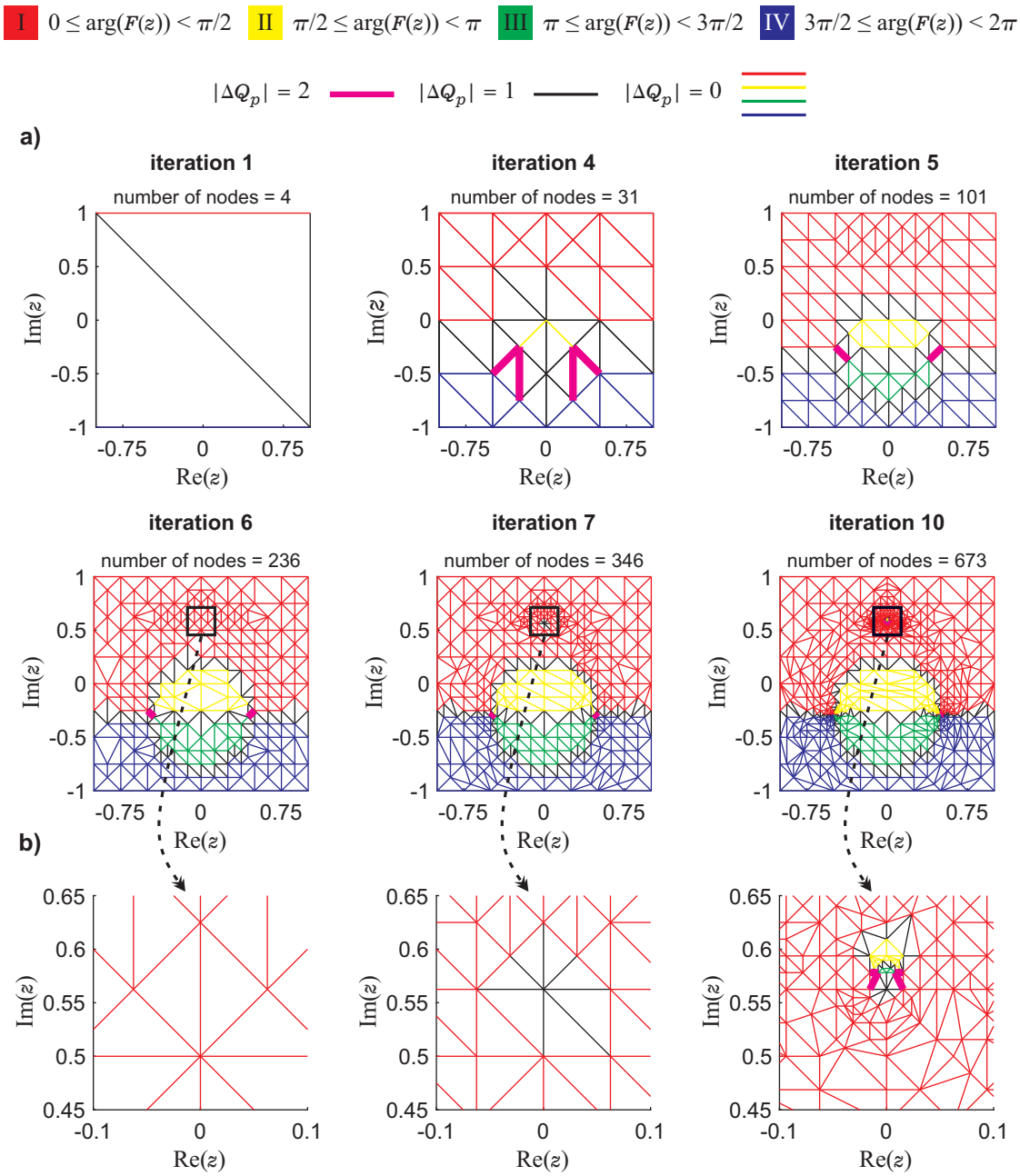

Figure 3.30: The iterative process of self-adaptive mesh generation, for function from (3.11) with $\epsilon = 0.01$ and $\delta = 0.001$, a) chosen example stages, b) zoom of "island" locations, respectively, in the demonstrated iterations.

Table 3.7 shows the number of nodes GRPF utilizes to find the assumed zeros and poles using the adaptive and regular meshes. As the data show for large ”islands,” i.e., considerably spaced between the zeros and poles, the benefit of using an adaptive mesh is slight. Nevertheless, for smaller epsilon, an adaptive mesh can reduce the number of nodes required by up to a thousand-fold compared to a regular mesh.

Table 3.7: The minimum number of nodes used by the GRPF (using a self-adaptive initial mesh generator) to find all zero and poles (including ”islands” of different sizes ϵ) for various accuracies δ. The number of nodes in the regular initial mesh is given in brackets.

”island” size ϵ	**accuracy** δ	**total no. of function calls**	**no. of initial nodes**
10^{-2}	10^{-3}	704 (3 137)	673 (3 038)
	10^{-6}	862 (3 323)	673 (3 038)
	10^{-9}	1 036 (3 561)	673 (3 038)
10^{-3}	10^{-3}	922 (1 157 748)	919 (1 157 734)
	10^{-6}	1 054 (1 157 931)	935 (1 157 734)
	10^{-9}	1 210 (1 158 077)	935 (1 157 734)
10^{-4}	10^{-6}	3 287 (28 883 805)	3 248 (28 883 662)
	10^{-9}	3 817 (28 883 971)	3 715 (28 883 662)

Dual-band cross-coupled eighth-order microwave filter

As a first applicable numerical example, the transfer function of the microwave filter presented in subsection 2.3.8 is investigated. The model concerns the function describing the filter scattering parameter $\mathrm{S}_{11}(s)$. The domain to be calculated is:

$$\Omega = \{z \in \mathbb{C} : -0.4 < \mathrm{Re}(z) < 0.2 \wedge 0 < \mathrm{Im}(z) < 1.2\}$$

Table 3.8: The minimum number of nodes used by the GRPF (using a self-adaptive initial mesh generator) to find all zero and poles (including ”islands”) for various accuracies δ. The number of nodes in the regular initial mesh is given in brackets.

accuracy δ	**total no. of function calls**	**no. of initial nodes**
10^{-3}	1 344 (1 567)	1 262 (1 435)
10^{-6}	1 656 (1 880)	1 262 (1 435)
10^{-9}	1 918 (2 253)	1 262 (1 435)

The investigated parameter z is $s = \sigma + j\omega$. The function is symmetric concerning ω, so only the region $\omega \geq 0$ is analyzed. A similar analysis was also carried out in [112] as a problematic case where an "island" seemed, which is a limitation for that algorithm. The zeros and poles found, enumerated below, make it possible to describe the $\mathbf{S}_{11}$ parameter from the scattering matrix of the filter:

roots:

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

poles:

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

As can be seen in Table 3.8, the number of points required to find all zeros and poles is similar for both methods - the adaptive mesh required $1\,262$ function calls, while the regular mesh required $1\,435$. This is mainly because the size of the "island" is not small relative to the dimension of the domain. However, it is worth emphasizing that finding all zeros and poles using a minimal regular mesh required a series of tests with different mesh densities. Using a sparser mesh than shown in Figure 3.31 does not ensure finding all zeros and poles in the domain. Such tests do not need to be conducted in the adaptive approach. The user can observe the changing mesh in the following steps and increase the number of points if necessary. In addition, as evidenced in Figure 3.31, the essential region from the point of view of finding zeros and poles has shorter edges than regions with more minor changes in the function's argument. It results in speeding up the final refinement step of the GRPF algorithm.

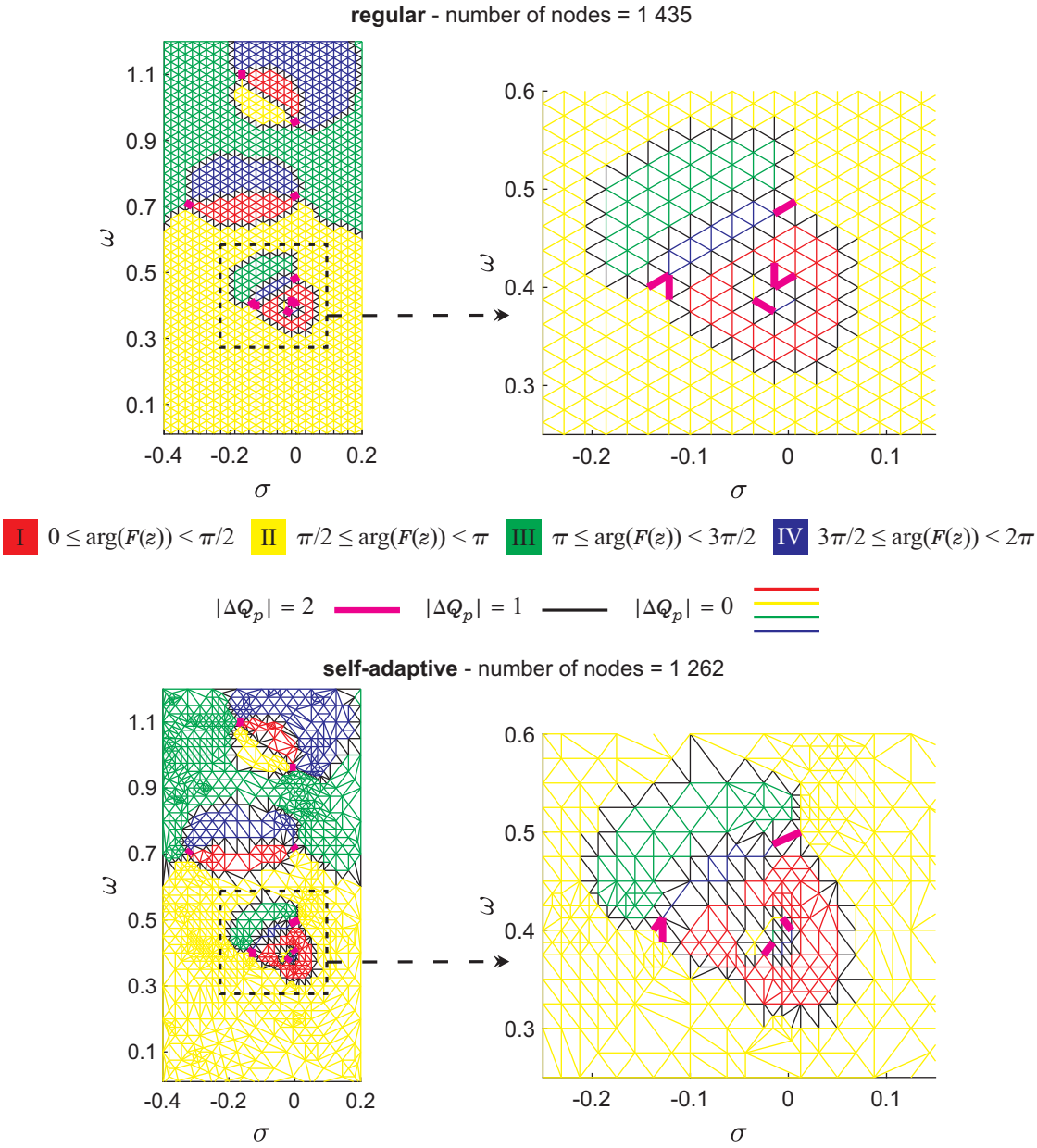

Figure 3.31: The initial meshes utilized to find all zeros and poles (including ”island”) in the investigated domain Ω with assumed accuracy $\delta = 10^{-3}$ for both approaches: regular and self-adaptive. On the right, a close-up of the ”island” region.

Fabry-Pérot open resonator

Another example includes the structure of the Fabry-Pérot open resonator presented in subsection 2.3.7. The system in the configuration with a plane mirror is analyzed. This example assumes material losses in the plate with permittivity $\varepsilon_r = 5 - 10j$. Therefore, the investigation region for the frequency $z = f$ contains a positive imaginary part. The real part of the domain is based on the reference publication [17]:

$$\Omega = \{z \in \mathbb{C} : 20 \cdot 10^9 < \mathrm{Re}(z) < 20.5 \cdot 10^9 \wedge 0.05 \cdot 10^9 < \mathrm{Im}(z) < 0.55 \cdot 10^9\}$$

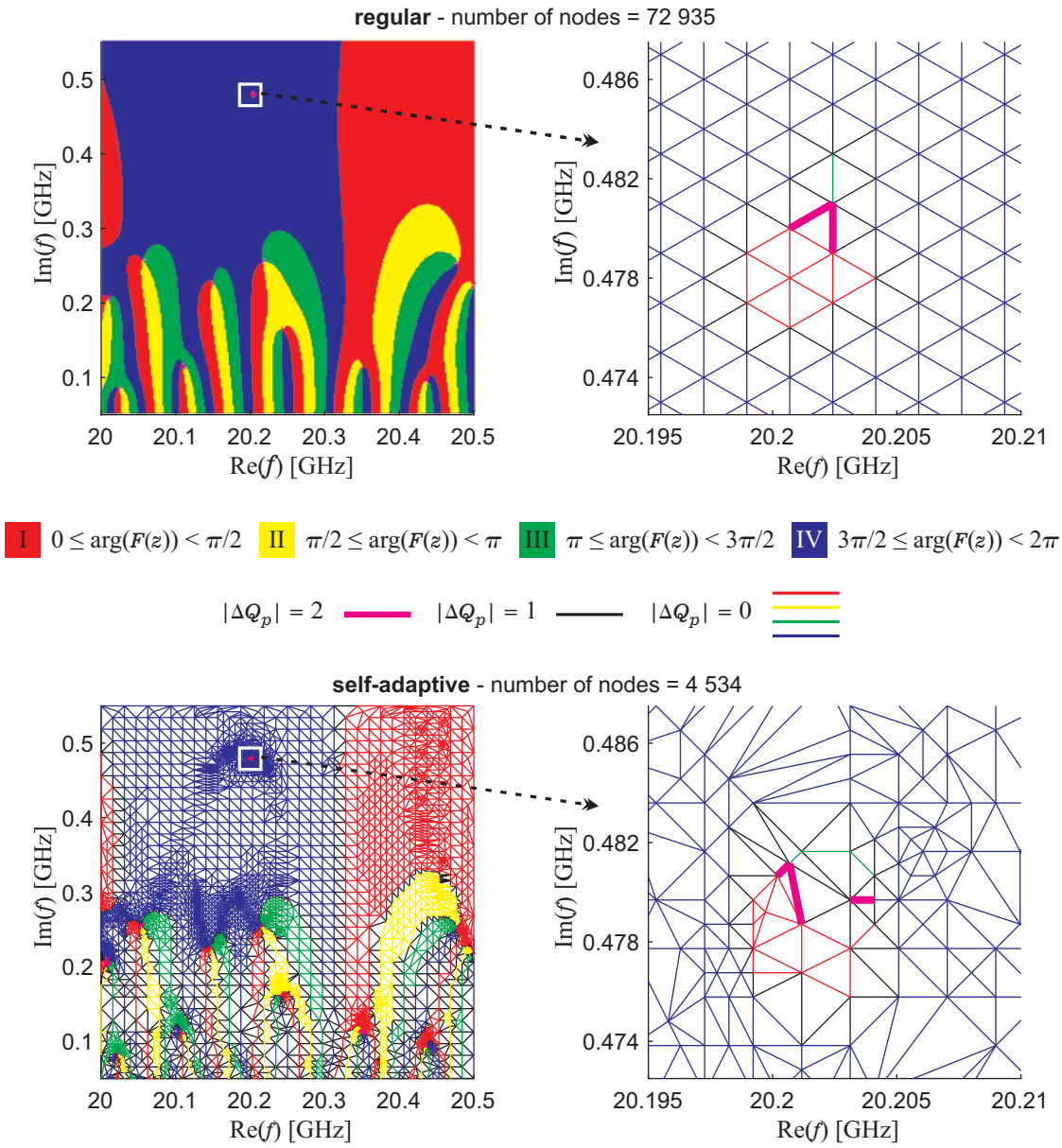

Figure 3.32: The initial meshes utilized to find all zeros and poles (including "island") in the investigated domain Ω with assumed accuracy $\delta = 10^{-3}$ for both approaches: regular and self-adaptive. On the right, a close-up of the "island" region.

Table 3.9: The minimum number of nodes used by the GRPF (using a self-adaptive initial mesh generator) to find all zero and poles (including "islands") for various accuracies δ. The number of nodes in the regular initial mesh is given in brackets.

accuracy δ	**total no. of function calls**	**no. of initial nodes**
10^{-3}	4 569 (73 071)	4 534 (72 935)
10^{-6}	5 104 (73 703)	4 605 (72 935)
10^{-9}	5 616 (74 099)	4 605 (72 935)

The following zeros and poles are found in the analysis domain:

eleven single zeros:

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

an additional single zero (see right side in Figure 3.32):

$$f_r^{(12)} = 20.200693325 + j\,0.480861643\ \mathrm{GHz}$$

an additional single pole (see right side in Figure 3.32):

$$f^{(13)} = 20.203665927 + j\,0.479927477\ \mathrm{GHz}$$

The analysis is completed for three different δ accuracies, and the efficiency of the processes is summarized in Table 3.9. Figure 3.32 shows the final meshes for both approaches. Notably, the self-adaptive initial mesh generator for GRPF uses almost 20 times fewer nodes than a regular mesh to find all zeros and poles of the resonator function in the considered Ω region.

Graphene transmission line

In the last example in this fragment of the work, a simple transmission line containing a graphene layer is investigated as outlined in subsection 2.3.3. In this problem, the normalized propagation coefficient $z = \gamma$ for TM modes is determined in the following range:

$$\Omega = \{z \in \mathbb{C} : -100 < \mathrm{Re}(z) < 400 \wedge -100 < \mathrm{Im}(z) < 400\}$$

Table 3.10: The minimum number of nodes used by the GRPF (using a self-adaptive initial mesh generator) to find all zero and poles (including "islands") for various accuracies δ. The number of nodes in the regular initial mesh is given in brackets.

accuracy δ	**total no. of function calls**	**no. of initial nodes**
10^{-3}	17 744 (4 510 733 668)	17 608 (4 510 733 255)
10^{-6}	18 546 (4 510 734 734)	17 728 (4 510 733 255)
10^{-9}	19 267 (4 510 735 232)	17 728 (4 510 733 255)

The following zeros and poles are found in the analysis domain:

eight single zeros:

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

two poles of the second order:

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

additional single zeros (see ”island” in Figure 3.33):

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

additional poles (see ”island” in Figure 3.33:

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

Similarly to the previous example, this calculation is performed for various δ accuracies. The numerical results are listed in Table 3.10, and the final meshes are shown in Figure 3.33[2]. By utilizing a self-adaptive technique to the initial mesh, the proposed algorithm is able to detect four additional zeros and two additional poles in the assumed domain. Detecting these regions through a regular mesh for GRPF would require a huge number of function calls. Furthermore, it requires additional primary information that particular regions have an ”island.” Without multiple pre-analyses, the detection of such regions would not be possible. It is unsurprising, as it would require a mesh with more than four billion nodes. In comparison, the adaptive mesh used hundreds of thousands of times fewer nodes and pointed all ”candidate regions” with just one analysis.

[2] In the background is a phase portrait obtained from a very dense function sampling.

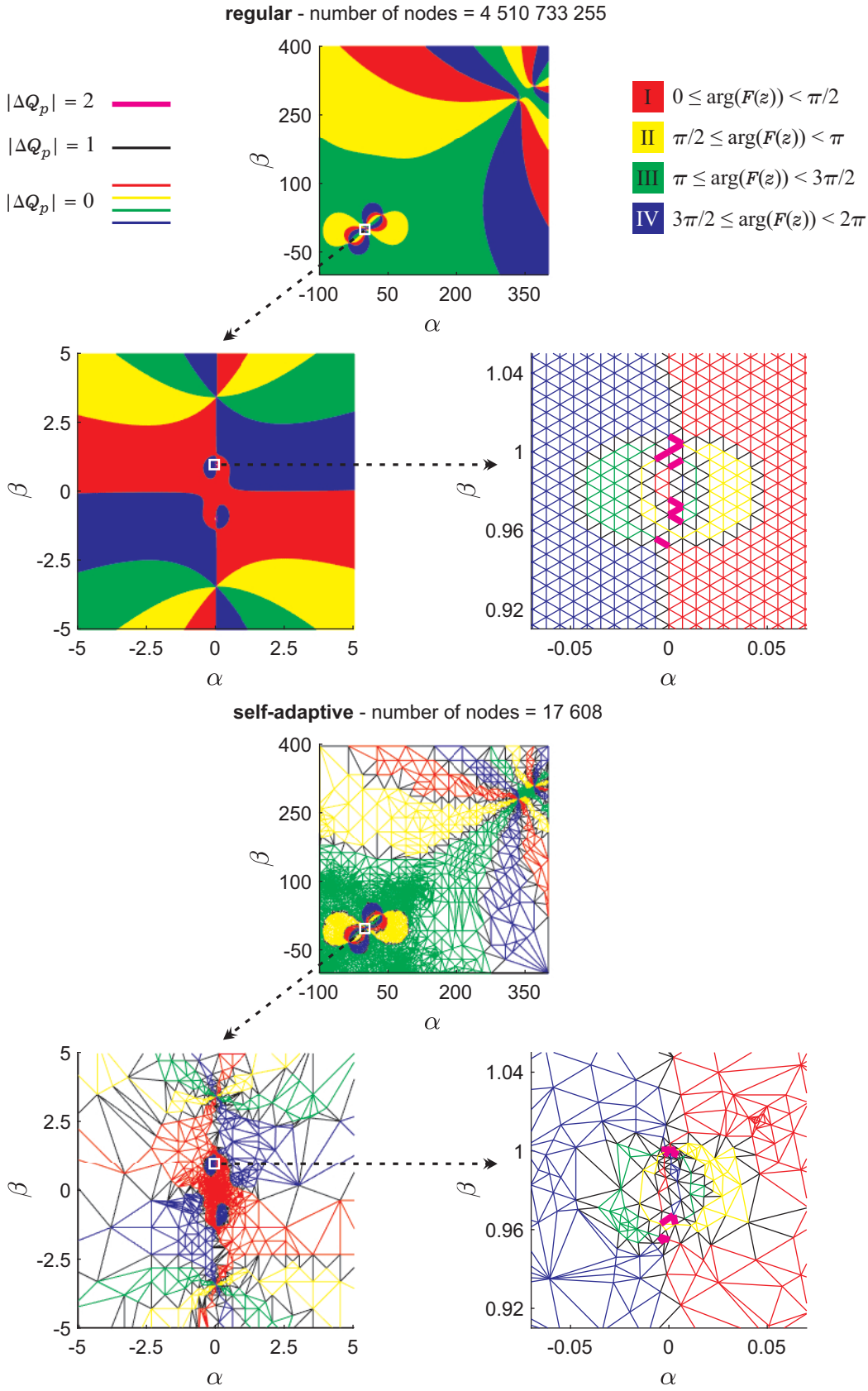

Figure 3.33: The initial meshes utilized to find all zeros and poles (including ”island”) in the investigated domain Ω with assumed accuracy $\delta = 10^{-3}$ for both approaches: regular and self-adaptive. A double magnification of the critical region of the ”island” is shown at the bottom and right.

3.6.5 Comparison with other algorithms

The following subsection compares the numerical results of the AAA algorithm [54] (in the circular and rectangular domain versions) with GRPF (including a self-adaptive mesh generator and improvement mesh refinement) for the same examples presented the previous subsection. The analysis is performed on the domains with similar sizes and number of points (number of function calls taken from subsection 3.6.4). The figures show the results obtained for the assumed number of points (thick colored continuous line at the domain boundary) in the AAA algorithm where "$\mathbf{x}$" denotes zero and "$\mathbf{o}$" denotes pole. The results are also gathered in the tables. The phase portrait is placed in the background to visualize the function behavior and illustrate the zeros and poles distribution.

Dual-band cross-coupled eighth-order microwave filter

As a first example, the transfer function of the microwave filter presented in subsection 2.3.8 is re-calculated. The investigated parameter $z \in \mathbb{C}$ is $s = \sigma + j\omega$. The analyzed rectangle is defined by the following curve:

$$C_r = \left\{ z \in \mathbb{C} : \begin{array}{c} -0.4 \leq \mathrm{Re(z)} \leq 0.2 \wedge \mathrm{Im(z)} = 0 \\ -0.4 \leq \mathrm{Re(z)} \leq 0.2 \wedge \mathrm{Im(z)} = 1.4 \\ \mathrm{Re(z)} = -0.4 \wedge 0 \leq \mathrm{Im(z)} \leq 1.4 \\ \mathrm{Re(z)} = 0.2 \wedge 0 \leq \mathrm{Im(z)} \leq 1.4 \end{array} \right\}$$

and circle is described by the following curve:

$$C_c = \{ z \in \mathbb{C} : |z - j\,0.7| = 0.65 \}$$

Table 3.11: Overview of found roots and singularities for the proposed self-adaptive initial mesh generator for GRPF and results obtained using the AAA method in two configurations.

zeros

z	**SA-GRPF**	**AAA circle**	**AAA rectangular**
$s^{(1)}$	$0.000000 + j\,0.407949$	$0.000000 + j\,0.407949$	$0.000000 + j\,0.407949$
$s^{(2)}$	$0.000000 + j\,0.491541$	$0.000000 + j\,0.491541$	not found
$s^{(3)}$	$0.000000 + j\,0.720326$	$0.000000 + j\,0.720326$	$0.000000 + j\,0.720326$
$s^{(4)}$	$0.000000 + j\,0.961957$	$0.000000 + j\,0.961957$	$0.000000 + j\,0.961957$

poles

	SA-GRPF	**AAA circle**	**AAA rectangular**
$s^{(5)}$	$-0.022514 + j\,0.378089$	$-0.022514 + j\,0.378089$	not found
$s^{(6)}$	$-0.128689 + j\,0.408631$	$-0.128689 + j\,0.408634$	$-0.128689 + j\,0.408634$
$s^{(7)}$	$-0.324725 + j\,0.702182$	$-0.324725 + j\,0.702182$	not found
$s^{(8)}$	$-0.167916 + j\,1.095438$	$-0.167916 + j\,1.095438$	$-0.167916 + j\,1.095438$

The visualization of the results is presented in Figure 3.34, and the summary of the found zeros and poles is in Table 3.11. The $1\,000$ points located evenly spaced from each other along the perimeter of the rectangle and circle are used for both analyses.

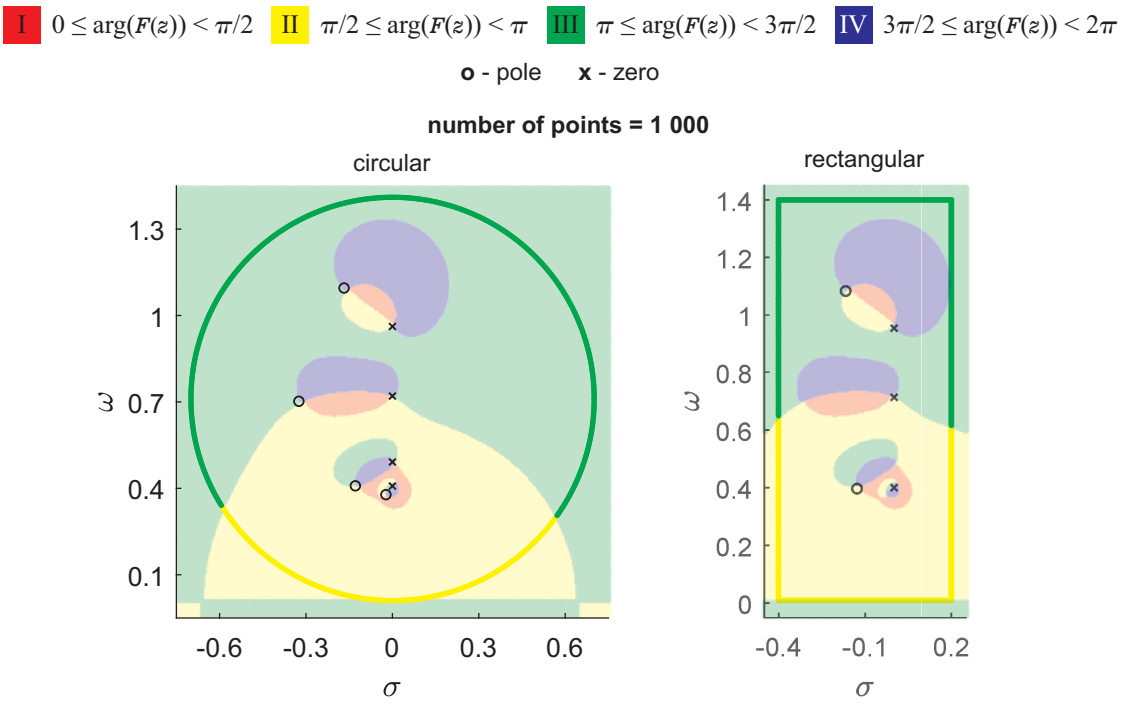

Figure 3.34: The results from the findings of zeros and poles using the AAA algorithm (circular domain on the left and rectangular domain on the right). In the background is a phase portrait obtained from a very dense function sampling.

One hundred points are enough for a circular domain in AAA to obtain the same results as from GRPF. A higher number of points does not bring an improvement in root finding. In the case of the rectangular domain in AAA, some of the zeros and poles are omitted.

Fabry-Pérot open resonator

The next example includes the structure of the Fabry-Pérot open resonator presented in subsection 2.3.7. The system in the configuration with a plane mirror is analyzed. This example assumes material losses in the plate with permittivity $\varepsilon_r = 5 - 10j$. The investigated parameter $z \in \mathbb{C}$ is frequency f [GHz]. The analyzed rectangle is defined by the following curve:

$$C_r = \left\{ z \in \mathbb{C} : \begin{array}{c} 20 \le \mathrm{Re(z)} \le 20.5 \wedge \mathrm{Im(z)} = 0.05 \\ 20 \le \mathrm{Re(z)} \le 20.5 \wedge \mathrm{Im(z)} = 0.55 \\ \mathrm{Re(z)} = 20 \wedge 0.05 \le \mathrm{Im(z)} \le 0.55 \\ \mathrm{Re(z)} = 20.5 \wedge 0.05 \le \mathrm{Im(z)} \le 0.55 \end{array} \right\}$$

and circle is described by the following curve:

$$C_c = \{ z \in \mathbb{C} : |z - 20.25 - j\,0.275| = 0.275 \}$$

The visualization of the results is presented in Figure 3.35, and the summary of the found zeros and poles is in Table 3.12.

Table 3.12: Overview of found roots and singularities for the proposed self-adaptive initial mesh generator for GRPF and results obtained using the AAA method in two configurations.

zeros

z	**SA-GRPF**	**AAA circle**	**AAA rectangular**
$f_r^{(1)}$ [GHz]	$20.007761 + j\,0.204953$	$20.084013 + j\,0.140736$	not found
$f_r^{(2)}$ [GHz]	$20.021725 + j\,0.082218$	not found	$20.030878 + j\,0.08804$
$f_r^{(3)}$ [GHz]	$20.053612 + j\,0.255720$	$20.010173 + j\,0.264031$	not found
$f_r^{(4)}$ [GHz]	$20.107273 + j\,0.112346$	$20.118882 + j\,0.080254$	$20.084013 + j\,0.140735$
$f_r^{(5)}$ [GHz]	$20.146603 + j\,0.226089$	$20.114028 + j\,0.177647$	not found
$f_r^{(6)}$ [GHz]	$20.214322 + j\,0.253186$	not found	not found
$f_r^{(7)}$ [GHz]	$20.241272 + j\,0.161734$	$20.217773 + j\,0.149273$	$20.209273 + j\,0.183226$
$f_r^{(8)}$ [GHz]	$20.349967 + j\,0.111563$	$20.338619 + j\,0.068928$	not found
$f_r^{(9)}$ [GHz]	$20.437490 + j\,0.092513$	$20.433241 + j\,0.085308$	$20.463069 + j\,0.085994$
$f_r^{(10)}$ [GHz]	$20.477526 + j\,0.251887$	$20.417040 + j\,0.215997$	not found
$f_r^{(11)}$ [GHz]	$20.499251 + j\,0.220886$	$20.488543 + j\,0.212953$	$20.498675 + j\,0.220884$
$f_r^{(12)}$ [GHz]	$20.200693 + j\,0.480862$	$20.202756 + j\,0.481418$	not found

poles

	SA-GRPF	**AAA circle**	**AAA rectangular**
$f^{(13)}$ [GHz]	$20.203666 + j\,0.479928$	$20.205040 + j\,0.480290$	not found

For this example, $20\ 000$ points (evenly spaced from each other along the perimeter of the rectangle) are taken for the analysis. In the case of the circular domain, more results are found than in the case of the rectangular one. However, the results are not accurately evaluated. In the case of circular and rectangular domains, the AAA algorithm finds only a few particular roots (including the ”island”) with acceptable accuracy. In addition, as can be seen in Figure 3.35, there are many artificial poles and zeros (especially at the boundary of the domain). The significant impact of the shape of the domain on the results is worrying.

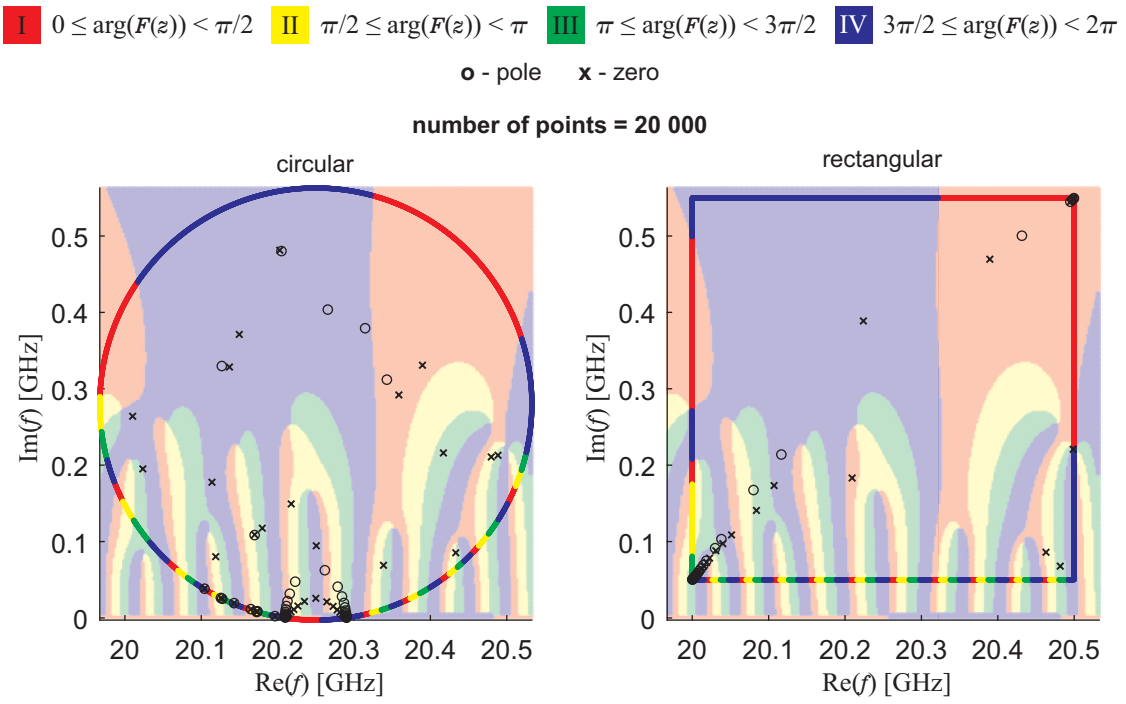

Figure 3.35: The results from the findings of zeros and poles using the AAA algorithm (circular domain on the left and rectangular domain on the right). In the background is a phase portrait obtained from a very dense function sampling.

Graphene transmission line

In the last example, a simple transmission line containing a graphene layer is investigated as presented in subsection 2.3.3. In this problem, the normalized propagation coefficient $z \in \mathbb{C} = \gamma$ is determined. The analyzed rectangle is defined by the following curve:

$$C_r = \left\{ z \in \mathbb{C} : \begin{array}{c} -100 \leq \mathrm{Re(z)} \leq 400 \wedge \mathrm{Im(z)} = -100 \\ -100 \leq \mathrm{Re(z)} \leq 400 \wedge \mathrm{Im(z)} = 400 \\ \mathrm{Re(z)} = -100 \wedge -100 \leq \mathrm{Im(z)} \leq 400 \\ \mathrm{Re(z)} = 400 \wedge -100 \leq \mathrm{Im(z)} \leq 400 \end{array} \right\}$$

and circle is described by the following curve:

$$C_c = \{ z \in \mathbb{C} : |z - 150 - j\,150| = 350 \}$$

The visualization of the results is presented in Figure 3.36, and the summary of the found zeros and poles is in Table 3.13. The $20\,000$ points located evenly spaced from each other along the perimeter of the rectangle and circle are used for both analyses.

Table 3.13: Overview of found roots and singularities for the proposed self-adaptive initial mesh generator for GRPF and results obtained using the AAA method in two configurations.

zeros

z	**SA-GRPF**	**AAA circle**	**AAA rectangular**
$\gamma^{(1)}$	$0.004527 - j \quad 0.955902$	not found	not found
$\gamma^{(2)}$	$-0.004527 + j \quad 0.955902$	not found	not found
$\gamma^{(3)}$	$0.003207 + j \quad 0.964811$	not found	not found
$\gamma^{(4)}$	$-0.003207 - j \quad 0.964810$	not found	not found
$\gamma^{(5)}$	$-32.101962 - j \quad 27.430862$	not found	not found
$\gamma^{(6)}$	$32.101962 + j \quad 27.430862$	not found	not found
$\gamma^{(7)}$	$-38.177725 - j \quad 32.529521$	not found	not found
$\gamma^{(8)}$	$38.177725 + j \quad 32.529521$	not found	not found
$\gamma^{(9)}$	$332.744889 + j\, 282.243080$	not found	not found
$\gamma^{(10)}$	$336.220287 + j\, 285.191091$	not found	not found
$\gamma^{(11)}$	$368.439467 + j\, 312.522078$	not found	not found
$\gamma^{(12)}$	$371.007571 + j\, 314.700408$	not found	not found

poles

	SA-GRPF	**AAA circle**	**AAA rectangular**
$\gamma^{(13)}$	$0.000000 + j\, 1.000000$	not found	not found
$\gamma^{(14)}$	$0.000000 - j\, 1.000000$	not found	not found
$\gamma^{(15)}$	$0.000000 + j\, 3.449638$ (second order)	$0.188613 + j\, 3.465189$ $-0.188169 + j\, 3.466824$	not found
$\gamma^{(16)}$	$0.000000 - j\, 3.449638$ (second order)	$-0.188583 - j\, 3.465199$ $0.188139 - j\, 3.466814$	$0.188119 - j\, 3.466744$ $-0.188555 - j\, 3.465267$

In this case, the AAA algorithm found only some poles with debatable accuracy (at least one in the case of a rectangular domain). The algorithm do not find any of the existing roots, and only the artificial results are obtained (see the phase portrait in Figure 3.36). In order to analyze this case in more detail, it is decided to decompose the domain into smaller subregions. The analyzed rectangle is defined by the following curve:

$$C_r = \left\{ z \in \mathbb{C} : \begin{array}{c} -5 \leq \mathrm{Re(z)} \leq 5 \wedge \mathrm{Im(z)} = -5 \\ -5 \leq \mathrm{Re(z)} \leq 5 \wedge \mathrm{Im(z)} = 5 \\ \mathrm{Re(z)} = -5 \wedge -5 \leq \mathrm{Im(z)} \leq 5 \\ \mathrm{Re(z)} = 5 \wedge -5 \leq \mathrm{Im(z)} \leq 5 \end{array} \right\}$$

and circle is described by the following curve:

$$C_c = \{ z \in \mathbb{C} : |z| = 6.5 \}$$

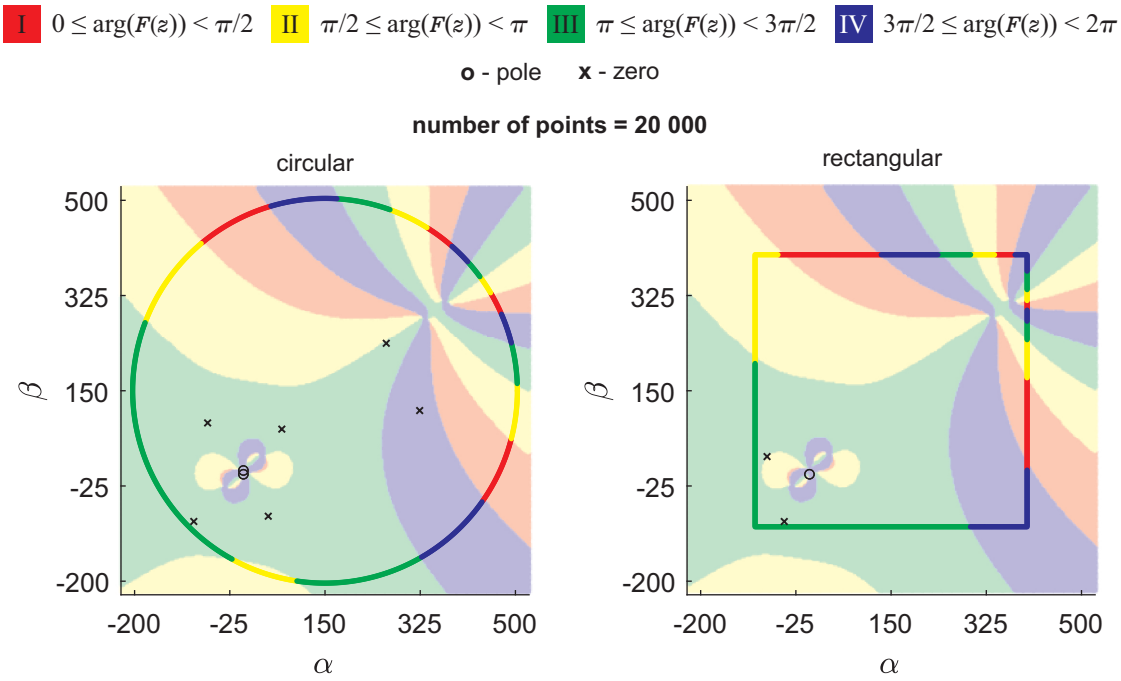

Figure 3.36: The results from the findings of zeros and poles using the AAA algorithm (circular domain on the left and rectangular domain on the right). In the background is a phase portrait obtained from a very dense function sampling.

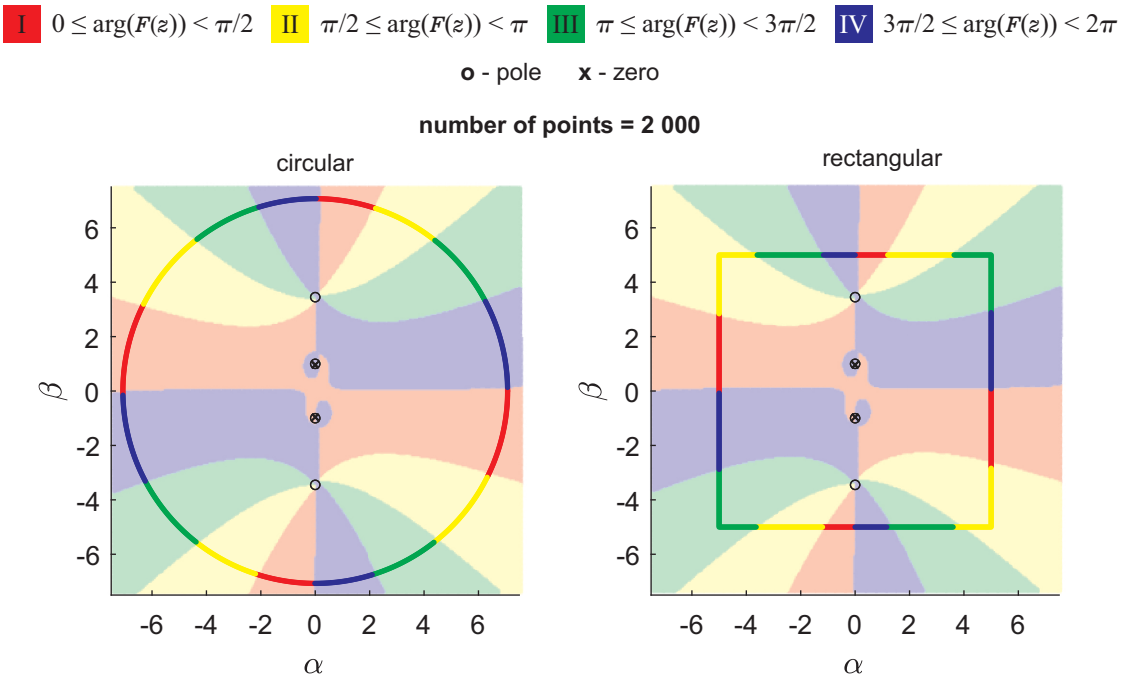

Figure 3.37: The results from the findings of zeros and poles using the AAA algorithm (circular domain on the left and rectangular domain on the right). In the background is a phase portrait obtained from a very dense function sampling.

The first one involves the vicinity of $z = 0$, where four poles and four zeros are located. The visualization of the results is presented in Figure 3.37, and the summary of the found zeros and poles is in Table 3.14. For a smaller domain, ten times fewer points are used.

Table 3.14: Overview of found roots and singularities for the proposed self-adaptive initial mesh generator for GRPF and results obtained using the AAA method in two configurations.

zeros

z	**SA-GRPF**	**AAA circle**	**AAA rectangular**
$\gamma^{(1)}$	$0.004527 - j\,0.955901$	$0.012542 - j\,0.978939$	$0.003490 - j\,0.954127$
$\gamma^{(2)}$	$-0.004527 + j\,0.955902$	$-0.041528 + j\,0.990652$	$-0.002948 + j\,0.949466$
$\gamma^{(3)}$	$-0.003207 - j\,0.964810$	$-0.012860 - j\,0.942240$	$-0.002155 - j\,0.966510$
$\gamma^{(4)}$	$0.003207 + j\,0.964810$	$0.041862 + j\,0.930490$	$0.001614 + j\,0.971167$

poles

	SA-GRPF	**AAA circle**	**AAA rectangular**
$\gamma^{(5)}$	$0.000000 + j\,1.000000$	$0.000538 + j\,0.999760$	$0.000076 + j\,0.999870$
	(second order)	$-0.000525 + j\,1.000252$	$-0.000075 + j\,1.000140$
$\gamma^{(6)}$	$0.000000 - j\,1.000000$	$0.000414 - j\,0.999887$	$0.000040 - j\,0.999905$
	(second order)	$-0.000409 - j\,1.000121$	$-0.000039 - j\,1.000095$
$\gamma^{(7)}$	$0.000000 - j\,3.449638$	$0.000001 - j\,3.449637$	$0.000000 - j\,3.449637$
	(second order)	$-0.000001 - j\,3.449639$	$0.000000 - j\,3.449638$
$\gamma^{(8)}$	$0.000000 + j\,3.449639$	$0.000002 + j\,3.449637$	$0.000000 + j\,3.449637$
	(second order)	$-0.000002 + j\,3.449639$	$0.000000 + j\,3.449638$

Reducing the search region allows one to get correct results; however, their accuracy is still questionable (and depends on the domain shape). In the next step, the focus is on the region containing four zeros. The analyzed rectangle is defined by the following curve:

$$C_r = \left\{ z \in \mathbb{C} : \begin{array}{l} 200 \leq \mathrm{Re(z)} \leq 400 \wedge \mathrm{Im(z)} = 200 \\ 200 \leq \mathrm{Re(z)} \leq 400 \wedge \mathrm{Im(z)} = 400 \\ \mathrm{Re(z)} = 200 \wedge 200 \leq \mathrm{Im(z)} \leq 400 \\ \mathrm{Re(z)} = 400 \wedge 200 \leq \mathrm{Im(z)} \leq 400 \end{array} \right\}$$

and circle is described by the following curve:

$$C_c = \{z \in \mathbb{C} : |z - 300 - j\,300| = 140\}$$

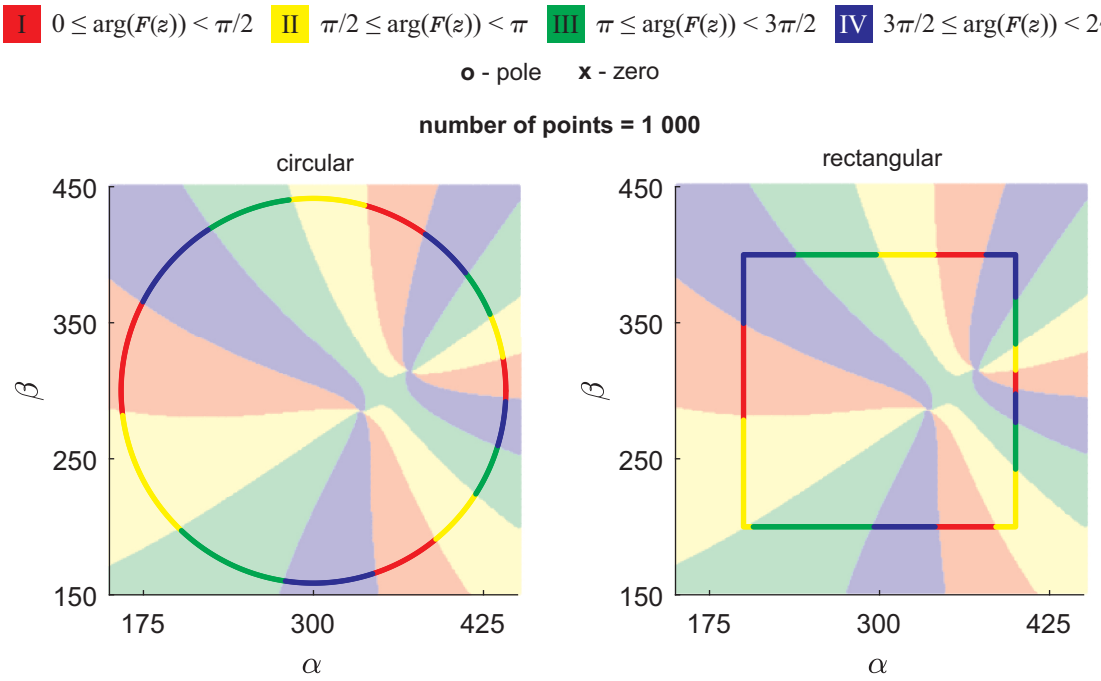

Figure 3.38: The results from the findings of zeros and poles using the AAA algorithm (circular domain on the left and rectangular domain on the right). In the background is a phase portrait obtained from a very dense function sampling.

Table 3.15: Overview of found roots and singularities for the proposed self-adaptive initial mesh generator for GRPF and results obtained using the AAA method in two configurations.

zeros

z	SA-GRPF	AAA circle	AAA rectangular
$\gamma^{(1)}$	$332.744889 + j\ 282.243080$	not found	not found
$\gamma^{(2)}$	$3.362203 + j\ 285.191091$	not found	not found
$\gamma^{(3)}$	$368.439467 + j\ 312.522078$	not found	not found
$\gamma^{(4)}$	$3.710076 + j\ \ 3.147004$	not found	not found

The visualization of the results is presented in Figure 3.38, and the summary of the found zeros and poles is in Table 3.15. For this example, $1\ 000$ points are taken for the analysis. The AAA algorithm could not calculate any zeros used for both circular and rectangular shape domains (even if a hundred times more points were used).

In summary, there are circumstances in which the AAA algorithm is very efficient, especially if the function is rational. In general, however, the results of the AAA algorithm are highly influenced by the domain shape and size choice. The presence of artificial zeros and the omissions of some results may raise some doubts about the algorithm's reliability. The new self-adaptive GRPF approach is free of these disadvantages, as it assumes analysis for any function and domain at which the number of starting points will automatically be chosen.

3.7 Final comments

The above chapter presented the idea of the GPRF algorithm and its improvements on various levels, i.e., initial mesh, mesh refinement, or control of the stability of root finding (”skinny” triangle issue). It is shown that by using appropriate mesh densification, it is possible to reduce the number of nodes and, thus, the total number of function calls. Each reduction in the number of function calls directly impacts the algorithm’s performance and efficiency, especially for more complex functions (e.g., numerically complicated).

The main achievement is the proposed self-adaptive mesh generator for the GRPF algorithm. As can be seen from the presented numerical examples, the proposed technique has low numerical complexity. The benefit of using an adaptive mesh is insensible in straightforward models and problems where the zeros and poles are well separated. However, the algorithm’s performance can increase by several orders of magnitude in the case of zeros and poles located close together. It was proved that the benefits of using the proposed approach can be huge with near-located zero-pole pairs. In a situation where the roots are separated and evenly distributed within the domain, using an adaptive mesh is not necessary, and its effectiveness is similar to using a uniform mesh. Nevertheless, it is challenging to determine priori the distribution of zeros and poles for any function, so regardless of the variant, it will be better to use the presented adaptive algorithm.

In summary, the self-adaptive initial mesh generator for GRPF allows faster and more accurate finding of zero and poles of complex functions. An excellent instance of this is the last of the numerical examples - the graphene transmission line. The proposed technique reduced the number of function evaluations by about $200\,000$ times. It allows the user to perform a faster, more accurate, in-depth analysis of the investigated electromagnetics problem (not only but every root finding issue). The above results and the supporting conference proceeding [119] and journal publications [116] are proof of the first thesis of this dissertation: ”The proper generation of the initial mesh and the optimal process of its densification in the GRPF algorithm [104] will improve the efficiency of the procedure and increase the probability of finding all roots in a fixed search region”.

Chapter 4

Self-adaptive complex root tracing algorithm

In this chapter, the technique for tracing the complex roots $z \in \mathbb{C}$ as a function of the additional parameter $t \in \mathbb{R}$, presented in [61], will be introduced. This powerful method has successfully addressed electrodynamics, various engineering fields, and mathematical problems. Furthermore, within the scope of this work, an optimization of this algorithm is proposed based on the self-adaptive discretization of the CAP condition, also supported by the publication [117]. At the end of this section of the thesis, a numerical experiment will be presented, confirming the reduced number of function calls during analyses without affecting the achieved accuracy of the results. In order to better understand the essence of the modifications, first, the features of the algorithm in the basic version will be discussed.

4.1 Formulation of the problem

The root tracing technique presented in [94] was originally proposed for real functions of two variables. As it is well known for continuous real functions, changing the sign of the function value is a necessary and sufficient condition for the existence of a root. Hence, the idea of tracing is to iteratively check the changes in the sign of the function value at new systematically evaluated points.

4.1.1 Root tracing of a real function of a two variables

Let us discuss an example of a two-dimensional tracing algorithm for a certain real function, $F(x, y)$, in the $\Omega = \{x \in \mathbb{R}, y \in \mathbb{R} : x \geq 0 \wedge y \geq 0\}$ region (see Figure 4.1). In order to start the tracing procedure, it is necessary to determine the starting point, which is the zero of the analyzed function $F(x_0, y_0) = 0$. For the example shown in Figure 4.1, the tracing process starts from the line segment on the y-axis, i.e., two points near (x_0, y_0) where the value function changes its sign. In the first step, a new point is added at the vertex of the equilateral triangle. The value of the function is evaluated there, and its sign is determined. Then, the difference in the sign of the function between vertices is compared. One of the sides of the triangle becomes the new starting

element for the next step. This procedure is repeated until the edge of the Ω domain is reached or until there is no zero of the function in a given region.

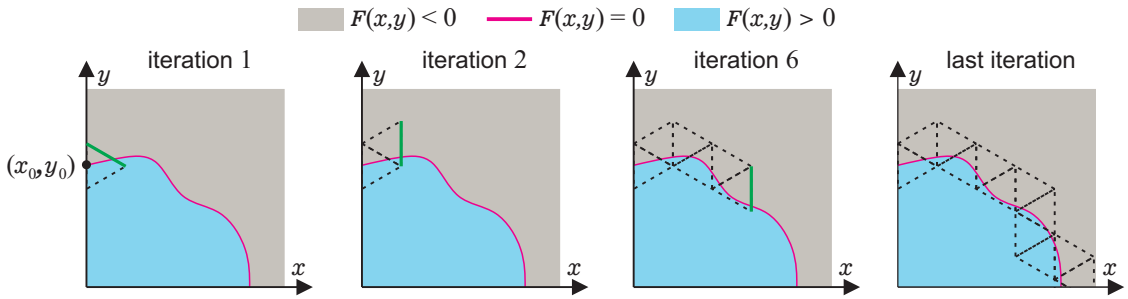

Figure 4.1: Some example steps of the two-dimensional tracing procedure of a function of a real variables.

An effective technique of tracing the root of real functions became the ground for developing global algorithms for finding the zeros and poles of complex functions [55], [60], [95]. As mentioned earlier, very often in practical electromagnetic applications (but also in other fields), the object of study is to change the place of zero or pole on the complex plane as a function of an additional variable. The problem then boils down to the three-dimensional space $\mathbb{C} \times \mathbb{R}$, in which the method based on changing the sign of the real function turns out to be ineffective, as proven in [61]. It was shown that different signs of the real and imaginary parts of the function are a necessary but not sufficient condition for the existence of roots or poles within the considered region. Therefore, in the complex domain, verification of the existence of a zero in a fixed region is a much more challenging task.

4.1.2 Generalization of the Cauchy's argument principle to $\mathbb{C} \times \mathbb{R}$ space

As discussed in earlier chapters, the approach based on Cauchy's Argument Principle is widely used in global methods for finding complex roots and poles (refer to subsection 3.1.2). In the article [61], a necessary modification of CAP-based methods (such as [60], [95]) is proposed, which enables the correct application of the tracing algorithm in the complex domain. In order to verify the existence of zero or singularity of the function in an assumed domain, the discretized version of this principle is used (see subsection 3.2.2). Although DCAP was originally defined on the complex plane, it turned out that its simple generalization can be derived from the $\mathbb{C} \times \mathbb{R}$ space, as shown in Figure 4.2.

Let $F(z,t)$ be a continuous holomorphic function in the $\Omega = \{z \in \mathbb{C}, t \in \mathbb{R}\}$ region. The root of this function may depend on the parameter t if z changes as t changes, such that:

$$F(\xi(t), t) = 0 \tag{4.1}$$

where:

$\xi(t)$ - curve in the analyzed space $\mathbb{C} \times \mathbb{R}$.

In such a case, if a $C \subset \Omega$ curve of arbitrary shape surrounds $\xi(t)$ - not necessarily being in the $t = \text{const}$ plane, then CAP (as well as DCAP) can be applied. The principles are identical to those of the two-dimensional complex domain. Thus, the formulas presented in subsection 3.1.2 and their interpretation are valid. Consequently, the value of q (see equation (3.3)) represents the sum of all zeros with their multiplicities minus the sum of all poles with their multiplicities inside the curve C. According to the discrete version of CAP, q can be estimated by sampling the function and changing the quadrant Qn (consistent with changing the function's argument) around the curve C, as illustrated in Figure 4.2.

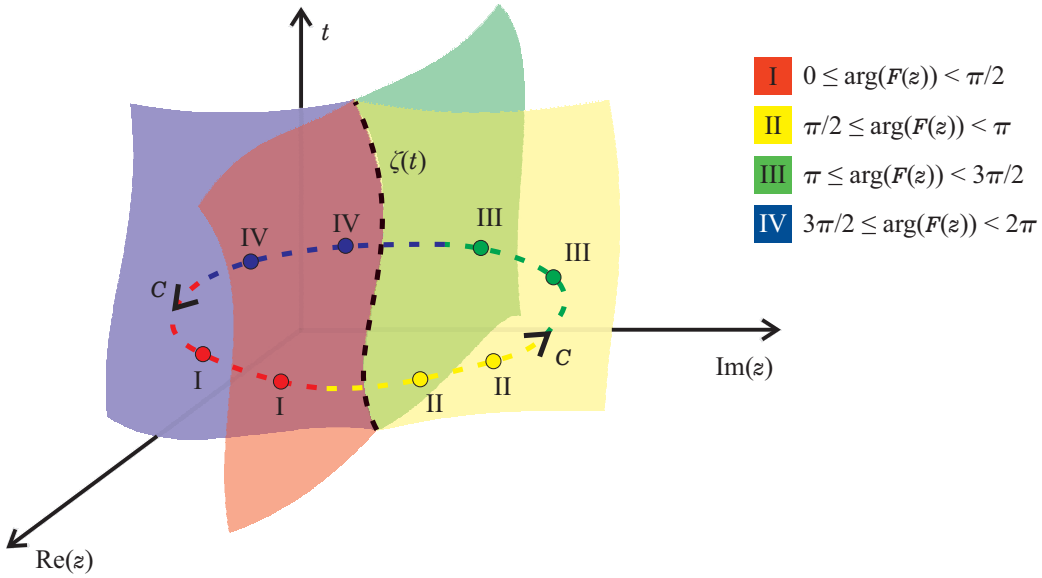

Figure 4.2: The $\xi(t)$ curve in the $\mathbb{C} \times \mathbb{R}$ space and the curve C confirm the existence of a root inside it.

4.1.3 Chain of regular tetrahedrons

The tracing algorithm aims to build a chain of regular tetrahedra (three-dimensional $\mathbb{C} \times \mathbb{R}$ space). This chain will contain a curve $\xi(t)$ representing the change in the place of the root as a function of the parameter t. The solution will consist of a set of zeros or poles $z_n(t_n)$ located at the center of each internal face of the chain. In order to begin the tracing procedure, it is necessary to determine the starting point (z_0, t_0), which is the root of a given function in the complex plane for a fixed value of the initial parameter $F(\mathrm{Re}(z_0), \mathrm{Im}(z_0), t_0)) = 0$, as shown in Figure 4.3. This value can be obtained through various means, such as prior knowledge of the function, experimental or numerical experimentation, or using global algorithms to determine the zeros of complex functions. The tracing algorithm can be broken down into the following steps:

1. To begin, an equilateral triangle ABC is constructed with its centroid located at the starting point ((z_0, t_0)). The length of the triangle's side is equal to the r, the parameter that determines the resolution (accuracy) of the algorithm. At each of the vertex of $\mathrm{A}, \mathrm{B}, \mathrm{C}$, the value of the function is evaluated, and its quadrant is determined, as shown in Figure 4.4. According to DCAP, to confirm the existence of a root inside any closed curve, it must contain at

least four points representing the value of the function being in four different quadrants of the coordinate system, and they must be set in the correct order.

2. In the next step, new points are added at the midpoint of the edges of the triangle. A standard bisection approach is applied until the DCAP condition (3.7) is met, i.e., $q = 1$ for zero, $q = -1$ for singularity. Fulfillment of this condition confirms the existence of a root or pole inside the triangle and allows the move on to the next step.

3. A new point D is added in this stage to form a regular tetrahedron (see Figure 4.4). This way, triangles are created: ABD, BCD, and CAD enclosing the $\xi(t)$ curve inside. Assuming its continuity, if $\xi(t)$ intersects the surface of triangle ABC, it must consequently intersect one of the other faces of the tetrahedron. Determining this face requires, once again, the use of DCAP. It is therefore necessary to determine the value of the function at vertex D and at new points added on segments AD, BD, and CD of the tetrahedron until the condition (3.7) is met on any of the faces ABD, BCD, or CAD. This procedure step is crucial to the algorithm's performance, especially if evaluating a single function value is time-consuming.

4. Once the proper face containing the root (or pole) is identified (ACD for the example in Figure 4.4), it becomes the bottom face of the new tetrahedron. This process is repeated, starting from the face where the zero is located (intersecting with the $\xi(t)$ curve). Once again, this can be achieved by systematically increasing the density of points and adding new points in the middle of current segments.

5. After several iterations, one gets a set of interconnected tetrahedrons representing the $\xi(t)$ curve. The process of tracing is repeated until no point has a coordinate t greater than t_{max}. This procedure will result in a set of points (z_n, t_n) consisting of the centroids of all the inner faces of the chain through which $\xi(t)$ passes.

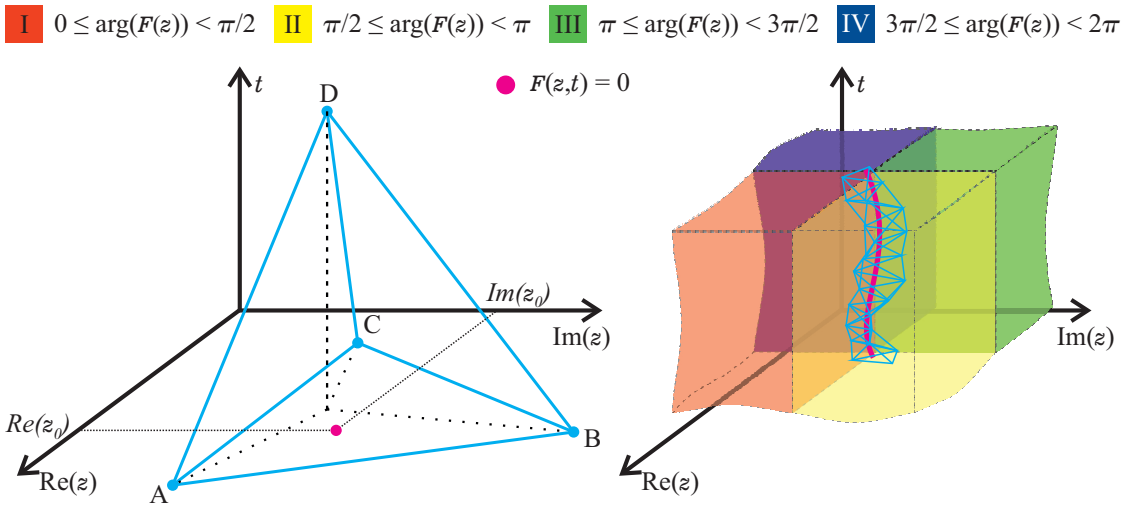

Figure 4.3: On the left, a single regular tetrahedron ABCD in $\mathbb{C} \times \mathbb{R}$ space with the root at point z_0; on the right, a chain of tetrahedra representing the investigated curve $\xi(t)$ and a phase portrait in the background.

The total number of new points (function calls) cannot be determined a priori. Nevertheless, it depends on the length of the edge r and the complexity of the function. It also depends on the arrangement of the initial triangle, which will be discussed in more detail in subsection 4.2.3.

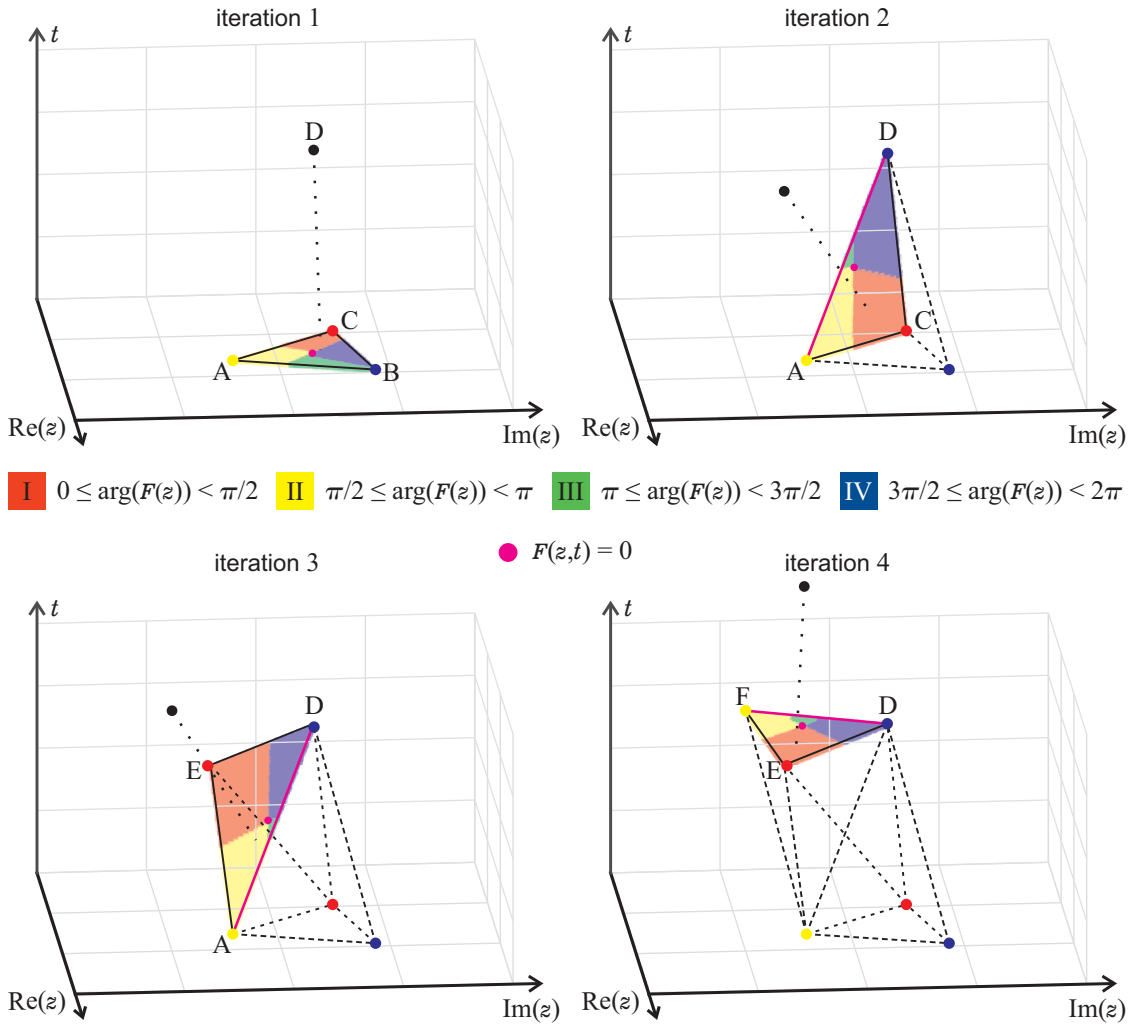

Figure 4.4: The forming of a chain of regular tetrahedrons in the first four example iterations of the tracing algorithm; the phase portraits illustrated on individual faces.

4.2 The improvement of the algorithm

The Self-Adaptive Complex Root Tracing Algorithm [117] (SACTRA for short) is a proposed improvement to the tracing algorithm based on the technique presented in the previous section. As mentioned above, the most important fragment of the tracing technique in terms of its efficiency is the number of function calls needed to verify which face the curve $\xi(t)$ passes through. In the regular tracing algorithm [61], the DCAP condition is analyzed in each triangle (in each face of the tetrahedron) see equation (3.7). A discrete change in the function phase along each side containing the z_p nodes is investigated. Generally, the phase value itself is unimportant, but the $Q_n = \{\mathrm{I}, \mathrm{II}, \mathrm{III}, \mathrm{IV}\}$, i.e., quadrant in the coordinate system in which the function value is. Suppose

the discretization considers all changes of individual quadrants (by one, e.g., from I to II , from II to III, from III to IV, from IV to I) along the circumference of the triangle. In that case, the DCAP condition is met. As mentioned earlier, verifying a single root requires at least four points. Figure 4.5 shows two types of discretization (the base approach and the proposed modification [117]), which confirm the existence of the root inside the triangle. In the regular tracing algorithm, new points are added to the center of each triangle segment. However, this approach requires evaluating the function value at a minimum of six points, which is redundant. A more efficient approach is proposed to add a point only in the section with a double change of quadrants at terminal nodes.

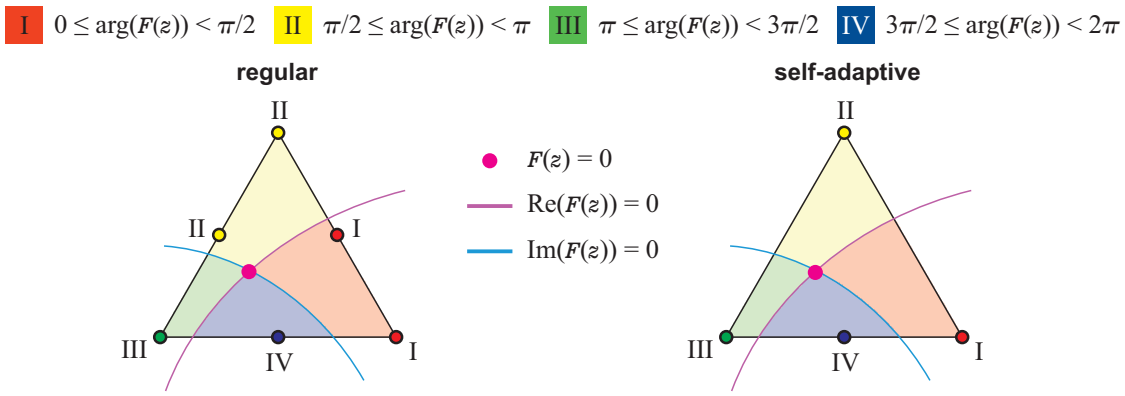

Figure 4.5: Two types of discretizations verify the existance of the root inside the triangle.

The example shown in Figure 4.5 is trivial, and the improvement in algorithm efficiency by reducing the number of function calls is only 6 to 4 times. In practice, the angle of intersection of the curves representing the zeros of the real and imaginary parts of the function with the triangle side could be slight, or the root could be close to the edge of the triangle. Therefore, it may be necessary to add multiple points along the sides of the triangle, requiring an iterative approach. New points are added until the existence of a root is confirmed on any face of the tetrahedron. Each of the four quadrants must occur in the correct order without changing the quadrant double (i.e., $|\Delta Q| = 2$) along a single segment. In Figure 4.6, a comparison of iterative procedures in regular tracing and the new algorithm using self-adaptive discretization is presented for an example function. It clearly shows that the algorithm's performance increases significantly with the number of iterations, respectively: $(3 \cdot 2^n)$ to $(n+3)$ times, where n is the number of iterations. A summary of the algorithm of the self-adaptive approach for the complex root tracing procedure is presented in the form of a block diagram in Figure 4.7. The source code of the Self-Adaptive Complex Root Tracing Algorithm can be found at: https://github.com/PioKow/SACRTA.

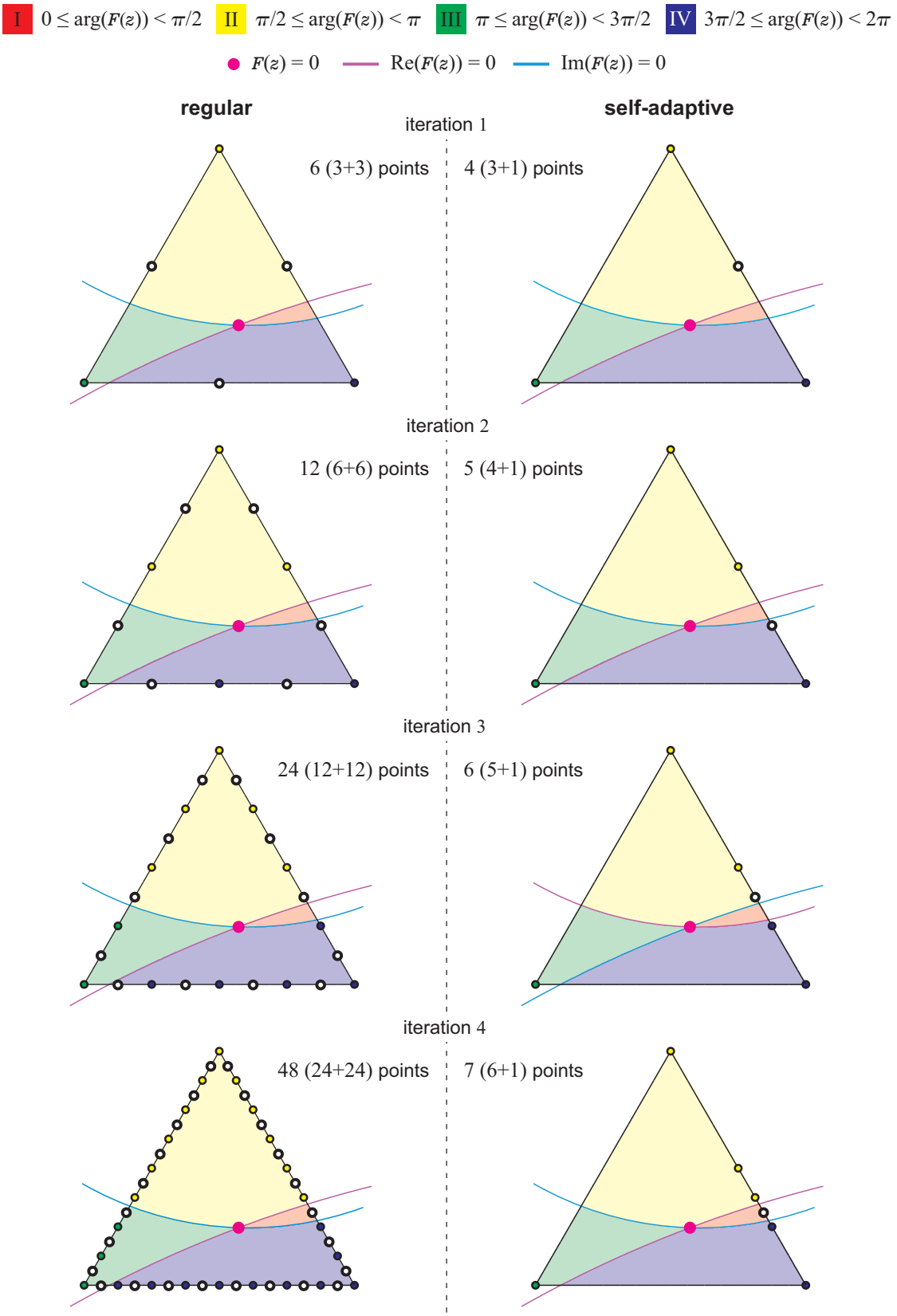

Figure 4.6: The example of four successive iterations of verifying a single root within a triangle. The tracing algorithm is displayed on the left, while the proposed algorithm utilizing self-adaptive discretization is shown on the right. The total number of points at each stage is depicted, reflecting the sum of both current and added nodes (in brackets). Empty circles with black borders indicate the new points added in each iteration.

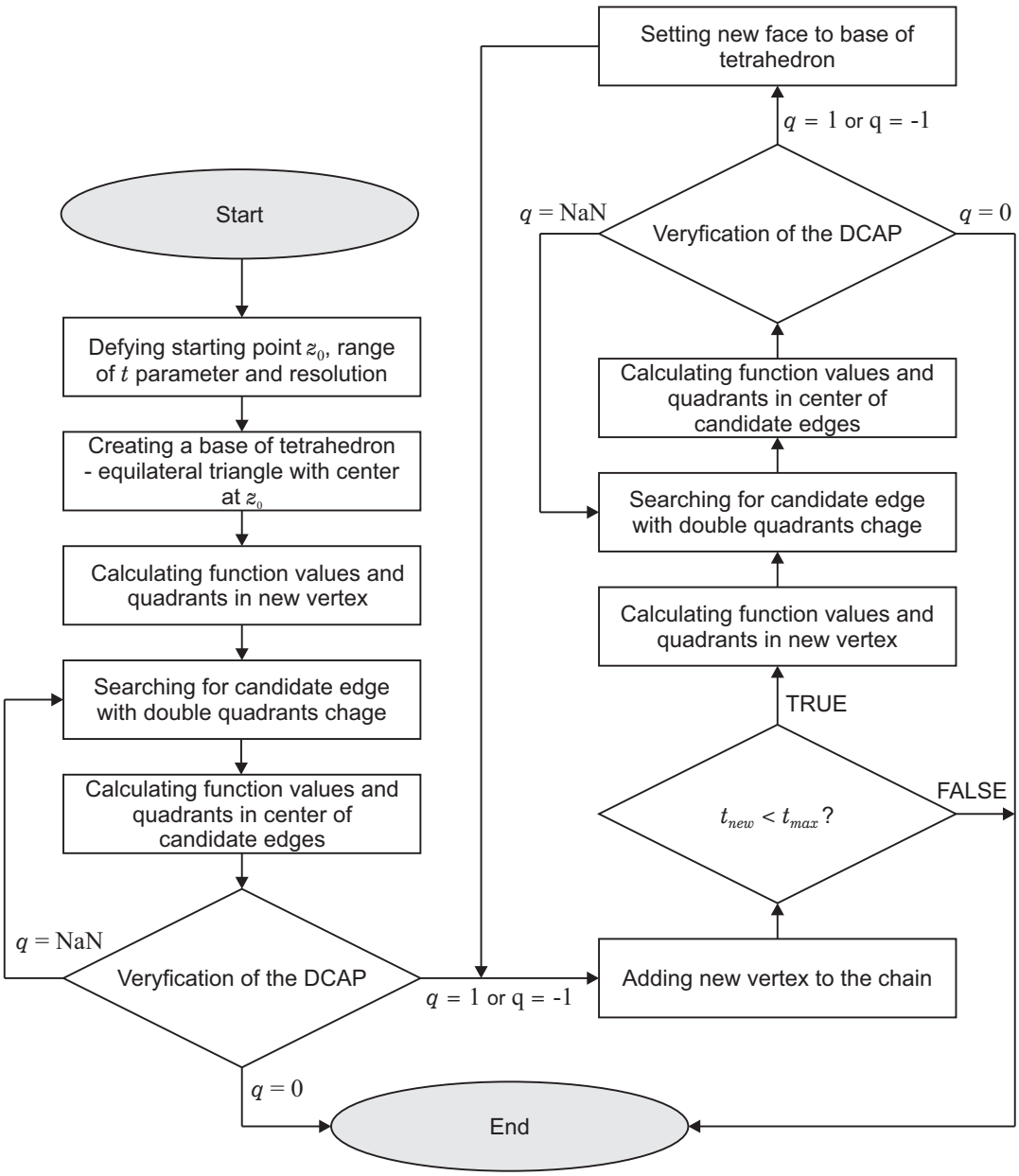

Figure 4.7: Block diagram of operating the proposed Self-Adaptive Complex Root Tracing Algorithm.

4.2.1 Numerical examples

In order to validate the effectiveness of the new, self-adaptive approach for root tracing, the examples presented in the regular algorithm [61] are calculated. The accuracy of the results, the number of function calls, and the analysis time are compared between the regular tracing algorithm and the modified version. In each example, the starting point z_0 for the initial value of the additional parameter t_0 is determined using the regular GRPF [104]. The examples shown here account for the structures presented in subsection 2.3 and are the graphene transmission line, the coaxially loaded cylindrical waveguide, and the cylindrical-rectangular microstrip resonant structure.

Graphene transmission line

In the first example in this section, an analysis of the propagation coefficient $\gamma(f)$ as a function of frequency in a graphene transmission line (shown in 2.3.3) is carried out. The tracing is performed in the $f \in \mathbb{R} : 1 \leq f \leq 7\,\mathrm{THz}$ range. The procedure is run for two TM mods. At the initial frequency $f_0 = 1\ \mathrm{THz}$, the propagation coefficients are $\gamma_{01}(f_0) = 336 + j\ 285$ and $\gamma_{02}(f_0) = 32 + j\ 27$, respectively. The resolution of the algorithm is set to $r = 1$ (the length of the side of the equilateral tetrahedron). Fitting the domain to the investigated problem increases the algorithm's efficiency, so frequency normalization to $100\ \mathrm{GHz}$ is assumed. The resulting curve, $\xi(t)$, is identical for both approaches and is shown in Figure 4.8. Table 4.1 compares the number of function calls and analysis times between the approaches.

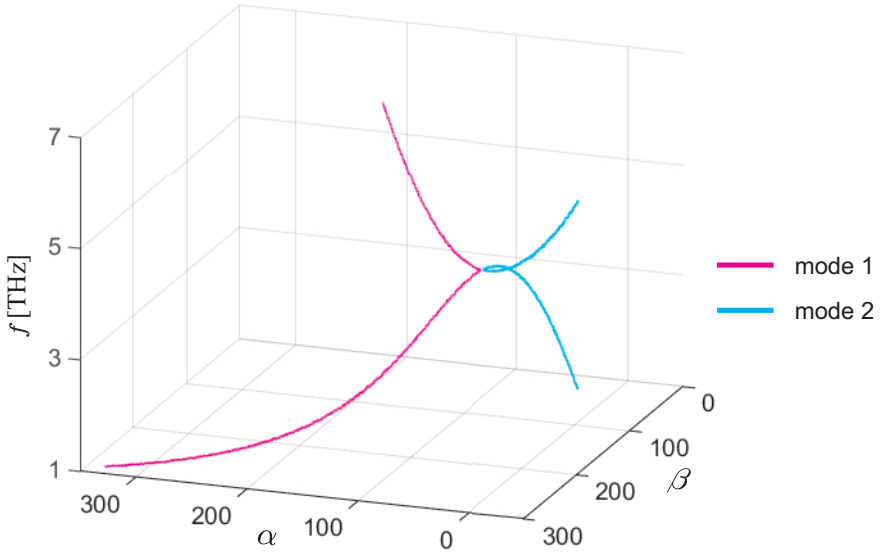

Figure 4.8: The obtained dispersion characteristics of two TM modes in the assumed frequency range (from both approaches, results are identical).

Table 4.1: The obtained numerical effectiveness for the proposed self-adaptive approach, compared with regular tracing (results in brackets).

mode	**no. of function calls**	**time** [s]
1	7 342 (269 767)	0.51 (2.82)
2	3 780 (102 670)	0.39 (1.51)
1+2	11 122 (372 437)	0.90 (4.33)
speed-up ratio	33.49	4.83

This example achieved an impressive reduction in function calls by over 33 times, resulting in a speed-up analyzing time of almost five times. The difference between these values is due to the short time of a single function evaluation because the line is modeled analytically.

Coaxially loaded cylindrical waveguide

The second example pertains to the coaxial cylindrical waveguide loaded with a dielectric. Subsection 2.3.2 provides the parameters of this structure. As with the previous example, the propagation coefficient of $\gamma(f)$ is traced as a function of frequency in the range of $f \in \mathbb{R} : 3 \leq f \leq 7$ GHz. Again, two separate mods, EH_{11} and HE_{11}, are considered with propagation coefficients $\gamma_{0\mathrm{HE}_{11}}(f_0) = -2.93$ and $\gamma_{0\mathrm{EH}_{11}}(f_0) = 3.68$, respectively, at $f_0 = 3$ GHz. Also, in this case, the resolution of the algorithm is set to $r = 0.01$ with frequency normalization to 1 GHz. Figure 4.9 presents the obtained characteristics, which are identical for both approaches. Table 4.2 compares the number of function calls and analysis time between the approaches.

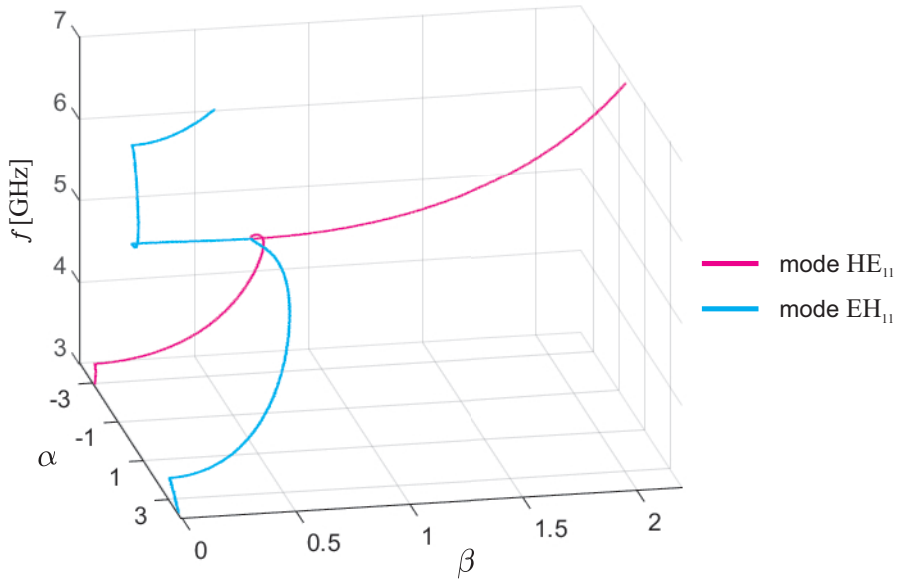

Figure 4.9: The obtained dispersion characteristics of EH_{11} and HE_{11} modes in the assumed frequency range (from both approaches, results are identical).

Table 4.2: The obtained numerical effectiveness for the proposed self-adaptive approach, compared with regular tracing (results in brackets).

mode	**no. of function calls**	**time** [s]
HE_{11}	5 381 (41 419)	0.85 (4.07)
EH_{11}	7 115 (135 714)	1.11 (10.98)
HE_{11}+EH_{11}	12 496 (177 133)	1.96 (15.05)
speed-up ratio	14.18	7.66

In this example, the reduction in function calls is about 14 times, a speed-up time of more than seven times. Again, the time for a single function evaluation is short because the analytical method is used to solve the electromagnetic wave propagation equation in such a structure.

Cylindrical-rectangular microstrip resonant structure

The last example involves a cylindrical-rectangular microstrip resonant structure (see subsection 2.3.6). The tracing concerned the change of the complex resonance frequency $f_r(s)$ as a function of one of the dimensions of the structure, $s \in \mathbb{R}$, in the range $0 \leq s \leq 10 \ \ \mathrm{mm}$. For the starting point with an air gap thickness of $s_0 = 0\,\mathrm{mm}$ the resonant frequency is $f_{r0}(s_0) = 1.18 - j\,0.01\,\mathrm{GHz}$. The resolution of the algorithm is set to $r = 0.01$ (without domain normalization). The obtained characteristics are equivalent for both approaches and are presented in Figure 4.10. The comparison of the number of function calls and analysis time is shown in Table 4.3.

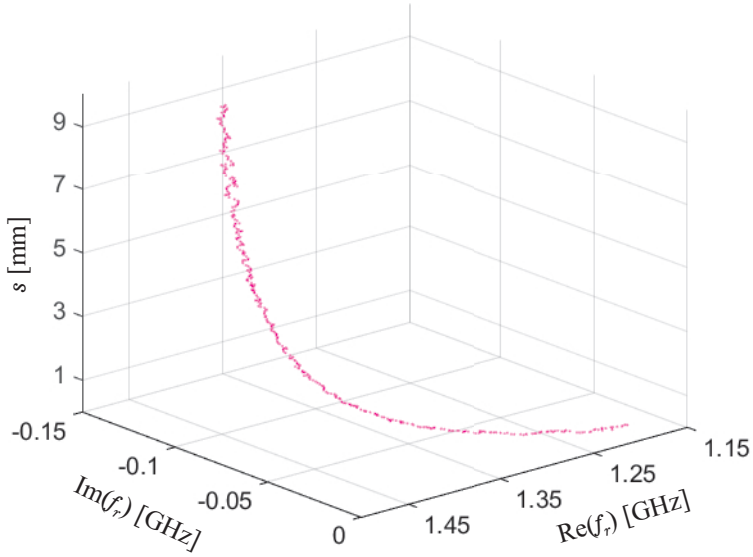

Figure 4.10: The obtained resonance frequency characteristics, as a function of the air gap thickness (from both approaches, results are identical).

Table 4.3: The obtained numerical effectiveness for the proposed self-adaptive approach, compared with regular tracing (results in brackets).

	no. of function calls	**time** [s]
	915 (6 792)	1 470 (10 045)
speed-up ratio	7.42	8.21

Opposite to the previous examples, evaluating a single function takes much longer because the problem requires a numerical analysis. The reduction in function calls is approximately seven times, which resulted in an eight-fold speed-up of time. As can be seen in the presented examples, the longer the time of a single function call, the greater the acceleration in terms of analysis time in relation to the reduction of function calls.

4.2.2 Effectiveness and limitations of the proposed algorithm

The previous section presents a modified and efficient complex root tracing algorithm using self-adaptive CAP discretization. The proposed modification features significantly fewer function calls than the regular tracing algorithm. The result is faster analysis without affecting accuracy, which can be utilized in design and optimization processes. Nevertheless, the tracing algorithm still has its deficiencies. They are exact in both approaches (regular and self-adaptive). When considering efficiency and limitations, the following few issues can be highlighted:

- Before starting the tracing process, it is problematic to estimate the number of function calls. By analyzing the given examples, on average, there are about three function calls per tetrahedron. Nonetheless, the number of tetrahedrons in the chain strongly depends on the nature of the curve $\xi(t)$ being analyzed, making it difficult to provide general conditions. The examples presented ranged from 6 to 40 tetrahedrons per unit $\Delta t/r$, where Δt is the size of the domain relative to the extra parameter t.

- The appropriate setting of the r, i.e., the length of the edges, impacts the algorithm's efficiency. A constant value of this parameter is assumed during one analysis. If it were variable while building the tetrahedral chain, it would get a chain of irregular tetrahedrons. In the case of irregular tetrahedrons, it is difficult to control the chain stability, which may become ill-conditioned, e.g., the side face may consist of "skinny" elements. Furthermore, there is no evidence indicating a positive impact on the efficiency of the tracing algorithm or the accuracy of the results obtained when the length of the edges is modified during the analysis.

- The algorithm exhibits optimal efficiency when the traced characteristic follows a singular path, $\xi(t)$. However, the potential trouble occurs when the characteristic bifurcates or when the curve representing the traced root is very close to the different curve. When another zero (or singularity) is in the area surrounded by the tetrahedron, the algorithm should take this possible path of characteristics into account.

Upon careful examination of the statement above, it can be deduced that the final point is of utmost importance. To address this concern, there exist multiple solutions. One possible approach involves initiating the tracing process from various initial points. By doing so, individual chains of tetrahedrons can be obtained. Further elaboration on the topic of multipath root tracing can be found in the subsequent subsection and the conference proceeding [120] authored by the writer of this thesis.

4.2.3 Multipath complex root tracing

In the presented tracing algorithm (both regular and self-adaptive), new points are added iteratively until a single root or pole is detected. The procedure then automatically moves on to the next tetrahedron. In general, multiple zeros or poles can occur on the same or separate faces of

a tetrahedron. The situation is relatively not complicated when, for a branching point, two or more roots are found on different tetrahedral faces. The algorithm can remember this critical place. It continues to trace a single root and may return to the second root after receiving the first characteristic. Unfortunately, this strategy is impossible when multiple roots appear in one tetrahedron due to the defined termination condition of analysis. After finding the first root on a given face, the algorithm moves to the next face. Searching for further roots could result in the procedure never leaving the first tetrahedron, and the single face would be refined to infinity. Therefore, only a single root value existing in the one face is guaranteed. Searching for additional roots would require adding an infinite number of points due to the lack of prior knowledge of their existence (no termination condition for CAP verification). This problem is a different form of the "islands" described in subsection 3.3 and [116], that is, zero-pole pairs located close to each other. It is challenging to detect the diffraction of isophase in a small area. In addition, it is yet to be known whether multipath characteristics are somewhere. Generally, there are no grounds to assume or reject such circumstances. In critical situations, any global root finding algorithm, e.g., GRPF, can be used to verify the plane $t = \text{const}$ where the branch is possible. However, regardless of the multipath problem, a reduced resolution r can also be used for the entire chain.

In the approach presented above, the initial point t_0 is surrounded by an initial triangle (on the complex plane), which is the bottom face of the initial tetrahedron. The orientation of this triangle on the complex plane is fixed so that for $t = t_0$, one of the edges is parallel to the real axis. Then, three additional triangles are built to construct the initial tetrahedron (see Figure 4.11). In each successive step, a new root value is determined on each new face of the tetrahedron. The face on which the new root is found becomes the next bottom triangle, and three new triangles are built to form a new tetrahedron. The procedure is repeated until the tetrahedral chain is obtained within the assumed range of the additional parameter, forming the characteristic $\xi(t)$ from the centroid points of faces.

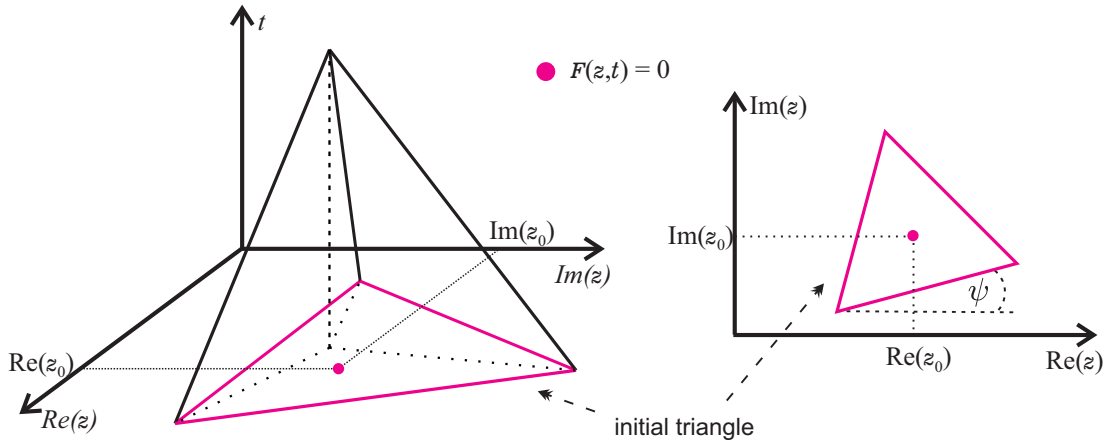

Figure 4.11: On the left is a regular tetrahedron in $\mathbb{C} \times \mathbb{R}$ space; on the right is the orientation of the initial triangle in the complex plane, defined by angle ψ.

The shape of the result characteristic $\xi(t)$ obtained in the tracing procedure may depend on specific algorithm parameters. When tracing a single curve ($\xi(t)$ has no branches), the initial orientation of the triangle does not influence the form of the obtained characteristics. Only the configuration of the tetrahedrons inside the chain may be slightly different. Therefore, it is recommended to set the triangle with one edge parallel to the real axis (as applied in the examples in the previous subsection). However, it turns out that this orientation can be crucial when bifurcations appear on the traced path. In such a case, the algorithm may assume a different path after going through a critical point. This is mainly because other nodes from the close vicinity of the roots will be analyzed. Therefore, they may differ from each other in quadrants, resulting from a change in the phase of the function in the $\mathbb{C} \times \mathbb{R}$ space. Hence, it is proposed to introduce an extra parameter to the algorithm in the form of the initial angle ψ (see Figure 4.11). Its appropriate selection enables obtaining each possible solution. This parameter defines the rotation of the initial triangle relative to the axis OX (real) within the range from 0 to 120 degrees.

To further support the earlier discussion, an example involving the structure of the coaxial cylindrical waveguide loaded with a dielectric is subjected to additional analysis. The analysis is carried out over a slightly more sweeping frequency range, from $f_{min} = 1\ \mathrm{GHz}$ to $f_{max} = 7\ \mathrm{GHz}$. The starting point for the tracing routine is also modified compared to the one investigated in the previous subsection. In this example, the tracing is applied in the reversed direction, i.e., with a decreasing t parameter. The initial point (determined employing the regular GRPF) corresponded to the normalized propagation coefficient $\gamma(f_0) = 2.172$ at the frequency $f_0 = 7\ \mathrm{GHz}$.

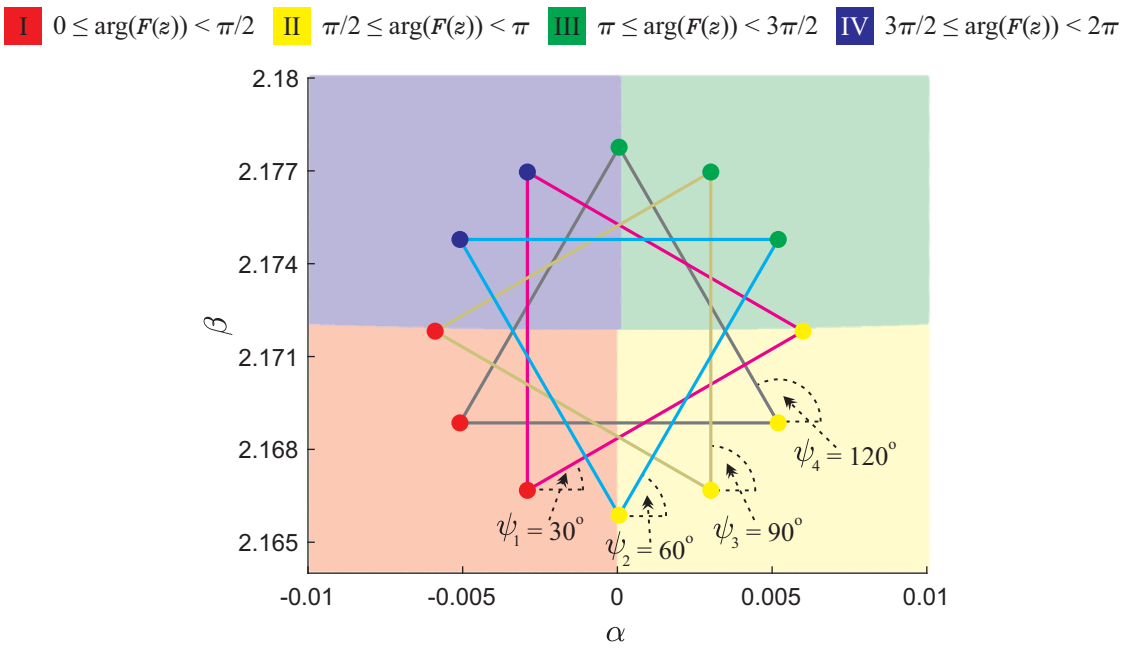

Figure 4.12: The examples of initial triangle configurations in the $t = \mathrm{const}$ plane, equal $f = 7\ \mathrm{GHz}$, are used to analyze multipath problems in the coaxial cylindrical waveguide example. The phase portrait of the function is displayed in the background, which shows that for different values of the angle ψ, the vertices of the triangle are located in the different quadrants of the coordinate system.

The tracing procedure is carried out with the resolution of the algorithm set to $r = 0.01$ and frequency normalization to 1 GHz. By changing the value of ψ, that is, the orientation of the initial triangle (see Figure 4.12), one obtains four separate characteristics, $\xi(t)$, corresponding to different modes, as shown in Figure 4.14. Figure 4.13 displays the projections of the final characteristic generated by executing four separate tracing procedures.

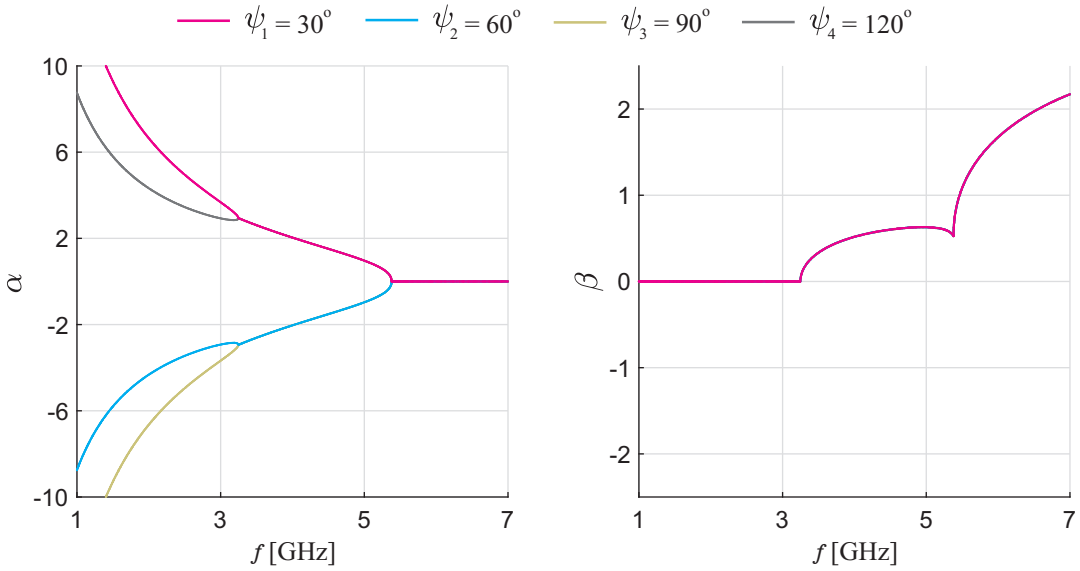

Figure 4.13: The normalized propagation coefficient is shown in the function of frequency. In the analyzed example, the single modes are obtained by setting initial conditions for the triangle (ψ parameter). On the left is the projection on the real axis (attenuation constant), and on the right is the imaginary axis (phase constant).

To sum up, the proposed modification of the initial condition for the starting triangle expands the possibilities of the algorithm. However, determining all possible paths requires multiple analyses for different values of the ψ angle. The number of such algorithm calls depends on the complexity of the path and the scope of the analysis. The above tests indicate that the bisection strategy can help set a range of angles ψ: {0°, 60°, 120°}, {30°, 90°}, {15°, 45°, 75°, 105°}, and so on. Regardless, the results obtained may strongly depend on the lucky alignment of the chain to the location of the bifurcations. In order to indicate the more possible paths, the additional analysis should be carried out with any global complex root finding algorithm (such as GRPF) performed for several planes (for fixed values of $t = \text{const}$). The above results and the supporting conference proceeding [120].

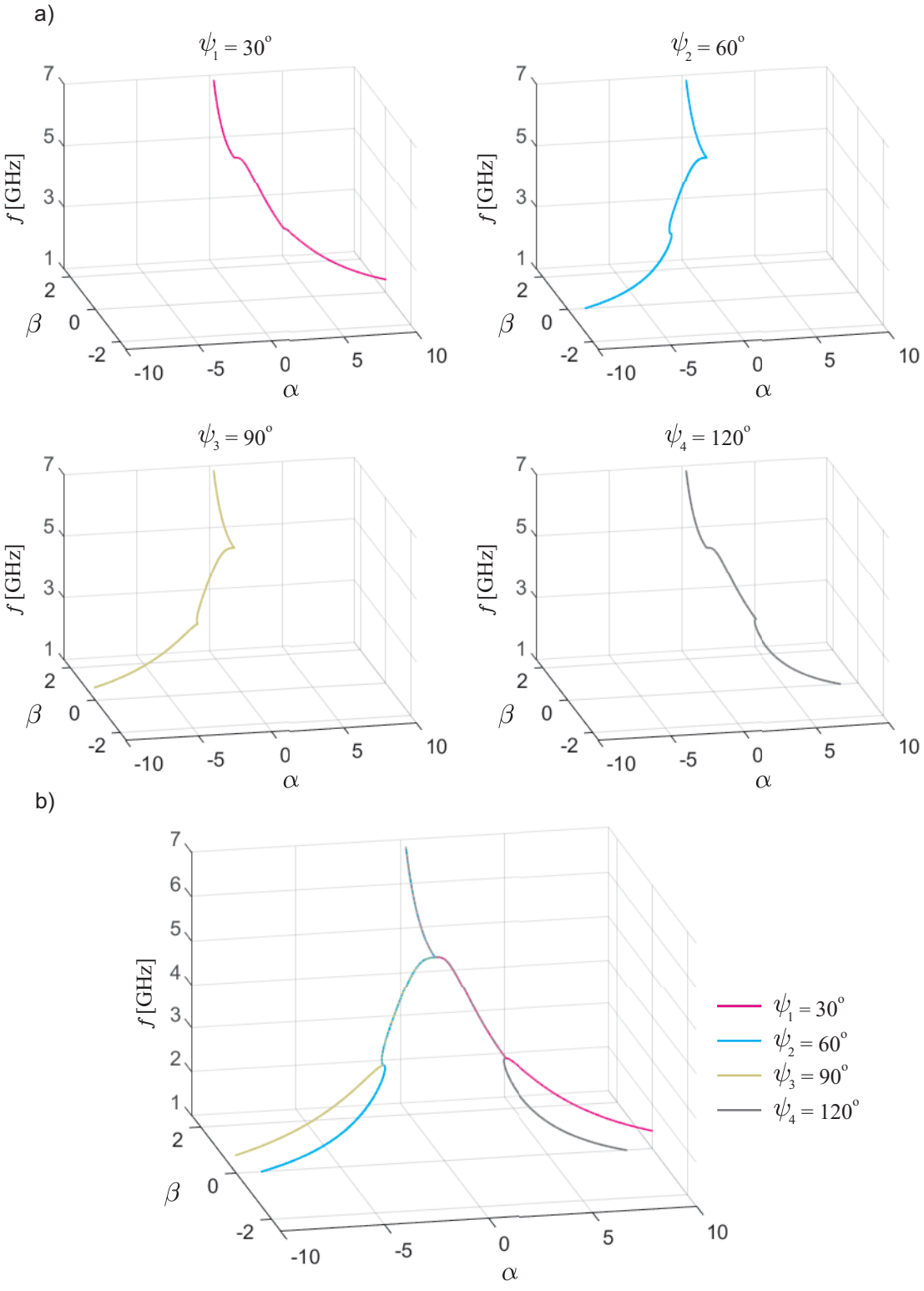

Figure 4.14: The characteristics of the normalized propagation coefficient for the analyzed example: a) obtained from four separate analyses for individual ψ, b) merged results in one plot.

4.3 Final comments

The proposed Self-adaptive complex root tracing algorithm effectively modifies the regular tracing routine and significantly reduces the number of function calls. The fundamental difference is only in changing the approach for discretizing the CAP condition using a self-adaptive method. It is a slight change in the algorithm operations, but it gives an impressive growth in its performance. In summary, for each of the above examples, a reduction in the number of function calls is obtained, which is also confirmed by the theoretical analysis and proves the thesis: ”The use of self-adaptive discretization of the CAP condition will reduce the number of function calls in the tracing algorithm [61] without impacting the achieved accuracy of the results”.

Chapter 5

Global complex roots and poles finding algorithm in $\mathbb{C} \times \mathbb{R}$ domain

Although there are many methods for complex root finding as a function of the extra parameter t, including the efficient global algorithms presented in the previous chapters (with all their improvements), they do not have a direct generalization for the case of functions of multiple variables, i.e., for example, in $\mathbb{C} \times \mathbb{R}$ space. As part of this dissertation, a new technique is proposed to increase the efficiency of finding zeros or poles for functions $F(z,t)$ with an additional parameter. This approach takes advantage of the features of global algorithms and eliminates problems that occur in tracing. Its goal is to obtain the full picture of zeros and poles of an investigated function $F(z,t)$ in the entire analyzed $\mathbb{C} \times \mathbb{R}$ space. It is done by iteratively discretizing the investigated region (based on domain Ω) and using the generalization of the DCAP condition to the $\mathbb{C} \times \mathbb{R}$ space (see section 4.1.2). The outcome is the determination of all $\xi(t)$ curves in a given region representing the roots of complex functions dependent on an additional parameter.

5.1 Formulation of the problem

One way to solve the problem of complex root finding as a function of extra parameters $F(z,t) = 0$ is to use the ”slicing method.” It involves looking for roots in the complex plane for the discredited value of the additional parameter t. Such an approach requires multiple calls to a global algorithm, e.g., GRPF, in a set number of planes $t = \mathrm{const}$. Depending on the complexness of the function, the discretization may be regular or require greater density (local) relative to t. In a given area of analysis, there may be many $\xi(t)$ curves; they may cross, intersect, branch, and have any arbitrary distribution. The ”slicing method” is especially inefficient when the $\xi(t)$ curve is parallel to the $t = \mathrm{const}$ plane. Creating an autonomous and flexible method of discretization of the t parameter is challenging due to the lack of a general rule. In addition, the ”slicing method” is an iterative procedure in which repeatedly performing the same operations is time-consuming and numerically expensive. In the case of complex numerical analyses, the time needed to evaluate a single function value can be long, so the efficiency of the algorithms is crucial.

As outlined in subsection 3.3, finding all zeros (and poles) with the minimal number of function calls (in the assumed domain) cannot be guaranteed by any global algorithm. Sometimes, some roots may be missed; in the case of the ”slicing method,” they may be in an adjacent plane (or not). So, it is difficult in the preliminary analysis to estimate the number of points and ensure no solution has been missed. For more efficient analysis, tracing techniques can be used. However, this type of algorithm also has its limitations, as presented in subsection 4.2.2. Tracing algorithms require a starting point and can only find one $\xi(t)$ curve (multipath is possible but inefficient). When the characteristic bifurcates or when the curves intersect, tracing algorithms may miss some of the solutions. It can be minimized by reducing the tracing resolution (smaller edge length of the tetrahedrons), but this leads to an increase in the number of function calls and more numerical cost.

5.1.1 Algorithm concept

The proposed technique is an iterative method based on regular GRPF (refer to subsection 3.2) and the generalization of DCAP to $\mathbb{C} \times \mathbb{R}$ space, which is also used in tracing (see subsection 4.1.2). First, a function is defined in the form $F(z,t)$ and the domain $\Omega \subset \mathbb{C} \times \mathbb{R}$, where the root will be searched. This area may be any geometry. Generally, the investigated region may be a range set for each of the three independent variables $\mathrm{Re}(z)$, $\mathrm{Im(z)}$, and t, creating a definite shape in the $\mathbb{R}^3$ space. Therefore, it is possible to discretize the entire Ω search area using a three-dimensional mesh composed of tetrahedrons. It is the fundamental difference between regular GRPF. At each of the mesh vertices, the value of the function is evaluated, and the argument of the function $arg(F(z,t))$ at the nodes is analyzed. By analyzing the phase differences between each vertex in edges, it is possible to determine ”candidate edges,” i.e., segments with a double change of quadrants at their ends. In each subsequent step, new nodes (additional points) are added to the mesh in the midpoint of the ”candidate edges” (analogous to the GRPF regular algorithm). A set of ”candidate edges” is thus determined, and the points that are the centers of these edges correspond to the found root of the function. Consequently, the calculated characteristic is a set of points in the $\mathbb{C} \times \mathbb{R}$ space. To summarize, the algorithm can be presented in the following steps:

1. The first step involves the initial discretization of the Ω region. Based on the defined initial edges length parameter r, a uniform three-dimensional initial mesh is created, constructed from K_0 tetrahedra based on nodes at N_0 points ($\mathrm{Re}(z_n)$,$\mathrm{Im}(z_n)$,t_n). For this purpose, Delaunay triangulation in the $\mathbb{R}^3$ domain is applied.

2. Then, the function values in each node of the initial mesh are evaluated $F_n = F(z_n, t_n)$.

3. The edges of the mesh are based on points in the $\mathbb{C} \times \mathbb{R}$ space, so it is possible to apply DCAP and determine the difference of the argument of the function $arg(F(z,t))$ along each edge p. From this, the phase change $\Delta\Theta_p = arg(F_{n_{p2}}) - arg(F_{n_{p1}})$ or, in a discrete version, the corresponding quadrant increment $|\Delta Q_p|$ is determined.

4. Next, a set of ”candidate edges” is determined depending on the version of the algorithm (see subsections 5.1.2 and 5.2). In the preliminary version (similar to regular GRPF), it is defined by the condition of quadrant change ($|\Delta Q_p| = 2$). In the final proposed version of this approach, ”candidate edges” are determined by meeting the absolute change of phase value ($|\Delta\Theta_p| > \pi/2$).

5. Then mesh is refinement, i.e., all ”candidate edges” are split into two equal parts by adding additional nodes in their centers. Adding new nodes requires adjusting the mesh and evaluating subsequent function values. Therefore, in the following iterations, the mesh is refined near the curves representing the zeros or poles of the function.

6. Finally, the $\xi(t)$ curves expressing the results of a given problem are formed based on the ”candidate edges.” More specifically, a simple relation is used: the midpoint of the ”candidate edges” represents zeros or poles in space $\mathbb{C} \times \mathbb{R}$ (without preliminary separation).

5.1.2 The primary version of the algorithm

The essential part of the proposed method is the shape of the mesh. Its appropriate refinement, i.e., reducing the length of edges, especially ”candidate edges,” makes the algorithm close to the true value of the root. The initial mesh is created from N_0 evenly distributed points ($\mathrm{Re}(z_n)$,$\mathrm{Im}(z_n)$,t_n), resulting from the initial edges length r. As it is done in the case of tracing, the normalization of the domain (mainly extra parameter) allows for improved efficiency. It increases the algorithm's stability, especially in cases where the additional parameter t is in units of a larger order, e.g., GHz. In the algorithm, by default, the Ω domain is normalized to the cube (hereinafter referred to as ”cubical” domain normalization). Nevertheless, the parameter t, but also the $\mathrm{Re}(z)$ and $\mathrm{Im}(z)$ domain range, can be scaled separately to the required range so that all variables are of a similar order of magnitude (hereinafter referred to as ”manual” domain normalization).

The primary version of the algorithm proposed that the condition for ending the analysis is to reduce each ”candidate edges” length to the specific value (defined by the accuracy parameter δ). In cases where the mesh is normalized, the edge length is normalized as well (δ also). It is important to note that this algorithm does not establish the ”candidate regions” (and does not analyze them) as typically found in GRPF. To ensure the validity of the proposed technique, numerical tests are performed on the examples outlined in subsection 2.3, particularly the acoustic waveguide and resonant structure. The objective is to confirm the effectiveness of the domain normalization to cube and assess the impact of the δ parameter on the results.

Anti-resonant reflecting acoustic waveguide

To start, the correctness of the results and domain normalization is checked by investigating the example of the ARRAW structure (depicted in subsection 2.3.4). An analysis of the complex longitudinal wavenumber $\beta(d)$ as a function of the cladding thickness $d\,[\mu\mathrm{m}]$ is performed. The inves-

tigation area aligns with that of reference [72]:

$$\Omega = \{\beta \in \mathbb{C}, d \in \mathbb{R} : 0.177 < \mathrm{Re}(\beta) < 15.887 \wedge 0.177 < \mathrm{Im}(\beta) < 15.887 \wedge 0.5 < d < 0.95\}$$

Table 5.1: The comparison of the range in mesh for two types of analysis, ”cubical” and ”manual” normalization of the domain Ω; x refers to $\mathrm{Re}(\beta)$, y to $\mathrm{Im}(\beta)$, and z corresponds to d.

normalization type	x **range**	y **range**	z **range**
”cubical”	$0 \leq x \leq 1$	$0 \leq y \leq 1$	$0 \leq z \leq 1$
”manual”	$0.177 \leq x \leq 15.887$	$0.177 \leq y \leq 15.887$	$5 \leq z \leq 9.5$

Each point from Ω is recalculated for the mesh range and vice versa to evaluate the function value. Table 5.1 shows the modified range of the domain from which the mesh is created. In the case of ”cubical” mode, a cube is created. For the ”manual” normalization, the scopes of $\mathrm{Re}(\beta)$ and $\mathrm{Im}(\beta)$ are not changed; only the thickness value d is multiplied by 10. The true points at which the function value is evaluated are scaled relative to the coordinates of the nodes on the mesh.

Table 5.2: The comparison of analysis parameters used for two domain normalization approaches. The number of starting points results from the assumed edge length in the initial mesh and the size of the normalized domain. The accuracy can not be compared directly because of the mesh volume (normalization), but the δ affects the precision of the results.

normalization type	**accuracy** δ	**no. of initial nodes** N_0	**initial edges length** r
”manual”	$2 \cdot 10^{-2}$	2 904	0.7
”cubical”	10^{-3}	1 000	0.1

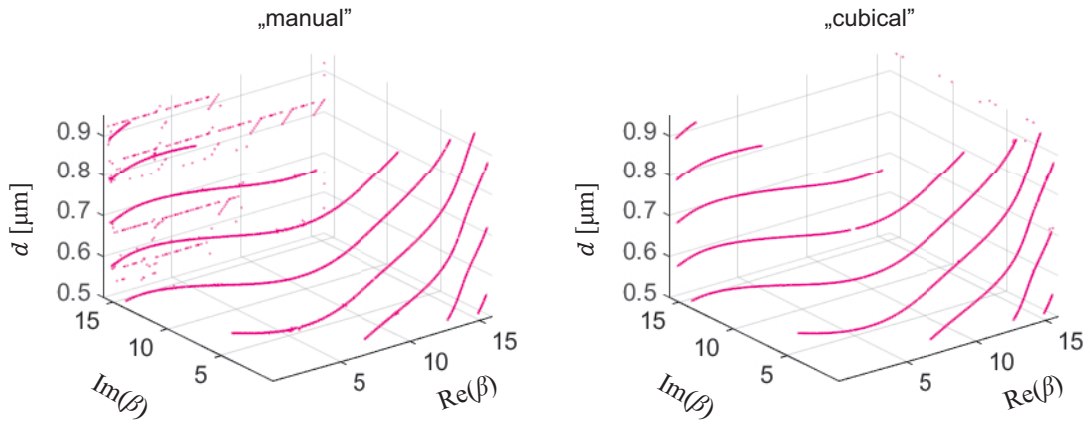

Figure 5.1: The obtained roots in the form of characteristics (after scaling to the initial Ω domain) of the complex longitudinal wavenumber β as a function of the cladding thickness d for two types of analyses with ”manual” and ”cubical” domain normalization.

Table 5.3: The comparison of the number of function calls, analysis times, and the number of iterations performed using the ”manual” and ”cubical” normalization types. The number of ”c.edges” refers to the found roots (number of points in characteristics).

normalization type	no. of function calls	no. of ”c.edges”	time	iterations
”manual”	79 526	23 656	116.3 s	18
”cubical”	71 888	25 431	114.6 s	18

According to Table 5.3, both normalization ”cubical” and ”manual” yield similar numerical effectiveness. It is due, among other things, to the fact that the algorithm parameters are selected to be equivalent for both types. Although the sizes of the domains are different, they are proportional to each other, as shown in Table 5.2. In general, setting the same number of initial nodes or assumed accuracy (obtained after re-scaling to the Ω input domain) is not feasible. Nevertheless, the advantage of normalizing to a cube is the more straightforward setting of parameters (no need to recalculate the edge length for the initial mesh and accuracy) and the ability to directly compare the results from various analyses. The obtained characteristics coincide with the references, which confirms the correctness of the proposed technique. In addition, analyzing the results of Figure 5.1, it can be noted that the characteristics obtained by ”manual” normalization of the domain include so-called ”noise,” i.e., fragments that are not the true solution. This is due to the greater disproportion between long and short edges on the mesh. It is worth recalling that the presented results are only the centers of ”candidate edges.” So, the worse the edge distribution (the mesh is ill-conditioned), the more artificial solutions or artifacts in the characteristics may appear. A mesh with a cubic shape is not free of these limitations (due to local mesh refinement around the ”candidate edges”). However, they are noticeable to a more secondary degree and are not a significant issue.

Cylindrical-rectangular microstrip resonant structure

The second example involves a cylindrical-rectangular microstrip resonant structure (see subsection 2.3.6). The analysis concerns the behavior of the complex resonance frequency $f_r(s)$ [GHz] in a function of the air gap thickness s [mm] in the investigated structure in the range:

$$\Omega = \{f_r \in \mathbb{C}, s \in \mathbb{R} : 1 < \mathrm{Re}(f_r) < 1.5 \wedge -0.2 < \mathrm{Im}(f_r) < 0 \wedge 0 < s < 10\}$$

.

In this instance, the computational domain is normalized to a cube using ”cubical” type. The initial mesh contains $N_0 = 1\,000$ nodes defined by the $r = 0.1$ parameter (normalized edges length). Four independent calculations are carried out for varying accuracies (parameter δ is swept). The results of the numerical tests are presented in Table 5.4. It is widely acknowledged that as the accuracy increases, the number of function calls, as well as the total computation time, increases.

Figure 5.2 shows the obtained characteristics (obtained after re-scaling to the Ω input domain) for each tested accuracy. It is evident that a smaller delta parameter means a denser mesh with shorter ”candidate edges,” and therefore, the obtained results are more continuous and expressive.

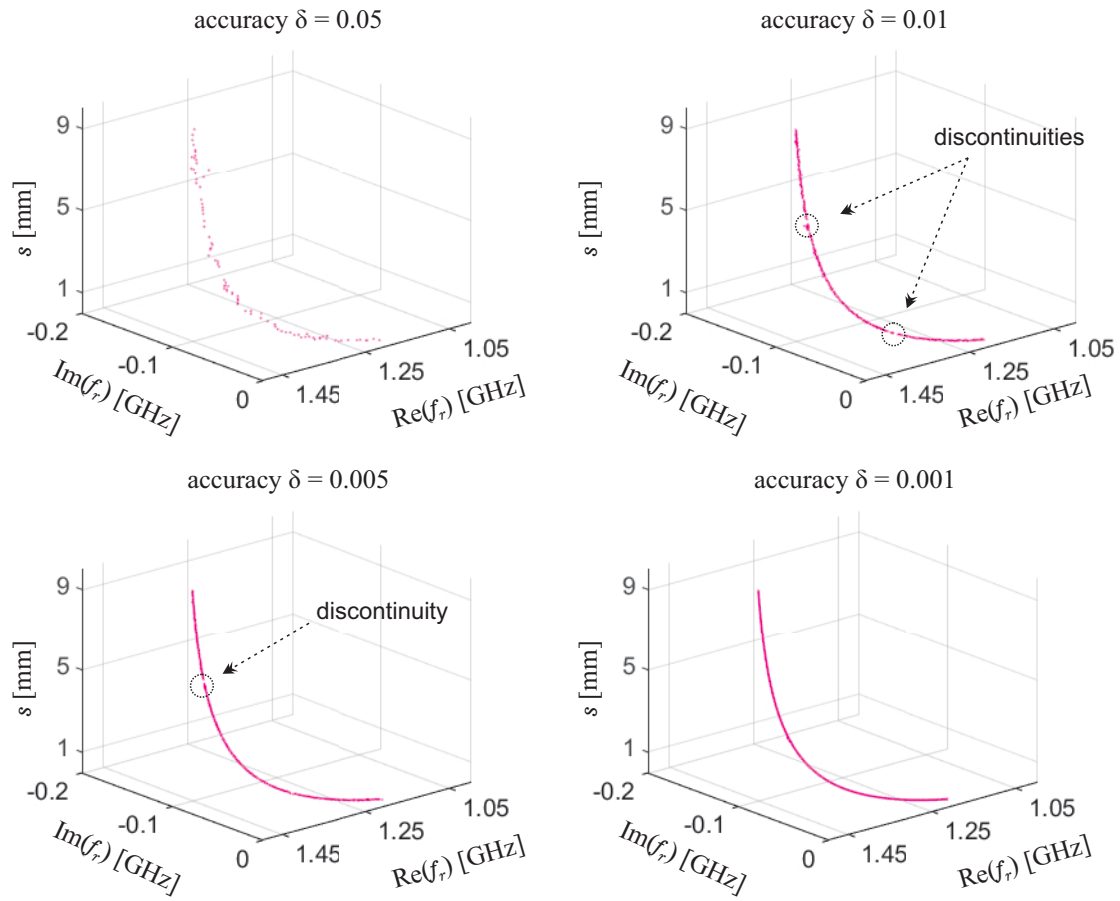

Figure 5.2: The obtained roots in the form of characteristics (after scaling to the initial Ω domain) of the complex resonant frequency f_r as a function of one of the structure dimensions s for four analyses with different accuracies. Places where the characteristics show discontinuities are indicated.

Table 5.4: The comparison of the number of function calls, analysis times, and the number of iterations performed for four analyses with different accuracies. The number of ”c.edges” refers to the found roots (number of points in characteristics).

accuracy δ	**no. of function calls**	**no. of ”c.edges”**	**time**	**iterations**
0.05	1 149	87	36 min 19 s	5
0.01	2 025	526	1 h 4 min 1 s	12
0.005	2 918	894	1 h 31 min 45 s	15
0.001	10 945	4 624	5 h 39 min 10 s	22

In general, due to the specific arrangement of edges in the mesh relative to the investigated curve ξ, some areas will not be densified because the ”candidate edges” with a double quadrant change will not appear there. As a result of such a situation, discontinuities appear in the outcome characteristics. One way to eliminate such ”gaps” is to use post-processing. Missing parts of the characteristics could be detected by creating chains of tetrahedrons (similar to tracing) and discontinuities removed by refinement of these critical mesh parts. It requires additional analysis and creating an appropriate condition for new analysis (domain range, initial edge length, accuracy, and others). It is generally challenging to automate and decrease the efficiency.

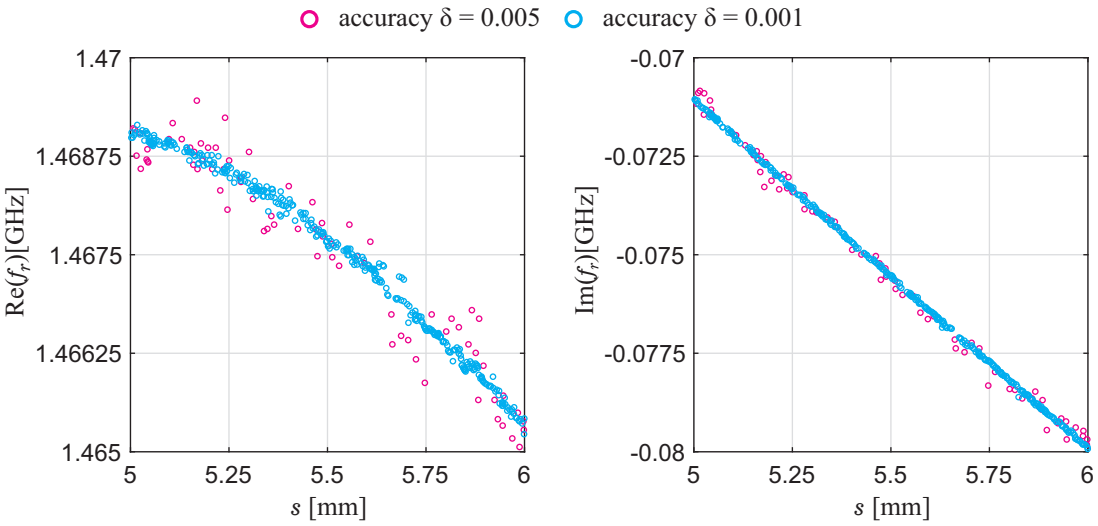

Figure 5.3: The complex resonant frequency is shown in the function of one of the structure dimensions for two accuracies. The zoom of the crucial domain fragment where there are discontinuities in the obtained characteristics. On the left is the projection on the real axis, and the right is the imaginary axis.

In summary, the proposed technique provides a comprehensive view of the characteristics (curves $\xi(t)$) and serves as a ground for potential in-depth analysis. Local algorithms may be suitable for this purpose, as they can achieve high accuracy with minimal function calls and can be applied to any plane (regardless of whether $t = \text{const}$). Nevertheless, the proposed algorithm is highly efficient, as confirmed by the tabled results. The distribution of ”candidate edges” and their lengths reflects the resolution and accuracy of the results obtained. When shorter edges are in close proximity to each other, the distribution is more solid, leading to more precise outcomes (refer to Figure 5.3).

5.2 The final approach

After careful analysis of the previous subsection’s conclusion and limitations, some modifications are made to the primary version of the algorithm. The final version of the proposed approach, outlined below, emphasizes the area of zeros and poles near ”candidate edges.” In the primary approach, ”candidate edges” are assumed to be edges with double quadrant changes along their

vertices. It is a discrete version of the general CAP condition, in which the function's argument changes more than $\pi/2$ in the vicinity of roots. Thus, to eliminate the problematic issue of discontinuity in characteristics, the condition on "candidate edges" is just modified formally. As a result, in the proposed approach, an edge p is considered a "candidate edge" if it satisfies the following condition:

$$|\Delta\Theta_p| > \pi/2 \tag{5.1}$$

This notation is, in some special cases, equivalent to changing the argument by two quadrants, but it works more generally. For example, when the phase change from 359° to 107°, it is a change about two quadrants. However, when the phase changes from 1° to 109°, it is a change of one quadrant, even though both cases have $|\Delta\Theta_p| = 3\pi/5\,(108^\circ) > \pi/2$. It is not necessary to discretize the function's argument with respect to the angles $0, \pi/2, \pi, 3\pi/2$.. Hence, there is no argumentation for applying such a discretization concerning another one shifted by any angle.

5.2.1 Operation of the algorithm

The primary version of the algorithm assumes determining the length of the edges (r) of the initial mesh. This parameter is associated with the size of the domain and defines the number of starting points (N_0). The number of final points is not imposed in advance, but the higher the accuracy δ, the greater the number of function calls. In order to simplify these two parameters, the new approach introduces one input parameter that must be set and is the maximum number of function calls N_{max}. This value can vary depending on the complexity of the function and determines the effectiveness and accuracy of the analysis performed. The numerical effort involved in evaluating a single function value allows one to estimate in advance the running time of the whole algorithm. If the obtained resolution of results is insufficient, the N_{max} number can be increased in further stages of the algorithm. The user observes the individual stages of the formation of the curves $\xi(t)$ and can decide during the process to increase the number of function calls to improve the quality of the results.

The number of nodes in the initial mesh N_0 depends on the total number of nodes N_{max} and is usually one percent of that number. The remaining part of the assumed number of nodes (99% of N_{max}) will be added during the mesh refinement in successive stages of the algorithm. In special circumstances, the user can modify the ratio between the number of initial and final nodes $N_0 = \eta N_{max}$, such as increasing N_0 when N_{max} is relatively low.

In the GRPF in $\mathbb{C} \times \mathbb{R}$ domain analogously to the regular GRPF procedure, according to the new condition (5.1), edges along which the phase of the function changes by more than $\pi/2$ are included in the set of "candidate edges." It is near these edges that the zeros or poles of the function are located. However, detecting the roots in this domain region does not require the evaluation of q. It should also be mentioned that, as in regular GRPF, if the mesh becomes ill-defined - "skinny" tetrahedrons may occur. To avoid such situations, all faces (triangles) in the mesh are verified as "skinny" according to the condition described in subsection 3.4.2. If the ratio of the longest edge to

the lowest height on the face exceeds an assumed value[1], the triangle is supposed to be ”skinny,” and a new node is added at the center of the longest edge of such a triangle.

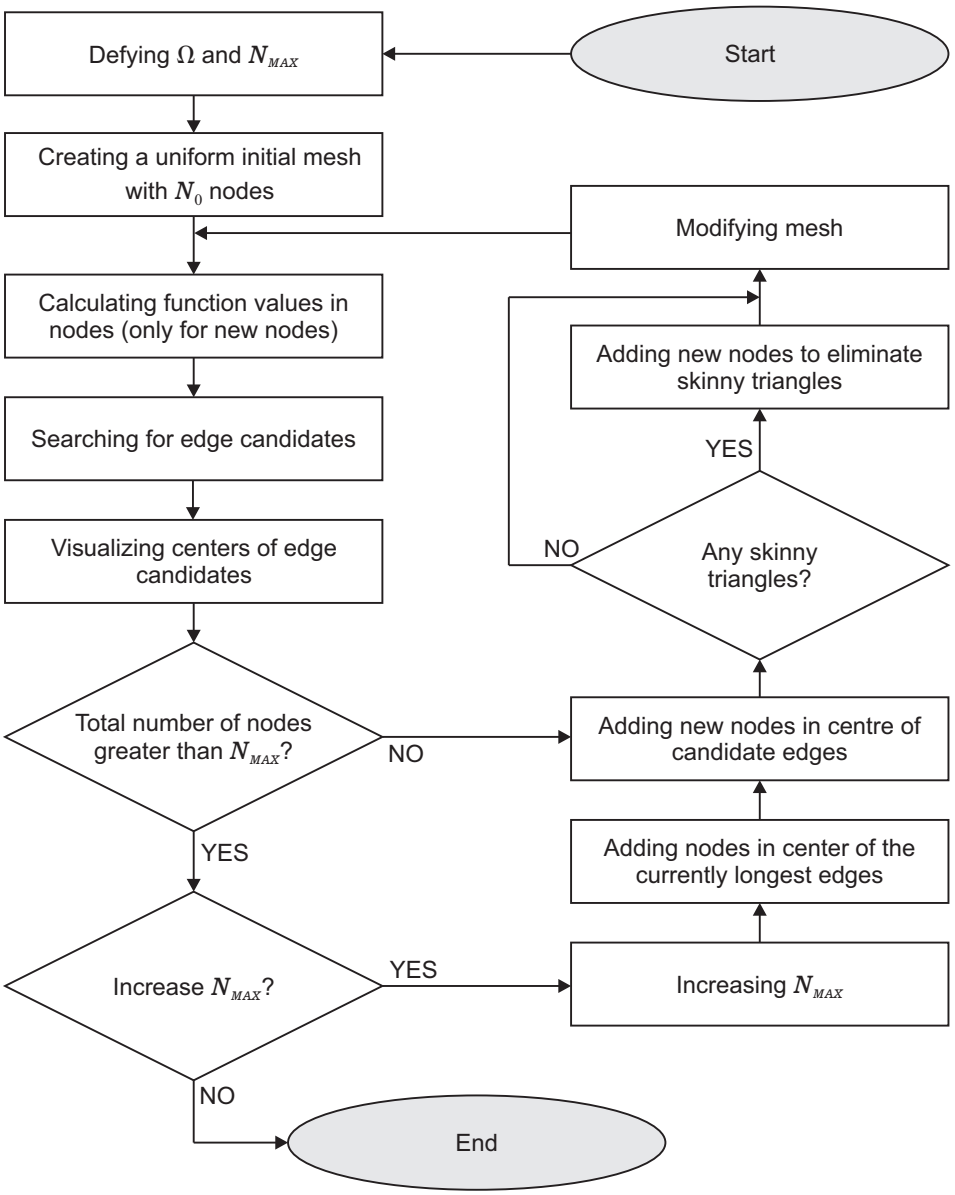

Figure 5.4: Block diagram of operating the proposed GRPF in $\mathbb{C} \times \mathbb{R}$ domain algorithm.

The algorithm stops operating when the number of function calls reaches N_{max}. The entire procedure is shown in Figure 5.4. The above description of the algorithm and the examples presented in the following subsection are parts of the publication [118] of the author of this dissertation. The source code of the Global Complex Roots and Poles Finding Algorithm in the in $\mathbb{C} \times \mathbb{R}$ Domain can be found at: https://github.com/PioKow/GRPF3D.

[1]Typically, this value is ten based on prior experience and tests.

5.2.2 Algorithm details

To gain a better understanding of the proposed algorithm at hand, it is important to note the following details:

- If the value of a function at a particular node is calculated to be a numerical zero (below machine precision) or infinity, the function argument becomes impossible to evaluate. In such cases, every edge connected to this node is permanently designated as ”candidate edges” (as in the regular GRPF).
- The algorithm accounts for a potential increase in N_{max} following the analysis. If the user decides to adjust this number, nodes equivalent to one percent of the total are evenly added across the entire domain, just like in the initial mesh process. These nodes are added to the middle of the longest edges, not just on ”candidate edges.”

The algorithm may seem computationally expensive at first peek. Nevertheless, the iterative process itself, including triangulation, is relatively inexpensive in terms of CPU resources and memory usage. The most significant numerical complexity lies in function evaluation. It is clear that the size of the investigation domain Ω and the level of complexity of curves in this region are also crucial in the context of the analysis time. It is these that directly affect the necessary number of function calls. Like standard global algorithms (operating only in the complex plane), this algorithm does not provide a hundred percent guarantee of finding all zeros and poles for any complex function with a finite number of calls. However, when performing an investigation by additional parameter t, the chance of skipping one of the results is lower (assuming continuity of the function). This is due to the higher probability of convergence for a previous or next t value. Furthermore, increasing N_{max} during the analysis reduces the risk of missing one of the curves $\xi(t)$ while improving the resolution and accuracy of the obtained results.

5.2.3 Numerical examples

In order to confirm the stability and efficiency of the proposed approach, four various example propagation and resonance problems are analyzed. Additionally, to prove the assumed thesis in the first chapter (see subsection 1.2), the ”slicing method” is implemented to compare the results. It involves calculating roots using the Self-Adaptive Mesh Generator for GRPF for different values of the extra parameter t.

Anti-resonant reflecting acoustic waveguide

The first example in this fragment of work concerns the ARRAW structure described in subsection 2.3.4. As in the example for the primary version of this algorithm, an analysis of complex longitudinal wavenumber $\beta(d)$ as a function of the cladding thickness $d\,[\mu\mathrm{m}]$ is performed.

However, in this case, the β_n parameter is normalized such that:

$$\begin{gathered}\beta_n = n_{eff} + j\ \mathrm{loss} \\ n_{eff} = \frac{\mathrm{Re}(\beta)}{k_s^1} \quad \mathrm{loss} = \frac{\mathrm{Im}(\beta)}{k_s^1}\end{gathered} \tag{5.2}$$

where:

n_{eff} - effecitve mode index,

k_s^1 - shear transverse wavenumber in the first layer $[1/\mathrm{m}]$.

The analysis is carried out setting $N_{max} = 100\,000$ (the number of initial nodes is $N_0 = 1\,000$) in the domain:

$$\Omega = \{\beta_n \in \mathbb{C}, d \in \mathbb{R} : 0.01 < \mathrm{Re}(\beta_n) < 0.9 \wedge 0.01 < \mathrm{Im}(\beta_n) < 0.9 \wedge 0.5 < \mathrm{d} < 0.95\}$$

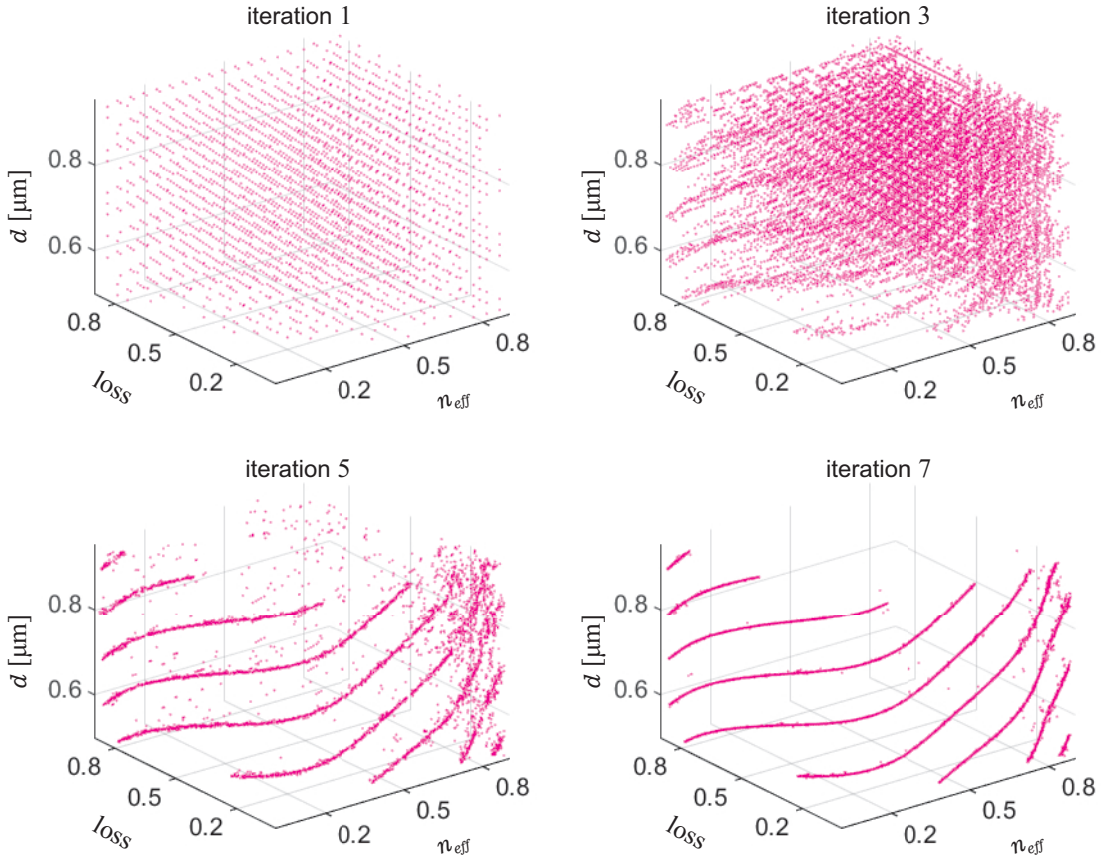

Figure 5.5: The iterative process of the proposed algorithm: GRPF in $\mathbb{C} \times \mathbb{R}$ domain for the ARRAW example. The normalized longitudinal wavenumber as a function of one of the structure dimensions is refined in each step.

Table 5.5: The number of nodes in the mesh and the number of ”candidate edges” in each iteration of the proposed algorithm for the ARRAW example. The number of ”candidate edges” refers to the found roots (number of points in characteristics).

iteration	no. of nodes	no. of ”candidate edges”
1	1 000	2 726
2	3 726	9 516
3	13 233	14 880
4	28 113	7 922
5	36 027	5 630
6	41 654	7 876
7	49 524	13 610
8	63 111	25 176
9	88 255	47 226
10	99 994	55 974
11	100 000	55 974

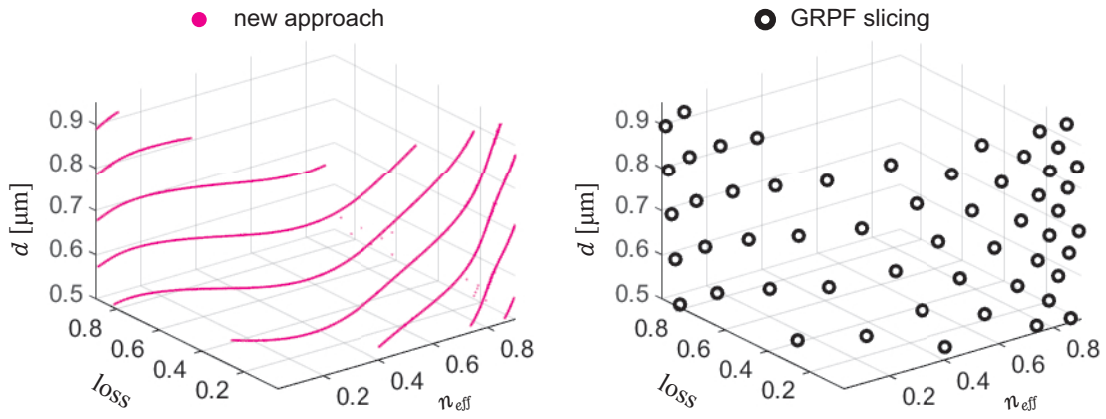

Figure 5.6: The obtained roots in the form of characteristics of the normalized complex longitudinal wavenumber β as a function of the cladding thickness d for two types of analyses; On the left are results from GRPF in $\mathbb{C} \times \mathbb{R}$ domain; On the right is the ”slicing method” applying the Self-Adaptive GRPF in the ten planes.

Figure 5.5 shows the operation of the algorithm for selected stages of the analysis. Table 5.5 displays the total number of mesh nodes and the number of ”candidate edges” for all iterations. After the 11th iteration, the assumed limit of Nmax function calls is reached. The final characteristics obtained include nine modes (see Figure 5.6). In order to verify the results with reference [72], the projections of the real and imaginary axes are illustrated in Figure 5.7. Furthermore, the results from the ”slicing method” at ten regular distant planes (different thicknesses d) are added for comparison. The analysis of each plane $d = \text{const}$ is carried out with Self-Adaptive GRPF

using about[2] $10\,000$ nodes, resulting in a total number of function calls equal to around $100\,000$ in the assumed Ω.

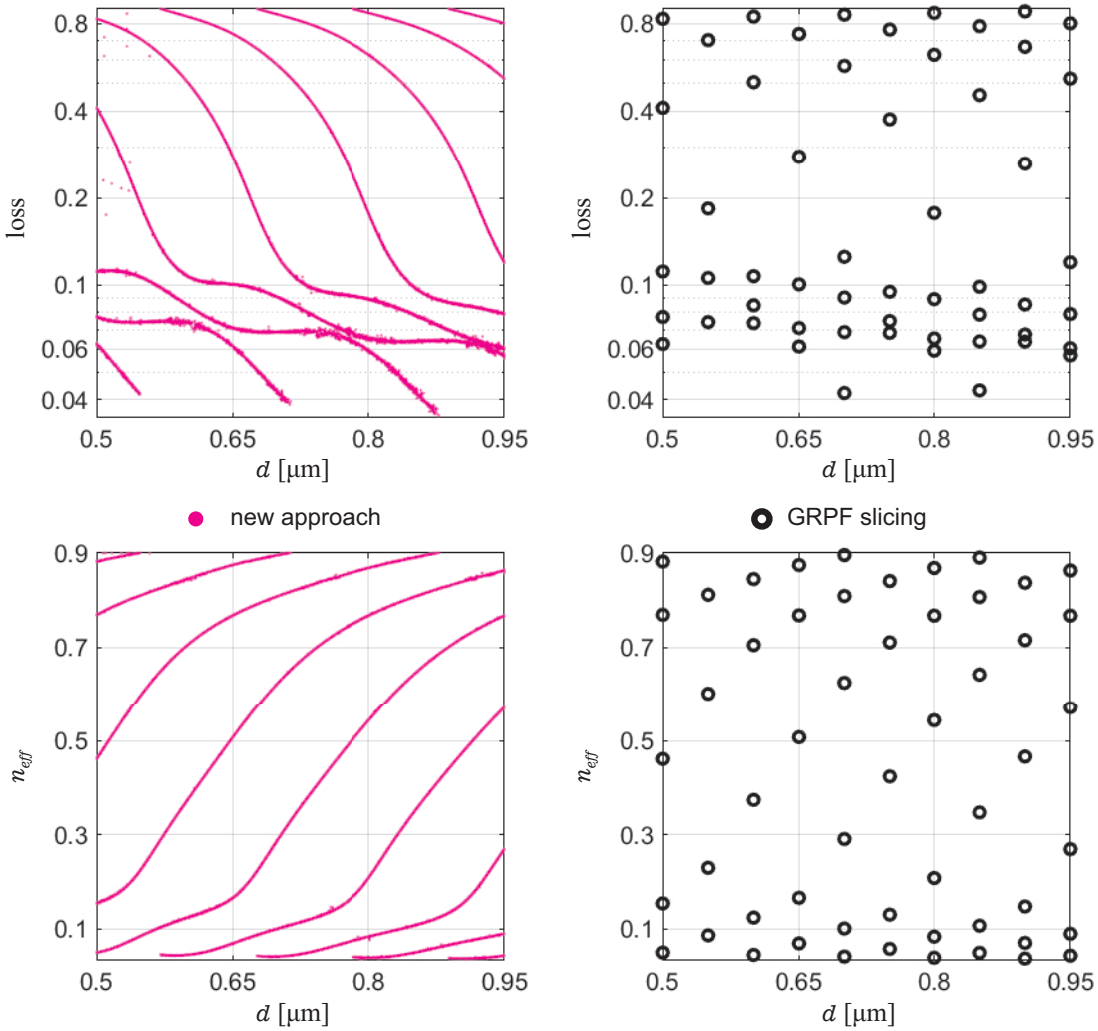

Figure 5.7: The projections on the real axis (n_{eff}), and on the imaginary axis (loss) of the obtained characteristics. The imaginary part representing losses is shown on a logarithmic scale. On the left are results from GRPF in $\mathbb{C} \times \mathbb{R}$ domain; On the right is the "slicing method" applying the Self-Adaptive GRPF in the ten planes.

Table 5.6: The comparison of the analysis times of the ARRAW example for equivalent the number of function calls in two types of analysis: the proposed technique and the "slicing method."

algorithm	**no. of function calls**	**time** [s]
new approach	$100\,000$	45
GRPF slicing	$10 \times \approx 10\,000$	173

[2]In the family of GRPF approach, it is not possible to set precisely a number of the function calls, only the maximum number of nodes when the self-adaptive mesh generator is stopped; after that is done the regular mesh refinement where the total number of points relies on the assumed accuracy.

Analyzing the above results in the form of characteristics in the drawings and the times included in Table 5.6, one can see a significant advantage of the proposed technique over the ”slicing method.” GRPF in $\mathbb{C} \times \mathbb{R}$ domain gives a full picture of the curves ξ occurring in an investigated region. In the ”slicing method,” the analysis time is significantly longer (with a similar number of points as in the new approach). It results from repeated re-initializations of Self-Adaptive GRPF with the new initial conditions. Additionally, although the results are correct, their resolution needs to be higher to make the obtained characteristics interpretable. Using the ”slicing” approach requires a large number of points (calculated redundantly). Increasing the accuracy would require extra analyses in additional planes $d = \text{const}$. It increases the total number of points by several orders of magnitude. The GRPF in $\mathbb{C} \times \mathbb{R}$ domain is free of these disadvantages and is very efficient and effective.

Coaxially loaded cylindrical waveguide

The next example is coaxial cylindrical waveguide loaded with a dielectric. This structure is described in the subsection 2.3.2 but also investigated in the tracing chapter. The normalized propagation coefficient γ as a function of frequency f is analyzed. The analysis is carried out setting $N_{max} = 100\,000$ (the number of initial nodes is $N_0 = 1\,000$) in the domain (comparable to tracing analysis):

$$\Omega = \{\gamma \in \mathbb{C}, f \in \mathbb{R} : -4.5 < \text{Re}(\gamma) < 4.5 \wedge -0.25 < \text{Im}(\gamma) < 2.25 \wedge 2 < f < 7\}$$

The calculated dispersion characteristics are shown in Figure 5.8 as a three-dimensional graph. Figure 5.9 shows the projections on the real and imaginary axes. Table 5.7 compares the analysis times for the proposed technique and the ”slicing method.” As one can see, the new approach is characterized by higher performance.

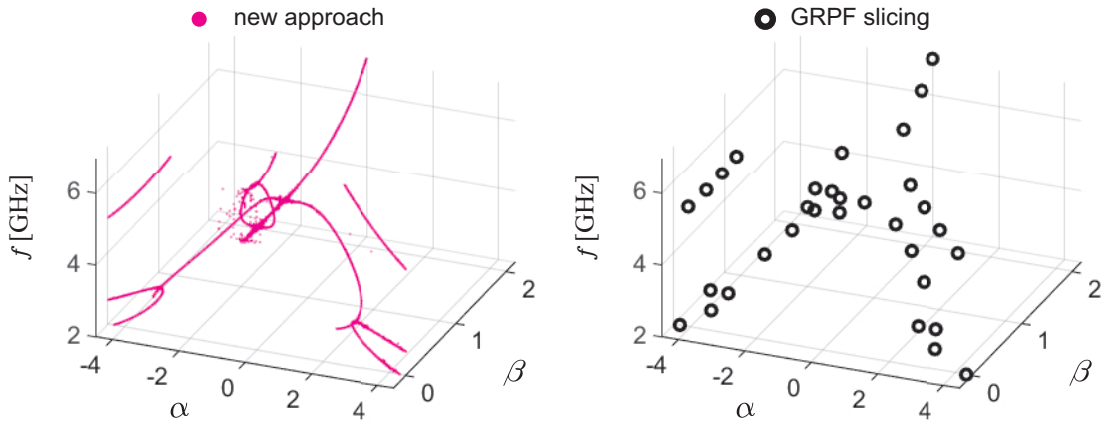

Figure 5.8: The obtained roots in the form of dispersion characteristics of the normalized propagation coefficient γ as a function of frequency f for two types of analyses. On the left are results from GRPF in $\mathbb{C} \times \mathbb{R}$ domain; On the right is the ”slicing method” applying the Self-Adaptive GRPF in the ten planes.

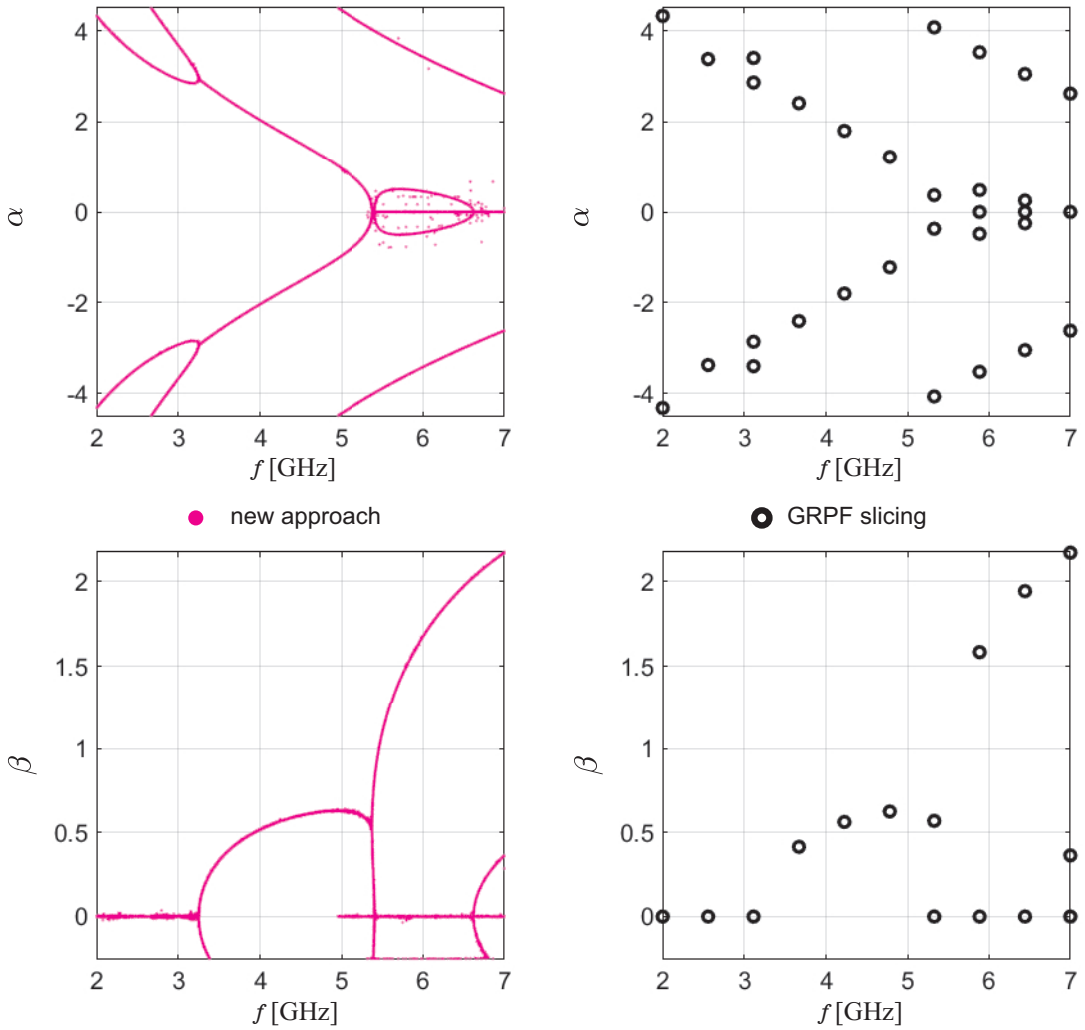

Figure 5.9: The projections on the real axis (attenuation constant) and on the imaginary axis (phase constant) of the obtained dispersion characteristics. On the left are results from GRPF in $\mathbb{C} \times \mathbb{R}$ domain; On the right is the ”slicing method” applying the Self-Adaptive GRPF in the ten planes.

It is noted that the presented new approach provides a better overview of solutions in the analyzed region as compared to the characteristics obtained in subsection 4.2.3 (see Figure 4.13 b)). The results obtained from the tracings are extended with additional mods. The limitations of multipath tracing are eliminated, meaning one analysis gets the full picture, and the branches are not a problem to find. Similar to the previous example, the results from the ”slicing method” at ten regular distant planes (different frequencies f) are also in Figures 5.8 and 5.9. The analysis of each plane $f = \text{const}$ is carried out with Self-Adaptive GRPF using about $10\,000$ nodes, resulting in a total number of function calls equal to around $100\,000$ in the assumed Ω. The results obtained from both algorithms are convergent. However, much denser discretization as a function of frequency is required to obtain a full picture of the solutions by the ”slicing method.”

Table 5.7: The comparison of the analysis times for equivalent the number of function calls in two types of analysis: the proposed technique and the ”slicing method.”

algorithm	**no. of function calls**	**time** [s]
new approach	$100\,000$	24
GRPF slicing	$10 \times \approx 10\,000$	102

Graphene transmission line

In the third example in this section, an analysis of the propagation coefficient γ as a function of frequency f [THz] in a graphene transmission line (see subsection 2.3.3) is conducted. A similar investigation is also performed using the tracing method, which is presented in subsection 4.2.1. The analysis is done by setting $N_{max} = 100\,000$ (the number of initial nodes is $N_0 = 1\,000$) in the following domain:

$$\Omega = \{\gamma \in \mathbb{C}, f \in \mathbb{R} : -100 < \mathrm{Re}(\gamma) < 350 \wedge 25 < \mathrm{Im}(\gamma) < 300 \wedge 1 < f < 7\}$$

Once more, the calculated dispersion characteristics are presented in Figure 5.10 as a three-dimensional graph. In Figure 5.11, the projections on the real and imaginary axes are shown. Table 5.8 compares the analysis times for two approaches: the GRPF in $\mathbb{C} \times \mathbb{R}$ domain and the ”slicing method” (at ten regular distant planes with different frequencies f). The analysis of each plane $f = \mathrm{const}$ is carried out with Self-Adaptive GRPF using about $10\,000$ nodes, resulting in a total number of function calls equal to around $100\,000$ in the assumed domain. Likewise, the results obtained from both algorithms are convergent. However, it can be observed that the proposed new approach is more efficient than the other method.

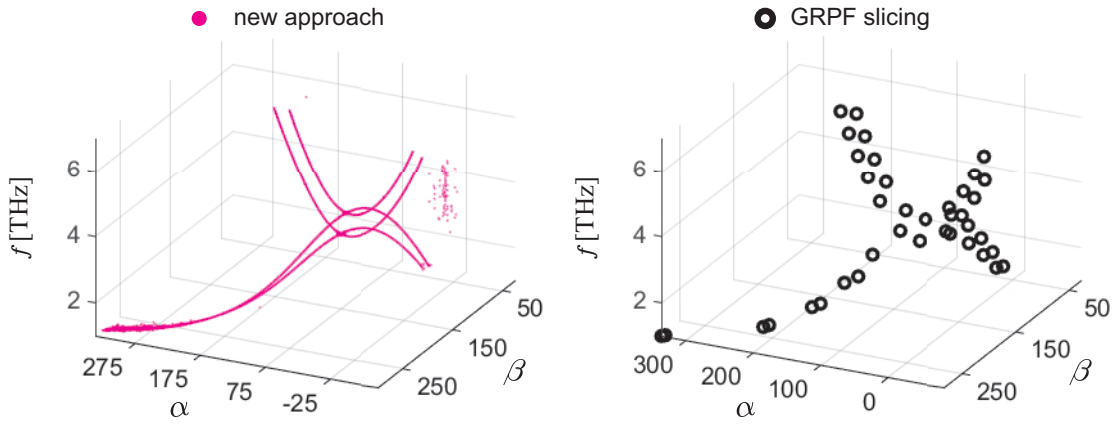

Figure 5.10: The obtained roots in the form of dispersion characteristics of the normalized propagation coefficient γ as a function of frequency f for two types of analyses. On the left are results from GRPF in $\mathbb{C} \times \mathbb{R}$ domain; On the right is the ”slicing method” applying the Self-Adaptive GRPF in the ten planes.

Table 5.8: The comparison of the analysis times for equivalent the number of function calls in two types of analysis: the proposed technique and the ”slicing method.”

algorithm	**no. of function calls**	**time** [s]
new approach	100 000	15
GRPF slicing	$10 \times \approx 10\,000$	82

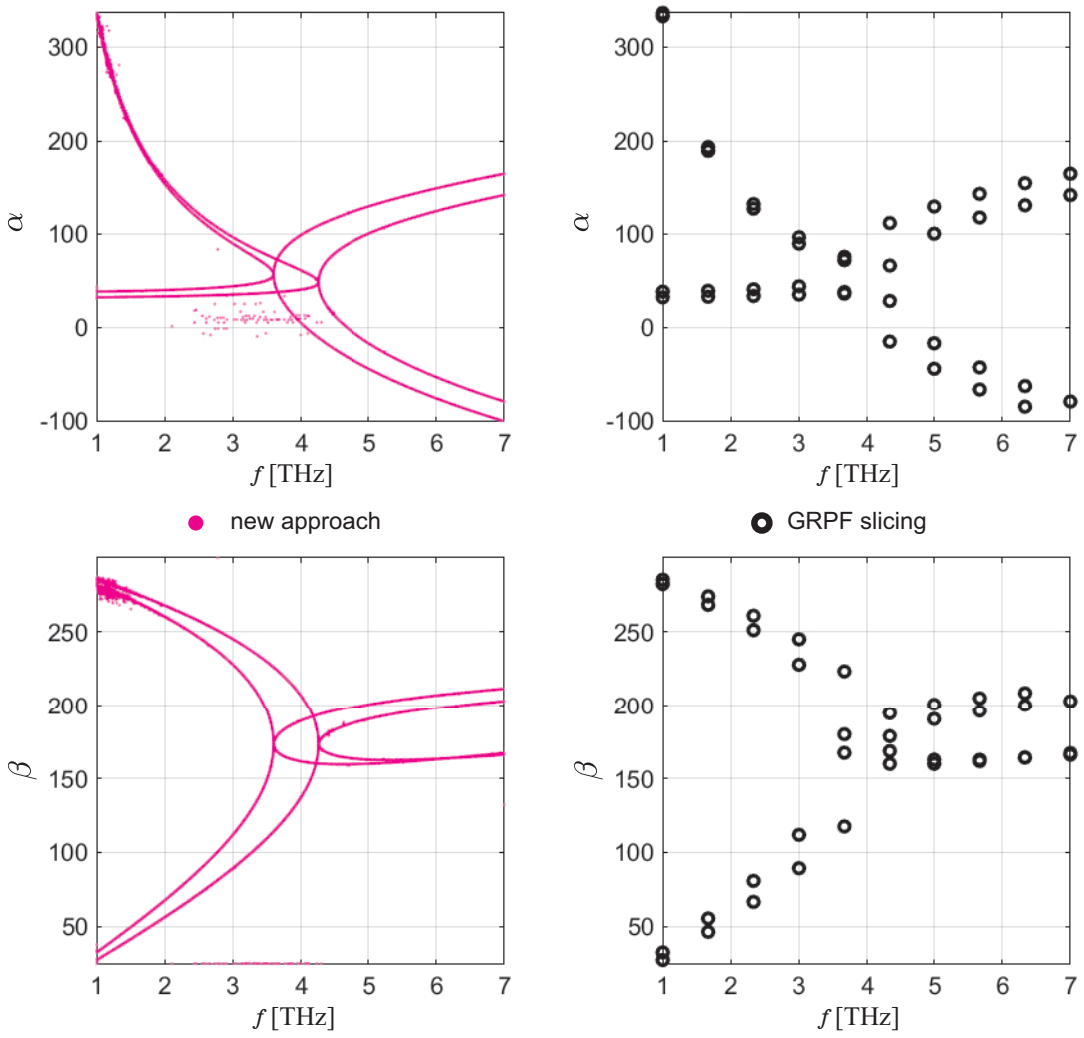

Figure 5.11: The projections on the real axis (attenuation constant) and on the imaginary axis (phase constant) of the obtained dispersion characteristics. On the left are results from GRPF in $\mathbb{C} \times \mathbb{R}$ domain; On the right is the ”slicing method” applying the Self-Adaptive GRPF in the ten planes.

As can be seen in Figure 5.10, the GRPF in $\mathbb{C} \times \mathbb{R}$ domain results contain some scattered points that are not necessarily roots (i.e., artificial solutions). It is the so-called ”noise”, a phenomenon already discussed in the primary version of this algorithm. In critical domain regions, such as mode intersection, the mesh needs to be denser to separate the curves precisely. In order to improve

the resolution of the obtained result, further analysis is carried out in the necessary subdomains:

$$\Omega_1' = \{\gamma \in \mathbb{C}, f \in \mathbb{R} : 35 < \mathrm{Re}(\gamma) < 70 \wedge 165 < \mathrm{Im}(\gamma) < 180 \wedge 3.56 < f < 3.70\}$$
$$\Omega_2' = \{\gamma \in \mathbb{C}, f \in \mathbb{R} : 35 < \mathrm{Re}(\gamma) < 70 \wedge 165 < \mathrm{Im}(\gamma) < 180 \wedge 4.24 < f < 4.38\}$$

For both subregions Ω_1' and Ω_2', the $N_{max} = 25\,000$ with the number of starting nodes $N_0 = 216$ is assumed for analysis. Figure 5.12 illustrates the roots found in the subregions Ω' from the initial analysis of the entire domain, as well as the results of an extra analysis that involves only the critical region. The data from the full-domain analysis does not provide conclusive evidence as to whether the curves intersect or merely come close to each other. Additional analysis is necessary to obtain a clear understanding of the curve behavior.

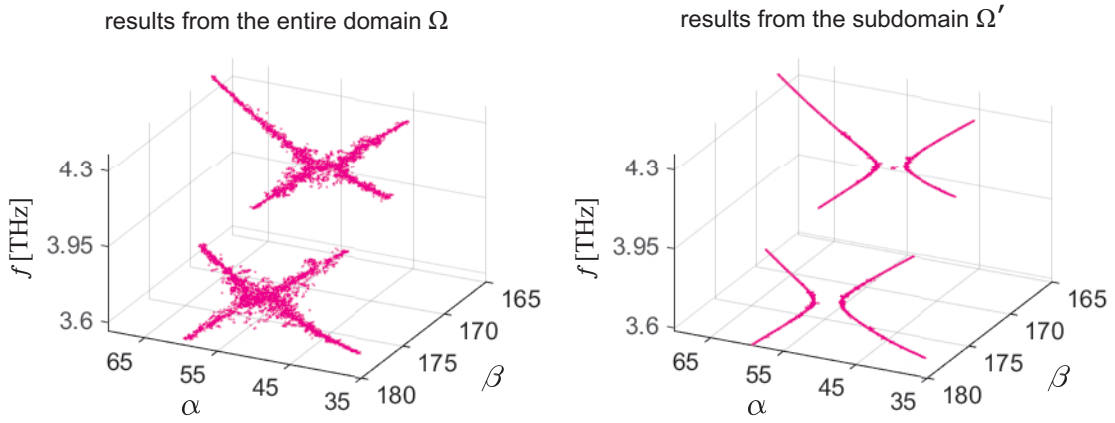

Figure 5.12: The normalized propagation coefficient as a function of the frequency in the investigated subregions. On the left is a zoom of the critical region of the intersection of characteristics from Figure 5.8. On the right are results obtained from separate analyses of the two for the Ω_1' and Ω_2' subdomains.

One method to obtain a more accurate characterization in the indicated area is to increase N_{max} at the end of the analysis. Initially, this may seem like the obvious solution. Nevertheless, the number of new nodes increases directly proportional to the number of ”candidate edges” in each iteration. So, their quantity may quickly exceed even $1\,000\,000$. New points are added in all parts of the curves, which are already redundant in some fragments. Consequently, it seems a reasonable solution to focus only on the ”critical” regions by re-analyzing the reduced part of the domain.

Fabry-Pérot open resonator

The last example includes the structure of the Fabry-Pérot open resonator presented in subsection 2.3.7. The system in the configuration with two spherical mirrors is analyzed. According to reference [140], the real part of the permittivity of the investigated plate is $\varepsilon_r = 5.1217$. In this example, the subject of research is the impact of dielectric losses defined by the loss tangent $\tan\delta$ on the resonance frequency $f_r(\tan\delta)[\mathrm{GHz}]$. Since the dimensions of the structure are large relative to the wavelength (resulting from the operating frequency), the discretization of the computational

domain in the finite element method used is numerically demanding. Therefore, the search for roots is limited to relatively narrow ranges of frequency and loss tangent. The analysis is done by setting $N_{max} = 10\,000$ (the number of initial nodes is $N_0 = 125$) in the following domain:

$$\Omega = \{f_r \in \mathbb{C}, \tan\delta \in \mathbb{R} : 20.35 < \mathrm{Re}(z) < 20.4 \wedge 0.05 < \mathrm{Im}(z) < 0.1 \wedge 0.01 < \tan\delta < 0.1\}$$

Based on the convergence analysis detailed in [17], the number of elements that should be used in the finite element method equals $5\,887$. As shown in Figure 5.13, the level of discretization used proved to be sufficient to obtain a clear overview of the three-dimensional characteristics.

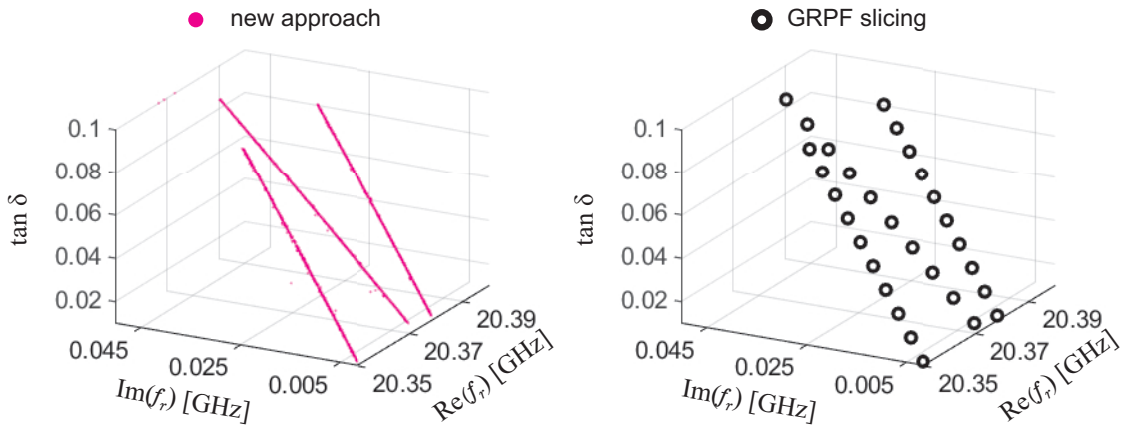

Figure 5.13: The obtained roots in the form of characteristics of the resonant frequency f_r as a function of loss tangent $\tan\delta$ for two types of analyses. On the left are results from GRPF in $\mathbb{C} \times \mathbb{R}$ domain; On the right is the "slicing method" applying the Self-Adaptive GRPF in the ten planes.

Table 5.9 compares analysis times between two approaches: the GRPF in the $\mathbb{C} \times \mathbb{R}$ domain and the "slicing method" (at ten regular distant planes with different dielectric losses). For each plane at a constant $\tan\delta$, the Self-Adaptive GRPF method is used with approximately $1\,000$ nodes, resulting in a total of around $10\,000$ function calls in the given domain. In contrast to the previous examples, evaluating a single function takes much longer because the problem requires a numerical analysis. Hence, every improvement in the algorithm's operation is directly expressed in the total computational time.

Table 5.9: The comparison of the analysis times for equivalent the number of function calls in two types of analysis: the proposed technique and the "slicing method."

algorithm	no. of function calls	time
new approach	$10\,000$	78 min 42 s
GRPF slicing	$10 \times \approx 1\,000$	90 min 55 s

Figure 5.14 displays the projections on the real and imaginary axes. As can be seen, the increase in dielectric losses yields a linear shift (increases) in the imaginary part of the frequency while the real part remains almost unchanged. The calculated values coincide with the measured results published in [17], where $\mathrm{Re}(f_r) = 20.372\,\mathrm{GHz}$ for the resonant mode of $TEM_{0,0,27}$.

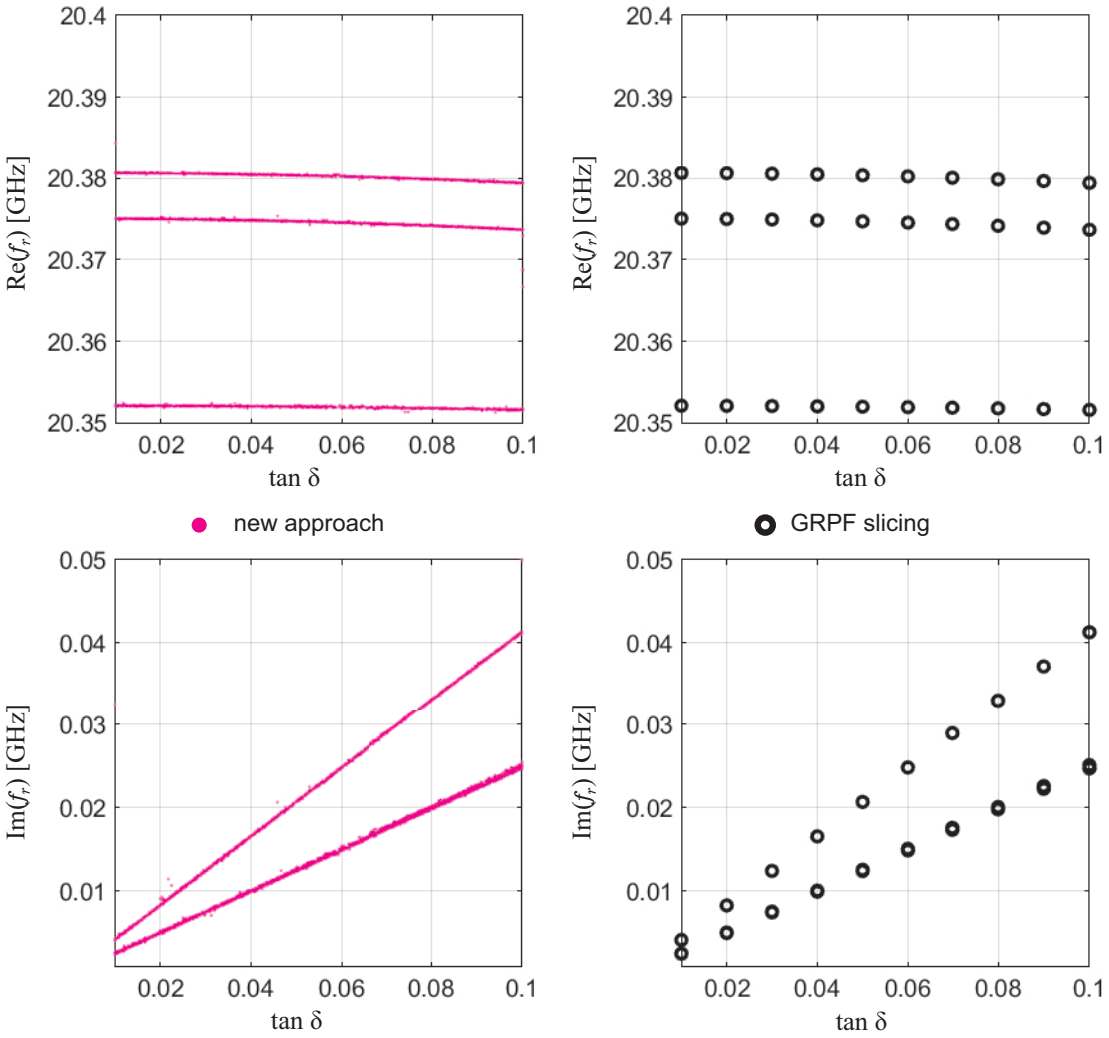

Figure 5.14: The projections on the real axis and the imaginary axis of the obtained characteristics. On the left are results from GRPF in $\mathbb{C} \times \mathbb{R}$ domain; On the right is the ”slicing method” applying the Self-Adaptive GRPF in the ten planes.

5.3 Final comments

Summarizing the obtained numerical results, the proposed GRPF in $\mathbb{C} \times \mathbb{R}$ domain algorithm extends the possibilities of finding zeros of complex functions as a function of an additional parameter. It is an effective and stable global method that gives a full picture of the characteristics in the investigated $\mathbb{C} \times \mathbb{R}$ domain. Nevertheless, it has some limitations. The results provide a general overview of the behavior of the curves rather than a precise solution. While it is an approximate result, it is obtained quickly and efficiently, which can be seen as a benefit if it meets the user's

needs. Furthermore, the results obtained can be improved in further analysis or using other methods, such as tracing, ”slicing,” and local techniques.

An undeniable advantage of the proposed GRPF in $\mathbb{C} \times \mathbb{R}$ domain technique over tracing is that a single employment of the algorithm allows one to obtain several curves representing zero or poles in the investigated area. Moreover, the algorithm can handle multipath characteristics; no a priori knowledge of these curves is needed, such as their number, beginning and end, the distance between them, or potential intersection. Therefore, the user does not have to determine the edge length of the mesh and perform preliminary analyses.

On the other hand, when comparing the new approach with an alternative ”slicing method” in the proposed algorithm, there is no need to control the level of discretization of the t parameter, which reduces the user's involvement and allows for lower numerical costs. The algorithm can find curves $\xi(t)$ parallel to $t = \mathrm{const}$ (in the case of ”slicing,” this is problematic). Additionally, the level of reliability in GRPF in $\mathbb{C} \times \mathbb{R}$ domain technique is more significant compared to ”slicing,” in which the discontinuity of the domain (in the t variable) causes the risk of omitting any of the roots in some plane.

Moreover, the final version of the algorithm largely eliminates ”noise” in the obtained results, i.e., edges not being zeros or poles. Such artificial solutions are usually the effect of edges that are too long in a specific part of the mesh. Thus, the main limitation of the proposed method is the finite number of function calls. Generally speaking, precisely determining all $\xi(t)$ curves in an assumed domain Ω is possible, but N_{max} must go to infinity.

To sum up, the above discussion confirms that generalization of the GRPF algorithm to $\mathbb{C} \times \mathbb{R}$ space allows one to obtain a full picture of the distribution of zeros and poles of the function $F(z, t)$ in the entire analyzed region $\mathbb{C} \times \mathbb{R}$, eliminating the limitations existing in tracing and the ”slicing methods.” So, the last of the theses assumed at the opening of the dissertation is proven.

Chapter 6

Summary

This dissertation underscores the significance of root finding problems for studying the properties of waveguides and resonators applicable in microwave and terahertz technology. The primary objective of the research was to improve existing algorithms and develop new numerical tools to address these issues, and that goal has been accomplished.

The outcome of the work is to create a package of algorithms (free and available on the GitHub platform) that can be utilized to investigate phenomena such as backward wave propagation complex and leaking mods in waveguides. These methods are not limited to this; the proposed methods can also be used to determine resonant frequencies and parameters of materials, as well as the propagation properties of, for example, conformal antennas.

Additionally, solving nonlinear eigenvalue and complex root findings problems is crucial when designing microwave devices like couplers, baluns, phase shifters, circulators, isolators, beam splitters, and multiplexers. Proposed techniques aid in analyzing the radiation and material losses in these devices, providing a comprehensive understanding of their phenomena. This dissertation's findings are expected to contribute significantly to knowledge of the theoretical basis of electromagnetic wave phenomena. Notably, the tools created, specifically nonlinear eigenvalue problem-solving methods, have practical applications beyond electrodynamic problems. The developed tool is flexible and can be used in various scientific fields, including acoustics, control theory, quantum mechanics, and more.

6.1 Conclusion

The research conducted for this dissertation led to the attainment of the following novel outcomes:

- The proposed is a straightforward modification and improvement to the mesh refinement stage of the GRPF algorithm. The modification involves adding new points only at the centers of "candidate edges", which leads to a significant reduction in computation time by minimizing the number of function calls in successive iterations. This approach is sufficient and effective in improving the performance of the algorithm.

- In the GRPF algorithm, a new procedure for detecting and eliminating ”skinny” triangles is implemented. It is accomplished by determining the ratio of the longest edge to the shortest height. Whenever an ill-conditioned triangle is found, a new point is inserted at the midpoint of the longest edge. The numerical differences are minor, but the essence is that the shape of the mesh is more regular and stable.

- The modified and efficient root tracing algorithm involving self-adaptive discretization in CAP condition is developed. The procedure significantly reduces the number of function calls, resulting in faster analysis without affecting accuracy. It can be utilized in design and optimization processes.

- The problem of multipath root tracing has been discussed. Even though it is a limitation of this approach, the proposed modification of the initial condition for the starting triangle expands the possibilities of the algorithm.

- The new algorithm for finding the roots and poles of a complex function with an additional parameter is presented. By generalizing the DCAP condition in the $\mathbb{C}\times\mathbb{R}$ space, it is possible to obtain a full picture of the characteristics in a given domain. The proposed algorithm is straightforward, flexible, and very efficient. Numerical tests have confirmed the effectiveness of the proposed technique.

- The iterative application of the regular GRPF algorithm with sweeping the t parameter, commonly known as the ”slicing method,” is an accurate but inefficient approach for the $\mathbb{C}\times\mathbb{R}$ space. On the other hand, tracing enables a rapid analysis of a single curve with comparable precision. In contrast, the recently introduced GRPF in $\mathbb{C}\times\mathbb{R}$ domain provides a full and fast overview of the roots in the entire domain. To achieve high accuracy, one can increase the number of function calls or employ further analyses or local methods.

Each of the research results (from Chapters 3,4,5) in more detail can also be found in the following publications:

1. M. Jasinski, **S. Dziedziewicz**, M. Jozwicka and P. Kowalczyk, ”An Improvement of Global Complex Roots and Poles Finding Algorithm for Propagation and Radiation Problems,” 2019 13th European Conference on Antennas and Propagation (EuCAP), Krakow, Poland, 2019, pp. 1-5.

2. **S. Dziedziewicz**, M. Warecka, R. Lech and P. Kowalczyk, ”Self-Adaptive Mesh Generator for Global Complex Roots and Poles Finding Algorithm,” in IEEE Transactions on Microwave Theory and Techniques, vol. 71, no. 7, pp. 2854-2863, July 2023.

3. **S. Dziedziewicz**, R. Lech and P. Kowalczyk, ”A Self-Adaptive Complex Root Tracing Algorithm for the Analysis of Propagation and Radiation Problem,” in IEEE Transactions on Antennas and Propagation, vol. 69, no. 8, pp. 5171-5174, Aug. 2021.

4. **S. Dziedziewicz**, M. Warecka, R. Lech and P. Kowalczyk, "Multipath Complex Root Tracing," 2022 24th International Microwave and Radar Conference (MIKON), Gdansk, Poland, 2022, pp. 1-4.

5. **S. Dziedziewicz**, M. Warecka, R. Lech and P. Kowalczyk, "Global Complex Roots and Poles Finding Algorithm in $\mathbb{C} \times \mathbb{R}$ Domain," in IEEE Access, vol. 11, pp. 68809-68817, 2023.

In the author's belief, all the thesis presented at the beginning of the dissertation has been confirmed by the achieved results.

6.2 The way forward

Even though there are stable and efficient algorithms for eigenvalue problems, there are still many opportunities for further research in root finding approaches, particularly in their development. Taking into account the presented limitations of the GRPF family algorithms, the following issues can be considered for further research:

- future developments for regular GRPF and tracing methods;
- exploring development opportunities and challenges in advancing algorithm GRPF in $\mathbb{C} \times \mathbb{R}$ domain;
- improving the characteristics obtained utilizing GRPF in $\mathbb{C} \times \mathbb{R}$ domain employing the post-processing method;
- possible application of the presented algorithms for solving nonlinear eigenvalue problems.

Even though the regular GRPF and tracing methods have already undergone improvements, their continued development is highly valued. Further enhancements of these techniques will involve optimizing the visualization process and implementing the algorithms in additional programming languages, such as Python. Alternatively, hardware optimization using graphics cards is also under consideration.

One of the most promising development opportunities lies in the further advancement of the algorithm GRPF in $\mathbb{C} \times \mathbb{R}$ domain. One issue that needs to be addressed is implementing the self-adaptive mesh generator for regular GRPF to three-dimensional space. Despite this, there is no evidence to suggest that the idea and concept of isophases cannot be utilized in the $\mathbb{C} \times \mathbb{R}$ space. To achieve this, determining the gradient of the function argument for individual edges of a three-dimensional mesh is necessary. Introducing such a solution could positively influence the efficiency of the algorithm. It could lead to the creation of a self-active detection of critical regions of the mesh that require densification. This is especially beneficial in situations where many curves intersect in different planes, e.g., bifurcation characteristics. Nevertheless, it is essential to remember to verify the stability of the mesh permanently and avoid "skinny" triangles, which are still a limitation of the general family of mesh algorithms.

Furthermore, a crucial area needing improvement in GRPF in $\mathbb{C} \times \mathbb{R}$ domain is the ability of the algorithm to characterize individual solutions (e.g., based on a physical interpretation) or determine whether the obtained points are zeros or poles. To address this issue, it is recommended that an additional post-processing method be developed (a final verification technique). One possible strategy involves constructing a chain of tetrahedrons using a set of ”candidate edges” and the generated mesh, somewhat similar to the tracing method. These ”candidate regions” can aid in verifying the CAP condition, enabling the interpretation of the identified curve (especially zeros and poles verification).

In summary, any of the presented algorithms have a wide range of applications in solving nonlinear eigenvalue problems. Specifically, the algorithms proposed in this thesis are dedicated and highly efficient for solving electrodynamic problems. These problems are often related to the propagation of electromagnetic waves and require the investigation of complex and leaky modes. In the field of modern microwave engineering, understanding the relationship between these modes is crucial for analyzing both shielded and open structures. Unfortunately, these issues are usually poorly described or omitted in the literature, especially the conditions of their existence. Generally, these problems boil down to finding the roots of complex parameters, e.g., propagation coefficient. When dealing with huge nonlinear problems, traditional methods based on calculating the determinant of a function may not be effective. Thus, it becomes essential to develop flexible projection algorithms to the appropriate subspace so that it is possible to reduce the problem to a small determinant size (or a large sparse matrix). Nevertheless, in the author's plans, the package of numerical tools developed in this dissertation will also be employed in other areas of modern technologies, such as automatic systems and control theory.

Appendices

Appendix A

Initial mesh generators

This appendix delves into the proposed solutions that try to overcome the limitation of the regular GRPF approach by addressing the issue of initial mesh generation. The research in this area aims to develop a stable mesh algorithm by examining the uneven refinement process in detail. The focus is on identifying the most critical edges on the mesh and finding ways to refine them effectively. To this end, two methods are discussed: a simple mesh refinement algorithm and a rational interpolation approach.

A.1 The simple mesh refinement

The simple algorithm proposed involves determining the importance of each edge based on a standard DCAP approach-based analysis of increasing the function's argument at the mesh nodes. Determining the importance of an edge is crucial to the whole issue. The algorithm is developed based on trial and error methods and is based on seven irregular iterations. This approach helps identify the most critical edges that need to be refined.

A.1.1 Algorithm details

The procedure starts with five points. Iterations first to four will divide all kinds of edges, but this must be done unevenly. The first iteration will be preliminary. In the following steps, the length of the edges will be checked so that the mesh created is regularly distributed. The length check will only apply to single and no quadrants change edges to avoid redundant points. Iterations five and six will assume higher density nodes in the ”candidate regions.” From the seventh iteration, refinement will occur as in the standard GRPF algorithm.

The distribution of quadrants (resulting from the function's argument), as well as the zeros and poles, can be challenging. The following method is used to identify all ”candidate regions,” particularly those without a noticeable phase change. For edges with zero quadrant change, all edges are divided into four equal parts, resulting in three additional points. For edges with a single quadrant change, all edges are divided into three equal parts, resulting in two additional points. Finally, for edges with double quadrant change, all edges are divided into two equal parts,

resulting in one additional point. Initially, the mesh comprises eight edges, so the number of new points added will range from 8 to 24.

In the second iteration of this process, the mesh remains somewhat sparse, with edges ranging from 28 to a maximum of 68. The following step is to give more attention to the areas where a single quadrant change occurs. The following stages will be more sensitive in detecting ”candidate regions.” It means that if an edge with zero or single quadrant change is shorter than a specified length, it will not be reduced further. This method for dividing edges is based on a certain subjective observation. For edges with zero quadrant change over one-tenth of the domain side, half of the longest edges are divided into three equal parts, while the remaining parts are divided into two equal parts. For edges with a single quadrant change longer than one-tenth of the domain side, half of the longest edges are divided into four equal parts, and the remaining parts are divided into three equal parts. Double quadrant change edges are divided into two equal parts regardless of length. Additionally, one adds new points at the centers of the ”skinny” triangle on the entire domain. From this stage until iteration no. 5, ”skinny” triangles will be checked in the entire region and only in the first zone of the ”candidate regions” (see subsection 3.4.2). At this point, the number of new points added will range from 28 to 170.

In the third iteration of this method, edges with different quadrant changes are split. For edges without quadrant change and over one-tenth of the domain size, half of the longest edges are divided into two equal parts, resulting in one additional point. For edges with a single quadrant change and longer than one-tenth of the area size, half of the longest edges are divided into three equal parts, and the remaining parts are divided into two equal parts, resulting in two additional points. Edges with double phase change are divided into two equal parts, resulting in one additional point, regardless of length. Lastly, new points are added in the centers of the ”skinny” triangle of the entire domain, resulting in one additional point.

In the fourth step, the third step is repeated but with fewer new points. The process involves dividing long edges with no quadrant change into smaller parts that are one-tenth of the side of the domain. Only half of the longest edges are divided into two equal parts, adding one extra point. For edges with a single quadrant change that is longer than one-tenth of the side of the domain, only half of the longest edges are divided into two equal parts, which adds one extra point. Edges with double quadrant changes are divided into two equal parts, adding one extra point. Lastly, new points are added in the middle of ”skinny” triangles across the entire domain, which adds one extra point.

The first four steps of this process aim to detect all ”candidate regions” with some resolution. Subsequent stages focus on the density of regions around the ”candidate edges,” specifically those with edges that are a double quadrant change. A specific procedure is followed to add new points. All ”double-quadrant” edges are divided into four equal parts, which gives three extra points. New points are added at the centers of all triangles from the second ”candidate regions” zone, giving one more point. Additionally, new points are added at the centers of all ”skinny” triangles on the entire domain, which gives another point. Lastly, the edges of the first ”candidate regions” zone

are divided in half, which gives one more point.

The following steps are taken in the sixth iteration to refine the mesh further:

- All edges that have a double quadrant change are divided into three equal parts, resulting in the addition of two new points.
- A new point is introduced at the center of each triangle in the first ”candidate regions” zone. Then, new points are added at the centers of all ”skinny” triangles across the entire domain.
- The edges of the first ”candidate regions” zone are divided in half, resulting in the addition of one new point.

Table A.1: The number of points added by refinement of the particular regions in iterations from first to fourth. A ”skinny” triangle involves the additional points in the centroid; in other cases, the new points are added by splitting the edges.

iteration	longer than one-tenth of the domain side				each	
	$\lvert\Delta Q_p\rvert = 0$		$\lvert\Delta Q_p\rvert = 1$		$\lvert\Delta Q_p\rvert = 2$	**”skinny”**
	longer	shorter	longer	shorter		
1	3	3	2	2	1	0
2	2	1	3	2	1	1
3	1	0	2	1	1	1
4	1	0	1	0	1	1

Table A.2: The number of points added by refinement of the particular regions in iterations from fifth to seventh. In the case of a $|\Delta Q_p| = 2$, this involves points on the edges; in other cases, the new point is in the centroid (”skinny,” first zone).

iteration	$\lvert\Delta Q_p\rvert = 2$	**”skinny”**	**first zone**	**boundary in first zone**
5	3	1	1	1
6	2	1	1	1
7	1	1	0	1

A summary of the algorithm for each iteration is provided in Tables A.1 and A.2. These steps help to create a more refined and specific mesh. The standard refinement GRPF method is used from the seventh iteration onwards. It involves dividing all edges with a double quadrant change into two equal parts, resulting in an additional point. Furthermore, new points are added to the centers of the ”skinny” triangle in the first ”candidate regions” zone, resulting in one additional point. Finally, the edges of the first ”candidate regions” zone are further divided in half, resulting in another point.

A.1.2 Numerical example

The numerical test is done for the identical example discussed in the subsections 3.6.4 and 3.3. The rational function from (3.12) is analyzed in the following domain:

$$\Omega = \{z \in \mathbb{C} : -1 < \mathrm{Re}(z) < 1 \wedge -1 < \mathrm{Im}(z) < 1\}$$

Figure A.1 shows the successive stages in forming a mesh for a function with an ”island” of size $\epsilon = 0.05$. As can be seen, the effect of the last sixth iteration is the island’s detection. In this case, the method worked successfully. However, the mesh contains too many redundant points, i.e., the nodes in the areas with slight variation in function.

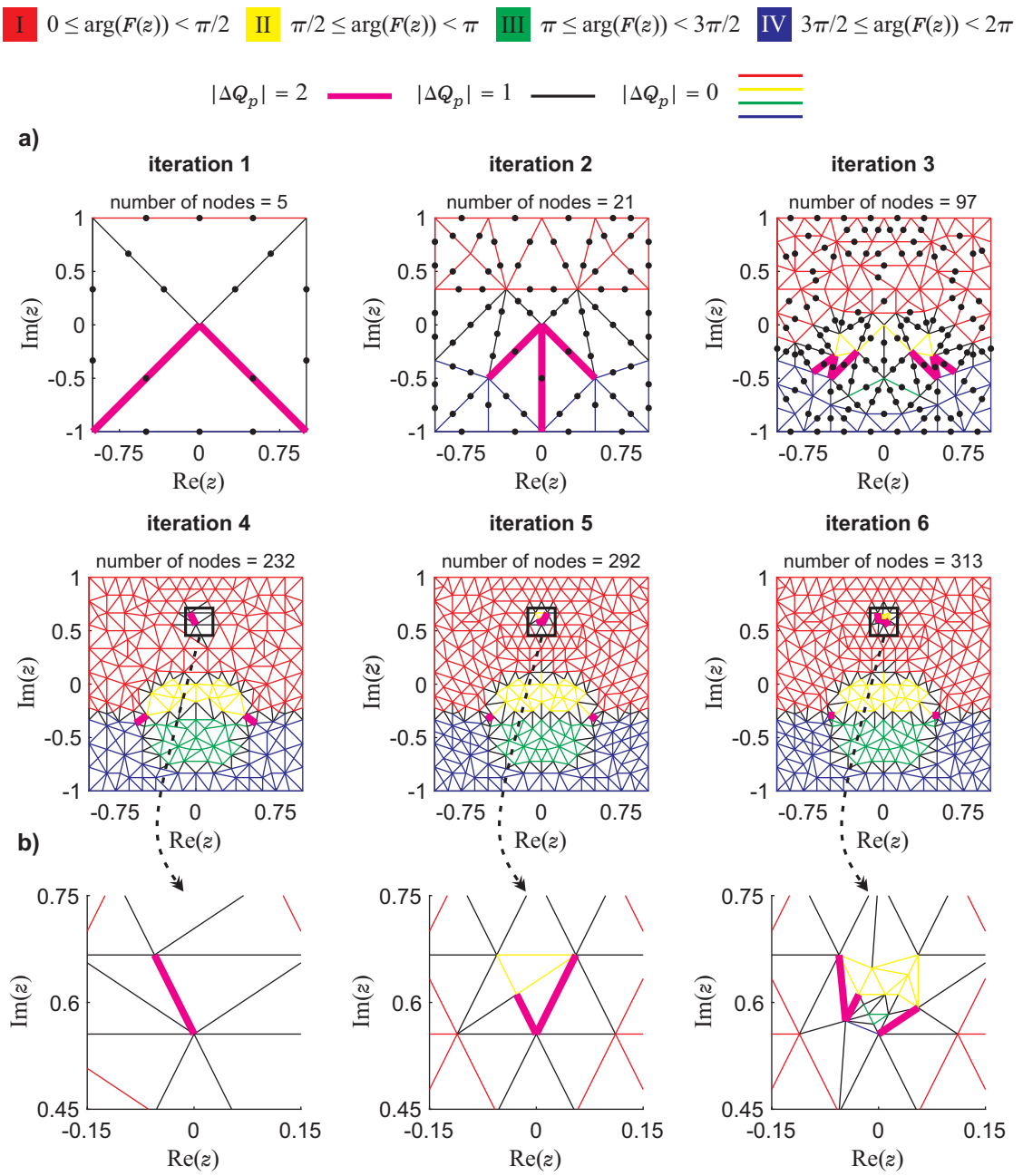

Figure A.1: The iterative process of simple mesh refinements solution, for function from (3.11) with $\epsilon = 0.05$ a) first six example stages, b) zoom of ”island” locations, respectively, in the demonstrated iterations.

A.2 The local rational interpolation

The next algorithm is based on rational interpolation, which combines classical interpolation with GRPF technique. This approach involves finding the zeros of complex functions based on interpolation methods. Given a set of points (e.g., function values evaluated in nodes on the mesh), one can use different functions to interpolate the function between them. For instance, it can use polynomials of a higher order than the number of points or various trigonometric functions. If the function has singularities, it can use a set of rational functions with different variants of the order of the numerator and denominator. So, regarding the zeros and poles finding issue, rational interpolation seems to be the most appropriate. Rational interpolation is a numerical method involving estimating a function using a ratio of two polynomials. Generalizing such approaches creates a rational function, where the order of the polynomial (in numerator and denominator) is related to the current number of nodes, according to the following formula:

$$F(z) = \frac{a_0 + a_1 z + a_2 z^2 + ... + a_n z^n}{b_0 + b_1 z + b_2 z^2 + ... + b_m b^m} \tag{A.1}$$

where:

$z \in \mathbb{C}$,

a_n, b_m - are unknown coefficients,

$n + m$ - order of the rational (number of points).

The goal of rational interpolation is to find a set of coefficients of a rational function that accurately represents the original function being interpolated while minimizing the error between the two. The resulting function can be used to make predictions or to generate a smooth curve that fits the data see Figure A.2.

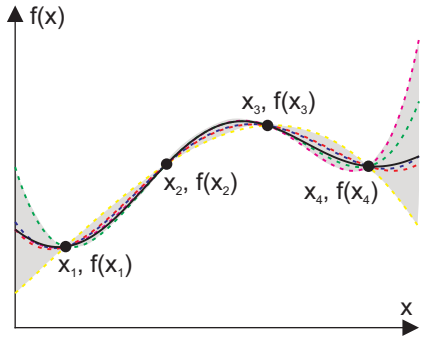

Figure A.2: An example of a real function $f(x)$, interpolated using polynomials based on four points.

A.2.1 Algorithm details

This subsection briefly introduces an adaptive approach for generating an initial mesh. The proposed method utilizes local rational interpolation. The algorithm implemented in the present solution consists of three primary stages: *Mesh nesting*, *Validation of the accuracy of the interpolation*, and the *Selection of a densified edge*.

Mesh nesting

The adaptive initial mesh generation based on rational interpolation is an iterative-recursive process. At the beginning of each series, four points from the domain are selected (see Figure A.3). Next, the values of the functions at points no. $1, 2, 3, 4$ are evaluated. If it is impossible to create a correct interpolation from these points (see *Validation of the accuracy of the rational interpolation*), then points no. $5, 6, 7, 8, 9$ will be used in successive iterations. They will be grouped into smaller subdomains, each consisting of four points, as shown in Figure A.3.

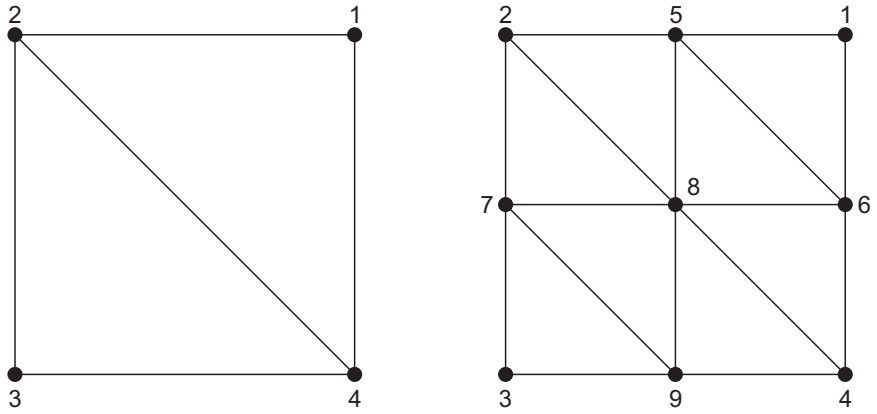

Figure A.3: On the left, the initial mesh from points $(1, 2, 3, 4)$. On the right, smaller subdomains are formulated by dividing the edges in half based on possible combinations of $(1, 5, 8, 6)$,$(5, 2, 7, 8)$,$(8, 7, 3, 9)$,$(6, 8, 9, 4)$.

Validation of the accuracy of the rational interpolation

In order to ensure an appropriate interpolation and validation, it is necessary to divide the nodes into ”training” and ”validation” points. The function value in ”training” nodes is known (evaluated directly) versus the function value in ”validation” nodes, which is estimated using the interpolation. Two points are selected for the ”validation” set in each iteration, and the interpolating rational is made from the remaining points. All possible combinations for four points are presented in Table A.3; the number of combinations results in the rational degree and the number of ”training” points. The mesh can consist of 9 points, so there are 36 ”training” sets (7th-degree rationals).
Upon analyzing the case presented in the table above, it can be inferred that for each set of points (totaling six), two potential interpolating rationals exist, considering the number of ”training” points. There are:

- the first type of rational: $\frac{a_0}{b_1 x + b_0}$
- the second type of rational: $\frac{a_1 x + a_0}{b_0}$.

The coefficients a_0, a_1, b_0, b_1 have different values for each (i.e., ”training” and ”validation”) set of points. This results in 12 possible rational functions (two rationals in six configurations). Consequently, for each point in the ”validation” set, two possible estimated values of the function exist, one from each rational. Each rational type has 12 ”validation” points, encompassing two possible

points in six configurations. Lastly, it is worth noting that each mentioned point $(1, 2, 3, 4)$ corresponds to three known points and three ”validation” points.

Table A.3: The example of combinations points for sets (”training” and ”validation”) is presented to clarify the idea.

set of points	**”training” set**	**”validation” set**
$\mathrm{no.1}$	$1, 2$	$3, 4$
$\mathrm{no.2}$	$1, 3$	$2, 4$
$\mathrm{no.3}$	$1, 4$	$2, 3$
$\mathrm{no.4}$	$2, 3$	$1, 4$
$\mathrm{no.5}$	$2, 4$	$1, 3$
$\mathrm{no.6}$	$3, 4$	$1, 2$

In a nutshell, six interpolate function values can be determined for each point. It should generally be the same value as the directly evaluated one. The algorithm aims to identify the most accurate rational type from $2\mathrm{nd}$ to $7\mathrm{th}$ degree (depending on the number of points) and then calculates its coefficients by averaging their combinations. For instance, Table A.4 shows the function values for particular points of one of the rational, where:

f_k^i - interpolated value of the function for k and i indices, respectively,

k - corresponds to the node number,

i - corresponds to the ”training” set number (rational coefficients),

F_k - true function value evaluated directly in k node.

Table A.4: An example of function values for each point in ”training” set.

	”training” set						**function**
	$\mathrm{no.1}$	$\mathrm{no.2}$	$\mathrm{no.3}$	$\mathrm{no.4}$	$\mathrm{no.5}$	$\mathrm{no.6}$	**value**
node $\mathrm{no.1}$	f_1^1	f_1^2	f_1^3	f_1^4	f_1^5	f_1^6	F_1
node $\mathrm{no.2}$	f_2^1	f_2^2	f_2^3	f_2^4	f_2^5	f_2^6	F_2
node $\mathrm{no.3}$	f_3^1	f_3^2	f_3^3	f_3^4	f_3^5	f_3^6	F_3
node $\mathrm{no.4}$	f_4^1	f_4^2	f_4^3	f_4^4	f_4^5	f_4^6	F_4

Due to it being an interpolation, the estimated values of the function have some error and are not equal to each other $f_k^1 \neq f_k^2 \neq f_k^3 \neq f_k^4 \neq f_k^5 \neq f_k^6 \neq F_k$. For each k, i indices, the relative error can be determined $e_k^i = \frac{|f_k^i - F_k|}{|F_k|}$. It creates the error matrix $\mathbf{E}_{rr}$, presented in Table A.5 for a specific rational type, where:

e_k^i - the error for individual k and i indices, respectively,

k - corresponds to the point number,

i - corresponds to the ”training” set number (rational coefficients),

E_x - mean error for each k node.

Table A.5: An example of interpolation error for each point in ”training” set.

	”training” set						**mean**
	no.1	no.2	no.3	no.4	no.5	no.6	**value**
node no.1	0	0	0	e_1^4	e_1^5	e_1^6	E_1
node no.2	0	e_2^2	e_2^3	0	0	e_2^6	E_2
node no.3	e_3^1	0	e_1^4	0	e_3^5	0	E_3
node no.4	e_4^1	e_4^2	0	e_4^4	0	0	E_4

In order to determine the global interpolation error for a specific rational, the errors for each point should be averaged. This parameter indicates how accurate the method is and determines which rational most reliably represents the analyze function (the averaged error should be as small as possible). Additionally, the algorithm assumes the general condition for the correctness of the interpolation defined in percent[1]. If none of the interpolated rationals meet this condition, adding an extra point on the mesh (out of $5, 6, 7, 8, 9$) is necessary. The procedure of validating the rational interpolation is repeated with more points in ”training”, up to seven. If the mean interpolation error is still more significant than accepted, the domain is recursively split according to the abovementioned rule (see *Mesh nesting*).

Selection of a densified edge

The way of selecting the extra point from among $(5, 6, 7, 8, 9)$ is not arbitrary. Instead, the interpolate value of the function is estimated based on the different coefficients of the rationals. Ideally, for a particular node, the value of the function should be equal regardless of the chosen coefficients. The point at which the value of the function differs most is critical. It informs on the possible variability of the function in that region. Hence, the point added to the mesh can be chosen by determining the most significant standard deviation from the calculated mean error.

A.2.2 Numerical example

In order to illustrate the functionality of the algorithm, one of the most challenging examples, specifically the graphene transmission line, is examined. This investigation is outlined in subsection 2.3.3 and is also recalculated in subsection 3.6.4 for further clarity. In this problem, the normalized prop-

[1] This value depends on the investigated problems (function, domain, etc.) but is usually about one percent.

agation coefficient $z = \gamma$ for TM modes is determined in the following range:

$$\Omega = \{z \in \mathbb{C} : -100 < \mathrm{Re}(z) < 400 \wedge -100 < \mathrm{Im}(z) < 400\}$$

The investigated domain has eight single zeros, two poles of the second order, and an additional single zero and pole ("island"). Figure A.4 shows the examples of stages in forming a mesh.

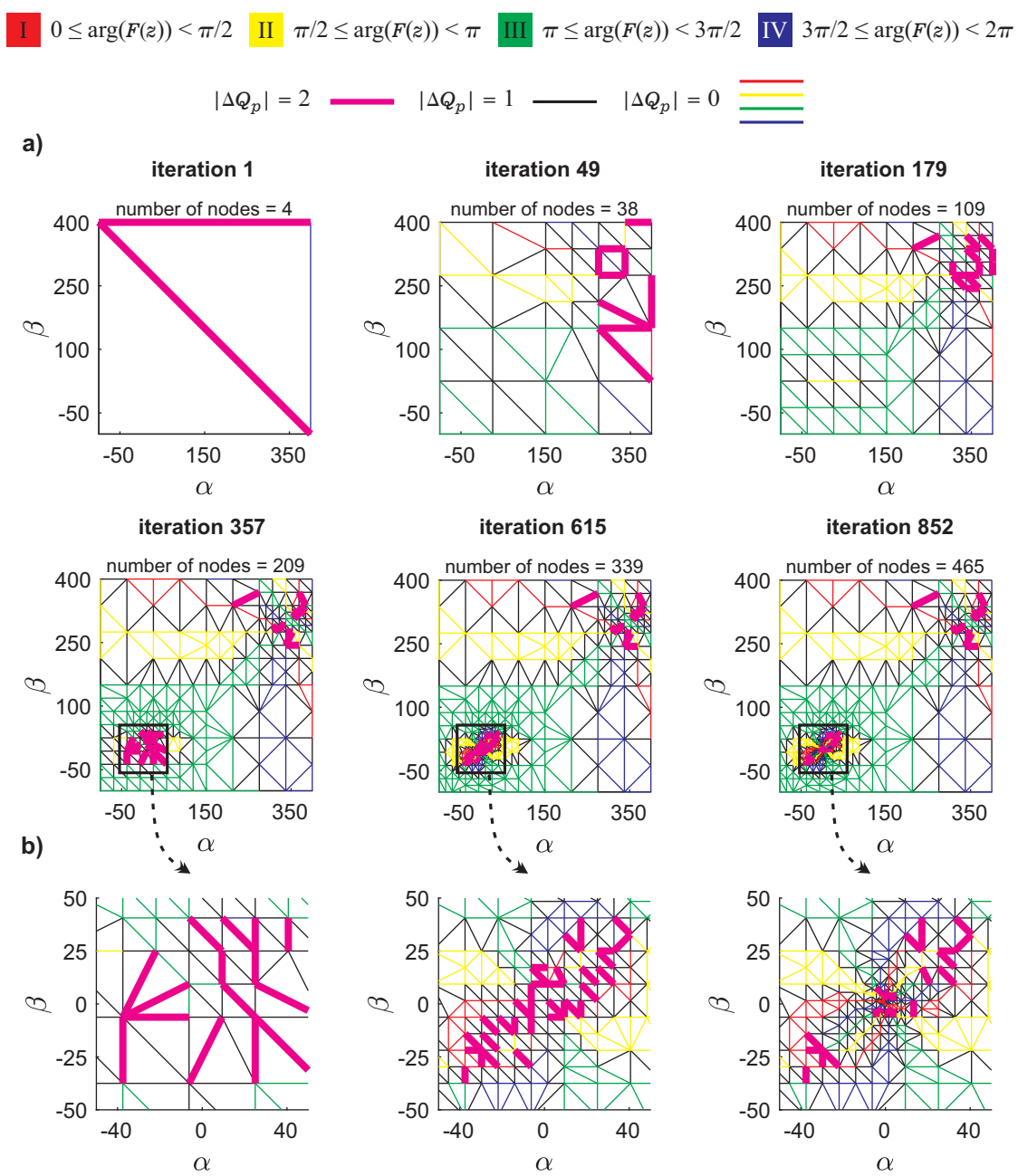

Figure A.4: The iterative process of local rational interpolation technique, for level accuracy 0.5% (mean interpolation error), a) first-sixth example stages, b) zoom of "island" locations, respectively, in the demonstrated iterations.

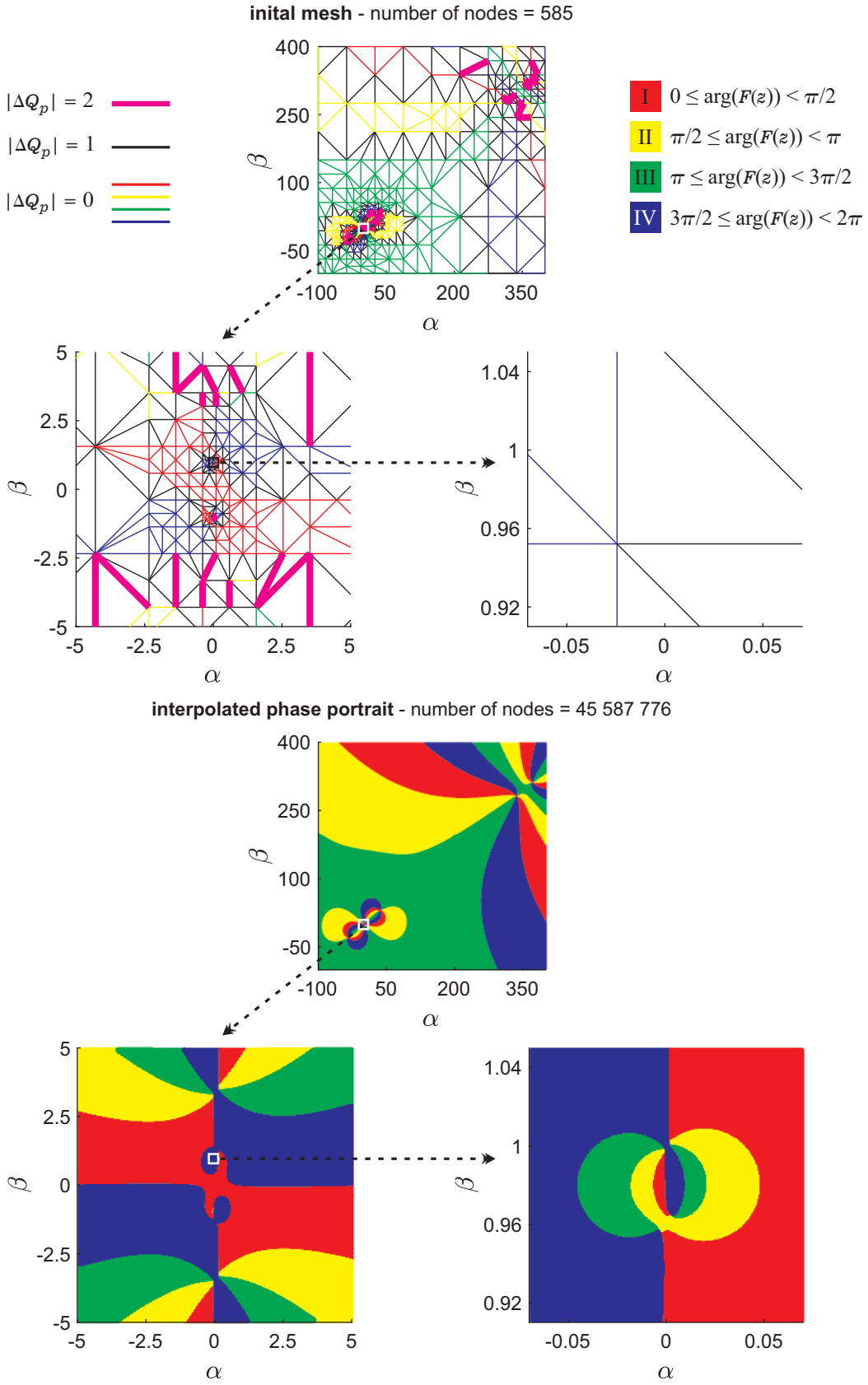

Figure A.5: The final initial meshes that can be utilized to find all zeros and poles (including ”island”) in the investigated domain Ω and the interpolated phase portrait based on the calculated rational function. A double magnification of the critical region of the ”island” is shown at the bottom and right.

Similar to the previous example in this appendix, the mesh displayed only represents the initial shape (not the final refinement). The analysis resulted in a mesh consisting of $1\,069$ nodes created within 586 iterations, illustrated in Figure A.4. Moreover, calculated coefficients of the rational function are used to interpolate the function values at additional points. In each rectangular grid, 250 regular points are added, resulting in a total number of points: $45\,587\,776$. Based on these estimated values, a phase portrait is drawn. However, it is important to note that these values are approximate and are obtained quickly due to the interpolation technique used rather than rational calculation (see Figure A.5). By implementing an interpolation technique to the initial mesh, the proposed algorithm successfully identified four additional zeros and two additional poles in the assumed domain. In comparison (with the results in Figure 3.33), detecting these regions through a regular mesh for GRPF would require a large number of function calls (in this case, below 600 is enough to find the island).

A.3 Final comments

The effectiveness of the proposed techniques is mostly limited to specific function classes. In the first presented algorithm (simple refinament method), no theoretical or mathematical evidence supports the described scheme. The approach has many parameters and needs to be more flexible. It was discovered that the phase-quadrant approach generated insufficient information about the function's variability and character. As a result, the technique should take additional information, such as correlations between points.

On the other hand, in the interpolation approach, the degree of the rationals plays a crucial role in the algorithm's performance, mainly when dealing with rapidly varying functions and exploring the large domain Ω. However, due to the finite number of mesh nodes used for "training" and "validation," this method cannot guarantee the discovery of all zeros and poles. Another drawback is the issue of procedure convergence, which can lead to a decrease in efficiency by causing the algorithm to loop repeatedly. The recursion level can be high, resulting in increased computational complexity and longer analysis times, especially for simple functions where the time required to execute a single iteration of the interpolation procedure exceeds that of evaluating a single function value. Even though only a few nodes are added in each iteration, the overall impact can be significant.

Overall, the research in this appendix identified several promising algorithms that can be used to refine the initial mesh effectively. Exploring each algorithm's strengths and weaknesses helped develop a new self-adaptive approach to provide a stable and reliable mesh for further analysis and simulation.

Bibliography

[1] J. Nilsson and S. Riedel, *Electric Circuits*. Pearson/Prentice Hall, 2008, ISBN: 9780131989252.

[2] F. Najm, *Circuit Simulation* (IEEE Press). Wiley, 2010, ISBN: 9780470538715.

[3] J. Lange, "Interdigitated stripline quadrature hybrid (correspondence)," *IEEE Transactions on Microwave Theory and Techniques*, vol. 17, no. 12, pp. 1150–1151, 1969. DOI: 10.1109/TMTT.1969.1127115.

[4] D. Pozar, *Microwave Engineering, 4th Edition*. Wiley, 2011, ISBN: 9781118213636.

[5] C. Balanis, *Advanced Engineering Electromagnetics, 2nd Edition*. Wiley, 2012, ISBN: 9781118213483.

[6] D. Pozar, *Microwave and RF Design of Wireless Systems*. Wiley, 2000, ISBN: 9780471322825.

[7] R. Collin, *Foundations for Microwave Engineering* (IEEE Press Series on Electromagnetic Wave Theory). Wiley, 2001, ISBN: 9780780360310.

[8] M. Mrozowski, *Guided electromagnetic waves: properties and analysis*. Research Studies Press, Wiley, New York, 1997.

[9] R. F. Harrington, *Time-Harmonic Electromagnetic Fields*. John Wiley & Sons, 2001, ISBN: SBN: 978-0-471-20806-8.

[10] D. Cook, *The Theory of the Electromagnetic Field* (Dover Books on Physics Series). Dover Publications, 2002, ISBN: 9780486425672.

[11] M. Sadiku, *Computational Electromagnetics with MATLAB*. CRC Press, 2018, ISBN: 9781138558151.

[12] A. Kirsch and F. Hettlich, *The Mathematical Theory of Time-Harmonic Maxwell's Equations: Expansion-, Integral-, and Variational Methods* (Applied Mathematical Sciences). Springer International Publishing, 2014, ISBN: 9783319110851.

[13] J. W. Brown and R. V. Churchill, *Complex Variables and Applications*. McGraw-Hill, New York, NY, USA, 2009.

[14] H. Priestley, *Introduction to Complex Analysis*. OUP Oxford, 2003, ISBN: 9780191583339.

[15] P. K. Tien, R. J. Martin, and S. Riva-Sanseverino, "Novel metal-clad optical components and method of isolating high-index substrates for forming integrated optical circuits," *Applied Physics Letters*, vol. 27, no. 4, pp. 251–253, Sep. 2008, ISSN: 0003-6951. DOI: 10.1063/1.88412.

[16] T. Habashy, S. Ali, and J. Kong, “Input impedance and radiation pattern of cylindrical-rectangular and wraparound microstrip antennas,” *IEEE Transactions on Antennas and Propagation*, vol. 38, no. 5, pp. 722–731, 1990. DOI: 10.1109/8.53500.

[17] 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 Transactions on Microwave Theory and Techniques*, vol. 70, no. 5, pp. 2732–2742, 2022. DOI: 10.1109/TMTT.2022.3152511.

[18] D. Djurdjevic, T. Benson, P. Sewell, and A. Vukovic, “Fast and accurate analysis of 3-d curved optical waveguide couplers,” *Journal of Lightwave Technology*, vol. 22, no. 10, pp. 2333–2340, 2004. DOI: 10.1109/JLT.2004.833280.

[19] R. Schwindt and C. Nguyen, “Spectral domain analysis of three symmetric coupled lines and application to a new bandpass filter,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 42, no. 7, pp. 1183–1189, 1994. DOI: 10.1109/22.299755.

[20] E. Anemogiannis and E. Glytsis, “Multilayer waveguides: Efficient numerical analysis of general structures,” *Journal of Lightwave Technology*, vol. 10, no. 10, pp. 1344–1351, 1992. DOI: 10.1109/50.166774.

[21] E. Yamashita, *Analysis Methods for Electromagnetic Wave Problems* (Analysis Methods for Electromagnetic Wave Problems). Artech House, 1990, ISBN: 9780890063644.

[22] 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 Transactions on Antennas and Propagation*, vol. 66, no. 6, pp. 3237–3240, 2018. DOI: 10.1109/TAP.2018.2819903.

[23] R. Gesche and N. Lochel, “Scattering by a lossy dielectric cylinder in a rectangular waveguide,” *IEEE Trans. Microw. Theory Tech.*, vol. 36, no. 1, pp. 137–144, Jan. 1988.

[24] Z. Sipus, P.-S. Kildal, R. Leijon, and M. Johansson, “An algorithm for calculating green’s functions of planar, circular cylindrical, and spherical multilayer substrates,” *Applied Computational Electromagnetics Society Journal*, vol. 13, pp. 243–54, Jan. 1998.

[25] W. Yu and R. Mittra, “A conformal fdtd software package modeling antennas and microstrip circuit components,” *IEEE Antennas and Propagation Magazine*, vol. 42, no. 5, pp. 28–39, 2000. DOI: 10.1109/74.883505.

[26] R. Lech, “Calculation of resonance in planar and cylindrical microstrip structures using a hybrid technique,” *IEEE Transactions on Antennas and Propagation*, vol. 66, no. 1, pp. 497–500, 2018. DOI: 10.1109/TAP.2017.2768578.

[27] D. Arena, M. Ludovico, G. Manara, and A. Monorchio, “Analysis of waveguide discontinuities using edge elements in a hybrid mode matching/finite elements approach,” *IEEE Microwave and Wireless Components Letters*, vol. 11, no. 9, pp. 379–381, 2001. DOI: 10.1109/7260.950767.

[28] A. Kusiek, R. Lech, and J. Mazur, “A new hybrid method for analysis of scattering from arbitrary configuration of cylindrical objects,” *IEEE Transactions on Antennas and Propagation*, vol. 56, no. 6, pp. 1725–1733, Jun. 2008. DOI: 10.1109/TAP.2008.923339.

[29] M. Mrozowski and J. Mazur, “Matrix theory approach to complex waves (in shielded lossless guides),” *IEEE Transactions on Microwave Theory and Techniques*, vol. 40, no. 4, pp. 781–785, 1992. DOI: 10.1109/22.127534.

[30] J. Dobrowolski and W. Ostrowski, *Computer-aided Analysis, Modeling, and Design of Microwave Networks: The Wave Approach* (Artech House microwave library). Artech House, 1996, ISBN: 9780890066690.

[31] S. Güttel and F. Tisseur, “The nonlinear eigenvalue problem,” *Acta Numerica*, vol. 26, pp. 1–94, May 2017. DOI: 10.1017/S0962492917000034.

[32] O. Zienkiewicz, R. Taylor, and J. Zhu, *The Finite Element Method: Its Basis and Fundamentals*. Elsevier Science, 2005, ISBN: 9780080472775.

[33] J.-M. Jin, *The finite element method in electromagnetics*. John Wiley & Sons, 2015.

[34] G. Pelosi, R. Coccioli, and S. Selleri, *Quick Finite Elements for Electromagnetic Waves* (Artech House electromagnetic analysis series). Artech House, 2009, ISBN: 9781596933453.

[35] F. A. Fernandez and Y. Lu, *Microwave and Optical Waveguide Analysis by the Finite Element Method*. USA: John Wiley & Sons, Inc., 1996, ISBN: 0471957488.

[36] A. Taflove and S. C. Hagness, *Computational electrodynamics: the finite-difference time-domain method*, 3nd. Artech House, Boston, Jun. 2005.

[37] M. Beaubien and A. Wexler, “An accurate finite-difference method for higher order waveguide modes,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 16, no. 12, pp. 1007–1017, 1968. DOI: 10.1109/TMTT.1968.1126856.

[38] P. Kowalczyk, M. Wiktor, and M. Mrozowski, “Efficient finite difference analysis of microstructured optical fibers,” *Opt. Express*, vol. 13, no. 25, pp. 10 349–10 359, Dec. 2005. DOI: 10.1364/OPEX.13.010349.

[39] R. Harrington, “Matrix methods for field problems,” *Proceedings of the IEEE*, vol. 55, no. 2, pp. 136–149, 1967. DOI: 10.1109/PROC.1967.5433.

[40] *PathWave Advanced Design System (ADS)*, https://keysight.com/.

[41] *High Frequency Structure Simulator (HFSS)*, http://ansys.com.

[42] *Quick Wave 3D (QWED)*, https://qwed.com.pl.

[43] *InventSim*, https://inventsim.com.

[44] M. Warecka, G. Fotyga, P. Kowalczyk, *et al.*, “Modal fem analysis of ferrite resonant structures,” *IEEE Microwave and Wireless Components Letters*, vol. 32, no. 7, pp. 819–822, 2022. DOI: 10.1109/LMWC.2022.3154532.

[45] W. Ford, “Large sparse eigenvalue problems,” in *Numerical Linear Algebra with Applications*, Boston: Academic Press, 2015, pp. 533–549, ISBN: 978-0-12-394435-1. DOI: 10.1016/B978-0-12-394435-1.00022-3.

[46] W.-J. Beyn, “An integral method for solving nonlinear eigenvalue problems,” *Linear Algebra and its Applications*, vol. 436, no. 10, pp. 3839–3863, 2012, ISSN: 0024-3795. DOI: 10.1016/j.laa.2011.03.030.

[47] R. Lech, P. Kowalczyk, and A. Kusiek, “Scattering from a cylindrical object of arbitrary cross section with the use of field matching method,” *IEEE Transactions on Antennas and Propagation*, vol. 64, no. 11, pp. 4883–4887, Jul. 2016. DOI: 10.1109/TAP.2016.2594841.

[48] M. Warecka, P. Kowalczyk, and R. Lech, “Evaluation of propagation parameters of open guiding structures with the use of complex root finding algorithms,” in *2017 IEEE MTT-S International Microwave Workshop Series on Advanced Materials and Processes for RF and THz Applications (IMWS-AMP)*, 2017, pp. 1–3. DOI: 10.1109/IMWS-AMP.2017.8247347.

[49] L. M. Delves and J. N. Lyness, “A numerical method for locating the zeros of an analytic function,” *Mathematics of computation*, vol. 21, pp. 543–560, 1967. DOI: 10.1090/S0025-5718-1967-0228165-4.

[50] M. Meylan and L. Gross, “A parallel algorithm to find the zeros of a complex analytic function,” *ANZIAM J.*, vol. 44 (E), E216–E234, 2003. DOI: 10.21914/anziamj.v44i0.495.

[51] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in Fortran 77: The Art of Scientific Computing*, 2nd ed. Cambridge University Press, England, 1992, ISBN: 052143064X.

[52] M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions, With Formulas, Graphs and Mathematical Tables*. New York, NY, USA, 1972, ISBN: 9780318117300.

[53] A. P. Austin, P. Kravanja, and L. N. Trefethen, “Numerical algorithms based on analytic function values at roots of unity,” *SIAM Journal on Numerical Analysis*, vol. 52, no. 4, pp. 1795–1821, 2014. DOI: 10.1137/130931035.

[54] Y. Nakatsukasa, O. Sète, and L. N.Trefethen, “The AAA algorithm for rational approximation,” *SIAM Journal on Scientific Computing*, vol. 40, no. 3, A1494–A1522, Jan. 2018. DOI: 10.1137/16m1106122.

[55] P. Kowalczyk, “Complex root finding algorithm based on delaunay triangulation,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 41, pp. 1–13, 3 Jun. 2015. DOI: 10.1145/2699457.

[56] G. Zouros, “Ccomp: An efficient algorithm for complex roots computation of determinantal equations,” *Computer Physics Communications*, vol. 222, pp. 339–250, 2018. DOI: 10.1016/j.cpc.2017.09.023.

[57] 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, 2018. DOI: 10.1109/TMTT.2018.2880778.

[58] R. Lech, W. Marynowski, and A. Kusiek, “An analysis of elliptical-rectangular multipatch structure on dielectric-coated confocal and nonconfocal elliptic cylinders,” *IEEE Transactions on Antennas and Propagation*, vol. 63, no. 1, pp. 97–105, 2015. DOI: 10.1109/TAP.2014.2365573.

[59] D. Khedrouche, F. Bouttout, T. Fortaki, and A. Benghalia, “Spectral-domain analysis of multilayer cylindrical–rectangular microstrip antennas,” *Engineering Analysis with Boundary Elements*, vol. 33, no. 7, pp. 930–939, 2009, ISSN: 0955-7997. DOI: 10.1016/j.enganabound.2009.02.002.

[60] J. J. Michalski and P. Kowalczyk, “Efficient and systematic solution of real and complex eigenvalue problems employing simplex chain vertices searching procedure,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 59, no. 9, pp. 2197–2205, 2011. DOI: 10.1109/TMTT.2011.2160277.

[61] P. Kowalczyk and W. Marynowski, “Efficient complex root tracing algorithm for propagation and radiation problems,” *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 5, pp. 2540–2546, Jun. 2017. DOI: 10.1109/TAP.2017.2684198.

[62] P. Kowalczyk, L. Kulas, and M. Mrozowski, “Analysis of microstructured optical fibers using compact macromodels,” *Opt. Express*, vol. 19, no. 20, pp. 19 354–19 364, Sep. 2011. DOI: 10.1364/OE.19.019354.

[63] P. Kowalczyk and M. Mrozowski, “A new conformal radiation boundary condition for high accuracy finite difference analysis of open waveguides,” *Opt. Express*, vol. 15, no. 20, pp. 12 605–12 618, Oct. 2007. DOI: 10.1364/OE.15.012605.

[64] M. Warecka, R. Lech, and P. Kowalczyk, “Hybrid analysis of structures composed of axially symmetric objects,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 68, no. 11, pp. 4528–4535, 2020. DOI: 10.1109/TMTT.2020.3014944.

[65] *CST Studio Suite*, https://cst.com/.

[66] H. Wang, C. Wu, T. Jia, and D. Li, “A phase unwrapping method based on sparse depth for depth perception,” *Optics and Lasers in Engineering*, vol. 109, pp. 60–67, 2018, ISSN: 0143-8166. DOI: 10.1016/j.optlaseng.2018.05.001.

[67] P. Y. Chen and Y. Sivan, “Robust location of optical fiber modes via the argument principle method,” *Computer Physics Communications*, vol. 214, pp. 105–116, 2017, ISSN: 0010-4655. DOI: 10.1016/j.cpc.2016.11.009.

[68] E. P. Popov, *The Dynamics of Automatic Control Systems*. Pergamon Press, Oxford, 1962, ISBN: 978-0-08-009588-2. DOI: 10.1016/B978-0-08-009588-2.50001-1.

[69] D. Trofimowicz and T. P. Stefański, "Testing stability of digital filters using optimization methods with phase analysis," *Energies*, vol. 14, no. 5, 2021, ISSN: 1996-1073. DOI: 10.3390/en14051488.

[70] 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. Express*, vol. 30, no. 2, pp. 3113–3124, Jan. 2022. DOI: 10.1364/OE.449147.

[71] F. B. Jensen, W. A. Kuperman, M. B. Porter, and H. Schmidt, *Computational Ocean Acoustics*, 2nd. Springer Publishing Company, Incorporated, 2011, ISBN: 1441986774. DOI: 10.1007/978-1-4419-8678-8.

[72] M. K. Schmidt, M. C. O'Brien, M. J. Steel, and C. G. Poulton, "ARRAW: Anti-resonant reflecting acoustic waveguides," *New Journal of Physics*, vol. 22, no. 5, p. 053 011, May 2020. DOI: 10.1088/1367-2630/ab7d79.

[73] G. Tytko, "Locating defects in conductive materials using the eddy current model of the filamentary coil," *Journal of Nondestructive Evaluation*, vol. 40, Sep. 2021. DOI: 10.1007/s10921-021-00798-2.

[74] E. C. G. Sudarshan, C. B. Chiu, and V. Gorini, "Decaying states as complex energy eigenvectors in generalized quantum mechanics," *Phys. Rev. D*, vol. 18, pp. 2914–2929, 8 Oct. 1978. DOI: 10.1103/PhysRevD.18.2914.

[75] F. M. Fernández, *Introduction to Perturbation Theory in Quantum Mechanics*. RC Press, Boca Raton, 2001, ISBN: 9780849318771. DOI: 10.1201/9781420039641.

[76] R. e. a. Aaij, "Study of the lineshape of the $\chi_{c1}(3872)$ state," *Phys. Rev. D*, vol. 102, p. 092 005, 9 Nov. 2020. DOI: 10.1103/PhysRevD.102.092005.

[77] I. Chekhovskoy, S. Medvedev, I. Vaseva, E. Sedov, and M. Fedoruk, "Introducing phase jump tracking - a fast method for eigenvalue evaluation of the direct zakharov-shabat problem," *Communications in Nonlinear Science and Numerical Simulation*, vol. 96, p. 105 718, 2021, ISSN: 1007-5704. DOI: 10.1016/j.cnsns.2021.105718.

[78] Z. Zhong, V. Cardoso, and E. Maggio, "Instability of ultracompact horizonless spacetimes," *Phys. Rev. D*, vol. 107, p. 044 035, 4 Feb. 2023. DOI: 10.1103/PhysRevD.107.044035.

[79] R. Schechter, *The Variational Method in Engineering* (McGraw-Hill chemical engineering series). McGraw-Hill, 1967.

[80] H. Anton, *Elementary Linear Algebra*. John Wiley & Sons, 2010, ISBN: 9780470458211.

[81] Y. Saad, *Numerical Methods for Large Eigenvalue Problems* (Algorithms and Architectures for Advanced Scientific Computing). Halsted Press-John Wiley & Sons Inc., New York, 1992, ISBN: 9780719033865.

[82] G. Sleijpen and H. V. D. Vorst, "A jacobi-davidson iteration method for linear eigenvalue problems," *SIAM Rev.*, vol. 42, pp. 267–293, 2000.

[83] C. Lanczos, “An iteration method for the solution of the eigenvalue problem of linear differential and integral operators,” *Journal of research of the National Bureau of Standards*, vol. 45, pp. 255–282, 1950.

[84] J. Francis, “The qr transformation a unitary analogue to the lr transformation - part 1,” *Comput. J.*, vol. 4, pp. 265–271, 1961.

[85] I. T. Gohberg, *Matrix Polynomials*. Academic Press, New York, 1982.

[86] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, *Templates for the Solution of Algebraic Eigenvalue Problems*. SIAM Publications, Philadelphia, 2000. DOI: 10.1137/1.9780898719581.ch1.

[87] L. Mazurenko and H. Voss, “On the number of eigenvalues of a rational eigenproblem,” *Technical Report 61, Section of Mathematics,*, Jul. 2003.

[88] P. Benner and H. Faßbender, “The symplectic eigenvalue problem, the butterfly form, the sr algorithm, and the lanczos method,” *Linear Algebra and its Applications*, vol. 275-276, pp. 19–47, 1998, Proceedings of the Sixth Conference of the International Linear Algebra Society, ISSN: 0024-3795. DOI: 10.1016/S0024-3795(97)10049-0.

[89] V. Mehrmann and D. Watkins, “Polynomial eigenvalue problems with hamiltonian structure,” *ETNA. Electronic Transactions on Numerical Analysis*, vol. 13, Jan. 2002.

[90] S. Yokota and T. Sakurai, “A projection method for nonlinear eigenvalue problems using contour integrals,” *JSIAM Lett.*, vol. 5, pp. 41–44, 2013.

[91] T. Yu-Bo, “Solving complex transcendental equations based on swarm intelligence,” *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 4, no. 6, pp. 755–762, 2009. DOI: 10.1002/tee.20477.

[92] P. Gonnet, R. Pachon, and L. Trefethen, “Robust rational interpolation and least-squares,” *Electronic Transactions on Numerical Analysis*, vol. 38, pp. 146–167, Jan. 2011.

[93] 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 *2016 IEEE Congress on Evolutionary Computation (CEC)*, 2016, pp. 1477–1484. DOI: 10.1109/CEC.2016.7743964.

[94] M. Mrozowski, “An efficient algorithm for finding zeros of a real function of two variables,” *IEEE Trans. Microw. Theory Tech.*, vol. 36, no. 3, pp. 601–604, Mar. 1988.

[95] J. Gulgowski and J. J. Michalski, “Topological attitude towards path following, applied to localization of complex dispersion characteristics for a lossy microwave, ferrite-coupled transmission line,” *IMA Journal of Applied Mathematics*, vol. 80, no. 2, pp. 494–507, Apr. 2015. DOI: 10.1093/imamat/hxt055.

[96] S. Weerakoon and T. Fernando, “A variant of newton’s method with accelerated third-order convergence,” *Applied Mathematics Letters*, vol. 13, no. 8, pp. 87–93, 2000, ISSN: 0893-9659. DOI: 10.1016/S0893-9659(00)00100-2.

[97] C. Chun, “Iterative methods improving newton’s method by the decomposition method,” *Computers & Mathematics with Applications*, vol. 50, no. 10, pp. 1559–1568, 2005, ISSN: 0898-1221. DOI: 10.1016/j.camwa.2005.08.022.

[98] E. Babolian and J. Biazar, “Solution of nonlinear equations by modified adomian decomposition method,” *Applied Mathematics and Computation*, vol. 132, no. 1, pp. 167–172, 2002, ISSN: 0096-3003. DOI: https://doi.org/10.1016/S0096-3003(01)00184-9.

[99] K. Sikorski, “Bisection is optimal,” *Numerische Mathematik*, vol. 40, pp. 111–117, 1982.

[100] R. Brent, “Algorithms for minimization without derivatives,” *Englewood Cliffs, Prentice Hall*, vol. 19, Jan. 2002. DOI: 10.2307/2005713.

[101] M. Dowell and P. Jarratt, “A modified regula falsi method for computing the root of an equation,” *BIT Numerical Mathematics*, vol. 11, pp. 168–174, 1971.

[102] X. Wang, “A simple proof of descartes’s rule of signs,” *American Mathematical Monthly*, vol. 111, Jun. 2004. DOI: 10.2307/4145072.

[103] S. Basu, R. Pollack, and M. Coste-Roy, *Algorithms in Real Algebraic Geometry* (Algorithms and Computation in Mathematics). Springer Berlin Heidelberg, 2007, ISBN: 9783540330998.

[104] P. Kowalczyk, “Global complex roots and poles finding algorithm based on phase analysis for propagation and radiation problems,” *IEEE Transactions on Antennas and Propagation*, vol. 66, no. 12, pp. 7198–7205, Sep. 2018. DOI: 10.1109/TAP.2018.2869213.

[105] A. Schönhage, “The fundamental theorem of algebra in terms of computational complexity,” in *Technical report*, Mathematisches Institut der Universitat Tubingen, 1982.

[106] J. R. Pinkert, “An exact method for finding the roots of a complex polynomial,” *ACM Trans. Math. Softw.*, vol. 2, no. 4, pp. 351–363, Dec. 1976, ISSN: 0098-3500. DOI: 10.1145/355705.355710.

[107] Y. Long and H. Jiang, “Rigorous numerical solution to complex transcendental equations,” *International Journal of Infrared and Millimeter Waves*, vol. 19, pp. 785–790, 1998.

[108] C. Wu, J. Li, G. Wei, and J. Xu, “A novel method to solve the complex transcendental equation for the permittivity determination in short-circuited line,” in *PIERS Proceedings*, vol. 2, 2010, pp. 1739–1742.

[109] P. Kravanja, M. Van Barel, O. Ragos, M. Vrahatis, and F. Zafiropoulos, “Zeal: A mathematical software package for computing zeros of analytic functions,” *Computer Physics Communications*, vol. 124, no. 2-3, pp. 212–232, Feb. 2000. DOI: 10.1016/S0010-4655(99)00429-4.

[110] J. H. Wilkinson, *Rounding Errors in Algebraic Processes*. USA: Dover Publications, Inc., New York, NY, USA, 1994, ISBN: 0486679993.

[111] J. Gilewicz and Y. Kryakin, “Froissart doublets in padé approximation in the case of polynomial noise,” *Journal of Computational and Applied Mathematics*, vol. 153, no. 1, pp. 235–242, 2003, Proceedings of the 6th International Symposium on Orthogonal Poly nomials, Special Functions and their Applications, Rome, Italy, 18-22 June 2001, ISSN: 0377-0427. DOI: 10.1016/S0377-0427(02)00674-X.

[112] J. J. Michalski, “Algorithm for locating complex zeros and poles with the use of border tracking on complex plane, and its application in dispersion characteristics calculation,” in *2018 IEEE/MTT-S International Microwave Symposium - IMS*, 2018, pp. 327–329.

[113] E. Anderson, Z. Bai, C. Bischof, *et al.*, *LAPACK Users' Guide*, 2nd. SIAM Publications, Philadelphia, 1995.

[114] R. Lehoucq, D. Sorensen, and C. Yang, “Arpack users' guide - solution of large-scale eigenvalue problems with implicitly restarted arnoldi methods,” in *Software, environments, tools*, 1998.

[115] MATLAB, *version 9.10 (R2021a)*. Natick, Massachusetts USA: The MathWorks Inc., 2021.

[116] S. Dziedziewicz, M. Warecka, R. Lech, and P. Kowalczyk, “Self-adaptive mesh generator for global complex roots and poles finding algorithm,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 71, no. 7, pp. 2854–2863, 2023. DOI: 10.1109/TMTT.2023.3238014.

[117] S. Dziedziewicz, R. Lech, and P. Kowalczyk, “A self-adaptive complex root tracing algorithm for the analysis of propagation and radiation problem,” *IEEE Transactions on Antennas and Propagation*, vol. 69, no. 8, pp. 5171–5174, 2021. DOI: 10.1109/TAP.2021.3060047.

[118] S. Dziedziewicz, M. Warecka, R. Lech, and P. Kowalczyk, “Global complex roots and poles finding algorithm in c × r domain,” *IEEE Access*, vol. 11, pp. 68 809–68 817, 2023. DOI: 10.1109/ACCESS.2023.3292961.

[119] 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 *2019 13th European Conference on Antennas and Propagation (EuCAP)*, 2019, pp. 1–5.

[120] S. Dziedziewicz, M. Warecka, R. Lech, and P. Kowalczyk, “Multipath complex root tracing,” in *2022 24th International Microwave and Radar Conference (MIKON)*, 2022, pp. 1–4. DOI: 10.23919/MIKON54314.2022.9924911.

[121] F. Mesa, G. Valerio, R. Rodríguez-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 and Propagation Magazine*, vol. 63, no. 5, pp. 33–45, 2021. DOI: 10.1109/MAP.2020.3003210.

[122] F. Gasdia and R. A. Marshall, “A new longwave mode propagator for the earth–ionosphere waveguide,” *IEEE Transactions on Antennas and Propagation*, vol. 69, no. 12, pp. 8675–8688, 2021. DOI: 10.1109/TAP.2021.3083753.

[123] M. Warecka, R. Lech, and P. Kowalczyk, “Hybrid method analysis of unshielded guiding structures,” in *2020 23rd International Microwave and Radar Conference (MIKON)*, 2020, pp. 398–401. DOI: 10.23919/MIKON48703.2020.9253827.

[124] 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, 2018, ISSN: 2073-8994. DOI: 10.3390/sym10080307.

[125] 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 Transactions on Microwave Theory and Techniques*, vol. 66, no. 11, pp. 4669–4676, 2018. DOI: 10.1109/TMTT.2018.2865928.

[126] 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 Transactions on Antennas and Propagation*, vol. 62, no. 6, pp. 2935–2945, Jun. 2014. DOI: 10.1109/TAP.2014.2313137.

[127] J.-P. Berenger, “A perfectly matched layer for the absorption of electromagnetic waves,” *Journal of Computational Physics*, vol. 114, no. 2, pp. 185–200, 1994, ISSN: 0021-9991. DOI: 10.1006/jcph.1994.1159.

[128] X.-Q. Sheng, J.-M. Jin, J. Song, C.-C. Lu, and W. C. Chew, “On the formulation of hybrid finite-element and boundary-integral methods for 3-d scattering,” *IEEE Transactions on Antennas and Propagation*, vol. 46, no. 3, pp. 303–311, Mar. 1998. DOI: 10.1109/8.662648.

[129] J. Rubio, J. Arroyo, and J. Zapata, “Analysis of passive microwave circuits by using a hybrid 2-D and 3-D finite-element mode-matching method,” *IEEE Trans. Microw. Theory Tech.*, vol. 47, no. 9, pp. 1746–1749, Sep. 1999.

[130] M. Warecka, R. Lech, and P. Kowalczyk, “Efficient finite element analysis of axially symmetrical waveguides and waveguide discontinuities,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 67, no. 11, pp. 4291–4297, 2019. DOI: 10.1109/TMTT.2019.2940021.

[131] M. Baranowski, S. Dziedziewicz, R. Lech, and P. Kowalczyk, “Hybrid technique for the em scattering analysis with the use of ring domain decomposition,” in *2020 Baltic URSI Symposium (URSI)*, 2020, pp. 81–84. DOI: 10.23919/URSI48707.2020.9254017.

[132] P. Kowalczyk, R. Lech, M. Warecka, and A. Kusiek, “Electromagnetic plane wave scattering from a cylindrical object with an arbitrary cross section using a hybrid technique,” *Journal of Electromagnetic Waves and Applications*, vol. 33, no. 2, pp. 178–192, 2019. DOI: 10.1080/09205071.2018.1532323.

[133] M. Polewski, R. Lech, and J. Mazur, “Rigorous modal analysis of structures containing inhomogeneous dielectric cylinders,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 52, no. 5, pp. 1508–1516, May 2004. DOI: 10.1109/TMTT.2004.827030.

[134] 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, Mar. 2013.

[135] C. M. Cameron R. J. Kudsia and R. R. Mansour, *Synthesis of Network-Circuit Approach*. John Wiley & Sons, Ltd, 2018, ch. 7, pp. 215–245, ISBN: 9781119292371. DOI: 10.1002/9781119292371.ch7.

[136] Y. Yamamoto, T. Kamiya, and H. Yanai, “Characteristics of optical guided modes in multilayer metal-clad planar optical guide with low-index dielectric buffer layer,” *IEEE Journal of Quantum Electronics*, vol. 11, no. 9, pp. 729–736, 1975. DOI: 10.1109/JQE.1975.1068998.

[137] I. P. Kaminow, W. L. Mammel, and H. P. Weber, “Metal-clad optical waveguides: Analytical and experimental study,” *Appl. Opt.*, vol. 13, no. 2, pp. 396–405, Feb. 1974. DOI: 10.1364/AO.13.000396.

[138] G. W. Hanson, “Dyadic green’s functions for an anisotropic, non-local model of biased graphene,” *IEEE Transactions on Antennas and Propagation*, vol. 56, no. 3, pp. 747–757, 2008. DOI: 10.1109/TAP.2008.917005.

[139] K.-L. Wong, Y.-T. Cheng, and J.-S. Row, “Analysis of a cylindrical-rectangular microstrip structure with an airgap,” *IEEE Trans. Microw. Theory Tech.*, vol. 42, no. 6, pp. 1032–1037, 1994.

[140] B. Salski, T. Karpisz, P. Kopyt, and J. Krupka, “Rigorous scattering matrix analysis of a fabry–perot open resonator,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 68, no. 12, pp. 5093–5102, 2020. DOI: 10.1109/TMTT.2020.3021716.

[141] P. Kowalczyk, “On root finding algorithms for complex functions with branch cuts,” *Journal of Computational and Applied Mathematics*, vol. 314, pp. 1–9, Apr. 2017, ISSN: 0377-0427. DOI: 10.1016/j.cam.2016.10.015.

[142] E. Wegert, *Visual complex functions: An introduction with phase portraits*. Birkhauser Basel, Jan. 2012, ISBN: 978-3-0348-0179-9. DOI: 10.1007/978-3-0348-0180-5.

[143] P. Kravanja and M. Van Barel, “A derivative-free algorithm for computing zeros of analytic functions,” *Computing*, vol. 63, no. 1, pp. 69–91, Jul. 1999. DOI: 10.1007/s006070050051.

[144] X. Ying and I. Norman Katz, “A reliable argument principle algorithm to find the number of zeros of an analytic function in a bounded domain,” *Numerische Mathematik*, vol. 53, pp. 143–163, Jan. 1988. DOI: 10.1007/BF01395882.

[145] P. Lampariello and R. Sorrentino, “The zepls program for solving characteristic equations of electromagnetic structures (computer program descriptions),” *IEEE Transactions on Microwave Theory and Techniques*, vol. 23, no. 5, pp. 457–458, 1975. DOI: 10.1109/TMTT.1975.1128598.

[146] W. Zieniutycz, “Comments on ”the zepls program for solving characteristic equations of electromagnetic structures”,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 31, no. 5, pp. 420–420, 1983. DOI: 10.1109/TMTT.1983.1131515.

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

[148] C. J. Gillan, A. Schuchinsky, and I. Spence, “Computing zeros of analytic functions in the complex plane without using derivatives,” *Computer Physics Communications*, vol. 175, no. 4, pp. 304–313, Aug. 2006, ISSN: 0010-4655. DOI: 10.1016/j.cpc.2006.04.007.

[149] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, “The quickhull algorithm for convex hulls,” *ACM Trans. Math. Softw.*, vol. 22, no. 4, pp. 469–483, Dec. 1996, ISSN: 0098-3500. DOI: 10.1145/235815.235821.

[150] A. Eiger, K. Sikorski, and F. Stenger, “A bisection method for systems of nonlinear equations,” *ACM Trans. Math. Softw.*, vol. 10, no. 4, pp. 367–377, Dec. 1984, ISSN: 0098-3500. DOI: 10.1145/2701.2705.

[151] W. Li, *A new method to find full complex roots of a complex dispersion equation for light propagation*, ArXiv e-prints, 2011. arXiv: 1109.0879.

[152] M.-C. Rivara, “Mesh refinement processes based on the generalized bisection of simplices,” *SIAM Journal on Numerical Analysis*, vol. 21, no. 3, pp. 604–613, 1984. DOI: 10.1137/0721042.