ORIGINAL ARTICLE OPEN ACCESS

AGORA: Adaptive Generation of Orthogonal Rational Approximations for Frequency-Response Data

Andria Lemus | Arif Ege Engin

Department of Electrical and Computer Engineering, San Diego State University, San Diego, California, USA

Correspondence Arif Ege Engin (aengin@sdsu.edu)

Received: 2 November 2024 | **Revised:** 2 January 2025 | **Accepted:** 29 January 2025

Keywords: macromodeling | rational functions | scattering parameters | stability | vector fitting

ABSTRACT

This paper introduces the adaptive generation of orthogonal rational approximations (AGORA) method for rational function approximation of measured or simulated microwave network parameters. The output of AGORA is a state-space model or a rational function model that is representative of the input data. The generated model can then be used to interpolate the data or to evaluate the data quality. A typical application is the time-domain signal and power integrity analysis in circuit simulations using the generated model from measured or simulated interconnnects and power distribution networks. AGORA allows estimating the model order for a given error tolerance and does not require any initial estimates or adjustment of hyperparameters.

1 | Introduction

Signal and power integrity analysis often involves time-domain waveforms such as eye diagrams or power supply voltage noise that are simulated in a circuit solver. An important aspect is the integration of interconnect and power distribution network models in circuit solvers. These components usually do not have closed-form equivalent circuit models at the package and board levels. Their frequency-response data can be obtained from electromagnetic solvers or high-frequency measurements. Integration of this frequency-response data into a circuit simulator can be achieved by approximating it with a rational function [1]. Methods available to generate rational function approximations from tabulated simulated or measured data include the widely popular vector fitting (VF) [2], Loewner framework [3], Sanathanan–Koerner (SK) iteration [4], RKFIT [5], ORA [6], and AAA (adaptive Antoulas–Anderson) [7–9].

The modeling methods outlined earlier continue to be used and developed to analyze a variety of high frequency systems as shown in Figure 1. For example, VF was used to model transceivers by using feedforward neural networks [11], to assist with postproduction tuning of coupled-resonator networks [12], to simplify the model of a power delivery network which is comprised of fully incorporated voltage regulators [13], and to create a model for a noisy package structure [14]. The design of a PCB cavity and locations of vias was expedited by using the parameterized Loewner matrix framework [15]. The speed of different CPUs in calculating large VF models was examined, towards handling increasingly complex representations of systems in examining signal and power integrity issues [16]. A microring resonator photonic filter was also modeled and tested by using complex vector fitting (CVF) [17].

The output of adaptive generation of orthogonal rational approximation (AGORA) is the rational function $r(s) = b(s)/a(s)$ where s is the Laplace variable, $a(s)$ is the denominator polynomial, and $b(s)$ is the numerator polynomial, which becomes a polynomial matrix for multi-port data. The input is the measured or simulated matrix data H_i, typically provided at the real frequency points $s_i = j\omega_i$. The rational function $r(s)$ represents a blackbox macromodel that characterizes the component, where the roots of the numerator and denominator are denoted as zeros and poles, respectively [18].

A primary difficulty in rational function approximation is that it requires the nonlinear least-squares solution of $b(s)/a(s) = H$. The

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

© 2025 The Author(s). *International Journal of Circuit Theory and Applications* published by John Wiley & Sons Ltd.

https://doi.org/10.1002/cta.4468

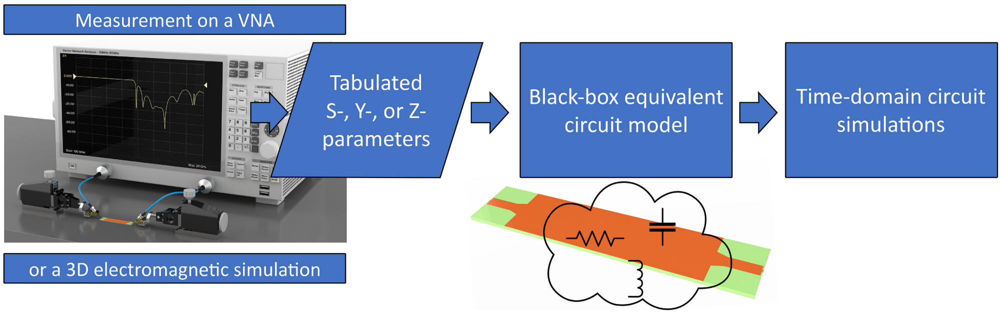

Measurement on a VNA
Tabulated S-, Y-, or Z-parameters
Black-box equivalent circuit model
Time-domain circuit simulations
or a 3D electromagnetic simulation

FIGURE 1 | Time-domain analysis for a passive system, such as this substrate integrated waveguide [10], may use a black-box macromodel.

nonlinearity comes from the unknown denominator polynomial $a(s)$. Available algorithms for rational function approximation differ in how they iteratively solve this nonlinear problem. The accuracy of the straightforward linearized solution of $b(s) = Ha(s)$ is, in general, unsatisfactory. Commonly used algorithms for rational approximation assume a known model order with the notable exception of AAA, which increments the model order at each iteration until convergence. AAA is however an interpolatory method; hence, it is most suitable for noise-free data and model orders close to the actual physical system. These properties of AAA present themselves as critical shortcomings for modeling of distributed networks including delay, such as transmission lines, which inherently requires an approximate lower model order than the physical system. An alternative is VF, which has been applied to generate rational approximations in a wide range of applications. Due to its noninterpolatory nature, it is in general more accurate than AAA for a least-squares approximation of noisy scientific data. However, VF is not adaptive, therefore computationally inefficient for estimation of model order. The proposed AGORA algorithm overcomes the shortcomings of both approaches. AGORA is both noninterpolatory and adaptive. Using the monomial basis for $r(s)$ is, of course, ill-conditioned for large orders. AGORA is based on the recent data-centered orthogonal rational basis in ORA [6], which is well-conditioned for larger orders as well.

There are several key differences between the introduced AGORA algorithm and the ORA algorithm [6]. First, in ORA, the orders of the numerator and denominator are set by the user. In AGORA, the model order does not need to be known. It is adaptively incremented until convergence. Next, ORA is based on SK iterations, which are time consuming, whereas AGORA does not need them. To solve the nonlinear least-squares problem, AGORA's approach of building up the model adaptively is unique and fundamentally different than existing approaches. The only existing adaptive approach is the AAA algorithm, which is however interpolatory and cannot provide a least-squares solutions at the support points. AGORA aims to solve the nonlinear least squares problem with no interpolation.

2 | Current Approaches for Rational Function Approximation

There are two major approaches for rational function approximation:

1. SK iteration: Various implementations include RKFIT, stabilized SK, ORA, and the popular VF algorithms. SK iteration is known to be accurate for approximating data with noise. The model order is fixed.
2. AAA: The barycentric basis of AAA comes from the Loewner framework. AAA is known to be accurate for closed-form functions or data without noise. The adaptive nature of AAA allows to estimate the model order.

SK-based methods, AAA, and AGORA all use different bases. However, more importantly, the process for solving the nonlinear least squares problem is different among them. They are all based on an iteratively updated polynomial $\hat{a}(s)$ as

$$r(s) = \frac{b(s)/\hat{a}(s)}{a(s)/\hat{a}(s)}, \tag{1}$$

where the linearized problem now seeks the least-squares solution of $b(s)/\hat{a}(s) \approx Ha(s)/\hat{a}(s)$. The presence of $\hat{a}(s)$ does not change the nonlinear problem; however, it has a profound effect on the accuracy of the linearized solution. The choice of the iteration polynomial $\hat{a}(s)$ dictates the convergence of the linearized problem to the solution of the actual non-linear problem $b(s)/a(s) \approx H$. This choice is the main difference between the three approaches.

In the following, we compare them using the partial fractions basis, as this is common to both VF and AAA. We also focus on the denominator $a(s)/\hat{a}(s)$, which is the main departure point among the three methods.

2.1 | VF

In SK iteration methods, such as VF, typical initial choice for $\hat{a}(s)$ is a polynomial with slightly damped complex roots. The monomial basis is of course not suitable for large-order approximations and replaced with a better-conditioned basis such as a partial fraction (VF) or an orthogonal rational basis (RKFIT or ORA [6]). In VF, the denominator of $r(s)$ is expressed in a partial-fractions basis as

$$\frac{a(s)}{\hat{a}(s)} = k_0 + \sum_{i=1}^{n} \frac{k_i}{s - \hat{p}_i}, \tag{2}$$

where $\hat{p}_i$ are the roots of $\hat{a}(s)$. The solution of the linearized problem yields an updated denominator $a(s)$. In the next iteration, this updated denominator $a(s)$ is used in place of $\hat{a}(s)$ and a new least squares problem is solved. As $\hat{a}(s)$ approaches $a(s)$, the linearized problem $b(s)/\hat{a}(s) \approx Ha(s)/\hat{a}(s)$ tends to get closer to the solution of the actual nonlinear problem $b(s)/a(s) \approx H$. Note that $a(s)$ and $\hat{a}(s)$ have the same polynomial degree n. Therefore, the

1097007x, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cta.4468, Wiley Online Library on [27/03/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

model order n remains the same at each iteration. This is a primary disadvantage of SK iteration-based methods such as VF, as n needs to be known in advance.

2.2 | AAA

The AAA algorithm is based on choosing roots of $\hat{a}(s)$ from a subset of the l frequency points as $\hat{p}_i = \hat{s}_i$, resulting in the denominator of

$$\frac{a(s)}{\hat{a}(s)} = \sum_{i=1}^{n} \frac{k_i}{s - \hat{p}_i}. \tag{3}$$

This form implies that the polynomial degree of $a(s)$ is one less than that of $\hat{a}(s)$. This is however not a problem, as AAA does not use the updated denominator $a(s)$ in place of $\hat{a}(s)$ at the next iteration. AAA is based on the property of barycentric functions as in (1) to yield interpolatory results $r(\hat{p}_i) = \hat{H}_i$, where $\hat{H}_i$ is the data provided at frequency $\hat{p}_i$. At each iteration, the frequency point with the maximum deviation is selected as the next node $\hat{p}_{n+1}$, and the iteration continues until a certain error tolerance is reached. This adaptive nature of AAA allows to estimate the model order, as each iteration increments the model order by one. This is an important feature, as a higher-than-necessary model order can result in spurious poles (poles with very small residues), affecting the model's accuracy.

2.3 | The Proposed Adaptive Generation Algorithm

The denominator in AAA has no constant or linear terms, whereas SK has the linear term k_0. Consider now the third option of adding a linear term sk_∞ to the denominator:

$$\frac{a(s)}{\hat{a}(s)} = k_0 + sk_\infty + \sum_{i=1}^{n-1} \frac{k_i}{s - \hat{p}_i} \tag{4}$$

The choice of the denominator in (4) is the main departure point for the proposed AGORA algorithm. Unlike SK iteration, the order n is not assumed to be known and can now be estimated in an adaptive manner similar to AAA. However, the iterations are aimed at achieving $a(s) = \hat{a}(s)$, similar to SK. The main feature of AGORA is that the order of $a(s)$ is one more than the order of $\hat{a}(s)$, due to the addition of the linear term. Hence, each iteration increments the model order n by one. In contrast, SK iteration maintains the same model order over iterations.

3 | AGORA

The presentation of the denominator in (4) using a partial-fractions basis is for illustration purposes only. Even though it is a possible option, we do not use this basis in this paper. AGORA instead uses the orthogonal rational approximation (ORA) basis [6]:

$$\frac{a(s)}{\hat{a}(s)} = \sum_{i=0}^{n} a_i o_i(s). \tag{5}$$

In ORA, the dividing polynomial $\hat{a}(s)$ is embedded into the orthogonal rational bases $o_i(s)$, which are not explicitly calculated, following a similar approach as in the Vandermonde with Arnoldi [19] method for generating polynomial approximations. ORA is data driven, in the sense that it allows calculating $o_i(s)$ at arbitrary frequency points s_i, without providing an explicit expression for $o_i(s)$. It also does not require the starting poles or an expression of $\hat{a}(s)$ directly; only the values of this polynomial at the given frequency points is needed. This property of ORA makes it the most convenient candidate for experimenting with the difference in the degrees of $a(s)$ and $\hat{a}(s)$ for adaptive generation, as it does not require a major modification of an existing algorithm.

ORA is based on two methods:

- `numfit`: This method creates an orthogonal basis, that is, a matrix containing the values of $o_i(s)$ at the given frequency points, using the Arnoldi iteration. The basis is created starting with a vector that incorporates the division by $\hat{a}(s)$, which extends the Vandermonde with Arnoldi [19] method from calculating polynomial approximations to rational approximations [6, 20]. This methods serves the same purpose as the residue extraction stage of VF.
- `denfit`: This method solves the linearized least-squares problem $b(s)/\hat{a}(s) \approx Ha(s)/\hat{a}(s)$ to calculate $a(s)$. Both $b(s)/\hat{a}(s)$ and $a(s)/\hat{a}(s)$ are expressed using the orthogonal rational basis in (5). This method serves the same purpose as the pole relocation stage of VF.

VF makes use of efficient state-space algorithms to generate large order models. To modify the VF basis in (2) into the form in (4) is complicated, as it requires switching from the proper state-space model in VF to a more general descriptor model, while ensuring the algorithms are applicable for large-order models. For example, the zeros of (3) are needed to find the poles obtained from AAA. The core AAA algorithm [7] extracts these poles by solving a generalized eigenvalue problem. An alternative method [9] is available that is numerically better conditioned without a frequency scaling. VF extracts the roots of $a(s)$ from the poles of the

Algorithm 1 AGORA

```
Require: response samples $H_i$, corresponding frequencies $s_i$, maximum order $n_{max}$, error treshold $\epsilon_{max}$
$n \leftarrow 1$, $\hat{a}(s) \leftarrow 1$
while $n \leq n_{max}$ and $\epsilon > \epsilon_{max}$ do
    denfit: Solve the linearized least-squares problem $b(s)/\hat{a}(s) \approx Ha(s)/\hat{a}(s)$ to calculate $a(s)$ using the basis in (5)
    numfit: Solve the linear least-squares problem $b(s)/a(s) \approx H$ to calculate $b(s)$
    $n \leftarrow n + 1$, $\hat{a}(s) \leftarrow a(s)$
end while
Create a state space model from calculated $b(s)/a(s)$
```

1097007x, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cta.4468, Wiley Online Library on [27/03/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

inverse state-space representation of (2). The partial fraction expansion in (4), on the other hand, is a descriptor model. Finding the zeros of (4) therefore requires a numerically well-conditioned method that is not immediately available from the established pole-extraction processes in AAA or VF. This is however easily accomplished using the orthogonal rational approximation (ORA) [6]. ORA is a data-centric approach and does not require the poles of $\hat{a}(s)$. The value of $\hat{a}(s_i)$ at the frequency points s_i is all that is needed. Hence, changing the order of $a(s)$ with respect to $\hat{a}(s)$ is straightforward in ORA. In AGORA, we therefore make use of the underlying ORA algorithms: `denfit` and `numfit`, where their Matlab implementations are given in [6]. The proposed AGORA algorithm is implemented in the Matlab function `agora` as shown in Figure A1. The pseudocode is shown in Algorithm 1.

TABLE 1 | Comparison of vector fitting (VF), adaptive Antoulas–Anderson (AAA), and the proposed AGORA approaches.

Method	Model order	Noninterpolatory
VF (i.e., SK iteration)	Fixed	Yes
AAA	Adaptive	No
AGORA	Adaptive	Yes

The accuracy of the linearized problem $b(s)/\hat{a}(s) = Ha(s)/\hat{a}(s)$ to solve the actual nonlinear problem $b(s)/a(s) = H$ depends on the convergence of $\hat{a}(s)$ to $a(s)$. In AGORA, increasing the model order in each step is meant to provide a better approximation of $\hat{a}(s)$ to $a(s)$. This heuristic goal is the same as in SK iteration or VF. AAA, on the other hand, relies on the interpolatory nature of the barycentric form. Obviously, convergence of $\hat{a}(s)$ to $a(s)$ is not expected in AAA, as the roots of $a(s)$ will in general not be a subset of the frequency points. On the other hand, each iteration increases the number of frequency points where interpolation occurs, with the goal of reaching a better least squares approximation overall. This interpolatory nature of AAA makes it suboptimal for least squares approximation of data including noise. The new AGORA approach therefore promises to incorporate the powerful features of both VF and AAA as shown in Table 1.

4 | Accuracy of AGORA

We provide several examples to compare the performance of AGORA with the `rationalfit` and `rational` MATLAB 2024a packages. Lower-triangular S-parameter values were used. The functions `rationalfit` and `rational` are based on VF and AAA, respectively. They are however proprietary implementations, and may differ from the original algorithms. The function `rational`, for example, ensures

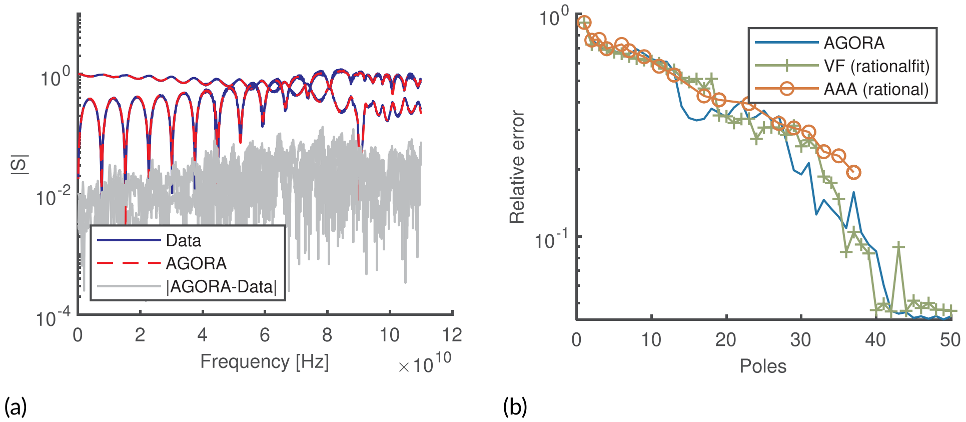


FIGURE 2 | Example 1: Noisy stripline data measured at 5001 frequency points was fitted using AGORA (a) Fitting results. (b) `rational` failed to generate models with more than 40 poles.

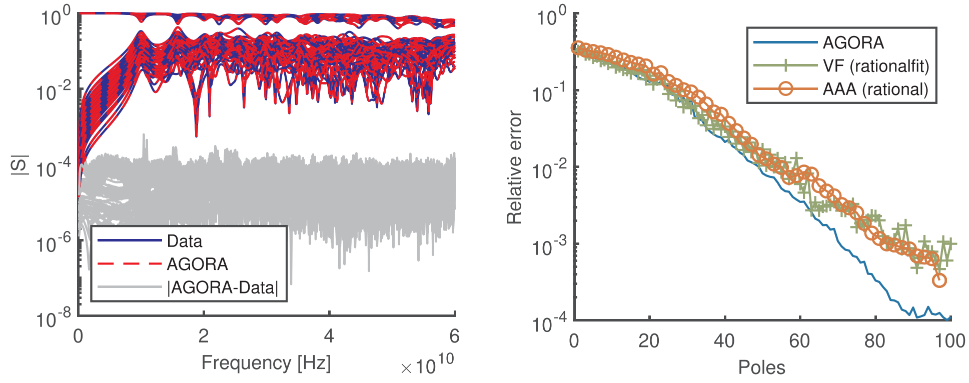


FIGURE 3 | Example 2: Simulated cavity resonator data at 10 ports with 600 frequency points. (a) Fitting results. (b) AGORA had a significantly lower RMS error when compared to `rationalfit` and `rational` at higher orders.

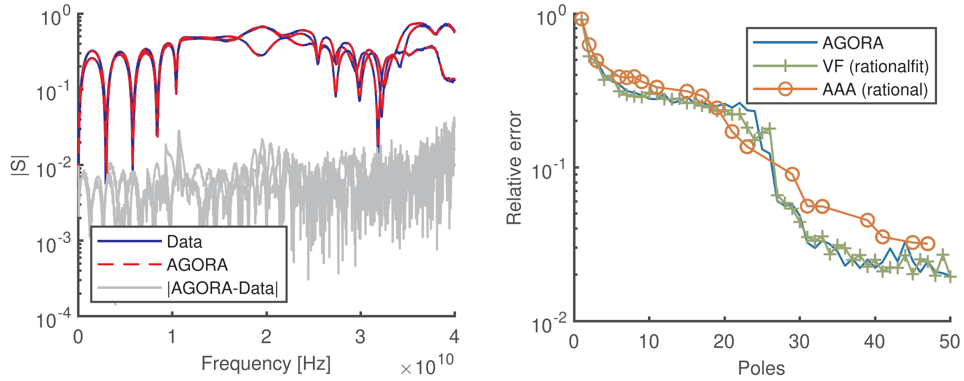
1097007x, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cta.4468, Wiley Online Library on [27/03/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

conjugate symmetry and stability of the poles, which are not true for the core AAA algorithm. We leave the number of SK iterations in `rationalfit` at its default settings of between 4 and 12. In the following study, the stability of poles was enforced in all the methods, including AGORA. Passivity can be another critical property. AGORA can be enhanced to generate passive models with the use of PASSOS [21, 22], which is based on a sum-of-squares rational function representation.

The two-port scattering parameter data from a stripline, which were measured from 100 MHz to 110 GHz on a vector network analyzer, were modeled with up to 50 poles, as depicted in Figure 2a. As shown in Figure 2b, AAA failed to generate models with more than 40 poles. The approximation by AGORA had lower RMS error than that of `rationalfit` and `rational` for high orders.

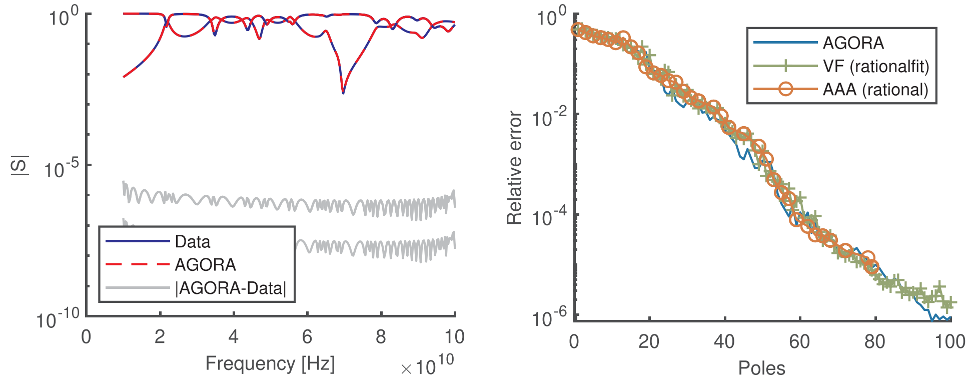
A second example, shown in Figure 3a,b, depicts the fit on scattering parameters from a cavity resonator resulting from a full-wave simulation using Sonnet with ten ports. In a comparison of models with up to 100 poles, Figure 3b shows that those with more than 85 poles had significantly lower RMS error when constructed with AGORA than with `rationalfit` or `rational`.

The third example uses the measured data from a four-port common-mode filter design. The fit is shown in Figure 4a. Figure 4b shows that the fit generated by AGORA had similar error



(a) (b)

FIGURE 4 | Example 3: Resulting fit on data from a four-port common-mode filter design from [10]. (a) Fitting results. (b) AGORA and `rationalfit` generated a fit to the function which had lower relative error than that of `rational` for higher numbers of poles.



(a) (b)

FIGURE 5 | Example 4: Resulting fit on the data from the finite-difference method simulation of a cavity resonator. (a) Fitting results. (b) AAA failed to generate models with more than 80 poles.

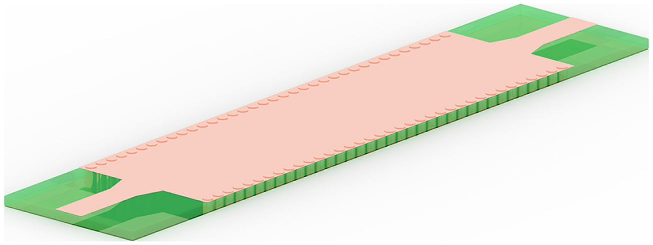


FIGURE 6 | Example 5:The simulated data from the pictured substrate integrated waveguide (SIW) [23, 24] was used for this example.

1097007x, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cta.4468, Wiley Online Library on [27/03/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

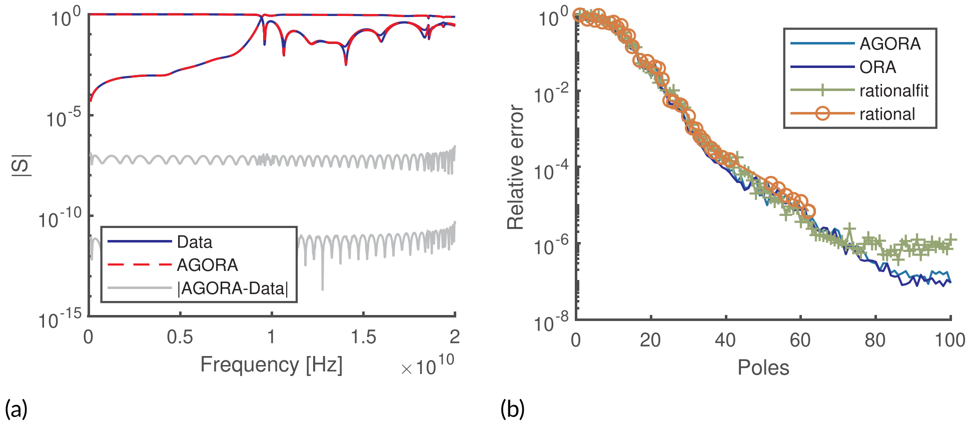


FIGURE 7 | Example 5: Fit of AG on a simulated two-port Substrate Integrated Waveguide (SIW). The model was created with 62 poles. (a) Fitting results. (b) The error of the fit by AG was lower than that of `rational` near 60 poles.

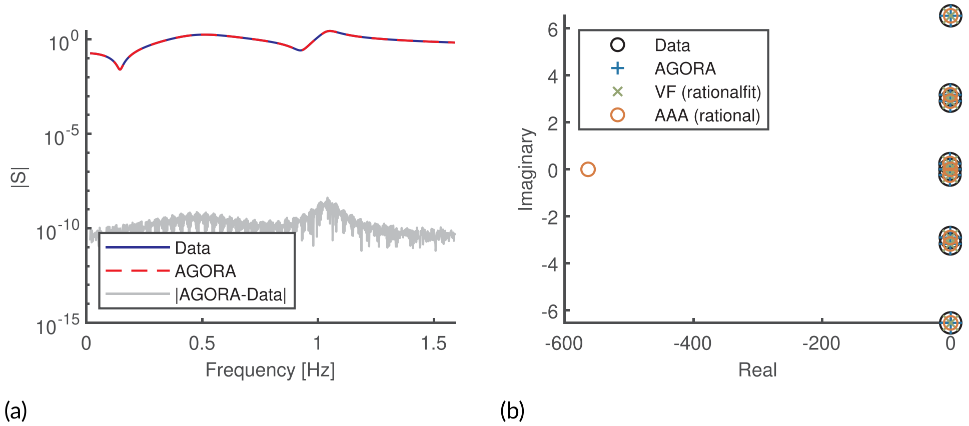


FIGURE 8 | Example 6: A 10-pole synthetic transfer function from [25] was used to generate S-parameter data. Shown are the (a) magnitude and (b) the extracted poles from the data.

as in `rationalfit`. `rational` was not able to generate models at certain orders and resulted in higher error for high orders.

For the fourth example, the simulated data using the finite-difference method from a two-port cavity resonator were investigated as shown in Figure 5a. Figure 5b shows that AGORA produced a fit with lower relative error than both `rational` and `rationalfit`, while `rational` failed to generate models with more than 80 poles.

For the fifth example, a two-port Substrate Integrated Waveguide (SIW) was simulated in Matlab by following the steps in [23]. 1000 frequency points were measured between 100 MHz and 20 GHz. The operating frequency range of the design is 10–15 GHz [24]. The SIW is shown in Figure 6. The fit of AG using 62 poles is shown in Figure 7a. The ORA implementation used 20 SK iterations. As shown in Figure 7b, the fit by AGORA had generally lower error than that of `rationalfit`.

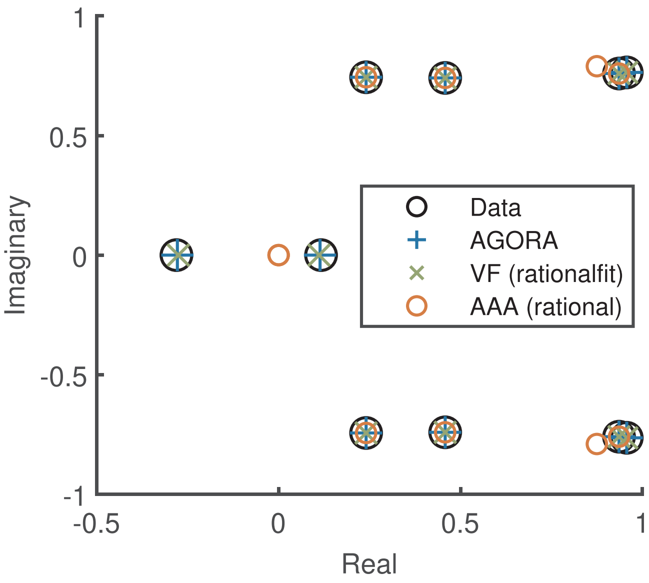


FIGURE 9 | Extracted residues for Example 6 in Figure 8a.

5 | Identifying the Correct Poles

If the order of the underlying system is known and the data is not noisy, it would be a desirable property for a rational function generator to extract the correct poles of the system. In this example, a 10-pole system from Example 8.3.3 in [25] was fit with `rationalfit`, `rational`, and AGORA. For this synthetic, one-port transfer function, the poles and residues are known. Therefore,

1097007x, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cta.4468, Wiley Online Library on [27/03/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

the ability of the algorithms to correctly determine the poles and residues by using 1000 frequency points was compared. The fit to the function is shown in Figure 8a. As shown in Figures 8b and 9, the poles and residues which were calculated by AGORA and VF were a good fit in comparison to the known values, whereas AAA was inaccurate in identifying a pole and several residues.

6 | Appearance of Spurious Poles

If the dataset is fit with a rational function of an order higher than that of the underlying system, there will be an excess of poles. Poles with small residues are considered spurious [7]. It is optimal if these poles are not generated. The residues calculated for different numbers of poles were compared in Figure 10. The function has ten known poles. For more than 10 poles, residues that are orders of magnitude smaller than those of the known values are present. These are considered spurious. It can also be seen that there was no fit available for every order when using `rational`. AGORA helps to detect such spurious poles or overfitting by adaptively building the model.

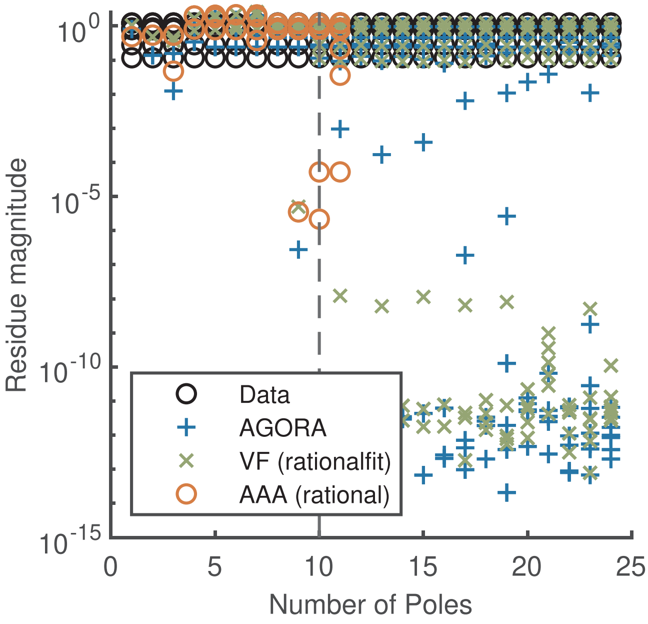


FIGURE 10 | Example 6: Residues corresponding to spurious poles for orders higher than the model order appear due to overfitting.

7 | Timing for Model Generation

In a typical setting, the user may not know the original model order. Therefore, if the user chooses VF, which requires SK iteration, they may need to select different orders and calculate the outcome before determining if additional orders should be tested. Therefore, the time required to use VF would increase if the original result was unsatisfactory. For a subsequent example, then, the cumulative time is defined as the total time which a function would require to fit models at all orders up to the given order.

As AGORA is adaptive, the time which it requires to increase in model order is expected to be lower than that of VF. Furthermore, it does not require SK iterations. By using the data in Example 2, the cumulative time required to fit the data were measured. As shown in Figure 11a, AGORA required less time per number of poles when compared to the `rationalfit` method. Figure 11b also shows how AGORA required less time per number of poles when fitting the simulated data from Example 5.

8 | Discussion

A complete convergence analysis is not yet available for AGORA, SK iteration [5], or AAA [7]. The residual error does not monotonously decrease with the number of poles. A theoretically better-studied method is the Levenberg–Marquardt approach that leads to the Whitfield estimator [26] for rational functions. Alternatively, fitting the derivative of the objective function in the instrumental variable approach aims to find the local minima [27]. These approaches have however not found widespread use, as they are either considered to be slow [28, 29] or not a significant improvement over SK iteration [20]. The convergence of AGORA especially in the presence of noise requires further research. VF as an example may fail to locate the poles for high noise levels [30] due to the presence of spurious poles. For the synthetic-transfer function example discussed in this paper, both AGORA and `rationalfit` were able to identify the correct poles.

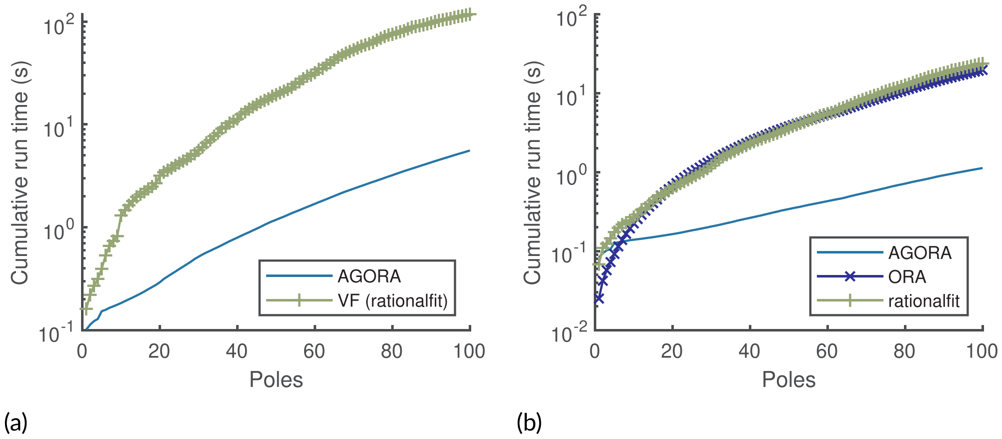


FIGURE 11 | Cumulative time from fitting (a) a simulated cavity resonator data set with no known poles or residues from Example 2 and (b) a simulated substrate integrated waveguide from Example 5.

1097007x, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cta.4468, Wiley Online Library on [27/03/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

A commonly desired property for the rational function is that it should have real coefficients and stable poles. This is achieved in the current implementation by the underlying ORA algorithm. Passivity on the other hand is typically not preserved in rational approximation of passive microwave network parameters. It can be enforced if necessary as a postprocessing step by perturbation of the residues, as commonly done in existing rational approximation approaches.

9 | Conclusion

We presented a new AGORA approach to linearize the least squares problem and estimate the model order for rational function approximation. AGORA is an alternative to the widely popular VF and AAA algorithms to compute rational function approximations. Initial results are promising in generating fast and accurate rational approximations by the new AGORA approach.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

1. P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, and L. M. Silveira, *System- and Data-Driven Methods and Algorithms* (De Gruyter, 2021), https://doi.org/10.1515/9783110498967.

2. B. Gustavsen and A. Semlyen, "Rational Approximation of Frequency Domain Responses by Vector Fitting," *IEEE Transactions on Power Delivery* 14, no. 3 (1999): 1052–1061.

3. A. J. Mayo and A. C. Antoulas, "A Framework for the Solution of the Generalized Realization Problem," *Linear Algebra and its Applications* 425, no. 2 (2007): 634–662, https://www.sciencedirect.com/science/article/pii/S0024379507001280. Special Issue in honor of Paul Fuhrmann.

4. C. Sanathanan and J. Koerner, "Transfer Function Synthesis as a Ratio of Two Complex Polynomials," *IEEE Transactions on Automatic Control* 8, no. 1 (1963): 56–58.

5. M. Berljafa and S. Güttel, "The RKFIT Algorithm for Nonlinear Rational Approximation," *SIAM Journal on Scientific Computing* 39, no. 5 (2017): A2049–A2071, https://doi.org/10.1137/15M1025426.

6. A. Ma and A. E. Engin, "Orthogonal Rational Approximation of Transfer Functions for High-Frequency Circuits," *International Journal of Circuit Theory and Applications* 51, no. 3 (2023): 1007–1019, https://onlinelibrary.wiley.com/doi/abs/10.1002/cta.3488.

7. Y. Nakatsukasa, O. Sète, and L. N. Trefethen, "The AAA Algorithm for Rational Approximation," *SIAM Journal on Scientific Computing* 40, no. 3 (2018): A1494–A1522, https://doi.org/10.1137/16M1106122.

8. I. V. Gosea and A. C. Antoulas, "Rational Approximation of the Absolute Value Function From Measurements: A Numerical Study of Recent Methods," (2020).

9. A. Valera-Rivera and A. E. Engin, "AAA Algorithm for Rational Transfer Function Approximation With Stable Poles," *IEEE Letters on Electromagnetic Compatibility Practice and Applications* 3, no. 3 (2021): 92–95.

10. W. A. Alarcon, A. E. Engin, I. Ndip, and K. Lang, "EBG Common-Mode Filter Design Using Uncoupled Coplanar Waveguide to Microstrip Transitions," *IEEE Letters on Electromagnetic Compatibility Practice and Applications* (2020): 1–1.

11. Y. Zhao, T. Nguyen, H. Ma, E.-P. Li, A. C. Cangellaris, and J. E. Schutt-Ainé, "Modular Neural Network-Based Models of High-Speed Link Transceivers," *IEEE Transactions on Components, Packaging and Manufacturing Technology* 13, no. 10 (2023): 1603–1612.

12. P. Zhao, "Phase De-Embedding of Narrowband Coupled- Resonator Networks by Vector Fitting," *IEEE Transactions on Microwave Theory and Techniques* 71, no. 4 (2023): 1439–1446.

13. A. Carlucci, T. Bradde, S. Grivet-Talocia, S. Mongrain, S. Kulasekaran, and K. Radhakrishnan, "A Compressed Multivariate Macromodeling Framework for Fast Transient Verification of System-Level Power Delivery Networks," *IEEE Transactions on Components, Packaging and Manufacturing Technology* 13, no. 10 (2023): 1553–1566.

14. A. Carlucci, T. Bradde, and S. Grivet-Talocia, "Addressing Load Sensitivity of Rational Macromodels," *IEEE Transactions on Components, Packaging and Manufacturing Technology* 13, no. 10 (2023): 1591–1602.

15. L. E. Carrera-Retana, M. Marín-Sánchez, C. Schuster, and R. Rimolo-Donadio, "Construction of Reciprocal Macromodels in the Loewner Matrix Framework," *IEEE Transactions on Microwave Theory and Techniques* 71, no. 8 (2023): 3561–3571.

16. V. Kukutla and R. Achar, "A Comparative Study of Performance of Eigenvalue Solvers for Parallel Vector Fitting in Multiport Tabulated Data Modeling," in *2023 ieee symposium on electromagnetic compatibility & signal/power integrity (emc+sipi)*, (2023): 306–310.

17. X. Jiang and Y. Ye, "Incorporating Frequency Responses of Passive Photonic Integrated Circuits Into Discrete Time-Domain Simulations Using Pole-Residue Fitting," *Optics & Laser Technology* 175 (2024): 110718, https://www.sciencedirect.com/science/article/pii/S0030399224001762.

18. S. Grivet-Talocia and B. Gustavsen, *Passive Macromodeling: Theory and Applications* (Hoboken, NJ: Wiley Professional, Reference & Trade, 2016).

19. P. D. Brubeck, Y. Nakatsukasa, and L. N. Trefethen, "Vandermonde With Arnoldi," *SIAM Review* 63, no. 2 (2021): 405–415, https://doi.org/10.1137/19M130100X.

20. J. M. Hokanson, "Multivariate Rational Approximation Using a Stabilized Sanathanan–Koerner Iteration," (2020).

21. F. Coronado and A. E. Engin, "Passos: Passive Approximation Through Sum-of-Squares Orthogonal Rational Functions," *IEEE Transactions on Electromagnetic Compatibility* 65, no. 2 (2023): 555–563.

22. A. Lemus, F. Coronado, and A. E. Engin, "Adaptive Generation of Passive Rational Function Approximations for Electromagnetic Simulation," in *2023 International Applied Computational Electromagnetics Society Symposium (aces)*, (2023): 1–2.

23. I. The MathWorks, "Prototype, Design, and Analysis of SIW Based Microstrip Tapered Transmission Line," https://www.mathworks.com/help/rfpcb/ug/prototype-design-and-analysis-of-siw-based-microstrip-tapered-transmission-line.html.

24. M. F. Bouchra Rahali, "Substrate Integrated Waveguide Power Divider, Circulator and Coupler in [10-15] ghz Band," *International Journal of Information Sciences and Techniques* 4, no. 1 (2014).

25. P. Benner, L. Feng, I. V. Gosea, et al., *Volume 1 System- and Data-Driven Methods and Algorithms* (Berlin, Boston: De Gruyter, 2021), https://doi.org/10.1515/9783110498967.

26. A. H. Whitfield, "Asymptotic Behaviour of Transfer Function Synthesis Methods," *International Journal of Control* 45, no. 3 (1987): 1083–1092, https://doi.org/10.1080/00207178708933791.

27. R. van Herpen, T. Oomen, and M. Steinbuch, "Optimally Conditioned Instrumental Variable Approach for Frequency-Domain System Identification," *Automatica* 50, no. 9 (2014): 2281–2293, https://www.sciencedirect.com/science/article/pii/S0005109814002775.

1097007x, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cta.4468, Wiley Online Library on [27/03/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

28. A. Arda Ozdemir and S. Gumussoy, “Transfer Function Estimation in System Identification Toolbox via Vector Fitting,” *IFAC-PapersOnLine* 50, no. 1 (2017): 6232–6237, https://www.sciencedirect.com/science/article/pii/S2405896317315045. 20th IFAC World Congress.

29. L. Knockaert, F. Ferranti, and T. Dhaene, “Vector Fitting vs. Levenberg-Marquardt: Some Experiments,” in *2009 IEEE Workshop on Signal Propagation on Interconnects*, (2009): 1–4.

30. S. Grivet-Talocia and M. Bandinu, “Improving the Convergence of Vector Fitting for Equivalent Circuit Extraction From Noisy Frequency Responses,” *IEEE Transactions on Electromagnetic Compatibility* 48, no. 1 (2006): 104–120.

Appendix A

```
function [poles,myss,errvec] = agora(S,nmax,errmax,ind)
% S: touchstone data, nmax: max number of poles (default = 100),
% errmax: max relative error (default = 0.0001)
% ind: index of matrix elements
% example: agora(sparameters('filename.s2p'),100,1e-5,[1 1;1 2]);
if nargin<2, nmax = 100; end
if nargin<3, errmax = 0.0001; end
p=size(S.Parameters,1);
if nargin<4
    ind=[];
    for ind1 = 1:p
        for ind2 = 1:p
            ind = [ind;[ind1 ind2]];
        end
    end
end
f = [];
for k = 1:size(ind,1)
    f = [f squeeze(S.Parameters(ind(k,1),ind(k,2),:))];
end
frequency = S.Frequencies;
den = ones(length(frequency),1); %initial denominator is a 1s vector
s=1i*2*pi*frequency;
err = inf;
errvec = [];
n=0;
while err>errmax && n<nmax
    n = n+1
    [den,poles] = denfit(den,s,n,n,f);
    [~,~,~,myss,err] = numfit(den,s,n,n,f);
    errvec = [errvec; n err];
    end
end
```

FIGURE A1 | Matlab implementation of AGORA that uses `numfit` and `denfit` functions defined in [6].

1097007x, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cta.4468, Wiley Online Library on [27/03/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License