



Article

Macromodeling High-Speed Circuit Data Using Rational Krylov Fitting Method

Mohamed Sahouli *  and Anestis Dounavis * 

Department of Electrical and Computer, Western University, London, ON N6A 3K7, Canada

* Correspondence: msahouli@uwo.ca (M.S.); adounavi@uwo.ca (A.D.)

Abstract: This paper presents the modeling of high speed distributed networks characterized by S-parameters frequency data using the rational Krylov fitting (RKFIT) algorithm. Numerical examples illustrate the effectiveness of the method to compute stable rational approximation that fit given S-parameters data. In addition, it is shown that RKFIT has some advantages when compared to the well-established Vector Fitting (VF) method, such as more accurate fitting, less dependence on the choice of the initial poles of the algorithm, and faster convergence. Numerical examples are implemented using RKFIT and the results are compared with VF and the Loewner Matrix (LM) algorithm.

Keywords: distributed networks; macromodeling; rational approximation; s-parameters; vector fitting



Citation: Sahouli, M.; Dounavis, A. Macromodeling High-Speed Circuit Data Using Rational Krylov Fitting Method. *Energies* **2021**, *14*, 7318. <https://doi.org/10.3390/en14217318>

Academic Editor: Stefano Grivet-Talocia

Received: 13 September 2021
Accepted: 27 October 2021
Published: 4 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Advances in integrated circuit technology have increased the design complexity and operating signal speeds of electronic devices and are placing significant demands on electronic design automation (EDA) tools to provide the same efficiency and accuracy. Aggressive design objectives such as system on chip or system in package coupled with increased operating frequencies require multidisciplinary design methodologies such as electrical, thermal and electromagnetic analysis to accurately model high-speed integrated circuits (IC). With higher operating frequencies, signal integrity issues such as signal delay, distortion, crosstalk, and attenuation dominate and significantly degrade circuit performance. These effects exist throughout the design process at the chip, packaging, printed-circuit-board, and backplane levels. Under these circumstances, the behavior of interconnects as well as other electromagnetic modules such as vias, connectors, and packages are often characterized by tabulated data, obtained by measurements, or by electromagnetic simulations [1–5]. As a result, macromodeling of distributed systems using tabulated data is an important issue for the analysis of high-speed circuits. Once the distributed models have been modeled, they can be converted to ordinary differential equations (ODE) or circuit equivalent models which can be used in circuit simulators for time-domain analysis for the purpose of signal integrity analysis [1,6].

Macromodeling of distributed networks characterized by frequency-domain data is usually performed by rational curve fitting techniques. The maximum frequency at which this fitting is done depends on the application and the type of input signal expected in the network, however, a general rule is to take the maximum frequency as $f_{max} = 0.35/t_r$, where t_r is the rise time of the input signal [6]. Among these techniques, the vector fitting (VF) algorithms [2,3,7–12] have emerged as a popular system identification tool since the rational approximation is formulated as a linear least square problem and uses an iterative pole relocation scheme to improve the approximation. This leads to better numerical stability and robustness when compared to non-iterative or polynomial approaches. Recently, the Loewner matrix (LM) has been proposed as an alternative to VF for multi-port networks [4,5], since LM models are more compact. However, unlike

VF, which builds stable models by construction, LM models may not always be stable. To tackle this problem, in [13], the unstable part of the LM model is fitted using a low order polynomial, however this increases the error of the LM approximation.

In recent years, rational Krylov fitting (RKFIT) [14,15] has emerged as a new iterative method for the computation of rational approximation from given data samples. Like VF, the RKFIT method aims to find accurate rational approximations by solving a least squares (LS) problem, but does it by using rational Krylov techniques that guarantee the stability of the model. RKFIT has been successfully used in numerous applications such as computational methods for 3-D transient electromagnetic modeling [16], the efficient computation of compression of finite difference grid for nonhomogeneous mediums [17], and very recently in the calculation of electromagnetic transients on overhead lines and underground cables [18]. In the latter work, models of cables were obtained from Y-parameters data and subsequently used to perform transient analysis.

In this work, RKFIT is used to create stable macromodels from S-parameter data characterizing high-speed interconnect circuits. Unlike VF, where the choice of the poles in the initial stage of the fitting process can affect the end-result of the model [19], RKFIT is less dependent on that initial choice of poles. Numerical examples show that the RKFIT generally approximates the data more accurately and in less iterations when compared to VF. Furthermore, the accuracy of RKFIT is also compared with the LM algorithm.

The paper first presents a brief overview of the LM and RKFIT algorithms in Section 2 followed by numerical examples and a conclusion in Sections 3 and 4, respectively.

2. Review of Rational Fitting Algorithms

A multiport network characterized by tabulated data can be expressed in the form of admittance, impedance, hybrid, or scattering parameters and have the following form:

$$\mathbf{H}(s) = [H_{ij}(s)], \quad i, j \in 1, \dots, P \quad (1)$$

where H_{ij} is a single transfer function (TF), P corresponds to the number of ports of the system, and s is the Laplace variable. Since the LM algorithm is more recent and less prominent than VF, an overview of the main steps of LM is presented next.

2.1. Review of the LM Algorithm

The objective of the LM method is to obtain a macromodel that seeks to accurately estimate the tabulated data of Equation (1) in the frequency domain and obtain the following system:

$$\mathbf{H}(s) = \mathbf{C}(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} \quad (2)$$

where $\mathbf{A}, \mathbf{E} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times P}$, $\mathbf{C} \in \mathbb{R}^{P \times n}$, and $\mathbf{D} \in \mathbb{R}^{P \times P}$ describe the multiport system of order n . How to obtain the descriptor state space (DSS) matrices of \mathbf{A} , \mathbf{B} , \mathbf{C} , \mathbf{D} , and \mathbf{E} is presented next.

The tabulated data of Equation (1) is first partitioned in two sets (one containing even frequencies and the other containing the odd ones) as:

$$\{s_m, \mathbf{H}(s_m)\} \rightarrow \{\tau_i, \mathbf{H}(\tau_i)\}, \{v_j, \mathbf{H}(v_j)\} \quad (3)$$

where $i = 1, \dots, \bar{m}$ and $j = 1, \dots, \underline{m}$. Here $\bar{m} + \underline{m} = M$.

$$\begin{cases} \bar{m} = \underline{m} = \frac{M}{2} & \text{if } M \text{ is even} \\ \bar{m} = \underline{m} + 1 = \frac{M+1}{2} & \text{if } M \text{ is odd} \end{cases}$$

The right and left data sets formed are expressed as:

$$\begin{aligned} \Gamma &= \text{diag}[\tau_1, \dots, \tau_m] \in \mathbb{C}^{m \times m}, \\ \mathbf{R} &= [R_1, \dots, R_m] \in \mathbb{C}^{P \times m}, \\ \mathbf{W} &= [W_1, \dots, W_m] \in \mathbb{C}^{P \times m} \\ \\ \mathbf{Y} &= \text{diag}[v_1, \dots, v_m] \in \mathbb{C}^{m \times m}, \\ \mathbf{L}^T &= [L_1, \dots, L_m] \in \mathbb{C}^{m \times P}, \\ \mathbf{V}^T &= [V_1, \dots, V_m] \in \mathbb{C}^{m \times P} \end{aligned}$$

where $W_i = \mathbf{H}(\tau_i)R_i$ and $V_i = L_i\mathbf{H}(\tau_i)$ and \mathbf{R} and \mathbf{L} are right and left tangential directions for right and left data. Columns of the identity matrix can be used for R_i and L_i as done in [4].

Next the Loewner matrix \mathbb{L} and shifted Loewner Matrix $\sigma\mathbb{L}$ are defined as:

$$\mathbb{L} = \begin{bmatrix} \frac{V_1 R_1 - L_1 W_1}{v_1 - \tau_1} & \dots & \frac{V_1 R_m - L_1 W_m}{v_1 - \tau_m} \\ \vdots & \ddots & \vdots \\ \frac{V_m R_1 - L_m W_1}{v_m - \tau_1} & \dots & \frac{V_m R_m - L_m W_m}{v_m - \tau_m} \end{bmatrix} \tag{4}$$

$$\sigma\mathbb{L} = \begin{bmatrix} \frac{v_1 V_1 R_1 - \tau_1 L_1 W_1}{v_1 - \tau_1} & \dots & \frac{v_1 V_1 R_m - \tau_m L_1 W_m}{v_1 - \tau_m} \\ \vdots & \ddots & \vdots \\ \frac{v_m V_m R_1 - \tau_1 L_m W_1}{v_m - \tau_1} & \dots & \frac{v_m V_m R_m - \tau_m L_m W_m}{v_m - \tau_m} \end{bmatrix}. \tag{5}$$

The matrices \mathbb{L} , $\sigma\mathbb{L}$, \mathbf{V} , and \mathbf{W} are complex. In order to obtain a macromodel with real DSS matrices, a similarity transformation is used as follows [4,20]:

$$\mathbb{L}_r = \mathbb{T}^* \mathbb{L} \mathbb{T} \tag{6a}$$

$$\sigma\mathbb{L}_r = \mathbb{T}^* \sigma\mathbb{L} \mathbb{T} \tag{6b}$$

$$\mathbb{V}_r = \mathbb{T}^* \mathbf{V} \tag{6c}$$

$$\mathbb{W}_r = \mathbb{T} \mathbf{W} \tag{6d}$$

where,

$$\mathbb{T} = \text{blkdiag}[\mathfrak{t}, \dots, \mathfrak{t}] \in \mathbb{C}^{m \times m}, \quad \mathfrak{t} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -j \\ 1 & +j \end{bmatrix}.$$

Once the Loewner and shifted Loewner are computed, the transfer function (TF) to fit the given data can be computed as:

$$\mathbf{H}(s) = \mathbf{W}(s\mathbb{L} - \sigma\mathbb{L})^{-1}\mathbf{V}. \tag{7}$$

An important property of Equation (7) is that it fits exactly at the given data points and uses interpolation for the in-between data. However, because this macromodel depends on the number of data points it can be large and inefficient and the matrices of Equation (7) are usually reduced using a singular value decomposition (SVD) on $(s\mathbb{L} - \sigma\mathbb{L})$. Any value of s can be chosen as long as it is not the eigenvalue of the $(\sigma\mathbb{L}, \mathbb{L})$ matrix pencil [4], resulting in the following expression:

$$\text{SVD}(s\mathbb{L}_r - \sigma\mathbb{L}_r) = \mathbf{Y}\mathbf{\Sigma}\mathbf{X} \tag{8}$$

where $\mathbf{\Sigma}$ is a diagonal matrix containing the singular values, and \mathbf{Y} and \mathbf{X} are orthonormal matrices corresponding to the left and right singular values. The order n of the approximation

is chosen as the location where a large drop of the normalized singular value happens as described in [4]. The time domain macromodel is constructed as:

$$\begin{aligned} E\dot{x}(t) &= \mathbf{A}x(t) + \mathbf{B}u(t) \\ y(t) &= \mathbf{C}x(t) + \mathbf{D}u(t) \end{aligned} \tag{9}$$

where the DSS matrices are constructed as:

$$\begin{aligned} \mathbf{A} &= -\mathbf{Y}_n^* \sigma \mathbb{L}_r \mathbf{X}_n, & \mathbf{B} &= \mathbf{Y}_n^* \mathbb{V}_r, \\ \mathbf{C} &= \mathbb{W}_r \mathbf{X}_n, & \mathbf{E} &= -\mathbf{Y}_n^* \mathbb{L}_r \mathbf{X}_n \end{aligned} \tag{10}$$

where $\mathbf{X}_n \in \mathbb{R}^{\bar{k} \times n}$ and $\mathbf{Y}_n \in \mathbb{R}^{k \times n}$ are constructed from the first n columns of \mathbf{X} and \mathbf{Y} of Equation (8) respectively [4]. If needed, the value of \mathbf{D} is extracted as described in [21]. The next section discusses the RKFIT algorithm.

2.2. RKFIT Algorithm

The objective of RKFIT algorithm [14,15] is to determine a rational function $r = p_{m+k}/q_m$, (where p_n and $q_n \in \mathcal{P}_n$ are n -th degree polynomials) that minimizes ε defined as:

$$\varepsilon = \frac{\|\mathbf{F}\mathbf{b} - r(\mathbf{A})\mathbf{b}\|_2}{\|\mathbf{F}\mathbf{b}\|_2} \tag{11}$$

where \mathbf{F} and \mathbf{A} are $N \times N$ matrices that contain $H_{ij}(s_k)$ and s_k ($k = 1, \dots, N$) in their diagonals respectively, and \mathbf{b} is an $N \times 1$ a nonzero starting vector. Since the denominator q_m of r is an unknown, Equation (11) is a nonlinear problem, RKFIT iteratively improves a starting guess for q_m by solving a linearized problem at each iteration. A more detailed description of the algorithm is presented next.

2.3. Review of the RKFIT Algorithm

This section presents a brief summary of the most important concepts of the RKFIT algorithm. For a more in-depth view about the theory and applications of the RKFIT see [14,15].

RKFIT is a curve fitting technique that uses the concept of rational Krylov spaces to solve the LS problem Equation (11) and finds rational approximation $r(s)$ to fit given data samples. A rational Krylov space of order m for matrix \mathbf{A} and vector \mathbf{b} is defined as:

$$\begin{aligned} \mathcal{Q}_{m+1}(\mathbf{A}, \mathbf{b}, q_m) &:= q_m(\mathbf{A})^{-1} \mathcal{K}_{m+1}(\mathbf{A}, \mathbf{b}) \\ \mathcal{K}_{m+1}(\mathbf{A}, \mathbf{b}) &:= \text{span}\{\mathbf{b}, \mathbf{A}\mathbf{b}, \dots, \mathbf{A}^m \mathbf{b}\} \end{aligned} \tag{12}$$

where $q_m \in \mathcal{P}_m$ is a polynomial with no roots in the spectrum of \mathbf{A} and \mathcal{K}_{m+1} is a Krylov subspace associated with \mathbf{A} and \mathbf{b} .

The method utilizes two rational Krylov spaces \mathcal{S} and \mathcal{T} , called search space and target spaces respectively, to look for poles in the search space \mathcal{S} that provides a better LS approximation to $\mathbf{F}\mathbf{b}$ from the target space \mathcal{T} .

There are two main steps in the RKFIT algorithm: Identification of the target space \mathcal{T} followed by solving a linear problem and relocating the poles. The first step of the algorithm is to compute the rational Arnoldi decomposition (RAD) of the search space $\mathcal{S} = \mathcal{Q}_{m+1}(\mathbf{A}, \mathbf{b}, q_m)$ defined as:

$$\mathbf{A}\mathbf{V}_{m+1}\mathbf{K}_m = \mathbf{V}_{m+1}\mathbf{H}_m \tag{14}$$

where \mathbf{V}_{m+1} is an orthogonal basis of \mathcal{S} , and \mathbf{K}_m and \mathbf{H}_m are called unreduced upper Hessenberg matrices. Once \mathcal{S} is computed and depending on the value of k (where k is an optional parameter of the algorithm that allows for the extension of the RAD [14,15]) target space $\mathcal{T} = \mathcal{R}(\hat{\mathbf{V}}_{m+k+1})$ is also computed.

Next, the numerator of Equation (11) can be rewritten as follows [14,15,18]:

$$\begin{aligned} \|\mathbf{F}\mathbf{b} - r(\mathbf{A})\mathbf{b}\|_2 &= \|\mathbf{F}\mathbf{b} - \hat{\mathbf{V}}_{m+k+1}\hat{\mathbf{V}}_{m+k+1}^*\mathbf{b}\|_2 \\ &= \|(\mathbf{I} - \mathcal{P}_{\mathcal{T}})\mathbf{F}\mathbf{b}\|_2 \end{aligned} \tag{15}$$

where $\mathcal{P}_{\mathcal{T}} = \hat{\mathbf{V}}_{m+k+1}\hat{\mathbf{V}}_{m+k+1}^*$ is the orthogonal projection onto \mathcal{T} . Once Equation (15) is computed, the RKFIT minimizes it by finding a vector \mathbf{v} defined as:

$$\mathbf{v} = \underset{\check{\mathbf{v}} \in \mathcal{S}, \|\check{\mathbf{v}}\|_2=1}{\operatorname{argmin}} \|(\mathbf{I} - \mathcal{P}_{\mathcal{T}})\mathbf{F}\check{\mathbf{v}}\|_2 \tag{16}$$

where a solution for Equation (16) is given by $\mathbf{v} = \mathbf{V}_{m+1}\hat{\mathbf{c}}$, where $\hat{\mathbf{c}}$ is a right singular vector corresponding to a smallest singular value of a matrix \mathbf{S} defined as:

$$\mathbf{S} = \mathbf{F}\mathbf{V}_{m+1} - \hat{\mathbf{V}}_{m+k+1}(\hat{\mathbf{V}}_{m+k+1}^*\mathbf{F}\mathbf{V}_{m+k+1}) \tag{17}$$

Finally, the RKFIT algorithm computes a new guess for the polynomial q_m defined as \check{q}_m such that $\mathbf{v} = \mathbf{V}_{m+1}\hat{\mathbf{c}} = \check{q}_m(\mathbf{A})q_m(\mathbf{A})^{-1}\mathbf{b}$ and if the error target ε of Equation (11) is not reached, a new iteration starts with a new search space \mathcal{S} using $q_m = \check{q}_m$. Once the poles converge, the rational approximation can be computed as $r(\mathbf{A})\mathbf{b} = \|\mathbf{b}\|_2\hat{\mathbf{V}}_{m+k+1}\mathbf{c}$ with the coefficient vector $\mathbf{c} = \hat{\mathbf{V}}_{m+k+1}^*(\mathbf{F}\mathbf{b})/\|\mathbf{b}\|_2$.

RKFIT with MIMO Data

To deal with MIMO transfer functions, a new ε is defined as follows:

$$\varepsilon = \sqrt{\frac{\sum_{j=1}^l \|D^j\mathbf{F}^j\mathbf{b} - r^j(\mathbf{A})\mathbf{b}\|_2}{\sum_{j=1}^l \|D^j\mathbf{F}^j\mathbf{b}\|_2}} \tag{18}$$

where F^j corresponds to the j -th transfer function and D^j corresponds to weighting values. Now, the goal of the RKFIT algorithm becomes to minimize Equation (18). To do so, Equation (16) becomes:

$$\mathbf{v} = \underset{\check{\mathbf{v}} \in \mathcal{S}, \|\check{\mathbf{v}}\|_2=1}{\operatorname{argmin}} \sum_{j=1}^l \|D^j(\mathbf{I} - \mathcal{P}_{\mathcal{T}})F^j\check{\mathbf{v}}\|_2 \tag{19}$$

where a solution for Equation (19) is also given by $\mathbf{v} = \mathbf{V}_{m+1}\hat{\mathbf{c}}$, where $\hat{\mathbf{c}}$ is a right singular vector corresponding to a smallest singular value of a matrix \mathbf{S} defined as:

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_1^T & \dots & \mathbf{S}_l^T \end{bmatrix} \tag{20}$$

$$\mathbf{S}_j = D^j \left[\mathbf{F}^j\mathbf{V}_{m+1} - \hat{\mathbf{V}}_{m+k+1}(\hat{\mathbf{V}}_{m+k+1}^*\mathbf{F}^j\mathbf{V}_{m+k+1}) \right]. \tag{21}$$

Finally, the coefficient vector \mathbf{c}^j for each j -th rational approximation is given by $\mathbf{c} = (D^j\hat{\mathbf{V}}_{m+k+1})^\dagger(D^j\mathbf{F}^j\mathbf{b})/\|\mathbf{b}\|_2$.

As presented, the rational approximation of RKFIT is not in terms of poles and residues. To get a rational approximation in a partial fraction basis form, \mathbf{c} can be transformed into residues and $\{r_l\}_{l=0}^m$ will be transformed to $\hat{r}_l(s) = \frac{1}{s-\zeta_l}$ partial fraction basis, see ([15], Chapter 7) for more details. In the next section, numerical examples are presented to validate the accuracy of the RKFIT when fitting S-parameter data from high-speed interconnect circuits.

3. Numerical Results

To demonstrate the effectiveness of RKFIT when fitting S-parameter data, two examples are presented. All the examples are from symmetric MIMO systems, therefore the fitting is

done for the data from $H(s)_{ij}$ where $i \leq j$ with common poles for all transfer functions. In addition all the fitting is compared with the relaxed version of the VF (RVF) [3] and LM [4,5] algorithms. The error between the given data $\mathbf{H}(s)$ and the rational approximations $\hat{\mathbf{H}}(s)$ is computed using the normalized \mathcal{H}_2 -norm defined as:

$$\mathcal{H}_2\text{-error} = \sqrt{\frac{\sum_{i=1}^{N_s} \|\hat{\mathbf{H}}(s_i) - \mathbf{H}(s_i)\|_F^2}{\sum_{i=1}^{N_s} \|\mathbf{H}(s_i)\|_F^2}} \quad (22)$$

where $\|A\|_F$ is the Frobenius norm of a matrix A . The simulation was done using the MATLAB toolboxes available at [22,23] for RKFIT and RVF respectively.

3.1. Measured Four Port Network

The first example corresponds to a four-port network of an 8-inch PCB interconnect (courtesy of Broadcom) measured using a vector network analyzer. The data consists of 1600 linearly spaced frequency points ranging from 0.3 GHz to 20 GHz. The fitting is done with an order of $n = 90$ for both RKFIT and RVF. The initial poles for the RKFIT algorithm were set to ∞ using the MATLAB command `inf`, while the initial poles of the RVF algorithm were linearly spaced out across the given bandwidth as recommended in [3,12]. For both algorithms, the poles refinement step is done for three iterations and then a model is formed to calculate the error Equation (22) between the computed rational approximations and the given data set. This process is repeated for 15 iterations for RKFIT and RVF.

As seen from Figure 1, RKFIT converges very quickly to a minimal value while RVF takes a few more iteration to converge to a stable error. In addition, Figure 2 and 3 show sample plots of the magnitude and phase of S_{11} and S_{12} , where it can be seen that both RKFIT and RVF do a good job capturing the behavior the data. However, as can be seen from the zoomed figure of Figure 4, RKFIT does a better job fitting the data compared to RVF. It should be noted that the results shown in the figures are from the fourth iteration for RKFIT since the error does not change in the subsequent iterations, however the results for RVF are from the fifteenth iteration. The computed poles obtained from RKFIT and RVF are also shown in Figure 5, as expected all the poles are stable since both methods guarantee stable models by construction.

Finally, Figures 2 and 3 also show fitting done with the LM algorithm, where the number of poles is chosen to get similar accuracies as RKFIT and RVF. It should be noted that even though the number of poles is greater for LM, due to the way the model is represented, the macromodel from the LM method is more compact than both RKFIT and RVF.

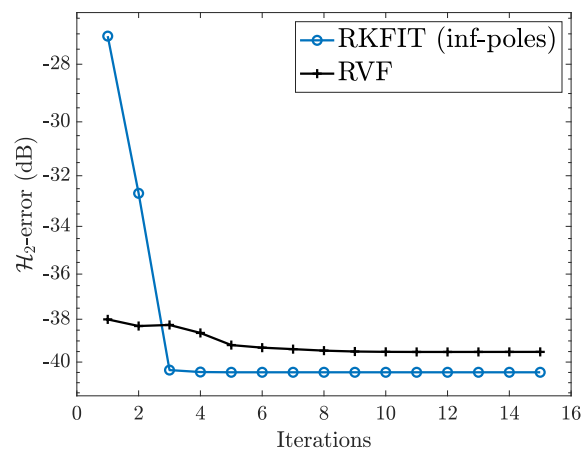


Figure 1. \mathcal{H}_2 -error per iteration comparison between RKFIT and RVF algorithms of Section 3.1.

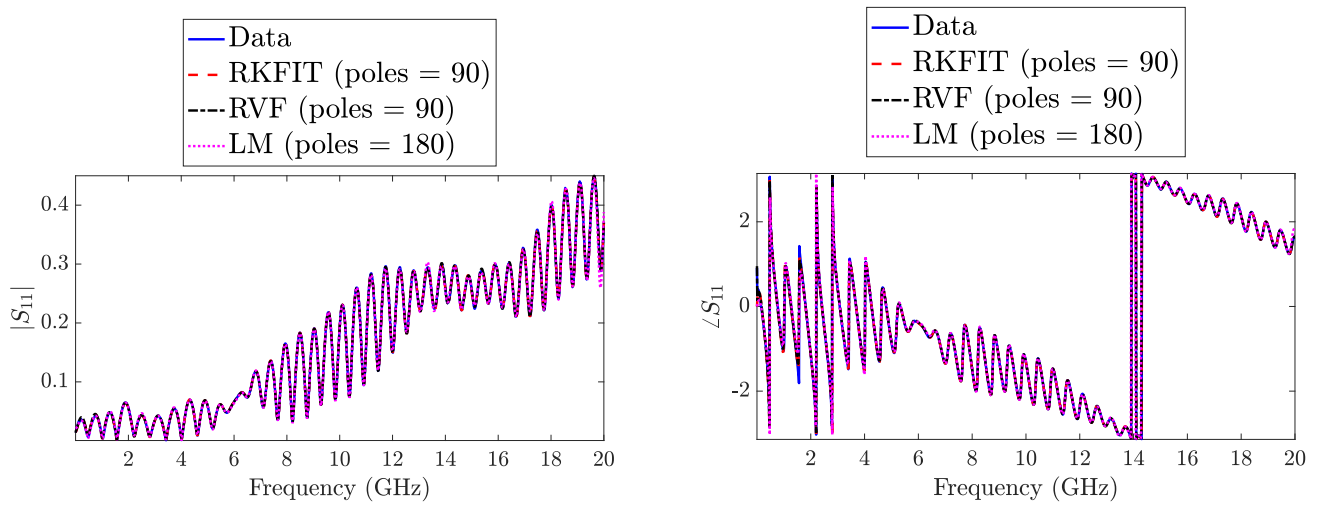


Figure 2. Comparison of RKFIT with RVF and LM models with given data for S_{11} of Section 3.1.

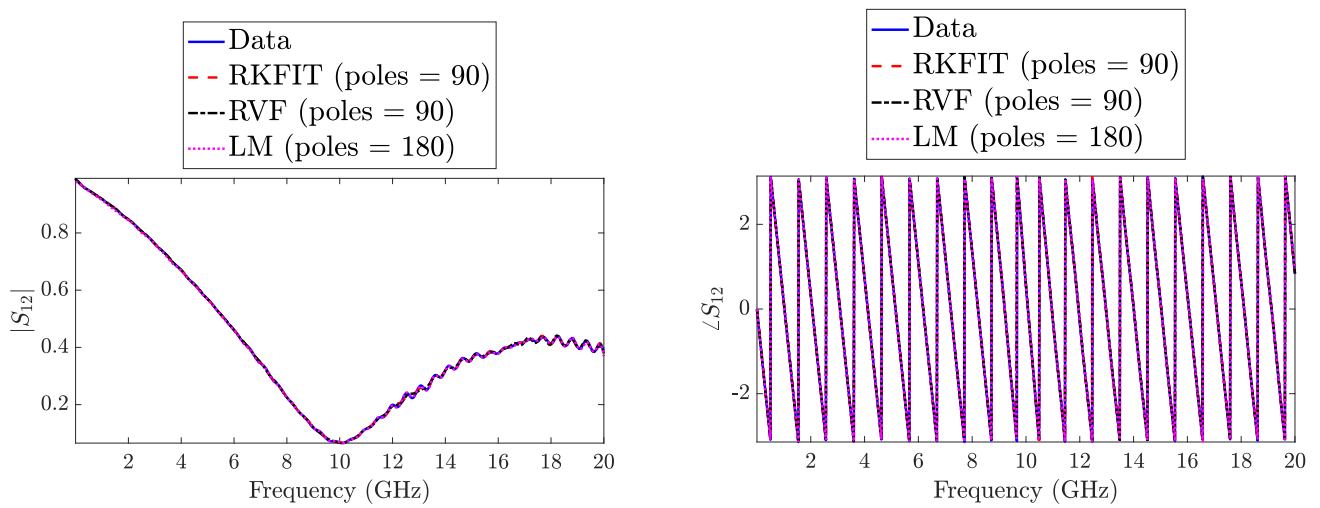


Figure 3. Comparison of RKFIT with RVF and LM models with given data for S_{12} of Section 3.1.

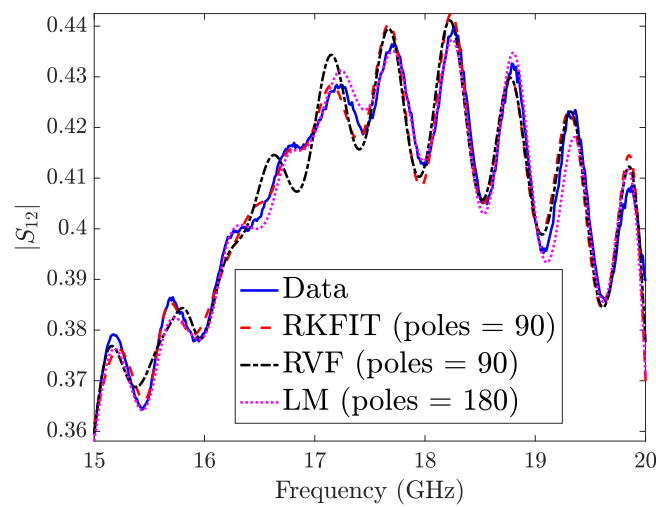


Figure 4. Comparison of RKFIT with RVF and LM models with given data for $|S_{12}|$ of Section 3.1 zoomed.

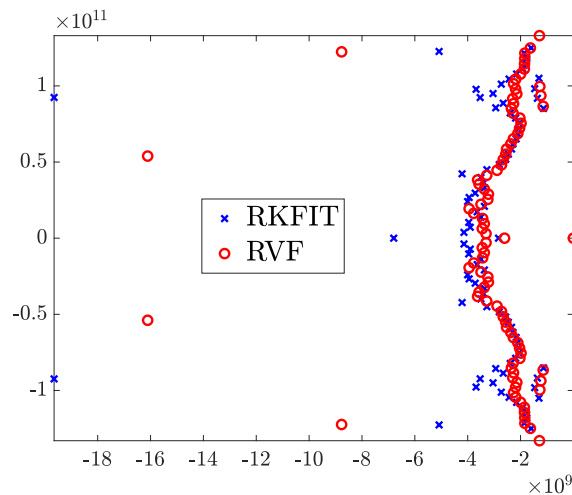


Figure 5. RKFIT and RVF poles.

3.2. 18-Port Network

The second example consists of an 18-port network generated from nine coupled lossless transmission lines (with length $l = 15$ cm) where the per-unit-length capacitance and inductance values were taken from [24]. The data consists of 1000 linearly-spaced frequency points ranging from 0.1 MHz to 10 GHz. The fitting is done with an order of $n = 26$ for both methods. The initial poles for the RKFIT algorithm were set to ∞ using the MATLAB command `inf`, while the initial poles of the RVF algorithm were linearly spaced out across the given bandwidth. As seen from Figure 6, RKFIT is once again able to be more accurate than the RVF algorithm. In addition Figures 7 and 8 show sample plots of the magnitude and phase of S_{11} and S_{12} , and it can be seen that both RKFIT and RVF do a good job capturing the behavior of the data.

For this example, the results shown are from the fourth iterations for both methods since the error does not change in the subsequent iterations. The computed poles obtained from RKFIT and RVF are also shown in Figure 9, as expected all the poles are stable since both methods guarantee stable models by construction.

Finally, Figures 7 and 8 also show fitting done with the LM algorithm, where the number of poles is chosen to gain similar accuracies as RKFIT and RVF. It should be noted that even though the number of poles is greater for LM, due to the way the model is represented, the macromodel from the LM method is more compact than both RKFIT and RVF.

Table 1 shows the average CPU time per iterations it took the methods for both examples, showing that the RKFIT does not need a prohibitive amount of time to get more accurate results compared to the RVF algorithm.

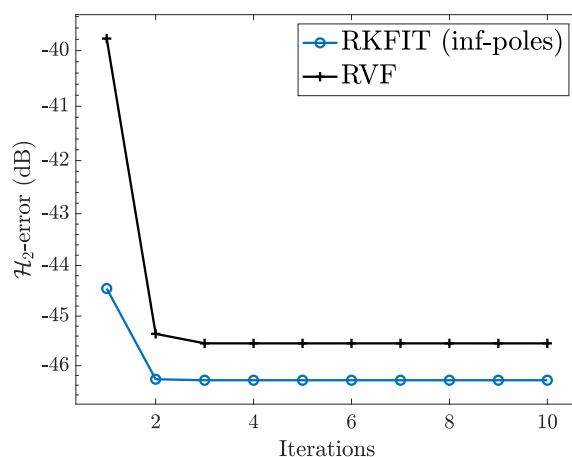


Figure 6. H_2 -error per iteration comparison between RKFIT and RVF algorithms of Section 3.2.

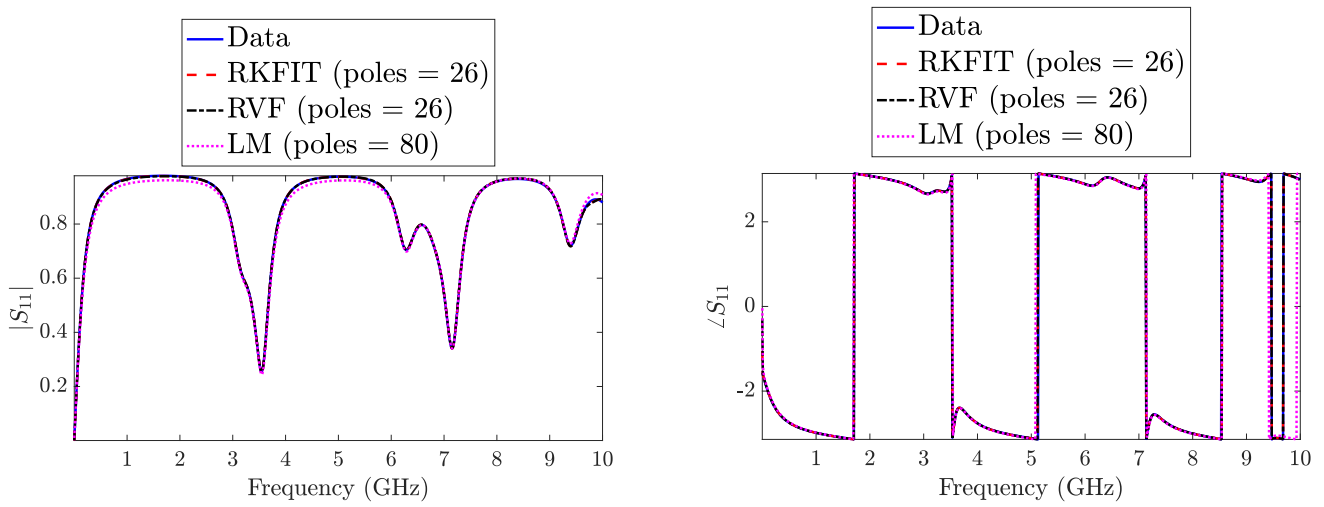


Figure 7. Comparison of RKFIT with RVF and LM models with given data for S_{11} of Section 3.2.

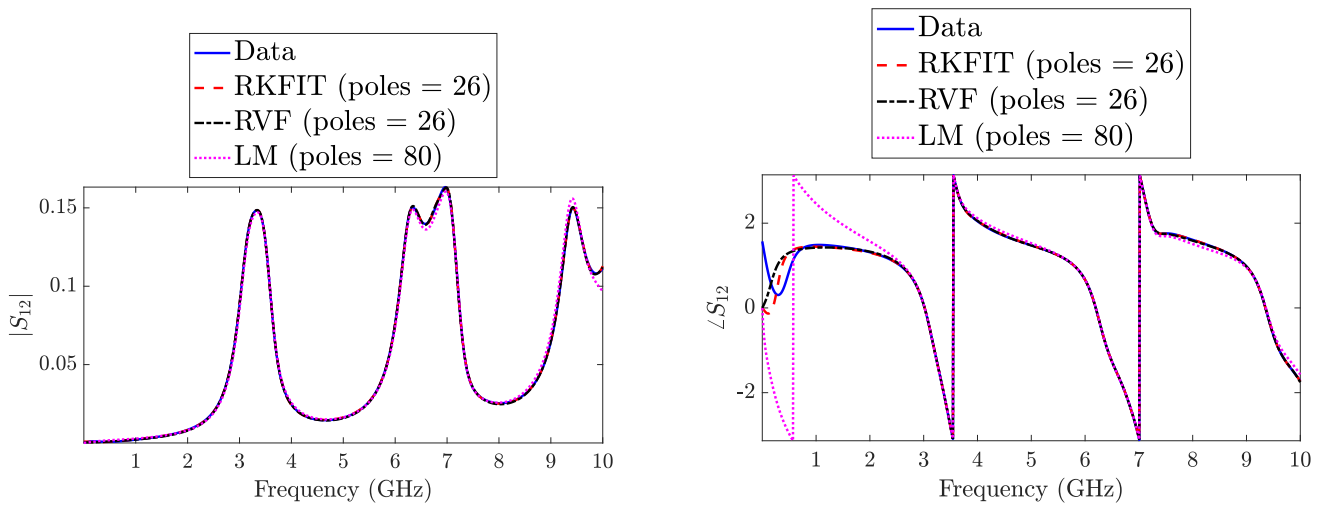


Figure 8. Comparison of RKFIT with RVF and LM models with given data for S_{12} of Section 3.2.

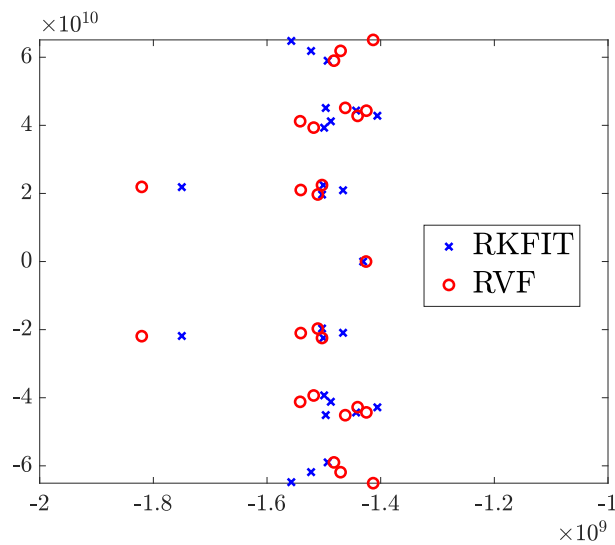


Figure 9. RKFIT and RVF poles.

Table 1. Average time per iteration for Sections 3.1 and 3.2.

Example	Method	Time (s)
4-Port	RVF	0.504
	RKFIT	0.770
18-Port	RVF	1.357
	RKFIT	0.470

4. Conclusions

This paper presented algorithms to model high-speed networks characterized by S-parameters frequency data. The goal of the models is capture the dominant behavior of high speed networks in such a way as to be able to perform time-domain analysis for the purpose of signal integrity. In this work, models of high speed networks were computed using RKFIT technique. This algorithm is a rational curve fitting technique that utilizes rational Krylov to model frequency domain data (obtained by measurements or simulations) of the system under study.

It was shown that the RKFIT has some advantages when compared to the well-established VF method, such as more accurate fitting, less dependence on the choice of the initial poles of the algorithm, and faster convergence. However, it should be noted that just like VF, RKFIT only guarantees the stability of the rational approximation and not the passivity. If any passivity issues arise, they can be tackled using the same methods used for VF.

Author Contributions: Conceptualization, M.S. and A.D.; methodology, M.S. and A.D.; software, M.S.; validation, M.S. and A.D.; formal analysis, M.S.; investigation, M.S.; resources, A.D.; data curation, M.S.; writing—original draft preparation, M.S.; writing—review and editing, M.S. and A.D.; visualization, M.S. and A.D.; supervision, A.D.; project administration, A.D.; funding acquisition, A.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by NSERC Discovery Grant RGPIN-2019-05341.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

LM	Loewner Matrix
RAD	Rational Arnoldi Decomposition
RKFIT	Rational Krylov Fitting
RVF	Relaxed Vector Fitting
VF	Vector Fitting

References

1. Grivet-Talocia, S.; Gustavsen, B. *Passive Macromodeling: Theory and Applications*; Wiley Series in Microwave and Optical Engineering; John Wiley & Sons, Incorporated: Hoboken, NJ, USA, 2016.
2. Gustavsen, B.; Semlyen, A. Rational approximation of frequency domain responses by vector fitting. *IEEE Trans. Power Deliv.* **1999**, *14*, 1052–1061. [[CrossRef](#)]
3. Gustavsen, B. Relaxed Vector Fitting Algorithm for Rational Approximation of Frequency Domain Responses. In Proceedings of the 2006 IEEE Workshop on Signal Propagation on Interconnects, Berlin, Germany, 9–12 May 2006; pp. 97–100. [[CrossRef](#)]
4. Lefteriu, S.; Antoulas, A.C. A New Approach to Modeling Multiport Systems From Frequency-Domain Data. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2010**, *29*, 14–27. [[CrossRef](#)]
5. Ionita, A. Lagrange Rational Interpolation and Its Applications to Approximation of Large-Scale Dynamical Systems. Ph.D. Thesis, Rice University, Houston, TX, USA, 2013.

6. Achar, R.; Nakhla, M.S. Simulation of high-speed interconnects. *Proc. IEEE* **2001**, *89*, 693–728. [[CrossRef](#)]
7. Semlyen, A.; Gustavsen, B. Vector fitting by pole relocation for the state equation approximation of nonrational transfer matrices. *Circuits Syst. Signal Process.* **2000**, *19*, 549–566. [[CrossRef](#)]
8. Gustavsen, B. Improving the pole relocating properties of vector fitting. *IEEE Trans. Power Deliv.* **2006**, *21*, 1587–1592. [[CrossRef](#)]
9. Deschrijver, D.; Haegeman, B.; Dhaene, T. Orthonormal Vector Fitting: A Robust Macromodeling Tool for Rational Approximation of Frequency Domain Responses. *IEEE Trans. Adv. Packag.* **2007**, *30*, 216–225. [[CrossRef](#)]
10. Grivet-Talocia, S.; Bandinu, M. Improving the convergence of vector fitting for equivalent circuit extraction from noisy frequency responses. *IEEE Trans. Electromagn. Compat.* **2006**, *48*, 104–120. [[CrossRef](#)]
11. Ferranti, F.; Rolain, Y.; Knockaert, L.; Dhaene, T. Variance Weighted Vector Fitting for Noisy Frequency Responses. *IEEE Microw. Wirel. Components Lett.* **2010**, *20*, 187–189. [[CrossRef](#)]
12. Beygi, A.; Dounavis, A. An Instrumental Variable Vector-Fitting Approach for Noisy Frequency Responses. *IEEE Trans. Microw. Theory Tech.* **2012**, *60*, 2702–2712. [[CrossRef](#)]
13. Kassis, M.T.; Kabir, M.; Xiao, Y.Q.; Khazaka, R. Passive Reduced Order Macromodeling Based on Loewner Matrix Interpolation. *IEEE Trans. Microw. Theory Tech.* **2016**, *64*, 2423–2432. [[CrossRef](#)]
14. Berljafa, M.; Güttel, S. Generalized rational Krylov decompositions with an application to rational approximation. *SIAM J. Matrix Anal. Appl.* **2015**, *36*, 894–916. [[CrossRef](#)]
15. Berljafa, M. Rational Krylov Decompositions: Theory and Applications. Ph.D. Thesis, The University of Manchester, Manchester, UK, 2017.
16. Börner, R.U.; Ernst, O.G.; Güttel, S. Three-dimensional transient electromagnetic modelling using Rational Krylov methods. *Geophys. J. Int.* **2015**, *202*, 2025–2043. [[CrossRef](#)]
17. Druskin, V.; Güttel, S.; Knizhnerman, L. Compressing Variable-Coefficient Exterior Helmholtz Problems via RKFIT. MIMS EPrint 2016.53. 2016; p. 1. Available online: <http://eprints.ma.man.ac.uk/2511/> (accessed on 20 October 2021).
18. Mouhaidali, A.; Dervout, D.T.; Chadebec, O.; Guichon, J.M.; Silvant, S. Electromagnetic Transient Analysis of Transmission line based on rational Krylov approximation. *IEEE Trans. Power Deliv.* **2020**. [[CrossRef](#)]
19. Morales, J.; Mahseredjian, J.; Ramirez, A.; Sheshyekani, K.; Kocar, I. A Loewner/MPM—VF Combined Rational Fitting Approach. *IEEE Trans. Power Deliv.* **2020**, *35*, 802–808. [[CrossRef](#)]
20. Kabir, M.; Khazaka, R. Macromodeling of Distributed Networks From Frequency-Domain Data Using the Loewner Matrix Approach. *IEEE Trans. Microw. Theory Tech.* **2012**, *60*, 3927–3938. [[CrossRef](#)]
21. Kabir, M.; Khazaka, R. Loewner Matrix Macromodeling for Y-Parameter Data With a Priori D Matrix Extraction. *IEEE Trans. Microw. Theory Tech.* **2016**, *64*, 4098–4107. [[CrossRef](#)]
22. Rational Krylov Toolbox for MATLAB. Available online: <http://guettel.com/rktoolbox/index.html> (accessed on 1 October 2021).
23. The Vector Fitting Website. Available online: <https://www.sintef.no/projectweb/vectorfitting/> (accessed on 1 October 2021).
24. Cangellaris, A.C.; Ruehli, A.E. Model order reduction techniques applied to electromagnetic problems. In Proceedings of the IEEE 9th Topical Meeting on Electrical Performance of Electronic Packaging (Cat. No.00TH8524), Scottsdale, AZ, USA, 23–25 October 2000; pp. 239–242. [[CrossRef](#)]