Abstract—The wave propagation problem in an on-chip interconnect network can be modeled as a generalized eigenvalue problem. For solving a generalized eigenvalue problem, the computational complexity of Arnoldi iteration is at best $O(k^2N)$, with $k$ being the number of dominant eigenvalues and $N$ the matrix size. In this work, we reduce the computational complexity of the Arnoldi iteration from $O(k^2N)$ to $O(N)$, thus paving the way for full-wave extraction of very large-scale on-chip interconnects, of which a typical value of $k$ is in the order of hundreds of thousands. Numerical and experimental results have demonstrated the accuracy and efficiency of the proposed fast eigenvalue solver.

Index Terms—Arnoldi iteration, generalized eigenvalue problem, on-chip interconnects, full-wave analysis, frequency domain.

I. INTRODUCTION

As the clock frequency of microprocessors entered the giga-hertz regime, full-wave models have become increasingly important since it is necessary to analyze the chip response to harmonics that are up to 5 times the clock frequency. In particular, full-wave-based analysis can be used to characterize global electromagnetic coupling through the common substrate and power delivery network. However, there are many modeling challenges associated with on-chip interconnect structures [1]. These challenges include large problem size, large number of non-uniform dielectric stacks with strong non-uniformity, large number of non-ideal conductors, the presence of silicon substrate, highly-skewed aspect ratios, etc. In recent years, solutions of formulations based on both partial differential equations and integral equations have been developed to address these challenges [1]–[12]. However, driven by the continued increase of the complexity of integrated circuit problems, there still exists a continued demand of reducing the computational complexity of full-wave modeling methods.

The wave propagation problem in an on-chip interconnect network can be modeled as a generalized eigenvalue problem $Ax = \lambda B x$ [4], [13]–[15]. Let the matrix size of $A$ and $B$ be $N$. In general, the number of propagation modes that can be supported by an on-chip interconnect structure is much less than $N$ [4], [13]–[15]. Therefore, what is really required is the computation of selected eigenpairs of the generalized eigenvalue system. Among all the existing eigenvalue solvers, the Arnoldi iteration [16] is particularly suited for this computing task. The Arnoldi process generates an orthonormal basis of the Krylov subspace of a significantly reduced size on which the original eigen-system is projected. The overall computational complexity of an Arnoldi process is $O(kN^2 + k^2N)$ where $k$ is the number of significant eigenvalues. The $O(kN^2)$ cost is due to the computation of $B^{-1}Ax$ at each Arnoldi iteration. The $O(k^2N)$ cost is attributed to the orthonormalization of the $k$ Arnoldi vectors that span the Krylov subspace. In each Arnoldi iteration, the current Arnoldi vector is made orthonormal to all previous Arnoldi vectors.

In [14], [15], a direct matrix solver of linear time complexity was developed. The solver allowed for an efficient computation of $B^{-1}Ax$ in $O(N)$ complexity, leading to an efficient solution to the generalized eigenvalue problem. The $O(N)$ complexity for computing $B^{-1}Ax$ in $O(N)$ was achieved by eigenvalue clustering, fast system reduction with negligible computational cost, and fast linear-time solution of the reduced system. As a result, the overall computational complexity of solving a generalized eigenvalue problem was reduced to $O(k^2N)$, which is linear when $k$ is a constant that is not related to $N$. However, for an on-chip interconnect, which involves a large number of conducting wires, even $O(k^2N)$ complexity is too high since $k$ is related to $N$ and is in the order of hundreds and thousands. However, due to the fact that the orthogonalization of Krylov subspace vectors is unavoidable in an Arnoldi process, the time complexity of Arnoldi iteration is at best $O(k^2N)$. Even if one intends to reduce the complexity further, there is no easy way forward.

The main contribution of this paper is the reduction of the computational complexity of Arnoldi iteration method for solving a generalized eigenvalue problem from $O(k^2N)$ to $O(N)$, thus paving the way for the full-wave extraction of very large-scale integrated circuits. The basic idea of this work has been outlined in [17]. In this paper, a detailed derivation of the proposed method is given. A theoretical proof is developed and an extensive number of numerical results are given to provide a rigorous validation of the proposed method.

II. OVERVIEW OF THE EIGENVALUE-BASED ANALYSIS OF ON-CHIP INTERCONNECTS

A typical on-chip interconnect structure is shown in Fig. 1. The generalized eigenvalue problem resulting from a finite-element-based analysis of such a structure can be written as
in which the eigenvalues correspond to the propagation constants $\gamma_j$ and the eigenvectors characterize the transverse electric field $e_t$ and longitudinal electric field $e_z$. Matrices $A$ and $B$ are complex-valued due to the penetration of fields into on-chip conductors. The entries of $A$ and $B$ are given by

$$A_{tt,ij} = \int \Omega \frac{1}{\mu_r} \{\nabla_t \times N_i\} \cdot \{\nabla_i \times N_j\} \, d\Omega,$$

$$B_{tt,ij} = \int \Omega \frac{1}{\mu_r} N_i \cdot N_j \, d\Omega,$$

$$B_{tz,ij} = \int \Omega \nabla_t \xi_j d\Omega, B_{zt,ij} = \int \Omega \frac{1}{\mu_r} \nabla_i \xi_j \cdot N_j \, d\Omega,$$

$$B_{zz,ij} = \int \Omega \frac{1}{\mu_r} \nabla_i \xi_j \cdot \nabla_\xi \xi_j \cdot N_j \, d\Omega,$$

where $\tau_\xi$ denotes the complex permittivity that accounts for conductivity $\sigma$, $N$ represents the edge basis function used to expand the transverse field [18], $\xi$ is the node basis function used to expand the longitudinal field [18], and $\Omega$ is the computational domain.

We can compactly write (1) as

$$Ax = \lambda Bx,$$

where $A$ and $B$ are sparse and of size $O(N)$. The task here is to find $k$ selected eigenpairs of the large sparse matrix system shown in (3), where $k$ is the number of significant modes. The Arnoldi iteration [16] is particularly suited for this computing task. Consider a standard eigenvalue problem $Gx = \lambda x$; a $k$-step Arnoldi process generates an orthonormal basis $\{\nu_j\}_{j=1}^k$ of the Krylov subspace $K_k(\nu_1, G)$ spanned by $\nu_1, G\nu_1, \cdots, G^{k-1}\nu_1$, where $\nu_1$ is an initial unit norm vector. The orthogonal projection of $G$ onto $K_k(\nu_1, G)$ is represented by a $k \times k$ upper Hessenberg matrix $H_k$, the Ritz pairs of which can be used to approximate the eigenpairs of $G$.

The algorithm of a $k$-step Arnoldi process is shown as below:

**Algorithm: A $k$-step Arnoldi process**

1. $\nu_1 = \nu_1 / ||\nu_1||$
2. for $j = 1, 2, \cdots, k$ do
   2.1. $w = G\nu_j$
   2.2. for $i = 1, 2, \cdots, j$ do
   2.3. $h_{ij} = \nu_i^* w$
   4. $w = w - h_{ij}\nu_j$
   5. $\nu_{j+1} = w / ||w||$

The complexity of this algorithm is $O(k^2N)$ if $G$ is sparse. However, in our problem, $G$ is dense because it is equal to $B^{-1}A$ and $B^{-1}$ is dense, as can be seen from (3). Therefore the complexity of a straightforward implementation of the Arnoldi process is at least $O(k^2N + kN^2)$, where the $O(kN^2)$ complexity accounts for the $k$ dense matrix-vector multiplication operations in Step 2.1 shown in (4), and the $O(k^2N)$ complexity accounts for the cost of orthogonalization in Step 2.2. The cost of Step 2.1 was reduced to $O(N)$ by a recent development in [14], [15]. As a result, the complexity of the Arnoldi process is dominated by that of Step 2.2. When $k$ is large, the computation complexity of $O(k^2N)$ could become prohibitively large. In the next Section, we show a method that can remove this computational bottleneck.

III. AN EIGENVALUE SOLUTION OF $O(N)$ COMPLEXITY

In this section, we first construct an alternative eigenvalue solution that is equivalent to the original one in terms of interconnect extraction; we then prove that the solution of the proposed alternative eigenvalue problem is local, from which we show the dependence of an eigenvalue solution on the number of eigenvalues can be eliminated, and hence an $O(N)$ complexity can be achieved.

A. An alternative eigenvalue solution

An examination of the field solution to the eigenvalue problem (1) reveals that the field distribution is global, i.e., fields spread all over the computational domain. As an example, in Fig. 2, we plot the longitudinal and transverse electric field solution of (1) in a typical on-chip interconnect at 1 GHz. The interconnect involves seven layers, the dielectric constants of which are, respectively, 4, 2.9, 2.9, 2.9, 2.9, 2.9, and 4. These layers are, respectively, 0.137, 0.256, 0.256, 0.32, 0.384, 0.576, and 0.972 $\mu$m thick. There are 9 parallel copper wires located in the third layer. The structure is backed by a silicon substrate, the conductivity of which is $10^4$ S/m. The cross-sectional view of the structure can be seen in Fig. 2. We plot in Fig. 2(a) the magnitude of longitudinal electric field $E_x$ at each discretized point in the cross section, from which the longitudinal current can be obtained by $J_x = \sigma E_x$. Clearly, the current is distributed all over the wires and the substrate.
Fig. 2 shows the transverse electric field distribution, which again reveals that the field solution of (1) is global.

Since the solution of (1) is found to be global, there is no apparent way to truncate the computational domain, i.e., reduce the problem size. However, we can remodel the problem such that the field solution becomes local. The details are given as follows.

An on-chip interconnect structure of multiple conductors can be thought of as a p-port network. Its property can be characterized by network parameters such as Impedance (Z)-, Admittance (Y)-, and Scattering (S)-parameters. Take the S-parameter extraction as an example. We first find from (1) all possible modes, i.e. field solutions, that can be supported by an interconnect. We then extract the S-parameters of the interconnect from all these possible field solutions. The procedure can be summarized as the following:

**S-Extraction Procedure**

1. Solve $Ax = \lambda Bx$
   \[x = (e_x, e_y)^T, \lambda = \gamma_m^2\]
2. Perform Superposition
   \[E = \sum_{m=1}^{n} [\alpha_m e_m(x, y)e^{-\gamma_m z} + \beta_m e_m(x, y)e^{\gamma_m z}]\]
   where $e_m = E_t \hat{t} + E_z \hat{z}$, $E_t = j\epsilon_0 c Z_e \gamma$, $E_z = je_z$.
3. Extract $S_{ij} = \frac{V_i - Z_{ref} I_i}{V_j + Z_{ref} I_j}$
   subject to $V_i + Z_c I_i = 0$, for $i \neq j$, $i = 1, 2, \ldots, p$

In the above, $\hat{t}$ denotes a unit vector along the tangential direction, $\hat{z}$ denotes a unit vector along the longitudinal direction, and $Z_{ref}$ denotes the reference impedance. An industry standard $Z_{ref}$ is 50 Ohms. Similarly, Y-parameters of the interconnect can be extracted from the following procedure:

**Y-Extraction Procedure**

1. Solve $Ax = \lambda Bx$
2. Perform Superposition

   \[E = \sum_{m=1}^{n} [\alpha_m e_m(x, y)e^{-\gamma_m z} + \beta_m e_m(x, y)e^{\gamma_m z}]\]
3. Extract $Y_{ij} = \frac{I_j}{V_i}$
   subject to $V_i = 0$, for $i \neq j$, $i = 1, 2, \ldots, p$

In the original eigenvalue solution defined by (1), all interconnects have the implicit boundary condition that they are matched to characteristic impedance of each mode. This is because the wave propagation along the longitudinal direction is analytically incorporated in the derivation of (1) via $e^{-\gamma_z}$ dependence. In other words, an exact absorbing boundary condition is imposed along the wave propagation direction, and hence in this direction, wave propagation has no reflections. This is equivalent to loading the interconnect structure by a matched impedance for each mode. Therefore, the circuit extraction procedure based on the original eigenvalue solution can be explicitly written as

**Original Extraction Procedure**

1. Solve $Ax = \lambda Bx$
   subject to $V_i + Z_c I_i = 0$
2. Perform Superposition
   \[E = \sum_{m=1}^{n} [\alpha_m e_m(x, y)e^{-\gamma_m z} + \beta_m e_m(x, y)e^{\gamma_m z}]\]
3. Extract Circuit Parameters

where $Z_c$ is the characteristic impedance. Comparing the above procedure to the S-Extraction Procedure, it is clear that the original eigenvalue solution based circuit extraction resembles an S-parameter-based extraction of the p-port network with reference impedance $Z_{ref}$ chosen as $Z_c$. Since the solution is found to be global, it suggests that the S-matrix of the interconnect network is dense.

It has been observed that admittance matrix $Y$ arising from the modeling of on-chip interconnects is usually sparse.
It has also been observed that a full-wave integral-equation-based method casts a problem into a system of linear equations of the form \( YI = V \), with \( Y \) being dense, whereas a partial-differential-equation-based method is able to model the same problem by \( YV = I \), with \( Y \) being sparse. These observations suggest that a \( Y \)-parameter-based extraction procedure can potentially render the solution local. Therefore, we constructed the following alternative procedure for interconnect extraction:

**Alternative Extraction Procedure**

1. Solve \( Ax = \lambda Bx \) subject to \( V_i = 0, \text{ for } i \neq j, \text{ for } i = 1, 2, \ldots, p \)
2. Perform Superposition
   \[ E = \sum_{m=1}^{n} [\alpha_m e_m(x, y)e^{-\gamma_m z} + \beta_m e_m(x, y)e^{\gamma_m z}] \]
3. Extract Circuit Parameters

A comparison of the above procedure to the \( Y \)-Extraction Procedure shows clearly that the two procedures are equivalent since they differ only in the ways the port boundary conditions are incorporated. In the Alternative Extraction Procedure, the port boundary conditions are incorporated at the stage of eigenvalue solution, whereas in the \( Y \)-Extraction Procedure, the port boundary conditions are incorporated at the stage of circuit extraction. Since the incorporation of boundary conditions at an earlier stage or at a later stage should not affect the solution, the alternative extraction procedure is equivalent to the original extraction procedure, from the perspective that the two procedures generate the same network parameters.

As a result, instead of solving (1), the solution of which is found to be global, we transform (1) to an alternative eigenvalue solution as shown below.

**Alternative Eigenvalue Solution**

Solve \( Ax = \lambda Bx \) subject to \( V_i = 0, \text{ for } i \neq j, \text{ for } i = 1, 2, \ldots, p \) \hfill (5)

In this solution, we ground all the ports except for one port (or a few selected ports) whose modes are to be extracted, i.e., we let each port float in turn while grounding the rest of the ports. This leads to \( p \) eigenvalue problems. We then find the solution of each eigenvalue problem. The total \( E \) field can then be obtained as a linear combination of all these possible solutions, from which the network parameters, such as \( S \)-parameters, can be extracted. The grounding of each port is achieved by explicitly setting the tangential electric field of corresponding edges to zero. By doing so, we enforce the port boundary condition in (5) without altering the original physical structure.

The solution of the alternative eigenvalue solution (5) is found to be local. In Fig. 3, we plot the solution of (5) in the same interconnect structure shown in Fig. 2. We let the 6th conductor float and ground other conductors. We then observe the field solution extracted from (5) at 1 GHz. As shown in Figs. 3(a) and 3(b), both longitudinal and transverse electric fields exhibit a fast decay. In Figs. 3(c) and 3(d), we plot the longitudinal and transverse electric field distribution at 20 GHz. Once again, a fast decay is observed.

In addition to the numerical proof shown in Fig. 3, we also theoretically proved that the solution of (5) is local, the detail of which is given in the following section.

**B. Proof on the locality of the alternative eigenvalue solution**

From the second row of (1), it can be seen clearly that the following equation satisfies

\[ B_{zt} e_t + B_{zz} e_z = 0. \]

In deriving (1), the following transformation was used [4], [13]

\[ e_t = -j \gamma E_t, \quad e_z = -j E_z. \]

Substituting (7) into (6), we obtain

\[ E_z = -B_{zz}^{-1} B_{zt} (\gamma E_t). \]

From \( E_t \), one can obtain the voltage at one end of the wire (a port) by performing a line integral from the terminal to the ground,

\[ V = \int E_z dl. \]

Due to the fact that all field components have \( e^{-\gamma z} \) dependence in a structure seed [4], [13], we have

\[ \gamma E_t = \frac{\partial E_z}{\partial z}. \]

Therefore, from \( \gamma E_t \), one can obtain the voltage difference between the two ends of the wire across a unit length

\[ V_1 - V_2 = \int \gamma E_z dl. \]

In addition, from \( E_z \), one can obtain the current flowing into the wire by performing an area integral of the current density over the wire cross section

\[ I = \int (j \omega z + \sigma) E_z dS, \]

in which both displacement and conduction currents are included. From the aforementioned analysis, \( E_z \) in (8) relates to voltage, \( \gamma E_t \) relates to the voltage difference across a unit length, and \( E_z \) relates to current. Therefore, although (8) is a field-based equation, it has an analogous circuit meaning [20]. The circuit interpretation of (8) is

\[ I = \frac{V_1 - V_2}{Z_{in}}, \]

which reveals the impedance experienced by a current \( I \) given a potential difference \( V_1 - V_2 \) at the two ends of a wire of a unit length.

When computing the alternative eigenvalue solution (5), we ground all the ports (wires) except for one port. Once the port (wire) is grounded, the tangential electric field \( E_t \) on each edge along the grounding path is set to zero. Hence, \( \gamma E_t \) is zero. As a result, the voltage difference across the wire...
length is set to zero as can be seen from (11). Hence, from (13), the current flowing into the grounded port is zero. As a result, in the alternative eigenvalue solution (5), the currents flowing into all the ports are zero except for the port that is not grounded. Therefore, except for the wire that is excited, all the other wires do not carry current. In other words, no current is coupled to other wires no matter how close they are to the wire that is excited, because the voltage difference across the length of these wires is enforced to be zero. As a result, in the alternative eigenvalue solution (5), the current loop is forced to be formed between the active conductor and the physical ground only instead of adjacent wires, and hence localized. Therefore, the locality of the alternative eigenvalue solution is proved.

As a validation of the above proof, we did the following experiment. We considered the same test-chip interconnect shown in Figs. 2 and 3. In Fig. 4 and Fig. 5, we plot the voltage and current distribution simulated from the original eigenvalue solution (1), and the alternative eigenvalue solution (5) respectively. The label of $x$-axis in Fig. 4 and Fig. 5 denotes the index of the conductors, and the label of $y$-axis represents the magnitude of voltage or current. As can be seen from Fig. 4, in the original eigenvalue solution, the voltage and current distribution over the conductors is global, i.e., no voltages and currents are significantly smaller than others. However, in Fig. 5, we see clearly that the current flowing into each conductor is zero except for the conductor that is not grounded. This behavior is in excellent agreement with our theoretical proof. In addition, we compared the $S$-parameters obtained from the alternative eigenvalue solution and those simulated from the original eigenvalue solution. Denoting the former by $\mathbf{S}$, and the latter by $\tilde{\mathbf{S}}$, and using Frobenius norm, the difference $||\mathbf{S} - \tilde{\mathbf{S}}||_F/||\mathbf{S}||_F$ was found to be 3.48%. Therefore, the proposed alternative eigenvalue solution localizes the field solution without sacrificing circuit extraction accuracy.

Since the solution of the alternative eigenvalue problem is local, this property can be utilized to significantly reduce the computational complexity of an eigenvalue solution, which is described in the next section.

C. Windowing technique

If the field solution becomes zero in the region that is away from the active conductor, there is no need to simulate fields in that region. Hence, for each eigenvalue problem defined by (5), we do not have to simulate the entire computational domain. We only have to simulate a small window in which fields are nonzero. We cannot do this in the framework of the original eigenvalue solution since the fields are global in that scenario. By developing an alternative eigenvalue solution described in Section III.A, we essentially localize
the solution of the system, and hence are able to simulate a sequence of much smaller problems to obtain the solution of the original large problem without any reduction cost. One might argue that if the original solution is global, there is no way to localize it. Note that here we have already remodeled the problem such that the solution of our problem becomes local. A basic windowing procedure is given below.

Windowing Procedure
1. Adaptively determine the window size
2. Compute the field solution inside the window based on (5)
3. Slide the window from left to right
4. Repeat the computation until the whole structure is solved
5. Perform superposition
6. Extract circuit parameters

The window size is adaptively decided by enlarging the window size progressively until the solution converges. As shown in Section III.B, in the proposed alternative eigenvalue solution, current loop is shrunk to the loop formed between the active conductor and ground. If the ground is perfect, it is known that fields on the ground only concentrate in a small region having a size similar to the active conductor; if the ground is not perfect, it can be viewed as a resistance ($R$)-inductance ($L$) network, the area that the fields/currents can spread over is also limited so that the impedance experienced by the current can be minimized (Note that larger area results in a larger $\omega L$). Therefore, the window size is, in general, very small.

D. Complexity analysis

Assuming $L$ windows are used, and each window includes $M$ number of unknowns. The proposed eigenvalue solver solves $L$ eigenvalue problems each having size $M$. In each eigenvalue problem, $k$ is $O(1)$ since in each window, we only let one or a few conductors float while grounding all the other conductors. As a result, the computational complexity for simulating the eigenvalue problem in each window is $O(M)$. Since there are $L$ windows in total, the total computational complexity of the proposed solver is $O(LM) \sim O(N)$, which does not depend on the total number of eigenvalues in the original system.

IV. NUMERICAL AND EXPERIMENTAL VALIDATION

We simulated a number of on-chip interconnect structures to evaluate the performance of the proposed eigenvalue solver. The first example is the same structure examined in Section II, the cross section of which can be seen in Fig. 2 and Fig. 3. The

Fig. 4: Voltage and current distribution in log scale simulated from the original eigenvalue solution (1) at 1 GHz.

Fig. 5: Voltage and current distribution in log scale simulated from the alternative eigenvalue solution (5) with all the conductors grounded except for the 5$^{th}$ conductor at 1 GHz.
structure involved 9 wires and 18 ports in total. We simulated the structure using the original eigenvalue solution as well as the proposed alternative one. In Fig. 6, we plot simulated \( S_{11} \) and \( S_{1,17} \), where port 1 was located at the near end of the leftmost wire, and port 17 was located at the near end of the rightmost wire. An excellent agreement can be observed, which validates the proposed alternative eigenvalue solution.

The second example is a test-chip interconnect structure [8], the cross-sectional-view of which is shown in Fig. 7. The structure was 2000 \( \mu \)m long, consisting of 11 inhomogeneous layers. It involved 12 parallel returns in M1 and M3 layers, respectively. These returns were 1.05 \( \mu \)m wide and 1 \( \mu \)m apart, and overlapped with each other vertically. Two wires were placed in the center of M2. The far-ends of the two center wires in M2 were left open. The \( S \)-parameters at the near-end of one M2 wire were extracted by the proposed fast eigenvalue solver. The physical ground was located at the bottom of the substrate. The window size was chosen between 14 \( \mu \)m and 16 \( \mu \)m. In total, six windows were used. As can been from Fig. 8, the simulated \( S \)-parameters agree very well with those generated by the conventional solver reported in [4], [13], the result of which was shown to agree with the measured data.

The last example was used to test the performance of the proposed fast eigenvalue solver in solving large-scale on-chip interconnects. The structure involves seven dielectric stacks, the dielectric constants of which are, respectively, 4, 2.9, 2.9, 2.9, 2.9, 2.9, and 4. These layers are, respectively, 0.972, 0.576, 0.384, 0.32, 0.256, 0.256, and 0.137 \( \mu \)m thick. The

Fig. 7: Cross-sectional view of a test-chip interconnect structure.

Fig. 8: Simulation of a test-chip interconnect. (a) \( |S_{11}| \). (b) \( S_{11} \) Phase(degree).
third, fifth, and seventh dielectric stack is populated with 10 - 200 parallel conductors. The smallest problem hence involves 30 conductors, and the largest problem involves 600 conductors. Each conductor is 0.5 µm wide, and separated from each other by 2.5 µm in each metal layer. These conductors, i.e., interconnect wires, do not overlap with each other vertically. The distance from the leftmost (rightmost) conductor to the left (right) boundary is 10 µm. The computational domain was discretized into triangular elements, resulting in 24,651 to 476,091 unknowns. If we use the number of conductors to estimate the number of significant eigenvalues $k$, the range of $k$ is from 30 to 600. We simulated the structure at 20 GHz frequency. The ground was located at the bottom. The left, right, and top boundary conditions were set to be open.

First, we used the 30-conductor case to examine the accuracy of the proposed solver with respect to window size. It was shown that when the window included 15, 21, and 27 conductors, the average error of the $S$-matrix was 3.96 %, 1.54 %, and 0.75 % respectively. The 21-conductor window was then decided as a window size across the simulations. In Fig. 10(a), we plot the Arnoldi iteration time, which is the time of Step 2 in (4), with respect to unknowns. Clearly, the performance of the proposed solver is linear, whereas the conventional solver developed in [15] scales as $O(k^2 N)$. Since $k$ increases with $N$ in this case, the performance of the conventional solver is much worse than linear. In Fig. 10(b), we plot the total CPU time with respect to the number of unknowns, which includes the factorization time needed.

Fig. 10: Simulation of a large-scale on-chip interconnect involving 30 - 600 conductors. (a) Arnoldi Time. (b) Total CPU time. (c) Average percentage error. (d) An error plot of $S$-matrix for 114-conductor (228-port) case.

Fig. 9: A large-scale interconnect having 30 to 600 conductors.
for the computation of $B^{-1}A$. Again, the proposed solver demonstrates a linear complexity that is independent of $k$. In Fig. 10(c), we use the results obtained from the conventional solver as a reference, and plot the average error of the extracted $S$-parameters with respect to the number of unknowns. The error is evaluated by using $\frac{|S_1 - S_\text{est}|}{\min_i |S_{\text{ref},i}|}$. Clearly, good accuracy can be observed in the entire range. In Fig. 10(d), we take the 114-conductor (228-port) case as an example, and plot the error of every $S$-matrix element. The ports are ordered layer by layer from M3 to M7, and assigned to the near-end and far-end of each conductor. Excellent accuracy can be observed. In addition, we used $\|S - S_1\|_F / \|S\|_F$ to assess the error, where $S$ is generated by the conventional solver, and $S_1$ is generated by the proposed fast solver. The error was shown to be 4.83%. Therefore, the proposed solver reduces the computational complexity of a generalized eigenvalue problem without sacrificing accuracy. In Fig. 10(a) and (b), the results for the conventional solver were only generated up to 261 conductors since it could not solve larger problems in reasonable run times.

V. SUMMARY

The computational complexity of a generalized eigenvalue problem generally depends on the number of eigenvalues $k$. A large-scale on-chip interconnect network involves a large number of conductors, and hence a large number of eigenvalues. As a result, state-of-the-art eigenvalue solutions become computationally prohibitive when analyzing large-scale interconnect structures. In this work, we transform the original eigenvalue solution to an alternative one. We show that the alternative eigenvalue solution is equivalent to the original one in terms of interconnect extraction. Most important, we prove that the alternative eigenvalue solution is local, and hence can be utilized to significantly reduce the computational complexity of an eigenvalue analysis. As a result, we are able to decompose the original large-scale eigenvalue problem into $L$ small eigenvalue problems, each having a constant number of dominant eigenvalues. Hence, we reduce the computational complexity of the generalized eigenvalue problem from $O(k^2 N)$ to $O(N)$. Numerical experiments demonstrated superior performance of the proposed method for solving large-scale on-chip interconnects.

ACKNOWLEDGMENT

This work was supported by NSF under award No. 0747578 and No. 0702567.

REFERENCES


