

# High-Order Moment-Matching MOR with Impedance Boundaries for Signal Integrity Analysis

Matthew B. Stephanson

ANSYS, Inc., San Jose, California  
matt.stephanson@ansys.com

**Abstract**—This paper presents an extension of high-order moment-matching model order reduction to frequency-dependent impedance boundary conditions. Such boundary conditions are essential for accurately and efficiently modeling conductors at high frequency, where skin effect is significant. These boundary conditions involve complicated transcendental functions, whereas previous MOR methods assume polynomial dependence. Automatic differentiation is used to easily and accurately calculate the higher derivatives of such functions. Substantial improvement is shown, not only compared to discrete frequency sweeps, but also compared to non-moment-matching S matrix interpolation.

## I. INTRODUCTION

Model order reduction (MOR) is a frequently-used technique to develop efficient approximations to discretizations of Maxwell's equations that depend on one or more parameters (e.g., frequency or material properties). Although there are many methods to accomplish this, they can broadly be classified according to how well the reduced-order solution matches the full-order solution (order) and how many points in parameter space the two solutions match (single- vs. multi-point). High-order methods, in which not only the value but also the derivatives of the solutions are equal, can be efficient for systems with smooth parameter dependence because the incremental cost of each order is a matrix forward-backward substitution, rather than a factorization [1]. The disadvantage is that the derivatives of the system matrix with respect to the parameters must be known.

For signal integrity (SI) analysis of electronic packages and circuit boards, this presents a problem. Modeling of high-speed interconnects requires analysis up to tens of gigahertz, where losses from skin effect significantly affect performance. One approach to capture this effect is to model the volume inside the conductors with a mesh fine enough to resolve the decaying field. The system matrix entries are then second-order polynomials of frequency and the derivatives are easily calculated. Alternatively, an impedance boundary condition (IBC) can be used, but the matrix entries are no longer polynomials.

This work shows that the accuracy of an IBC can be achieved with the ease-of-implementation of a polynomial formulation by using automatic differentiation to calculate the higher-order derivatives of the surface impedance. This is a method for transforming the computer code of a function so

that it not only calculates the value of the function, but also the derivatives with respect to the inputs. In contrast to finite difference methods, there is no discretization error, only round-off error. There are also advantages compared to symbolic differentiation, e.g., it is unnecessary to translate the computer code into a single mathematical expression.

The resulting method has several advantages for network analysis compared to S matrix interpolation. First, the use of higher-order moment-matching allows the same accuracy and bandwidth to be achieved with fewer matrix factorizations. Second, the reduced-order model automatically inherits several desirable properties from the original system, including passivity and stability [2]. Third, although not pursued here, knowledge of the S parameter's exact derivatives could be used to provide better error control choosing the moment-matching points adaptively.

## II. BACKGROUND

### A. Finite element modeling of SI problems

The finite element method is used to discretize Maxwell's equations, as applied to the problem geometry. For non-dispersive materials and ohmic conductors, this results in a matrix equation of the form:

$$A(s)\mathbf{E} = (S + sY + s^2M)\mathbf{E} = \mathbf{J}(s), \quad (1)$$

where  $s = j\omega$  is the complex frequency, and  $\mathbf{J}$  and  $\mathbf{E}$  are the current excitation and electric field solution vectors. The matrices  $S$ ,  $Y$ , and  $M$  are the stiffness, admittance, and mass matrices, and depend, respectively, on the materials' permeability  $\mu$ , conductivity  $\sigma$ , and permittivity  $\epsilon$ . Inside good conductors, the fields decay exponentially on a length scale given by the skin depth  $\delta = \sqrt{2/\omega\mu\sigma}$  (e.g., about 0.3  $\mu\text{m}$  for copper at 40 GHz). With trace cross sections on the order of tens of microns or more, this requires an extremely dense mesh to accurately resolve.

Because the traces and ground planes are typically wide compared to their thickness, it is common to model them with a layered media impedance boundary [3]. For a single layer, this has the form:

$$Z_{IBC} = Z_0 \frac{1 + \Gamma e^{-2\gamma T}}{1 - \Gamma e^{-2\gamma T}}, \quad (2)$$

where  $Z_0 = \sqrt{\mu/\epsilon}$  is the intrinsic impedance,  $\Gamma$  is the reflection coefficient from the surrounding dielectric,  $\gamma = j\omega\sqrt{\epsilon\mu}$  is the propagation constant, and  $T$  is the metal thickness. For

good conductors,  $\Gamma \approx 1$ ,  $\gamma \approx (1 + j)/\delta$ , and  $Z_0 \approx \gamma/\sigma$ , leading to a rather complicated expression for  $Z_{IBC}$ . It is this expression (actually, its inverse) that the matrix entries now depend on and which must be differentiated for moment-matching.

## B. Projection-based model order reduction

A projection-based reduced order model is obtained by Galerkin-testing the  $N \times N$  system matrix  $A$  with the  $N \times M$  reduced-order basis vectors  $V$ :

$$\widehat{A}(s) = V^H A(s) V. \quad (3)$$

The reduced-order matrix equation is then:

$$\widehat{A}(s) \widehat{\mathbf{E}} = V^H \mathbf{J}, \quad (4)$$

$$\mathbf{E} \approx V \widehat{\mathbf{E}}. \quad (5)$$

Because  $M \ll N$ , this equation can be solved much more rapidly than the original. In order for  $\widehat{\mathbf{E}}$  and  $\mathbf{E}$  to match up to the  $p$ -th order at frequency  $s_0$ , it can be shown that it is sufficient that the column space of  $V$  equal  $\text{span}(\mathbf{E}|_{s_0}, \mathbf{E}'|_{s_0}, \dots, \mathbf{E}^{(p)}|_{s_0})$  [4]. Applying the Leibniz product rule to  $A\mathbf{E} = \mathbf{J}$  shows that these are given by:

$$\mathbf{E}^{(p)} = A^{-1} \left[ \mathbf{J}^{(p)} - \sum_{k=1}^p \binom{p}{k} A^{(k)} \mathbf{E}^{(p-k)} \right]. \quad (6)$$

## C. Automatic differentiation

Automatic differentiation consists of analyzing each line of code that involves a dependent variable, for example  $Z_{IBC}$ , and augmenting it with a line that calculates the derivative with respect to the independent variable (frequency) as well [5]. Thus, for example, calculation of  $\delta$  would include:

$$\delta'(f) = \left( \frac{1}{f\pi\mu\sigma} \right)^{-1/2} \cdot \frac{-1}{f^2\pi\mu\sigma}. \quad (7)$$

The power of automatic differentiation is that this augmentation can be done automatically, either by source code translation or operator overloading. Furthermore, the process can be applied recursively to obtain arbitrarily high orders.

To see how this can be accomplished, consider first the simplest case, computing only the first derivative. Using the overloading approach, this means that every value  $u$  that depends on frequency is replaced by a class that contains both the value and its derivative with respect to frequency:  $\langle u, u' \rangle$ . Then, the basic mathematical operators are overloaded to use this new class. For example, multiplication and the exponential function become:

$$\langle u, u' \rangle \times \langle v, v' \rangle = \langle uv, uv' + u'v \rangle, \quad (8)$$

$$\exp \langle u, u' \rangle = \langle \exp u, u' \exp u \rangle. \quad (9)$$

To extend this to higher orders, the derivatives themselves can be treated as value were above, up to some maximum order, and the rules applied recursively. This can be illustrated

by working out the exponential function up to second order and verifying that it indeed gives the correct result:

$$\begin{aligned} \exp \langle u, u', u'' \rangle &= \langle \exp u, \langle u', u'' \rangle \times \exp \langle u, u' \rangle \rangle \\ &= \langle \exp u, \langle u', u'' \rangle \times \langle \exp u, u' \exp u \rangle \rangle \quad (10) \\ &= \langle \exp u, u' \exp u, [(u')^2 + u''] \exp u \rangle. \end{aligned}$$

In general, when applying a unary function  $f$  to a value and its  $n$  first derivatives, the chain rule provides:

$$\begin{aligned} f \langle u, u^{(1)}, u^{(2)}, \dots, u^{(n)} \rangle \\ = \langle f(u), f' \langle u, \dots, u^{(n-1)} \rangle \times \langle u^{(1)}, \dots, u^{(n)} \rangle \rangle. \quad (11) \end{aligned}$$

Notice that the argument to  $f'$  has one lower order, so that eventually we reach the scalar case the the recursion terminates.

There are two significant advantages to this approach. The first is the ease of understanding and maintaining the code. If we define ‘‘reduction’’ and ‘‘differentiation’’ operators as:

$$\mathbf{u} \equiv \langle u, u^{(1)}, \dots, u^{(n)} \rangle, \quad (12)$$

$$\mathbf{R}\mathbf{u} \equiv \langle u, u^{(1)}, \dots, u^{(n-1)} \rangle, \quad (13)$$

$$\mathbf{D}\mathbf{u} \equiv \langle u^{(1)}, \dots, u^{(n)} \rangle, \quad (14)$$

then many functions can be programmed using expressions that look nearly equivalent to what one might find in a math textbook, for example,

$$\exp \mathbf{u} = \langle \exp u, \exp(\mathbf{R}\mathbf{u}) \cdot \mathbf{D}\mathbf{u} \rangle, \quad (15)$$

$$\sin \mathbf{u} = \langle \sin u, \cos(\mathbf{R}\mathbf{u}) \cdot \mathbf{D}\mathbf{u} \rangle, \quad (16)$$

$$\cos \mathbf{u} = \langle \cos u, -\sin(\mathbf{R}\mathbf{u}) \cdot \mathbf{D}\mathbf{u} \rangle, \quad (17)$$

$$\mathbf{u}^p = \begin{cases} \langle u^p, p \cdot (\mathbf{R}\mathbf{u})^{p-1} \cdot \mathbf{D}\mathbf{u} \rangle & \text{if } p \neq 0 \\ \langle 1, \mathbf{0} \rangle & \text{if } p = 0. \end{cases} \quad (18)$$

The second advantage is ease of testing. Simply comparing the first derivative, calculated using recursive automatic differentiation, with a few values calculated manually is sufficient to verify the code’s correctness, because this simple test nonetheless achieves 100% code coverage.

Set against this, the primary disadvantage is efficiency. Multiplication can be done quadratically in terms of the maximum order, using the Leibniz rule, making the entire recursion cubic. There are two ways to mitigate this. One is to obtain and learn one of the many third-party automatic differential libraries, such as those listed at [www.autodiff.org](http://www.autodiff.org) [6]. These use more optimized algorithms, but few, if any, support complex numbers, making IBC calculations difficult. The second is to note, as explained in detail below, that derivative calculation can be made a minuscule part of the overall computation, so that an especially efficient implementation is actually unnecessary.

Because many of the system matrix entries have the same frequency dependence, up to a multiplicative constant, it would be inefficient to calculate them entry by entry. Rather, the

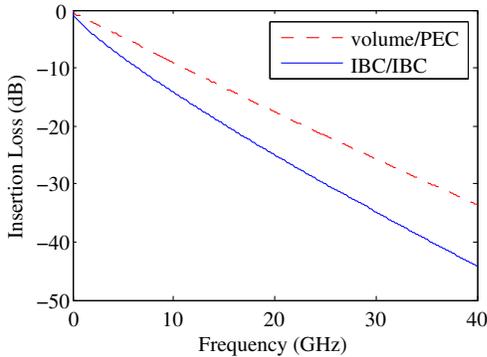


Fig. 1. Insertion loss for two different conductor models: (dashed red) volume conductor for traces, PEC for ground (solid blue) IBC for both.

matrix can be decomposed into a linear combination of scalar functions and constant matrices:

$$A(s) = \sum_i \phi_i(s) A_i. \quad (19)$$

The number of functions is typically less than a few dozen, e.g., one IBC-related function for each unique pair  $(T, \sigma)$ . The derivatives  $\phi_i^{(p)}$  can therefore be rapidly calculated, even with an unoptimized implementation, at which point evaluating  $A^{(p)}(s)$  becomes trivial. Furthermore, this decomposition permits rapid online computation of the reduced-order matrix  $\hat{A}$ , by precomputing each  $V^H A_i V$ .

### III. RESULTS AND CONCLUSIONS

#### A. Long microstrip

To demonstrate the benefit of MOR with IBCs, a simple test project consisting of a very long trace, as might be found on a PCB, is used. The trace is  $127 \mu\text{m}$  wide,  $48 \mu\text{m}$  thick, and 30 cm long. The ground plane underneath is  $33 \mu\text{m}$  thick, separated by  $69 \mu\text{m}$ -thick FR4 ( $\epsilon_r = 4.4$ ,  $\tan \delta = 0.02$ ). The S parameters are extracted to 40 GHz using the Sentinel-PSI solver in ANSYS SIwave, which uses a 5th-order moment matching algorithm.

Four different analysis methods are considered. The first is using IBC on both the trace and ground plane and using the MOR with automatic differentiation described above. Second, MOR is also used, but with the simplest conductor modeling that gives a polynomial system matrix and a similar number of unknowns. In particular, the trace is modeled as a volume conductor and the ground plane is PEC; note that the mesh is not heavily refined, so this is expected to result in less loss. Third, IBCs are again used on all metals, but S matrix interpolation is used for the frequency sweep. Finally, for reference, the IBC results are calculated without any fast-sweep algorithm.

The resulting insertion loss is shown in Fig. 1. As expected, the model using volume conductivity and PEC shows significantly less loss at high frequency, up to 10 dB at 40 GHz. The

Table 1: Performance Results

Model	Sweep Method	Freq. Samples	Time (min.)	Memory (GB)	% Error RMS/Peak
IBC	MOR	33	33.9	9.00	0.06/0.48
$\sigma$ /PEC	MOR	31	27.6	8.50	65/105
IBC	S interp.	262	112.3	3.93	3.6/18.5
IBC	none	800	344.9	3.95	—/—

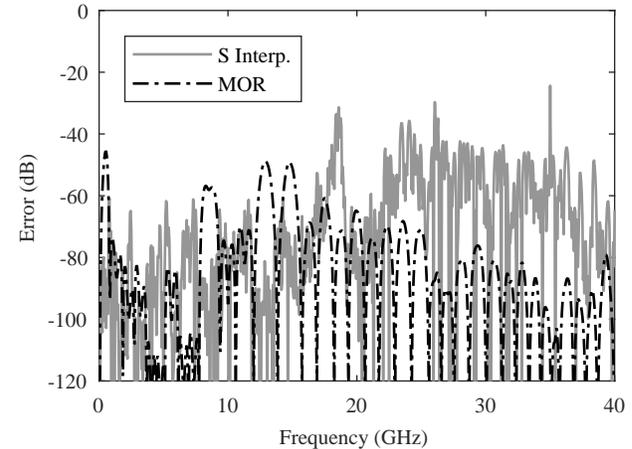


Fig. 2. Error in S matrix for long microstrip, using S matrix interpolation versus high-order moment matching MOR.

performance measurements are shown in Table 1. The error is defined as  $\|S - S_{ref}\|_F / \|S_{ref}\|_F$ . Two conclusions can be drawn. First, for a problem with a given number of unknowns, the addition of IBCs has no significant impact on the runtime of the MOR algorithm, compared to the polynomial case. Second, the use of higher-order moment matching allows for a significantly reduced number of frequency samples compared to S matrix interpolation. In addition, Fig. 2 shows the error in the two frequency sweep methods. It can be seen that the MOR results has much more spacing between the nulls at the sample points (where the error is exactly zero), demonstrating the utility of high-order moment matching in providing more bandwidth per sample.

#### B. Flip-chip package

Figure 3 shows a more complicated example, a flip-chip package with four layers, four signal conductors, and three power/ground nets. The fact that some parts of the signal traces lack a nearby ground reference, combined with the many cutouts in the ground plane, lead to many dips in the insertion loss. This is a challenging situation for methods that interpolate the S parameters based on value alone, without any knowledge of the derivatives.

Table 2 shows the performance results, while Figs. 4 and 5 illustrate the insertion loss and S parameter error, respectively. Once more, the superior runtime and accuracy performance, at the cost of more memory, can be seen. In this case, the error nulls at the sample frequencies are not evident because of numerical non-reproducibility, caused by, e.g., differing

unknown ordering and using SIMD instructions on unaligned data [7]. Nevertheless, besides the number of samples being smaller, the fact that the average error level is lower also implies larger bandwidth per sample, because the error does not increase as rapidly when the frequency moves away from a sample. With higher-order moment matching, not only is the error zero at the samples, but its derivatives are also (apart from numerical rounding). It is also worth noting that of the 6.5 minutes spent computing the frequency sweep, a mere 21 ms was spent performing automatic differentiation. This validates the earlier observation that, for this particular application, a simple and understandable implementation may be preferable to a highly-optimized one.

Table 2: Flip-Chip Performance Results

Sweep Method	Frequency Samples	Time (minutes)	Memory (GB)	% Error RMS/Peak
MOR	8	6.5	7.7	0.006/0.06
S interpolation	92	20.6	3.0	0.05/0.7
none	400	91.2	3.1	—/—

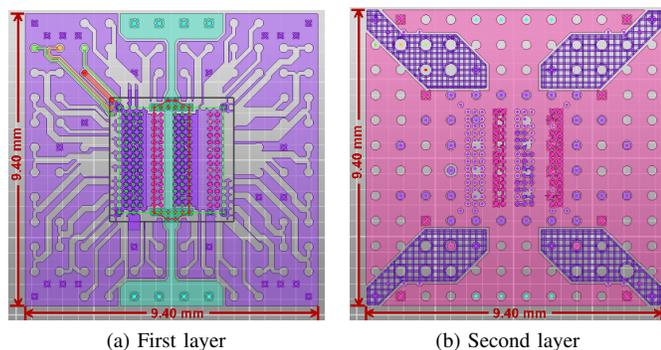


Fig. 3. First two layers of flip-chip package. The four signal nets are in the upper left-hand corner of the first layer. Note the cross-hatched grounds in the corners of the second layer, a source of additional reflections.

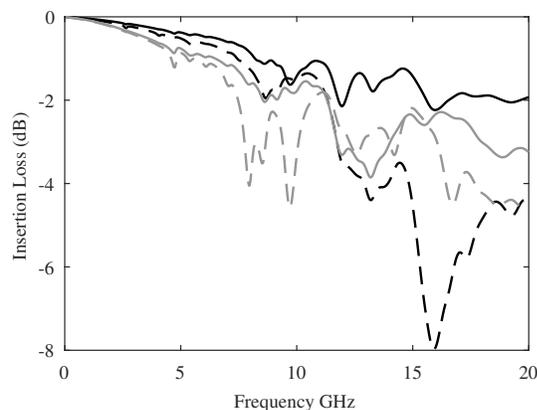


Fig. 4. Package insertion loss for each signal net.

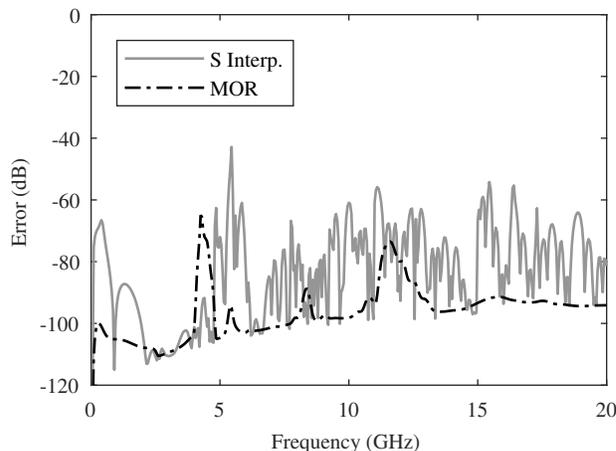


Fig. 5. Error in S matrix for flip-chip package, using S matrix interpolation versus high-order moment matching MOR.

## REFERENCES

- [1] R. D. Slone, R. Lee, and J.-F. Lee, "Multipoint Galerkin asymptotic waveform evaluation for model order reduction of frequency domain FEM electromagnetic radiation problems," *IEEE Trans. Antennas Propag.*, vol. 49, no. 10, pp. 1504–1513, Oct. 2001.
- [2] A. Odabasioglu, M. Celik, and L. T. Pileggi, "PRIMA: Passive reduced-order interconnect macromodeling algorithm," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 17, no. 8, pp. 645–654, Aug. 1998.
- [3] *HFSS Online Help (Release 19.0)*, ANSYS, Inc., Canonsburg, PA, "Finite conductivity boundaries".
- [4] O. Farle and R. Dyczij-Edlinger, "Numerically stable moment matching for linear systems parameterized by polynomials in multiple variables with applications to finite element models of microwave structures," *IEEE Trans. Antennas Propag.*, vol. 58, no. 11, pp. 3675–3684, Nov. 2010.
- [5] M. Bartholomew-Biggs, S. Brown, B. Christianson, and L. Dixon, "Automatic differentiation of algorithms," *J. Comput. and Appl. Math.*, vol. 124, no. 1, pp. 171–190, 2000.
- [6] Selected AD tools. [Online]. Available: <http://www.autodiff.org/?module=Tools>
- [7] *Intel Math Kernel Library for Windows: Developer Guide (MKL 2018)*, Intel, "Obtaining Numerically Reproducible Results," [Online]. Available: <https://software.intel.com/sites/default/files/managed/35/53/mkl-2018-developer-guide-win.pdf>