

Stable and Compact Inductance Modeling of 3-D Interconnect Structures

Hong Li, Venkataramanan Balakrishnan, and Cheng-Kok Koh
School of Electrical and Computer Engineering
Purdue University, West Lafayette, IN 47907-1285
{li73, ragu, chengkok}@ecn.purdue.edu

ABSTRACT

Recent successful techniques for the efficient simulation of large-scale interconnect models rely on the sparsification of the inverse of the inductance matrix L . While there are several techniques for sparsifying L^{-1} , the stability of these approximations for general interconnect structures has not been established, i.e., the sparsified reluctance and inductance matrices are not guaranteed to be positive-definite. In this paper, we present a novel technique for reluctance sparsification for general interconnect structures that enjoys several advantages: First, the resulting sparse approximation is *guaranteed to be positive definite*. Second, the approximation is *optimal*, in a certain well-defined sense. Third, owing to its computational efficiency and numerical stability, the algorithm is applicable for very large problem sizes. Finally our approach yields a *compact* representation of both inductance and reluctance matrices for general cases.

1. INTRODUCTION

With the aggressive scaling of VLSI technology, the problem of accurate modeling of interconnects has become increasingly important. The Partial Element Equivalent Circuit (PEEC) model has been widely used to analyze on-chip interconnects [10]. However the inductance matrix L obtained from PEEC is large and dense, direct simulation with which typically places unrealistic demands on both simulation time and memory.

One approach towards addressing this issue is to reduce the number of inductance parameters by sparsifying either L or L^{-1} . (We will use \hat{L} and \hat{L}^{-1} generically to denote the resulting approximations.) Devgan *et al.* [4], noting that the off-diagonal entries of L^{-1} diminish much faster than those in L , proposed the sparsification of L^{-1} by retaining only significant terms; the resulting \hat{L}^{-1} can reduce the number of coupling terms and speed up simulation. However, L^{-1} needs to be computed first, which is prohibitively expensive. Several window-based techniques have been used to approximate \hat{L}^{-1} [5, 1, 2]. With such techniques, a small inductance submatrix of wires strongly coupled to the wire of interest is inverted, and the row (or column) of the inverse corresponding to the wire of interest is used to construct significant entries in \hat{L}^{-1} .

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICCAD '06, November 5-9, 2006, San Jose, CA
Copyright 2006 ACM 1-59593-389-1/06/0011... ..\$5.00.

Beyond efficiency and accuracy, an important consideration is that of *stability*: It is desirable to have a guarantee that the approximate inductance \hat{L} is positive-definite. While above techniques provide faster simulation, stability of the approximation is not guaranteed with all of the above methods. Although stability of the approximate \hat{L} can be proven in the special case when L is diagonally dominant, it is easy to construct examples where this is not the case; see for example [6].

One approach for obtaining stable approximants \hat{L} for the one-dimensional case, i.e., parallel unsegmented wires in a single layer, was presented in [7]. However, this approach suffers from two main drawbacks: First, for large L matrices, the technique proposed in [7] to obtain \hat{L} exhibits numerical instability. Second, the technique in [7] is not applicable to the more general and practical case consisting of segmented parallel wires over multiple layers. We will henceforth refer to this general case as the three-dimensional (3-D) case: the segmentation in a single layer accounts for x - and y -dimensions, and the multiple layers account for the z -dimension.

In this paper, we present a novel *multi-band* matching method for the approximation of the inductance matrix in general interconnect structures: The *multi-band* entries of the approximate inductance matrix \tilde{L} match those of L , while \tilde{L}^{-1} is a multi-band matrix. Our proposed technique enjoys several advantages: (i) \tilde{L}^{-1} is sparse. (ii) $\tilde{L} > 0$. (iii) \tilde{L} is optimal, in a certain well-defined sense. (iv) \tilde{L} preserves the values of all significant couplings. (v) The technique provides a compact representation for \tilde{L} and is computationally efficient and numerically stable.

2. REVIEW OF EXISTING SPARSIFICATION METHODS

We briefly review existing methods for the sparsification of L^{-1} . Perhaps the simplest technique is that of direct truncation [5], where L is inverted, its small off-diagonal terms truncated to obtain \bar{L}^{-1} , which is then inverted to get \bar{L} . \bar{L} serves as an approximation to L in the sense that the in-band entries of \bar{L}^{-1} and L^{-1} are matched. However, the band entries of L and \bar{L} can be significantly different, resulting in a loss of simulation accuracy. Moreover, this technique requires two large matrix inversions, which can be expensive.

Window-based techniques provide an alternative that obviates the need for expensive large matrix inversion [5, 1, 2]. The work in [12] provides a good mathematical foundation for windowing techniques: If A is a $n \times n$ banded matrix with bandwidth $2b + 1$ and $B = A^{-1}$, the significant entries of A can be computed by using only a subset of the matrix B . We take the intersection of rows $i - b$ to $i + b$ and columns $i - b$ to $i + b$ of B to form a sub-matrix. Then, the center row and center column of the inverse of the sub-matrix are identical to the corresponding entries in the i th row and

i th column of the A matrix:

$$A(i, i-b:i+b) = (B(i-b:i+b, i-b:i+b))^{-1}(b+1, :), \quad (1)$$

$$A(i-b:i+b, i) = (B(i-b:i+b, i-b:i+b))^{-1}(:, b+1). \quad (2)$$

Here, $A(i:j, m:n)$ refers to the sub-matrix at the intersection of rows i to j and columns m to n of A , $A(:, m)$ refers to column m and $A(i, :)$ refers to row i .

In the window-based methods proposed in [5, 1, 2], the original L matrix, instead of \tilde{L} , is used to calculate \tilde{L}^{-1} . However, as L^{-1} is not exactly banded, rows calculated by (1) are different from columns calculated by formula (2), resulting in an asymmetric approximation of L . Moreover, stability is not guaranteed as \tilde{L} and \tilde{L} are not necessarily positive definite.

A new technique for the approximate sparsification of L^{-1} was presented for the one-dimensional case in [7], where the inverse of the inductance matrix is approximated by a banded matrix. The in-band entries of \hat{L} , constructed by this method, match those entries in L , with \hat{L}^{-1} being a banded matrix. The algorithm in [7] uses the following elegant structure of inverses of banded matrices to compute \hat{L} efficiently. A is a banded matrix with bandwidth of $2b+1$ if and only if $B = [b_{i,j}] = A^{-1}$ is given by two sequences of vectors $\{u\}_{i=1}^n, \{v\}_{i=1}^n, u_i, v_i \in R^b$ such that

$$B = \begin{bmatrix} u_1^T & u_1^T & \cdots & u_1^T \\ u_1^T & u_2^T & \cdots & u_2^T \\ \vdots & \vdots & \ddots & \vdots \\ u_1^T & u_2^T & \cdots & u_n^T \end{bmatrix} \circ \begin{bmatrix} v_1 & v_2 & \cdots & v_n \\ v_2 & v_2 & \cdots & v_n \\ \vdots & \vdots & \ddots & \vdots \\ v_n & v_n & \cdots & v_n \end{bmatrix}.$$

However, the inverse of the inductance matrix for 3-D interconnect structures is no longer a banded matrix. Hence, the algorithm in [7] applies only to the 1-D case. Moreover, the algorithm exhibits poor numerical properties when large-sized inductance matrices are considered. The contribution of this paper is to present an algorithm that addresses both deficiencies: We present an efficient, numerically stable multi-band matching method that applies to 3-D interconnect structures, resulting in an approximate inductance matrix that is guaranteed to be positive definite. Moreover, the approximation is optimal in a sense that we will define in the sequel.

3. BAND MATCHING METHOD: 2-D AND 3-D CASES

To illustrate the approximate multi-band nature of the reluctance matrix in 3-D interconnect structures, consider an interconnect structure consisting of three layers with five wires in each layer, where each wire is divided into four segments, as shown in Fig. 1(a). (Our proposed technique can easily handle wires that are of arbitrary lengths, widths, and heights.) The wire segments are indexed first in the y direction, then in the x direction, followed by the z direction, as shown in Fig. 1(a). Truncating the small entries in the inverse of the inductance matrix (obtained with above mentioned ordering) [4] produces a sparse matrix with the pattern shown in Fig. 1(b). This sparse matrix can be partitioned into 3×3 blocks with each block size being 20×20 (20 segments in one layer). Diagonal blocks characterize the interactions between wires in the same layer and off-diagonal blocks represent the interactions between different layers. Each 20×20 block again can be partitioned into 5×5 blocks (5 wires in one layer) with each block being 4×4 (4 segments for one wire). Here, the diagonal blocks represent the couplings between different segments in the same wire and the off-diagonal blocks represent the interactions between different wires

in the same layer. Division of wires into segments that are properly aligned has been used in [8, 2].

The inverse of the inductance matrix can be understood in terms of current induced by the magnetic vector potential drop along a set of conductor segments [5]. Since the magnetic field generated by each neighbor cancels part of the field induced on the aggressor line, and shields the field induced on the aggressor line from influencing farther wires, the (i, j) entries in L^{-1} are small when wires i and j are not neighbors. This is the so-called shielding effect between neighbors. All the sparsification techniques assume perfect shielding by neighboring wires so that when wire i and j are not neighbors, the (i, j) entry in the inverse of inductance matrix is assumed to be zero. In general, the bands of significant entries can be wider if more nearby wires are required to capture the shielding effect more accurately.

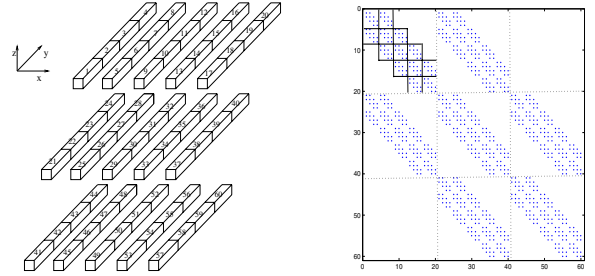


Figure 1: (a) 3-D interconnects. (b) Sparsity pattern for the inverse of the inductance matrix when shielding is perfect.

For convenience, we term the sparse matrices such as the one shown in Fig. 1(b) as *multi-band matrices*, and their nonzero entries as *multi-band entries*. In practice however, shielding is never perfect. Thus arises the problem of approximating the inductance matrix L with \tilde{L} whose inverse is a *multi-band matrix*, i.e. when every wire is assumed to be perfectly shielded by its neighboring wires, an assumption central to most sparsification techniques. Moreover, in our proposed method, we require that the mutual inductance values between neighboring wires remain the same. In other words, we require the multi-band entries of \tilde{L} to match those of L .

The solution that we describe relies on the elegant structure of the inverse of block-tridiagonal matrices which provides a compact parametrization for matrices with block-tridiagonal inverses.

LEMMA 3.1 ([9]). *Suppose that the inverse of a symmetric $nm \times nm$ matrix A is block-tridiagonal, with $m \times m$ blocks. Then, there exist two sequences of $m \times m$ matrices U_i and V_i such that for $j \geq i$, $[A]_{ij} = U_i V_j$. Thus, A can be written as*

$$A = \begin{pmatrix} U_1 V_1 & U_1 V_2 & \cdots & U_1 V_n \\ V_2^T U_1^T & U_2 V_2 & \cdots & U_2 V_n \\ \vdots & \vdots & \ddots & \vdots \\ V_n^T U_1^T & V_n^T U_2^T & \cdots & U_n V_n \end{pmatrix}. \quad (3)$$

The sequences U_i and V_i can be computed from the tridiagonal blocks of A as follows. With A_{ij} denoting the (i, j) $m \times m$ block-matrix of A ,

$$U_1 = I, V_1 = A_{11}, \\ V_i = U_{i-1}^{-1} A_{i-1, i}, U_i = A_{ii} V_i^{-1}, \text{ for } i = 1, \dots, n. \quad (4)$$

Using the sequences $\{U_i\}$ and $\{V_i\}$, the matrix A can be compactly described by $(2n-1)m^2$ independent parameters (U_1 can be chosen as any positive definite matrix), matching the number of

independent parameters in A^{-1} . If A^{-1} is block-banded, the formula (4) still applies, but U_i and V_i become rectangular matrices. However treating block-banded matrices as block-tridiagonal matrices with a larger size for each block results in more efficient computation. Therefore, we will henceforth focus on block-tridiagonal matrices.

Lemma 3.1 can be readily used to solve the band matching problem for block tridiagonal case. Given a block matrix L , the following algorithm generates a matrix \tilde{L} which has the same block tridiagonal entries as L , with \tilde{L}^{-1} being block tridiagonal matrix.

BLOCK-TRIDIAGONAL BAND MATCHING ALGORITHM

Algorithm $\tilde{L} = \text{BTBM}(L)$;
 1. Via (4), generate the matrix sequences U_i and V_i , using only the block-tridiagonal entries of L .
 2. Set the block-tridiagonal entries of \tilde{L} equal to those of L .
 3. Generate the remaining entries of \tilde{L} from U_i and V_i using (3).

Note that if only a *representation* of \tilde{L} rather than all of its entries is sought, only Step 1 of the algorithm is needed.

3.1 An Iterative Approach

The band-matching problem for 3-D interconnect structures requires multi-band matching, not simple block tridiagonal matching. To illustrate this, consider a simple 2-D problem of 3 layers with 5 wires in each layer without segmentation. The cross-sectional view and the numbering of the wires are shown in Fig. 2(a). Truncating the small entries in the inverse of the inductance matrix results in a sparse matrix with the sparsity pattern shown in Fig. 2(b). This implies that the shieldings from two directions are perfect at the same time. Hence, the problem now is to find an approximation to L whose inverse has the desired sparsity pattern as Fig. 2(b).

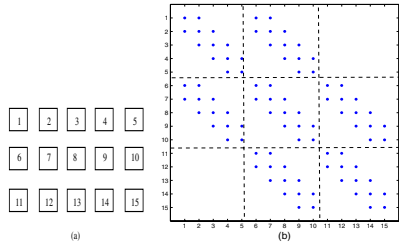


Figure 2: (a) Cross-sectional view of a 2-D example. (b) The desired multi-band structure for \tilde{L}^{-1} .

Let Φ denote the set of sparse matrices with the desired multi-band structure, as shown in Fig. 2(b). Let T_Φ denote the projection of a matrix onto Φ , i.e., for a given matrix M , $T_\Phi M \in \Phi$ with the multi-band entries of $T_\Phi M$ and M being the same. We can also view $T_\Phi M$ as a matrix that keeps only the multi-band entries of M . Then the band matching problem is: Find \tilde{L} such that

$$1. T_\Phi L = T_\Phi \tilde{L}, \quad 2. \tilde{L}^{-1} \in \Phi \quad (\text{P}).$$

Our approach towards solving (P) is via alternating between two standard block-tridiagonal band matching problems. Each of these problems can be readily solved using Algorithm BTBM.

For the sake of clarity, we use the 2-D example in Fig.2 to explain the basic idea behind our algorithm for multi-band matching. The same idea can be readily extended to the 3-D problem.

We begin by defining two auxiliary problems. The first problem, (P1), is a block-tridiagonal band matching problem where perfect shielding is assumed between horizontal layers, and the in-layer

shielding is ignored. This results in a block tridiagonal band matching problem: Find \tilde{L} such that

$$1. T_{\Phi_1} L = T_{\Phi_1} \tilde{L}, \quad 2. \tilde{L}^{-1} \in \Phi_1, \quad (\text{P1})$$

where Φ_1 consists of block tridiagonal matrices, as shown in Fig. 4(a). Three horizontal layers for this example are shown by the dotted circle in Fig. 3(a).

To define the second auxiliary problem, (P2), we define ‘‘vertical’’ layers by grouping wires as $(1, 6, 11), (2, 7, 12), \dots, (5, 10, 15)$, as shown in Fig. 3(b). Assume that the vertical layers are perfectly shielded and ignoring the shielding between wires in each vertical layer, we obtain another block-tridiagonal band-matching problem after an appropriate permutation of indices: Find \tilde{L} such that

$$1. T_{\Phi_2}(P^T \tilde{L} P) = T_{\Phi_2}(P^T L P), \quad 2. (P^T \tilde{L} P)^{-1} \in \Phi_2, \quad (\text{P2})$$

where Φ_2 consists of block tridiagonal matrices, as shown in Fig. 4(b). Note that we have reordered the indices of the matrix via a permutation P .

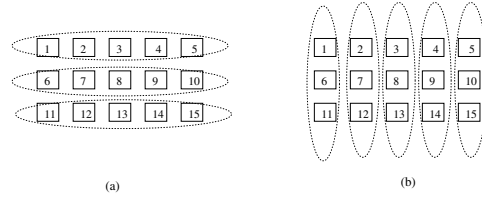


Figure 3: (a) Horizontal layers. (b) Vertical layers.

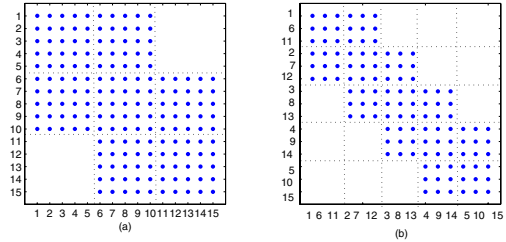


Figure 4: Illustration of sets Φ_1 and $P\Phi_2P^T$ for the 2-D example. Their intersection matches the structure in Fig. 2(b).

We now make the following key observation: $\Phi = \Phi_1 \cap (P\Phi_2P^T)$. (Fig. 5 illustrates this for the example under consideration.) This motivates the following iterative multi-band matching algorithm to solve (P) by alternating between solving (P1) and (P2):

MULTI-BAND MATCHING ALGORITHM

Algorithm $\tilde{L} = \text{MBM}(L)$;
 Set $L_2 = P^T L P$;
 repeat {
 solve $L_1 = \text{BTBM}(P L_2 P^T)$;
 solve $L_2 = \text{BTBM}(P^T L_1 P)$;
 } until $L_1 - P L_2 P^T$ is small
 return $\tilde{L} = L_1$;

If L^{-1} is close to a multi-band matrix to begin with, it typically takes only a few iterations for the algorithm to converge (see Section 4 for numerical results). Moreover, note that the algorithm alternates between solving two block-tridiagonal band matching problems (P1) and (P2). The *intersection* of these matched entries, i.e., the multi-band entries, are unchanged with iterations, so that at every iteration, L_1 and $P L_2 P^T$ match L in multi-band positions,

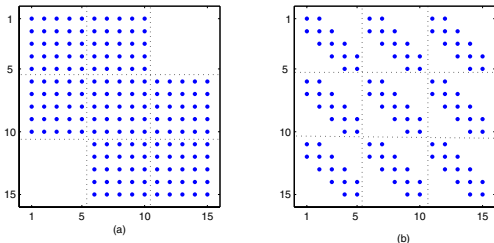


Figure 5: (a) Sparsity pattern of matrices in Φ_1 (b) Sparsity pattern of matrices in $P\Phi_2P^T$

with L_1^{-1} and L_2^{-1} belonging to Φ_1 and Φ_2 respectively. At convergence, $L_1^{-1} = P(L_2)^{-1}P^T \in \Phi_1 \cap (P\Phi_2P^T)$.

Each iteration of the MBM algorithm requires the execution of only Step 1 of the BTBM algorithm. For example, suppose that the U_i and V_i sequences that compactly represent the BTBM solution to $L_1 = \text{BTBM}(P^T L_2 P)$ have been generated. For the next iteration, $L_2 = \text{BTBM}(PL_1P^T)$, only the block-tridiagonal entries of PL_1P^T are needed; these and only these can be generated efficiently using the matrices U_i and V_i that represent L_1 . This compact representation can also be useful in reducing computation when using simulation tools (see for example [12]) that use only a subset of the entries of the inductance matrix.

The algorithm enjoys the following properties:

- At any iteration $\tilde{L} > 0$ (see Appendix A for proof).
- The algorithm is guaranteed to converge (see Appendix B for proof).
- The approximate \tilde{L} given by the algorithm is optimal in the following sense: Among all \tilde{L} such that $\tilde{L}^{-1} \in \Phi$, the algorithm yields the one that minimizes

$$d(L, \tilde{L}) = \frac{1}{N} \{ \text{tr}(\tilde{L}L^{-1}) - \log \det(\tilde{L}L^{-1}) \} - 1.$$

(This is the so called Kullback-Leibler distance in the information theory; see Appendix B for proof.) Moreover, it can be seen from the simulation results that the error obtained from band matching approximation is smaller compared to those obtained from other sparsification techniques.

4. EXPERIMENTAL RESULTS

We demonstrate the band matching method by using two circuits: structured wires in 3-D space and randomly generated wires in 3-D space.

4.1 Experiment 1

First, we demonstrate the band matching method on a three-layer bus structure with 32 signals on each layer. The wire length is 1mm , the cross section $1 \times 1\mu\text{m}$, the wire separation $1\mu\text{m}$, and the separation between layers $3\mu\text{m}$. Wires are divided into 5 segments along the length. The driver resistance is 30Ω and the load capacitance is 50fF . A 1V 20ps ramp is applied to the first signal of the middle layer, and the rest are quiet. Shields are inserted in several wires randomly. We compare the simulation results obtained from our band matching (BM) method with other models obtained from a direct truncation (DT) method and the Wire Duplication (WD) method. (WD is a window-based method. As other window-based methods are similar, we have chosen the Wire Duplication method to report our results.) All experiments were run on a Pentium III 1GHz processor.

The waveforms for the first and second signal of the middle layer are shown in Figs. 6 and Figs. 7 respectively. For a fair comparison, the width of the coupling window in horizontal direction is set to be 5 for every method, and vertical couplings are also considered.

Table 1: Comparison of AER and PER.

wire	AER			PER		
	WD	DT	BM	WD	DT	BM
1	0.251	0.400	0.183	0.017	0.011	0.007
2	0.443	0.251	0.183	0.235	0.174	0.110
3	0.426	0.260	0.178	0.245	0.011	0.007
4	0.412	0.282	0.175	0.264	0.213	0.125
33	0.013	0.008	0.005	0.018	0.012	0.008
34	0.353	0.229	0.149	0.019	0.013	0.008
All	0.486	0.409	0.158	0.272	0.193	0.112

We denote the voltage in i th wire obtained by the full L matrix by V_i (i.e., the ‘‘true’’ voltage), and by \hat{V}_i the i th voltage obtained by the approximation methods. We use the $|\cdot|_1$ norm to measure the errors (the results are essentially unchanged when other norms such as $|\cdot|_2$ are used). Define average error ratio (AER) and peak error ratio (PER) for the i th wire as

$$\text{AER} = \frac{\sum_t |\hat{V}_i - V_i|}{\sum_t |V_i|}, \quad \text{PER} = \frac{\max(|\hat{V}_i - V_i|)}{\max(|V_i|)},$$

average error ratio (AER) and peak error ratio (PER) for all wires are defined as

$$\text{AER for all wires} = \frac{\sum_i \sum_t |\hat{V}_i - V_i|}{\sum_i \sum_t |V_i|},$$

$$\text{PER for all wires} = \max_i (\text{Peak error in } i\text{th wire}).$$

Table 2: Memory and run time usage for different methods.

Method	Memory (MB)	Run Time(s)
full L matrix	206.4	8.7×10^5
direct truncation	65.2	832
wire duplication	49.6	657
multi-band matching	49.8	661

Note that these definitions typically result in a PER that is smaller than the corresponding AER. As can be seen from Table 1, the BM method gives the smallest average error ratio and peak error ratio. Table 2 provides the runtime and memory usage for different methods. Fig. 8 offers data on the run time and the number of iterations for the band matching method for inductance matrices corresponding to n layers with n wires in each layer, for various n .

4.2 Experiment 2

In Experiment 1, the regularity of the layout plays a role in the inverse of the inductance matrix being close to multi-band. In this experiment, we randomly generated 100 wires in 3-D space: The length, width, and height for every wire, as well as their placement are random, with no overlap allowed between any two wires. The response to a 1V 20ps ramp, applied to a randomly chosen signal, as well as the waveforms of its two neighboring signals in the horizontal and vertical directions are shown in Fig. 9(a), Fig. 9(c), Fig. 9(e). As shown in the figures, DT method yields significant simulation errors, as a consequence of L^{-1} no longer being very close to a multi-band matrix. However, both WD and BM methods yields much better simulation results, as they both preserve the multi-band entries in L matrix. Fig. 9(b), Fig. 9(d), Fig. 9(f) present an enlarged picture for a closer comparison of WD and BM. As mentioned earlier, the WD method produces an asymmetric approximation and without a stability guarantee. Both these issues are resolved with our proposed BM method.

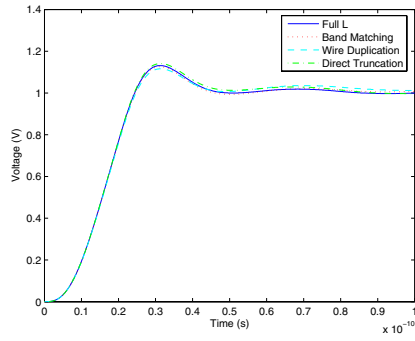


Figure 6: Simulation results for signal 1 on the middle layer.

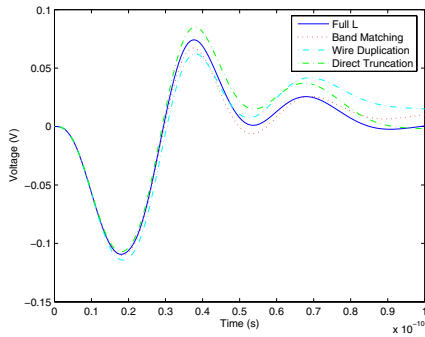


Figure 7: Simulation results for signal 2 on the middle layer.

5. REFERENCES

- [1] M. Beattie and L. Pileggi. Modeling magnetic coupling for on-chip interconnect. In *Proc. Design Automation Conf.*, pages 335–340, 2001.
- [2] T. H. Chen, H. Kim C. Luk, and C. C.-P. Chen. INDUCTWISE: Inductance-wise interconnect simulator and extractor. In *Proc. Int. Conf. on Computer Aided Design*, pages 215–220, 2002.
- [3] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications. John Wiley and Sons, Inc., 1991.
- [4] A. Devgan, H. Ji, and W. Dai. How to efficiently capture on-chip inductance effects: Introducing a new circuit element k. In *Proc. Int. Conf. on Computer Aided Design*, pages 150–155, 2000.

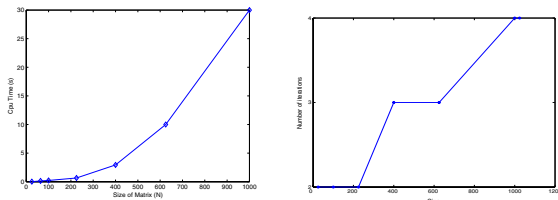


Figure 8: (a) Run times. (b) Number of iterations required.

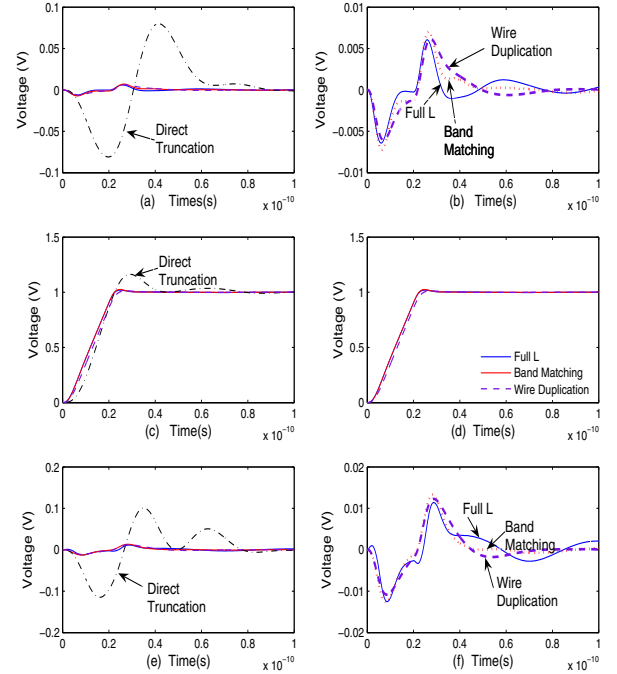


Figure 9: Simulation results for randomly generated wires

- [5] H. Ji, A. Devgan, and W. Dai. KSim: a stable and efficient RKC simulator for capturing on-chip inductance effect. In *Proc. Asia South Pacific Design Automation Conf.*, pages 379–384, 2001.
- [6] H. Li, V. Balakrishnan, and C.-K. Koh. <http://web.ics.purdue.edu/~li73/report1.pdf>.
- [7] H. Li, V. Balakrishnan, C.-K. Koh, and G. Zhong. Compact and stable modeling of partial inductance and reluctance matrices. In *Proc. Asia South Pacific Design Automation Conf.*, pages 507–510, 2005.
- [8] T. Lin, Michael W. Beattie, and Lawrence T. Pileggi. On the efficacy of simplified 2D on-chip inductance models. In *Proc. Design Automation Conf.*, pages 757–762, 2003.
- [9] P. Rozsa. On the inverse of band matrices. *Integral Equations and Operator Theory*, 10:82–95, 1987.
- [10] A. E. Ruehli. Equivalent circuit models for three dimensional multiconductor systems. In *IEEE Trans. on Microwave Theory and Techniques*, pages 216–221, 1974.
- [11] G. Strang. *Linear Algebra and its Applications*. Thomson Learning, 1986.
- [12] G. Zhong, C.-K. Koh, and K. Roy. On-chip interconnect modeling by wire duplication. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 22:1521–1532, 2003.

Appendix A: Proof of stability

LEMMA 5.1. [6] Let the matrix A^{-1} be a $q \times q$ block tridiagonal matrix with each diagonal block $p_i \times p_i$, $i = 1, \dots, q$. Then the determinant of A is

$$\det A = \frac{D(1 : p_1 + p_2) \cdots D(\sum_1^{q-2} p_i + 1 : \sum_1^q p_i)}{D(p_1 + 1 : p_2) \cdots D(\sum_1^{q-2} p_i + 1 : \sum_1^{q-1} p_i)}, \quad (5)$$

where $D(i : j)$ is the determinant of $A(i : j, i : j)$.

An important consequence of Lemma 5.1 is that the determinant of a matrix with an inverse that is block tridiagonal depends only on the entries in the tridiagonal blocks.

THEOREM 5.1. *If L is positive definite, its band matching approximation \tilde{L} obtained by the proposed iterative algorithm is also positive definite.*

Proof. We prove that the matrix we obtain in each iteration, denoted as \tilde{L}^i , is positive definite. We use the notation that for any $n \times n$ matrix A , A_m is the $m \times m$ principal sub-matrix of A (i.e., the $m \times m$ submatrix that forms the top left-hand corner of A).

A necessary and sufficient condition for \tilde{L}^i to be positive definite is that $\det \tilde{L}_m^i > 0$ for $m = 1, \dots, n$; see for example [11]. As L is positive definite, $L_m > 0$, $m = 1, \dots, n$.

Let x denote the size of diagonal blocks. In the first iteration, \tilde{L}^1 is a BTBM solution for matrix L . Hence all the tridiagonal blocks in L and \tilde{L}^1 are the same. When $m \leq 2x$, as $\tilde{L}^1(1 : 2x, 1 : 2x) = L(1 : 2x, 1 : 2x)$, we obtain,

$$\det \tilde{L}_m^1 = \det L_m > 0.$$

For the case $m > 2x$, all the determinants of \tilde{L}_m^1 , $m = 1, \dots, n$ can be calculated from Lemma 5.1 as they can be described by U_i and V_i , and their inverses are block tridiagonal matrices. Note that determinant of \tilde{L}_m^1 depends only on the block tridiagonal entries of \tilde{L}_m^1 from Lemma 5.1. As all the block tridiagonal entries of \tilde{L}^1 match the block tridiagonal entries of L , all the $D(i : j)$ in (5) are positive (as L is positive definite). Therefore $\det \tilde{L}_m^1 > 0$, for the case $m > 2x$.

Finally, permutation matrices are orthogonal, hence, $P^{-1} = P^T$, and $\det(P^T \tilde{L}^1 P) = \det \tilde{L}^1$. With this observation and the same line of reasoning as above, it follows that the matrix \tilde{L}^2 obtained after the second iteration is positive definite as well. Extending this argument further leads to the conclusion that the matrices obtained at the end of every iteration is positive definite, concluding the proof. \square

Appendix B: Proof of Optimality and convergence

In Information Theory, the Kullback-Leibler (KL) distance is used to measure the distance between probability distributions [3]. The KL distance can also be used to define the distance between two positive definite matrices: The KL distance from L_1 to L_2 is

$$d(L_1, L_2) = \frac{1}{N} \{ \text{Tr}(L\tilde{L}^{-1}) - \log \det(L\tilde{L}^{-1}) \} - 1.$$

$d(L_1, L_2) \geq 0$, and vanishes if and only if $L_1 = L_2$.

We now state the result on the optimality of the multi-band matching approximant.

THEOREM 5.2. *Given a matrix L , the multi-band matching approximation \tilde{L} is at the smallest KL distance from L .*

Proof. Suppose L_a is any other matrix with the inverse having the desired multi-band structure. We note that $\text{Tr}(L\tilde{L}^{-1}) = N$, since L and \tilde{L} have the same multi-band entries and \tilde{L}^{-1} is a multi-band matrix. Then,

$$\begin{aligned} N(d(L, \tilde{L}) + d(\tilde{L}, L_a)) &= \text{Tr}(L\tilde{L}^{-1}) - \log |L\tilde{L}^{-1}| - N \\ &\quad + \text{Tr}(\tilde{L}L_a^{-1}) - \log |\tilde{L}L_a^{-1}| - N \\ &= \text{Tr}(\tilde{L}L_a^{-1}) - \log |L| - \log |L_a^{-1}| - N \\ &= \text{Tr}(LL_a^{-1}) - \log |LL_a^{-1}| - N \\ &= N(d(L, L_a)). \end{aligned} \tag{6}$$

Thus, $d(L, \tilde{L}) + d(\tilde{L}, L_a) = d(L, L_a)$. As $d(\tilde{L}, L_a) \geq 0$, $d(L, L_a) \geq d(L, \tilde{L})$, or \tilde{L} has the minimum KL distance from L . \square

Note that Theorem 5.2 applies to the simpler block-tridiagonal matching scenario: Let L_{Φ_1} denote any matrix whose inverse belongs to Φ_1 . Let L_1 denote the matrix which matches L along the block tridiagonal. Then

$$d(L, L_1) + d(L_1, L_{\Phi_1}) = d(L, L_{\Phi_1}). \tag{7}$$

In other words, the BTBM algorithm can be thought of as ‘‘projecting’’ L to Φ_1 in the KL distance sense.

We next turn to the problem of convergence of the iterative algorithm in Section 3.

THEOREM 5.3. *Algorithm MBM converges.*

Proof. For simplify we sketch the proof only for the 2-D case. Suppose L is the original inductance matrix. Denote by $\overline{\Phi}_i$, $i = 1, 2$ the set of matrices whose inverses belong to Φ_i $i = 1, 2$, and by $\overline{P\Phi_2P^T}$ the set of matrices whose inverses belong to $P\Phi_2P^T$.

Let k be the iteration index of the MBM algorithm, and let $L_1^{(k)}$ and $L_2^{(k)}$ respectively denote the solutions to problems (P1) and (P2), during iteration k . Then, $L_1^{(1)} = \text{BTBM}(L)$, and from Theorem 5.2, $L_1^{(1)}$ is the element in $\overline{\Phi}_1$ with the smallest KL distance from L . Next, $L_2^{(1)}$ is the block tridiagonal match of $P^T L_1^{(1)} P$, and therefore is the element in $\overline{\Phi}_2$ with the smallest KL distance from $P^T L_1^{(1)} P$. As permutation matrices are orthogonal, we have $P^{-1} = P^T$, and consequently we have $d(P^T L_1^{(1)} P, L_2^{(1)}) = d(L_1^{(1)}, PL_2^{(1)} P^T)$. Hence, it follows that $PL_2^{(1)} P^T$ is the element in $\overline{P\Phi_2P^T}$ with the smallest KL distance from $L_1^{(1)}$. Subsequent iterations continue along similar lines. Fig. 10 provides an illustration.

Let K be any matrix belonging to $\overline{\Phi}_1 \cap \overline{P\Phi_2P^T}$. As $d(L_1^{(1)}, PL_2^{(1)} P^T)$ is the minimum distance from $L_1^{(1)}$ to $\overline{P\Phi_2P^T}$, it follows from (7) that $d(L_1^{(1)}, K) = d(L_1^{(1)}, PL_2^{(1)} P^T) + d(PL_2^{(1)} P^T, K)$. Therefore, $d(L_1^{(1)}, K) \geq d(PL_2^{(1)} P^T, K)$. Similarly, $d(PL_2^{(1)} P^T, K) \geq d(L_1^{(2)}, K)$. Hence the series

$$\{d(L_1^{(1)}, K), d(PL_2^{(1)} P^T, K), d(L_1^{(2)}, K), d(PL_2^{(2)} P^T, K), \dots\}$$

is monotonically decreasing. Also, it is bounded below by zero, and therefore converges to a unique limit, and consequently the iterations obtained by the algorithm converge to a unique limit $\tilde{L} \in \overline{\Phi}_1 \cap \overline{P\Phi_2P^T}$ with the smallest KL distance from L . As $\tilde{L} \in \overline{\Phi}_1 \cap \overline{P\Phi_2P^T}$, $\tilde{L}^{-1} \in \Phi_1 \cap P\Phi_2P^T$. \square

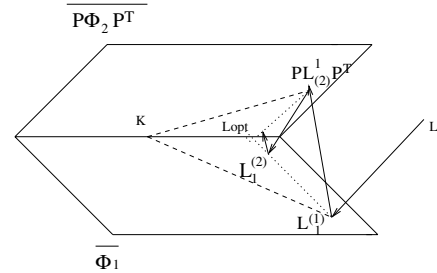


Figure 10: Illustrating the convergence of the iterative algorithm.