

# Partitioned Latency Insertion Method (PLIM) with Stability Considerations

Patrick Goh<sup>1</sup>, José E. Schutt-Ainé<sup>1</sup>, Dmitri Klokov<sup>1</sup>, Jilin Tan<sup>2</sup>, Ping Liu<sup>2</sup>, Wenliang Dai<sup>2</sup>, and Feras Al-Hawari<sup>2</sup>

<sup>1</sup>Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61820, USA.

<sup>2</sup>Cadence Design Systems, Chelmsford, MA 01824, USA.

<sup>1</sup>{pgoh, jesa, dklokot2}@illinois.edu, <sup>2</sup>jtan@cadence.com

## Abstract

In this paper, we present a modular approach to the high frequency simulation of large networks by utilizing the latency insertion method (LIM) and considering the stability criteria of partitions of different latencies in the circuit. This results in a robust algorithm that is able to preserve the stability condition while improving the runtime of the overall transient simulation. An extension to the LIM to handle dependent sources is also presented, along with a generalized stability criteria for selecting a maximum stable time step.

## Introduction

With the increase in density of interconnects and the complexity of high-speed packages, signal integrity becomes an important aspect in the design of modern devices. Circuit designers are constantly in need of newer and better methods that are able to capture the complicated electromagnetic behaviors of complex circuits. In this aspect, the ability to accurately simulate large circuits in a reasonable amount of time is invaluable in the design phase of any circuit. As a result, there is a constant need and push towards faster circuit simulation methods that are able to handle large circuits in a fraction of the time of conventional circuit simulators.

The Latency Insertion Method (LIM) [1] has recently emerged as an efficient approach for performing fast simulations of very large circuits. By exploiting latencies in the circuit, LIM is able to solve the voltages and the currents in the circuit explicitly at each time step. This results in a computationally efficient algorithm that is able to simulate large circuits significantly faster than conventional matrix-inversion based methods.

The two main advantages of using LIM for transient simulations are its linear numerical complexity and its linear memory requirement [2]. For this reason, LIM is most efficiently suited for solving large circuits where the number of elements is in the order of millions. The main disadvantage of LIM on the other hand, is in its reliance on latency elements in the circuit in order to perform its time stepping algorithm. The LIM algorithm is only conditionally stable, with an upper bound on the maximum time step which depends mainly on the smallest inductance and capacitance in the circuit. When the circuit contains very small latency elements, the time step required for a stable simulation could be equally small, which would result in a large number of time steps in a transient simulation. In order to alleviate this problem, a block processing technique has been proposed [3]–[4] which utilizes different time steps for different parts of the circuit. However, selecting the maximum time step for each part of the circuit is still a challenging task and the basic method used in [3]–[4] to select the time step can only be applied under very restrictive assumptions [5]. Specifically, each node in the circuit has to

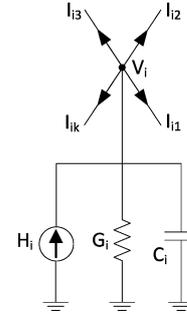


Fig. 1. Node topology.

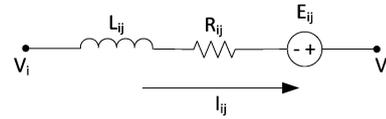


Fig. 2. Branch topology.

be connected to only two branches, and the values of the circuit elements have to be the same everywhere in each sub-circuit.

In this paper, we will extend the basic LIM to handle dependent sources and propose a robust method to select the maximum time step of the LIM simulation, which is independent of the circuit topology. We then apply the method, along with the block processing technique to the simulation of circuits with partitions of different latencies and demonstrate the speed improvements of the proposed method over the basic LIM.

## Basic LIM Formulation

LIM can be applied to any arbitrary network, where it is assumed that through the use of Thévenin and Norton transformations, the branches and nodes of the circuit can be described by the general topologies shown in Fig. 1 and Fig. 2. Each node is represented by a parallel combination of a current source, a conductance, and a capacitor to ground. The connection between two different nodes forms a branch and it is represented by a series combination of a voltage source, a resistor and an inductor. In order to solve for the voltages and currents in the circuit, LIM discretizes the time variable whereby the voltages and currents are collated in half time steps. Specifically, the voltages are solved at half time steps while the currents are solved at full time steps. From Fig. 1, writing Kirchhoff's current law (KCL) at node  $i$  yields

$$C_i \left( \frac{V_i^{n+1/2} - V_i^{n-1/2}}{\Delta t} \right) + G_i V_i^{n+1/2} - H_i^n = - \sum_{k=1}^{M_i} I_{ik}^n \quad (1)$$

where the superscript  $n$  is the index of the current time step,  $\Delta t$  is the time step and  $M_i$  is the number of branches connected to node  $i$ . Solving for the unknown voltage yields

$$V_i^{n+1/2} = \frac{\frac{C_i V_i^{n-1/2}}{\Delta t} + H_i^n - \sum_{k=1}^{M_i} I_{ik}^n}{\frac{C_i}{\Delta t} + G_i} \quad (2)$$

for  $i = 1, 2, \dots, N_n$  where  $N_n$  is the number of nodes in the circuit.

From Fig. 2, writing Kirchhoff's voltage law (KVL) at branch  $ij$  yields

$$V_i^{n+1/2} - V_j^{n+1/2} = L_{ij} \left( \frac{I_{ij}^{n+1} - I_{ij}^n}{\Delta t} \right) + R_{ij} I_{ij}^n - E_{ij}^{n+1/2}. \quad (3)$$

Solving for the unknown current yields

$$I_{ij}^{n+1} = I_{ij}^n + \frac{\Delta t}{L_{ij}} \left( V_i^{n+1/2} - V_j^{n+1/2} - R_{ij} I_{ij}^n + E_{ij}^{n+1/2} \right). \quad (4)$$

The computation of the node voltages and the branch currents are alternated as time progresses in a leapfrog manner. In this aspect, LIM is similar to the Yee's algorithm for the solution of Maxwell's equations in the finite difference time domain (FDTD) method [6]. It is clear that the LIM algorithm relies on the latencies in the network to perform the leapfrog time stepping formulation. Thus, at every node, a capacitor to ground has to be present. If it is not, a small fictitious capacitor is inserted. Similarly, small fictitious inductors are inserted into branches without latencies.

As with the FDTD method, LIM is only conditionally stable. In other words, there is an upper bound on the time step that will result in a numerically stable solution to (2) and (4). This will be analyzed in the next section.

### Vector-matrix Semi-implicit LIM Formulation with Dependent Sources

The LIM method can be readily extended to handle dependent sources. This is most easily done using the vector-matrix version of the semi-implicit LIM [7].

Fig. 3 shows the node topology with a voltage-controlled current source (VCCS) and a current-controlled current source (CCCS) connected to it. Writing the KCL at the node, in semi-implicit form gives

$$C_i \left( \frac{V_i^{n+1/2} - V_i^{n-1/2}}{\Delta t} \right) + G_i \left( \frac{V_i^{n+1/2} + V_i^{n-1/2}}{2} \right) - H_i^n - B_{ik} \left( \frac{V_k^{n+1/2} + V_k^{n-1/2}}{2} \right) - S_{ip} I_p^n = - \sum_{k=1}^{M_i} I_{ik}^n \quad (5)$$

where  $B_{ik}$  is the coefficient of the VCCS at node  $i$  due to node

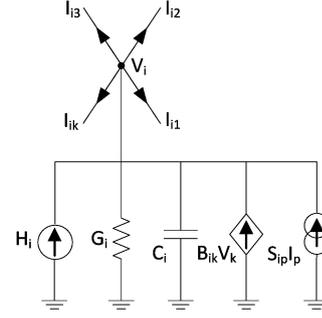


Fig. 3. Node topology with dependent sources.

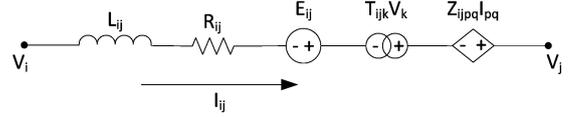


Fig. 4. Branch topology with dependent sources.

$k$  and  $S_{ip}$  is the coefficient of the CCCS at node  $i$  due to branch  $p$ . Equation (5) can then be written in vector-matrix form as

$$\mathbf{C} \left( \frac{\mathbf{v}^{n+1/2} - \mathbf{v}^{n-1/2}}{\Delta t} \right) + \frac{1}{2} \mathbf{G} (\mathbf{v}^{n+1/2} + \mathbf{v}^{n-1/2}) - \mathbf{h}^n - \frac{1}{2} \mathbf{B} (\mathbf{v}^{n+1/2} + \mathbf{v}^{n-1/2}) - \mathbf{S} \mathbf{i}^n = -\mathbf{M} \mathbf{i}^n \quad (6)$$

where  $\mathbf{v}$  is the node voltage vector of dimension  $N_n$ ,  $\mathbf{i}$  is the branch current vector of dimension  $N_b$ ,  $\mathbf{C}$  and  $\mathbf{G}$  are the capacitance and conductance matrices respectively of dimensions  $N_n$  by  $N_n$ ,  $\mathbf{h}$  is a vector of dimension  $N_n$  containing all the current sources at the nodes,  $\mathbf{B}$  and  $\mathbf{S}$  are the matrices of dimensions  $N_n$  by  $N_n$  and  $N_n$  by  $N_b$  containing the coefficients of the VCCS and CCCS respectively, and  $\mathbf{M}$  is the  $N_n$  by  $N_b$  incidence matrix defined as follows

$M_{qp} = 1$  if branch  $p$  is incident at node  $q$  and the current flows away from node  $q$ .

$M_{qp} = -1$  if branch  $p$  is incident at node  $q$  and the current flows into node  $q$ .

$M_{qp} = 0$  if branch  $p$  is not incident at node  $q$ .

Equation (6) can be solved for  $\mathbf{v}^{n+1/2}$  to yield

$$\mathbf{v}^{n+1/2} = \left( \frac{\mathbf{C}}{\Delta t} + \frac{\mathbf{G}'}{2} \right)^{-1} \left[ \left( \frac{\mathbf{C}}{\Delta t} - \frac{\mathbf{G}'}{2} \right) \mathbf{v}^{n-1/2} + \mathbf{h}^n - \mathbf{M} \mathbf{i}^n \right] \quad (7)$$

where

$$\mathbf{G}' = \mathbf{G} - \mathbf{B} \quad \text{and} \quad \mathbf{M}' = \mathbf{M} - \mathbf{S}. \quad (8)$$

Fig. 4 shows the branch topology with a voltage-controlled voltage source (VCVS) and a current-controlled voltage

source (CCVS) connected to it. KVL at the branch, in semi-implicit form gives

$$V_i^{n+1/2} - V_j^{n+1/2} = L_{ij} \left( \frac{I_{ij}^{n+1} - I_{ij}^n}{\Delta t} \right) + R_{ij} \left( \frac{I_{ij}^{n+1} + I_{ij}^n}{2} \right) \quad (9)$$

$$- E_{ij}^{n+1/2} - T_{ijk} V_k^{n+1/2} - Z_{ijpq} \left( \frac{I_{pq}^{n+1} + I_{pq}^n}{2} \right)$$

where  $T_{ijk}$  is the coefficient of the VCVS at branch  $ij$  due to node  $k$  and  $Z_{ijpq}$  is the coefficient of the CCVS at branch  $ij$  due to branch  $pq$ . Proceeding in a similar manner as before, we may solve for  $\mathbf{i}^{n+1}$  to yield

$$\mathbf{i}^{n+1} = \left( \frac{\mathbf{L}}{\Delta t} + \frac{\mathbf{R}'}{2} \right)^{-1} \left[ \left( \frac{\mathbf{L}}{\Delta t} - \frac{\mathbf{R}'}{2} \right) \mathbf{i}^n + \mathbf{e}^{n+1/2} + \mathbf{M}^T \mathbf{v}^{n+1/2} \right] \quad (10)$$

where

$$\mathbf{M}^T = \mathbf{M}^T + \mathbf{T} \quad \text{and} \quad \mathbf{R}' = \mathbf{R} - \mathbf{Z}. \quad (11)$$

In (10) and (11),  $\mathbf{L}$  and  $\mathbf{R}$  are the inductance and resistance matrices respectively of dimensions  $N_b$  by  $N_b$ ,  $\mathbf{e}$  is a vector of dimension  $N_b$  containing all the voltage sources at the branches,  $\mathbf{T}$  and  $\mathbf{Z}$  are the matrices of dimensions  $N_b$  by  $N_n$  and  $N_b$  by  $N_b$  containing the coefficients of the VCVS and CCVS respectively and  $\mathbf{M}^T$  is the transpose of  $\mathbf{M}$  as defined previously. Equations (7) and (10) then give the new update equations for circuits with dependent sources.

The advantage of the vector-matrix formulation lies in its ability to accurately predict if a time step will be stable. If we assume that all sources are zero, we can combine (7) and (10) to obtain

$$\begin{bmatrix} \mathbf{v}^{n+1/2} \\ \mathbf{i}^{n+1} \end{bmatrix} = \mathbf{A}' \begin{bmatrix} \mathbf{v}^{n-1/2} \\ \mathbf{i}^n \end{bmatrix} \quad (12)$$

where  $\mathbf{A}'$  is the amplification matrix given by

$$\mathbf{A}' = \begin{bmatrix} \mathbf{P}_+ \mathbf{P}'_- & -\mathbf{P}_+ \mathbf{M}' \\ \mathbf{Q}_+ \mathbf{M}'^T \mathbf{P}_+ \mathbf{P}'_- & \mathbf{Q}_+ \mathbf{Q}'_- - \mathbf{Q}_+ \mathbf{M}'^T \mathbf{P}_+ \mathbf{M}' \end{bmatrix} \quad (13)$$

where

$$\mathbf{P}_+ \mathbf{P}'_- = \left( \frac{\mathbf{C}}{\Delta t} + \frac{\mathbf{G}'}{2} \right)^{-1} \quad \mathbf{P}'_- = \left( \frac{\mathbf{C}}{\Delta t} - \frac{\mathbf{G}'}{2} \right) \quad (14)$$

$$\mathbf{Q}_+ \mathbf{Q}'_- = \left( \frac{\mathbf{L}}{\Delta t} + \frac{\mathbf{R}'}{2} \right)^{-1} \quad \mathbf{Q}'_- = \left( \frac{\mathbf{L}}{\Delta t} - \frac{\mathbf{R}'}{2} \right). \quad (15)$$

It is clear that the voltages and the currents in the circuit will be amplified by the matrix  $\mathbf{A}'$  at each time step. Thus *all the eigenvalues of the amplification matrix defined in (13)*

*must have magnitude strictly smaller than 1* for the simulation to be stable. This illustrates the usage of the amplification matrix to predict the stability of a time step  $\Delta t$ .

We write the stability criteria compactly as follows.

$$|\lambda_i(\mathbf{A}'(\Delta t))| < 1 \quad i = 1, 2, \dots, l_b. \quad (16)$$

### Simulating Circuits with Partitions of Different Latencies

Since the LIM method is only conditionally stable, there is a maximum time step that can be used in the time domain simulation, beyond which the simulation would be unstable. This maximum time step is in many cases due not to the general properties of the entire circuit under consideration, but instead on a small subnetwork or partition of the circuit [3]–[4]. The partitioned latency insertion method (PLIM) arose as a technique to improve the overall simulation speed of such networks while preserving the stability criteria. In PLIM, the network is first partitioned into smaller subnetworks. Then the maximum stable time step of each partition is determined. Once all the time steps have been determined, the smallest time step is used in LIM to simulate the circuit, but each partition is only updated as needed, depending on its maximum stable time step. This results in a computationally efficient algorithm, with large speed-ups in the simulation time, especially when the partition with the smallest latency is small compared to the rest of the circuit.

However, selecting the maximum stable time step for each partition is still a challenging task. The method used in [3]–[4] can only be applied if each node in the circuit is connected to only two branches, and the values of the circuit elements are the same everywhere in each sub-circuit [5]. The method in [2] on the other hand, can only be applied to RLC circuits. In this paper, we apply (16) on each individual partition, since it does not impose any restriction on the circuit topology.

### Numerical Results

The circuit in Fig. 5 is simulated in the LIM and PLIM environment. The circuit consists of three partitions detailed as follows:

- 1) Partition 1: High latency partition.
- 2) Partition 2: Low latency partition. (In practice, this could be a partition with no latency, whereby small fictitious elements have been inserted to enable LIM.)
- 3) Partition 3: Dependent sources.

The input is a current source with a single trapezoidal pulse of rise and fall times equal to 1 ns and a pulse width of 4 ns. The maximum amplitude is 0.02 A.

The maximum time step of each partition is determined using (16) to be  $\Delta t_1 = 1.0486 \times 10^{-10}$  s,  $\Delta t_2 = 1.07 \times 10^{-12}$  s and  $\Delta t_3 = 6.741 \times 10^{-11}$  s corresponding to partitions one, two and three respectively. Note that we have chosen the time steps to be integer multiples of the smallest time step for ease of implementation. The simulation results at the input (node 1a) and output (node 4c) are shown in Fig. 6. No loss in accuracy is observed when using PLIM compared to the regular LIM.

Next, PLIM is used to simulate a large circuit where the partition with the smallest latency is small compared to the rest

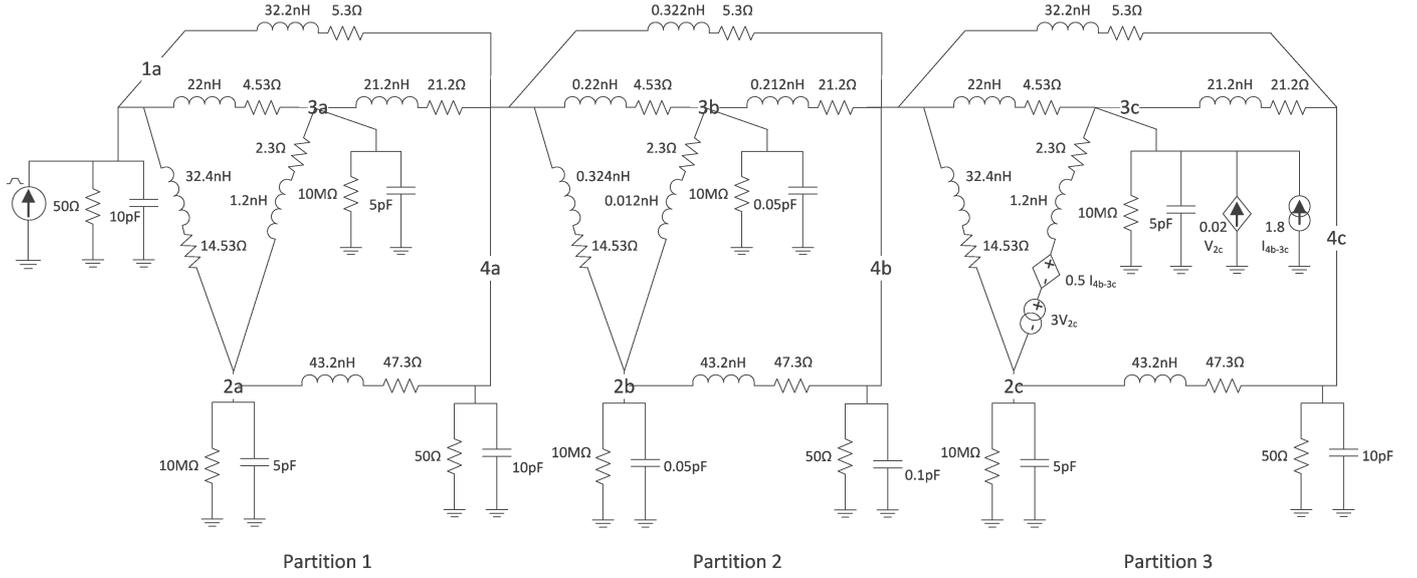


Fig. 5. Example circuit.

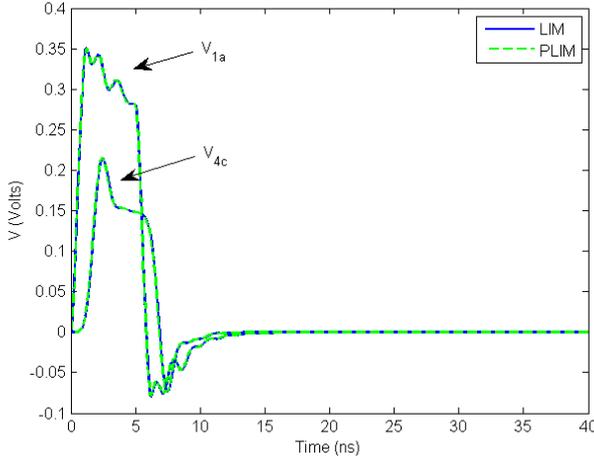


Fig. 6. Simulation of circuit in Fig. 12.

of the circuit. To construct this circuit, partition 1 is cascaded  $N$  times and the simulation time is recorded for PLIM and LIM. The results are summarized in Table 1. We observe that when the sizes of the partitions are comparable, a small speed-up is obtained when using PLIM. On the other hand, when the partition with the smallest latency is small compared to the rest of the circuit, a large speed-up is obtained which approaches the limit of  $\Delta t_{\text{large}}/\Delta t_{\text{min}}$  where  $\Delta t_{\text{large}}$  is the time step of the largest partition and  $\Delta t_{\text{min}}$  is the smallest time step in the circuit, which also dictates the maximum time step of the regular LIM.

### Conclusions

An improved version of the latency insertion method has been proposed along with extensions to handle dependent sources, and a general stability condition for selecting the time step of the simulations with multiple partitions has been presented. It is found that by using PLIM along with the developed stability criteria, a significant speedup in the run time is achieved while preserving the accuracy of the general LIM.

N	LIM (sec)	PLIM (sec)	Speed-up
1	0.11	0.01	11.0
10	0.23	0.02	11.5
100	1.40	0.03	46.7
1000	13.52	0.15	90.1
2000	26.21	0.28	93.6

Table 1. Comparison of run times

### References

- [1] J. E. Schutt-Ainé, "Latency insertion method (LIM) for the fast transient simulation of large networks," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 48, no. 1, pp. 81–89, Jan. 2001.
- [2] S. N. Lalgudi, M. Swaminathan, and Y. Kretchmer, "On-chip power grid simulation using latency insertion method," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 3, pp. 914–931, Apr. 2008.
- [3] R. Gao and J. E. Schutt-Ainé, "Improved latency insertion method for simulation of large networks with low latency," in *Proc. IEEE (EPEP 2002)*, Monterey, CA, Oct., pp. 37–41.
- [4] H. Asai and N. Tsuboi, "Multi-rate latency insertion method with RLCG-MNA formulation for fast transient simulation of large-scale interconnect and plane networks," in *Proc. IEEE (ECTC 2007)*, Reno, NV, May, pp. 1667–1672.
- [5] Z. Deng and J. E. Schutt-Ainé, "Stability analysis of Latency Insertion Method (LIM)," in *Proc. IEEE (EPEP 2004)*, Portland, OR, Oct. 2004, pp. 167–170.
- [6] K. S. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media," *IEEE Trans. Antennas Propagat.*, vol. 14, pp. 302–307, May 1966.
- [7] J. E. Schutt-Ainé, "Stability analysis of the latency insertion method using a block matrix formulation," in *Proc. IEEE (EDAPS 2008)*, Seoul, Korea, Dec., pp. 155–158.