Multi-Thread Assembling for Fast FEM Power Delivery DC Integrity Analysis

Ke Yang*, Shaoyi Peng†, Sheldon X.-D. Tan†, Hai-Bao Chen*
*Department of Micro/Nano-electronics, Shanghai Jiao Tong University, Shanghai 200240, China
†Department of Electrical and Computer Engineering, University of California, Riverside, CA 92521, USA

***Abstract*—Power integrity analysis is of great significance in the field of circuit design, especially the design of modern high speed circuit system. For the high performance printed circuit boards (PCBs) and IC design, power delivery network DC integrity checks play an important role. However, the element assembling process in finite element method (FEM) can take significant portion of total computing time. In this paper, a fast finite element assembling method for power network DC integrity checks of PCBs is proposed. We divided the mesh into a serious of bins and elements in different bins could be assembled in parallel. Further more, a dynamic circle shape approximation method is introduced to further control the number of elements due to vias and circular objectives. As a result, the new solver can easily perform progressive trade off between speed and accuracy. Experimental results of two PCB examples on a 3.6-GHz Intel i7 Dual-core CPU show that the proposed multi-thread assembling method can achieve 2X speedup over existing single-thread assembling methods. A dynamic circle shape approximation method is introduced to further control the number of elements and speed up the solver process. The resulting FEM solver leads to 3X speed over a commercial power integrity solver with no more than 0.7% errors.**

I. Introduction

In recent years, with the increasing complexity and integration of high-speed circuits, multi-power systems are gradually adopted and the supply voltage is continuously reduced which causes excessive DC voltage drop and current density in the power delivery networks (PDNs). These power integrity (PI) problem seriously affects the normal operation of the circuit and even burns the PCB board [1] [2]. Therefore, for high-speed printed circuit board (PCB) and package design, power integrity analysis is an essential and necessary part in the circuit design and verification process.

The electromagnetic parameters in the power delivery networks satisfy the classical electromagnetic field equation - Maxwell's equations which is essentially a set of partial differential equations (PDEs). Most problems in electromagnetic field can be reduced to boundary value problems of differential equations. Over the past decades, a great number of researchers focus on the task of the solution of PDE and have proposed many classical numerical methods including finite element method (FEM) [3]. Finite element method (FEM) is a general numerical approach to solve the problem of power integrity checks with good accuracy. However, FEM always suffers high computational costs and much time consumption due to large number of mesh elements generated.

Fig. 1 shows the practical view of two PDNs 3V3IO_GND selected from Ti WV5_rD and V1P0_S0_GND selected from Intel GalileoGeo which will be used as examples for testing purpose later. As shown in this figure, the practical power delivery networks have many complicated structures (vias, planes), finite element method (FEM) will be a good fit as it can deal with complicated structures and naturally handle the two boundary conditions.

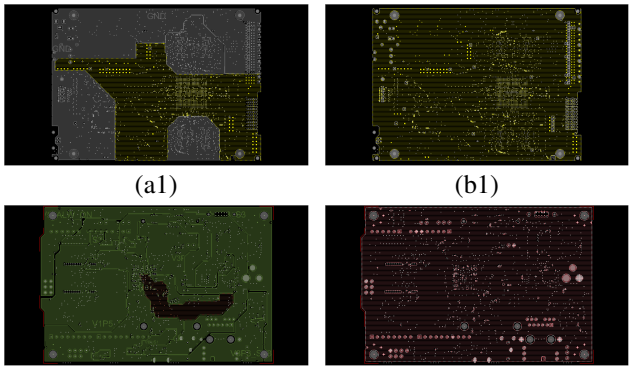


Fig. 1: Practical view of two power delivery networks 3V3IO_GND selected from Ti WV5_rD and V1P0_S0_GND selected from Intel GalileoGeo for testing purpose. (1) PDN 3V3IO_GND: (a1) net 3V3IO, (b1) net GND (WV5_rD); (2) PDN V1P0_S0_GND: (a2) net V1P0_S0, (b2) net GND (GalileoGeo).

In this paper, a fast finite element assembling method for power delivery network DC integrity checks of PCBs is proposed. We divided the mesh generated into a serious of bins and elements in different bins could be assembled in parallel. Experiments on test mesh examples in PCB examples on a 3.6-GHz Intel i7 Dual-core CPU show that the proposed multi-thread assembling method can achieve 2X speedup over existing single-thread assembling methods. Meanwhile, a dynamic circle shape approximation method is introduced to further control the number of elements and speed up the solver process. The resulting FEM solver leads to 3X speed over a commercial power integrity solver with no more than 0.7% errors.

II. Review of FEM for steady state electrical analysis

In this work, we use finite element method (FEM) for solving the current density problems of PCBs and ICs, which are described by the Poisson equations for steady state electrical field [4]. The finite element method divides the continuous solution domain into a group of elements. Thus a continuous infinite degree of freedom problem becomes a discrete finite degree of freedom problem [5]. The finite element method not only has high accuracy, but also can deal with the problem domain of arbitrary complex boundary. In this section we briefly review steady-state electrical field solving process which can be viewed as FEM for Poisson equation with Dirichlet (voltage) and Neumann (current) boundary conditions.

Generally, the steady-state (electrostatic) electric potential in an interconnect, can be modeled by the Poisson equation

978-1-7281-0735-6/19/$31.00 ©2019 IEEE

with Dirichlet and Neumann boundary conditions:

$$\begin{aligned} -\nabla^2 u &= f(x), x \in \Omega \\ u &= g(x), x \in \partial\Omega \cap \Gamma_D \\ \nabla u \cdot \vec{n} &= h(x), x \in \partial\Omega \cap \Gamma_N \end{aligned} \tag{1}$$

where x is the location. $\Omega \subset R^n$ is the solution domain and its boundary is written as $\partial\Omega$. $f(x)$ defined in Ω, represents the electrical charges. While the function $g(x)$ and $h(x)$ are respectively defined on Γ_D and Γ_N, setting the Dirichlet and Neumann boundary conditions which are voltage sources and current sources in the power integrity (PI) analysis. And $u(x)$ is the electrical field solution we want to obtain.

To perform FEM analysis, we try to build the weak form for the differential equation:

$$\nabla \cdot \nabla u_H + f = 0 \tag{2}$$

Given the arbitrary test function v_H, we have:

$$-\int_\Omega \nabla \cdot \nabla u_H \cdot v_H dV = \int_\Omega f \cdot v_H dV \tag{3}$$

By applying the divergence theorem and some manipulation, and considering the Neumann boundary equation, we obtain the weak form of the Poisson problem:

$$\int_\Omega \nabla u_H \cdot \nabla v_H dV = \int_\Omega f \cdot v_H dV + \int_{\partial\Omega \cap \Gamma_N} h v_H dS \tag{4}$$

This equation can be used for each element in the FEM. After we assemble all the equations of element, we will end up with following linear algebraic equation in matrix format:

$$AU = F \tag{5}$$

where vector U is the voltage potential unknown vector on mesh nodes, which can be solved by calculating $U = A^{-1}F$. Assembling matrices A and F, called system matrices, can be divided into calculating each components of them, which requires integration of products of shape functions and other prescribed variables depending on spatial coordinates.

III. Fast Finite Element Assembling

In this section, a new finite element method solver for power delivery network DC integrity checks of PCBs is proposed including multi-thread assembling and dynamic circle shape approximation method, which can save much solver time consumption with good accuracy.

Fig. 2 shows a 3D view of WV5_rD PCB layout in which a, b and c respectively represent sink component (IC load), vertical via and copper clad laminate (CCL). A power delivery network always contains the above three components with electrical source and discrete component. We first extract the geometry of the power delivery network and mesh the extracted geometry into a serious of 2D-triangle elements [6]. Then the FEM solver can start the matrices assembling process.

However, for large-scale PCB layouts and power delivery networks, the number of mesh nodes and elements always are extremely huge even up to several hundred thousand, which will lead to much memory footprint and time consumption. Below we will introduce our method of accelerating the whole finite element solution process.

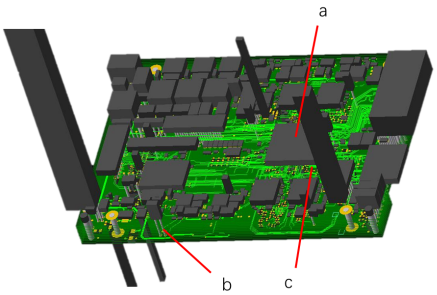


Fig. 2: 3D view of WV5_rD PCB layout. a, b and c respectively represent sink component (IC load), vertical via and copper clad laminate (CCL).

A. Multi-thread assembling

Assembling system matrices A and F and forming as the equation (5) play an important role in the FEM process and sometimes spend more than 10 seconds. In this subsection, we want to divide the whole mesh into a serious of mesh blocks, we called bins. In this way, the matrices assembling process can run at different bins at the same time, i.e., running against multiple threads which can speed up the matrices assembling process. The detail will be introduced below.

A geometry mesh bin generator is implemented based on QuadTree data structure using C++ in order to further speed up the FEM solver assembling process. Every element in original mesh will be identified by a new element type called bin-id. It means that elements with the same bin-id are located in same bin partition and elements in different bins can be assembled in parallel.

According to the literature [7], QuadTree data structure starts with a large rectangular region that contains all the geometries to be put into the tree (bounding box here). This large rectangle will be finally divided into several small rectangular regions i.e., bins. Every bin will be numbered with an integer greater than or equal to 2. If an element is contained by a specific bin, the bin-id of this element will be written as the number (id) of bin, so do other elements. It is important to note that elements intersected with the boundary of bins can not be contained by any bin. In this case, its bin-id will be given the integer 1. Based on property bin-id of elements, we can gather elements with same bin-id and assemble system matrices A and F in parallel.

We can control the number of bins by setting the depth of QuadTree. Fig. 3 provides a 4 bins example to clear the bin generation process. After the bounding box is obtained, subdivide it into four equal parts. Every part will be formed as a child node in QuadTree (store its position, geometry and all elements in it). The elements that hit the bisector line can be put into the root node. In this case, the depth of QuadTree is 2 and 4 bins can be generated at most (some bins can be empty). If you want to obtain more bins, you can subdivide one or more parts till the bins generated are tiny enough.

B. Dynamic circle shape approximation method

The second method we employed is to control the number of mesh elements with acceptable accuracy degradation. Usually, many vias exist in power delivery networks. However, in the finite element method meshing process, in order to restore shape better, the complex geometries such as circles always are with high order linear approximation which leads to dense meshes as shown Fig. 4 (b). Actually, those dense meshes

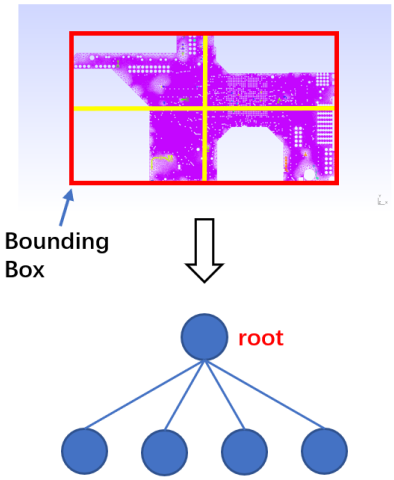
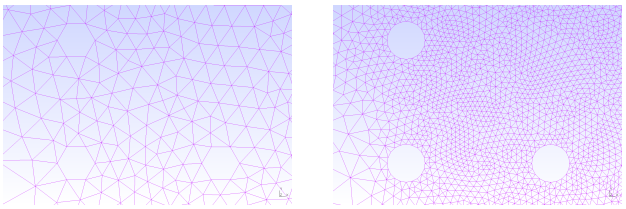


Fig. 3: Bin generation based on QuadTree data structure.



(a) (b)

Fig. 4: Mesh generation results under circle approximation with $n = 6$ (a) and $n = 20$ (b) respectively for same network.

are not necessary because there is no significant change in voltage around these complex shapes at least in our test PCB examples.

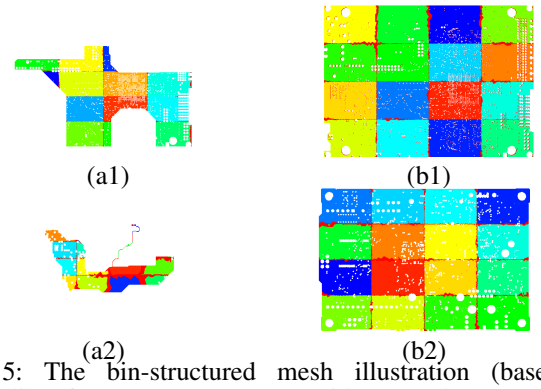
As a result, we introduced a dynamic approximation method for circle shapes. In the geometry generation process, we approximate circular shape with regular polygons. Specifically, take one point p_0 arbitrarily on the circle C, let $p_0 = (x_0, y_0)$ rotate $i\frac{2\pi}{n}, i = 1, 2, ..., n-1$ around the circle center c using Rodrigues' rotation formula:

$$p_i^T = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} p_0^T, \theta = i\frac{2\pi}{n} \quad (6)$$

Sequentially connect points $p_0, p_1, ..., p_{n-1}, p_0$ and the approximation polygon shape is obtained. Fig. 4 shows the mesh generation results under circle approximation with $n = 6$ and $n = 20$ respectively for same network. The situation $n = 6$ achieves significantly decrease on the number of mesh nodes and elements. Most of all, we can control the number of mesh nodes and elements by setting the value of n. Usually, n is between 3 and 20. The bigger the number of polygon edges is, the more dense mesh will be generated. However, too small value causes too sparse mesh which will cause serious accuracy less. In our experiments, n is suitable for about 6.

IV. Numerical Results and Analysis

In order to confirm the performance of the proposed method, the new FEM solver with multi-thread assembling and dynamic circle shape approximation method is implemented using C++. We use the Gmsh program to generate the 2D mesh for our FEM solver [6]. All experiments are implemented on a PC with 8-GB RAM and 3.6-GHz Intel i7 Dual-Core CPU.



(a1) (b1)

(a2) (b2)

Fig. 5: The bin-structured mesh illustration (based on VTK lib) for two test power grid networks. (a1) Mesh of net 3V3IO in PDN 3V3IO_GND. (b1) Mesh of net GND in PDN 3V3IO_GND. (a2) Mesh of net V1P0_S0 in PDN V1P0_S0_GND. (b2) Mesh of net GND in PDN V1P0_S0_GND.

TABLE I: Time consumption of single-thread assembling and multi-thread assembling on two test PDNs 3V3IO_GND and V1P0_S0_GND .

PDN	Net	Single-thread assembling (sec)	Multi-thread assembling (sec)	Speed up
3V3IO_GND	3V3IO	6.1057	2.2069	2.77X
	GND	14.0390	7.0102	2.00X
V1P0_S0_GND	V1P0_S0	0.0962	0.0423	2.27X
	GND	1.7597	0.8064	2.17X

We measure the time consumption of FEM solver with single-thread assembling and multi-thread assembling on two industry PCB layout designs including Ti WV5_rD (see Fig. 1 (1) and Fig. 2) and Intel GalileoGeo (see Fig. 1 (2)). Meanwhile, a 3D bin visualization program based on VTK lib has been developed for displaying the bin partitions intuitively. Two power delivery networks tested here are selected from Ti WV5_rD and Intel GalileoGen respectively named 3V3IO_GND (see Fig. 1 (1)) and V1P0_S0_GND (see Fig. 1 (2)). 3V3IO_GND consists of one source component pin C75.1 with voltage 1V, several pins of sink component U1 with total 10A current and 3V3IO net, GND net. While V1P0_S0_GND consists of one source component pin C3B9.1 with voltage 1V, several pins of sink component U2A5 with total 10A current and V1P0_S0 net, GND net. Besides, we set the copper conductivity of two examples with 5.959×10^7S/m.

It is important to note that for those industry examples, a large number of elements are generated. In PDN 3V3IO_GND, the mesh of net 3V3IO has 304687 nodes and 640201 elements while mesh of net GND has 884577 nodes and 1961773 elements. In PDN V1P0_S0_GND, the mesh of net V1P0_S0 has 11450 nodes and 26397 elements while mesh of net GND has 156612 nodes and 364555 elements. Fig. 5 shows the bin-structured mesh illustration (based on VTK lib) for two

TABLE II: Performance comparisons of different circle approximation $n = 6$ and $n = 20$ on the test PDN 3V3IO_GND from PCB layout Ti WV5_rD.

Indicator	3V3IO	3V3IO	GND	GND
Circle approximation	$n = 6$	$n = 20$	$n = 6$	$n = 20$
Meshing time (sec)	5.6591	29.7241	43.8979	125.7343
Number of Nodes	64344	304687	421580	884577
Number of elements	142255	640201	944918	1961773
Solver time (sec)	0.6056	4.8563	13.6199	46.9803

TABLE III: Performance comparisons of different circle approximation $n = 6$ and $n = 20$ on the test PDN V1P0_S0_GND from PCB layout Intel GalileoGen.

Indicator	V1P0_S0	V1P0_S0	GND	GND
Circle approximation	$n = 6$	$n = 20$	$n = 6$	$n = 20$
Meshing time (sec)	0.4435	0.8546	7.8216	14.7443
Number of Nodes	5298	11450	82720	156612
Number of elements	12404	26397	191044	364555
Solver time (sec)	0.4262	0.9719	0.5709	1.1022

TABLE IV: Comparisons of the pure solver CPU time between FEM Solver ($n = 6$) and PowerDC on two test PDNs 3V3IO_GND and V1P0_S0_GND.

PDN	Net	Solver time (PowerDC, sec)	Solver time (FEM Solver, $n = 6$, sec)	Speed up
3V3IO_GND	3V3IO	1.5529	0.6056	2.56X
	GND	15.2497	13.6199	1.12X
V1P0_S0_GND	V1P0_S0	0.8735	0.4262	2.05X
	GND	1.7317	0.5709	3.03X

test PDNs. Nets 3V3IO and GND of PDN 3V3IO_GND are divided into 13 bins and 16 bins, respectively. Considered elements on the bin boundary, 14 and 17 threads are applied to the system matrices assembling process. Similarly, 16 and 17 threads are applied to the system matrices assembling for two nets of PDN V1P0_S0_GND. Time consumption of single-thread assembling and multi-thread assembling on the test PDNs is presented in Table I. The matrices assembling process on all four nets in two PDNs achieves 2X speedup with multi-thread assembling over existing single-thread assembling methods.

Besides, the dynamic approximation method for circle shapes is implemented. Table II and Table III respectively show the performance in meshing for $n = 6$ and $n = 20$ on the test PDNs 3V3IO_GND and V1P0_S0_GND. Meanwhile, Fig. 6, shows the comparisons of simulation results with Cadence Sigrity PowerDC using same PCB layouts, PDN and simulation settings. As we can see the errors are consistent over different pins for each example. Besides, we show the average relative error between the FEM solver ($n = 6$, $n = 20$) and PowerDC for two power grid examples in Table V. As we can see, for PDN 3V3IO_GND, the error of FEM solver ($n = 6$) is only about 0.1% higher than the FEM solver ($n = 20$) but with 2X speedup both in solver and meshing process. For PDN V1P0_S0_GND, the FEM solver ($n = 6$) can be 2X faster than PowerDC in solver and meshing process. As a result, the new solver can easily perform progressive trade off between speed and accuracy by setting the value of n. However, no matter what value n takes (between 3 and 20), the error would be no more than 0.7% in our experiments.

Further more, Table IV shows the pure solver CPU time comparisons of the FEM solver ($n = 6$) and PowerDC. Thanks to the 2D finite elements for space discretization method in literature [6], the pure solver CPU time of our FEM solver under the situation $n = 6$ achieves 1-3X speedup compared with PowerDC on the test PDNs.

V. Conclusion

In this paper, a fast finite element assembling method for power delivery network DC integrity checks of PCBs has been proposed. We divided the mesh generated into a serious of bins and elements in different bins could be assembled in parallel. Experimental results of two PCB examples show that that the proposed multi-thread assembling method can achieve 2X speedup over existing single-thread assembling methods.

TABLE V: Average relative error of voltage between FEM solver ($n = 6$, $n = 20$) and PowerDC on two test PDNs 3V3IO_GND and V1P0_S0_GND.

PDN	3V3IO_GND		V1P0_S0_GND	
Circle approximation	$n = 6$	$n = 20$	$n = 6$	$n = 20$
Average relative error	0.6393%	0.5401%	0.0621%	0.0221%

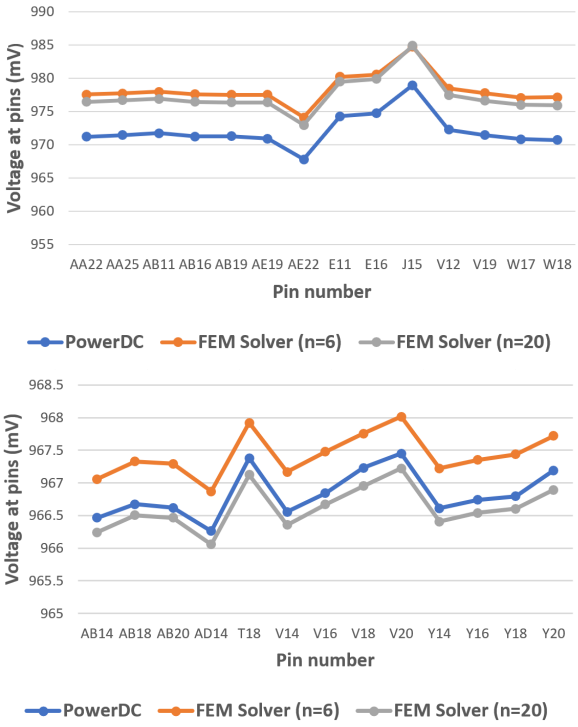


Fig. 6: Voltage at pins comparisons between FEM solver ($n = 6$), FEM solver ($n = 20$) and PowerDC used two test PDNs. The first curve is corresponding with PDN 3V3IO_GND and the other is corresponding with PDN V1P0_S0_GND.

Further more, a shape approximation method is proposed and the resulting FEM solver can lead to 3X speed over a commercial power integrity solver with no more than 0.7% errors.

References

[1] V. Sridharan, M. Swaminathan, and T. Bandyopadhyay, "Enhancing signal and power integrity using double sided silicon interposer," *IEEE Microwave and Wireless Components Letters*, vol. 21, no. 11, pp. 598–600, 2011.

[2] I. Erdin, R. Achar, and K. Erdin, "Power integrity aware approach to dynamic analysis of buck converters," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 8, no. 1, pp. 32–40, 2018.

[3] A. Ruehli, "Equivalent circuit models for three-dimensional multiconductor systems," *IEEE Transactions on Microwave Theory and Techniques*, vol. 22, no. 3, pp. 216–221, 1974.

[4] Y. Takahashi and S. Wakao, "Large-scale analysis of eddy-current problems by the hybrid finite element-boundary element method combined with the fast multipole method," *IEEE Transactions on Magnetics*, vol. 42, no. 4, pp. 671–674, 2006.

[5] L. Qi, X. Cui, Z. Zhao, and H. Li, "Grounding performance analysis of the substation grounding grids by finite element method in frequency domain," *IEEE Transactions on Magnetics*, vol. 43, no. 4, pp. 1181–1184, 2007.

[6] W. He, H. Zhao, Z. Qi, H.-B. Chen, and S. X.-D. Tan, "Fast two-dimensional finite element analysis for power network dc integrity checks of pcbs," in *11th IEEE International Conference on ASIC (ASICON 2017)*, October 2017.

[7] J. Rosenberg, "Geographical data structures compared: A study of data structures supporting region queries," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 4, no. 1, pp. 53–67, 1985.