3-D Impedance Extraction Algorithm Using Mixed Boundary Element Method

Fang Gong[1], Wenjian Yu[1], Zeyi Wang[1],
Zhiping Yu[2], Changhao Yan[3]

[1]EDA Lab, Dept. Computer Science & Technology,
Tsinghua University, Beijing, 100084, China

[2]Institute of Microelectronics,
Tsinghua University, Beijing, 100084, China

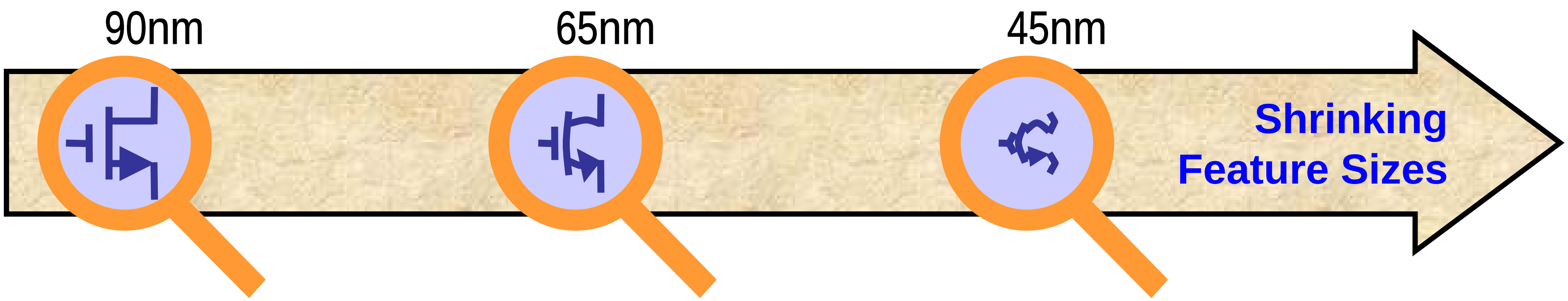
[3]Department of Microelectronics,
Fudan University, Shanghai, 201203, China

Outline

- **Introduction**
- **Background**
- **Extraction with Mixed BEM**
- **Numerical examples**
- **Conclusion**

Introduction

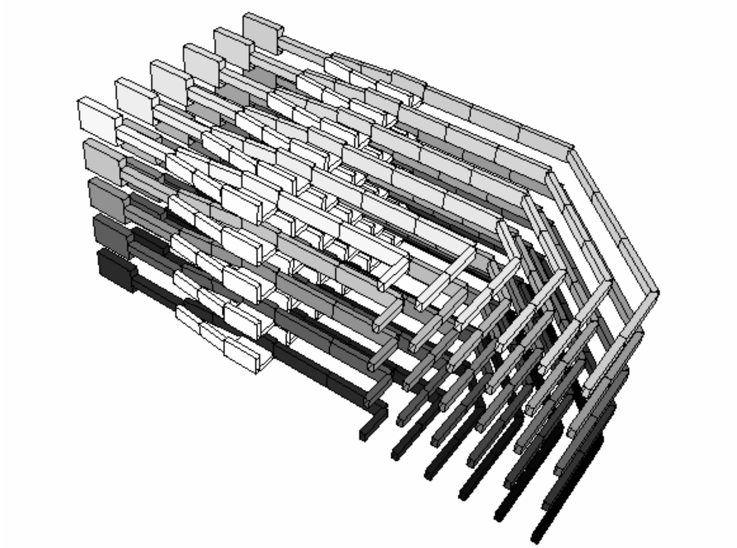
- Ultra Deep Submicron Design and Challenges
  - Shrinking feature Size (45nm)
  - Operating frequency: multiple GHz
  - Circuit Scale: larger and more complicated
  - High Frequency Effects (Especially Inductance effects)



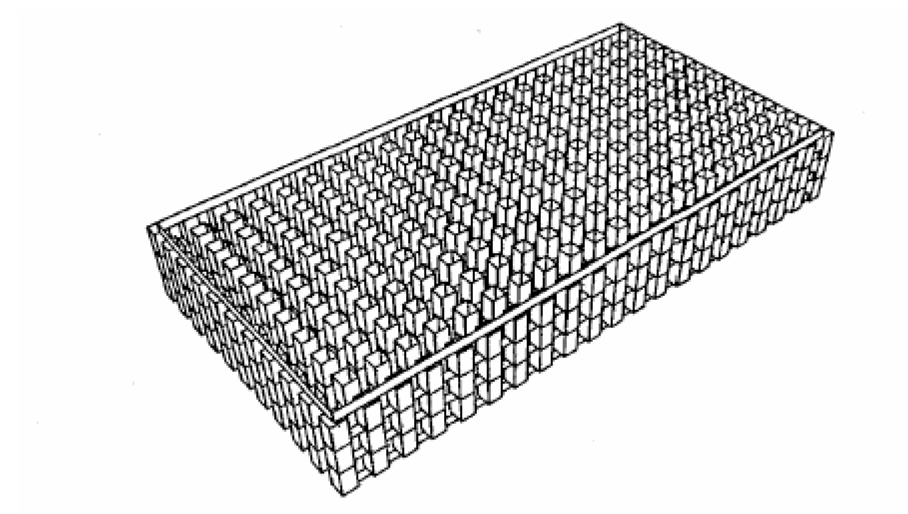
Accurate extraction of impedance becomes much more important!

Inductance Extraction Methods

- Partial element equivalent circuit (PEEC) Method
  - Volume discretization;
  - Finer discretization for high-frequency effects;
  - Large variables (Especially for “block” conductors).



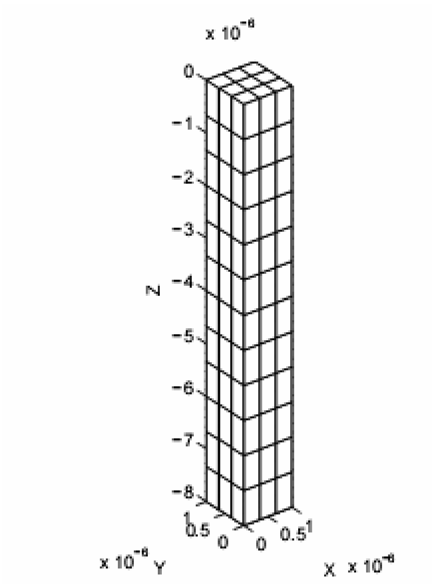
(a)pin-connect (FastHenry)



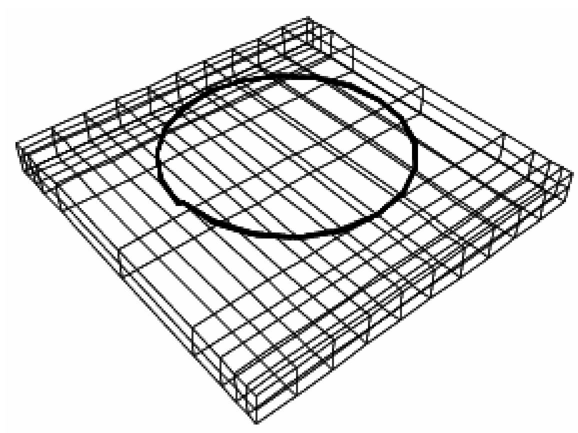
(b) ground plane

Inductance Extraction Methods

- Boundary Element Method (BEM)
  - discretize only the conductor surface into panels (fewer variables);
  - flexible for complicated structures;
  - convenient to capture the skin and proximity effect accurately;



(a) A single wire



(b) A spiral upon ground

Outline

- ❖ Introduction
- ❖ **Background**
- ❖ Extraction with Mixed BEM
- ❖ Numerical examples
- ❖ Conclusion

Background

- Direct Boundary Element Method
  - Junfeng Wang proposed direct boundary element integral equations (Ph.D. dissertation, Mass. Inst. Technol., Cambridge, MA, 1999.) .
  - Zhenhai Zhu (Ph.D,2005 M.I.T.) implemented a prototype “FastImp”. (*IEEE Trans. CAD*, vol. 24, no. 7, pp. 981-998, 2005)
  - Extract Impedance accurately in high frequencies;
  - Reduce calculation consumption.

- Existing Problems
  - Accuracy problem in low-frequencies.
  - Large number of variables.

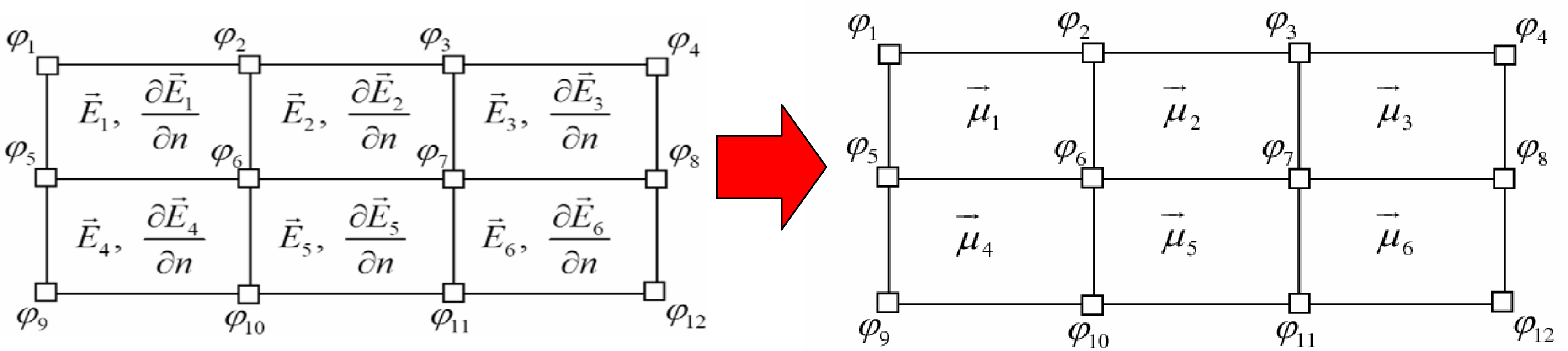
Background

- Mixed Boundary Element Method
  - electric field within each conductor is described with indirect boundary integral equations;
  - electric field outside conductors, direct boundary integral equations as FastImp is adopted.

	Differential Equations		Integral Equations & boundary condition
Interior	$\nabla^2\vec{E} - i\omega\mu\sigma\vec{E} = 0$	**Direct**	$\int G_1(x,y)\frac{\partial \vec{E}(y)}{\partial n_y}dy - \int \frac{\partial G_1(x,y)}{\partial n_y}\vec{E}(y)dy = \frac{1}{2}\vec{E}(x)$
		Indirect	$\begin{cases} \vec{E}(y) = \int_{S_i} \frac{\partial G(y,x)}{\partial n(x)}\vec{\mu}(x)dS_x \\ \frac{\partial \vec{E}(y)}{\partial n(y)} = \frac{\partial}{\partial n(y)}\left[\int_{S_i}\frac{\partial G(y,x)}{\partial n(x)}\vec{\mu}(x)dS_x\right] \end{cases}$
Exterior	$\left.\begin{array}{c} \nabla^2\vec{E} = i\omega\mu\vec{J} \\ -\nabla\varphi = \vec{E} + i\omega\vec{A} \\ \vec{A} = \int \mu G_0(x,y)\vec{J}dy \end{array}\right\}$		$\int G_0(x,y)\frac{\partial \vec{E}(y)}{\partial n_y}dy - \int \frac{\partial G_0(x,y)}{\partial n_y}\vec{E}(y)dy + \nabla\varphi(x) = 0$

Background

- Advantage of Mixed BEM
  - fewer variables than direct BEM.
  - Fast convergence of Krylov subspace iterative solver (GMRES)



- Drawbacks of Mixed BEM
  - Complex boundary integrals.
  - More dense-matrix computations

Outline

- Introduction
- Background
- **Extraction with Mixed BEM**
- Numerical examples
- Conclusion

Problem Analysis (MQS)

$$
\begin{pmatrix}
\boldsymbol{T}_{1x}\boldsymbol{Q}_1 & \boldsymbol{T}_{1y}\boldsymbol{Q}_1 & \boldsymbol{T}_{1z}\boldsymbol{Q}_1 & \boldsymbol{A}_{t1} \\
\boldsymbol{T}_{2x}\boldsymbol{Q}_1 & \boldsymbol{T}_{2y}\boldsymbol{Q}_1 & \boldsymbol{T}_{2z}\boldsymbol{Q}_1 & \boldsymbol{A}_{t2} \\
\boldsymbol{N}_{ncx}\boldsymbol{D}_1 & \boldsymbol{N}_{ncy}\boldsymbol{D}_1 & \boldsymbol{N}_{ncz}\boldsymbol{D}_1 & 0 \\
\boldsymbol{T}_{1x}\boldsymbol{D}_1 & \boldsymbol{T}_{1y}\boldsymbol{D}_1 & \boldsymbol{T}_{1z}\boldsymbol{D}_1 & 0 \\
\boldsymbol{T}_{2x}\boldsymbol{D}_1 & \boldsymbol{T}_{2y}\boldsymbol{D}_1 & \boldsymbol{T}_{2z}\boldsymbol{D}_1 & 0 \\
\boldsymbol{N}_{cx}\boldsymbol{D}_2 & \boldsymbol{N}_{cy}\boldsymbol{D}_2 & \boldsymbol{N}_{cz}\boldsymbol{D}_2 & 0 \\
\boldsymbol{Q}_{2x} & \boldsymbol{Q}_{2y} & \boldsymbol{Q}_{2z} & 0 \\
0 & 0 & 0 & \boldsymbol{I}_c
\end{pmatrix}
\begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \\ \boldsymbol{\mu}_z \\ \boldsymbol{\varphi} \end{bmatrix}
=
\begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \boldsymbol{b}_3 \\ \boldsymbol{b}_4 \\ \boldsymbol{b}_5 \\ \boldsymbol{b}_6 \\ \boldsymbol{b}_7 \\ \boldsymbol{b}_8 \end{bmatrix}
$$

$$
\begin{cases}
P_{\boldsymbol{0}}(a,b) = \int_{panel_b} \frac{1}{4\pi r(y_a, x)} dx \\
D_{\boldsymbol{0}}(a,b) = \int_{panel_b} \frac{\partial}{\partial n_x}\left[\frac{1}{4\pi r(y_a, x)}\right] dx \\
D_1(a,b) = \int_{panel_b} \frac{\partial}{\partial n_x}\left[\frac{e^{ikr(y_a, x)}}{4\pi r(y_a, x)}\right] dx \\
D_2(a,b) = \frac{\partial}{\partial n(y_a)} \int_{panel_b} \frac{\partial}{\partial n_x}\left[\frac{e^{-ikr(y_a, x)}}{4\pi r(y_a, x)}\right] dx
\end{cases}
$$

- Extra Matrix Multiplication
- Matrix-Vector Product
- Post-Process Matrices
- Efficient techniques for linear equation system solution.

$$\boldsymbol{Q}_1 = \boldsymbol{P}_0\boldsymbol{D}_2 - \boldsymbol{D}_0\boldsymbol{D}_1 \ ,$$

$$\boldsymbol{Q}_{2\alpha} = \boldsymbol{C}_{d\alpha}\boldsymbol{D}_2 + \boldsymbol{C}_{\alpha}\boldsymbol{D}_1, \ \ \alpha = x, y, z,$$

(1) *Extra Matrix Multiplication & Matrix Vector Product*

$$
\begin{pmatrix}
\boldsymbol{T}_{1x}\boldsymbol{Q}_1 & \boldsymbol{T}_{1y}\boldsymbol{Q}_1 & \boldsymbol{T}_{1z}\boldsymbol{Q}_1 & \boldsymbol{A}_{t1} \\
\boldsymbol{T}_{2x}\boldsymbol{Q}_1 & \boldsymbol{T}_{2y}\boldsymbol{Q}_1 & \boldsymbol{T}_{2z}\boldsymbol{Q}_1 & \boldsymbol{A}_{t2} \\
\boldsymbol{N}_{ncx}\boldsymbol{D}_1 & \boldsymbol{N}_{ncy}\boldsymbol{D}_1 & \boldsymbol{N}_{ncz}\boldsymbol{D}_1 & 0 \\
\boldsymbol{T}_{1x}\boldsymbol{D}_1 & \boldsymbol{T}_{1y}\boldsymbol{D}_1 & \boldsymbol{T}_{1z}\boldsymbol{D}_1 & 0 \\
\boldsymbol{T}_{2x}\boldsymbol{D}_1 & \boldsymbol{T}_{2y}\boldsymbol{D}_1 & \boldsymbol{T}_{2z}\boldsymbol{D}_1 & 0 \\
\boldsymbol{N}_{cx}\boldsymbol{D}_2 & \boldsymbol{N}_{cy}\boldsymbol{D}_2 & \boldsymbol{N}_{cz}\boldsymbol{D}_2 & 0 \\
\boldsymbol{Q}_{2x} & \boldsymbol{Q}_{2y} & \boldsymbol{Q}_{2z} & 0 \\
0 & 0 & 0 & \boldsymbol{I}_c
\end{pmatrix}
\begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \\ \boldsymbol{\mu}_z \\ \boldsymbol{\varphi} \end{bmatrix}
=
\begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \boldsymbol{b}_3 \\ \boldsymbol{b}_4 \\ \boldsymbol{b}_5 \\ \boldsymbol{b}_6 \\ \boldsymbol{b}_7 \\ \boldsymbol{b}_8 \end{bmatrix}
$$

$$\boldsymbol{Q}_1 = \boldsymbol{P}_0\boldsymbol{D}_2 - \boldsymbol{D}_0\boldsymbol{D}_1 \ ,$$

$$\boldsymbol{Q}_{2\alpha} = \boldsymbol{C}_{d\alpha}\boldsymbol{D}_2 + \boldsymbol{C}_{\alpha}\boldsymbol{D}_1, \ \alpha = x, y, z,$$

- Computation complexity of $\boldsymbol{Q}_1$ Matrix is $O(n^3)$

$$\boldsymbol{Q}_1 = \boldsymbol{P}_0\boldsymbol{D}_2 - \boldsymbol{D}_0\boldsymbol{D}_1$$

- Krylov subspace iterative solver only need to calculate matrix-vector product (*Ax*)

$$\boldsymbol{Q}_1\boldsymbol{\mu}_x = (\boldsymbol{P}_0\boldsymbol{D}_2 - \boldsymbol{D}_0\boldsymbol{D}_1)\boldsymbol{\mu}_x = [\boldsymbol{P}_0(\boldsymbol{D}_2\boldsymbol{\mu}_x) - \boldsymbol{D}_0(\boldsymbol{D}_1\boldsymbol{\mu}_x)]$$

- With reuse scheme, the integral-related matrix operations in “*Ax*” can be **reduced from 42 to 12**.

(1) *Extra Matrix Multiplication & Matrix Vector Product (cont.)*

- pFFT algorithm (*precorrected fast fourier transform*)
  - pFFT algorithm can accelerate the integral-related Matrix-vector product, and reduce the complexity from $\boldsymbol{O(n^2)}$ to $\mathbf{O(\boldsymbol{n})}$**.** (*IEEE Trans. CAD*, vol. 16, no. 10, pp. 1059-1072, Oct. 1997.)
  - Accelerate the integral computation of integral entries in matrices.
  - Reduce the memory used by storing integral matrices "**implicitly**".

- The efficiency of pFFT acceleration can be verified by experiments in "Numerical Results" Section.

(2) *Post-Process Matrices*

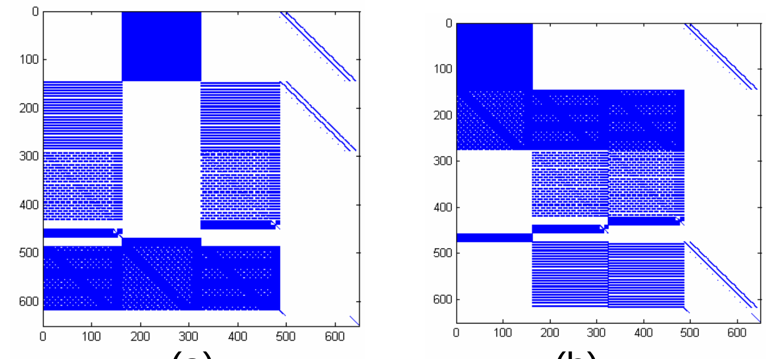
- Problem description
  - In Mixed BEM, $\vec{E}$ and $\partial\vec{E}/\partial n$ can expressed with imaginary dipole distribution $\vec{\mu}$. But the computation of current *Ic* need to convert the solved dipole $\vec{\mu}$ into electrical field $\vec{E}$ and its derivative;
  - Post-Process Matrices D_1 and D_2 are used for this conversion;
  - With pFFT algorithm, D_1 and D_2 are not formed explicitly;
  - Only rows corresponding to contact panels can be used.
- Possible solution
  - pFFT method: form the complete D_1 and D_2 with pFFT
  - Integration Method: Compute necessary row entries with integration.

Comparison on Post-process Matrices among Different Methods
Time unit is Second. Memory unit is MB.

	Total panel	Contact panel	Ratio	pFFT Method		Integration Method	
				Time	Memory	Time	Memory
Example1	6364	508	8%	181.4	83	32.54	3
Example2	17472	192	1%	653.9	387	56.29	4

(3) *Scaling Technique*

- Matrix organization
  - Disperse nonzero entries at or near the diagonal in order to reduce the condition number.
  - The operation can be shown as follows, and resultant matrix is shown in Fig. 1



(a) (b)

The nonzero distribution of: (a) the coefficient matrix without matrix organization, (b) the coefficient matrix with matrix organization.

$$
\begin{pmatrix}
\boldsymbol{T}_{1x}\boldsymbol{Q}_1 & \boldsymbol{T}_{1y}\boldsymbol{Q}_1 & \boldsymbol{T}_{1z}\boldsymbol{Q}_1 & \boldsymbol{A}_{t1} \\
\boldsymbol{T}_{2x}\boldsymbol{Q}_1 & \boldsymbol{T}_{2y}\boldsymbol{Q}_1 & \boldsymbol{T}_{2z}\boldsymbol{Q}_1 & \boldsymbol{A}_{t2} \\
\boldsymbol{N}_{ncx}\boldsymbol{D}_1 & \boldsymbol{N}_{ncy}\boldsymbol{D}_1 & \boldsymbol{N}_{ncz}\boldsymbol{D}_1 & 0 \\
\boldsymbol{T}_{1x}\boldsymbol{D}_1 & \boldsymbol{T}_{1y}\boldsymbol{D}_1 & \boldsymbol{T}_{1z}\boldsymbol{D}_1 & 0 \\
\boldsymbol{T}_{2x}\boldsymbol{D}_1 & \boldsymbol{T}_{2y}\boldsymbol{D}_1 & \boldsymbol{T}_{2z}\boldsymbol{D}_1 & 0 \\
\boldsymbol{N}_{cx}\boldsymbol{D}_2 & \boldsymbol{N}_{cy}\boldsymbol{D}_2 & \boldsymbol{N}_{cz}\boldsymbol{D}_2 & 0 \\
\boldsymbol{Q}_{2x} & \boldsymbol{Q}_{2y} & \boldsymbol{Q}_{2z} & 0 \\
0 & 0 & 0 & \boldsymbol{I}_c
\end{pmatrix}
\begin{bmatrix} \mu_x \\ \mu_y \\ \mu_z \\ \varphi \end{bmatrix}
=
\begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \boldsymbol{b}_3 \\ \boldsymbol{b}_4 \\ \boldsymbol{b}_5 \\ \boldsymbol{b}_6 \\ \boldsymbol{b}_7 \\ \boldsymbol{b}_8 \end{bmatrix}
\quad \Longrightarrow \quad
\begin{pmatrix}
\boldsymbol{T}_{1y}\boldsymbol{Q}_1 & \boldsymbol{T}_{1x}\boldsymbol{Q}_1 & \boldsymbol{T}_{1z}\boldsymbol{Q}_1 & \boldsymbol{A}_{t1} \\
\boldsymbol{Q}_{2y} & \boldsymbol{Q}_{2x} & \boldsymbol{Q}_{2z} & 0 \\
\boldsymbol{N}_{ncy}\boldsymbol{D}_1 & \boldsymbol{N}_{ncx}\boldsymbol{D}_1 & \boldsymbol{N}_{ncz}\boldsymbol{D}_1 & 0 \\
\boldsymbol{T}_{1y}\boldsymbol{D}_1 & \boldsymbol{T}_{1x}\boldsymbol{D}_1 & \boldsymbol{T}_{1z}\boldsymbol{D}_1 & 0 \\
\boldsymbol{T}_{2y}\boldsymbol{D}_1 & \boldsymbol{T}_{2x}\boldsymbol{D}_1 & \boldsymbol{T}_{2z}\boldsymbol{D}_1 & 0 \\
\boldsymbol{N}_{cy}\boldsymbol{D}_2 & \boldsymbol{N}_{cx}\boldsymbol{D}_2 & \boldsymbol{N}_{cz}\boldsymbol{D}_2 & 0 \\
\boldsymbol{T}_{2y}\boldsymbol{Q}_1 & \boldsymbol{T}_{2x}\boldsymbol{Q}_1 & \boldsymbol{T}_{2z}\boldsymbol{Q}_1 & \boldsymbol{A}_{t2} \\
0 & 0 & 0 & \boldsymbol{I}_c
\end{pmatrix}
\begin{bmatrix} \mu_y \\ \mu_x \\ \mu_z \\ \varphi \end{bmatrix}
=
\begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_7 \\ \boldsymbol{b}_3 \\ \boldsymbol{b}_4 \\ \boldsymbol{b}_5 \\ \boldsymbol{b}_6 \\ \boldsymbol{b}_2 \\ \boldsymbol{b}_8 \end{bmatrix}
$$

(3) *Scaling Technique (cont.)*

- Scaling techniques
  - Suppose the average edge length of panels is *O(u)*, the scale of each matrix block can be estimated in terms of *u*.

$$\begin{pmatrix} \boldsymbol{T}_{1y}\boldsymbol{Q}_1(1) & \boldsymbol{T}_{1x}\boldsymbol{Q}_1(1) & \boldsymbol{T}_{1z}\boldsymbol{Q}_1(1) & \boldsymbol{A}_{t1}(1) \\ \boldsymbol{Q}_{2y}(u) & \boldsymbol{Q}_{2x}(u) & \boldsymbol{Q}_{2z}(u) & 0 \\ \boldsymbol{N}_{ncy}\boldsymbol{D}_1(1) & \boldsymbol{N}_{ncx}\boldsymbol{D}_1(1) & \boldsymbol{N}_{ncz}\boldsymbol{D}_1(1) & 0 \\ \boldsymbol{T}_{1y}\boldsymbol{D}_1(1) & \boldsymbol{T}_{1x}\boldsymbol{D}_1(1) & \boldsymbol{T}_{1z}\boldsymbol{D}_1(1) & 0 \\ \boldsymbol{T}_{2y}\boldsymbol{D}_1(1) & \boldsymbol{T}_{2x}\boldsymbol{D}_1(1) & \boldsymbol{T}_{2z}\boldsymbol{D}_1(1) & 0 \\ \boldsymbol{N}_{cy}\boldsymbol{D}_2(1) & \boldsymbol{N}_{cx}\boldsymbol{D}_2(1) & \boldsymbol{N}_{cz}\boldsymbol{D}_2(1) & 0 \\ \boldsymbol{T}_{2y}\boldsymbol{Q}_1(1) & \boldsymbol{T}_{2x}\boldsymbol{Q}_1(1) & \boldsymbol{T}_{2z}\boldsymbol{Q}_1(1) & \boldsymbol{A}_{t2}(1) \\ 0 & 0 & 0 & \boldsymbol{I}_c(1) \end{pmatrix} \begin{bmatrix} \mu_y \\ \mu_x \\ \mu_z \\ \varphi \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_7 \\ \boldsymbol{b}_3 \\ \boldsymbol{b}_4 \\ \boldsymbol{b}_5 \\ \boldsymbol{b}_6 \\ \boldsymbol{b}_2 \\ \boldsymbol{b}_8 \end{bmatrix}$$

$$\Rightarrow$$

$$\begin{pmatrix} \boldsymbol{T}_{1y}\boldsymbol{Q}_1(1) & \boldsymbol{T}_{1x}\boldsymbol{Q}_1(1) & \boldsymbol{T}_{1z}\boldsymbol{Q}_1(1) & \boldsymbol{A}_{t1}(1) \\ \frac{1}{u}\boldsymbol{Q}_{2y}(u) & \frac{1}{u}\boldsymbol{Q}_{2x}(u) & \frac{1}{u}\boldsymbol{Q}_{2z}(u) & 0 \\ \boldsymbol{N}_{ncy}\boldsymbol{D}_1(1) & \boldsymbol{N}_{ncx}\boldsymbol{D}_1(1) & \boldsymbol{N}_{ncz}\boldsymbol{D}_1(1) & 0 \\ \boldsymbol{T}_{1y}\boldsymbol{D}_1(1) & \boldsymbol{T}_{1x}\boldsymbol{D}_1(1) & \boldsymbol{T}_{1z}\boldsymbol{D}_1(1) & 0 \\ \boldsymbol{T}_{2y}\boldsymbol{D}_1(1) & \boldsymbol{T}_{2x}\boldsymbol{D}_1(1) & \boldsymbol{T}_{2z}\boldsymbol{D}_1(1) & 0 \\ \boldsymbol{N}_{cy}\boldsymbol{D}_2(1) & \boldsymbol{N}_{cx}\boldsymbol{D}_2(1) & \boldsymbol{N}_{cz}\boldsymbol{D}_2(1) & 0 \\ \boldsymbol{T}_{2y}\boldsymbol{Q}_1(1) & \boldsymbol{T}_{2x}\boldsymbol{Q}_1(1) & \boldsymbol{T}_{2z}\boldsymbol{Q}_1(1) & \boldsymbol{A}_{t2}(1) \\ 0 & 0 & 0 & \boldsymbol{I}_c(1) \end{pmatrix} \begin{bmatrix} \mu_y \\ \mu_x \\ \mu_z \\ \varphi \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1 \\ \frac{1}{u}\boldsymbol{b}_7 \\ \boldsymbol{b}_3 \\ \boldsymbol{b}_4 \\ \boldsymbol{b}_5 \\ \boldsymbol{b}_6 \\ \boldsymbol{b}_2 \\ \boldsymbol{b}_8 \end{bmatrix}$$

The Number of GMRES Iteration With or Without Organization& Scaling

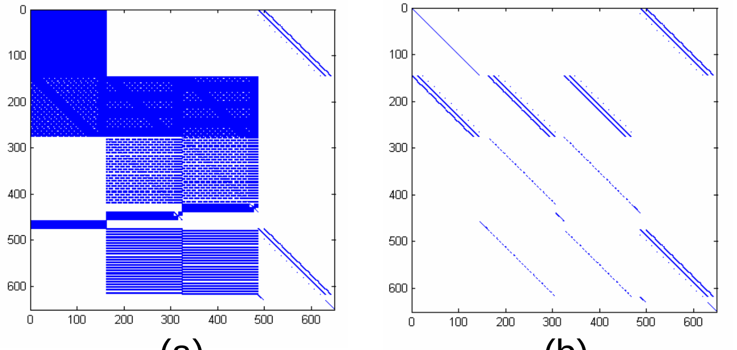
	Number of Iteration in GMRES	
	Without	With
Wire	9	7
Bus2x2	22	13

(4) *Preconditioning*

- The efficiency of preconditioner is very important for the convergence rate of the Krylov subspace iterative solver.

- The construction of preconditioning matrix (the superscript *D* indicates the diagonal part of matrix)

$$
\begin{pmatrix}
\boldsymbol{T}_{1y}\boldsymbol{Q}_1^D & \boldsymbol{T}_{1x}\boldsymbol{Q}_1^D & \boldsymbol{T}_{1z}\boldsymbol{Q}_1^D & \boldsymbol{A}_{t1} \\
\frac{1}{u}\boldsymbol{Q}'_{2y} & \frac{1}{u}\boldsymbol{Q}'_{2x} & \frac{1}{u}\boldsymbol{Q}'_{2z} & 0 \\
\boldsymbol{N}_{ncy}\boldsymbol{D}_1^D & \boldsymbol{N}_{ncx}\boldsymbol{D}_1^D & \boldsymbol{N}_{ncz}\boldsymbol{D}_1^D & 0 \\
\boldsymbol{T}_{1y}\boldsymbol{D}_1^D & \boldsymbol{T}_{1x}\boldsymbol{D}_1^D & \boldsymbol{T}_{1z}\boldsymbol{D}_1^D & 0 \\
\boldsymbol{T}_{2y}\boldsymbol{D}_1^D & \boldsymbol{T}_{2x}\boldsymbol{D}_1^D & \boldsymbol{T}_{2z}\boldsymbol{D}_1^D & 0 \\
\boldsymbol{N}_{cy}\boldsymbol{D}_2^D & \boldsymbol{N}_{cx}\boldsymbol{D}_2^D & \boldsymbol{N}_{cz}\boldsymbol{D}_2^D & 0 \\
\boldsymbol{T}_{2y}\boldsymbol{Q}_1^D & \boldsymbol{T}_{2x}\boldsymbol{Q}_1^D & \boldsymbol{T}_{2z}\boldsymbol{Q}_1^D & \boldsymbol{A}_{t2} \\
0 & 0 & 0 & \boldsymbol{I}_c
\end{pmatrix}
$$



(a) (b)

The nonzero distribution of: (a) the coefficient matrix with matrix reorganization, (b) the preconditioning matrix.

$$\boldsymbol{Q}_1^D = \left(\boldsymbol{P}_0\boldsymbol{D}_2^D - \boldsymbol{D}_0\boldsymbol{D}_1^D\right)^D$$

$$\boldsymbol{Q}_{2\alpha} = \boldsymbol{C}_{d\alpha}\boldsymbol{D}_2^D + \boldsymbol{C}_{\alpha}\boldsymbol{D}_1^D, \ \alpha = x, y, z$$

(4) *Preconditioning (cont.)*

- The comparison among three different examples.
  - The preconditioning costs little time for construction;
  - Be robust and efficient for different examples.

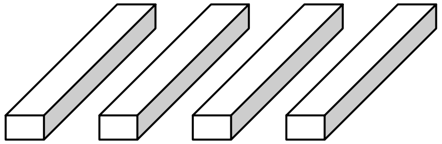
The Preconditioner Construction Time and Iteration Number In MBEM And FastImp For Different Structures

	Number of Unknows		Construction (Second)		LU Factorization (Second)		Number of Iteration	
	FastImp	MBEM	FastImp	MBEM	FastImp	MBEM	FastImp	MBEM
Wire	1136	650	0.04	0.19	0.01	0.07	9	**8**
Bus1x1	2272	1300	0.08	0.74	0.01	0.42	18	**8.5**
Bus2x2	3536	2024	0.16	1.22	0.02	1.24	21	**9**

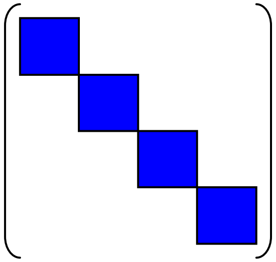
(5) *Integral Reuse Technique*

- The integral computations in D_1 and D_2 are very time-consuming, so that integral reuse technique was proposed.

- Method:
  - Matrix D_1 and D_2 are block diagonal matrices.
  - No coupling between variables belonging to different conductors.
  - The integral results can be reused for the same conductors.



(a)



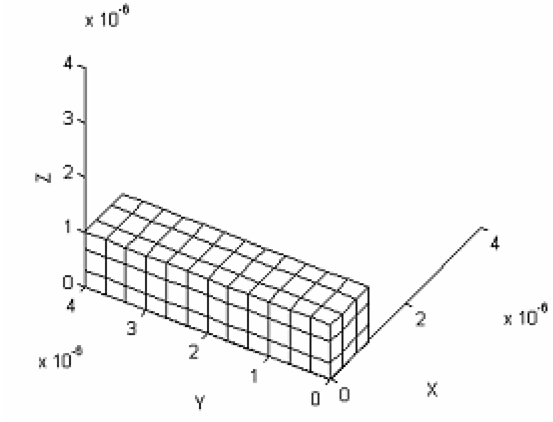
(b)

(a) A problem with four conductors, and (b) the corresponding population of matrix $\boldsymbol{D}_1$ or $\boldsymbol{D}_2$.

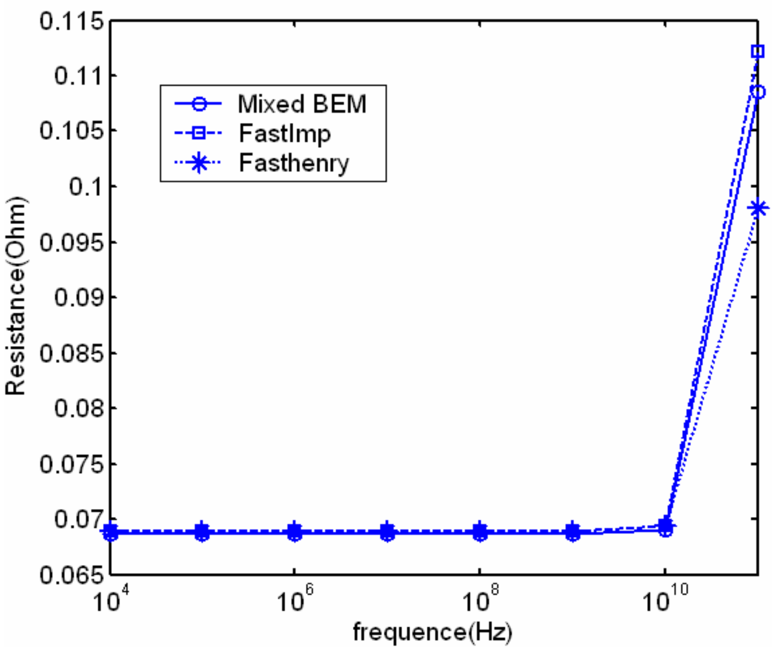
Outline

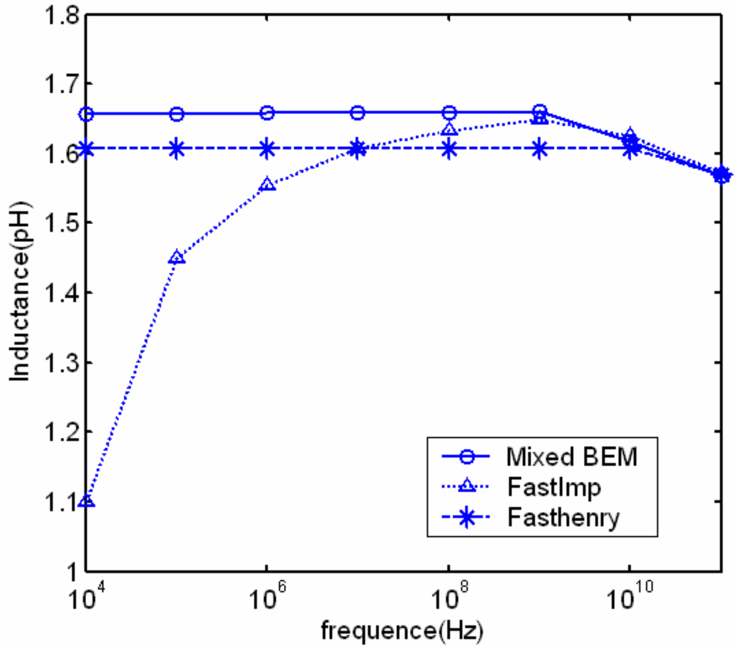
- **Introduction**
- **Background**
- **Extraction with Mixed BEM**
- **Numerical examples**
- **Conclusion**

Accuracy Comparison

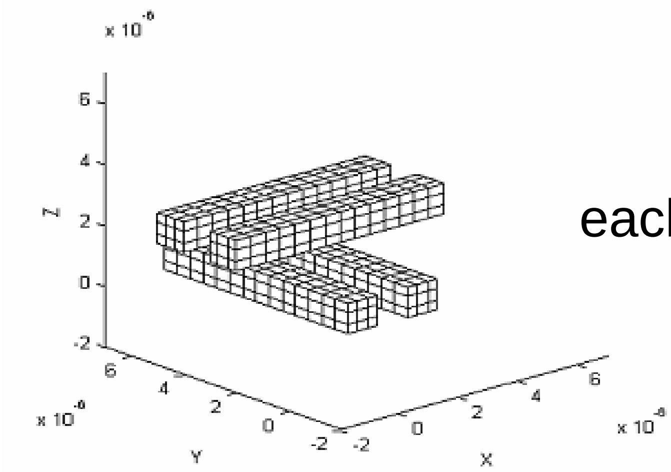


Example 1: a single wire example
size: 1x1x4 (um)
partitioned level: 3x3x12

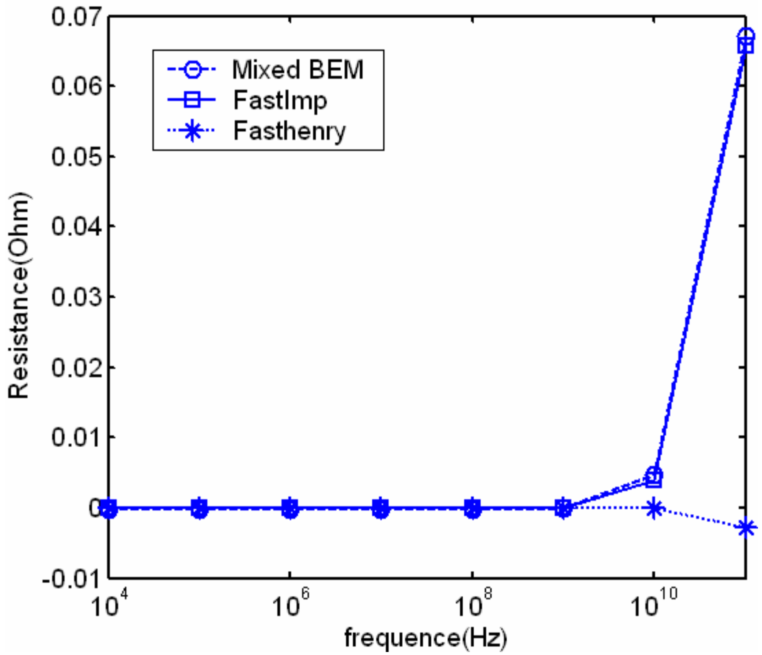


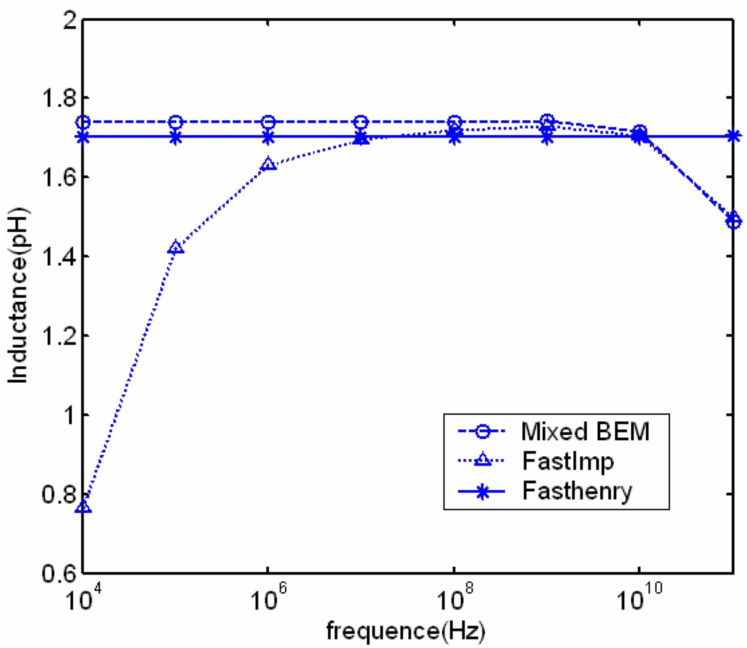


Accuracy Comparison



Example 2: a cross-over structure
each conductor size: 1x1x7 (um)
partitioned level: 3x3x14



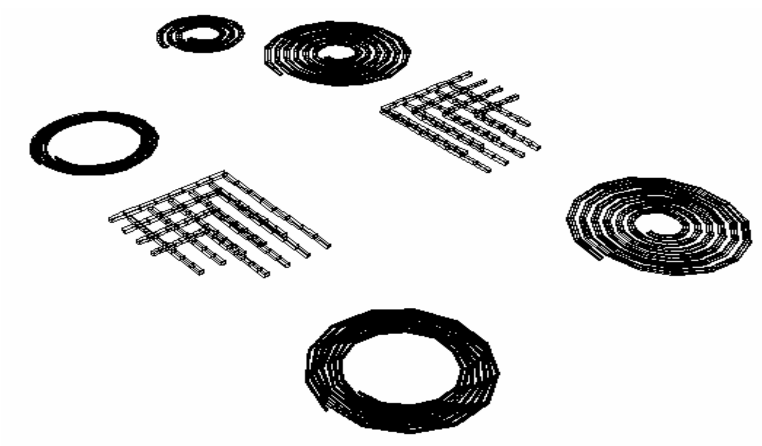


Speed Comparison

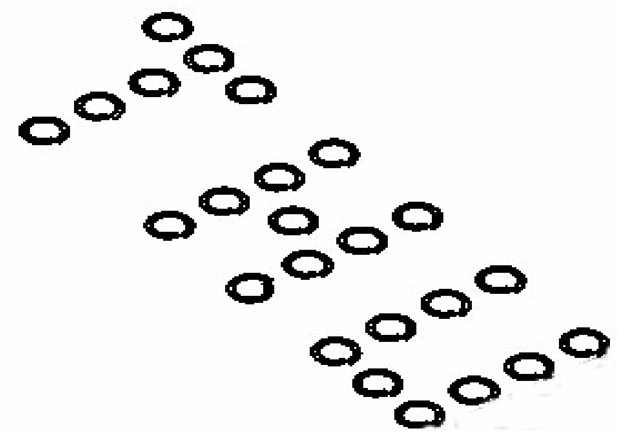
- Two typical structures
  - Example 1: Part of RF circuit
  - Example 2: THU logo

Discretization of Two Large 3-D Structures

		RF circuit	THU logo
Number of Panel		**6,364**	**17,472**
Number of Vertex		**6,426**	**17,520**
variables	FastImp	**44,610**	**122,352**
	MBEM	**25,518**	**69,936**



(a) Part of RF circuit



(b) THU logo

Speed Comparison

- Time breakdown for the two large examples (Unit is Second)

	RF circuit		THU logo	
	FastImp	MBEM	FastImp	MBEM
Form PFFT Matrices	335.0	263.2	811.9	106.2
Form Post-Process Matrix	--	32.5	--	56.3
Form Preconditioner $\boldsymbol{P}_r$	2.0	4.2	7.8	13.0
LU Factorization	0.16	18.6	0.5	60.0
Matrix-Vector Product ($\boldsymbol{Ax}$)	7.0	6.9	18.4	14.9
GMRES (tol =1e-3)	7558 (29 iter)	3011(10 iter)	7265(15 iter)	4280(8 iter)
Total time	7893 (2.19 hour)	3447 (0.96hour)	7919 (2.2hour)	4594 (1.27hour)

The speedup of two examples are 56% and 42%, respectively. (Approximate 2X)

- Memory comparison (Unit is MB)

	RF circuit	THU logo
FastImp	173	474
MBEM	192	501

Outline

- ❖ **Introduction**
- ❖ **Background**
- ❖ **Extraction with Mixed BEM**
- ❖ **Numerical examples**
- ❖ **Conclusion**

Conclusion

- Efficient algorithm for 3-D Impedance Extraction
  - Less accuracy across the whole frequency range, too much calculation consumption, are shortcomings of most existing methods
  - In this paper, many efficient techniques are proposed, and we combined Mixed BEM with pFFT to accelerate the extraction.
  - The algorithm eliminates the low-frequency accuracy problem in FastImp.
  - Achieve more than 2x speedup ratio over FastImp (M.I.T. 2005)
- Future work
  - Further improvement of our extractor.
  - To perform EMQS and Full-wave analysis

Thank you !

For more information:

gongf05@mails.tsinghua.edu.cn