Direct Finite-Element-Based Solver for 3D-IC Thermal Analysis via $\mathcal{H}$-Matrix Representation

Ying-Chi Li[1], Sheldon. X.-D. Tan[2], Tan Yu[2], Xin Huang[2] and Ngai Wong[1]
[1]Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong
Email: tycli@eee.hku.hk, nwong@eee.hku.hk
[2]Department of Electrical Engineering, The University of California, Riverside, USA
Email: stan@ee.ucr.edu, tyu008@ucr.edu, xhuan009@ucr.edu

***Abstract*—We propose, for the first time, the use of hierarchical matrix ($\mathcal{H}$-matrix) in the efficient finite-element-based (FE-based) direct solver implementation for both steady and transient thermal analyses of three-dimensional integrated circuits (3D ICs). $\mathcal{H}$-matrix was shown to provide a data-sparse way to approximate the matrices and their inverses with almost linear space and time complexities. We show this is also true for FE-based transient analysis of thermal parabolic partial differential equations (PDEs). Specifically, we show that the stiffness matrix from a FE-based steady and transient thermal analysis can be represented by $\mathcal{H}$-matrix without any approximation, and its inverse and Cholesky factors can be evaluated by $\mathcal{H}$-matrix with controlled accuracy. We then show that the memory and time complexities of the solver are bounded by $\mathcal{O}(k_1 N \log N)$ and $\mathcal{O}(k_1^2 N \log^2 N)$, respectively, for very large scale thermal systems, where k is a small quantity determined by accuracy requirements and N is the number of unknowns in the system. Numerical results validate and demonstrate the effectiveness of the proposed method in terms of predicted theoretical scalability.**

***Index Terms*—Finite element method, 3D IC, $\mathcal{H}$-matrix, thermal analysis.**

I. Introduction

3D IC technology is viewed as a necessary driving force to maintain the trend described by Moore's law [1], [2]. 3D stacked integration based on TSV connection technique enables heterogeneous integration of cores, memories, and analog devices, and overcomes the barriers in interconnect scaling. However, it also introduces new challenges for TSV-based 3D ICs. Thermal issues are one of the primary challenges in 3D ICs. As a result, fast and efficient thermal analysis and estimation at different granularity levels become critical for thermal-aware design and optimization.

Traditional thermal analysis solves the partial thermal diffusion equations using numerical approaches such as the finite difference method (FDM) [3] and the FEM [4]. The FE discretization of thermal initial-value problems typically leads to a large sparse FE system, which can be solved by iterative and direct methods. The iterative methods are usually efficient. However, the convergence of the iteration highly depends on an appropriate preconditioner which is often problem-dependent. For LU-based direct method, it typically is more expensive for both memory and computation with super-linear time complexity [5].

This work is supported by NSF grant No.CCF-1255899, in part by Semiconductor Research Corporation (SRC) Grant under No. 2013-TJ-2417 and UC Academic Senate Committee on Research (COR) Award.

There exists a new representation of matrices, the so-called $\mathcal{H}$-matrix [6], [7], [8]. $\mathcal{H}$-matrix was shown to be well-suited for data-sparse representation of dense or sparse matrices arising in FEM or for the approximation of the inverse to the FE matrices for elliptic partial differential equations (PDEs). These matrices are not necessarily sparse, but they are data-sparse in the sense that these matrices are described by only few data. The significance of this $\mathcal{H}$-matrix technique is that those matrices can be stored by almost linear space complexities. The resulting matrix operations such as addition, multiplication, and even inverse can be done in terms of space complexities, which is almost linear. Recently, the $\mathcal{H}$-matrix structure has been used in solving electro- and magneto-static problems in which the resulting wave PDEs are hyperbolic [9], [10], [11], [12].

In this paper, the $\mathcal{H}$-matrix technique is applied to solve both steady and transient thermal analysis problems using FE-based approaches. For heat diffusion PDEs, the steady state PDEs are elliptic while the transient PDEs are parabolic [3]. Not many works have been reported for $\mathcal{H}$-matrix FEM solvers for parabolic PDEs so far as most of works focused on either elliptic or hyperbolic PDEs. Contributions of this chapter are:

- It is demonstrated that $\mathcal{H}$-matrix techniques can be applied to parabolic PDEs such as transient thermal equations. We show that the stiffness matrix generated from a steady and transient thermal analysis based on FEM can be represented by $\mathcal{H}$-matrix without approximation.
- It is revealed that the inverse and Cholesky factors of the thermal stiffness matrices generated from the parabolic PDEs by the FEM can be also approximated and represented representation using the $\mathcal{H}$-matrix arithmetics.
- It is shown that the memory and the time complexities of the solver are bounded by $\mathcal{O}(k_1 N \log N)$ and $\mathcal{O}(k_1^2 N \log^2 N)$, respectively, for very large scale thermal systems, where k_1 is a small quantity determined by the accuracy requirements and N is the number of unknowns in the system.

To the best knowledge of the authors, this is the first approach to investigating the $\mathcal{H}$-matrix based solver for parabolic thermal PDEs of complicated 3D IC structures. Numerical results on one 3D IC structure have validated and demonstrated the effectiveness of the proposed method in terms of predicted theoretical scalability.

The paper is organized as follows: Section II presents the basics for FE-based thermal analysis. Section III shows and proves that thermal stiffness matrices from FEM can be

represented by $\mathcal{H}$ matrices without any approximation. Then, Section IV shows the inverse of thermal stiffness matrices can be represented by $\mathcal{H}$-matrices with controlled accuracy as well as the arithmetics to compute its inverse and its Cholesky factors. Section VI presents some numerical results and Section VII concludes this paper.

II. Finite-element-based thermal analysis

In the circuit, package and board levels, the heat transfer phenomena are governed by the following heat differential equation [13], [14]:

$$\rho C_p \frac{\partial T(\vec{r},t)}{\partial t} = \nabla \cdot [\kappa(\vec{r},T) \cdot \nabla T(\vec{r},t)] + g(\vec{r},t), \quad (1)$$

subject to the following general thermal boundary condition (Robin's boundary condition)

$$\kappa(\vec{r},T) \frac{\partial T(\vec{r},t)}{\partial \mathbf{n}_i} = h_i(T(\vec{r},t) - T_a). \quad (2)$$

In (1), T is the temperature (K), ρ is the density of the material (kg/m^3), C_p is the mass heat capacity (Jkg^{-1}K^{-1}), κ is the thermal conductivity (Wm^{-1}K^{-1}), and g is the heat energy generation rate (W/m^3). In (2), $\mathbf{n}_i$ is the outward direction normal to the boundary condition i, h_i is the heat-transfer coefficient (Wm^{-2}K^{-1}) (for the convective interface), and T_a is the ambient temperature surrounding the thermal systems. If $h_i = 0$, the boundary condition is adiabatic (isolated) , otherwise, it is convective. Note that the thermal conductivity κ differs for different materials and also depends on the temperature. In our work, we assume that κ is constant for each material. Then, (1) can be written as

$$\kappa \left[\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right] + g = \rho C_p \frac{\partial T}{\partial t}, \quad (3)$$

subject to the following boundary conditions:

$$T = T_b \text{ on } S_1, \quad (4)$$

$$\kappa \left[\frac{\partial T}{\partial x}\mathbf{n_1} + \frac{\partial T}{\partial y}\mathbf{n_2} + \frac{\partial T}{\partial z}\mathbf{n_3} \right] + q = 0 \text{ on } S_2, \quad (5)$$

$$\kappa \left[\frac{\partial T}{\partial x}\mathbf{n_1} + \frac{\partial T}{\partial y}\mathbf{n_2} + \frac{\partial T}{\partial z}\mathbf{n_3} \right] + h\left(T - T_a\right) = 0 \text{ on } S_3, \quad (6)$$

and initial condition $T = T_0$ at $t = 0$, where $\mathbf{n_1}$, $\mathbf{n_2}$ and $\mathbf{n_3}$ are surface normals, q is the heat flux and h is the heat transfer coefficient.

To solve the PDE from the thermal diffusion equations (3), the FDM and FEM are among the popular methods. In this work, we focus on the FE-based thermal analysis. FEM discretizes the solution region Ω into small elements Ω_e such as triangles for 2D problems and tetrahedrons for 3D problems by a specific set of shape function φ_i such that:

$$\begin{aligned} T\left(x,y,z,t\right) &= \sum_{i=1}^{m} \varphi_i\left(x,y,z\right) T_i\left(t\right) \\ &= \left[\mathbf{\Phi}\right]\left[\mathbf{T}\right], \end{aligned} \quad (7)$$

where m is the number of nodes on the element (e.g. 4 for a tetrahedron) $\mathbf{\Phi} = [\varphi_1\ \varphi_2\ \cdots\ \varphi_m]$ is the shape function matrix and $\mathbf{T} = [T_1\ T_2\ \cdots\ T_m]^{\mathrm{T}}$ is the vector of unknown temperature at corresponding nodes.

In this paper, tetrahedron elements are used, thus

$$\varphi_i = \frac{1}{6V}\left(c_{i1} + c_{i2}x + c_{i3}y + c_{i4}z\right), \quad (8)$$

where

$$6V = \det \begin{bmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4, \end{bmatrix} \quad (9)$$

and c_{ij} is the (i,j) cofactor of the matrix in (9) with i and $j = 1,2,3,4$.

Assume the conductivity κ is independent of temperature and the heat flow due to radiation is negligible. The Galerkin representation of (3) for a element is

$$\int_{\Omega_e} \varphi_i \left[\kappa \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) + g - \rho C_p \frac{\partial T}{\partial t} \right] d\Omega_e = 0. \quad (10)$$

By Green's theorem and integration by parts, the second derivatives can be rewritten in two parts:

$$\begin{aligned} &\int_{\Omega_e} \varphi_i \left(k \frac{\partial^2 T}{\partial x^2} \right) d\Omega_e \\ = &\int_{S_e} \varphi_i \left(k \frac{\partial T}{\partial x} \right) \mathbf{n_1} dS_e - \int_{\Omega_e} k \frac{\partial \varphi_i}{\partial x} \frac{\partial T}{\partial x} d\Omega_e \end{aligned} \quad (11)$$

Substituting (11) and boundary condition equations (5) and (6) into (10), we get

$$\begin{aligned} &- \int_{\Omega_e} k \left[\frac{\partial \varphi_i}{\partial x}\frac{\partial T}{\partial x} + \frac{\partial \varphi_i}{\partial y}\frac{\partial T}{\partial y} + \frac{\partial \varphi_i}{\partial z}\frac{\partial T}{\partial z} \right] d\Omega_e \\ &+ \int_{\Omega_e} \left[\varphi_i g - \varphi_i \rho C_p \frac{\partial T}{\partial t} \right] d\Omega_e - \int_{S_{2e}} \varphi_i q dS_{2e} \\ &- \int_{S_{3e}} \varphi_i h \left(T - T_a\right) dS_{3e} = 0. \end{aligned} \quad (12)$$

Rearrange (12) and substitute (7) into it, we have

$$\mathbf{C_e} \frac{\partial \mathbf{T}}{\partial t} + \mathbf{G_e}\mathbf{T} = \mathbf{f_e}, \quad (13)$$

where $\mathbf{f_e} \subset \mathbb{R}^{4\times 1}$ is the local thermal load, $\mathbf{C_e} \subset \mathbb{R}^{4\times 4}$ is the local heat capacity matrix and $\mathbf{G_e} \subset \mathbb{R}^{4\times 4}$ is the local stiffness matrix, where

$$\mathbf{C_e} = \int_{\Omega_e} \rho C_p \mathbf{\Phi}^{\mathrm{T}} \mathbf{\Phi} d\Omega_e, \quad (14)$$

$$\mathbf{G_e} = \int_{\Omega_e} \kappa \mathbf{B}^{\mathrm{T}} \mathbf{B} d\Omega_e, + \int_{S_{3e}} h \mathbf{\Phi}^{\mathrm{T}} \mathbf{\Phi} dS_{3e}, \quad (15)$$

$$\mathbf{f_e} = \int_{\Omega_e} g \mathbf{\Phi}^{\mathrm{T}} d\Omega_e - \int_{S_{2e}} q \mathbf{\Phi}^{\mathrm{T}} dS_{2e} + \int_{S_{3e}} h T_a \mathbf{\Phi}^{\mathrm{T}} dS_{3e}, \quad (16)$$

and

$$\mathbf{B} = \begin{bmatrix} \frac{\partial \varphi_1}{\partial x} & \frac{\partial \varphi_2}{\partial x} & \frac{\partial \varphi_3}{\partial x} & \frac{\partial \varphi_4}{\partial x} \\ \frac{\partial \varphi_1}{\partial y} & \frac{\partial \varphi_2}{\partial y} & \frac{\partial \varphi_3}{\partial y} & \frac{\partial \varphi_4}{\partial y} \\ \frac{\partial \varphi_1}{\partial z} & \frac{\partial \varphi_2}{\partial z} & \frac{\partial \varphi_3}{\partial z} & \frac{\partial \varphi_4}{\partial z} \end{bmatrix} = \frac{1}{6V} \begin{bmatrix} c_{12} & c_{22} & c_{32} & c_{42} \\ c_{13} & c_{23} & c_{33} & c_{43} \\ c_{14} & c_{24} & c_{34} & c_{44} \end{bmatrix} \quad (17)$$

is the gradient matrix of the shape functions.

The global thermal load $\mathbf{f}$, heat capacity matrix $\mathbf{C}$ and stiffness matrix $\mathbf{G}$ can be assembled by adding the values on corresponding nodes, hence we can get the representation of entire domain with the following equation:

$$\mathbf{C}\frac{\partial \mathbf{T}}{\partial t} + \mathbf{G}\mathbf{T} = \mathbf{f}. \tag{18}$$

In steady stage analysis, there is no time-dependent term ($\frac{\partial \mathbf{T}}{\partial t} = 0$). Therefore (18) can be reduced to:

$$\mathbf{G}\mathbf{T} = \mathbf{f}. \tag{19}$$

In transient analysis, FDM is applied to evaluate the result. Using a backward difference method, the differential operator in (18) can be approximated as follow:

$$\frac{\partial \mathbf{T}}{\partial t} \approx \frac{\mathbf{T}^{n+1} - \mathbf{T}^{n}}{\Delta t}. \tag{20}$$

Substituting (20) into (18), we have

$$\left[\mathbf{G} + \frac{\mathbf{C}}{\Delta t}\right]\mathbf{T}^{n+1} = \mathbf{f}^{n+1} + \frac{\mathbf{C}}{\Delta t}\mathbf{T}^{n}. \tag{21}$$

III. $\mathcal{H}$-MATRIX REPRESENTATION FOR THE THERMAL STIFFNESS MATRIX

A. $\mathcal{H}$-matrix in a nutshell

An $\mathcal{H}$-matrix $\mathbf{H} \in \mathbb{C}^{\mathcal{I}\times\mathcal{I}}$ is generally associated with an admissibility condition (Section 2.2.2 in [8]), where $\mathcal{I} = \{i_1, i_2, \cdots, i_N\}$ is the re-ordered index set containing the indices of the basis functions φ_i or nodes in global domain Ω with the total number of N unknowns in the entire FEM model. If any two subsets t and $s \subseteq \mathcal{I}$ are admissible, the following admissibility condition holds:

$$\min\{\operatorname{diam}(\Omega_t), \operatorname{diam}(\Omega_s)\} \leq \eta \operatorname{dist}(\Omega_t, \Omega_s), \tag{22}$$

where Ω_t and $\Omega_s \subseteq \Omega$ represented by axis-parallel box (Section 2.2.3 in [8]), contain the supports of all the basis functions subject to t and s , diam$(\cdot)$ is the Euclidean diameter of the subsets, dist$(\cdot,\cdot)$ is the Euclidean distance of two subsets, and η is a positive constant. In addition, the sub-matrix $\mathbf{H}_{t\times s}$ can be represented by a low-rank matrix with rank k and stored in a factorized form such that

$$\mathbf{H}_{t\times s} \approx \mathbf{A}\mathbf{B}^{\mathrm{T}}, \tag{23}$$

where $\mathbf{A}$ is a $t \times k$ matrix and $\mathbf{B}$ is a $s \times k$ matrix. The other sub-matrices in $\mathbf{H}$ corresponding to the inadmissible subsets in $\mathcal{I}$ will be represented by full matrices. In order to construct the suitable structure of $\mathbf{H}$, a block cluster tree $\mathcal{T}_{\mathcal{I}\times\mathcal{I}}$ is necessary based on the geometric information of the FEM model. It is built according to the definition in Section 1.5 of [8].

In order to build $\mathcal{T}_{\mathcal{I}\times\mathcal{I}}$, a cluster tree $\mathcal{T}_{\mathcal{I}}$ for multi-dimensional meshes is constructed first. We find a bounding box that contains the full index of the basis functions then store into $\mathcal{I}$, split the box into two subsets based on the geometrical position of nodes, and repeat the splitting until each subset has no more than n_{min} (leafsize) indices which is a tree depth control parameter. $\mathcal{T}_{\mathcal{I}}$ is built in which each subset is a cluster and $\mathcal{I}$ is the root of the tree.

The block cluster tree $T_{\mathcal{I}\times\mathcal{I}}$ is constructed from two cluster trees $T_{\mathcal{I}}$ and $T_{\mathcal{I}}$ in the thermal FEM analysis. The admissibility condition of each pair of clusters$(t, s) \in T_{\mathcal{I}}$ is checked

Fig. 1. One-dimensional FEM mesh.

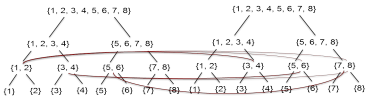


Fig. 2. An example for the block cluster tree.

level by level, which starts from the root and descends in the tree. If the cluster pair $(t, s) \in T_{\mathcal{I}}$ is admissible, then make it a leaf of $T_{\mathcal{I}\times\mathcal{I}}$,which corresponds to a disjoint partition of the index set $\mathcal{I} \times \mathcal{I}$. Otherwise repeat the procedure recursively for all pairs of its children.

Take one-dimensional structure as an example, the node index set $\mathcal{I} = \{1, \cdots, 8\}$ with corresponding domain $[0, 1]$ is shown in Fig. 1 and Fig. 2. The root of the block cluster tree is $\mathcal{I} \times \mathcal{I}$. No admissible clusters are found in level 0 and level 1. The clusters that are admissible in level 3 are linked and stored as leaves of the block cluster tree. The construction of block cluster tree defines the $\mathcal{H}$-matrix structure as shown in Fig. 3, in which shaded matrix blocks are represented by $\mathbf{A}\mathbf{B}^{\mathrm{T}}$ while others are represented by full matrices.

B. $\mathcal{H}$ matrix representation for the thermal matrices

We first show that the thermal stiffness matrices for both steady and transient FEM analyses can be converted to $\mathcal{H}$-matrices without any accuracy loss.

In the thermal FEM model, $\mathbf{G}$ and $\mathbf{C} \in \mathbb{R}^{\hat{\mathcal{I}}\times\hat{\mathcal{I}}}$ can be obtained from (21), where $\hat{\mathcal{I}} = \{1, 2, \cdots, N\}$ is the node indices set. If the indices of nodes belong to subset $t, s \subseteq \hat{\mathcal{I}}$ and are admissible, they must be physically disconnected (their corresponding elements are not connected directly). Therefore, the sub-matrices $\mathbf{G}_{t\times s}$ and $\mathbf{C}_{t\times s}$ will be all zeros. After applying the block cluster tree construction algorithm, $\hat{\mathcal{I}}$ is re-ordered into $\mathcal{I}$, the global stiffness matrix $\mathbf{G}$ and $\left[\mathbf{G} + \frac{\mathbf{C}}{\Delta t}\right]$ can be then stored into $\mathbf{H}_{\text{steady}}$ and $\mathbf{H}_{\text{transient}}$, respectively, without any approximation.

IV. $\mathcal{H}$-MATRIX BASED INVERSE AND CHOLESKY FACTORS OF FEM SYSTEM AND THEIR ACCURACY CONTROL

Now we look at the problem of $\mathcal{H}$-matrix presentation of the inverse and Cholesky factors of the thermal stiffness matrices in both steady state and transient FEM analyses.

Note that heat equation for steady state analysis is an elliptic PDE (3), the inverse of $\mathbf{H}_{\text{steady}}$ is also a $\mathcal{H}$-matrix [15]. But heat equation for transient analysis is a parabolic PDE (3) [3],

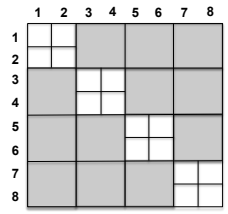


Fig. 3. $\mathcal{H}$-matrix structure of 1D example

and not many studies have been reported for $\mathcal{H}$-matrix representation of FEM matrices from parabolic PDE so far. On the other hand, we observe that the structure of $\left[\mathbf{G}+\frac{\mathbf{C}}{\Delta t}\right]$ is identical to that of $\mathbf{G}$ at each time interval by comparing (14) and the second term of (15). Therefore, $\mathbf{H}_{\text{transient}}$ and $\mathbf{H}_{\text{steady}}$ have the same structure and the inverse of $\mathbf{H}_{\text{transient}}$ is also a $\mathcal{H}$-matrix. The detailed theoretical proof of this is quite involved and is omitted here due to space limitation. But the both space and time complexities for the resulting thermal stiffness matrices will be confirmed in numerical result section. Therefore, $\mathcal{H}$-matrix based inverse and Cholesky factorization (stiffness matrix from (21) is symmetric) can also be applied, which is also called $\mathcal{H}$-arithmetics (Section 6.2 in [8]).

During the inversion procedures, the $\mathcal{H}$-matrix format in $\mathbf{H}$ is preserved such that both $\mathbf{H}$ and $\mathbf{H}^{-1}$ have the same structure. Fig. 4 shows the structure of $\mathbf{H}$. For the Cholesky factorization, the $\mathcal{H}$-matrix format is cut into half along the diagonal, Fig. 5 shows the matrix structure of the resultant Cholesky factors. The results are stored in the form of $\mathbf{AB}^{\mathrm{T}}$ in (23) for admissible blocks in both inverse and Cholesky factors.

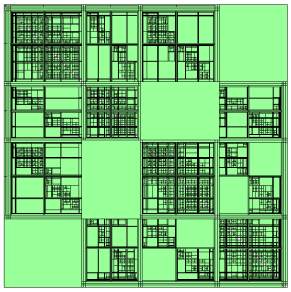


Fig. 4. $\mathcal{H}$-matrix structure of $\mathbf{H}$ and $\mathbf{H}^{-1}$

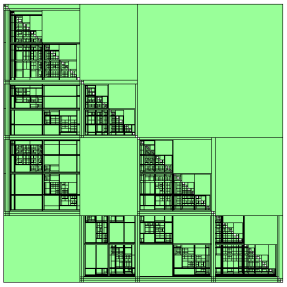


Fig. 5. $\mathcal{H}$-matrix structure of Cholesky factors

A. Direct inverse of $\mathcal{H}$-matrix

Consider the $\mathcal{H}$-matrix format of the stiffness matrix $\mathbf{H}$, it can be written as:

$$\mathbf{H}=\begin{bmatrix}\mathbf{H}_{11} & \mathbf{H}_{12}\\ \mathbf{H}_{21} & \mathbf{H}_{22}\end{bmatrix}. \tag{24}$$

The inverse if $\mathbf{H}$ can be computed by use of the equation:

$$\mathbf{H}^{-1}=\begin{bmatrix}\mathbf{H}_{11}^{-1}+\mathbf{H}_{11}^{-1}\mathbf{H}_{12}\mathbf{S}^{-1}\mathbf{H}_{21}\mathbf{H}_{11}^{-1} & -\mathbf{H}_{11}\mathbf{H}_{12}\mathbf{S}^{-1}\\ \mathbf{S}^{-1}\mathbf{H}_{21}\mathbf{H}_{11}^{-1} & \mathbf{S}^{-1}\end{bmatrix}, \tag{25}$$

where $\mathbf{S}=\mathbf{H}_{22}-\mathbf{H}_{21}\mathbf{H}_{11}^{-1}\mathbf{H}_{12}$. This process only requires computing the inverse of $\mathbf{H}_{11}$ and $\mathbf{S}$, which is much faster than computing the whole inverse of $\mathbf{H}$.

B. Cholesky factorization of $\mathcal{H}$-matrix

In most of the FEM application, computing the whole inverse of $\mathbf{H}$ is unnecessary. Instead, only a method to solve $\mathbf{Hx}=\mathbf{b}$ is required. Therefore, Cholesky or LU decomposition is applied. In our the thermal FEM analysis, the stiffness matrix is symmetric and positive semidefinite, so Cholesky decomposition is chosen. Assume $\mathbf{H}$ can be factorized into $\mathbf{LL}^{\mathrm{T}}$, then

$$\begin{aligned}\mathbf{H} &= \mathbf{LL}^{\mathrm{T}}=\begin{bmatrix}\mathbf{L}_{11} & \mathbf{0}\\ \mathbf{L}_{21} & \mathbf{L}_{22}\end{bmatrix}\begin{bmatrix}\mathbf{L}_{11}^{\mathrm{T}} & \mathbf{L}_{21}^{\mathrm{T}}\\ \mathbf{0} & \mathbf{L}_{22}^{\mathrm{T}}\end{bmatrix}\\ &= \begin{bmatrix}\mathbf{L}_{11}\mathbf{L}_{11}^{\mathrm{T}} & \mathbf{L}_{11}\mathbf{L}_{21}^{\mathrm{T}}\\ \mathbf{L}_{21}\mathbf{L}_{11}^{\mathrm{T}} & \mathbf{L}_{22}\mathbf{L}_{22}^{\mathrm{T}}\end{bmatrix}.\end{aligned} \tag{26}$$

Comparing (26) with (24), the Cholesky factorization can be recursively computed as:

1) Compute $\mathbf{L}_{11}$ by $\mathcal{H}$-Cholesky factorization $\mathbf{H}_{11}=\mathbf{L}_{11}\mathbf{L}_{11}^{\mathrm{T}}$;
2) Compute $\mathbf{L}_{21}$ by solving $\mathbf{H}_{21}=\mathbf{L}_{21}\mathbf{L}_{11}^{\mathrm{T}}$;
3) Compute $\mathbf{L}_{22}$ by $\mathcal{H}$-Cholesky factorization $\mathbf{H}_{22}=\mathbf{L}_{22}\mathbf{L}_{22}^{\mathrm{T}}$.

After obtaining the factor $\mathbf{L}$, the FEM system can be solved in two steps:

1) Compute $\mathbf{LX}=\mathbf{f}$ by $\mathcal{H}$-arithmetics;
2) Compute $\mathbf{L}^{\mathrm{T}}\mathbf{T}=\mathbf{X}$ by $\mathcal{H}$-arithmetics.

The $\mathcal{H}$-arithmetic for solving the above equations follows:

$$\mathbf{LX}=\begin{bmatrix}\mathbf{L}_{11} & \mathbf{0}\\ \mathbf{L}_{21} & \mathbf{L}_{22}\end{bmatrix}\begin{bmatrix}\mathbf{X}_1\\ \mathbf{X}_2\end{bmatrix}=\begin{bmatrix}\mathbf{f}_1\\ \mathbf{f}_2\end{bmatrix}, \tag{27}$$

solve $\mathbf{X}_1$ from $\mathbf{L}_{11}\mathbf{X}_1=\mathbf{f}_1$, then solve $\mathbf{X}_2$ from $\mathbf{L}_{22}\mathbf{X}_2=\mathbf{f}_2-\mathbf{L}_{21}\mathbf{X}_1$.

C. Accuracy Control

Since the admissible blocks in $\mathbf{H}$ are stored in the form of low-rank matrices as in (23), the accuracy of inversion and Cholesky factorization depend on the choice of rank k. However, there are many admissible blocks with different sizes. If a constant rank k is used, the accuracy in large admissible block may not be guaranteed for a small k or the efficiency of the computation may be lower for a large k. Therefore, adaptive rank scheme in section 6.1 of [8] should be used instead of a constant rank. Assume $\mathbf{H}_{t\times s}$ is in admissible block $\mathcal{T}_{t\times s}$, then it can be represented in the form of (23):

$$\mathbf{H}_{t\times s}\approx\mathbf{AB}^{\mathrm{T}}=\hat{\mathbf{H}}_{t\times s} \tag{28}$$

By introducing the relative truncation error ε, the rank k for $\mathbf{H}_{t\times s}$ is determined by:

$$k:=\min\left\{\hat{k}\in\mathbb{N}_0 \mid \exists\hat{\mathbf{H}}_{t\times s}\in\mathcal{R}\left(\hat{k},t,s\right)\right\} \tag{29}$$

subject to

$$\left\|\mathbf{H}_{t\times s}-\hat{\mathbf{H}}_{t\times s}\right\|\leq\varepsilon\left\|\mathbf{H}_{t\times s}\right\|. \tag{30}$$

where

$$\mathcal{R}\left(k,s,t\right):=\left\{\mathbf{M}\in\mathbb{R}^{s\times t} \mid \operatorname{rank}\left(\mathbf{M}\right)\leq k\right\}. \tag{31}$$

Therefore, the accuracy of the results can be controlled by choosing an appropriate ε. The corresponding vectors can be found by singular value decomposition with $\mathcal{H}$-arithmetic, which has a linear complexity of $\mathcal{O}\left(k^2\max\left(|s|,|t|\right)\right)$ [8].

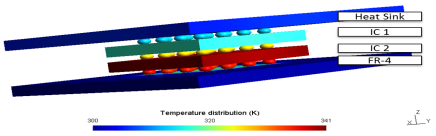


Fig. 6. Simple 3D IC with TSV.

V. Complexity Analysis

Complexity analysis for general $\mathcal{H}$-arithmetics [8] and for FE-based electromagnetic analysis [16] are well developed. Since FE-based thermal analysis is similar to those in electromagnetic, the complexity analysis of inverse in [16] can be adopted in this paper. During the analysis, a important parameter, sparsity constant C_{sp} is introduced, which is define as follow:

$$C_{sp} := \max \left\{ \begin{array}{l} \max\limits_{t \in T_{\mathcal{I}}} \#\{s \in T_{\mathcal{I}} \mid s \times t \in T_{\mathcal{I}\times\mathcal{I}}\}, \\ \max\limits_{s \in T_{\mathcal{I}}} \#\{t \in T_{\mathcal{I}} \mid s \times t \in T_{\mathcal{I}\times\mathcal{I}}\} \end{array} \right\} \tag{32}$$

For any $\mathcal{H}$-matrix $\mathbf{H}$ built from the block cluster tree $T_{\mathcal{I}\times\mathcal{I}}$ with tree depth $p = \log_N$, minimum leave size n_{min} and rank k, its storage complexity $H_{St}(T, k)$ is bounded by

$$\begin{aligned} H_{St}(T,k) &\leq 2C_{sp}(p+1)\max(k, n_{min})N \\ &\sim \mathcal{O}(k_1 N \log_2 N), \end{aligned} \tag{33}$$

where $k_1 = \max(k, n_{min})$ [8]. The complexity of computing inverse is the sum of the complexity of $\mathcal{H}$-based multiplication and addition, while Cholesky factorization only depends on the complexity of multiplication. Both $\mathcal{H}$-based multiplication and addition for the entire cluster tree are bounded by the complexity of $\mathcal{O}\left(k_1^2 N \log_2^2 N\right)$ [16]. Therefore, the complexity of computing inverse and Cholesky factors are also bounded by the complexity of $\mathcal{O}\left(k_1^2 N \log_2^2 N\right)$. After Cholesky factorization, solving (21) for each time interval only has a complexity of $\mathcal{O}(k_1 N \log_2 N)$.

VI. Numerical Results and Discussions

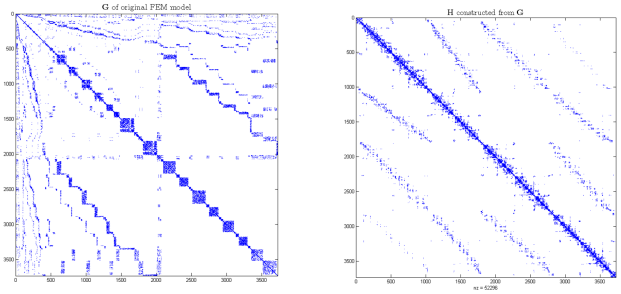
The proposed $\mathcal{H}$-based finite element direct solver is prototyped in C based on the hierarchical matrices library [17]. The temperature distribution of the TSV is shown in Fig. 6. The experiments are carried out on Window's server with Intel Core i7 CPU and 32GB memory.

The meshes are generated by the Gmsh program for our finite element solver [18]. We create a 3D-IC structure, which consists of two active layers with TSVs and micro-bumps connecting the layers as shown in Fig. 6 as a driving example. The ambient temperature is set to be 300 K. To demonstrate the effectiveness of $\mathcal{H}$-based solver, we need to show the almost linear complexity of $\mathcal{H}$-based inverse and Cholesky factorization for the FEM thermal matrices obtained in terms of both memory space and CPU time.

The 3D IC structure is discretized into tetrahedral elements with four nodes in each element to generate 7239 to 124344 unknowns for the complexity analysis. The stiffness matrix $\mathbf{G}$ generated from 3D IC is directly transfered into $\mathbf{H}_{\text{steady}}$ for steady analysis while the stiffness matrix with the capacity matrix $\left[\mathbf{G} + \frac{\mathbf{C}}{\Delta t}\right]$ is transfered into $\mathbf{H}_{\text{transient}}$ for transient analysis. Fig. 7 shows the sparsity patterns before and after the translation. The simulation parameters of $\eta = 1$ and $n_{min} = 32$ are applied into both steady and transient scenario. $\varepsilon = 1\text{e}{-9}$ is chosen to control the adaptive rank k in each tree level. Since the rank varies, the mean of the rank $\bar{k}$ is used to evaluate the complexity such that $k_1 = \max(\bar{k}, n_{min})$. The mean rank $\bar{k}$ of each case is shown in Table. I.

TABLE I
$\bar{k}$ IN INVERSE AND CHOLESKY FACTORS

No. of unknowns	$\bar{k}$ in $\mathcal{H}$-Inverse	$\bar{k}$ in $\mathcal{H}$-Cholesky
7239	13.84	8.00
12300	16.07	8.55
31888	20.98	10.03
49687	23.75	10.81
76840	26.84	11.63
124344	30.70	12.61



(a) The stiffness matrix from the thermal FEM method

(b) $\mathcal{H}$ representation of the stiffness matrix

Fig. 7. Stiffness matrix $\mathbf{H}$ and the transformed $\mathcal{H}$-matrix

It is noticed that the structure of $\mathbf{H}_{\text{transient}}$ at each time step is identical to the structure of $\mathbf{H}_{\text{steady}}$, the storage complexity analysis of $\mathbf{H}_{\text{transient}}$ at any time interval can also apply to $\mathbf{H}_{\text{steady}}$ as well. For CPU time complexity analysis, the solution time t_i for a time step i of $\mathbf{H}_{\text{transient}}$ are the same for every step, the total time for solving (21) is a multiple of t_i, which also has a time complexity bounded by $\mathcal{O}\left(k_1^2 N \log^2 N\right)$.

Table II and III show the comparison between the storage of $\mathcal{H}$-matrix (denoted as [H] in the tables) and the storage of full matrix (denoted as [F] in the tables), as well as the comparison between the CPU time computed by $\mathcal{H}$-arithmetics (denoted as [H] in the tables) and the CPU time computed by Matlab built in command (denoted as [M] in the tables). According to the tables, memory required for storing inverse and Cholesky factors in the form of $\mathcal{H}$-matrix are much less than that in the form of full-matrix. In addition, the $\mathcal{H}$-matrix-based inverse and Cholesky factorization require less computation time compared with the Matlab solvers if the size of matrix is large. The advantage becomes obvious if the size of matrix is beyond 49687×49687 for inverse and 76840×76840 for Cholesky factorization.

Fig. 8 are plotted from Table II and III (blue lines) and the theoretical complexities (red line). They show the storage and CPU time complexities for computing the inverse and Cholesky factors of $\mathbf{H}_{\text{transient}}$. From those figures we can clearly see that the storage complexities and CPU time required for both inverse and Cholesky factor match very well with the theoretical trends, $\mathcal{O}(k_1 N \log N)$ and

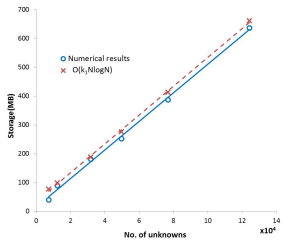
TABLE II
NUMERICAL RESULTS FOR CHOLESKY FACTORIZATION OF $\mathbf{H}_{\text{transient}}$

No. of unknowns	Storage (MB)[H]	Storage (GB)[F]	Memory saved (times)	CPU time (s)[H]	CPU time (s)[M]	Speedup (times)
7239	38.35	0.20	5.21	17.13	1.70	-10.08
12300	87.07	0.56	6.63	45.92	18.04	-2.56
31888	177.27	3.79	21.88	82.57	197.75	+2.39
49687	246.48	9.19	38.21	105.98	358.41	+3.38
76840	377.87	22.00	59.61	150.65	Fail	Inf
124344	622.55	57.60	57.60	258.09	Fail	Inf

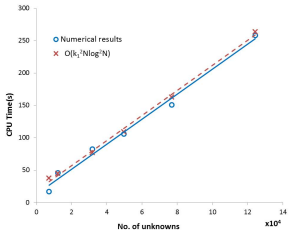
TABLE III
NUMERICAL RESULTS FOR DIRECT INVERSE OF $\mathbf{H}_{\text{transient}}$

No. of unknowns	Storage (MB)[H]	Storage (GB)[F]	Memory saved (times)	CPU time (s)[H]	CPU time (s)[M]	Speedup (times)
7239	186.29	0.39	2.15	243.23	42.55	-5.72
12300	419.79	1.13	2.75	598.88	235.25	-2.55
31888	1076.75	7.58	7.20	1392.86	2375.90	+1.71
49687	1711.33	18.39	11.01	2004.19	Fail	Inf
76840	2979.47	43.99	15.12	3550.83	Fail	Inf
124344	5610.69	115.19	21.02	7029.04	Fail	Inf

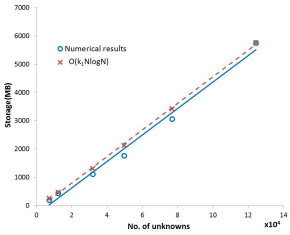
$\mathcal{O}(k_1^2 N \log^2 N)$, which are almost linear. Both results for inverse and Cholesky factors have error norm less than $1\text{e}{-6}$ K.



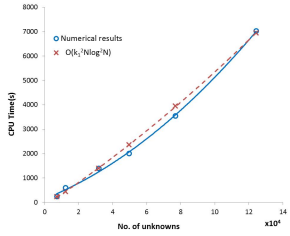
(a) Storage scalability of $\mathcal{H}$-Cholesky factors



(b) CPU time scalability for computing $\mathcal{H}$-Cholesky factors



(c) Storage scalability of $\mathcal{H}$-Inverse matices



(d) CPU time scalability for computing $\mathcal{H}$-Inverse matrices

Fig. 8. Storage and CPU time values of $\mathcal{H}$-Cholesky factors and inverse matrices and their predicted theoretical values.

VII. CONCLUSION

This paper has shown that the stiffness matrix from a FE-based steady and transient thermal analysis can be represented by $\mathcal{H}$-matrix without any approximation, and its inverse and Cholesky factors can be evaluated by $\mathcal{H}$-matrix with controlled accuracy. Numerical results match perfectly with the predicted theoretical scalability that the memory and time complexities of the solver are bounded by $\mathcal{O}(k_1 N \log N)$ and $\mathcal{O}(k_1^2 N \log^2 N)$ for very large scale thermal systems.

REFERENCES

[1] R. Patti, "3D integration: New opportunities for advanced packaging," in *Electrical Performance of Electronic Package and Systems (EPEPS)*, pp. 1–41, October 2011.

[2] J. Burns, "TSV-based 3D integration," in *Three Dimensional System Integration* (A. Papanikolaou, D. Soudris, and R. Radojcic, eds.), pp. 13–22, Springer, November 2010.

[3] M. N. Ozisik, *Finite Difference Methods in Heat Transfer*. Taylor & Francis, Inc., 1994.

[4] R. W. Lewis, P. Nithiarasu, and K. N. Seetharamu, *Fundamentals of the Finite Element Method for Heat and Fluid Flow*. John Wiley & Son, 2004.

[5] T. A. Davis, *Direct Methods for Sparse Linear Systems*. SIAM, Philadelphia, 2006.

[6] W. Hackbusch, "A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: Introduction to $\mathcal{H}$-matrices," *Computing*, vol. 62, no. 2, pp. 89–108, 1999.

[7] L. Grasedyck and W. Hackbusch, "Construction and arithmetics of $\mathcal{H}$-matrices," *Computing*, vol. 70, no. 4, pp. 295–334, 2003.

[8] S. Borm, L. Grasedyck, and W. Hackbusch, "Hierarchical Matrices." Lecture notes 21 of the Max Planck Institute for Mathematics in the Sciences, 2003.

[9] H. Liu and D. Jiao, "$\mathcal{H}$-Matrix-Based Fast Direct Finite Element Solver for Large-Scale Electromagnetic Analysis," *ECE Technical Reports, Purdue University*, 2010.

[10] H. Liu and D. Jiao, "Performance analysis of the $\mathcal{H}$-matrix-based fast direct solver for finite-element-based analysis of electromagnetic problems," in *Antennas and Propagation Society International Symposium, 2009. APSURSI '09. IEEE*, pp. 1–4, 2009.

[11] W. Chai and D. Jiao, "$\mathcal{H}$- and $\mathcal{H}^2$-matrix-based fast integral-equation solvers for large-scale electromagnetic analysis," *IET Microwaves, Antennas & Propagation*, vol. 4, no. 10, pp. 1583–1596, 2010.

[12] T. Wan, X. Q. Hu, and R. S. Chen, "An $\mathcal{H}$-matrix direct method with improved solution for finite element analysis of scattering problems," in *International Conference on Microwave and Millimeter Wave Technology (ICMMT)*, 2010.

[13] Y.-K. Cheng, C.-H. Tsai, C.-C. Teng, and S.-M. Kang, *Electrothermal Analysis of VLSI Systems*. Kluwer Academic Publishers, 2000.

[14] T. Bergman, A. Lavine, F. P. Incropera, and D. P. DeWitt, *Fundamentals of Heat and Mass Transfer*. New York: John Wiley & Sons, 7th ed., 2011.

[15] M. Bebendorf and W. Hackbusch, "Existence of $\mathcal{H}$-matrix approximants to the inverse FE-matrix of elliptic operators with L^∞-coefficients," *Numerische Mathematik*, vol. 95, no. 1, pp. 1–28, 2003.

[16] H. Liu and D. Jiao, "A Direct Finite-Element-Based Solver of Significantly Reduced Complexity for Solving Large-Scale Electromagnetic Problems," *Microwave Symposium Digest*, pp. 177–180, 2009.

[17] "Hlib: Hierarchical matries library." http://www.hlib.org.

[18] "Gmsh: a three-dimensional finite element mesh generator." http://geuz.org/gmsh/.