INTERNATIONAL JOURNAL OF CIRCUIT THEORY AND APPLICATIONS
Int. J. Circ. Theor. Appl. 2015; **43**:1953–1970
Published online 25 November 2014 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cta.2051

$\mathcal{H}^2$-matrix-based finite element linear solver for fast transient thermal analysis of high-performance ICs

Hai-Bao Chen[1], Sheldon X.-D. Tan[2,*,†], David H. Shin[2], Xin Huang[2], Hai Wang[3] and Guoyong Shi[1]

[1]*Department of Micro/Nano-electronics, Shanghai Jiao Tong University, Shanghai, China*
[2]*Department of Electrical and Computer Engineering, University of California, Riverside, CA, USA*
[3]*School of Microelectronics & Solid-State Electronics, University of Electronic Science & Technology of China, Chengdu, China*

SUMMARY

In this article, we propose $\mathcal{H}^2$-based finite element (FE) solver for transient thermal analysis of high-performance integrated circuits (ICs). $\mathcal{H}^2$-matrix is a special subclass of hierarchical matrix or $\mathcal{H}$-matrix, which was shown to provide a data-sparse way to approximate the matrices and their inverses with almost linear space and time complexities. In this work, we show that $\mathcal{H}^2$-based mathematical framework can also be applied to FE-based transient analysis of thermal parabolic partial differential equations. We show how the thermal matrix can be approximated by $\mathcal{H}^2$-representations with controlled error. Then, we demonstrate that both storage and time complexities of the new solver are bounded by $\mathcal{O}(N)$, where N is the matrix size. The method can be applied to any thermal structures for both steady and transient analysis. The numerical results from 3D ICs demonstrate the linear scalability of the proposed method in terms of both memory footprint and CPU time. The comparison with existing product-quality LU solvers, CSPARSE and UMFPACK, on a number of 3D IC thermal matrices, shows that the new method is much more memory efficient than these methods, which however prevents the demonstration of the potential speedup of the proposed method over those methods. Copyright © 2014 John Wiley & Sons, Ltd.

Received 23 June 2014; Revised 16 September 2014; Accepted 31 October 2014

KEY WORDS: finite element method; integrated circuits; H^2-matrix; thermal analysis

1. INTRODUCTION

Elevated on-chip temperature has become a top concern for high-performance microprocessor and integrated circuits (ICs) design as more devices are integrated on a chip. As the power density continues to increase, the excessively high temperature spots would adversely lead to performance degradation, increased cooling costs, reduced reliability, and signal integrity issues for high-performance ICs and microprocessors [1–4]. Thus, accurate and efficient thermal modeling and analysis are vital for the thermal-aware VLSI design [5–7] to improve performance, reliability, power reduction as well as online temperature regulation techniques [3,8].

Traditional thermal analysis solves the partial thermal diffusion equations using numerical approaches such as the finite difference method (FDM) [9, 10] and the finite element method (FEM) [11, 12]. 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-

*Correspondence to: Sheldon X.-D. Tan, Department of Electrical and Computer Engineering, University of California, Riverside, CA 92521, USA.
†E-mail: stan@ece.ucr.edu

dependent. For LU-based direct method, it typically is more expensive for both memory and computation with super-linear time complexity [13]

There exists a general mathematical framework called the hierarchical matrix ($\mathcal{H}$-matrix) framework [14–16], which enables a highly compact representation and efficient numerical computation of the matrices (dense or sparse). The $\mathcal{H}$-matrix method splits a large matrix into a hierarchy of blocks, which are approximated by low-rank matrices in factored form. Storage requirements of LU decomposition and matrix inverse using $\mathcal{H}$-matrix arithmetics have been shown to be of almost linear complexities $\mathcal{O}(N\log N)$ and $\mathcal{O}\left(N\log^2 N\right)$, where N is the size of the matrix. $\mathcal{H}$-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 can be described by only few data. The $\mathcal{H}$-matrix structure has been used in solving electrostatic and magneto-static problems in which the resulting electromagnetic wave PDEs are hyperbolic [17]. Recently, we proposed an FE-based method for transient thermal analysis of high-performance ICs, which is a parabolic problem, based on the $\mathcal{H}$-matrix representation [18, 19]. We approved that the $\mathcal{H}$-matrix based approach can still work for the parabolic problems. We also demonstrated that the $\mathcal{H}$-matrix representation method for thermal analysis of high-performance ICs can reduce the original dense storage $\mathcal{O}\left(N^2\right)$ to the sparse storage $\mathcal{O}(N\log N)$.

Later on, $\mathcal{H}^2$-matrices were further proposed [20–22], which are a specialized subclass of hierarchical matrices. It was shown that $\mathcal{H}^2$-matrices can be stored by linear space complexities. The resulting $\mathcal{H}^2$-matrix operations such as addition, multiplication, LU factorization, and even inverse can be performed in almost linear complexity. The nested structure is the key difference between general $\mathcal{H}$-matrices and $\mathcal{H}^2$-matrices, because it permits an efficient reuse of information across the entire cluster tree. The $\mathcal{H}^2$-matrix structure has been used to solve the elliptic PDEs [21, 22], integral equations [20], and recently for solving electromagnetic wave PDEs [23, 24].

It has been shown that the $\mathcal{H}^2$-matrix method can significantly reduce storage requirements for large problems arising from integral equations and elliptic PDEs. This article presents an $\mathcal{H}^2$-based FE linear solver that allows us to construct an $\mathcal{H}^2$-matrix without storing the entire original matrix. The main idea of the $\mathcal{H}^2$-matrix method is to approximate submatrices and combine them by row-index and column-index cluster spaces to form approximations of larger matrices until the entire thermal matrix has been treated. The $\mathcal{H}^2$-LU decomposition can be computed in the algebra of hierarchical matrices with almost liner complexity and with the same robustness as the existing LU-based CSPARSE [25] and UMFPACK [26] solvers. Low-precision approximations of the $\mathcal{H}^2$-representation can be used to precondition iterative solvers. Using the $\mathcal{H}^2$-matrix method, the storage requirements could be far less than modern direct solvers such as CSPARSE and UMFPACK, which will be shown later in the numerical results of Section 5.

In this article, the $\mathcal{H}^2$-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 [9]. Not many works have been reported for $\mathcal{H}^2$-matrix FE-based solvers for parabolic PDEs so far as most of works focused on either elliptic or hyperbolic PDEs. In this article, we propose an $\mathcal{H}^2$-matrix-based FE linear solver, which employs a refined representation of thermal matrices. We will show the $\mathcal{H}^2$-representation possesses a nested structure, which can be used to reduce the storage requirements of hierarchical matrices. With the multilevel structure of $\mathcal{H}^2$-matrices, the proposed $\mathcal{H}^2$-based solver can reduce the dense storage to the $\mathcal{O}(N)$ sparse storage. The contributions of this article are summarized as follows:

- We show that the $\mathcal{H}^2$-matrix techniques can be used to simulate the transient thermal model, which is described by a parabolic PDE. We demonstrate that the thermal matrix built from the thermal model by the FEM can be represented by an $\mathcal{H}^2$-matrix.
- We demonstrate that good approximates could be constructed with an upper bound on the error in the Frobenius norm. This error bound is related to the small singular values of a Gram matrix, which is used for the $\mathcal{H}^2$-representation of the thermal matrix.
- We reveal that the inverse of the thermal matrix generated from the FE model of the dynamic heat equation can be also approximated as an $\mathcal{H}^2$-representation. It is shown that for very large

DOI: 10.1002/cta

scale thermal systems, the memory of the $\mathcal{H}^2$-based solver can be bounded by $\mathcal{O}(N)$, where N is the matrix size.

The numerical results from a 3D IC demonstrate the linear scalability of the proposed method in terms of both memory footprint and CPU time. The comparison with existing product-quality LU solvers, CSPARSE, and UMFPACK, on a number of 3D IC thermal matrices, shows that the new method is much more memory efficient than these methods. Because those methods fail to work on some large examples on the given workstation, we cannot perform the CPU time comparisons on larger examples and thus cannot demonstrate the potential speedup of the proposed method over those methods.

The rest of this article is organized as follows. In Section 2, we introduce the basics for the FE model of the 3D thermal equation. In Section 3, we discuss the $\mathcal{H}^2$-representation of the thermal matrix and analyze the error for this $\mathcal{H}^2$-representation. In Section 4, the $\mathcal{H}^2$-matrix arithmetics in linear complexity are described. Section 5 presents some numerical results, and Section 6 concludes this article.

2. REVIEW OF FE-BASED THERMAL ANALYSIS

The transient heat conduction equation [27] for the circuit, package, and board levels is given by

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

with the Robin's boundary condition $\kappa(\vec{r},T)\frac{\partial T(\vec{r},t)}{\partial \mathbf{n}_i} = h_i(T(\vec{r},t) - T_a)$. In Equation (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 the boundary condition, n_i is the outward direction normal to the boundary condition i, h_i is the heat-transfer coefficient (Wm^{-2}K^{-1}), and T_a is the ambient temperature surrounding the thermal systems. If $h_i = 0$, the boundary condition is adiabatic; 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, Equation (1) can be written as

$$\kappa\left[\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2}\right] + g = \rho C_p \frac{\partial T}{\partial t}. \tag{2}$$

The corresponding boundary condition is given by

$$\begin{cases} T = T_b \text{ on } S_1, \\ \kappa\left[\frac{\partial T}{\partial x}\mathrm{n}_1 + \frac{\partial T}{\partial y}\mathrm{n}_2 + \frac{\partial T}{\partial z}\mathrm{n}_3\right] + q = 0 \text{ on } S_2, \\ \kappa\left[\frac{\partial T}{\partial x}\mathrm{n}_1 + \frac{\partial T}{\partial y}\mathrm{n}_2 + \frac{\partial T}{\partial z}\mathrm{n}_3\right] + h(T - T_a) = 0 \text{ on } S_3, \end{cases}$$

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

By the Galerkin FEM [11], the temperature is discretized over space as $T(x,y,z,t) = \sum_{i=1}^{N} \varphi_i(x,y,z)T_i(t)$, where $\varphi_i(x,y,z)$ are the shape functions, N is the number of nodes on the element, and $T_i(t)$ are the nodal temperatures. For simplicity, let $\Phi = [\varphi_1\varphi_2\cdots\varphi_N]$, $\mathbf{T} = [T_1T_2\cdots T_N]^T$, and

$$\mathbf{B} = \begin{bmatrix} \frac{\partial\varphi_1}{\partial x} & \frac{\partial\varphi_2}{\partial x} & \cdots & \frac{\partial\varphi_N}{\partial x} \\ \frac{\partial\varphi_1}{\partial y} & \frac{\partial\varphi_2}{\partial y} & \cdots & \frac{\partial\varphi_N}{\partial y} \\ \frac{\partial\varphi_1}{\partial z} & \frac{\partial\varphi_2}{\partial z} & \cdots & \frac{\partial\varphi_N}{\partial z} \end{bmatrix}.$$

DOI: 10.1002/cta

By using the FE discretization, we will have the following discretized equation:

$$\mathrm{C}\frac{\partial \mathrm{T}}{\partial t} + \mathrm{GT} = \mathrm{f}, \tag{3}$$

where the capacity matrix C, the conductance matrix G, and the thermal load f are given by

$$\begin{cases} \mathrm{C} = \int_{\Omega} \rho C_p \Phi^{\mathrm{T}} \Phi d\Omega, \\ \mathrm{G} = \int_{\Omega} \kappa \mathrm{B}^{\mathrm{T}} \mathrm{B} d\Omega + \int_{S} h \Phi^{\mathrm{T}} \Phi dS, \\ \mathrm{f} = \int_{\Omega} g \Phi^{\mathrm{T}} d\Omega - \int_{S_2} q \Phi^{\mathrm{T}} dS_2 + \int_{S_3} h T_a \Phi^{\mathrm{T}} dS_3. \end{cases} \tag{4}$$

For the steady state analysis of Equation (3), the time-dependent term $\frac{\partial \mathrm{T}}{\partial t}$ equals to zero. Thus, Equation (3) can be reduced to GT = f. For the transient response analysis of Equation (3), the FDM can be applied to compute this result. By using the backward difference method [9], Equation (3) can be approximated by the following linear system:

$$\left(\mathrm{G} + \frac{\mathrm{C}}{\Delta t}\right)\mathrm{T}^{n+1} = \mathrm{f}^{n+1} + \frac{\mathrm{C}}{\Delta t}\mathrm{T}^{n}, \tag{5}$$

where Δt is the time discretization step, T^{n+1} and f^{n+1} are the approximations of the temperature vector **T** and the thermal load vector f at time t_{n+1}, respectively.

3. $\mathcal{H}^2$-MATRIX FOR FE-BASED THERMAL ANALYSIS

When analyzing the transient thermal problem of high-performance ICs, the resulting FE model with high order is sometimes too expensive to solve. It is essential to seek more efficient algorithms for computational purpose. In this section, we present $\mathcal{H}^2$-matrix representations for the capacity matrix, the conductance matrix, and the thermal matrix.

3.1. Introduction of $\mathcal{H}^2$-matrix

The hierarchical matrix ($\mathcal{H}$-matrix) was introduced in treating elliptic PDEs. It enables a compact representation of dense matrices using low rank approximation. For an $\mathcal{H}$-matrix, its certain off-diagonal block $\mathbf{H}_b$ (how to determine the block will be shown later) with dimension $m \times n$ can be written as $\mathbf{H}_b = \mathbf{V}\mathbf{W}^{\mathrm{T}}$. **V** and **W** have dimensions of $m \times q$ and $n \times q$, respectively, where q is much smaller than m and n. This low rank approximation is made by interpolating the first integration variable of the kernel function of the original dense matrix. If the kernel function is smooth in the first variable, the interpolation will have good accuracy and resulting in the accurate and efficient sparse approximation of the $\mathcal{H}$-matrix against the original dense matrix. The $\mathcal{H}$-matrix reduces the original dense storage $\mathcal{O}(N^2)$ to the sparse storage $\mathcal{O}(NlogN)$.

Recently, new advances in hierarchical matrix research bring $\mathcal{H}^2$-matrix. The basic idea of $\mathcal{H}^2$-matrix is that instead of only interpolating the first integration variable of the kernel function integration, two integration variables are interpolated together to make the low rank approximation matrix even more compact. Different from the $\mathcal{H}$-matrix case, for $\mathcal{H}^2$-matrix, its certain off-diagonal block $\mathbf{H}_b$ with dimension $m \times n$ can be written as $\mathbf{H}_b = \mathbf{V}\mathbf{S}\mathbf{W}^{\mathrm{T}}$, and **V**, **S**, and **W** have the dimensions $m \times r$, $r \times r$, and $n \times r$, respectively, where r is much smaller than m and n. With the new low rank approximation technique, the $\mathcal{H}^2$-matrix further reduces the dense storage to the $\mathcal{O}(N)$ sparse storage.

3.2. Construction of $\mathcal{H}^2$-matrix

The $\mathcal{H}^2$-matrix [20] enables a data sparse representation with high accuracy for matrices, which often arise in modeling of complex physical processes. To define an $\mathcal{H}^2$-matrix, we have to consider a tree

DOI: 10.1002/cta

structure. Let t be the vertex of the tree $\mathcal{T}$, then we denote by $S(t)$ the set of sons of t. If $S(t)=\Phi$, t is a leaf of $\mathcal{T}$. For simplicity, $\mathfrak{L}(\mathcal{T})$ and $r(\mathcal{T})$ denote the set of leaves and the root of $\mathcal{T}$, respectively. Given the index set $\mathcal{I}=\{1,2,\cdots,N\}$, we define the cluster tree $\mathcal{T}_{\mathcal{I}}$ satisfying the following conditions: (i) $\mathcal{I}\in\mathcal{T}_{\mathcal{I}}$ and (ii) if $t\in\mathcal{T}_{\mathcal{I}}$ has no leaf, then $t=\dot{\bigcup}_{s\in\mathcal{S}(t)} s$, where the notation $\dot{\cdot}$ denotes the disjoint union. If we define the sets $\mathcal{I}_i^l=\{(i-1)2^{p-l}+1,\cdots,i2^{p-l}\}$, where $\mathcal{I}_1^0=\mathcal{I}$, $0\le l\le p$, and $1\le i\le 2^l$, obviously these sets form a cluster tree $\mathcal{T}_{\mathcal{I}}$ of height p. The superscript l indicates the level of the tree $\mathcal{T}_{\mathcal{I}}$. In the following, we denote by $|\mathcal{I}|$ the cardinality of the set $\mathcal{I}$ and call the vertex of the tree $\mathcal{T}_{\mathcal{I}}$ a cluster.

To build an $\mathcal{H}^2$-matrix $\mathbf{H}\in\mathbb{R}^{|\mathcal{I}|\times|\mathcal{I}|}$ for the entire FE model, we need to introduce the concept of admissibility condition, 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. For any two clusters r and s of the tree $\mathcal{T}_{\mathcal{I}}$, the admissibility condition is defined as follows:

$$\min\{\operatorname{diam}(\Omega_r),\operatorname{diam}(\Omega_s)\}\le\eta\operatorname{dist}(\Omega_r,\Omega_s), \tag{6}$$

where Ω_r and $\Omega_s\subseteq\Omega$ contain the supports of all the basis functions subject to r and s, $\operatorname{diam}(\cdot)$ is the Euclidean diameter of the subsets, $\operatorname{dist}(\cdot,\cdot)$ is the Euclidean distance of two subsets, and η is a positive constant. Given two clusters r and s satisfying the admissibility condition (6), the corresponding matrix block can be represented as an $\mathcal{H}^2$-matrix. According to the definition of the $\mathcal{H}^2$-matrix in [20], the admissible matrix block has a factorized form

$$\mathbf{H}_{r\times s}=\mathbf{V}_r\mathbf{S}_{r\times s}\mathbf{W}_s^{\mathrm{T}}, \tag{7}$$

where $\mathbf{V}_r\in\mathbb{R}^{|r|\times k_r}$, $\mathbf{S}_{r\times s}\in\mathbb{R}^{k_r\times k_s}$, and $\mathbf{W}_{r\times s}\in\mathbb{R}^{|s|\times k_s}$, in which k_r and k_s are less than $|r|$ and $|s|$. Because $\operatorname{rank}\left(\mathbf{V}_r\mathbf{S}_{r\times s}\mathbf{W}_s^{\mathrm{T}}\right)\le\operatorname{rank}(\mathbf{S}_{r\times s})\le min\{k_r,k_s\}$, an $\mathcal{H}^2$-matrix is also a hierarchical matrix. The families $(\mathbf{V}_r)_{r\in\mathcal{T}_{\mathcal{I}}}$ and $(\mathbf{W}_s)_{s\in\mathcal{T}_{\mathcal{I}}}$ are row and column cluster basis for the tree $\mathcal{T}_{\mathcal{I}}$, which are used for an $\mathcal{H}^2$-representation.

As an example, let us consider one-dimensional structure. The node index set $\mathcal{I}=\{1,2,\cdots,8\}$ with corresponding domain $[0,1]$ is shown in Figure 1. In Figure 2, we show the corresponding block cluster tree for $p=3$. The root of the block cluster tree is $\mathcal{I}\times\mathcal{I}$. The clusters that are admissible in Figure 2 are connected with red line and stored as leaves of the block cluster tree. No admissible clusters are found in other levels. The construction of block cluster tree defines the $\mathcal{H}$-matrix

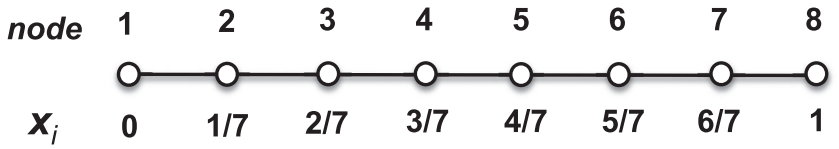

Figure 1. One-dimensional FEM mesh.

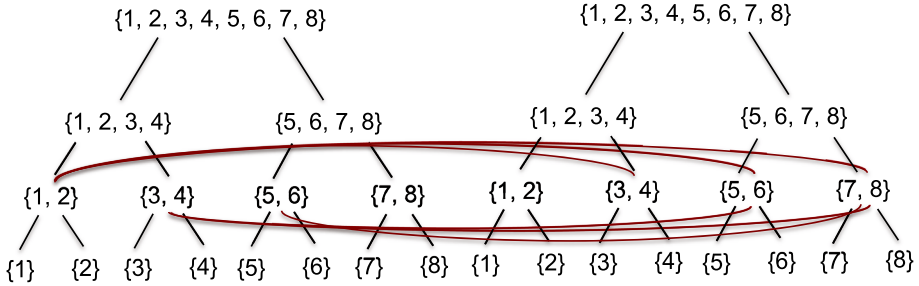

Figure 2. The block cluster tree for $p=3$.

DOI: 10.1002/cta

structure as shown in Figure 3, in which shaded matrix blocks are represented by Equation (7) (which are admissible blocks), while others are represented by full matrices.

3.3. $\mathcal{H}^2$-matrix for the thermal capacity matrix

In order to convert an arbitrary matrix into an $\mathcal{H}^2$-form, we need to construct the corresponding basis. For the heat capacity matrix $\mathbf{C}$ in Equation (4), because of its special property, orthogonal cluster basis can be built to simplify the conversion process. Before we approximate the matrix $\mathbf{C}$ into an $\mathcal{H}^2$-matrix, we first state and prove the following result.

Theorem 1
The heat capacity matrix, $\mathbf{C}=\int_\Omega \rho C_p \Phi^{\mathrm{T}}\Phi \mathrm{d}\Omega$ is a hierarchical matrix.

Proof
Given two admissible clusters $r, s \subseteq \mathcal{I}$, according to the structure of $\mathbf{C}$, we can see that there exist four matrices $\mathbf{K}_i$ and $\mathbf{L}_i$ for the matrix $\mathbf{C}_{r\times s}$ such that

$$\mathbf{C}_{r\times s} = \mathbf{K}_1\mathbf{L}_1^{\mathrm{T}} + \mathbf{K}_2\mathbf{L}_2^{\mathrm{T}} = [\mathbf{K}_1 \quad \mathbf{K}_2][\mathbf{L}_1 \quad \mathbf{L}_2]^{\mathrm{T}}.$$

From the previous expression, we know that

$$\mathrm{rank}(\mathbf{C}_{r\times s}) \le min\{\mathrm{rank}([\mathbf{K}_1 \quad \mathbf{K}_2]),\ \mathrm{rank}([\mathbf{L}_1 \quad \mathbf{L}_2])\},$$

which implies that the matrix $\mathbf{C}$ is a hierarchical matrix. □

We now represent the capacity matrix $\mathbf{C}$ as an $\mathcal{H}^2$-matrix. For the two admissible clusters r and s, we can see from theorem 1 that the matrix $\mathbf{C}_{r\times s}$ has the factorization $\mathbf{C}_{r\times s} = \mathbf{K}_{\mathbf{C}}\mathbf{L}_{\mathbf{C}}^{\mathrm{T}}$, where $\mathbf{K}_{\mathbf{C}}\in\mathbb{R}^{|r|\times k_{\mathcal{H}}}$ and $\mathbf{L}_{\mathbf{C}}\in\mathbb{R}^{|s|\times k_{\mathcal{H}}}$. To get a suitable row cluster basis for the matrix $\mathbf{C}_{r\times s}$, we need to compute the Gram matrix $\mathbf{C}_{r\times s}\mathbf{C}_{r\times s}^{\mathrm{T}} = \mathbf{K}_{\mathbf{C}}\left(\mathbf{L}_{\mathbf{C}}^{\mathrm{T}}\mathbf{L}\right)_{\mathbf{C}}\mathbf{K}_{\mathbf{C}}^{\mathrm{T}}$. From [16], the computation of all matrices $\mathbf{L}_{\mathbf{C}}^{\mathrm{T}}\mathbf{L}_{\mathbf{C}}$ requires a total complexity of $\mathcal{O}\left(k_{\mathcal{H}}^2 pN\right)$, where p is the height of the tree. Performing a singular value matrix decomposition on the matrix $\mathbf{K}_{\mathbf{C}}\left(\mathbf{L}_{\mathbf{C}}^{\mathrm{T}}\mathbf{L}_{\mathbf{C}}\right)\mathbf{K}_{\mathbf{C}}$ yields $\mathbf{K}_{\mathbf{C}}\left(\mathbf{L}_{\mathbf{C}}^{\mathrm{T}}\mathbf{L}_{\mathbf{C}}\right)\mathbf{K}_{\mathbf{C}} = \mathbf{V}_{\mathbf{C}}\Sigma_{\mathbf{C}}^2\mathbf{V}_{\mathbf{C}}^{\mathrm{T}}$, where $\mathbf{V}_{\mathbf{C}}\in\mathbb{R}^{|r|\times k_{\mathcal{H}}}$ and $\Sigma_{\mathbf{C}} = \mathrm{diag}\left\{\sigma_{\mathbf{C},1},\ \cdots,\sigma_{\mathbf{C},k_{\mathcal{H}}}\right\}\in\mathbb{R}^{k_{\mathcal{H}}\times k_{\mathcal{H}}}$ in which $\sigma_{\mathbf{C},1}\ge\cdots\ge\sigma_{\mathbf{C},k_r}\gg\sigma_{\mathbf{C},k_r+1}\ge\cdots\ge\sigma_{\mathbf{C},k_{\mathcal{H}}}\ge 0$. Thus, we can compute the row cluster basis $\mathbf{V}_{\mathbf{C},r}$ for the matrix $\mathbf{C}_{r\times s}$ as $\mathbf{V}_{\mathbf{C},r} = \mathbf{V}_{\mathbf{C}}\big|_{|r|\times k_r}$. A similar process of computing the row cluster basis can be used to build the column cluster basis for the matrix $\mathbf{C}_{r\times s}$. Concretely, we compute the following full singular decomposition:

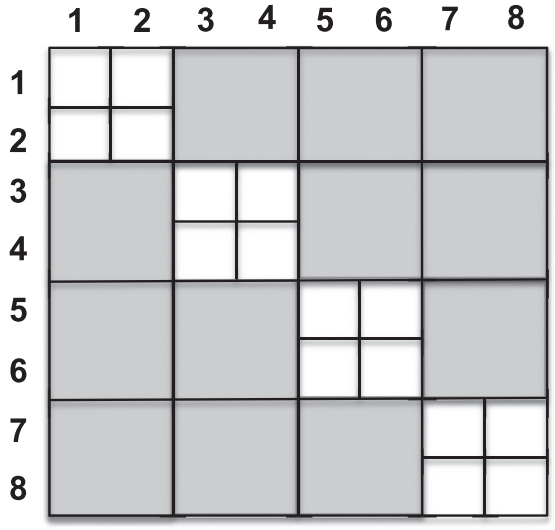

Figure 3. $\mathcal{H}^2$-matrix structure of 1D example.

DOI: 10.1002/cta

$$\mathbf{C}_{r\times s}^{\mathrm{T}}\mathbf{C}_{r\times s}=\mathbf{K}_{\mathbf{C}}\left(\mathbf{L}_{\mathbf{C}}^{\mathrm{T}}\mathbf{L}_{\mathbf{C}}\right)\mathbf{K}_{\mathbf{C}}=\mathbf{W}_{\mathbf{C}}\Sigma_{\mathbf{C}}^{2}\mathbf{W}_{\mathbf{C}}^{\mathrm{T}},$$

where $\mathbf{W}_{\mathbf{C}}\in\mathbb{R}^{|s|\times k_{\mathcal{H}}}$. Assuming that $\sigma_{\mathbf{C},1}\geq\cdots\geq\sigma_{\mathbf{C},k_s}\gg\sigma_{\mathbf{C},k_s+1}\geq\cdots\geq\sigma_{\mathbf{C},k_{\mathcal{H}}}\geq 0$, then the column cluster basis $\mathbf{W}_{\mathbf{C},s}=\mathbf{W}_{\mathbf{C}}\big|_{|s|\times k_s}$. Finally, the coupling matrix $\mathbf{S}_{\mathbf{C},r\times s}$ for the $\mathcal{H}^2$-representation of the matrix $\mathbf{C}_{r\times s}$ can be defined by $\mathbf{S}_{\mathbf{C},r\times s}=\mathbf{V}_{\mathbf{C},r}^{\mathrm{T}}\left(\mathbf{K}_{\mathbf{C}}\mathbf{L}_{\mathbf{C}}^{\mathrm{T}}\right)\mathbf{W}_{\mathbf{C},s}$. This representation has an upper error bound, which can be seen from the following theorem.

Theorem 2
For the matrix $\mathbf{C}_{r\times s}$ and its $\mathcal{H}^2$-representation form, we have

$$\left\|\mathbf{C}_{r\times s}-\mathbf{V}_{\mathbf{C},r}\mathbf{S}_{\mathbf{C},r\times s}\mathbf{W}_{\mathbf{C},s}^{\mathrm{T}}\right\|_F\leq\sqrt{\sum_{i=min\{k_r,k_s\}+1}^{k_{\mathcal{H}}}2\sigma_{\mathbf{C},i}^{2}}.$$

Proof
According to the definition of the Frobenius norm, we have

$$\begin{aligned}
&\left\|\mathbf{C}_{r\times s}-\mathbf{V}_{\mathbf{C},r}\mathbf{S}_{\mathbf{C},r\times s}\mathbf{W}_{\mathbf{C},s}^{\mathrm{T}}\right\|_F^2\\
=&\left\|\mathbf{C}_{r\times s}-\mathbf{V}_{\mathbf{C},r}\mathbf{V}_{\mathbf{C},r}^{\mathrm{T}}\mathbf{C}_{r\times s}\mathbf{W}_{\mathbf{C},s}\mathbf{W}_{\mathbf{C},s}^{\mathrm{T}}\right\|_F^2\\
=&\left\|\mathbf{C}_{r\times s}-\mathbf{V}_{\mathbf{C},r}\mathbf{V}_{\mathbf{C},r}^{\mathrm{T}}\mathbf{C}_{r\times s}+\mathbf{V}_{\mathbf{C},r}\mathbf{V}_{\mathbf{C},r}^{\mathrm{T}}\left(\mathbf{C}_{r\times s}-\mathbf{C}_{r\times s}\mathbf{W}_{\mathbf{C},s}\mathbf{W}_{\mathbf{C},s}^{\mathrm{T}}\right)\right\|_F^2\\
\leq&\left\|\mathbf{C}_{r\times s}-\mathbf{V}_{\mathbf{C},r}\mathbf{V}_{\mathbf{C},r}^{\mathrm{T}}\mathbf{C}_{r\times s}\right\|_F^2+\left\|\mathbf{V}_{\mathbf{C},r}\mathbf{V}_{\mathbf{C},r}^{\mathrm{T}}\right\|_2^2\left\|\left(\mathbf{C}_{r\times s}-\mathbf{C}_{r\times s}\mathbf{W}_{\mathbf{C},s}\mathbf{W}_{\mathbf{C},s}^{\mathrm{T}}\right)\right\|_F^2\\
\leq&\left\|\mathbf{C}_{r\times s}-\mathbf{V}_{\mathbf{C},r}\mathbf{V}_{\mathbf{C},r}^{\mathrm{T}}\mathbf{C}_{r\times s}\right\|_F^2+\left\|\mathbf{C}_{r\times s}^{\mathrm{T}}-\mathbf{W}_{\mathbf{C},s}\mathbf{W}_{\mathbf{C},s}^{\mathrm{T}}\mathbf{C}_{r\times s}^{\mathrm{T}}\right\|_F^2.
\end{aligned}$$

Based on the aforementioned singular value decompositions of the matrices $\mathbf{C}_{r\times s}\mathbf{C}_{r\times s}^{\mathrm{T}}$ and $\mathbf{C}_{r\times s}^{\mathrm{T}}\mathbf{C}_{r\times s}$, we can compute the singular value decompositions of the matrices $\mathbf{C}_{r\times s}$ and $\mathbf{C}_{r\times s}^{\mathrm{T}}$ as $\mathbf{C}_{r\times s}=\mathbf{V}_{\mathbf{C}}\Sigma_{\mathbf{C}}\mathbf{U}_1^{\mathrm{T}}$ and $\mathbf{C}_{r\times s}^{\mathrm{T}}=\mathbf{W}_{\mathbf{C}}\Sigma_{\mathbf{C}}\mathbf{U}_2^{\mathrm{T}}$, respectively, where $\mathbf{U}_1\in\mathbb{R}^{|s|\times k_{\mathcal{H}}}$ and $\mathbf{U}_2\in\mathbb{R}^{|r|\times k_{\mathcal{H}}}$ are both unitary matrices. Now we can see that

$$\begin{aligned}
&\left\|\mathbf{C}_{r\times s}-\mathbf{V}_{\mathbf{C},r}\mathbf{S}_{\mathbf{C},r\times s}\mathbf{W}_{\mathbf{C},s}^{\mathrm{T}}\right\|_F^2\\
\leq&\left\|\mathbf{V}_{\mathbf{C}}\Sigma_{\mathbf{C}}\mathbf{U}_1^{\mathrm{T}}-\mathbf{V}_{\mathbf{C},r}\mathbf{V}_{\mathbf{C},r}^{\mathrm{T}}\mathbf{C}_{r\times s}\right\|_F^2+\left\|\mathbf{W}_{\mathbf{C}}\textstyle\sum_{\mathbf{C}}\mathbf{U}_2^{\mathrm{T}}-\mathbf{W}_{\mathbf{C},s}\mathbf{W}_{\mathbf{C},s}^{\mathrm{T}}\mathbf{C}_{r\times s}^{\mathrm{T}}\right\|_F^2\\
\leq&\left\|\mathbf{V}_{\mathbf{C}}\Sigma_{\mathbf{C}}\mathbf{U}_1^{\mathrm{T}}-\mathbf{V}_{\mathbf{C}}|_{|r|\times k_r}\mathbf{V}_{\mathbf{C}}|_{|r|\times k_r}^{\mathrm{T}}\mathbf{V}_{\mathbf{C}}\Sigma_{\mathbf{C}}\mathbf{U}_1^{\mathrm{T}}\right\|_F^2\\
+&\left\|\mathbf{W}_{\mathbf{C}}\Sigma_{\mathbf{C}}\mathbf{U}_2^{\mathrm{T}}-\mathbf{W}_{\mathbf{C}}|_{|s|\times k_s}\mathbf{W}_{\mathbf{C}}|_{|s|\times k_s}^{\mathrm{T}}\mathbf{W}_{\mathbf{C}}\Sigma_{\mathbf{C}}\mathbf{U}_2^{\mathrm{T}}\right\|_F^2\\
=&\left\|\left(\mathbf{V}_{\mathbf{C}}-\mathbf{V}_{\mathbf{C}}|_{|r|\times k_r}\mathbf{V}_{\mathbf{C}}|_{|r|\times k_r}^{\mathrm{T}}\mathbf{V}_{\mathbf{C}}\right)\Sigma_{\mathbf{C}}\mathbf{U}_1^{\mathrm{T}}\right\|_F^2+\left\|\left(\mathbf{W}_{\mathbf{C}}-\mathbf{W}_{\mathbf{C}}|_{s|\times k_s}\mathbf{W}_{\mathbf{C}}|_{|s|\times k_s}^{\mathrm{T}}\mathbf{W}_{\mathbf{C}}\right)\Sigma_{\mathbf{C}}\mathbf{U}_2^{\mathrm{T}}\right\|_F^2\\
\leq&\sum_{i=k_r+1}^{k_{\mathcal{H}}}\sigma_{\mathbf{C},i}^2+\sum_{i=k_s+1}^{k_{\mathcal{H}}}\sigma_{\mathbf{C},i}^2\leq\sum_{i=\min\{k_r,k_s\}+1}^{k_{\mathcal{H}}}2\sigma_{\mathbf{C},i}^2,
\end{aligned}$$

which indicates our claim. This completes the proof. □

We can conclude that the truncation of the singular values related to the singular value decompositions of the Gram matrices does not change the $\mathcal{H}^2$-representation form of the heat capacity matrix. Theorem 2 shows that there exists a global error bound for this $\mathcal{H}^2$-representation method. The truncation of the small nonzero improper singular values may lead to an inaccurate approximation.

DOI: 10.1002/cta

3.4. $\mathcal{H}^2$-matrix for the thermal conductance matrix

In this section, we look at the problem of the $\mathcal{H}^2$-representation, the thermal conductance matrix $\mathbf{G}$ in Equation (4). Note that the matrix $\mathbf{G}$ has the same structure as the matrix $\mathbf{C}$. Similar to theorem 1, we can also prove that the matrix $\mathbf{G}=\int_\Omega \kappa \mathbf{B}^{\mathrm{T}}\mathbf{B}d\Omega+\int_S h\Phi^{\mathrm{T}}\Phi dS$ is a hierarchical matrix.

Now, we make use of the aforementioned method to compute the $\mathcal{H}^2$-representation of the matrix $\mathbf{G}$. Given two admissible clusters r and s, because of the hierarchical structure of the matrix $\mathbf{G}$, there exist two matrices $\mathbf{K}_{\mathbf{G}}\in\mathbb{R}^{|r|\times k_{\mathcal{H}}}$ and $\mathbf{L}_{\mathbf{G}}\in\mathbb{R}^{|s|\times k_{\mathcal{H}}}$ such that $\mathbf{G}_{r\times s}=\mathbf{K}_{\mathbf{G}}\mathbf{L}_{\mathbf{G}}^{\mathrm{T}}$. Taking a singular value decomposition of the matrix $\mathbf{G}_{r\times s}\mathbf{G}_{r\times s}^{\mathrm{T}}$, we get $\mathbf{G}_{r\times s}\mathbf{G}_{r\times s}^{\mathrm{T}}=\mathbf{V}_{\mathbf{G}}\Sigma_{\mathbf{G}}^2\mathbf{V}_{\mathbf{G}}^{\mathrm{T}}$, where $\mathbf{V}_{\mathbf{G}}\in\mathbb{R}^{|r|\times k_{\mathcal{H}}}$ and $\Sigma_{\mathbf{G}}=\mathrm{diag}\{\sigma_{\mathbf{G},1},\ \cdots,\sigma_{\mathbf{G},k_{\mathcal{H}}}\}\in\mathbb{R}^{k_{\mathcal{H}}\times k_{\mathcal{H}}}$. Similarly, performing the singular value decomposition of the matrix $\mathbf{G}_{r\times s}^{\mathrm{T}}\mathbf{G}_{r\times s}$ yields $\mathbf{G}_{r\times s}^{\mathrm{T}}\mathbf{G}_{r\times s}=\mathbf{W}_{\mathbf{G}}\Sigma_{\mathbf{G}}^2\mathbf{W}_{\mathbf{G}}^{\mathrm{T}}$, where $\mathbf{W}_{\mathbf{G}}\in\mathbb{R}^{|s|\times k_{\mathcal{H}}}$. We also assume that $\sigma_{\mathbf{G},1}\geq\cdots\geq\sigma_{\mathbf{G},min\{k_r,k_s\}}\gg\sigma_{\mathbf{G},min\{k_r,k_s\}+1}\geq\cdots\geq\sigma_{\mathbf{G},k_{\mathcal{H}}}\geq 0$. Therefore, by truncating the small singular values, the row and column orthogonal cluster basis can be constructed as $\mathbf{V}_{\mathbf{G},r}=\mathbf{V}_{\mathbf{G}}\big|_{|r|\times k_r}$ and $\mathbf{W}_{\mathbf{G},s}=\mathbf{W}_{\mathbf{G}}\big|_{|s|\times k_s}$, respectively. The corresponding coupling matrix $\mathbf{S}_{\mathbf{G},r\times s}$ can be given by $\mathbf{S}_{\mathbf{G},r\times s}=\mathbf{V}_{\mathbf{G},r}^{\mathrm{T}}\left(\mathbf{K}_{\mathbf{G}}\mathbf{L}_{\mathbf{G}}^{\mathrm{T}}\right)\mathbf{W}_{\mathrm{G},s}$. Similar to the result stated in theorem 2, there is a global error bound for the $\mathcal{H}^2$-representation $\mathbf{G}_{r\times s}=\mathbf{V}_{\mathbf{G},r}\mathbf{S}_{\mathbf{G},r\times s}\mathbf{W}_{\mathbf{G},s}^{\mathrm{T}}$.

3.5. $\mathcal{H}^2$-matrix for the transient thermal matrix

We now proceed to compute the $\mathcal{H}^2$-representation for the thermal matrix $\mathbf{G}+\frac{\mathbf{C}}{\Delta t}$. For simplicity, we denote by $\mathbf{H}$ the thermal matrix, that is, $\mathbf{H}=\mathbf{G}+\frac{\mathbf{C}}{\Delta t}$. From Equation (4), we can see that

$$\mathbf{H}=\int_\Omega \kappa\mathbf{B}^{\mathrm{T}}\mathbf{B}d\Omega+\int_S h\Phi^{\mathrm{T}}\Phi dS+\frac{1}{\Delta t}\int_\Omega \rho C_p\Phi^{\mathrm{T}}\Phi d\Omega,$$

which implies that the structure of the matrix $\mathbf{H}$ is the same as the matrix $\mathbf{G}$. Therefore, the matrix $\mathbf{H}$ is also a hierarchical matrix. Following the computation process discussed in the previous section, we can compute the row and column orthogonal cluster basis for the $\mathcal{H}^2$-representation of the matrix $\mathbf{H}$. The aforementioned technique tries to directly approximate the matrix $\mathbf{H}$ as an $\mathcal{H}^2$-representation by computing its corresponding cluster basis and coupling matrix. However, if the $\mathcal{H}^2$-representations of the matrices $\mathbf{G}$ and $\mathbf{C}$ have been computed, then we can compute the $\mathcal{H}^2$-representation of the matrix $\mathbf{H}$ by using the cluster basis and coupling matrices of the matrices $\mathbf{G}$ and $\mathbf{C}$. The following theorem is the computation process toward the goal of this representation.

Theorem 3

Suppose that the matrices $\mathbf{G}$ and $\mathbf{C}$ have been represented as the $\mathcal{H}^2$-matrices. Then the matrix $\mathbf{H}$ also has an $\mathcal{H}^2$-representation $\mathbf{H}_{r\times s}=\mathbf{V}_{\mathbf{H},r}\mathbf{S}_{\mathbf{H},r\times s}\mathbf{W}_{\mathbf{H},s}^{\mathrm{T}}$ for the admissible clusters r and s.

Proof

Without loss of generality, suppose that for two arbitrary clusters r and s, the matrices $\mathbf{C}_{r\times s}$ and $\mathbf{G}_{r\times s}$ can be represented as $\mathbf{C}_{r\times s}=\mathbf{V}_{\mathbf{C},r}\mathbf{S}_{\mathbf{C},r\times s}\mathbf{W}_{\mathbf{C},s}^{\mathrm{T}}$ and $\mathbf{G}_{r\times s}=\mathbf{V}_{\mathbf{G},r}\mathbf{S}_{\mathbf{G},r\times s}\mathbf{W}_{\mathbf{G},s}^{\mathrm{T}}$, respectively, then we have

$$\begin{aligned}\mathbf{H}_{r\times s}&=\frac{1}{\Delta t}\mathbf{C}_{r\times s}+\mathbf{G}_{r\times s}\\&=\frac{1}{\Delta t}\mathbf{V}_{\mathbf{C},r}\mathbf{S}_{\mathbf{C},r\times s}\mathbf{W}_{\mathbf{C},s}^{\mathrm{T}}+\mathbf{V}_{\mathbf{G},r}\mathbf{S}_{\mathbf{G},r\times s}\mathbf{W}_{\mathbf{G},s}^{\mathrm{T}}\\&=\begin{bmatrix}\frac{1}{\Delta t}\mathbf{V}_{\mathbf{C},r} & \mathbf{V}_{\mathbf{G},r}\end{bmatrix}\begin{bmatrix}\mathbf{S}_{\mathbf{C},r\times s} & 0\\ 0 & \mathbf{S}_{\mathbf{G},r\times s}\end{bmatrix}\begin{bmatrix}\mathbf{W}_{\mathbf{C},s}^{\mathrm{T}}\\ \mathbf{W}_{\mathbf{G},s}^{\mathrm{T}}\end{bmatrix}.\end{aligned}$$

Let $\mathbf{V}_{\mathbf{H},r}=\begin{bmatrix}\frac{\mathbf{V}_{\mathbf{C},r}}{\Delta t} & \mathbf{V}_{\mathbf{G},r}\end{bmatrix}$, $\mathbf{S}_{\mathbf{H},r\times s}=\mathrm{diag}\{\mathbf{S}_{\mathbf{C},r\times s},\mathbf{S}_{\mathbf{G},r\times s}\}$, and $\mathbf{W}_{\mathbf{H},s}=[\mathbf{W}_{\mathbf{C},s}\ \mathbf{W}_{\mathbf{G},s}]$, then it has $\mathbf{H}_{r\times s}=\mathbf{V}_{\mathbf{H},r}\mathbf{S}_{\mathbf{H},r\times s}\mathbf{W}_{\mathbf{H},s}^{\mathrm{T}}$, which indicates an $\mathcal{H}^2$-representation of the matrix $\mathbf{H}$. This completes the proof. □

We are therefore led to a methodology for the $\mathcal{H}^2$-representation of the matrix $\mathbf{H}$ by the proof process of theorem 3. To build an $\mathcal{H}^2$-representation of $\mathbf{H}$, one might compute the row and column cluster basis

DOI: 10.1002/cta

and coupling matrices of the matrices **G** and **C**. There exists a global error bound for this representation, which can be stated as the following result.

Theorem 4
For the thermal matrix **H** and its $\mathcal{H}^2$-representation form, we have

$$\left\|\mathbf{H}_{r\times s}-\mathbf{V}_{\mathbf{H},r}\mathbf{S}_{\mathbf{H},r\times s}\mathbf{W}_{\mathbf{H},s}^{\mathrm{T}}\right\|_F\leq\sqrt{\sum_{i=min\{k_r,,k_s\}+1}^{k_{\mathcal{H}}}2\left(\frac{\sigma_{\mathbf{C},i}^2}{\Delta t}+\sigma_{\mathbf{G},i}^2\right)}.$$

Proof
According to the $\mathcal{H}^2$-representations of the matrices $\mathbf{C}_{r\times s}$ and $\mathbf{G}_{r\times s}$, we obtain

$$\begin{aligned}&\left\|\mathbf{H}_{r\times s}-\mathbf{V}_{\mathbf{H},r}\mathbf{S}_{\mathbf{H},r\times s}\mathbf{W}_{\mathbf{H},s}^{\mathrm{T}}\right\|_F^2\\=&\left\|\frac{1}{\Delta t}\left(\mathbf{C}_{r\times s}-\mathbf{V}_{\mathbf{C},r}\mathbf{S}_{\mathbf{C},r\times s}\mathbf{W}_{\mathbf{C},s}^{\mathrm{T}}\right)+\left(\mathbf{G}_{r\times s}-\mathbf{V}_{\mathbf{G},r}\mathbf{S}_{\mathbf{G},r\times s}\mathbf{W}_{\mathbf{G},s}^{\mathrm{T}}\right)\right\|_F^2\\\leq&\frac{1}{\Delta t}\left\|\mathbf{C}_{r\times s}-\mathbf{V}_{\mathbf{C},r}\mathbf{S}_{\mathbf{C},r\times s}\mathbf{W}_{\mathbf{C},s}^{\mathrm{T}}\right\|_F^2+\left\|\mathbf{G}_{r\times s}-\mathbf{V}_{\mathbf{G},r}\mathbf{S}_{\mathbf{G},r\times s}\mathbf{W}_{\mathbf{G},s}^{\mathrm{T}}\right\|_F^2.\end{aligned}$$

Applying theorem 2 on the previous inequality, we get

$$\left\|\mathbf{H}_{r\times s}-\mathbf{V}_{\mathbf{H},r}\mathbf{S}_{\mathbf{H},r\times s}\mathbf{W}_{\mathbf{H},s}^{\mathrm{T}}\right\|_F^2\leq\sum_{i=min\{k_r,,k_s\}+1}^{k_{\mathcal{H}}}\frac{2\sigma_{\mathbf{C},i}^2}{\Delta t}+2\sigma_{\mathbf{G},i}^2.$$

This completes the proof.

The previous theorem leads to a global error result for the $\mathcal{H}^2$-representation of the matrix **H**. Note that, as aforementioned, this bound is in terms of the small singular values of the matrices $\mathbf{G}_{r\times s}\mathbf{G}_{r\times s}^{\mathrm{T}}$ and $\mathbf{C}_{r\times s}\mathbf{C}_{r\times s}^{\mathrm{T}}$. This error result is analogous to the error result of theorem 2. In the following, we present a general procedure for calculating the $\mathcal{H}^2$-representation of the thermal matrix **H** using the $\mathcal{H}^2$-representations of the matrices **G** and **C**. We summarize the algorithm as follows:

Step 1. For the clusters r and s, we compute the hierarchical $\mathcal{H}$-matrix representation as $\mathbf{G}_{r\times s}=\mathbf{K}_{\mathbf{G}}\mathbf{L}_{\mathbf{G}}^{\mathrm{T}}$.
Step 2. We perform the singular value decompositions on the Gram matrices $\mathbf{G}_{r\times s}\mathbf{G}_{r\times s}$ and $\mathbf{G}_{r\times s}^{\mathrm{T}}\mathbf{G}_{r\times s}$ as $\mathbf{G}_{r\times s}\mathbf{G}_{r\times s}^{\mathrm{T}}=\mathbf{V}_{\mathbf{G}}\Sigma_{\mathbf{G}}^2\mathbf{V}_{\mathbf{G}}^{\mathrm{T}}$ and $\mathbf{G}_{r\times s}^{\mathrm{T}}\mathbf{G}_{r\times s}=\mathbf{W}_{\mathbf{G}}\Sigma_{\mathbf{G}}^2\mathbf{W}_{\mathbf{G}}^{\mathrm{T}}$, respectively.
Step 3. For the capacity matrix **C**, we compute its hierarchical representation as $\mathbf{C}_{r\times s}=\mathbf{K}_{\mathbf{C}}\mathbf{L}_{\mathbf{C}}^{\mathrm{T}}$ and compute the singular value decompositions of the Gram matrices $\mathbf{C}_{r\times s}\mathbf{C}_{r\times s}^{\mathrm{T}}$ and $\mathbf{C}_{r\times s}^{\mathrm{T}}\mathbf{C}_{r\times s}$ as $\mathbf{C}_{r\times s}\mathbf{C}_{r\times s}^{\mathrm{T}}=\mathbf{V}_{\mathbf{C}}\Sigma_{\mathbf{C}}^2\mathbf{V}_{\mathbf{C}}^{\mathrm{T}}$ and $\mathbf{C}_{r\times s}^{\mathrm{T}}\mathbf{C}_{r\times s}=\mathbf{W}_{\mathbf{C}}\Sigma_{\mathbf{C}}^2\mathbf{W}_{\mathbf{C}}^{\mathrm{T}}$, respectively.
Step 4. For the matrices **G** and **C**, we compute their row and column cluster basis as $\mathbf{V}_{\mathbf{G},r}=\mathbf{V}_{\mathbf{G}}\big|_{|r|\times k_r}$, $\mathbf{W}_{\mathbf{G},s}=\mathbf{W}_{\mathbf{G}}\big|_{|s|\times k_s}$, $\mathbf{V}_{\mathbf{C},r}=\mathbf{V}_{\mathbf{C}}\big|_{|r|\times k_r}$, and $\mathbf{W}_{\mathbf{C},s}=\mathbf{W}_{\mathbf{C}}\big|_{|s|\times k_s}$, respectively. The corresponding coupling matrices are given by $\mathbf{S}_{\mathbf{G},r\times s}=\mathbf{V}_{\mathbf{G},r}^{\mathrm{T}}\left(\mathbf{K}_{\mathbf{G}}\mathbf{L}_{\mathbf{G}}^{\mathrm{T}}\right)\mathbf{W}_{\mathbf{G},s}$ and $\mathbf{S}_{\mathbf{C},r\times s}=\mathbf{V}_{\mathbf{C},r}^{\mathrm{T}}\left(\mathbf{K}_{\mathbf{C}}\mathbf{L}_{\mathbf{C}}^{\mathrm{T}}\right)\mathbf{W}_{\mathbf{C},s}$.
Step 5. For the thermal matrix **H**, we compute its $\mathcal{H}^2$-representation as $\mathbf{H}_{r\times s}=\mathbf{V}_{\mathbf{H},r}\mathbf{S}_{\mathbf{H},r\times s}\mathbf{W}_{\mathbf{H},s}^{\mathrm{T}}$, where $\mathbf{V}_{\mathbf{H},r}=\begin{bmatrix}\frac{\mathbf{V}_{\mathbf{C},r}}{\Delta t} & \mathbf{V}_{\mathbf{G},r}\end{bmatrix}$, $\mathbf{S}_{\mathbf{H},r\times s}=\mathrm{diag}\{\mathbf{S}_{\mathbf{C},r\times s},\mathbf{S}_{\mathbf{G},r\times s}\}$, and $\mathbf{W}_{\mathbf{H},s}=[\mathbf{W}_{\mathbf{C},s}\ \mathbf{W}_{\mathbf{G},s}]$.

4. $\mathcal{H}^2$-MATRIX-BASED LU DECOMPOSITION

In this section, we show that the inverse of the thermal matrix can be represented as an $\mathcal{H}^2$-matrix, and the $\mathcal{H}^2$-representation possesses a nested structure. Particular emphasis is on the $\mathcal{H}^2$-LU factorization and its complexity analysis.

DOI: 10.1002/cta

4.1. Inverse of the thermal matrix

Because the inverse of the thermal matrix $\mathbf{H}$ is rather expensive, we need to develop an algorithm that computes an approximation to the inverse of $\mathbf{H}$ without the need to invert $\mathbf{H}$. Let the matrix $\mathbf{H}$ be given in 2×2 block form

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

with the submatrix $\mathbf{H}_{11}$ regular. Thus, the inverse of $\mathbf{H}$ can be computed by use of the following equation:

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

where $\mathbf{S} = \mathbf{H}_{22} - \mathbf{H}_{21}\mathbf{H}_{11}^{-1}\mathbf{H}_{12}$. If the inverses $\mathbf{H}_{11}^{-1}$ and $\mathbf{S}^{-1}$ are already done, then only multiplications and additions of submatrices are required to be performed. This process only requires computing the inverse of $\mathbf{H}_{11}$ and $\mathbf{S}$, which is much faster than computing the whole inverse of $\mathbf{H}$.

To solve the system (5), one does not need the whole inverse but only an algorithm to perform matrix vector multiplications. In that case, it is sufficient to compute a Cholesky or LU decomposition of the matrix $\mathbf{H}$. Before introducing the $\mathcal{H}^2$-LU factorization, we show that the matrix $\mathbf{H}^{-1}$ has an $\mathcal{H}^2$-matrix format because of the $\mathcal{H}^2$-matrix structure of the matrices $\mathbf{G}$ and $\mathbf{C}$. This result can be seen from the following theorem.

Theorem 5
Suppose that the matrices $\mathbf{G}^{-1}$ and $\mathbf{C}^{-1}$ can be represented as $\mathcal{H}^2$-matrices, then the matrix $\mathbf{H}^{-1}$ also has an $\mathcal{H}^2$-representation.

Proof
Given two arbitrary clusters $r, r, s \subset \mathcal{I}$, we assume that the matrices $\mathbf{G}^{-1}$ and $\mathbf{C}^{-1}$ have the $\mathcal{H}^2$-representations $\mathbf{G}^{-1}_{r\times s} = \mathbf{V}_{\mathbf{G}^{-1},r}\mathbf{S}_{\mathbf{G}^{-1},r\times s}\mathbf{W}^{\mathrm{T}}_{\mathbf{G}^{-1},s}$ and $\mathbf{C}^{-1}_{r\times s} = \mathbf{V}_{\mathbf{C}^{-1},r}\mathbf{S}_{\mathbf{C}^{-1},r\times s}\mathbf{W}^{\mathrm{T}}_{\mathbf{C}^{-1},s}$, respectively. Because the matrices $\mathbf{G}$ and $\mathbf{C}$ are both nonsingular, we have

$$\mathbf{H}^{-1} = \mathbf{G}^{-1} - \mathbf{G}^{-1}\left(\left(\frac{\mathbf{C}}{\Delta t}\right)^{-1} + \mathbf{G}^{-1}\right)^{-1}\mathbf{G}^{-1}.$$

Let $\mathbf{J} = \mathbf{G}^{-1}\left(\left(\frac{\mathbf{C}}{\Delta t}\right)^{-1} + \mathbf{G}^{-1}\right)^{-1}\mathbf{G}^{-1}$. By the arithmetic process of matrix multiplication, we know that the matrix $\mathbf{J}$ is a hierarchical matrix. Using the aforementioned conversion of a hierarchical matrix to an $\mathcal{H}^2$-matrix, it is easy to verify that there exists an $\mathcal{H}^2$-representation for the matrix $\mathbf{J}$. Let $\mathbf{J}_{r\times s} = \mathbf{V}_{\mathbf{J},r}\mathbf{S}_{\mathbf{J},r\times s}\mathbf{W}^{\mathrm{T}}_{\mathbf{J},s}$, where $\mathbf{V}_{\mathbf{J},r}$ and $\mathbf{W}_{\mathbf{J},s}$ are the row and column cluster basis, respectively, the matrix $\mathbf{S}_{\mathbf{J},r\times s}$ is the corresponding coupling matrix. According to $\mathbf{H}^{-1}_{r\times s} = \mathbf{G}^{-1}_{r\times s} - \mathbf{J}_{r\times s}$, we arrive at $\mathbf{H}^{-1}_{r\times s} = \mathbf{V}_{\mathbf{H}^{-1},r}\mathbf{S}_{\mathbf{H}^{-1},r\times s}\mathbf{W}^{\mathrm{T}}_{\mathbf{H}^{-1},s}$, where the coupling matrix is defined as $\mathbf{S}_{\mathbf{H}^{-1},r\times s} = \mathrm{diag}\left\{\mathbf{S}_{\mathbf{G}^{-1},r\times s}, \mathbf{S}_{\mathbf{J},r\times s}\right\}$ and the row and column cluster basis are given by $\mathbf{V}_{\mathbf{H}^{-1},r} = \begin{bmatrix}\mathbf{V}_{\mathbf{G}^{-1},r} & \mathbf{V}_{\mathbf{J},r}\end{bmatrix}$ and $\mathbf{W}_{\mathbf{H}^{-1},s} = \begin{bmatrix}\mathbf{W}_{\mathbf{G}^{-1},s} & \mathbf{W}_{\mathbf{J},s}\end{bmatrix}$. This completes the proof.

4.2. Construction of nested cluster basis in $\mathcal{H}^2$

We now show that for the clusters r and s, the $\mathcal{H}^2$-representation $\mathbf{H}_{r\times s} = \mathbf{V}_r\mathbf{S}_{r\times s}\mathbf{W}^{\mathrm{T}}_s$ possesses a nested structure [20]. For simplicity, let $k_r = k_s = k_l$, where k_l is a function of the level l, then we get $\mathbf{V}_r \in \mathbb{R}^{|r|\times k_l}$, $\mathbf{S}_{r\times s} \in \mathbb{R}^{k_l\times k_l}$, and $\mathbf{W}_s \in \mathbb{R}^{|s|\times k_l}$. Let $\mathbf{V}_r = \begin{bmatrix} v_{r,1}\ v_{r,2}\ \cdots\ v_{r,k_l}\end{bmatrix}$ and $\mathbf{W}_s = \begin{bmatrix} w_{s,1}\ w_{s,2}\ \cdots\ w_{s,k_l}\end{bmatrix}$. Based on the matrices $\mathbf{V}_r$ and $\mathbf{W}_s$, we define an $\mathcal{R}$-matrix in the form $\sum_{i,j=1}^{k_l}\left[v_{r,i}, w_{s,j}\right]$, where $\left[v_{r,i}, w_{s,j}\right] = v_{r,i}w^{\mathrm{T}}_{s,j}$ with column vectors $v_{r,i}$ and row vectors $w^{\mathrm{T}}_{s,j}$. Corresponding to the vectors $v_{r,i}$ and $w_{s,j}$, the row-

DOI: 10.1002/cta

index and column-index cluster spaces can be defined as $\mathcal{V}(r) = \mathrm{span}\{v_{r,i} : \quad 1 \le i \le k_l\} \subseteq \mathbb{R}^{|r|}$ and $\mathcal{W}(s) = \mathrm{span}\{w_{r,j} : \quad 1 \le j \le k_l\} \subseteq \mathbb{R}^{|s|}$, respectively. The corresponding $\mathcal{R}$-matrices for the clusters r and s belong to the following tensor vector space:

$$S(r \times s) = \mathrm{span}\{[v_{r,i}, w_{s,j}] : \quad 1 \le i,j \le k_l\} = \mathcal{V}(r) \times \mathcal{W}(s).$$

If $\mathcal{V}(r)$ and $\mathcal{W}(s)$ have been computed, then it requires k_l^2 coefficients to represent an $\mathcal{R}$-matrix spanned by the space $\mathcal{S}(r \times s)$.

Let r be not a leaf of the tree $\mathcal{T}_{\mathcal{I}}$. Its sons are denoted by r_1 and r_2. From [20], we can compute the restriction operators $\mathbf{T}_v^{r_z,r}$ $(z = 1, 2)$ such that $v_{\mathrm{r,i}} = \begin{bmatrix} \mathbf{T}_v^{r_1,r} v_{r,i} \\ \mathbf{T}_v^{r_2,r} v_{r,i} \end{bmatrix}$. Similarly, for the vectors $w_{s,j}$, we define the restriction operators $\mathbf{T}_w^{s_z,s}$ such that $w_{s,j} = \begin{bmatrix} \mathbf{T}_w^{s_1,s} w_{s,j} \\ \mathbf{T}_w^{s_2,r} w_{s,j} \end{bmatrix}$. Let $v_{r_z,i} = \mathbf{T}_v^{r_z,r} v_{r,i}$ and $w_{s_z,j} = \mathbf{T}_w^{s_z,s} w_{s,j}$, then for these spaces $\mathcal{V}(r_z)$, $\mathcal{V}(r)$, $\mathcal{W}(s_z)$, and $\mathcal{W}(s)$, we get $\mathcal{V}(r_z) = \mathbf{T}_v^{r_z,r} \mathcal{V}(r)$ and $\mathcal{W}(s_z) = \mathbf{T}_w^{s_z,r} \mathcal{W}(s)$. From the earlier equalities, it is easy to see that we can define the coefficients $\alpha_{i,j}^{r_z,r}$ and $\beta_{j,i}^{s_z,s}$ such that

$$\mathbf{T}_v^{r_z,r} v_{r,i} = \sum_{j=1}^{k_{l+1}} \alpha_{i,j}^{r_z,r} v_{r_z,i}, \ \mathbf{T}_w^{s_z,r} w_{s,j} = \sum_{i=1}^{k_{l+1}} \beta_{j,i}^{s_z,s} w_{s_z,i}.$$

In practice, we need not to store the vectors $v_{r,i}$ and $w_{s,j}$ explicitly. Instead, one can only store the coefficients $\alpha_{i,j}^{r_z,r}$ and $\beta_{j,i}^{s_z,s}$. According to the nested structure, efficient algorithms can be constructed by re-using information across the cluster tree.

4.3. $\mathcal{H}^2$-LU factorization

For the 3D FE-based thermal extraction, to solve the system (5) we need not to compute the inverse of the $\mathcal{H}^2$-matrix $\mathbf{H}$. It is effective to perform LU decomposition on the matrix $\mathbf{H}$. Note that the thermal matrix $\mathbf{H}$ is symmetric and positive semidefinite. Therefore, we can define an $\mathcal{H}^2$-LU decomposition as the form $\mathbf{H} = \mathbf{L}_{\mathcal{H}^2} \mathbf{U}_{\mathcal{H}^2}$, where the matrices $\mathbf{L}_{\mathcal{H}^2}$ are $\mathbf{U}_{\mathcal{H}^2}$ are stored in the hierarchal matrix. Without loss of generality, suppose that the thermal matrix $\mathbf{H}$ can be partitioned into 2×2 submatrices. Thus, we assume that the matrices $\mathbf{L}_{\mathcal{H}^2}$ and $\mathbf{U}_{\mathcal{H}^2}$ can be written as follows:

$$\mathbf{L}_{\mathcal{H}^2} = \begin{bmatrix} \mathbf{L}_{11} & 0 \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{bmatrix}, \ \mathbf{U}_{\mathcal{H}^2} = \begin{bmatrix} \mathbf{U}_{11} & \mathbf{U}_{12} \\ 0 & \mathbf{U}_{22} \end{bmatrix}.$$

For the admissible block, the $\mathcal{H}^2$-LU decomposition of the matrix $\mathbf{H}$ can recursively be realized from the following procedure:

Step 1. Compute the matrices $\mathbf{L}_{11}$ and $\mathbf{U}_{11}$ by performing the $\mathcal{H}^2$-LU decomposition on the matrix $\mathbf{H}_{11}$.
Step 2. Compute the matrix $\mathbf{U}_{12}$ by solving the linear system $\mathbf{L}_{11}\mathbf{U}_{12} = \mathbf{H}_{12}$.
Step 3. Compute the matrix $\mathbf{L}_{21}$ by solving the linear system $\mathbf{L}_{21}\mathbf{U}_{11} = \mathbf{H}_{21}$.
Step 4. Compute the matrices $\mathbf{L}_{22}$ and $\mathbf{U}_{22}$ by performing the $\mathcal{H}^2$-LU decomposition of the matrix $\mathbf{H}_{22} - \mathbf{L}_{21}\mathbf{U}_{12}$.

In the earlier $\mathcal{H}^2$-LU computation process, we need to solve a lower triangular system $\mathbf{L}_{11}\mathbf{U}_{12} = \mathbf{H}_{12}$, where $\mathbf{L}_{11}$ and $\mathbf{H}_{12}$ are the input matrices with the hierarchical structure, an upper triangular system $\mathbf{L}_{21}\mathbf{U}_{11} = \mathbf{H}_{21}$ with the hierarchical input matrices $\mathbf{L}_{21}$ and $\mathbf{H}_{21}$. It should be pointed out that for the inadmissible block, a normal full LU decomposition can be performed straightforwardly. After the thermal matrix $\mathbf{H}$ is factorized as $\mathbf{H} = \mathbf{L}_{\mathcal{H}^2} \mathbf{U}_{\mathcal{H}^2}$, the linear system (5) can be solved in the following two steps:

DOI: 10.1002/cta

Step 1. Compute $\mathbf{X}^{n+1}$ by solving the lower triangular system $\mathbf{L}_{\mathcal{H}^2}\mathbf{X}^{n+1} = \mathbf{Y}^{n+1}$, where $\mathbf{Y}^{n+1} = \mathbf{f}^{n+1} + \frac{\mathbf{C}}{\Delta t}\mathbf{T}^n$.
Step 2. Compute $\mathbf{T}^{n+1}$ by solving the upper triangular system $\mathbf{U}_2\mathbf{T}^{n+1} = \mathbf{X}^{n+1}$.

Let $\mathbf{X}^{n+1} = \begin{bmatrix} \mathbf{X}_1^{n+1} \\ \mathbf{X}_2^{n+1} \end{bmatrix}$ and $\mathbf{Y}^{n+1} = \begin{bmatrix} \mathbf{Y}_1^{n+1} \\ \mathbf{Y}_2^{n+1} \end{bmatrix}$, then using the $\mathcal{H}^2$-matrix arithmetics, we get $\mathbf{X}_1^{n+1}$ by solving the system $\mathbf{L}_{11}\mathbf{X}_1^{n+1} = \mathbf{Y}_1^{n+1}$ and $\mathbf{X}_2^{n+1}$ from the system $\mathbf{L}_{22}\mathbf{X}_2^{n+1} = \mathbf{Y}_2^{n+1} - \mathbf{L}_{21}\mathbf{X}_1^{n+1}$.

4.4. *Complexity analysis*

According to Section 4.2 and [20], for the $\mathcal{H}^2$-representation $\mathbf{H} = \mathbf{V}_{\mathbf{H},r}\mathbf{S}_{\mathbf{H},r\times s}\mathbf{W}_{\mathbf{H},s}^{\mathrm{T}}$, we need to consider the storage of the coupling matrices $\mathbf{S}_{\mathbf{H},r\times s}$ and the coefficients $\alpha_{i,j}^{r_z,r}$ and $\beta_{j,i}^{s_z,s}$, where $r\times s \subset \mathcal{I}\times\mathcal{I}$. Let $P(l) = \{r\times s \subset \mathcal{I}\times\mathcal{I} : \mathrm{level}(r\times s) = l\}$ for the block tree $\mathcal{T}_{\mathcal{I}\times\mathcal{I}}$. It is easy to prove that $|P(l)| \le 2^l$. Hence, the required storage for all block coupling matrices $\mathbf{S}_{\mathbf{H},r\times s}$ amounts to

$$\sum_{r\times s\subset\mathcal{I}\times\mathcal{I}} k_r k_s = \sum_{l=0}^{p-1} 2^l k_r k_s \le \max_{r\times s\subset\mathcal{I}\times\mathcal{I}}\{k_r k_s\}\sum_{l=0}^{p-1} 2^l = \max_{r\times s\subset\mathcal{I}\times\mathcal{I}}\{k_r k_s\}(2^p - 1)\sim\mathcal{O}(N),$$

which implies that the storage needed for all coupling matrices is bounded by $\mathcal{O}(N)$. For the storage of the coefficients $\alpha_{i,j}^{r_z,r}$ and $\beta_{j,i}^{s_z,s}$, assuming that $k_l \le \tau_1(p-l)+\tau_2$, then we can see that the required storage can be bounded by

$$\sum_{r\times s\subset\mathcal{I}\times\mathcal{I}} k_l k_{l+1} = \sum_{l=0}^{p-1} 2^{l+1} k_l k_{l+1} \le \sum_{l=0}^{p-1} 2^{l+1}(\tau_1(p-l)+\tau_2)(\tau_1(p-l-1)+\tau_2)\sim\mathcal{O}(N).$$

From the aforementioned analysis, it can be seen that the total storage size of the $\mathcal{H}^2$-representation can be bounded by $\mathcal{O}(N)$. Moreover, the $\mathcal{H}^2$-based matrix-vector multiplication algorithm also has linear complexity, and the $\mathcal{H}^2$-LU factorization only requires $\mathcal{O}(N)$ operations [23].

5. NUMERICAL RESULTS AND DISCUSSIONS

To test the proposed $\mathcal{H}^2$-based method, we consider a 3D-IC structure, which consists of two active layers with through-silicon vias (TSVs) and microbumps connecting the layers shown in Figure 4. All numerical results are computed in C based on the hierarchical matrices library [28] and run on Linux server with a 2.4 GHz Intel Xeon Quad-Core CPU and 36 GB memory.

In order to construct an FE model, the *Gmsh* program can be used to generate the meshes [29]. To show the effectiveness of the $\mathcal{H}^2$-preconditioning, we need to demonstrate the linear complexity of the $\mathcal{H}^2$-LU factorization for the FE-based thermal matrices in terms of both memory space and CPU time.

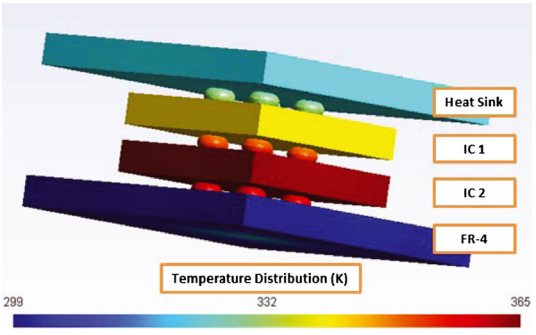

Figure 4. A 3D IC model with TSV.

DOI: 10.1002/cta

We discretize the 3D-IC structure into tetrahedral elements with four nodes in each element to generate 7239–825,875 unknowns for the thermal extraction. The simulation parameter η in Equation (6) is set to be one for transient simulation. By using the admissibility condition (6), we can construct an admissible block cluster tree $\mathcal{T}_{\mathcal{I}\times\mathcal{I}}$ for the transient thermal matrix **H** having the structure shown in Figure 5(a). The constructing procedure maps the admissible block cluster tree to an $\mathcal{H}^2$-matrix shown in Figure 5(b).

To show the performance of the proposed solver, we compare it with the known CSPARSE method [25], which represents the state-of-the-art of industry-strength sparse LU solver for general problems. The $\mathcal{H}^2$-based solver is a direct method for solving large linear systems. The $\mathcal{H}^2$-matrix uses a refined representation that employs a nested structure in order to reduce the storage requirement. We have shown that the storage requirement of $\mathcal{H}^2$-representation is of almost linear complexity. The CSPARSE method is also a direct solver for solving sparse linear systems. The CSPARSE method uses the compressed sparse row format, which is a popular, general-purpose sparse matrix representation, to explicitly store column indices and nonzero values in arrays column indices and data. Table I shows the comparison between the storage of $\mathcal{H}^2$-matrix and the storage for the CSPARSE method, as well as the comparison between the CPU time spent in $\mathcal{H}^2$-based arithmetics and the CPU time computed for the CSPARSE method. We can see from Table I that the required memory in the form of $\mathcal{H}^2$-matrix is much lesser than that in the form of the CSPARSE representation. If the size of the thermal matrix is large, the $\mathcal{H}^2$-LU factorization requires less CPU time compared with the CSPARSE LU factorization. The advantage of computation time becomes

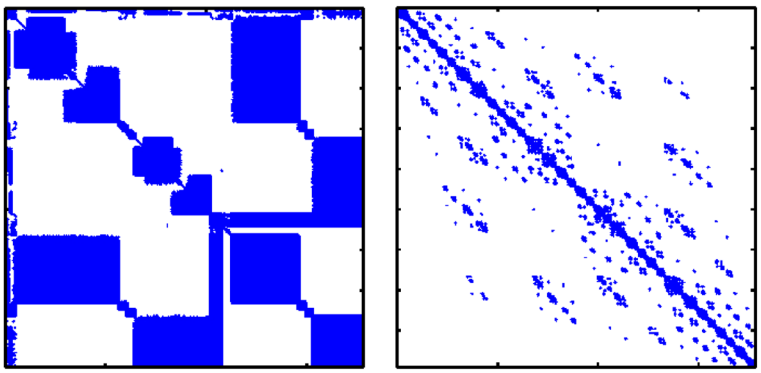

Figure 5. (a) The thermal matrix **H**. (b) The $\mathcal{H}^2$-representation.

Table I. Numerical results for the three dimensional integrated circuit model with through-silicon via.

	Memory space			CPU time			
No. of unknowns	$\mathcal{H}^2$-matrix (MB)	CSPARSE (MB)	Saved	$\mathcal{H}^2$-matrix (s)	CSPARSE (s)	Speedup	Error $\frac{\|\mathbf{L}_{\mathcal{H}^2}\mathbf{U}_{\mathcal{H}^2}-\mathbf{H}\|}{\|\mathbf{H}\|}$
7239	77.14	38.35	0.50	33.89	2.39	0.07	2.02e−8
12,300	175.43	97.78	0.56	82.42	8.75	0.11	5.54e−8
31,888	356.32	304.31	0.84	158.92	34.40	0.22	2.62e−8
49,687	494.38	554.8	1.12	205.63	75.54	0.37	5.82e−8
76,840	758.10	988.8	1.30	291.42	196.78	0.68	6.76e−9
124,344	1250.26	1871.9	1.44	502.75	493.05	0.98	6.49e−9
177,933	1851.67	2952.6	1.59	766.79	1089.74	1.42	5.23e−8
239,028	2591.40	4333.6	1.67	1091.75	1841.82	1.69	4.96e−8
354,099	4053.30	7109.8	1.75	1664.64	3838.58	2.31	2.66e−8
597,436	7355.99	13,474.2	1.84	2893.71	10,512.42	3.63	1.01e−7
825,875	11,345.39	Failed	N/A	4568.44	Failed	N/A	9.10e−7

DOI: 10.1002/cta

very obvious when the size of the thermal matrix is beyond 177,933 for LU factorization. The CSPARSE method requires twice as much storage as the $\mathcal{H}^2$-matrix method when the matrix size is beyond 597,436. Moreover, we find from our experiments that the direct LU factorization needs 30 times as much storage as the proposed method when the matrix size is 12,300, and the direct solver fails to work when the size is beyond 49,687.

We also demonstrate the complexity from Table I and the theoretical complexity by Figure 6(a) and Figure 6(b). It can be seen from these figures that the memory storage and CPU time for the $\mathcal{H}^2$-based arithmetics match very well with the theoretical analysis. For the whole space complexity analysis, because each thermal matrix **H** has the same structure for every time step, the total time for solving the system (5) has a linear storage complexity. From Figure 6, we can see that the proposed method has the obvious advantage in terms of memory space and computation time compared with the CSPARSE method.

For the 3D TSV model, Figure 7 shows the transient responses of both the proposed method and the CSPARSE method. Figure 7(a) shows the waveform simulation results computed by the two methods. Figure 7(b) shows the relative error of the proposed method over the CSPARSE method. Clearly, the $\mathcal{H}^2$-LU factorization also possesses high accuracy for the waveform simulation. In addition, it can be seen from Table I that the CSPARSE solver fails to work when the size of the thermal matrix is beyond

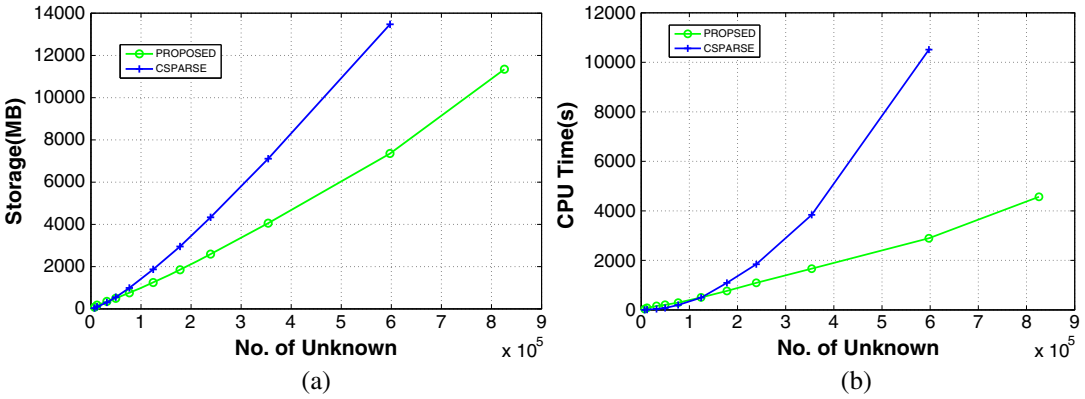

Figure 6. (a) Memory scalability comparison of LU factorization for the 3D IC model. (b) CPU time scalability comparison of LU factorization for the 3D IC model.

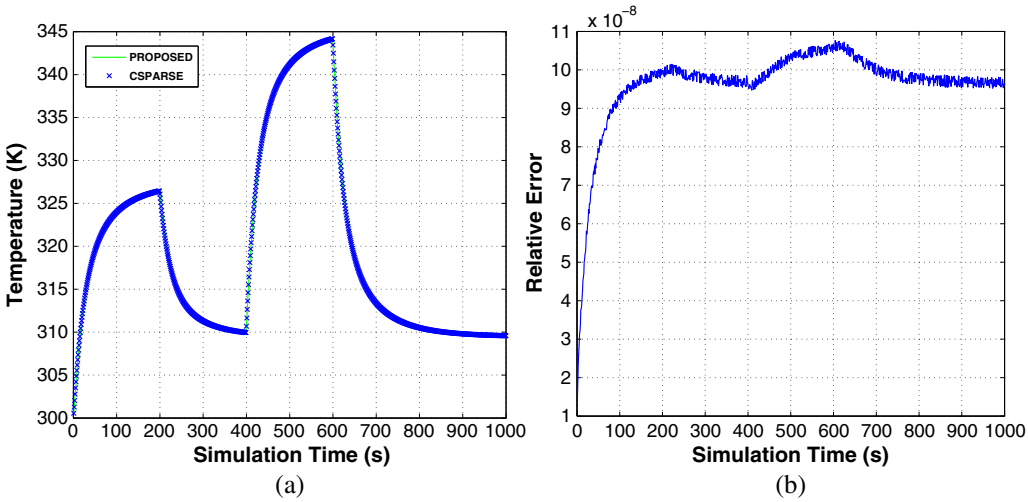

Figure 7. (a) Thermal transient responses by the $\mathcal{H}^2$ and CSPARSE methods. (b) Relative error of the $\mathcal{H}^2$ method over the CSPARSE method.

DOI: 10.1002/cta

825,875, but the $\mathcal{H}^2$-based solver can work effectively for the same matrix and even a matrix greater than that. From the simulation results, we can see that the $\mathcal{H}^2$-matrix method is more effective than the CSPARSE method for the simulation of the 3D TSV model.

In addition to the CSPARSE, we also did comparison with UMFPACK, which is a right-looking multifrontal solver [26]. Our numerical simulation results show that for the given thermal matrices in Table I, UMFPACK indeed is slightly faster than the $\mathcal{H}^2$-based solver. But the UMFPACK solver fails when the thermal matrix size is greater than 597,436 because of memory failure. This indeed shows the memory efficiency and scalability of the proposed method. UMFPACK is highly optimized LU solver, and it has very small overhead compared with the $\mathcal{H}^2$-based solver. But as the sizes of the problem increase, the proposed method will be faster because of its linear scalability and amortization of its overhead over large problem sizes. We remark that CSPARSE and UMFPACK fail to work on some large examples on the given workstation, we cannot perform the CPU time comparisons on larger examples and thus cannot demonstrate the potential speedup of the proposed method over those methods.

Another solver is the CHOLMOD [30], which is a Cholesky factorization solver. We did not perform the comparison as the proposed $\mathcal{H}^2$-based solver is the general unsymmetric solver. It is unfair to compare them directly. It was shown that the 3D ICs with microchannel-based liquid cooling technique can lead to nonsymmetric matrices as the microchannel is modeled as voltage-controlled current sources [31, 32]. Hence, CHOLMOD will not work for general thermal matrices.

In the following, we demonstrate the effectiveness of the proposed $\mathcal{H}^2$-based solver from a TSV array model [33]. A two-layer stack structure is shown in Figure 8(a), which represents part of a 3D stacked chip structure that can be meshed by the *Gmsh* program. The power sources are placed at the bottom of the structure where there is no TSV as shown in Figure 8(a). By using the *Gmsh* program, the tetrahedron meshes for the two-layer stack structure can be generated. Figure 8(b) shows a tetrahedron mesh structure for the 3D TSV array. For the numerical simulation, the two-layer chip structure is discretized into tetrahedral elements with four nodes in each element to generate 5452–886,154 unknowns. The comparison of the complexities of the $\mathcal{H}^2$-based LU factorization and the CSPARSE solver is shown in Table II. We can see from the table that less memory is required when the size of the thermal matrix exceeds $70,496 \times 70,496$. The CSPARSE solver failed to work for the $886,154 \times 886,154$ thermal matrix but the proposed $\mathcal{H}^2$-matrix method is still effective for this case. The complexities of the proposed solver from Table II are also demonstrated by Figure 9(a) and Figure 9(b). It can be seen from these figures that the complexities of the proposed $\mathcal{H}^2$-matrix method are almost linear.

The speedup of the proposed $\mathcal{H}^2$-matrix method over the LU-based method by the CSPARSE solver is not very impressive for small thermal examples. For small cases, the proposed method may be even slower. The reason is that the $\mathcal{H}^2$-matrix solver has pretty high upfront overheads as it needs to build the hierarchical matrix and the nested cluster basis first from the existing thermal matrices. The CSPARSE

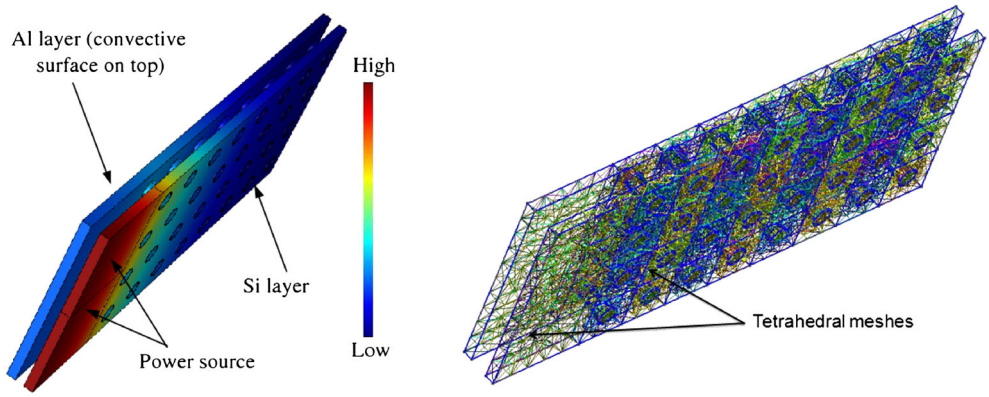

Figure 8. (a) Two-layer stack structure with a TSV array. (b) 3D mesh for the two-layer stack structure.

DOI: 10.1002/cta

Table II. Numerical results for the through-silicon via array model.

No. of unknowns	Memory space			CPU time			Error
	$\mathcal{H}^2$-matrix (MB)	CSPARSE (MB)	Saved	$\mathcal{H}^2$-matrix (s)	CSPARSE (s)	Speedup	$\frac{\|\mathbf{L}_{\mathcal{H}^2}\mathbf{U}_{\mathcal{H}^2}-\mathbf{H}\|}{\|\mathbf{H}\|}$
5452	37.51	11.91	0.32	6.48	1.18	0.18	1.21e−8
11,115	105.67	42.69	0.40	26.76	7.36	0.28	5.42e−8
22,439	235.05	127.62	0.54	83.81	26.35	0.31	4.73e−8
32,525	351.68	271.34	0.77	170.84	72.41	0.42	6.15e−8
70,496	585.12	649.86	1.12	201.95	136.83	0.68	2.27e−8
129,445	1178.39	1401.74	1.19	403.28	346.31	0.85	7.35e−9
215,850	2162.17	3285.73	1.52	1246.35	1451.04	1.16	5.74e−8
486,260	5734.42	10,466.84	1.83	2987.62	4591.42	1.54	3.72e−8
886,154	11,554.07	Failed	N/A	4775.49	Failed	N/A	2.75e−8

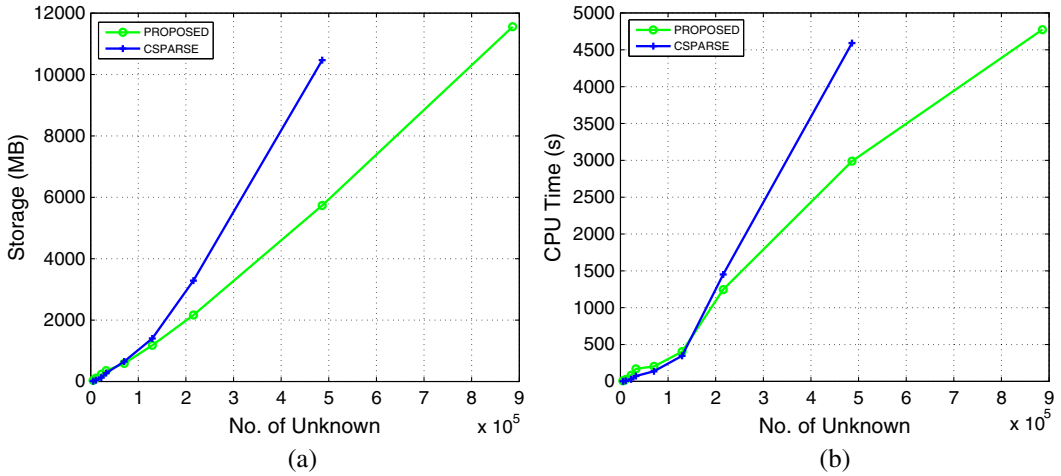

Figure 9. (a) Memory scalability comparison of LU factorization for the TSV array model. (b) CPU time scalability comparison of LU factorization for the TSV array model.

is well designed and industry-strength solver. The LU solver actually is very fast if the memory is not limited as demonstrated by recent power grid analysis contest, where all the wining solvers are LU-based solver [34, 35]. But for large examples, the proposed method starts to deliver increasing speedups over the LU solver as shown in Tables I and II. The CSPARSE solver cannot work for very large examples in our server with 32 GB memory. The key difference between the two methods is the growth rates of the CPU time and memory storage depending on the problem sizes, which makes the $\mathcal{H}^2$-matrix solver much more attractive for attacking very large problem sizes for limited computing resources.

6. CONCLUSION

In this article, we have proposed a new scalable transient thermal analysis technique, which shows *linear* time and storage complexity. The new technique is based on the recently proposed $\mathcal{H}^2$ mathematical framework, which exploits the data sparsity of many matrices from the FE-based solutions of PDEs. We showed and proved that $\mathcal{H}^2$ techniques can be used to efficiently solve the parabolic PDEs with controlled errors coming from the transient thermal analysis of ICs. The numerical results from 3D ICs demonstrate the linear scalability of the proposed method in terms of both memory footprint and CPU time. The comparison with existing product-quality LU solvers, CSPARSE and UMFPACK, on a number of 3D IC thermal matrices clearly shows the memory space advantage of the new method over these methods.

DOI: 10.1002/cta

ACKNOWLEDGEMENTS

This work is supported in part by NSF grant under no. CCF-1017090, in part by NSF grant under no. CCF-1255899, in part by Semiconductor Research Corporation grant under no. 2013-TJ-2417, in part by a 985 research fund from Shanghai Jiao Tong University, and in part by NSFC grant under no. 61404024.

REFERENCES

1. Manoj S, Yu H, Shang Y, Tan CS, Lim SK. Reliable 3D clock-tree synthesis considering nonlinear capacitive TSV model with electrical-thermal-mechanical coupling. *IEEE Transactions on Computer-Aid Design of Integrated Circuits and Systems* 2013; **32**:1734–1747.
2. Gunther S, Binns F, Carmean D, Hall J. Managing the impact of increasing microprocessor power consumption. *Intel Technology Journal* 2001; **5**:1–9.
3. Brooks D, Martonosi M. Dynamic thermal management for high-performance microprocessors. *Proceedings of International Symposium on High-Performance Computer Architecture* 2001:171–182.
4. Conte G. A general method for small-signal circuit analysis considering thermal effects. *International Journal of Circuit Theory and Applications* 1978; **6**:41–47.
5. Yu H, Shi Y, He L, Karnik T. Thermal via allocation for 3D ICs considering temporally and spatially variant thermal power. *IEEE Transactions on Very Large Scale Integration Systems* 2008; **16**:1609–1619.
6. Yu H, Ho J, He L. Allocating power ground vias in 3D ICs for simultaneous power and thermal integrity. *ACM Transactions on Design Automation of Electronic Systems* 2009; **14**:1–31.
7. Pedram M, Nazarian S. Thermal modeling, analysis, and management in VLSI circuits: principles and methods. *Proceedings of the IEEE* 2006:1487–1501.
8. Skadron K, Stan MR, Huang W, Velusamy S, Sankaranarayanan K, Tarjan D. Temperature-aware microarchitecture. *International Symposium on Computer Architecture* 2003:2–13.
9. Ozisik M. Finite Difference Methods in Heat Transfer. Taylor & Francis: Boca Raton, 1994.
10. Shen Y, Wong N, Lam EY, Koh CK. Finite difference schemes for heat conduction analysis in integrated circuit design and manufacturing. *International Journal of Circuit Theory and Applications* 2011; **39**:905–921.
11. Lewis R, Nithiarasu P, Seetharamu K. Fundamentals of the Finite Element Method for Heat and Fluid Flow. John Wiley & Son: New York, 2004.
12. Jurić-Grgić I, Lucić R, Bernadić A. Transient analysis of coupled non-uniform transmission line using finite element method. *International Journal of Circuit Theory and Applications*. doi: 10.1002/cta.2002
13. Davis T. Direct Methods for Sparse Linear Systems. SIAM: Philadelphia, 2006.
14. Hackbusch W. A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: introduction to $\mathcal{H}$-matrices. *Computing* 1999; **62**(2):89–108.
15. Grasedyck L, Hackbusch W. Construction and arithmetics of $\mathcal{H}$-matrices. *Computing* 2003; **70**(4):295–334.
16. Borm S, Grasedyck L, Hackbusch W. Hierarchical matrices. Lecture Notes 21 of the Max Planck Institute for Mathematics in the Sciences, 2003.
17. Liu H, Jiao D. Performance analysis of the $\mathcal{H}$-matrix-based fast direct solver for finite-element-based analysis of electromagnetic problems. *Antennas and Propagation Society International Symposium* 2009:1–4.
18. Li YC, Tan S, Yu T, Huang X, Wong N. Direct finite-element-based solver for 3D-IC thermal analysis via $\mathcal{H}$-matrix representation. *International Symposium on Quality Electronic Design* 2014:386–391.
19. Chen HB, Li YC, Tan S, Huang X, Wang H, Wong N. $\mathcal{H}$-matrix based finite-element-based thermal analysis for 3D ICs. Submitted to *ACM Transactions on Design Automation of Electronic Systems*, 2014.
20. Hackbusch W, Khoromskij B, Sauter S. On $\mathcal{H}^2$-matrices. Lecture Notes on Applied Mathematics, 2000.
21. Börm S. Construction of data-sparse $\mathcal{H}^2$-matrices by hierarchical compression. *SIAM Journal Scientific Computing* 2009; **31**(3):1820–1839.
22. Börm S. Approximation of solution operators of elliptic partial differential equations by $\mathcal{H}$ and $\mathcal{H}^2$-matrices. *Numerische Mathematik* 2010; **115**(2):165–193.
23. Chai W, Jiao D. $\mathcal{H}$- and $\mathcal{H}^2$-matrix-based fast integral-equation solvers for large-scale electromagnetic analysis. *IET Microwaves, Antennas & Propagation* 2010; **4**(10):1583–1596.
24. Chai W, Jiao D. An $\mathcal{H}^2$-matrix-based integral-equation solver of reduced complexity and controlled accuracy for solving electrodynamic problems. *IEEE Transactions on Antennas and Propagation* 2009; **57**(10):3147–3159.
25. CSPARSE: (Available from: http://www.cise.ufl.edu/research/sparse/CSparse/, 2014)
26. UMFPACK: (Available from:http://www.cise.ufl.edu/research/sparse/umfpack/, 2014)
27. Bergman T, Lavine A, Incropera F, DeWitt D. Fundamentals of Heat and Mass Transfer. John Wiley & Son: New York, 2011.
28. Hlib: (Available from:http://www.hlib.org, 2014)
29. Gmsh: (Available from:http://geuz.org/gmsh/, 2014)
30. Cholmod: (Available from:http://www.cise.ufl.edu/research/sparse/cholmod/, 2014)
31. IBM inter-layer: (Available from:http://www.zurich.ibm.com/st/cooling/integrated.html, 2014)
32. Sridhar AM, Vincenzi A, *et al*. 3D-ICE: fast compact transient thermal modeling for 3D-ICs with inter-tier liquid cooling. *Proceedings of International Conference on Computer Aided Design (ICCAD)* 2010:463–470.
33. Liu Z, Swarup S, Tan S, Chen HB, Wang H. Compact lateral thermal resistance model of TSVs for fast finite-difference based thermal analysis of 3D stacked ICs. To Appear in *IEEE Trans. on Computer-Aid Design of Integrated Circuits and Systems*, 2014.

DOI: 10.1002/cta

34. Yu T, Wong MDF. PGTSOLVER: an efficient solver for power grid transient analysis. *Proceedings of International Conference on Computer Aided Design (ICCAD)* 2012:647–652.
35. Yang J, Li Z, Cai Y, Zhuo Q. PowerRush: efficient transient simulation for power grid analysis. *Proceedings of International Conference on Computer Aided Design (ICCAD)* 2012:653–659.

DOI: 10.1002/cta