MANCHESTER 1824
The University of Manchester
The University of Manchester Research

The MTA: An Advanced and Versatile Thermal Simulator for Integrated Systems

DOI:
10.1109/TCAD.2018.2789729

Document Version
Accepted author manuscript

Link to publication record in Manchester Research Explorer

Citation for published version (APA):
Chen, Y-C., Ladenheim, S., Mihajlovic, M., & Pavlidis, V. (2018). The MTA: An Advanced and Versatile Thermal Simulator for Integrated Systems. *IEEE Transactions on Computer - Aided Design of Integrated Circuits and Systems* , *37*(12), 3123-3136. https://doi.org/10.1109/TCAD.2018.2789729

Published in:
IEEE Transactions on Computer - Aided Design of Integrated Circuits and Systems

Citing this paper
Please note that where the full-text provided on Manchester Research Explorer is the Author Accepted Manuscript or Proof version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version.

General rights
Copyright and moral rights for the publications made accessible in the Research Explorer are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Takedown policy
If you believe that this document breaches copyright please refer to the University of Manchester's Takedown Procedures [http://man.ac.uk/04Y6Bo] or contact uml.scholarlycommunications@manchester.ac.uk providing relevant details, so we can investigate your claim.

Download date:26. Aug. 2022

The MTA: An Advanced and Versatile Thermal Simulator for Integrated Systems

Scott Ladenheim, Yi-Chung Chen, Milan Mihajlović, and Vasilis F. Pavlidis

***Abstract*—Fast and accurate thermal analysis is crucial for determining the propagation of heat and tracking the formation of hot spots in integrated circuits (ICs). Existing academic thermal analysis tools primarily use compact models to accelerate thermal simulations but are limited to linear problems on relatively simple circuit geometries. The Manchester Thermal Analyzer (MTA) is a comprehensive tool that allows for fast and highly accurate linear and nonlinear thermal simulations of complex physical structures including the IC, the package, and the heatsink. The MTA is targeted for 2.5/3-D IC designs but also handles standard planar ICs. The MTA discretizes the heat equation in space using the finite element method and performs the time integration with unconditionally stable implicit time stepping methods. To improve the computational efficiency without sacrificing accuracy, the MTA features adaptive spatiotemporal refinement. The large-scale linear systems that arise during the simulation are solved with fast preconditioned Krylov subspace methods. The MTA supports the thermal analysis of realistic integrated systems and surpasses the computational abilities and performance of existing academic thermal simulators. For example, the simulation of a processor in a package attached to a heat sink, modeled by a computational grid consisting of over 3 million nodes, takes less than 3 minutes. The MTA is fully parallel and publicly available.*[*]**

***Index Terms*—Transient thermal analysis, Integrated circuits, Finite element method, Adaptive spatiotemporal refinement, Krylov solvers, Algebraic multigrid**

I. INTRODUCTION

Thermal issues created by higher power densities in sub-90 nm advanced CMOS technologies lead to performance degradation and excessive leakage currents [1]. Device technologies, such as FinFET and SOI, have been introduced to underpin further scaling of silicon technology by suppressing leakage currents. However, these technologies can be more sensitive to thermal issues due to self-heating and the low thermal conductivity of the materials involved [2]. Circuits of "post-Dennardian" scaling also face challenges of exponentially increasing power densities [3], [4]. Emerging 3-D technologies (i.e., multi-tier systems with vertical interconnections) also suffer from thermal issues since 3-D integrated circuits (IC) involve complex heat dissipation paths and have more active regions than a single tier in a package [1], [5], [6].

S. Ladenheim, Y.-C. Chen, M. Mihajlović and V.F. Pavlidis are with the School of Computer Science, University of Manchester, Manchester, UK. (e-mail: {scott.ladenheim, yi-chung.chen, milan.mihajlovic, vasileios.pavlidis}@manchester.ac.uk)

[*]The MTA is available to download at:
https://staffnet.cs.manchester.ac.uk/acso/thermal-analyzer/

To account for these thermal effects, detailed thermal analysis is required at several stages of the design process [7]. Academic thermal analysis tools, for instance [8]–[13], were previously introduced to compute the temperature profile of an IC. However, obtaining fast and accurate thermal simulations, especially transient, remains a challenging (high performance) computing problem which requires novel algorithms and approaches. Tools, such as [10], [11], use compact thermal models to simplify the heat flow model and reduce the size of the computational problem. However, such tools suffer from reduced accuracy and have limited scope in terms of the circuit and surrounding package features they can model, such as TSVs or heat sinks. Many of these tools use direct methods to compute the solutions of the linear systems arising from the problem discretization which hinders the scalability of the solver. This approach is ill-suited to solve the large-scale linear systems that result from the discretization of the 3-D structures in an IC. Furthermore, the numerical methods used by existing tools to compute the thermal profile of an IC are often either low order accurate or possess numerical stability constraints. Such limitations prohibit the analysis of complex micrometer scale features of ICs and do not allow for useful computational tradeoffs between the architecture level simulators and the numerical methods used for the thermal analysis. In addition, temperature-dependent thermal parameters are not included in the utilized models.

The Manchester Thermal Analyzer (MTA) is a new academic thermal analysis tool for circuits and their package that has been developed to address these limitations. Based only on the floorplan information of the circuit, the MTA initially generates the corresponding computational grid required by the thermal simulation. Power information obtained from an external power simulator is an input to the MTA which then computes the temperature profile of the circuit. The MTA deploys a unified methodology that involves advanced spatiotemporal refinement techniques and fast preconditioned iterative solvers that can be tuned to deliver efficient and highly accurate thermal simulations.

The contributions of this paper to the problem of thermal analysis are:

- An advanced and versatile tool for computing highly accurate steady state and transient thermal profiles of ICs and their packages using either a linear or a nonlinear model with temperature-dependent material parameters.
- A fast mesh generator for the discretization of 3-D IC geometries with built-in local refinement capabilities prior to the simulation.
- Fully adaptive spatiotemporal refinement and model

parameter selection based on established error analysis theory to compute highly accurate temperature profiles at a near optimal computational cost.
- The shared memory parallelization capability for multicore architectures.

With these capabalities, the MTA is able to thermally analyze a modern processor at the floorplan level in an LGA package attached to a heat sink with 72 fins, resulting in a mesh of over 3M nodes in under 3 minutes, of which the actual computation time is 40 seconds. Extensive numerical simulations demonstrate that the MTA provides fast and accurate linear and nonlinear thermal analyses for more complex geometries than existing academic tools.

The paper is organized as follows. Section II discusses related work and the motivation for the MTA. This section contains an overview of existing thermal analysis tools. Section III presents the new and extended version of the MTA, which includes a description of the solution methodology and lists the main features of the tool. Section IV covers the computational and implementation details of the MTA. Section V presents results of both linear and nonlinear thermal simulations and comparisons with existing tools in terms of both speed and accuracy. Section VI offers some concluding remarks.

II. Related Work and Motivation

There are a number of existing thermal analysis tools for ICs, see, e.g., [8]–[13]. Tools, such as Hotpost [10] and 3D-ICE [11], invoke the electro-thermal duality and utilize compact models to approximate the heat flow within the IC. This approach approximates the underlying physics of the problem by an analogous electrical circuit analysis. In these tools, the linear systems arising from the circuit equations are solved by direct methods [14], which become prohibitively expensive as the size of the linear system increases. For example, Hotspot [10], which uses the SuperLU solver [15], computes a steady-state temperature on a circuit discretized with 250K cuboids in 40 seconds, whereas a problem with 1M cuboids takes over 6 minutes.

Other tools, such as ICTherm [12], [13], discretize the heat equation using finite difference methods and solve the resulting linear systems using operator splitting methods. The solution time for these methods scales linearly with the problem size, but their use is limited to linear problems and structures discretized by tensor product grids, which prevents the use of adaptive spatial refinement. In all of these tools, transient temperature profiles are computed using a time integration method. The time integration methods are either explicit [10], [11], [16], semi-implicit [12], or low order implicit [17] and in most cases only support constant time step sizes. Stability requirements of explicit methods are expressed by the CFL condition [18], which bounds the maximum allowable time step size by the size of the spatial discretization parameter. The micrometer (or nanometer) scale features of an IC require fine spatial grids, which place a severe restriction on the time step size and increase the computational cost.

Existing state-of-the-art tools are effective for a limited range of problems, specifically, simple floorplan geometries that lead to small-scale linear systems that do not change during the simulation, i.e., the spatial and time discretizations remain fixed. Practically, these tools model 2.5-D geometries, where each layer of material cannot be freely discretized in the vertical direction. Limited support or oversimplified models are also provided for the surrounding structures of an IC, such as the package and heat sink, which hinders design and package space exploration.

The MTA differs from most of the previously mentioned tools by using the finite element method (FEM) [19] to discretize the heat equation directly. In addition, the resulting linear systems are solved with fast preconditioned iterative methods, specifically, multigrid preconditioned Krylov subspace methods [20], [21]. The MTA utilizes the FEM with varying order of the polynomial interpolation and non-damping time integration methods. To improve computational speed and to fully include the adaptive refinement features supported by previous thermal analysis tools, the MTA supports simultaneous adaptive time integration and adaptive spatial refinement. Tools, such as [8], [22], have previously employed simultaneous spatiotemporal adaptivity but use explicit time integration methods.

In [8], both the grid and time step size are adaptively refined and each element in the discretization can have a different time step size. However, this approach uses explicit methods which are still constrained by the CFL condition. In [22], frequency-domain methods are used to integrate over longer time intervals to avoid long simulations that take many small time steps. In contrast, the MTA uses unconditionally stable and non-dissipative adaptive synchronous time integration methods in conjunction with adaptive spatial refinement. As is shown, by enabling these features simultaneously, the MTA produces as accurate thermal profiles over long simulation intervals with comparable or improved speed compared to the previous tools and fixed discretization approaches. At the same time, the MTA removes many of the inherent limitations of existing academic tools while providing a more versatile tool capable of simulating more complex system structures.

In all of the aforementioned tools, the material properties are considered constant yielding linear heat flow models. This assumption has been shown to considerably affect the accuracy even for the simple case study of steady-state analysis for a silicon cube [23]. Recognizing the requirement to provide highly accurate thermal analysis over wide temperature ranges, the MTA incorporates, to the best of the authors' knowledge for the first time, transient nonlinear thermal analysis to model complex circuit geometries. The MTA also provides a novel and advanced mesh generation process that rapidly produces finite element meshes for complex IC and package geometries. This comprehensive set of computational features make the MTA well-suited for the thermal analysis of modern integrated systems characterized by increasing complexity and heterogeneity.

III. Overview of the MTA

Based only on a user-supplied XML file containing the thermal parameters and coordinates of the structures within

the integrated system, the MTA automatically generates the finite element mesh and numerically solves the (nonlinear) heat equation. The power density dissipated in the active components of the circuit is supplied from an external power trace file that is generated either by architecture level simulators with a power modeling framework such as [24] or commercial power analysis tools. Further details on the XML and power trace files are given in [25]. In the remainder of this section, the governing heat equation is reviewed, followed by the details of the mesh generation process, the spatial discretization, and the time integration methods. The adaptive spatiotemporal refinement features are also summarized. A discussion on the effects of spatial and temporal discretization errors to the accuracy and computational efficiency of the simulation is also presented. Lastly, the utilized preconditioned iterative solvers are discussed. A flow chart of the simulation process of the MTA is given in Figure 1.

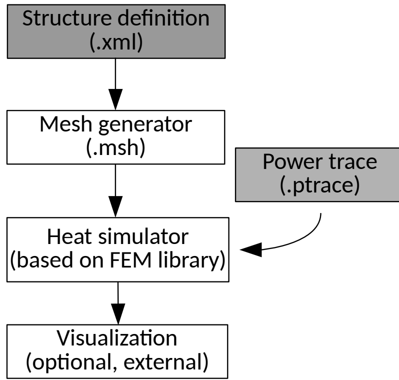


Fig. 1. The MTA flow. User-supplied input files are highlighted in grey.

A. Nonlinear Heat Transfer Model

The flow of heat in an IC is governed by the following nonlinear initial boundary value problem

$$\rho C(u)\frac{\partial u(\mathbf{x},t)}{\partial t} - \nabla\cdot(\kappa(u)\nabla u(\mathbf{x},t)) = f(u,\mathbf{x},t) \text{ in } \Omega\times[0,T], \tag{1a}$$

$$(\kappa(u)\nabla u)\cdot\mathbf{n} = \eta(u_a - u) \text{ on } \partial\Omega\times[0,T], \tag{1b}$$

$$u(\mathbf{x},0) = u_0 \text{ in } \bar{\Omega} = \Omega\cup\partial\Omega. \tag{1c}$$

The function $u(\mathbf{x},t)$ denotes the temperature (in [K]) at time t and a spatial point $\mathbf{x} = (x,y,z)$ in the IC domain $\Omega\subset\mathbb{R}^3$. The physical parameters are the material density ρ $[kg/m^3]$, the specific heat $C(u)$ $[J/kgK]$, the thermal conductivity $\kappa(u)$ $[W/mK]$, and the thermal transmissivity η $[W/m^2K]$ at the boundaries. The nonlinearity in the problem arises from the temperature dependence of the specific heat and thermal conductivity. Note that $c_V(u) = \rho C(u)$ is the volumetric specific heat.

The function $f(u,\mathbf{x},t)$ $[W/m^3]$ is the power density dissipated by the active layer(s) of the system. The power density is comprised of two parts, the dynamic power $f_1(\mathbf{x},t)$ and the leakage power $f_2(u,\mathbf{x},t)$ so that $f(u,\mathbf{x},t) = f_1(\mathbf{x},t) + f_2(u,\mathbf{x},t)$. The leakage power has an exponential dependence on the temperature, i.e., $f_2(u,\mathbf{x},t) = \alpha e^{\beta(u-u_b)}$, where α $[W/m^3]$, β (a dimensionless parameter), and the base temperature u_b depend on the architecture [26]–[28].

The Robin boundary condition described by (1b) represents Newton's law of cooling, i.e., the heat flux at the boundary is proportional (with constant heat transfer coefficient η) to the difference between the ambient temperature u_a and the temperature u at the boundary $\partial\Omega$. The initial temperature of the circuit in (1c) is denoted by u_0. The values of C and κ also vary across the circuit, since different materials have different thermal properties. A linear heat flow model is obtained by setting the thermal conductivity and specific heat to constant values in (1a). In the MTA, the initial boundary value problem (1) is solved numerically using the FEM and implicit time integration methods.

B. Mesh Generation

The required first step in the FEM is the generation of the computational mesh, i.e., the subdivision of the IC and package into smaller hexahedral (cuboid) elements with a characteristic length h referred to as the mesh width. The MTA provides a built-in and fully automatic 3-D mesh generator, which creates the mesh by parsing the coordinates of the IC components from the XML file. The developed mesh generator is extremely fast and efficient with the ability to create meshes containing millions of nodes in seconds.

To ensure mesh quality, the mesh generation routines offer refinement capabilities prior to the simulation. For example, the geometry of specific structures can produce coarse initial meshes that contain stretched elements. Other structures can produce meshes with reentrant corners at overhanging boundary surfaces. Without additional local mesh refinement, such meshes can lead to thermal profiles with low accuracy, producing, for example, an unphysical damping of the temperature. To avoid these computational issues, the MTA supports local mesh refinement for such structures before performing the thermal simulation.

C. FEM Discretization

The FEM discretization of (1) results in a system of initial value problems (IVP) of the form

$$M(\boldsymbol{u}_h)\dot{\boldsymbol{u}}_h + K(\boldsymbol{u}_h)\boldsymbol{u}_h = \mathbf{f}_h(\mathbf{u}_h), \tag{2}$$

or, equivalently,

$$M(\boldsymbol{u}_h)\dot{\boldsymbol{u}}_h + \underbrace{K(\boldsymbol{u}_h)\boldsymbol{u}_h - \mathbf{f}_h(\mathbf{u}_h)}_{F(\boldsymbol{u}_h)} = 0. \tag{3}$$

The solution (temperature) and power density vectors are denoted by $\boldsymbol{u}_h, \mathbf{f}(\mathbf{u}_h)_h \in \mathbb{R}^n$, respectively. The (nonlinear) mass and stiffness matrices are denoted by $M(\boldsymbol{u}_h), K(\boldsymbol{u}_h) \in \mathbb{R}^{n\times n}$. For further details of the discretization process see [25]. The length n (dimension) of the above vectors (matrices) is equal to the number of unknown nodal temperatures and is commonly referred to as the number of degrees of freedom (DOF). Note that as the mesh is refined ($h\to 0$), the discrete problem size increases ($n\to\infty$), and the

discrete solution tends to the continuous exact solution at the asymptotic rate. The asymptotic convergence of the solution and the relation to the computational efficiency of the thermal simulator are discussed further in Section III-F.

Transient thermal simulations are computed by solving (2) using a time integration method. The MTA uses unconditionally stable implicit time integration methods [18] and the specific methods offered by the tool are described in Section III-E. Steady-state thermal simulations are recovered by setting $\dot{\boldsymbol{u}}_h = 0$ in (2) and solving $K(\boldsymbol{u}_h)\boldsymbol{u}_h = \mathbf{f}_h(\mathbf{u}_h)$.

The computational cost of assembling the system matrices in (2) and the solution of the underlying linear systems increases with the dimension n. It is therefore essential that the assembly routines have optimal computational cost and to minimize the required number of assemblies during the simulation. Efficient and scalable parallelization of the assembly and solve routines is also crucial. For linear problems where the spatial and time discretization parameters are fixed, the assembly of these matrices is required only once. For nonlinear problems, these matrices must be constructed at each Newton iteration.

D. Adaptive Spatial Refinement

The temperature can vary significantly across the area of a circuit due to highly localized heat sources and the different properties of the constituent materials. Capturing this behavior accurately while maintaining the overall computational efficiency requires locally fine meshes. To achieve the required level of accuracy with uniformly refined meshes would lead to unnecessarily large computational times. In order to minimize the computational effort without sacrificing accuracy, the mesh is adaptively refined only in parts of the domain with significant variations (large gradients) in temperature. Adaptively refined solutions achieve the same asymptotic level of accuracy with more economical spatial discretizations. The adaptive mesh refinement (AMR) strategy in the MTA is based on computed a-posteriori error estimates.

The elements of the computational mesh are marked for refinement based on computed error estimates of the solution. In the MTA, these estimates are computed by the error estimator [29]. This estimator computes an approximation of the error in each element by averaging the jumps in the solution gradients at the faces of the adjacent elements. The refinement/merging of the elements is triggered based on user-specified tolerances described in IV-C. The effect of changing these tolerances on the speed and accuracy of the simulation is presented in Section V-B.

E. Adaptive Time Integration Methods

For transient problems, the MTA solves equation (3) using implicit time integration methods. When the time step size is fixed, the time interval $[0, T]$ is partitioned into N subintervals such that $t_0 = 0$ and $t_k = k\Delta t$ for $k = 1, \ldots, N$, with $\Delta t = T/N$. The discrete solution and right hand side vector at time t_k are denoted by $\mathbf{u}_h^k$ and $\mathbf{f}_h^k$, respectively.

Implicit time integration methods replace the time derivative in (3) by a finite difference approximation, i.e., $\dot{\mathbf{u}}_h \approx D\mathbf{u}_h^{k+1}$, for some difference operator D. Then the following nonlinear equation is solved for $\mathbf{u}_h^{k+1}$

$$M\left(\mathbf{u}_h^{k+1}\right) D\mathbf{u}_h^{k+1} + F\left(\mathbf{u}_h^{k+1}\right) = 0. \tag{4}$$

The MTA supports the following methods:

- the 1st order backward Euler (BE) method,
- the 2nd order backward differentiation formula (BDF2),
- the 2nd order trapezoidal rule (TR) and the stabilized version of this method [30],
- the 2nd order implicit midpoint rule (IMR).

For an overview of these methods, see, e.g., [18].

The BDF methods, both the first order BE and second order BDF2, are included as they have been previously used [10], [17]. The main drawback is their dissipative nature, making them unsuitable for accurate integration over long time materials. In contrast, the TR and the IMR methods do not introduce spurious damping into the computed solution and accurately reproduce the energy of the system (the geometric integration property [31]). These methods should be used for the cases when the integration over long time intervals with many subintervals, i.e., when $\mathcal{O}(N) = 1000$. The TR method is appropriate for linear models, while the IMR should be used for the nonlinear model.

Each time integration method implemented in the MTA supports adaptive time step selection of Δt. This increases the computational efficiency of transient thermal simulations whilst controlling the temporal error. The new time step size is determined based on the computed local truncation error (LTE) estimate of the implicit method. This is accomplished using predictor-corrector methods [31]. These methods compute an LTE estimate to sufficient asymptotic accuracy based on the computed predictor (explicit method) solution and corrector (implicit method) solution and the asymptotic order of convergence for both of these methods should be equal. Let $\mathbf{e}$ denote the local truncation error (LTE) estimate [18] and ε_t a user-prescribed tolerance. A standard heuristic for computing the adaptive time step Δt_{n+1} of a method of order q is

$$\Delta t_{n+1} = \Delta t_n \left(\frac{\varepsilon_t}{\|\mathbf{e}\|_2}\right)^{\frac{1}{q+1}}, \tag{5}$$

where $\|\mathbf{e}\|_2$ is the standard Euclidean norm and Δt_n is the previous time step.

F. Error Analysis

Utilizing theoretical error analysis results that exist for the FEM [19] and the IVP solvers [18] is a key advantage of MTA's methodology compared to other tools that are based on the electrothermal duality. The computed FEM solution $\mathbf{u}_h^{k+1}$ introduces a certain amount of error with respect to the exact solution $\mathbf{u}$ of (1). The contributions to this error are a combination of the spatial and temporal discretization errors. As both the mesh width and time step size are refined, i.e., $h,\ \Delta t \to 0$, this error decreases at a specific asymptotic rate related to the order of the discretization methods used and in the limit $\mathbf{u}_h^{k+1} \to \mathbf{u}$. Note that for meshes consisting of different sized volume elements, h is identified as the maximum element length, width, or height.

The global error in the solution of (1)-(3) behaves as

$$\|\mathbf{u} - \mathbf{u}_h\| = \mathcal{O}\left(h^p + \Delta t^q\right), \quad (6)$$

where p and q are the asymptotic convergence orders of the space and time discretization methods [18], [19]. The order p in the FEM is determined by the order of the basis functions on the elements. The order q of the time integration method is based on the order of the LTE. For numerical schemes using equal order methods ($p = q$), it is advisable to set $\Delta t = \mathcal{O}(h)$. Otherwise, either the spatial or temporal error dominates the solution and computational effort is wasted.

Based on these error considerations external architecture simulators used to produce the power vector $\mathbf{f}_h^{k+1}$ should be reconfigured based on the order of the grid resolution used in thermal simulations. If the grid resolution is coarse, for instance, on the order of tens or hundreds of micrometers, configuring a power simulator to output power at the typical sub-microsecond time rate is computationally wasteful. This behavior can be exploited to accelerate the overall design process since the external architecture simulators produce smaller power trace files. Alternatively, if the external simulators cannot be optimally reconfigured in this way, the thermal simulator should allow for the tuning of the mesh width h and the time step Δt. The MTA supports the tuning of these parameters in order to balance the error with the computational effort. The interplay between the spatial and temporal errors and how to set h and Δt in the MTA for computationally efficient thermal simulations is described in Section V.

G. Solvers

At each time step t_{k+1}, a system of nonlinear algebraic equations (4) is solved with Newton's method. Newton's method starts from the initial guess $\mathbf{u}_h^{k+1,0}$ and computes the approximate solution $\mathbf{u}_h^{k+1,\ell}$ for $\ell = 1, 2...$ as

$$J\left(\mathbf{u}_h^{k+1,\ell-1}\right)\delta\mathbf{u}_h^{k+1,\ell-1} = -\mathbf{r}\left(\mathbf{u}_h^{k+1,\ell-1}\right), \quad (7)$$

$$\mathbf{u}_h^{k+1,\ell} = \mathbf{u}_h^{k+1,\ell-1} + \delta\mathbf{u}_h^{k+1,\ell-1}, \quad (8)$$

where

$$J\left(\mathbf{u}_h^{k+1}\right) = \frac{\partial\mathbf{r}\left(\mathbf{u}_h^{k+1}\right)}{\partial\mathbf{u}_h^{k+1}}, \quad (9)$$

$$\mathbf{r}\left(\mathbf{u}_h^{k+1}\right) = M\left(\mathbf{u}_h^{k+1}\right)D\mathbf{u}_h^{k+1} + F\left(\mathbf{u}_h^{k+1}\right), \quad (10)$$

are the Jacobian matrix and the residual vector at iteration $\ell - 1$, respectively. Note that in (9)-(10) the iteration index is dropped for notational convenience. At each Newton iteration ℓ, a linear system is solved for the correction $\delta\mathbf{u}_h^{k+1,\ell-1}$, which is used to update the solution in (8). The stopping criterion for the Newton iteration adopted in the MTA is satisfied when the relative norm of the residual in (10) is less than a user-prescribed tolerance ε_N, which is set to 10^{-6} by default. Larger values of ε_N lead to faster simulations with fewer Newton iterations required but less accurate solutions. For linear problems, Newton's method converges in a single iteration.

In thermal simulations, the solution of the system of equations in (9) is typically a computational bottleneck, requiring the use of fast solvers. Since the Jacobian matrices that arise from the FE discretization of (1) are large, sparse, and symmetric positive definite, the MTA uses preconditioned Krylov subspace iterative solvers [21]; specifically, the conjugate gradient (CG) method [32] with the BoomerAMG algebraic multigrid (AMG) preconditioner [20].

The advantage of the CG method is the use of computationally inexpensive sparse matrix-vector products and a three-term recurrence to create the orthogonal Krylov basis (i.e., the Lanczos method [33]). Thus, the CG algorithm has fixed storage costs, making this technique more suitable for solving very large sparse linear systems that arise from the discretization of 3-D partial differential equations than sparse direct methods based on matrix factorizations. Using iterative solvers, the MTA reduces transient simulation times by two orders of magnitude compared to simulators that use direct methods, such as Hotspot [10]; see Section V-A. The implemented solvers in the MTA use the open-source software PETSc [34], [35].

IV. Computational Details

This section discusses key computational details of the MTA pertaining to the implementation, the choice of the thermal parameters, the operation modes, and the adaptive spatiotemporal refinement parameters.

A. Implementation Details

The numerical solution of the heat equation (1) in the current version of the simulator is implemented with the open-source finite element library deal.II [36], [37] for improved computational speed and parallel efficiency. As a result of this new implementation, the MTA supports larger and faster thermal simulations compared to [25].

The power dissipated by the circuit is supplied to the MTA from an external power trace file. This file contains the power dissipated to the active layer(s) of the circuit at a uniform time interval of size Δt_a, referred to as the atomic time step. The power information from this file is used to assemble the power density vector $\mathbf{f}_h^{k+1}$. When the time step of the simulation $\Delta t = \Delta t_a$, the power vectors are simply read in from the power trace file. However, when $\Delta t > \Delta t_a$, the power at time t_{k+1} is taken as the average power over the number N_a of atomic subintervals contained in $[t_k, t_{k+1}]$

$$\mathbf{f}_h^{k+1} = \frac{1}{N_a}\sum_{i=0}^{N_a}\mathbf{f}_h^{k+\frac{i}{N_a}}, \quad (11)$$

where $N_a = (t_{k+1} - t_k)/\Delta t_a$. Note that this implies that the time steps in the MTA are integer multiples of Δt_a. For adaptive time stepping methods the new time step Δt_{n+1} computed in (5) is rounded down to the nearest lower integer multiple of Δt_a. The justification for averaging the power is that there is a significant gap in the time scales of the circuit's switching activity and the thermal diffusion times of the materials involved.

B. Temperature-Dependent Thermal Parameters

The specific heat and thermal conductivity of the different materials used in the MTA are assumed to depend linearly on the temperature, i.e., $c(u) = c_0 + c_1 u$ and $\kappa(u) = k_0 + k_1 u$. Note that a linear heat flow model is recovered by setting $c_1 = k_1 = 0$. The coefficients c_0, k_0, and c_1, k_1, are determined by linearly interpolating the data from [38], [39] over the typical temperature range of an IC (e.g., 300-450 K). This represents a sufficiently good approximation over this temperature range. However, if required, the model can be improved by using a higher order interpolation. In the nonlinear thermal simulation presented in Section V-C, the thermal parameters of two silicon tiers are considered to be temperature-dependent. The nonlinear parameter values of silicon are listed in Table I.

TABLE I
NONLINEAR PARAMETER VALUES OF SILICON.

element	c_0	c_1	k_0	k_1
Silicon	$1.05 \cdot 10^6$	$2.03 \cdot 10^3$	300	-0.5

C. Spatiotemporal Refinement Parameters

For improved computational efficiency, the MTA supports fully adaptive spatiotemporal refinement. Utilizing several combinations of spatiotemporal adaptivity, the MTA operates in four different modes as listed in Table II.

TABLE II
SPATIOTEMPORAL CHARACTERISTICS OF THE MTA OPERATION MODES.

	fixed h	adaptive h
fixed Δt	M1	M3
adaptive Δt	M2	M4

The different modes available in the MTA provide more flexibility in tuning thermal analyses than any other academic thermal simulator. The use of these features requires setting specific parameters, which affects both the speed and accuracy of the simulation.

The mode M1 assumes fixed spatiotemporal discretization parameters through the simulation (note that Δt can be an integer multiple of Δt_a). In the case of a linear model, the Jacobian matrix in (9) needs to be assembled only once and can be reused throughout the entire transient simulation. Similarly, the coarsening phase of the AMG preconditioner needs to be computed once and can be reused. When the spatial discretization is fixed and the time step size varies (M2), similar computational savings can be achieved since the formation of the Jacobian only involves adding the mass matrix to the scaled stiffness matrix. Every time the spatial discretization changes in modes M3 and M4, the Jacobian matrix needs to be reassembled. For the nonlinear model, the Jacobian is assembled at each Newton iteration.

For adaptive time integration, the key parameter is the user-prescribed LTE tolerance ε_t. Decreasing ε_t leads to an increase in the number of time steps, but the temporal accuracy of the solution is improved. When ε_t is reduced by an order of magnitude, the number of time steps increase roughly by a factor $10^{1/1+q}$. The choice of ε_t is problem dependent, but even using relatively strict tolerances, i.e., $\varepsilon_t = \mathcal{O}(10^{-5})$ or smaller, decreases the simulation time by an order of magnitude compared to simulations with a fixed atomic time step size (see Section V-B). The optimal choice of ε_t is also influenced by the mesh width. For a time stepping method of order q, the LTE of the method is of order $q+1$. Thus, for a fixed mesh width h, it is suggested to set $\varepsilon_t = \mathcal{O}(h^{q+1})$.

For AMR, the key parameters are the AMR frequency, and the percentage of elements with the largest/smallest error estimates to be refined/merged, denoted by τ_r and τ_m, respectively. The parameters $0 < \tau_r, \tau_m < 1$ influence how many elements are marked for refinement/merging and thus control the reduction of the error in the adaptively refined solution. The refinement strategy is based on Dörfler marking [40]. The set of elements to be refined in the mesh is the smallest subset of hexahedrons whose error sum is larger than the product of τ_r and the total error estimate. The set of elements to be merged is the smallest subset of hexahedrons whose error sum is smaller than the product of τ_m and the total error. Uniform refinement with no merging is achieved by setting $\tau_r = 1$ and $\tau_m = 0$.

AMR is computationally expensive, requiring the re-assembly of the Jacobian matrix on the refined grid in addition to redistributing the DOFs and computing a new coarsening scheme for the AMG preconditioner. As a result, tuning the AMR frequency parameter is important to balance the accuracy and speed of the simulation. Note that fewer refinements during the simulation lead to shorter execution times, but the thermal profiles generally have slightly lower accuracy. The choice of these spatiotemporal refinement parameters is problem dependent and insights are offered in Section V-B related to the tradeoffs between the speed and accuracy of thermal simulations in each mode.

V. RESULTS

In this section, the accuracy and speed of the MTA are validated together with experiments that demonstrate the full range of the simulator's capabilities. In Section V-A, the accuracy of the MTA is verified with the proprietary multi-physics software COMSOL [41]. The simulation speed of the MTA is then compared to Hotspot version 6.0 [10]. Both comparisons are performed on a basic benchmark problem. These tests illustrate both the accuracy and favorable execution times of thermal simulations using the MTA. In addition, the effect of varying h and Δt on the global error and how to choose these parameters for efficient computations is presented. In Section V-B, the same benchmark problem is used to demonstrate the computational benefits of enabling the spatiotemporal refinement features of the MTA and to provide guidelines for the choice of the spatiotemporal parameters. In Section V-C, nonlinear thermal simulations are presented for a slightly more complex geometry. Lastly, in Section V-D, results from a packaged circuit, the LGA benchmark [42], are shown to demonstrate the parallel capabilities of the MTA for a more complex scenario. All of the experiments are performed on a PC with an Intel i7 4790 processor, 32 GB DRAM, and the CentOS 7 operating system.

A. Validation with COMSOL and Hotspot

The accuracy of the MTA is validated through a comparison with the proprietary software COMSOL [41], and the state-of-the-art academic simulator Hotspot [10]. A steady-state and transient thermal simulation are considered on a 3-D benchmark system. This system is chosen to match Hotspot's modeling capabilities in terms of geometries and structures. In addition, this benchmark serves to demonstrate the achievable computational gains needed for more complicated problems. The benchmark structure is a circuit consisting of two silicon tiers attached to a heat spreader and heat sink as depicted in Figure 2. Note that solid arrows at the boundary indicate the Robin boundary condition in (1b) and dashed arrows indicate an adiabatic boundary ($\eta = 0$).

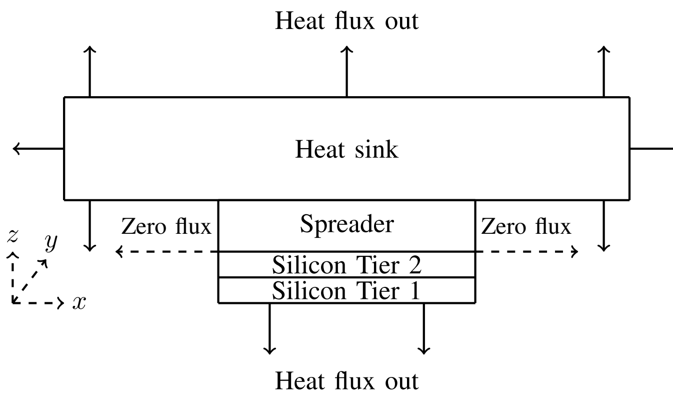


Fig. 2. Vertical cross-section of the basic benchmark circuit and package.

The dimensions are $10\ mm \times 10\ mm \times 0.5\ mm$ for the silicon tiers, $10\ mm \times 10\ mm \times 1\ mm$ for the heat spreader, and $50\ mm \times 50\ mm \times 10\ mm$ for the heat sink. Tier 2 dissipates 20 W throughout the entire simulation while Tier 1 is inactive. The parameters used to configure the linear thermal simulations are summarized in Table III, where the subscript T_i (HS) indicates a parameter of the silicon tier at room temperature (heat spreader or sink). The MTA uses hexahedral elements so the problem can be discretized with 1st (Q1) and 2nd (Q2) order elements. COMSOL uses tetrahedral elements with 1st (P1) and 2nd order (P2) elements. The length of the transient simulation is $T = 1$ sec with a fixed time step $\Delta t = 0.001$ sec.

TABLE III
PARAMETERS OF THE BENCHMARK SYSTEM FROM FIGURE 2.

Parameter	C_{T_i}, κ_{T_i}	C_{HS}, κ_{HS}	η	u_a
Value	$1.75 \frac{MJ}{kgK}$, $100\ \frac{W}{mK}$	$3.55 \frac{MJ}{kgK}$, $400\ \frac{W}{mK}$	$100\ \frac{W}{m^2K}$	$318.1\ K$

To compare the accuracy of the temperatures computed by the MTA and COMSOL as the order of the numerical scheme and grid resolution vary, the minimum u_{min} and maximum u_{max} temperatures in the entire domain are compared. These values are listed in Table IV for steady-state simulations, and in Table V at the end of transient simulations, respectively. The temperature $u(\mathbf{x}_c)$ at the center of the active layer $\mathbf{x}_c = (0.005, 0.005, 0.00075)$ is also listed.

TABLE IV
THE STEADY-STATE TEMPERATURES OBTAINED WITH THE MTA AND COMSOL.

Tool	Order	DOF	u_{min}	u_{max}	$u(\mathbf{x}_c)$
MTA	Q1	54,689	346.334	349.826	349.720
	Q1	415,041	346.334	349.836	349.729
	Q1	3,232,385	346.334	349.840	349.733
	Q2	54,689	346.334	349.834	349.727
	Q2	**415,041**	**346.334**	**349.839**	**349.732**
	Q2	3,232,385	346.333	349.841	349.734
Comsol	P1	77,816	346.334	349.831	349.595
	P2	**604,787**	**346.334**	**349.835**	**349.728**

TABLE V
THE TRANSIENT TEMPERATURES AT TIME $T = 1$ SEC OBTAINED WITH THE MTA AND COMSOL.

Tool	Scheme	DOF	u_{min}	u_{max}	$u(\mathbf{x}_c)$
MTA	Q1-BE	7,569	318.176	321.290	321.178
		54,689	318.176	321.313	321.200
		415,041	318.176	321.323	321.209
	Q2-TR	7,569	318.176	321.309	321.196
		54,689	318.176	321.321	321.208
		415,041	**318.176**	**321.326**	**321.213**
Comsol	P1-BE	77,816	318.176	321.319	321.077
	P2-BE	**604,787**	**318.176**	**321.324**	**321.211**

In Table VI, global measures of the error between the COMSOL and MTA solutions for the cases highlighted in bold in Tables IV and V are shown for both steady-state and transient simulations. For the steady-state simulation, the integral of the COMSOL and MTA solutions, highlighted in bold in Table IV, is shown. For the transient simulation, the maximum of the integrals over the time interval $[0, T]$ for the COMSOL and MTA solutions, highlighted in bold in Table V is shown. This global measure of error illustrates the near perfect agreement between the two models.

TABLE VI
GLOBAL ERROR COMPARISON OF THE MTA AND COMSOL.

Simulation	Error
Steady	$\int_\Omega \lvert u_{COMl} - u_{MTA} \rvert\ d\mathbf{x} = 8.32 \cdot 10^{-9}$
Transient	$\max_{[0,T]} \int_\Omega \lvert u_{COM} - u_{MTA} \rvert\ d\mathbf{x} = 5.75 \cdot 10^{-6}$

Furthermore, observe that the difference in the reported temperatures between the MTA and COMSOL is within a few thousandths of a degree, a negligible amount over the temperature range. In Figure 3, the transient temperatures at the point $\mathbf{x}_c$, i.e., $u(\mathbf{x}_c, t)$, for both the MTA and COMSOL are plotted for numerical schemes of different order, which shows almost perfect agreement between the two models.

SL: The thermal simulator Hotspot outputs the average temperature of the active layers. For comparison with the MTA, the average temperature of Silicon Tier 2 is computed in the MTA. This average is computed by summing the temperature at the nodes in Silicon Tier 2 and dividing by the number of nodes in Tier 2. The errors between the average temperatures are reported in Table VII.

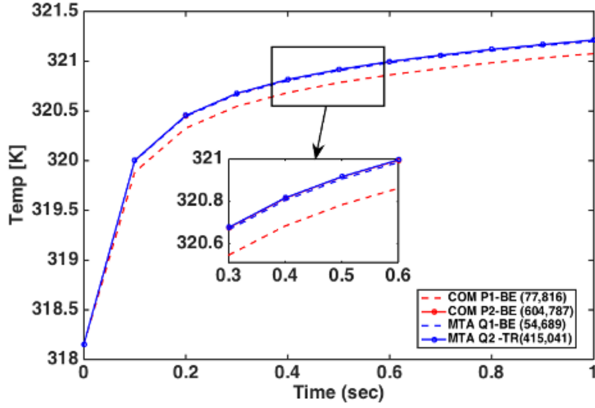


Fig. 3. Time evolution of $u(\mathbf{x}_c)$ computed with the MTA and Comsol.

TABLE VII
ERROR COMPARISON BETWEEN THE MTA AND HOTSPOT.

Simulation	MTA	Hotspot	Error
Steady			
Transient			

For the error estimation in subsequent computations, a highly accurate transient temperature profile computed with the MTA using Q2 elements on a grid with 415,041 DOF and the TR method serves as a reference solution $\mathbf{u}_{ref}$. A measure of accuracy of the solution $\mathbf{u}_h$ to the reference solution is the maximum pointwise error $\|\mathbf{u}_h - P\mathbf{u}_{ref}\|_\infty$[†], where P is the injection operator from the reference to the coarser grid.

In Table VIII, the simulation times and corresponding errors are reported with the simulator running in mode M1 as both the spatial discretization (DOF) and the time step size vary. The total execution times reported in this table are for simulations where the temperature is output once at the end of the simulation ($T = 1$ sec).

TABLE VIII
THE SIMULATION TIME AND ACCURACY IN MODE M1 AS h AND Δt VARY.

Scheme	DOF	Δt	Time (s)	$\|\mathbf{u}_h - P\mathbf{u}_{ref}\|_\infty$ [K]
Q1-BE	54,689 $h = 0.002$	0.1	8.45	0.090
		0.01	56.7	0.075
		0.001	295	0.074
	415,041 $h = 0.001$	0.1	70.7	0.034
		0.01	567	0.019
		0.001	3,924	0.018
Q2-TR	54,689 $h = 0.004$	0.1	16.3	0.123
		0.01	117	0.046
		0.001	615	0.046
	415,041 $h = 0.002$	0.1	135	0.108
		0.01	970	0.001
		0.001	7,433	-

There is a significant decrease in the execution time as the time step size increases. The Q1-BE solution on the coarsest grid with $\Delta t = 0.1$ is accurate to within 0.09 degrees, a difference of less than 3% relative to the temperature range of the system ($\sim$ 3 degrees K). The reduction of Δt by a factor of 10 for the fixed spatial grid does not lead to a tenfold decrease in execution time due to a one-off cost of the solution output (cf. Table VIII). The other computational phases, such as the assembly and solution times, however, do reduce by a factor of ten. Note that the reduction in the error as Δt decreases from 0.01 to 0.001 is almost negligible for each example in Table VIII. This behavior follows the discussion on the computational error from Section III-F and illustrates how both the spatial and time discretization errors contribute to the global error, which is often neglected in the thermal analysis of ICs. A fixed grid resolution (or time step) introduces a certain level of spatial (temporal) error. Thus, computational effort is wasted in simulations when the orders of the mesh width h and time step size Δt differ significantly. For this particular benchmark problem, observe that the mesh width $h = \mathcal{O}(10^{-3})$ and choosing a time step $\Delta t < 0.01$ is unnecessary. Furthermore, this behavior indicates that for this fixed grid resolution, the atomic time step size Δt_a of the external power simulator should be set to $\mathcal{O}(10^{-2})$. In the MTA, the mesh generator provides an estimate of the mesh width that helps to tune the time step.

To illustrate the improved performance the MTA offers over both COMSOL and academic open-source thermal simulators, the execution time of the MTA is compared to both COMSOL and Hotspot 6.0 [10] using the same system depicted in Figure 2. Since Hotspot uses first order accurate schemes, the MTA steady-state simulation is also computed with Q1 elements and the transient simulation is computed in mode M1 (cf. Table II) using the Q1-BE scheme with $\Delta t = 0.01$ sec. The execution times for each of the tools are reported in Table IX.

TABLE IX
TOTAL SIMULATION TIMES OF THE BENCHMARK FROM FIGURE 2.

Analysis	MTA (415,041 DOF)	Hotspot (262,144 DOF)	COMSOL (604,787 DOF)
Steady	33.9 sec	39.7 sec	25.0 sec
Transient ($\Delta t = 0.01$)	567 sec	78,000 sec	80,000 sec

For steady-state problems, the simulation time for the MTA is faster than Hotspot's on a discrete problem that is nearly twice the size. This indicates that for equal problem sizes, the MTA performs thermal simulations in less than half the time required by Hotspot. Moreover, as shown later in Section V-D, the MTA simulates systems with problem sizes that are well beyond the capabilities of Hotspot and other existing tools. A direct pointwise temperature comparison is not possible since Hotspot outputs the temperature at different grid points than the MTA, but the temperature of the two solutions is reasonably close. The improvement in the simulation time is considerably greater for transient thermal simulations, where the MTA is almost two orders of magnitude faster than Hotspot.

As described in [25], the proposed solution methodology allows the execution time to roughly scale linearly in the number of DOFs. This scaling trend can be seen for each phase of the simulation in Table X, which reports the breakdown of times (in secs) for a set of increasingly larger transient simulations conducted in mode M1 with Q1 elements and a fixed time step size $\Delta t = 0.01$ sec.

[†]The infinity norm is defined as $\|\mathbf{u}\|_\infty = \max_{1\leq i\leq n} |u_i|$

TABLE X
THE BREAKDOWN OF TRANSIENT SIMULATION TIMES (IN SEC).

Phase	DOF 7,569	54,689	415,041
Assemble $\mathbf{f}_h^{k+1}$	0.95	6.52	54.0
Assemble M, K	0.09	0.77	6.70
Copy mesh	0.07	0.74	6.90
Distribute DOF	0.02	0.17	1.29
Output	0.29	2.47	20.0
Solve	2.94	41.7	432
Total	5.36	57.3	567

The reported times are from a simulation with 100 time steps. In this simulation, the coefficient matrices M and K are assembled only once and the power source vector $\mathbf{f}_h^{k+1}$ is assembled at each time step. The generated mesh is read and copied by the MTA once at the start of the simulation. Distributing the DOF is a fast and necessary step that entails setting up the various finite element data structures in deal.II. Lastly, in this simulation, the solution is written to an output file once at the end of the simulation.

The mesh generation time, i.e., creating the .msh file, for the simulations from Table X is less than a second for the finest grid in this case study. The majority of the simulation time is spent solving the linear system. Only 10% of the time is spent assembling the power source vector $\mathbf{f}_h^{k+1}$ at each time step. Even with these fast simulation times, additional computational efficiency and gains in the simulation speed are possible by enabling the adaptive spatiotemporal refinement features. These features allow the MTA to analyze more refined and complex circuit designs.

B. Adaptive Spatiotemporal Refinement

The benchmark system from Figure 2 with the same parameters is used to demonstrate the computational advantages of enabling adaptive spatiotemporal refinement. The benefits of temporal adaptivity (mode M2) are discussed first, followed by spatial adaptivity (mode M3), and lastly when both adaptive modes are enabled simultaneously (mode M4).

A simulation in mode M2 on a fixed grid consisting of 415,041 DOF (with $h = 0.001$) is considered first. The initial time step size for this simulation is $\Delta t = 0.001$ sec. In Table XI, the number of time steps N, the simulation times, and the accuracy of the solution are reported for different adaptive time integration methods as the tolerance ε_t varies. Compared to the solution times reported in Table VIII for fixed time step simulations, there is a notable decrease in both the number of time steps and simulation times. If the parameters of the fixed and adaptive BE method are tuned so that they produce similar solution error for the problem under consideration, then the adaptive method will take 30% shorter execution time (cf. Table VIII). Similar or better computational savings are achieved with higher order adaptive time integration schemes, compared to the fixed step schemes. For example, a simulation with the adaptive Q2-STABTR ($k^* = 10$) scheme using $\varepsilon_t = 10^{-4}$ requires only 19 time steps and executes in just over 3 minutes. The error is 0.003 degrees, a difference of less than 1% within the temperature range. Mode M2 simulations have the advantage that the optimal time step size is automatically selected by the simulator at each time step.

The errors reported in the last column of Table XI start to stagnate as ε_t decreases, which indicates that spatial error dominates the solution similarly as in mode M1. Since computational times increase as ε_t decreases, reducing ε_t past a certain point results in extra and unnecessary computational effort. For this particular experiment setting $\varepsilon_t = 10^{-4}$ is sufficient, athough for finer spatial grids, a smaller value of ε_t is more appropriate. For a fixed h, there is a set ε_t value that leads to optimal computational effort. Based on the error analysis of the solution methodology (cf. Section III-F), for a fixed mesh width h, this value is $\varepsilon_t = \mathcal{O}(h^{q+1})$, where q is the order of the time stepping method. This experiment also highlights the importance of stabilization in the TR method. The adaptive TR method suffers from a ringing instability which limits the growth of the time step sizes [30]. The stabilization alleviates this restriction on the growth of the adaptive time step size.

TABLE XI
THE SIMULATION TIME AND ACCURACY IN MODE M2 AS ε_t VARIES .

Scheme	ε_t	N	Time (s)	$\Vert\mathbf{u}_h - P\mathbf{u}_{ref}\Vert_\infty$ [K]
Q1-BE 415,041	10^{-4}	15	102	0.048
	10^{-5}	33	188	0.032
	10^{-6}	88	443	0.023
Q2-BDF2 415,041	10^{-4}	17	181	0.012
	10^{-5}	27	273	0.004
	10^{-6}	47	451	0.001
Q2-TR 415,041	10^{-4}	78	759	0.001
	10^{-5}	477	3980	0.000
	10^{-6}	1000	8000	-
Q2-STABTR $k^* = 10$ 415,041	10^{-4}	19	201	0.003
	10^{-5}	29	286	0.002
	10^{-6}	44	429	0.002

To demonstrate the computational benefits of periodic adaptive spatial refinement, several simulations in mode M3 are run using the Q1-BE scheme with an initial problem size of 54,689 DOF over the time interval $T = 1$ sec. In this set of experiments, the effect on the simulation time and accuracy is considered as the AMR frequency, time step size, and the refinement threshold τ_r are varied. The merging threshold is set to $\tau_m = 0$ to ensure that no elements are merged. For $\Delta t = 0.1$, the AMR frequency is set to 5 giving a total of two adaptive refinements. For $\Delta t = 0.02$, the AMR frequency is set to 10 giving a total of five adaptive mesh refinements.

The results of these simulations are shown in Table XII. Note that increasing τ_r improves the accuracy of the solution but at a higher computational cost since these cases lead to finer grids. More importantly, the simulation where the final grid consists of 109,079 DOF is accurate to within 0.027 degrees, or a difference of roughly 1% within the temperature range. This achieved level of accuracy is better than the M1 temperature profile computed on a fixed grid of 415,041 DOF with $\Delta t = 0.1$ (0.034 degrees in Table VIII) and, more importantly, is obtained with a shorter execution time of 64.5 secs (compared to 70.7 secs). This result highlights

the benefit of AMR in the MTA, namely, the ability to compute temperatures on smaller, more economical grids while achieving the same level of accuracy as for the larger uniformly refined grid.

TABLE XII
THE SIMULATION TIME AND SOLUTION ACCURACY IN MODE M3 AS THE AMR FREQUENCY AND Δt VARY.

AMR freq.	Δt	τ_r	Grid size	Time (s)	$\|\mathbf{u}_{AMR} - P\mathbf{u}_{ref}\|_\infty$ [K]
5	0.1	0.25	58,713	15.4	0.091
		0.50	73,290	16.9	0.046
10	0.02	0.25	109,079	64.5	0.027
		0.50	411,812	164	0.020

Enabling adaptive spatial or temporal refinement individually improves the computational efficiency of thermal simulations. Mode M4 in the MTA enables both of these features simultaneously for fully adaptive spatiotemporal refinement. In Table XIII, a comparison of the simulation time and accuracy between all four operation modes is shown.

TABLE XIII
THE SIMULATION TIME (SECS) AND ACCURACY OF THE DIFFERENT MTA SIMULATION MODES.

Mode	Size	Time (N)	$\|\mathbf{u}_h - P\mathbf{u}_{ref}\|_\infty$ [K]
M1 ($\Delta t = 0.02$)	415,041	228 (50)	0.022
M2 ($\varepsilon_t = 10^{-5}$)	415,041	188 (33)	0.032
M3 ($\Delta t = 0.02$, AMR 10, $\tau_m = 0.5$)	411,812	164 (50)	0.020
M4 ($\varepsilon_t = 5.0 \cdot 10^{-5}$, AMR 8, $\tau_m = 0.5$)	367,470	110 (38)	0.024

To demonstrate the effects of a larger power variance on the performance of the spatiotemporal adaptive modes, Table XIII is reproduced below with a steady power of 100W applied to the active regions of the circuit.

TABLE XIV
THE SIMULATION TIME (SECS) AND ACCURACY OF THE DIFFERENT MTA SIMULATION MODES FOR LARGER POWER VARIANCE.

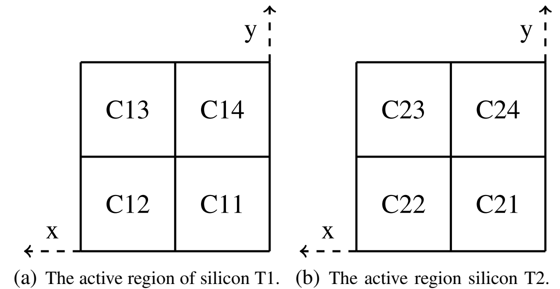
Mode	Size	Time (N)	$\|\mathbf{u}_h - P\mathbf{u}_{ref}\|_\infty$ [K]
M1 ($\Delta t = 0.02$)	415,041		
M2 ($\varepsilon_t = 10^{-5}$)	415,041		
M3 ($\Delta t = 0.02$, AMR 10, $\tau_m = 0.5$)			
M4 ($\varepsilon_t = 5.0 \cdot 10^{-5}$, AMR 8, $\tau_m = 0.5$)			

This table shows the impact of fully adaptive spatiotemporal refinement in reducing the computational cost of thermal simulations while maintaining the same level of accuracy compared to simulations with fixed discretization parameters. The M4 simulation is run using the Q1-BE scheme. The initial grid size is 54,689 DOF, the AMR frequency is 8, $\varepsilon_t = 5.0 \cdot 10^{-5}$, $\tau_r = 0.5$, and $\tau_m = 0$. The simulation time of the fully adaptive M4 simulation is 110 secs and requires 38 time steps, which is fastest amongst all modes. The final grid consists of only 367,470 DOFs and the temperature is accurate to within 0.024 degrees. Although the M2 simulation takes 5 fewer time steps, the total simulation time is longer since the fixed computational grid yields a larger discrete problem size. The M3 simulation is marginally more accurate than the M4 simulation but using a fixed time step size yields a longer simulation. For this set of experiments, enabling full spatiotemporal adaptivity halves the execution time while maintaining the same level of accuracy as a simulation with fixed spatiotemporal discretization parameters. In addition, simulations with simultaneous spatiotemporal adaptivity take shorter times to execute for the same level of accuracy than the simulations with only one adaptively refined discretization parameter (modes M2 and M3).

C. Nonlinear transient thermal simulations

The operating temperature range of modern IC technologies is typically between 300-400 K [43]. The thermal conductivity and specific heat of the silicon components of a circuit are temperature-dependent and vary significantly over this temperature range. Thus, a linear model is no longer sufficient to accurately represent the underlying physics and does not capture the hotspots of a circuit. The following experiments demonstrate that the linear and nonlinear models predict a different number of hotspots over this temperature range. This behavior demonstrates the need for added accuracy and favors the use of nonlinear models in transient simulations.

Nonlinear thermal simulations are performed on a similar benchmark system to the one depicted in Figure 2. However, for this case, the dimensions of the integrated circuit system have been modified. The dimensions are now $10\ mm \times 10\ mm \times 0.125\ mm$ for the silicon tiers, $10\ mm \times 10\ mm \times 0.25\ mm$ for the heat spreader, and $50\ mm \times 50\ mm \times 10\ mm$ for the heat sink. The two silicon tiers are now subdivided in the z-direction into an active layer of height 0.025 mm and an inactive bulk layer of height 0.1 mm. The active layer is then subdivided in the $x-y$ plane into quadrants representing a 4 core system as shown in Figure 4. This specific geometry



(a) The active region of silicon T1. (b) The active region silicon T2.

Fig. 4. The active cores of the silicon tiers.

which consists of 4 cores for the active silicon layer has been previously considered in [44]–[46]. The two silicon tiers and the heat spreader are attached to a heat sink with 11 fins.

The thermal parameters of the two silicon tiers, i.e., $C(u), \kappa(u)$ in (1a), use the values listed in Table I. The thermal parameters of the heat spreader and sink are constant over the operating temperature range of the circuit. The boundary condition at the bottom silicon layer is assumed to be adiabatic. A nonlinear M1 simulation using a first order scheme (Q1-BE) over the time interval $T = 50$ sec with $\Delta t = 0.01$ sec is performed. The initial and ambient temperatures are assumed to be 300 K. During the simulation, a core dissipates three set levels of power: 0 W (idle), 10 W, or 25 W. The 25 W of power represents an upper bound that a core of this size can dissipate [3], whereas the smaller power value is an intermediate value. It is assumed that during each

second 80% of the cores are active. During this interval, the active cores dissipate high power 80% of the time and low power the remaining 20%.

The hot spots in the circuit are identified as any node in the mesh where the temperature is above the threshold temperature of 358.15 K, which is a representative value for the safe operating temperature of the circuit [43]. The temperature trace at the spatial point $\mathbf{x}_* = (0.0045, 0.00625, 0.0001)$ is shown in Figure 5. Note that this point is in core C14.

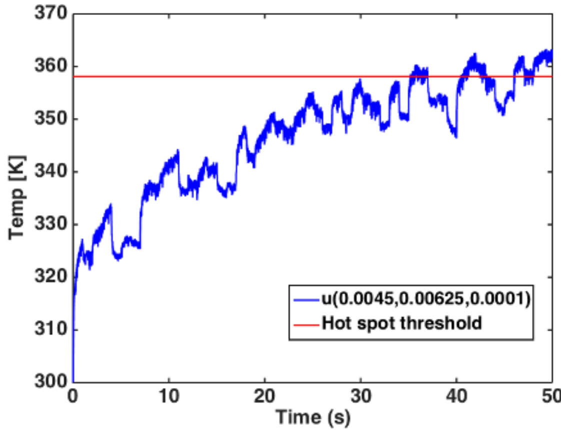
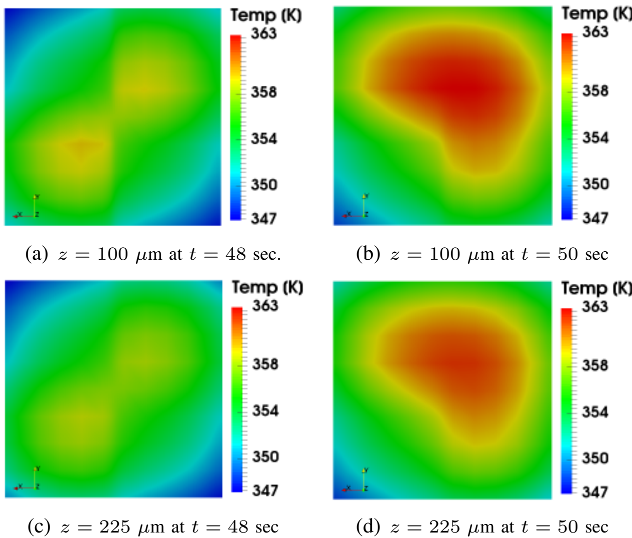


Fig. 5. Time evolution of $u(0.0045, 0.00625, 0.0001)$ with the nonlinear model.

The point $\mathbf{x}_*$ is selected as it is the location where the maximum temperature is encounterd during the simulation. Note that the temperature at this point is larger than the hot spot threshold temperature after 35 secs, and that the highest temperatures are observed between 48 and 50 secs. To better understand the formation of hot spots, cross-sections of the nonlinear thermal profile of the circuit at $z = 100$ μm and $z = 225$ μm at 48 and 50 secs are shown in Figure 6.



(a) $z = 100$ μm at $t = 48$ sec. (b) $z = 100$ μm at $t = 50$ sec

(c) $z = 225$ μm at $t = 48$ sec (d) $z = 225$ μm at $t = 50$ sec

Fig. 6. Cross-sections of the nonlinear active layer.

Next, the numbers of hot spots as identified by the linear and nonlinear models is presented. To this end, two linear simulations with $\kappa_1 = 100$ [W/mK] and $\kappa_2 = 150$ [W/mK] are performed in addition to the nonlinear simulation. Note that κ_1 and κ_2 are the thermal conductivities of silicon at 300 K and 400 K, respectively. At each time step in the simulation, the total number of nodes where the computed temperature is above the hot spot threshold is recorded. Figure 7 depicts these numbers over the interval $[48, 50]$ secs.

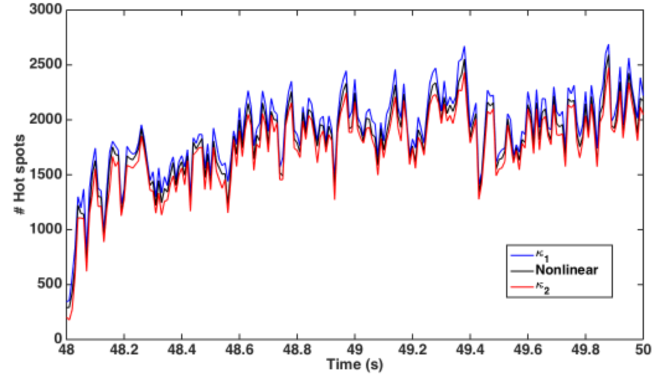


Fig. 7. Total number of hot spots for the linear and nonlinear simulation between 48-50 secs.

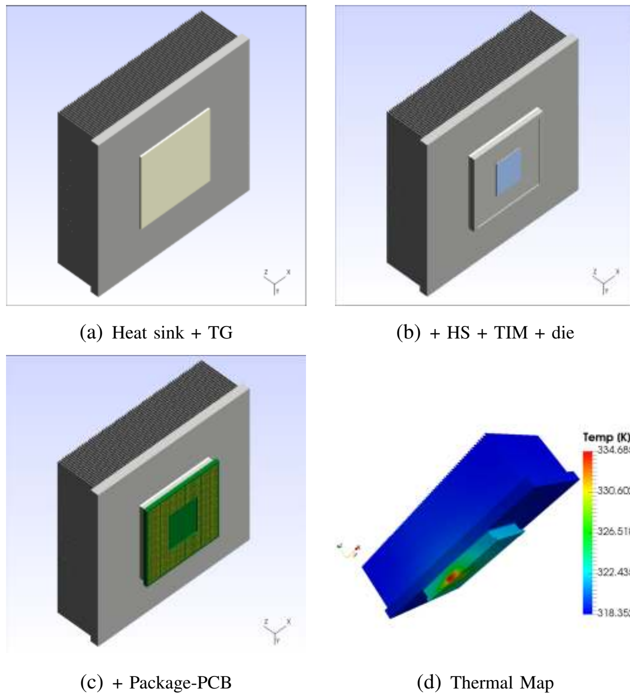
Observe that the total number of hot spots predicted by the nonlinear model is bounded above by the linear simulation with κ_1 and below by the linear simulation with κ_2. During this time interval, the maximum difference between the number of hot spots in the linear simulation with κ_1 and the nonlinear simulation is 221. This maximum difference occurs at time $t = 48.15$ sec. Similarly, the maximum difference between the number of hot spots in the nonlinear simulation and the linear simulation with κ_2 is 229, which occurs at time $t = 48.06$ secs, i.e., in the critical range when maximum temperatures are observed.

This behavior shows that linear simulations with lower thermal conductivities over predict the total number of hot spots, whereas linear simulations with higher thermal conductivities predict fewer hot spots. Utilizing nonlinear models, which incorporate temperature-dependent thermal parameters, allows the simulator to accurately capture the total number and the locations of hot spots, thereby avoiding either underdesign (κ_2) or overdesign (κ_1) of the circuit to appease exacerbating temperatures.

D. Complex Structures and Parallel Performance

A benchmark integrated system is considered to further demonstrate the capabilities of the MTA to perform the thermal analysis of circuits and complex package structures. The considered system, called the LGA benchmark, is the Intel Xeon processor (Nehalem Architecture) in a flip-chip land grid array (FCLGA) package described in [42]. Figures 8(a)-8(c) illustrate the assembly of a heat sink with 72 fins and thermal grease (TG), the CPU, and package components. Displayed in Figure 8(d) is the thermal profile of the LGA benchmark from a steady-state thermal simulation with the splash2-barnes power trace. The discretization of the LGA benchmark with Q1 elements results in a mesh with 3,604,605 DOF and the thermal simulation is executed in parallel.

In Table XV, the breakdown of the total simulation time into different phases as a function of the number of processors used is reported for the steady-state LGA benchmark thermal simulation.



(a) Heat sink + TG (b) + HS + TIM + die

(c) + Package-PCB (d) Thermal Map

Fig. 8. The LGA benchmark geometry and steady-state temperature map.

TABLE XV
TIME (IN SECS) SPENT IN EACH PHASE OF A STEADY-STATE SIMULATION WITH THE LGA BENCHMARK AS A FUNCTION OF THE NUMBER OF PROCESSORS.

	Phase	1 core	2 cores	4 cores
Computation	Assembly	26.0	13.0	7.00
	Distribution	11.0	7.00	6.00
	Solve	46.0	34.0	30.0
Total computation		83.0	55.0	43.0
Read/Write	Copy mesh	67.0	77.0	89.0
	Output	271	132	56.0
Total read/write		338	209	145
Total simulation time		421	264	188

Note that the time of the computational phase alone, which excludes the time to copy the mesh and output the results, for 1 core is roughly 80 secs and improves to 40 secs on 4 cores. A linear scaling with the number of cores is observed for the output and assembly phases of the simulation. The time of the copy mesh phase increases with the number of cores as the deal.II shared memory data structures require a copy of the initial fine mesh on each core to start the simulation. Although the distribution and solve phases do not scale optimally due to the constraints of shared memory parallelization, there is still a notable reduction in the solve time and thus the overall simulation time which enables the thermal analysis of complex integrated circuit systems.

VI. Conclusions and Future Work

In this paper, the MTA, an advanced and versatile academic tool for the thermal analysis of integrated systems is introduced. The MTA is a physics based model, as opposed to models which invoke the electrothermal duality. The tool employs a state-of-the-art numerical methodology for the discretization of the nonlinear heat equation. As shown through several detailed experiments, the MTA provides fast and accurate thermal simulations with the ability to model the complex geometries of realistic IC systems in parallel. The proposed thermal simulator has the ability to compute temperature profiles of ICs discretized on computational grids consisting of 3M nodes in under 3 minutes, surpassing the computational and modeling abilities of existing academic thermal simulators.

In addition to providing larger and faster simulation capabilities, the MTA provides more computationally efficient simulations by enabling adaptive spatiotemporal refinement features. Moreover, the established error analysis results of the employed numerical schemes are used to efficiently configure simulations in the MTA as well as other external architecture simulators. This feature improves the speed of the design flow process. Existing academic tools are either unable or do not support methodologies that allow for such optimal configurations.

The MTA supports nonlinear thermal simulations in four stable operation modes which enable different combinations of the adaptive spatiotemporal refinement features. As shown through an example simulation, the linear and nonlinear models predict different numbers of hot spots and their locations. The use of the highly accurate nonlinear model offers an important means for improved thermal management of integrated systems in order to avoid excessive (or under) utilization of thermal management techniques or policies. Overall, the MTA offers a publicly available, easy to use, multi-feature tool, approaching in a comprehensive manner the many aspects and requirements of the IC thermal analysis process.

Acknowledgment

This work is supported by the Engineering and Physical Sciences Research Council (EPSRC) under grant number EP/M009238/1.

References

[1] M. M. Waldrop, "The Chips Are Down for Moore's Law," *Nature News*, vol. 530, no. 7589, p. 144, February 2016.
[2] C. Xu, S. K. Kolluri, K. Endo, and K. Banerjee, "Analytical Thermal Model for Self-Heating in Advanced FinFET Devices with Implications for Design and Reliability," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 7, pp. 1045–1058, June 2013.
[3] A. Danowitz, K. Kelley, J. Mao, J. P. Stevenson, and M. Horowitz, "CPU DB: Recording Microprocessor History," *Communications of the ACM*, vol. 55, no. 4, pp. 55–63, April 2012.
[4] M. B. Taylor, "Is Dark Silicon Useful?: Harnessing the Four Horsemen of the Coming Dark Silicon Apocalypse," in *DAC Design Automation Conference*, June 2012, pp. 1131–1136.
[5] V. F. Pavlidis and E. G. Friedman, *Three-dimensional Integrated Circuit Design*, 1st ed. Morgan Kaufmann, 2009.

[6] S. Borkar, "3D Integration for Energy Efficient System Design," in *Proceedings of the 48th Design Automation Conference*, ser. DAC 2011. ACM, June 2011, pp. 214–219.

[7] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-Aware Microarchitecture," in *Proceedigns of the 30th Annual International Symposium on Computer Architecture*, June 2003, pp. 2–13.

[8] Y. Yang, C. Zhu, Z. Gu, L. Shang, and R. P. Dick, "Adaptive Multi-domain Thermal Modeling and Analysis for Integrated Circuit Synthesis and Design," in *Proceedings of the IEEE/ACM International Conference on Computer Aided Design*, November 2006, pp. 575–582.

[9] P. Li, L. T. Pileggi, M. Asheghi, and R. Chandra, "IC Thermal Simulation and Modeling via Efficient Multigrid-based Approaches," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 9, pp. 1763–1776, August 2006.

[10] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan, "HotSpot: A Compact Thermal Modeling Methodology for Early-Stage VLSI Design," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 14, no. 5, pp. 501–513, July 2006.

[11] A. Sridhar, A. Vincenzi, D. Atienza, and T. Brunschwiler, "3D-ICE: A Compact Thermal Model for Early-Stage Design of Liquid-Cooled ICs," *IEEE Transactions on Computers*, vol. 63, no. 10, pp. 2576–2589, October 2014.

[12] T.-Y. Wang and C. C.-P. Chen, "3-D Thermal-ADI: A Linear-Time Chip Level Transient Thermal Simulator," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 12, pp. 1434–1445, December 2002.

[13] A. Fourmigue, G. Beltrame, and G. Nicolescu, "Efficient Transient Thermal Simulation of 3D ICs with Liquid-Cooling and Through Silicon Vias," in *Proceedings of Design, Automation and Test in Europe Conference Exhibition*, March 2014, pp. 1–6.

[14] L. N. Trefethen and D. Bau III, *Numerical Linear Algebra*. SIAM, 1997.

[15] X. S. Li, J. W. Demmel, J. R. Gilbert, L. Grigori, M. Shao, and I. Yamazaki, "SuperLU Users' Guide," 2011, http://crd.lbl.gov/∼xiaoye/SuperLU/.

[16] A. Fourmigue, G. Beltrame, and G. Nicolescu, "Explicit Transient Thermal Simulation of Liquid-Cooled Transient," in *Proceedings of Design, Automation and Test in Europe Conference Exhibition*, March 2013, pp. 1385–1390.

[17] A. Fourmigue, G. Beltrame, G. Nicolescu, and E. M. Aboulhamid, "A Linear-Time Approach for the Transient Thermal Simulation of Liquid-Cooled 3D ICs," in *Proceedings of the IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, October 2011, pp. 197–206.

[18] R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-state and Time-dependent Problems*. SIAM, 2007.

[19] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite Elements and Fast Iterative Solvers: with applications in incompressible fluid dynamics*, 2nd ed. Oxford University Press, 2014.

[20] V. E. Henson and U. M. Yang, "BoomerAMG: A Parallel Algebraic Multigrid Solver and Preconditioner," *Applied Numerical Mathematics*, vol. 41, no. 1, pp. 155–177, April 2002.

[21] V. Simoncini and D. B. Szyld, "Recent Computational Developments in Krylov Subspace Methods for Linear Systems," *Numerical Linear Algebra with Applications*, vol. 14, no. 1, pp. 1–59, October 2006.

[22] Z. Hassan, N. Allec, F. Yang, L. Shang, R. P. Dick, and X. Zeng, "Full-Spectrum Spatial–Temporal Dynamic Thermal Analysis for Nanometer-Scale Integrated Circuits," *IEEE Transactions on Very Large Scale Integration*, vol. 19, no. 12, pp. 2276–2289, December 2011.

[23] A. Ramalingam, F. Liu, S. R. Nassif, and D. Z. Pan, "Accurate Thermal Analysis Considering Nonlinear Thermal Conductivity," in *Proceedings of the 7th International Symposium on Quality Electronic Design*, March 2006, pp. 644–649.

[24] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, "McPAT: An Integrated Power, Area, and Timing Modeling Framework for Multicore and Manycore Architectures," in *IEEE/ACM International Symposium on Microarchitecture*, Dec. 2009, pp. 469–480.

[25] S. Ladenheim, Y.-C. Chen, M. Mihajlović, and V. Pavlidis, "IC Thermal Analyzer for Versatile 3-D Structures Using Multigrid Preconditioned Krylov Methods," in *Proceedings of the International Conference on Computer-Aided Design*, November 2016, pp. 123:1–123:8.

[26] H. Sultan and S. R. Sarangi, "A fast leakage aware thermal simulator for 3d chips," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, March 2017, pp. 1733–1738.

[27] C. Yan, H. Zhu, D. Zhou, and X. Zeng, "An efficient leakage-aware thermal simulation approach for 3d-ics using corrected linearized model and algebraic multigrid," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, March 2017, pp. 1207–1212.

[28] Y. Ma, K. Wang, S. Dong, Y. Wang, and X. Hong, "Thermal effects of leakage power in 3D ICs," in *The 2010 International Conference on Green Circuits and Systems*, June 2010, pp. 578–583.

[29] D. W. Kelly, J. P. De S. R. Gago, O. C. Zienkiewicz, and I. Babuška, "A Posteriori Error Analysis and Adaptive Processes in the Finite Element Method: Part I–Error Analysis," *International Journal for Numerical Methods in Engineering*, vol. 19, pp. 1593–1619, November 1983.

[30] J. A. Lee, J. Nam, and M. Pasquali, "A New Stabilization of Adaptive Step Trapezoid Rule Based on Finite Difference Interrupts," *SIAM Journal on Scientific Computing*, vol. 37, no. 2, pp. A725–A746, March 2015.

[31] P. Gresho and R. Sani, *Incompressible Flow and the Finite Element Method Advection-Diffusion and Isothermal Laminar Flow*. John Wiley & Sons, 1998.

[32] M. R. Hestenes and E. Stiefel, "Methods of Conjugate Gradients for Solving Linear Systems," *Journal of Research of the National Bureau of Standards*, vol. 49, pp. 409–436, December 1952.

[33] Y. Saad, *Iterative methods for sparse linear systems*. SIAM, 2003.

[34] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang, "PETSc Web page," 2016. [Online]. Available: http://www.mcs.anl.gov/petsc

[35] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith, *Efficient Management of Parallelism in Object-Oriented Numerical Software Libraries*. Birkhäuser, 1997, pp. 163–202.

[36] W. Bangerth, R. Hartmann, and G. Kanschat, "deal.II – a General Purpose Object Oriented Finite Element Library," *ACM Transactions on Mathematical Software*, vol. 33, no. 4, pp. 24/1–24/27, August 2007.

[37] W. Bangerth, T. Heister, and G. Kanschat, *deal.II Differential Equations Analysis Library, Technical Reference*. [Online]. Available: http://www.dealii.org

[38] Y. S. Touloukian and C. Ho, *Thermophysical Properties of Matter: Specific Heat - Metallic Elements and Alloys*. New York: IFI/Plenum, 1970, vol. 4.

[39] Y. S. Touloukian, R. Powell, C. Ho, and P. Klemens, *Thermophysical Properties of Matter: Thermal Conductivity - Metallic Elements and Alloys*. New York: IFI/Plenum, 1970, vol. 1.

[40] W. Dörfler, "A Convergent Adaptive Algorithm for Poissons Equation," *SIAM Journal on Numerical Analysis*, vol. 33, pp. 1106–1124, June 1996.

[41] *Comsol Multiphysics 5.2*, 2016. [Online]. Available: http://www.comsol.com

[42] Intel, "Intel Xeon Processor 5500/5600 Series Thermal/Mechanical Design Guide," http://www.intel.com, 2010.

[43] D. Edwards, "Semiconductor and IC Package Thermal Metrics," 2012.

[44] X. Zhou, Y. Xu, Y. Du, Y. Zhang, and J. Yang, "Thermal Management for 3D Processors via Task Scheduling," in *Proceedings of the International Conference on Parallel Processing*, September 2008, pp. 115–122.

[45] C. Zhu, Z. Gu, L. Shang, R. P. Dick, and R. Joseph, "Three-Dimensional Chip-Multiprocessor Run-Time Thermal Management," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 8, pp. 1479–1492, Aug 2008.

[46] K. Kang, J. Kim, S. Yoo, and C. M. Kyung, "Runtime Power Management of 3-D Multi-Core Architectures Under Peak Power and Temperature Constraints," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 6, pp. 905–918, June 2011.