Interconnection of Subsystem Reduced-Order Models in the Electro-Thermal Analysis of Large Systems

Pramod Mathai, *Graduate Research Assistant, University of Maryland, College Park,*
and Benjamin Shapiro, *Associate Professor, University of Maryland, College Park.*

***Abstract*— Heat conduction in an electronic device is commonly modeled as a discretized thermal system (eg, finite element or finite difference models) that typically uses large matrices for solving complex problems. The large size of electronic-system heat transfer models can be reduced using model reduction methods and the resulting reduced-order models can yield accurate results with far less computational costs. Electronic devices are typically composed of components, like chips, printed circuit boards, and heat sinks that are coupled together. There are two ways of creating reduced-order models for devices that have many coupled components. The first way is to create a single reduced-order model of the entire device. The second way is to interconnect reduced-order models of the components that constitute the device. The second choice (which we call the "reduce then interconnect" approach) allows the heat transfer specialist to perform quick simulations of different architectures of the device by using a library of reduced-order models of the different components that make up the device. However, interconnecting reduced-order models in a straightforward manner can result in unstable behavior. The purpose of this paper is two-fold: creating reduced-order models of the components using a Krylov subspace algorithm and interconnecting the reduced-order models in a stable manner using concepts from control theory. In this paper we explain the logic behind the "reduce then interconnect" approach, formulate a control-theoretic method for it, and finally exhibit the whole process numerically, by applying it to an example heat conduction problem.**

***Index Terms*— Electro-thermal analysis, heat conduction, interconnecting reduced-order models, compact modeling, Krylov model reduction.**

I. INTRODUCTION

THERMAL management and design in electronic and MEMS devices has assumed greater importance because of the trend towards higher power densities and the continuous miniaturization of electronic devices. In order to be able to create good thermal models of electronic devices the designer has to have a thorough understanding of the heat transfer modes of the components of an electronic device. It is also necessary for the designer to be able to efficiently and accurately compute the impact of his design on the heat transfer properties of the device.

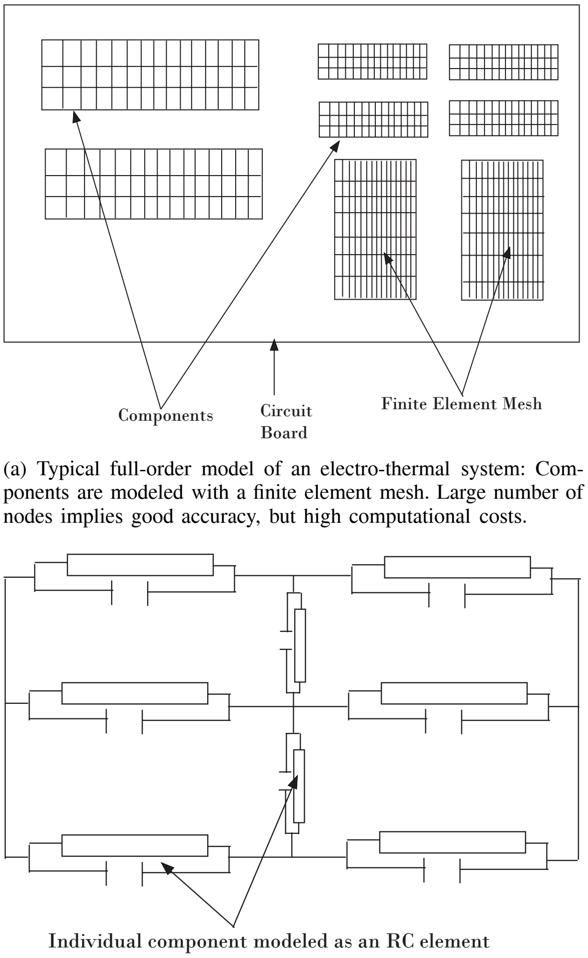
A large volume of work exists with regards to understanding the physics of heat transfer in electronic enclosures. Yang [1] provides overviews of studies in the area of convection heat transfer. Moffat and Ortega [2] and Peterson and Ortega [3] have detailed discussions of applications involving natural convection in electronic enclosures. Larson and Viskanta [5] determined that radiation heat transfer was orders of magnitude larger than convection for 2 dimensional enclosures. Although convection and radiation form a very important part of the heat transfer processes in an electronic system, in this paper we concentrate only on conduction. By itself, this is a very important problem (as discussed in [6]). It is hoped that the compact modeling ideas discussed in this paper can be extended to convection and radiation.

The thermal properties of an electronic device are influenced by a range of parameters like chip positions, cooling channel shapes and their placement, and fan speed. In order to effectively use a thermal model, a designer must be able to search as much of the parameter space that influences the operating regime of the device as possible. Finite element or finite difference methods are common numerical procedures in studying heat transfer problems in electronic devices [7], [8]. Resistance models [9], eigenfunction related methods [10], [11] and the deconvolution method [12] have also been applied to the analysis of thermal networks. Such numerical schemes create a discretized model that is an approximation of the actual thermal problem. If the designer needs to guarantee a high level of accuracy for this large parameter space, the number of states in an FEM model has to be large. This implies increased costs in terms of memory requirements and computational time.

Thus, computational costs for electro-thermal design of complex devices necessitate the development of compact modeling efforts. Kreuger and Bar-Cohen [22] presented one of the earliest compact modeling efforts in which a chip package was modeled with a simplified resistor network that reduced the mesh size and the computational time involved. Lasance et al [23], formulated a simplified resistance network that was independent of boundary conditions, while in [24], a hierarchial compact modeling effort for chip packages was developed. While this kind of approach reduces the computational costs, it might not yield the detailed information that a designer needs for electro-thermal design of a complex device. What one needs are compact modeling methods that, while retaining model accuracy, simultaneously reduces the size of (the typically large) FEM models, thus reducing computational costs. Across engineering fields, such techniques have been termed as dynamic reduced-order modeling (or just reduced-order modeling). There are various reduced-order modeling techniques like balanced truncation [13], Krylov

Manuscript received March 24, 2005; revised August 15, 2006.

subspace methods [14] and proper orthogonal decomposition [15]. These techniques have been succesfully demonstrated to work in a wide range of engineering fields including fluid dynamics [16], VLSI design [17], and control of fluid flow [18]. They have also been proposed in the MEMS and electronics packaging community [19],[20]. In [19], an Arnoldi-based thermal network has been proposed that gives very good results using a single reduced-order model for an entire device. Our goal is to apply reduced-order modeling methods to the conduction heat transfer problem. Fig.1 below is a graphical comparison between different approaches to heat transfer design.



(a) Typical full-order model of an electro-thermal system: Components are modeled with a finite element mesh. Large number of nodes implies good accuracy, but high computational costs.

(b) A commonly used reduced-order model: Modeling the system as an RC network can minimize computation time, but the accuracy achieved may not be sufficient for complex circuits.

Fig. 1. Typical methods that are used for modeling electro-thermal systems

This paper goes a step beyond the typical electro-thermal compact modeling paradigm. When faced with a complex device that is made up of many components, we contend that the ability to plug and play with reduced-order models of different components is an important tool in the hands of the designer. The designer should have a library of reduced-order models of different components that typically exist in a device and be able to experiment with the placement of these different components in order to optimize for heat dissipation. In the next section we explain the need for this kind of 'reduce then interconnect' approach to electro-thermal design. Keeping this need in mind, in this paper we propose a technique to interconnect the reduced-order models of different components so as to reliably reproduce full device behavior.

The remainder of this paper is divided into the following sections. Section II explains the logic and need behind the 'reduce then interconnect' approach that has been mentioned above. Section III discusses the structure of state space models that is derived from the FEM formulation of a conduction problem; Sections IV discusses reduced-order models and, in particular, the Krylov based reduction algorithm that we have used in creating the reduced-order models that are discussed in this paper; Section V discusses the approach that we use in interconnecting the various reduced-order models so that we can achieve accurate full system behavior; and finally, in section VI, we discuss the results of our approach with the help of numerical examples. In the rest of this paper, the terms 'system' and 'device' are used interchangeably, with the former being used in the mathematical portion of the text and the latter being used otherwise. The same rule of usage holds for the terms 'subsystem' and 'component'. The abbreviations FOM and ROM stand for Full-Order-Model and Reduced-Order-Model respectively.

II. THE 'REDUCE THEN INTERCONNECT' APPROACH

For model reduction of electronic devices composed of many connected components, there are two basic model reduction choices: a) connect the components and model reduce the device or b) model reduce the components and then connect the individual reduced models to get a reduced-order model of the entire device. The second option is preferable to the first. In the first option, a full (unreduced) finite element model must be created and reduced each time to create device models with different component architectures. In the second option, we have a library of reduced-order component models, and these components can be connected in different ways to get computationally cheap device models for different component architectures. We stress that the interconnection format that we present (using the C matrix that is introduced in section IV) is an exact mathematical description of the physical interconnection between components of the device. Any given physical interconnection that might exist between the components in terms of heat fluxes exchanged between them, can be represented in the mathematical format that we have in this paper. We focus on the reduce then connect method: we create reduced-order component models that are sufficiently rich to provide accurate results but are sufficiently small to enable fast simulation of heat conduction in complex devices. A central requirement in the above approach is that it should indeed be possible to interconnect the reduced-order component models in a stable and accurate manner. Anderson[21] explains why a straightforward mathematical representation of the physical interconnection of component models can result in inaccurate or stable behavior. Here we discuss a method to mathematically interconnect the reduced-order component component models in such a way that the resulting device models are stable and accurate.

The amount of computational time that can be saved by using the reduce then connect paradigm can be demonstrated by a simple two dimensional example. Consider a device made up of ten components. Suppose five of these components have external heating sources, and the device temperature is to be monitored at twenty junctions. If each component is discretized into twenty by twenty nodes, then each component will have 400 states, and the finite element model will have 4000 states. Thus the mapping from the five external heating sources q1, q2, q3, q4, and q5 to the twenty junction temperatures T1, ,T20 will be a 4000 state model. On a PC, such a model might take (say) 2 minutes of simulation time.

Suppose that we apply model reduction with the connect then reduce paradigm. If this finite element description is reduced to 100 states by model reduction techniques, then the 4000 state mapping from the heating sources q1, q2, q3, q4, and q5 to the junction temperatures T1, ,T20 is replaced by a 100 state mapping that can be evaluated in (say) 5 seconds. This means that we can evaluate the relation from the heating sources to the junction temperatures in 5 seconds for one device architecture. However, if we want to examine a different device architecture, i.e. if we wanted to connect the ten components in a different geometry, then we would have to *recreate* the reduced-order model. Since the initial creation of the reduced-order model requires an evaluation of the original finite element model, it would take us 2 minutes to evaluate each new architecture.

Now, lets choose to use the ‘reduce then connect’ paradigm instead. Assume that each 400 state component is replaced by a 10 state reduced-order model. Each model has inputs that will correspond to the fluxes and temperatures that it will receive from the adjoining components and the models for the five active components also have an input that corresponds to each of their external heating fluxes. Likewise, each component has outputs that will supply its neighboring components with temperature and heating fluxes that they need as their boundary conditions. Now the ten components that have 10 internal states each can be connected in different ways to achieve reduced-order models with different architectures. Each interconnected model will have 100 states and will evaluate in 5 seconds. In the reduce then connect paradigm, once a library of reduced-order models has been created, the designer can examine both different heating fluxes q1, q5 and different interconnection architectures, and can find the resulting temperatures at the twenty junctions in seconds.

Though the second approach is more useful, a simple interconnection of the reduced-order models in a state space manner can lead to stability problems as discussed in section V and in [21]. In section V, we will show that if we intend to interconnect the model reduced subsystems, then we must make sure that the model reduced subsystems are accurate at the dominant frequencies of the interconnected system. We describe a control theoretic approach to interconnecting the reduced-order models of the components in a way that replicates the behavior of the entire device in a stable and accurate manner.

III. DISCRETE STATE SPACE MODEL

In order to apply Krylov model reduction techniques to any given problem, it is first necessary to couch the problem numerically in a state space format. In order to do this, we first create a finite element formulation of the conduction (diffusion) equation using the commercial software package FEMLAB [25]. The state space model of an isolated component can be extracted from this finite element model by linearizing the problem around a nominal temperature (we again use FEMLAB to create the state space model from the finite element model). The governing equations of the state space model are

$$Ax(t) + E\dot{x}(t) = Bu(t). \tag{1}$$

$$T(t) = D^T x(t). \tag{2}$$

in which $x(t)$ is the M x 1 vector of the discretized temperature modes at the various nodes of the mesh, and u(t) is the p x 1 vector of external heat inputs to the system (power dissipated by the circuit, or flux received by the circuit). A and E are the constant M x M stiffness and mass matrices respectively. B is a constant M x p matrix that converts the external heat inputs $u(t)$ into an M x 1 input vector for the differential equation governing the state vector $x(t)$. For typical structures of electronic systems, M is usually very large. $T(t)$ represents a set of n junction temperatures at specified locations. D maps the M thermal modes to the n junction temperatures.

Assuming zero initial temperature offset $x(0) = 0$ (we can include an initial temperature in the formulation below but it does not change the basic results in this paper), and taking the Laplace transform of Eqn.1, we get the frequency domain formulation of the state space model as

$$Ax(s) + sEx(s) = Bu(s) \tag{3}$$

in which $x(s)$ and $u(s)$ are the Laplace transforms of $x(t)$ and $u(t)$. Hence, the Laplace transform from the heat input $u(s)$ to the junction temperatures $T(s)$ is given by

$$T(s) = D^T(A + sE)^{-1}Bu(s) \tag{4}$$

Equation (4) is the complete state space model of heat conduction in the component, which relates the heat inputs $u(s)$ and the junction temperatures $T(s)$ in the frequency domain. The frequency s, is basically, the speed of response of the system. The higher frequency modes of the system die out quickly, while the lower frequency modes are dominant and are mainly responsible for the long term response of the system.

IV. REDUCED-ORDER MODELING OF AN ISOLATED COMPONENT

In the reduce then connect paradigm that we explained in section II, we discussed the utility of creating reduced-order models of individual components before interconnecting them to form a reduced-order model of the entire device. In this section, we explain the procedure and give an algorithm [14] for creating reduced-order models for an isolated component. The reduced-order model for a given component can be obtained by projecting the original linear dynamic system (1)-(2) onto a smaller state-space of dimension $m << M$, by means of an M x m projection matrix U. Derivation of the matrix U is a critical step in our paper and will be outlined later in this section, but for now we assume that U is known. The state variable $x(t)$ can then be written as

$$x(t) = U\tilde{x}(t). \tag{5}$$

where $\tilde{x}(t)$ is the state vector projected onto the reduced space. Here $x(t)$, of dimension M is large and $\tilde{x}(t)$, of dimension m, is small. Substituting equation (5) into equation (1) and multiplying by U^T, a reduced-order model of the form

$$\tilde{A}\tilde{x}(t) + \tilde{E}\dot{\tilde{x}}(t) = \tilde{B}u(t). \tag{6}$$

$$T(t) = \tilde{D}^T\tilde{x}(t). \tag{7}$$

is obtained, in which $\tilde{A}, \tilde{E}, \tilde{B}$ and $\tilde{D}$ are given by

$$\tilde{A} = U^T AU,\ \tilde{E} = U^T EU,\ \tilde{B} = U^T B,\ \tilde{D} = U^T D. \tag{8}$$

Thus, the Laplace transform of the junction temperatures given by the reduced-order model of the component is

$$\tilde{T}(s) = \tilde{D}^T(\tilde{A} + s\tilde{E})^{-1}\tilde{B}u(s). \tag{9}$$

The projection matrix U must be chosen such that the input-output mapping of the reduced-order model approximates the input-output mapping of the unreduced model. In this paper, model reduction (which is basically making an appropriate choice of U) is achieved with the help of a Krylov Subspace Technique (KST) whose algorithm is explained below.

The basic idea behind KST is to create a reduced-order model such that the first few terms in its power series expansion, matches the first few terms in the power series expansion of the original (full-order) transfer function. In model reduction literature, the terms (or coefficients) in the power series expansion are also known as moments [14]. The moments are more rigorously defined as the value and subsequent derivatives of the transfer function (given in equation (4)) at $s = \sigma$, where σ is any particular frequency. In the model reduction literature, σ is commonly referred to as an *interpolation point*. The choice of σ depends on the relevant frequencies of interest in the physical problem. There can be multiple interpolation points in a typical model reduction algorithm if there are multiple bands of frequency in which we want the reduced-order model to match the full-order model. Hence KSTs ensure that once the relevant frequencies of interest are provided as inputs to the KST algorithm, the output is a reduced-order model whose first few terms (again, the number of terms is decided by the designer) in its power series expansions around those chosen frequencies matches those of the full-order model at and around those same frequencies [14]. Mathematically, if the original system has the transfer function $G(s)$ and the reduced-order model has the transfer function $\tilde{G}(s)$, then KSTs provide a transformation matrix U, that transforms the original system to the reduced system in such a way that the first j moments of the original system $G(s)$ match the first j moments of the reduced-order model $\tilde{G}(s)$. $G(s)$ and $\tilde{G}(s)$ will match better at those frequencies which are near the interpolation points and will differ at other frequencies (see Fig.2 below).

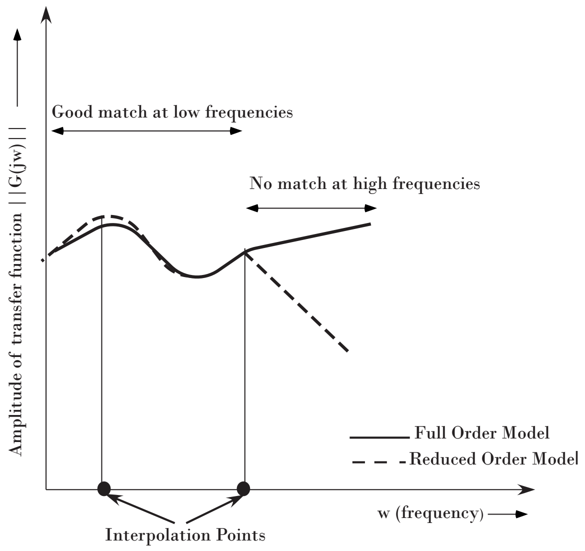


Fig. 2. Comparison of full-order model (FOM) and reduced-order model (ROM). They match at low frequencies because the interpolation points are in the low frequency range. At higher frequencies, the moments (the coefficients of the power series expansion of the transfer function) of the ROM and FOM are no longer equal and hence the behavior of the models diverge. This means that the ROM will match the FOM at low frequencies but not at higher frequencies.

In particular, the KST algorithm we used in this paper is as follows.

Algorithm 1: Krylov Subspace Reduction via Arnoldi [14]

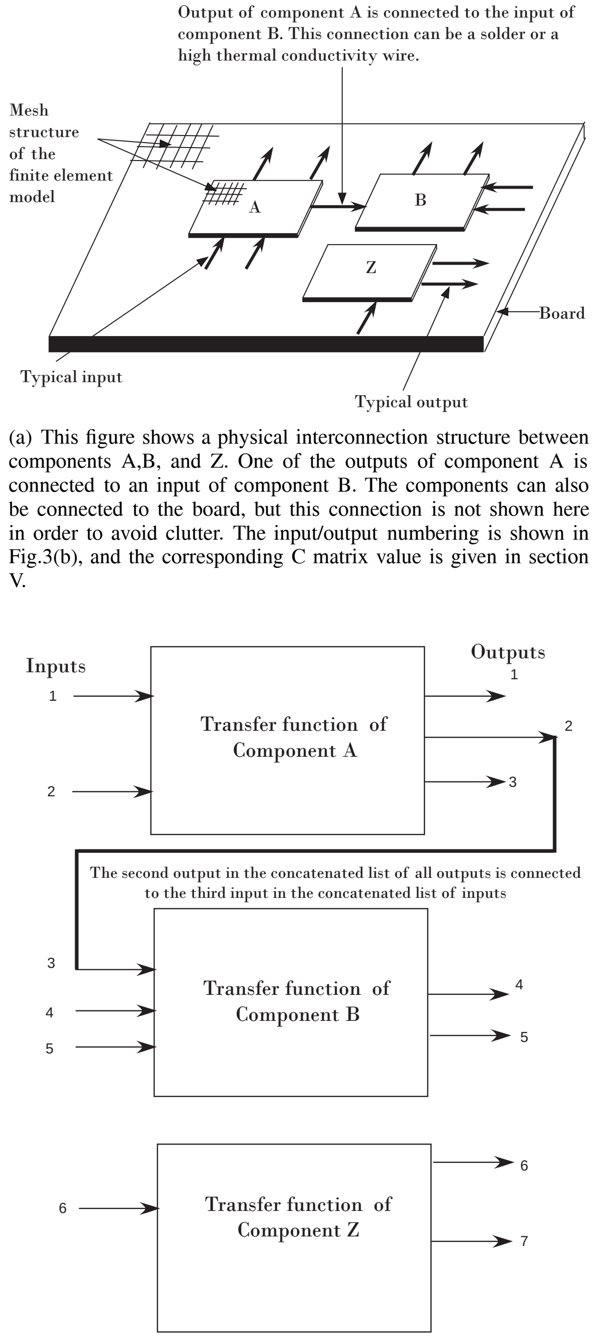
```
initialize $m = 0$, $U = [\ ]$ ($U$ is initialized as an empty matrix)
for $k = 1$ to $K$ , ($K$ is the total number of interpolation points)
  for $j_k = 1$ to $J_k$ ($J_k$ is the number of moments to be matched at the $k^{th}$ interpolation point)
    if $j_k = 1$ ,
      $\tilde{u}_m = (-A - \sigma^{(k)}E)^{-1}B$   else
      $\tilde{u}_m = (-A - \sigma^{(k)}E)^{-1}E\tilde{u}_{m-1}$
    end

    $\tilde{u}_m$ is the $m^{th}$ column of $U$.
    $\sigma^{(k)}$ is the $k^{th}$ interpolation point.

    Orthonormalize $\tilde{u}_m$ with respect to all the previously
    computed columns of $U$ to get $u_m$
    $U = [U\ \ u_m]$
    $m = m + 1$

  end
end
```

The transformation matrix U, is recursively built up by appending newly computed columns u_m, in each step of the loop. The newly added columns are orthonormalized with respect to the previously computed columns before appending them to the U matrix. Each subsequent column u_m (of the matrix U) contributes additional information about the system behavior in the reduced model. The inputs to this algorithm are the full-order model, the choice of interpolation points

$\sigma^{(k)}$, and the number of moments J_k to be matched at that interpolation point. The choice of the interpolation points will dictate the frequency range in which the reduced-order model is accurate. In the next section we will show that with an appropriate choice of interpolation points, one can in fact interconnect the reduced-order models of the various components in a state space fashion and get an accurate model of the entire device.

V. Interconnection of Reduced-Order Models

In the previous section, we have demonstrated a Krylov subspace technique for creating a reduced-order model for each component. We assume the following notation for the rest of this paper : the transfer function $g_j(s)$ of component j represents the full-order (unreduced) function that relates its inputs (heating from neighboring components and heat source in component j) to its outputs (fluxes to the neighbors adjacent to component j). The transfer function $\tilde{g}_j(s)$ is the corresponding reduced-order transfer function. The inputs of both, the reduced and unreduced models, are the same. They represent the exact same heat inputs - both, from the adjacent components as well as from their own internal heat sources. The outputs, for the reduced as well as the unreduced models, are exactly the same too. It is only the internal mapping between the inputs and outputs of an unreduced model of a component that changes in definition (and size) when compared to the internal mapping of the reduced model of the component (see equations (4) and (9)). Let $g_1(s), g_2(s), ..., g_N(s)$ be the complete (unreduced) transfer functions of the N subsystems of a system and let $\tilde{g}_1(s), \tilde{g}_2(s), ..., \tilde{g}_N(s)$ be the reduced transfer functions of the same subsystems. It is possible to represent the interconnection between the subsystems in a state space format with the help of a connection matrix C which has the following structure. Let α be a vector containing a concatenated list of all possible inputs from all subsystems and β be a concatenated list of all possible outputs from all subsystems. Then the connection matrix C depicts all the interconnections between the various subsystems. The matrix C has element $C_{pq} = 1$ if the q^{th} output β_q is connected to the p^{th} input α_p and $C_{pq} = 0$ otherwise. Physically, this means (referring to Fig.3 below) that if the second output of component A which is (for example) output 2 in the concatenated list of outputs β, is connected to the first input of component B which is (for example) input 3 in the concatenated list of inputs α, then $C_{32} = 1$. If they were not connected then $C_{32} = 0$.


Output of component A is connected to the input of component B. This connection can be a solder or a high thermal conductivity wire.
Mesh structure of the finite element model
A
B
Z
Board
Typical input
Typical output
Inputs
Outputs
1
2
3
4
5
6
7
Transfer function of Component A
Transfer function of Component B
Transfer function of Component Z
The second output in the concatenated list of all outputs is connected to the third input in the concatenated list of inputs

(a) This figure shows a physical interconnection structure between components A,B, and Z. One of the outputs of component A is connected to an input of component B. The components can also be connected to the board, but this connection is not shown here in order to avoid clutter. The input/output numbering is shown in Fig.3(b), and the corresponding C matrix value is given in section V.

(b) In the transfer function format, this figure shows that an output of A, the second in the concatenated list of outputs (of all components) is connected to an input of B, the third in the concatenated list of inputs. Hence the connection matrix C has entry C_{32}= 1. If they were not connected then entry C_{32} would be 0.

Fig. 3. Physical connection that is modeled in the interconnection matrix C and the corresponding block diagram representation.

Since the matrix C is defined solely on the basis of the interconnection of the inputs and outputs of the state space model (whether reduced or unreduced) the matrix C remains exactly the same whether we are dealing with the unreduced model or the reduced model. Referring to Fig.3, we can see that component A has 2 inputs and 3 outputs, component B has 3 inputs and 2 outputs, and component Z has 1 input and 2 outputs. Hence the concatenated list of outputs of all components has 7 elements and the concatenated list of inputs of all components has 6 elements. Hence the interconnection matrix C is a 6×7 matrix. To avoid clutter we have reduced the number of inputs, outputs, and interconnections between the components and we have also not explicitly shown the fact that each of the components actually exchange heat with (and hence is connected to) the board. In Fig.3, we have shown a connection between components A and B - an output of A (the second in the concatenated list of outputs of all components) is connected to an input of B (the third in the concatenated list of inputs of all components). Thus, element C_{32} of matrix C has value 1 and all the other elements of matrix C have value 0. Thus if we wish to represent the interconnection between the components A,B, and Z in the form of the interconnection matrix C for Fig 3(b), then

$$C = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

It is straightforward to show that this interconnection yields the complete system transfer function $F(s)$ between β (concatenated list of outputs of all components) and α (concatenated list of inputs of all components) as

$$F(s) = G(s)[I - CG(s)]^{-1}. \tag{10}$$

where

$$G(s) = \text{diag}[g_1(s), g_2(s), ..., g_N(s)]. \tag{11}$$

with $g_1(s), g_2(s), ..., g_N(s)$ being the transfer functions of the N subsystems that constitute the entire subsystem, $G(s)$ being the transfer function made by appending the transfer functions $g_1(s), g_2(s), ..., g_N(s)$ block-diagonally, and $\beta(s) = F(s)\alpha(s)$. Now, if we connect the N reduced subsystems $\tilde{g}_1(s), \tilde{g}_2(s), ..., \tilde{g}_N(s)$ in the same manner we get

$$\tilde{F}(s) = \tilde{G}(s)[I - C\tilde{G}(s)]^{-1}. \tag{12}$$

where

$$\tilde{G}(s) = \text{diag}[\tilde{g}_1(s), \tilde{g}_2(s), ..., \tilde{g}_N(s)]. \tag{13}$$

and $\beta(s) = \tilde{F}(s)\alpha(s)$.

Even if the reduced and unreduced models of the individual components match very well i.e., $g_j(s) \approx \tilde{g}_j(s)$, there is still no guarantee that the unreduced and reduced models ($F(s)$ and $\tilde{F}(s)$ respectively in equations (10) and (12) above) of the entire device will match. The reason is that if the errors between $g_j(s)$ and $\tilde{g}_j(s)$ are large enough at the dominant frequencies of the interconnected system, then these errors will multiply (in the feedback sense) in equation (12) and the interconnected system of reduced-order models might even be unstable. Hence, if we intend to interconnect the reduced subsystems, then we must make sure that the reduced subsystems are accurate at the dominant frequencies of the interconnected system. The question of how accurate the reduced subsystems must be, can in fact be specified in the form of a theorem.

This theorem relies largely on an application of the small gain theorem that can be found in a standard controls textbook [28]. Fig.4 aids in getting a better understanding of the basic idea behind the small gain theorem.

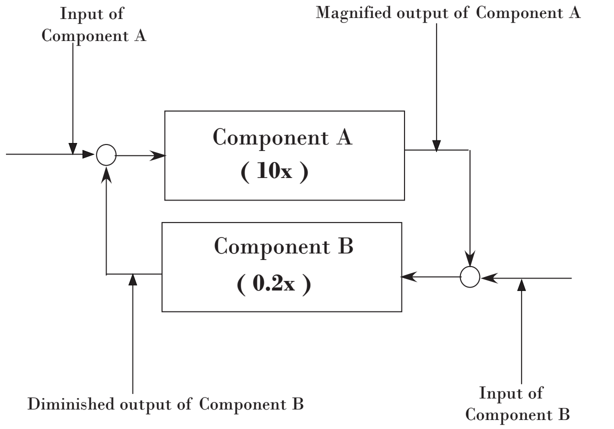


Fig. 4. Components A and B are connected in a cyclical fashion. Component A magnifies its input by 10 times in its output. Component B diminishes its input by 5 times in its output.

Suppose we have a device that is made up of two components that are connected in a cyclical fashion, i.e. each of them has exactly one input and exactly one output, with the output of the first component being connected to the input of the second component and vice versa. Suppose the first component amplifies it's input signal 10 times in its output and the second component diminishes its input signal by 5 times in its output. Then we can see that the composite device (made up of the two components) doubles any signal in a single cycle of the signal (a cycle is defined as a passage of the signal exactly once through both the components). Thus, the composite device made up of these two components is an inherently unstable system because of the net magnification of any signal that passes through it. The small gain theorem extends the above example to a device made up of multiple components that are connected to each other in an arbitrary way. It turns out that the question of stability of the device can in fact be answered for an input signal of any particular frequency. The small gain theorem explicitly states a condition (based on the transfer function of the individual components, the input signal frequency, and the interconnection matrix C) that decides whether or not the device is stable [28].

The theorem given below extends a previously derived result [21] (which itself makes use of the small gain theorem).

Theorem: Assume the subsystems $g_1(s), g_2(s), ..., g_N(s)$ are stable, that their full-order interconnection $F(s) = G(s)[I - CG(s)]^{-1}$ is stable, and the model reduced subsystems $\tilde{g}_1(s), \tilde{g}_2(s), ..., \tilde{g}_N(s)$ are also stable (the components in our model do have stable full-order and reduced-order transfer functions and the interconnection of full-order transfer functions is also stable). Then the connected system composed of reduced sub-systems, $\tilde{F}(s) = \tilde{G}(s)[I - C\tilde{G}(s)]^{-1}$ is guaranteed to be stable if

$$\|[I - CG(s)]^{-1}CE(s)\|_\infty = \sup_{\Omega} \|[I - CG(i\omega)]\|_2 < 1. \tag{14}$$

where the supremum is over all frequencies Ω and $E(s) =$

$G(s) - \tilde{G}(s)$, where G(s) and $\tilde{G}(s)$ are defined as diagonal block matrices of subsystem and full-order and reduced-order models respectively (as defined in equations (11) and (13) respectively).

We are basically concerned about reducing the infinity norm of the error between the full-order and reduced-order system, i.e. we intend to minimize $\Xi(s) = F(s) - \tilde{F}(s)$. Now, it can be shown by extending results in [21] that up to a first order approximation

$$\begin{aligned} \|\Xi(s)\|_\infty &\approx \|U(s)E(s)V(s)\|_\infty, \\ U(s) &\approx \tilde{U}(s), \ and \\ V(s) &\approx \tilde{V}(s). \end{aligned} \tag{15}$$

where
$U(s) = G(s)[I - CG(s)]^{-1}C + I$, $V(s) = [I - CG(s)]^{-1}$;
$\tilde{U}(s) = \tilde{G}(s)[I - C\tilde{G}(s)]^{-1}C + I$, $\tilde{V}(s) = [I - C\tilde{G}(s)]^{-1}$
and
$E(s) = G(s) - \tilde{G}(s)$.
Using the triangle inequality, we can show that

$$\|\Xi(s)\|_\infty \leq \sum_{p=1}^{p=N} \|U_{Ip}(s)\epsilon_p(s)V_{pJ}(s)\|_\infty. \tag{16}$$

where $U_{Ip}(s)$ and $V_{Ip}(s)$ denotes the Ip^{th} block of $U(s)$ and $V(s)$ respectively, and $\epsilon_p(s) = g_p(s) - \tilde{g}_p(s)$.

We can make use of the physics of the heat conduction problem for decreasing the time taken for the construction of the component reduced order models. The dominant poles of the system are negative and close to zero. Hence, the behavior of the full-order system is mainly determined by these dominant poles and we do not need to evaluate the full-order system's behavior in the entire frequency range. We only need to compute the dominant natural frequencies of the entire system and ensure that the reduced-order models of the subsystems are accurate in an appropriate dominant frequency range Δ (say, the low frequency range which is spanned by the two most dominant poles of the system as illustrated schematically in Fig.2).

Thus, we have to model reduce the sub-systems $g_k(s)$ in such a way so as to keep $\|U_{Ik}(s)\epsilon_k(s)V_{kJ}(s)\|_\Delta$ (the error in the dominant frequency range Δ) small for all k. Since $U(s) \approx \tilde{U}(s)$ and $V(s) \approx \tilde{V}(s)$, we can instead minimize $\|\tilde{U}_{Ik}(s)\epsilon_k(s)\tilde{V}_{kJ}(s)\|_\Delta$. In order to do this, we have to compute the entire system's frequency response, but this needs to be done only once (even a rough estimate of the frequency response of the entire system is enough). In fact, if the designer has knowledge about the dominant frequencies for a particular interconnection and if the arrangement of the components is not drastically changed for the next design iteration, then the dominant frequency range of the new interconnected system will be approximately the same as that of the previous system. In that case, the same set of reduced-order models of the components of the system may be used and interconnected with the new interconnection matrix (since the reduced-order models have already been computed in such a way that their interconnection yields an accurate behavior in the dominant frequency range). In our trials with device architectures that were not drastically different from each other we did observe a good agreement between the reduced and unreduced models of the device using just a single library of reduced-order models. In the industry, the placement of components on a board are primarily determined on the basis of VLSI design and hence the range of architectures that a heat transfer specialist can optimize over can be expected to be similar between different devices.

In the model reduction literature, ensuring that the reduced-order models are accurate at particular frequencies is termed as 'frequency weighted model reduction'. According to Krylov based frequency weighted model reduction theory [14], a rule of thumb to ensure frequency weighting is by choosing the interpolation points to be logarithmically spaced in the dominant frequency range.

In summation, the algorithm that we used to perform model reduction of the individual subsystems and interconnect the reduced-order models is given below:

Algorithm 2 (Creation and stable interconnection of reduced-order models):

1) Estimate the dominant natural frequencies of the complete (unreduced) system and choose the dominant frequency range Δ of the system based on the spacing between the eigenvalues of the system. Choose appropriate interpolation points based on Δ.

2) Using the Krylov model reduction method of section III, compute unweighted reduced-order models for each of the N subsystems. Call these initial reduced-order models as $\tilde{g}_1^0(s), \tilde{g}_2^0(s), ..., \tilde{g}_N^0(s)$.

3) Estimate $\tilde{U}(s)$ and $\tilde{V}(s)$ (refer to equations (15) and (16)), the model reduction projection matrices, based on $\tilde{g}_1^0(s), \tilde{g}_2^0(s), ..., \tilde{g}_N^0(s)$, namely $\tilde{U}^0(s) = \tilde{G}^0(s)[I - C\tilde{G}^0(s)]^{-1} + I$ and $\tilde{V}^0(s) = \tilde{G}^0(s)[I - C\tilde{G}^0(s)]^{-1}$.

4) Find $\tilde{g}_1^1(s), \tilde{g}_2^1(s), ..., \tilde{g}_N^1(s)$ by solving the frequency weighted Krylov subspace problem: min $\|\tilde{U}_{Ik}^0\epsilon(s)\tilde{V}_{kJ}^0\|_\Delta$ where the index k runs from 1 to N and $\epsilon_k(s)$ is the error transfer function between $g_k(s)$ and $\tilde{g}_k(s)$. Frequency weighting in the Krylov subspace method is done by choosing appropriate interpolation points to lie in the required frequency range. The initial choice of interpolation points is arbitrarily chosen in the dominant frequency range, but in order to find the minimum of the norm in this step, we have to vary the choice of interpolation points in the dominant frequency range so that the minimum is reached.

5) Repeat steps 3 and 4 until an acceptable set of reduced subsystems is found.

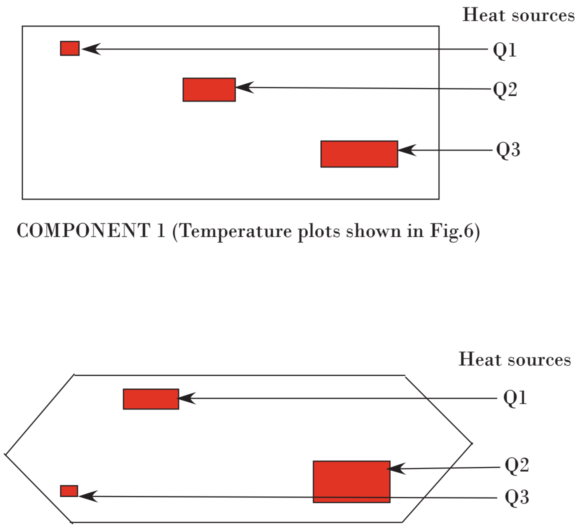
In brief, the above algorithm first estimates the dominant frequency range of the complete system, computes initial (unweighted) reduced-order models for each component and then iteratively refines these reduced-order models by solving the minimization problem mentioned in step 4 of the algorithm.

In each iteration of the algorithm, we add interpolation points for each component's reduced-order model (as mentioned before, one interpolation point in each logarithmic decade of the desired frequency range). A stopping criterion for the reduced-order model can be applied by requiring that the total error between the full-order and reduced-order system is less than a given required value. In our simulations, we needed 4 iterations for satisfactory results.

In the next section, we apply the proposed frequency weighted model reduction of the sub-systems and their interconnection mentioned above to numerical examples that have conduction heat transfer.

VI. NUMERICAL EXAMPLES

In Fig.6 and Fig.7 below, we demonstrate some results for model reduction of the heat conduction problem for a single component. The Krylov subspace algorithm (Algorithm 1) was used to create the reduced-order models for the components shown in Fig.5. They demonstrate the efficacy of algorithm 1 in reducing the number of states without losing accuracy.



COMPONENT 1 (Temperature plots shown in Fig.6)

COMPONENT 2 (Temperature plots shown in Fig.7)

Fig. 5. Component geometries and heat sources for the reduced-order modeling results shown in Fig.6 and Fig.7. For Components 1 and 2, $Q1 = 1 * 10^6\ W/m^3$, $Q2 = 3 * 10^6\ W/m^3$, $Q3 = 2.5 * 10^6\ W/m^3$.

The partial differential equation describing heat conduction is

$$\nabla \cdot (\kappa \nabla T) + Q - \rho C_p \frac{\partial T}{\partial t} = 0. \tag{17}$$

T and Q denote the temperature (in K) and the heat input (in W/m^3) respectively. The thermal conductivity κ, density ρ, and the specific heat capacity C_p of each of the two systems are constant throughout the system and have the same values as silicon at 163 $W/m.K$, 2330 kg/m^3, and 703 $J/kg.K$ respectively.

In Fig.6, we show full-order and reduced-order temperature profiles for 3 heat sources (values mentioned in the Fig.5) on a single rectangular component. The four boundaries of the component are maintained at $T = 300\ K$. The plots show the absolute temperature plot of the component above the nominal temperature of $T = 300\ K$ at the end of 3 s. They were computed by calculating the temperature rises for equally spaced mesh points of the component and interpolating for the values between these points. The reduced-order model has 30 states and took 0.01 s to compute the temperature response on a computer running 64-bit MATLAB on a 2.3 GHz AMD Opteron processor. The full-order model has 1243 states and took over 13 s to compute the temperature response.

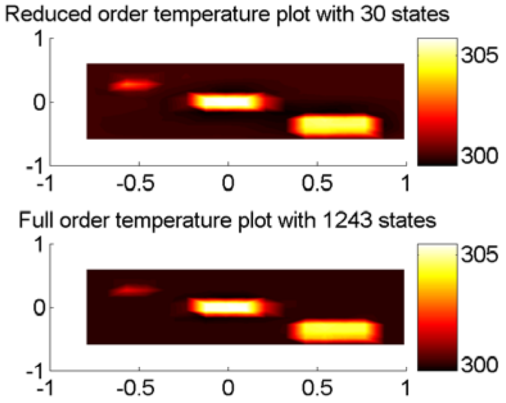


Fig. 6. Comparison of reduced and full-order temperature plots of the top (rectangular) component with the heat sources shown in Fig.5. The temperature profiles are shown for the final time $t = 3s$.

In Fig.7, we show temperature profiles for an arbitrarily shaped component having the same physical properties (silicon) as those in the above example (Fig.5(a)). The shape was chosen to demonstrate that there is no restriction on the component geometry when we create its reduced-order model. The boundaries of the component are thermally insulated. The value of the heat inputs are shown in the figure. The reduced-order model has 30 states and took 0.01 s to compute the temperature response. The full-order model has 1803 states and took over 49 s to compute the temperature response.

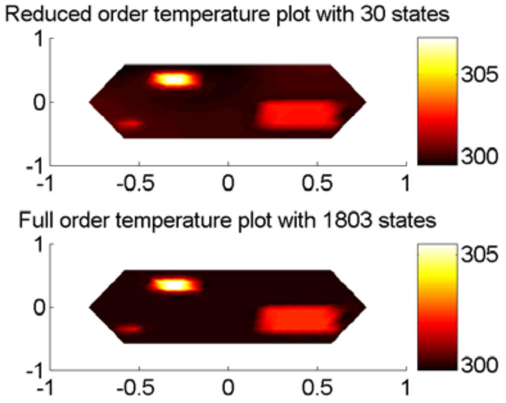


Fig. 7. Comparison of reduced and full-order temperature plots of the bottom component in Fig.5. The temperature profiles are shown for the final time $t = 3\ s$.

For each of the two components shown in Fig.5, the number of states were reduced from a full-order description of more than 1000 states to a reduced-order description of just 30 states. We computed the errors as the percentage difference between the final and reduced model's temperatures. We defined the error as the percentage difference between the temperatures computed by the FOM and ROM at a given point. The maximum error in temperature profiles was found to be below 1% for both the examples mentioned above.

We now demonstrate an example for model reduction done on a system of 20 components connected to a board. The reduced-order models for the complete system was done using Algorithm 1 (for creating ROMs of individual components) and Algorithm 2 (for creating a stable and accurate interconnection of the component ROMs). For our example, we have modeled a board with 20 components connected to it as shown in Fig.6 below.

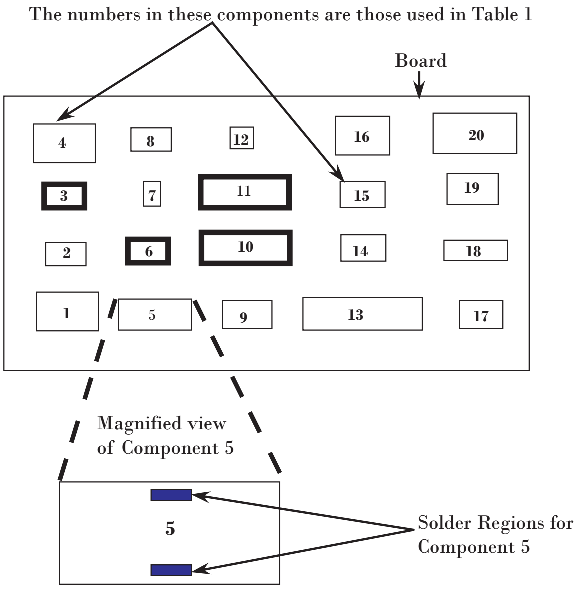


Fig. 8. Arrangement of the 20 components on the board. The components are numbered, and the corresponding step heat inputs applied to the components are mentioned in Table 1 below. The bold boundaries on components 3,6,10, and 11 denote that each of those components have all 4 of their boundaries kept at a constant temperature of 300 K.

There are 20 components on the board (shown above in Fig.8). Each of the components exchanges heat with the board and the only mode of heat transfer is conduction. Each of the components are joined to the board by two solder connections as shown in Fig.8 above. These solder connections were a cluster of 5-6 nodes near the top and bottom edges of each component. They model a conducting surface (like thermal paste) that might be typically sandwiched between the component and the board, but for this example we will term this cluster as a "solder" connection. For an actual component, the number of nodes in the cluster could be increased or decreased as per the situation. This decision should be made while building up the library of component ROMs (and before interconnecting the components). Each cluster in the solder region of the component exchanges heat with an adjacent cluster in the solder region of the board. All the nodes in each solder cluster of a component were modeled so as to exchange the same heat flux with the solder cluster of the adjacent component. This cluster is an idealization of a thermal contact between a component and the board. The heat flux for an entire cluster (solder contact) is modeled as just one input in the state space format and this kind of contact between the board and the component was captured as an input-output relationship with the connection matrix C explained in Eqn.10. This kind of interconnection between the components and the board reduced the effective number of inputs to just two for each component in addition to the source that generates heat in the component (which amounts to 3 inputs per component).

Each of the idealized components as well as the board were modeled as being made of silicon and having constant material properties that were independent of temperature. The physical properties of all the components and the board are the same as silicon (as mentioned for the examples in Fig.5). Components 3,6,10 and 11 (labeled in Fig.8) have each of their 4 boundaries at a constant temperature of 300 K. The boundaries of the rest of the 16 components as well as the board, are thermally insulated.

The interconnections between the components (subsystems) were modeled in the following way. The value of the system matrices A,B,E, and D (refer Eqn.1 and Eqn.2) for each of the sub-systems was individually extracted from FEMLAB (a finite element solver). Within FEMLAB, a linear time-dependent solver was used for simulating this problem. Each of the components was discretized with mesh sizes of the order of around 1000 nodes per component. We extracted complete state space models of each of the components with the inputs being the heat fluxes into each component. The state space matrices were obtained by linearizing the finite element model around a nominal temperature of 300 K. Though we created a reduced-order model for each component, we did not create a reduced-order model for the board. The reason is that when we initially connected the reduced-order model of the board to the reduced-order model of the components, the whole system's temperature response showed an increased amount of error (around 7%) in certain areas on the board. We consider this to be a result of an inherent numerical limitation in the Krylov subspace method (Algorithm 1), which to the best of our knowledge has not yet been resolved. A good explanation of this limitation can be found in [14]. Basically, in each inner loop of Algorithm 1, we add more information about the full-order model into the transformation matrix U, by appending new columns u_m to U. Now, if there are many inputs in the state space model (i.e. a B is a "fat" matrix, like in the state space model of the board in our numerical example, which has 40 inputs, 2 from each component connected to it), the new columns (u_m) that are added to U, in successive iterations of the Krylov subspace algorithm (Algorithm 1) can, because of limitations in computational accuracy, lie in the subspace of the previously computed columns of U. This makes the matrix U have less than full rank. When such a (less than full rank) transformation matrix is used for reduced-order modeling, it results in inaccurate reduced-order models.

Hence, we chose to connect the full-order model of the board to the reduced-order model of the components and avoid the above problem. We would like to stress that using a full-order model of the board has a one time, fixed computational cost. It is basically the total number of components on the board that influences the total number of states (and hence, computational cost) in the interconnected system reduced model. Specifically, in our example the full-order model of the board has 1429 states and the full-order model of each of the 20 components has around 1000 states and thus the full-order model of the interconnected system has 23109 states. The reduced-order model of each of the 20 components has 30 states. Thus even though, we connect the reduced-order models of the components to the full-order model of the board, the reduced-order model of the interconnected system has only 2029 states (20 x 30 + 1429). As the number of components on the board increases, the difference between the number of states in the FOM and ROM of the interconnected system will roughly scale by the total number of components.

We connected the state space systems of the components on MATLAB (with the connection matrix C) and computed the dominant frequency range of the entire system. The dominant eigenvalues were clustered in the 0-10 rad/s range (this is step 1 of Algorithm 2). We chose to have 3 interpolation points at 0.1 rad/s, 2 rad/s and 5 rad/s as the initial guesses for the interpolation points (with 1 moment to be matched at each interpolation point) and allowed them to vary in the dominant frequency range as mentioned in Algorithm 2.

In our entire simulation, we required 4 iterations of Algorithm 2 which took close to 40 minutes on a 2.3 GHz AMD Opteron processor (with 64 bit MATLAB). The final interpolation points were chosen as $\sigma_1 = 1$ rad/s, $\sigma_2 = 10^{-1}$ rad/s, $\sigma_3 = 10^{-2}$ rad/s, $\sigma_4 = 10^{-3}$ rad/s, and $\sigma_5 = 10^{-4}$ rad/s. After the reduced-order models were computed, it took less than 5 minutes to connect the reduced-order models of the 20 components (and the full-order model of the board) to form reduced-order models of the entire system. For ease of notation, the step (heat) inputs that have been applied to the components (labeled in Fig.8) have been denoted as follows : $q_i = u_i \cdot \mathbf{1}(t - \alpha_i)$ which denotes that $q_i = 0$ for $t \leq \alpha$, and $q_i = u_i$ for $t > \alpha$. The heat inputs applied all 20 components are listed in Table 1.

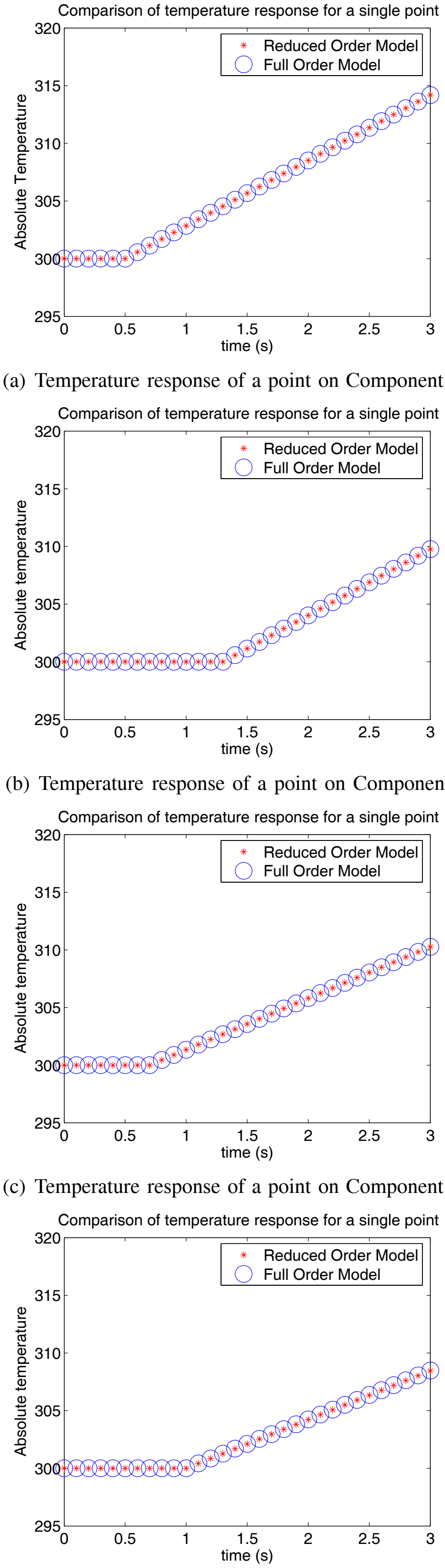
TABLE I
HEAT INPUTS APPLIED TO THE 20 COMPONENTS

Component	u_i (10^{-6} W/m^3)	α_i (s)
1	9	1.2
2	2	1.5
3	7	2.2
4	6.4	0.9
5	8.8	1.3
6	4.3	1.3
7	8.1	1.4
8	5.6	2.7
9	7.7	0.4
10	5.0	2.2
11	8.4	2.4
12	7.7	0.8
13	4.9	0.9
14	8.5	0.3
15	6.2	0.9
16	8.9	0.4
17	9.7	0.04
18	3.6	0.1
19	9.1	0.4
20	7.0	0.6

The resulting temperature response at 4 points of the system are shown in Fig.9 below. Each plot shows results for the full-order as well as the reduced-order model. The plots for the FOM and ROM response are indistinguishable because the errors are very small.

In Fig.10, we have provided a plot of the temperature profile for the entire system (components connected to the board). Fig.10(a) and Fig.10(b) show the reduced-order plots of the board and the components respectively. Fig.10(c) and Fig.10(d) show the full-order plots of the board and the components respectively. The plots of the temperature profile for the 20 components are shown separately (below) the plot of the temperature profile for the board, because the temperature rise in the components is higher than that of the board. The full-order model computations are very expensive (in terms of memory involved), because of which we could not use the "append" and "connect" commands in Matlab for creating the interconnected system model. For the reduced-order models we were able to use the "connect" and "append" commands for creating the interconnected system model, and this required 340 MB memory (once again, this memory usage is largely due to the *full-order model of the board*, which is being connected to the reduced-order models of the 20 components). For creating the interconnected system model with the full-order systems, we had to make use of the sparse structure of the system matrices (see Eqn.12 and Eqn.13), which are mostly block diagonal except for the few off block-diagonal terms corresponding to the solder connections. Due to this sparse (and almost block diagonal) structure of the system matrices, we were able to interconnect the individual state space matrices of the components to that of the board without using the "append" and "connect" commands. The plot was created by computing the temperature for a few evenly spaced points with the respective (full or reduced) order models in MATLAB, and interpolating for the values in between.

For each of the components, we can see that most of the component is at roughly the same temperature except for the two solder regions near the top and bottom of the component where the component exchanges heat flux with the component. The solder regions are cooler than the rest of the component because the heat flux flows out of the component and into the board. In the temperature plots of the board, one can see the regions that are supposed to be below the component have a higher temperature then the surrounding. For the initial set of 3 interpolation points (before Algorithm 2 was applied for interconnecting the board to the components), the reduced-order plots of the board temperature did not match that of the full-order plot of the board. However, one can see that in Fig.10 (after Algorithm 2 was used, and the number of interpolation points for the component ROMs was increased to 5), there is a very accurate match between the contours of the hot spots on the board on the full-order model and the contours on the reduced-order model. We computed the errors as the percentage difference between the full and reduced model's temperatures. The maximum error of the temperature response on various parts of the device was less than 1%. The number of states for the entire device was reduced from an original of around 23109 to 2029. In both, the full-order and



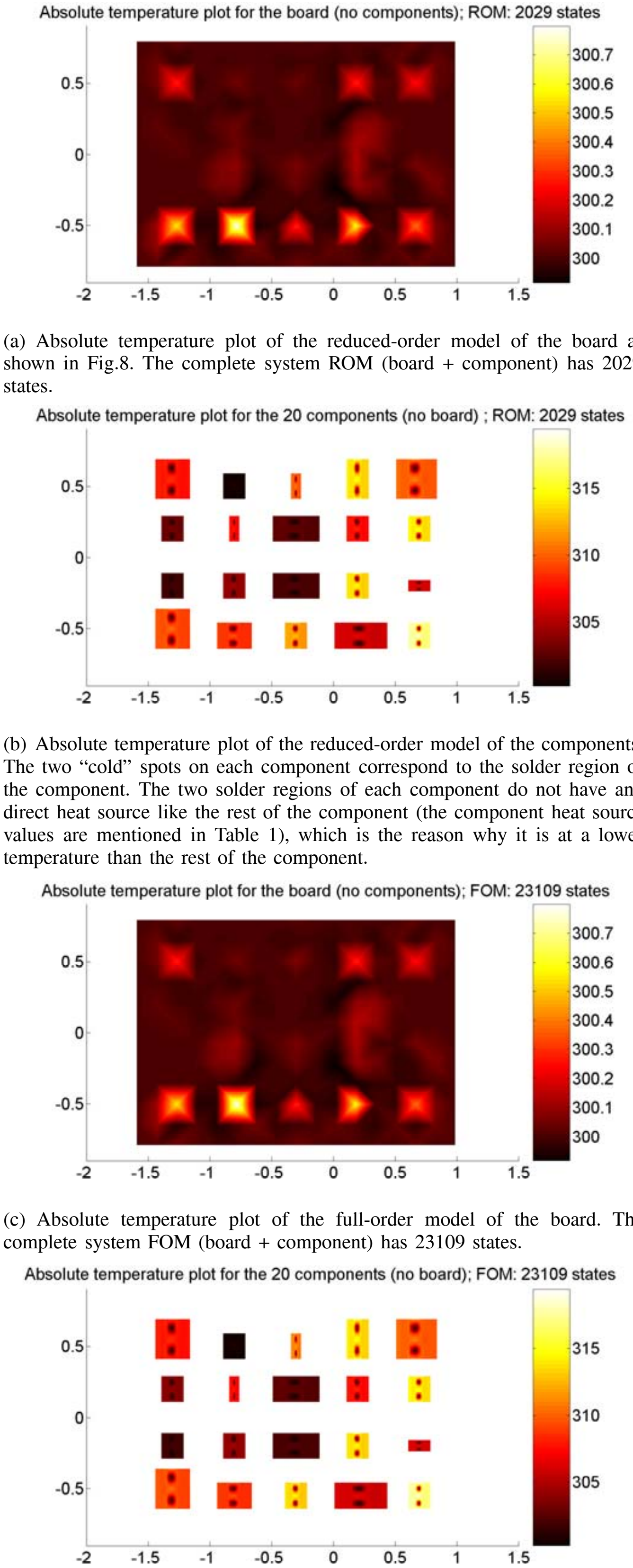
(a) Temperature response of a point on Component 16.

(b) Temperature response of a point on Component 1.

(c) Temperature response of a point on Component 20.

(d) Temperature response of a point on Component 4.

Fig. 9. Comparison between the full system and reduced system temperature response of the system (components and the board)from $t = 0s$ until $t = 3s$.



(a) Absolute temperature plot of the reduced-order model of the board as shown in Fig.8. The complete system ROM (board + component) has 2029 states.

(b) Absolute temperature plot of the reduced-order model of the components. The two "cold" spots on each component correspond to the solder region of the component. The two solder regions of each component do not have any direct heat source like the rest of the component (the component heat source values are mentioned in Table 1), which is the reason why it is at a lower temperature than the rest of the component.

(c) Absolute temperature plot of the full-order model of the board. The complete system FOM (board + component) has 23109 states.

(d) Absolute temperature plot of the full-order model of the components.

Fig. 10. Comparison between the full system and reduced system temperature profiles of the system (components and the board) at the end of $3s$.The components are shown separately from the board because the temperature rise in the components are much larger than on the board.

the reduced-order system, the board contributes 1429 elements. In the reduced-order system, each of the components have only got 30 states as compared to more than 1000 states in the full-order model. Though there is a fixed cost in the number of board elements, we can see that as the number of components on the board increase, the savings in computational time and memory are very significant in the reduced-order model of the system when compared to the full-order model.

Table 2 presents a comparison of modeling techniques that have been used for thermal simulation of microelectronic components. In our simulations, the initial iterations - which consists of computing the frequency weighted reduced-order model for all the components - took 40 minutes. The connection of the reduced-order models of the components took less than 5 minutes, and after that, computing the temperature responses took 55 seconds. Hence, the temperature distribution for new system architectures can be computed in 5 min 55 seconds. For the connection structure in our problem, the error was less than 1% as compared to the full-order system. The main advantage of our models is that we can interconnect many reduced-order models instead of having to compute a reduced-order model of the entire system for each different component layout.

TABLE II

COMPARISON OF MODEL REDUCTION TECHNIQUES

Modeling method	FOM/ROM number of elements (physical problem)	ROM run time	Error (compared to)	Recreate device ROM from ground up ?
Resistor modeling [9]:	ROM:2171 (9 components; 12 varying heat sources)	25 min	$<$ 7% (ANSYS)	YES
Pade approximation [29]	ROM:2375 (Voltage Regulator)	240s (as compared to 3110 s for FOM)	Almost exact (FOM)	YES
Arnoldi based single ROM [19]	FOM:30000 ROM:85 (17 components on a board)	10 min	$<$ 1% (FOM)	YES
Algorithm 2 of this paper	FOM:23109 ROM:2029 (20 components on a board)	$<$ 1 min after the initial interconnection of the system	$<$ 1% (FOM)	NO (refer to the discussion in section VII)

VII. DISCUSSION

In this paper we have been able to successfully demonstrate the use of an efficient Krylov subspace method to create reduced-order models of different components of a thermal conduction network of an electronic device. We also described a method of assembling various such reduced-order models in order to accurately compute the entire system behavior. The size of the system in the numerical example was reduced from 23109 states to approximately 2029 states, which reduced the time for the simulation to less than 1 minute (once the initial iterations for computing the frequency weighted reduced-order models of the components have been completed). A designer can use the method that we propose in this paper to create a library of reduced-order component models and then connect them together to achieve reduced-order system models.

The mathematical format for interconnection that we presented in this paper can be effectively applied to model any kind of physical interconnection between components. In a finite element package like ANSYS or FEMLAB, the kind of reduced-order results described in this paper can be offered as an option to the designer. Once the designer has modeled a particular component of the device, he or she can store the reduced-order description of the component in a library along with those of other components. The designer can then connect that library of components to create cheap but accurate system models.

The designer would find it convenient if there was a GUI incorporated in the software that would help him or her connect the (heat) inputs of a particular component to those of another component, instead of having to manually compute the interconnection matrix C. When faced with a new interconnection structure, the designer should first check whether the existing library contains component reduced-order models that satisfy the stopping criterion in algorithm 2. If it does meet the criterion, then the designer can use the same library, else he or she can use the existing library as a starting guess and use Algorithm 2 to compute new component reduced-order models. In Algorithm 2, one also needs to have an approximate idea of the dominant frequency range of the system model for different interconnections. One can potentially use a result from linear algebra to do this cheaply. Briefly, the idea is that if one has access to the dominant eigenvalues of the component reduced-order models that are being interconnected, the system's state space matrix A_{system} (which has only got a few *known* off block-diagonal terms corresponding to the entries due to the solder joints) can be treated as a perturbation of a matrix that is formed by block-diagonally appending the component matrices, A_i. Since we only need a rough approximation of the dominant frequency range, this can be computed by applying Gershgorin's theorem [30] to the block-diagonal matrices ($A_i, i = 1$ to N) in order to get the range in which the dominant eigenvalues will lie. Even a rough approximation of this range will suffice, because when creating ROMs of the component, it is enough to add interpolation points that are logarithmically spaced in the dominant frequency range.

In some of our simulations, we have modified the interconnection structure slightly and observed that one can still use the same library with accurate results, but there is still an open question: how much we can modify the interconnection structure without having to also modify the library of reduced-order component models (see section V , which discusses how the creation of component reduced-order models depends on the interconnection matrix C)?

The final aim is to couple these kinds of conduction ROMS to convection and radiation ROMs (convection can be model reduced using proper orthogonal decomposition) and provide a complete design capability for a heat transfer specialist using a similar 'reduce-then-interconnect' approach. We think that the answers to the above two questions are important topics for future research in order that designers apply model reduction techniques to practical problems.

Acknowledgment

The authors would like to thank the reviewers and Professor Yogendra Joshi at Georgia Tech for their valuable guidance.

References

[1] K.T. Yang, "Natural convection in enclosures," in Handbook of Single-Phase Convective Heat Transfer, S.Kakac, R.K.Shah, and W.Aung, Eds, New York : John Wiley and Sons, 1988, Chapter 13.

[2] R.J. Moffat and A. Ortega, "Direct air cooling of electronic components," Advances in Thermal modeling of thermal Components and Systems, Vol.1, A.Bar-Cohen and A.D.Krauss, Eds, Hemisphere New York, pp.129-282.

[3] G.P. Peterson and A. Ortega, "Thermal control of electronic equipment and devices," Advances in Heat Transfer, Vol.20, J.P. Hartnett and T.F. Irvine JR., Eds, Academic Press San Diego, 1990, pp.181-314.

[4] E. Papanicolau and S. Gopalakrishna, "Natural convection in shallow, horizontal air layers encountered in electronic cooling," ASME Journal of Electronic Packaging, Vol.117, 1993, pp. 307-316.

[5] D.W.Larson and R. Viskanta, "Transient combined free air convection and radiation in a rectangular enclosure," J.Fluid Mech., Vol.78, 1976, pp.65-85.

[6] C.J.M. Lasance, " The conceivable accuracy of experimental and numerical thermal analysis of electronic systems," IEEE Transactions on Components and Packaging Technologies, Vol.25, No.3, Sep 2002, pp.366-382.

[7] M. Clemens, E. Gjonaj, P.Pinder and T. Weiland, "Self-consistent simulations of transient heating effects in electrical devices using the Finite Integration Technique," IEEE Trans. Magnetics, pt.1, vol.37, pp.3375-3379, Sept 2001.

[8] J.T. Hsu and L. Vu-Quoc, "A rational formulation of thermal circuit models for electro-thermal simulation - Part 1: Finite Element Method," IEEE Trans. Circuits Syst., vol.43, pp.721-732, pp.721-732, Sept. 1996.

[9] M.-N.Sabry, et al., "Realistic and Efficient Simulation of Electro-thermal Effects in VLSI Circuits," IEEE Trans. VLSI Systems, Vol. 5, No. 3, 1997, pp. 283-289.

[10] Y.C. Gerstenmaier and G. Wachutka, "Time dependent temperature fields calculated using eigenfunctions and eigenvalues of the heat conduction equation," in Proc. 6th THERMINIC Workshop, Sept.24-27, 2000, pp.55-61.

[11] W. Batty, C.E. Christofferson, S. David, A.J. Panks, R.G. Johnson, C.M. Snowden and M.B. Steer, "Steady state and transient electro-thermal simulation of power devices and circuits based on fully physical thermal model," in Proc. 6th THERMINIC Workshop, Sept.24-27, 2000, pp.125-130.

[12] V. Szekely, "Identification of RC networks by deconvolution : Chances and Limits," IEEE Trans. CAS 1, vol. 45, Mar.1998, pp.244-258.

[13] B.C. Moore, "Principal component analysis in linear systems: controllability, observability and model reduction", IEEE Transactions in Automatic Control, AC-26, pp.17-32, 1981.

[14] E.J. Grimme, "Krylov projection methods for model reduction," Ph.D. Thesis, ECE Dept., U. of Illinois, Urbana-Champaign, (1997).

[15] L. Sirovich, "Turbulence and dynamics of coherent structures, parts I - III," Quarterly of Applied Mathematics, vol. 45, pp. 561-590, 1987.

[16] C. W. Rowley, T. Colonius, and R. M. Murray, "POD based models of self sustained oscillations in the flow past an open cavity," AIAA, Ed., 2000.

[17] E. Chiprout, "Model reduction for large circuit simulation", Institute for Mathematics and its Applications (IMA), Univ. of Minnesota, Sept. 1997.

[18] I. Mezic and S. Narayan, "Overview of some theoretical and experimental results on modeling and control of shear flows," Proc. Of 39th IEEE Conference on Decision and Control, Sydney, Australia, Dec.2000, pp.1709-1715.

[19] L. Codecasa, D. D'Amore and P. Maffezzoni, "An Arnoldi based thermal network reduction method for electro-thermal analysis," IEEE Transactions on Components and Packaging Technologies, Vol.26, No.1, March 2003, pp.186 - 192.

[20] Y.C.Liang, W.Z. Lin, H.P. Lee, S.P. Kim, K.H.Lee and H.Sun, "Proper orthogonal decomposition and its applications - Part 2: Model Reduction for MEMS Dynamical Systems," Journal of Sound and Vibration (2002), 256(3), pp. 515-532.

[21] B.D.O. Anderson, "Controller reduction- concepts and approaches," IEEE Transactions in Automatic Control, vo.l34, Aug 1989, pp.802-812.

[22] W. Kreuger and A. Bar-Cohen, "Thermal characterization of a PLCC-expanded Rjc methodology," IEEE Trans. Comp.,Hybrids Manufact. Technol.,vol.15, pp.691-698, May 1992.

[23] C.J.M. Lasance, H. Vinke, and H. Rosten, "Thermal characterization of electronic devices with boundary condition independent compact models," IEEE Trans. Comp.,Packag., Manufact. Technol., vol.18, no.4, pp.723-731, 1995.

[24] D.S. Boyalakuntla and J.Y. Murthy, "COBRA-Based compact models for simulation of electronic chip packages," Proc. Interpack., July 2001, IPACK2001-15 534.

[25] FEMLAB Reference Manual 2002.

[26] K. Ogata, "Modern Control Engineering," Prentice Hall India, 3rd Edition, 1998.

[27] F.P. Incropera and D.P. Dewitt, "Introduction to Heat Transfer," Wiley, 4th Edition, 2001.

[28] G.E. Dullerud and F. Paganini, "A course in Robust Control Theory," Springer 2000.

[29] S.S. Lee and D.J. Allstot, "Electrothermal simulation of Integrated Circuits," IEEE J. Solid-State Circuits, Vol.28, No.12, Dec. 1993, pp. 1283-1293.

[30] L.N. Trefethen, D.Bau III, "Numerical Linear Algebra," Society of Industrial and Applied Mathematics (SIAM), 1997.

Pramod Mathai was born in Bombay, India in 1979. He received the B.Tech degree in Aerospace Engineering from the Indian Institute of Technology (IIT), Bombay, India in 2001. He is currently a doctoral candidate at the University of Maryland (UMD), College Park, MD. His interests lie in control theory and in reduced-order modeling of complex systems.

Benjamin Shapiro was born in Jerusalem, Israel in 1973. He received the B.S. degree in Aerospace Engineering from Georgia Institute of Technology (Georgia Tech), Atlanta, GA in 1995, and the Ph.D. degree in control and dynamical systems from California Institute of Technology (Caltech), Pasadena, CA in 1999. He joined the Department of Aerospace Engineering at the University of Maryland (UMD), College Park, MD as an Assistant Professor in 2000 and is now an Associate Professor. His main interests are in modeling and control of microsystems, with a primary focus on modeling and control of microfluidic systems for biochemical applications. He is part of the interdisciplinary Small Smart Systems Center and is affiliated with the Bioengineering Graduate Program and the Applied Math And Scientific Computation Program at the University.

Dr. Shapiro was the organizer of the March 2004 NSF workshop on Control and System Integration of Micro- and Nano-Scale Systems and is a recipient of a 2003 NSF CAREER award.