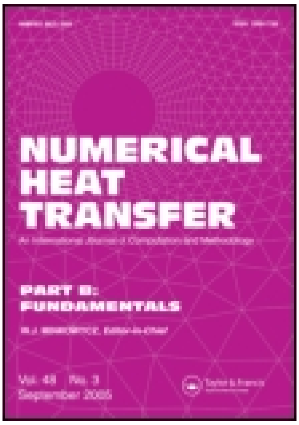


Numerical Heat Transfer, Part B: Fundamentals: An International Journal of Computation and Methodology

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/unhb20

PARALLEL DOMAIN DECOMPOSITION APPROACH FOR LARGE-SCALE THREE-DIMENSIONAL BOUNDARY-ELEMENT MODELS IN LINEAR AND NONLINEAR HEAT CONDUCTION

Eduardo Divo [a] , Alain J. Kassab [b] & Franklin Rodriguez [b]

[a] Department of Engineering Technology, University of Central Florida, Orlando, Florida 32816-2450, USA

[b] Mechanical, Materials, and Aerospace Engineering Department, University of Central Florida, Orlando, Florida, USA

Published online: 02 Feb 2011.

To cite this article: Eduardo Divo , Alain J. Kassab & Franklin Rodriguez (2003) PARALLEL DOMAIN DECOMPOSITION APPROACH FOR LARGE-SCALE THREE-DIMENSIONAL BOUNDARY-ELEMENT MODELS IN LINEAR AND NONLINEAR HEAT CONDUCTION, Numerical Heat Transfer, Part B: Fundamentals: An International Journal of Computation and Methodology, 44:5, 417-437

To link to this article: http://dx.doi.org/10.1080/716100489

Numerical Heat Transfer, Part B, 44: 417–437, 2003
Copyright © Taylor & Francis Inc.
ISSN: 1040-7790 print/1521-0626 online
DOI: 10.1080/10407790390235622

PARALLEL DOMAIN DECOMPOSITION APPROACH FOR LARGE-SCALE THREE-DIMENSIONAL BOUNDARY-ELEMENT MODELS IN LINEAR AND NONLINEAR HEAT CONDUCTION

Eduardo Divo
Department of Engineering Technology, University of Central Florida, Orlando, Florida, USA

Alain J. Kassab and Franklin Rodriguez
Mechanical, Materials, and Aerospace Engineering Department, University of Central Florida, Orlando, Florida, USA

The boundary-element method (BEM) requires only a surface mesh to solve linear and nonlinear heat conduction problems, but the resulting matrix is fully populated. This poses serious challenges for large-scale three-dimensional problems due to storage requirements and iterative solution of a large set of nonsymmetric equations. In this article, we develop a domain decomposition, or artificial subsectioning technique, along with a region-by-region iteration algorithm particularly tailored for parallel computation to address these issues. A coarse-surface grid solution coupled with an efficient physically based procedure provides an effective initial guess for a fine-surface grid model. The process converges very efficiently, offering substantial savings in memory. The iterative domain decomposition technique is ideally suited for parallel computation. We discuss its implementation on a modest Windows XP Pentium P4 PC cluster running under MPI with MPI2 extensions. Results from three-dimensional BEM heat conduction models including models of upwards of 85,000 nodes arising form an intricate film-cooled vane. We demonstrate that the BEM can readily be applied to solve large-scale linear and nonlinear heat conduction problems and that such solutions can be readily undertaken on modest PC clusters.

INTRODUCTION

The boundary-element method (BEM) requires only a surface mesh to solve a large class of field equations, and, further, the nodal unknowns appearing in the BEM equations are the surface values of the field variable and its normal derivative. Thus, the BEM lends itself ideally not only to the analysis of field problems, but also to modeling coupled field problems such as those arising in conjugate heat transfer (CHT). The authors have developed a so-called temperature-forward flux-back

Received 3 April 2003; accepted 2 May 2003.

This research is supported by grant NAG3-2691 from NASA-Glenn.

Address correspondence to Alain J. Kassab, University of Central Florida, Department of Mechanical and Aerospace Engineering, Box 162450, Orlando, FL 32816-2450, USA. E-mail: kassab@mail.ucf.edu

(TFFB), loosely coupled boundary-element, finite-volume method [1, 2] for CHT analysis of turbomachinery components such as film-cooled blades. However, in implementing the BEM for such intricate 3-D structures, the number of surface unknowns required to resolve the temperature field can readily number in the tens to hundreds of thousands. Since the ensuing matrix equation is fully populated, this poses a serious problem both in terms of storage requirements as well as in the need to solve a large set of nonsymmetric equations.

The BEM community has generally approached this problem by: (1) artificial subsectioning the 3-D model into a multiregion model, an idea originated for piecewise nonhomogeneous media [3], in conjunction with block solvers reminiscent of finite-element method (FEM) frontal solvers [4, 5] or iterative methods [6–9], and (2) fast-multipole methods adapted to BEM coupled to a General Minimization of Residuals (GMRES) nonsymmetric iterative solver [10, 11]. The first approach is readily adapted to existing BEM codes, while the multipole approach, although very efficient, requires rewriting of existing BEM codes. Recently, a technique using wavelet decomposition has been proposed to compress the BEM matrix once it is formed and stored, in order to accelerate the solution phase without major alteration of traditional BEM codes [12].

In this article, we develop a parallel BEM algorithm to solve large-scale, three-dimensional, steady nonlinear heat conduction problems, and we allow for multiple regions of different nonlinear conductivities. We adopt domain decomposition, or the artificial multiregion subsectioning technique, along with a region-by-region iteration algorithm tailored for parallel computation [13]. We formulate a nonsymmetric update of the interfacial fluxes to ensure equality of fluxes at the subdomain interfaces. In order to provide a sufficiently accurate initial guess for the iterative process, we derive a physically based initial guess for the temperatures at the domain interfaces, and we use a coarse-surface grid solution obtained with constant elements. The results of the constant grid model serves as an initial guess for finer discretizations obtained with bilinear and biquadratic boundary-element models. The process converges very efficiently, offers substantial savings in memory, and does not require complex data structure preparation required by block solvers or the multipole approaches. Moreover, we show that the process converges for both linear and nonlinear cases, the latter treated using the classical Kirchhoff transform. We present results from a series of numerical examples which first validate the approach against an exact solution, then consider increasingly intricate BEM heat conduction models including a plenum-cooled and a film-cooled turbine blade.

GOVERNING EQUATION AND BOUNDARY INTEGRAL EQUATION

The governing equation under consideration is the steady-state heat nonlinear conduction equation

$$\nabla \cdot [k(T)\nabla T] = 0 \tag{1}$$

where T denotes the temperature and k is the thermal conductivity of the material. If the thermal conductivity is taken as constant, then the above reduces to the Laplace equation for the temperature,

$$\nabla^2 T = 0 \tag{2}$$

When the thermal conductivity variation with temperature is an important concern, the nonlinearity in the steady-state heat conduction equation can readily be removed by introducing the classical Kirchhoff transform, $U(T)$ [14–16],

$$U(T) = \frac{1}{k_o}\int_{T_o}^{T} k(T)\, dT \tag{3}$$

where T_o is the reference temperature and k_o is the reference thermal conductivity. The transform and its inverse are readily evaluated, either analytically or numerically. As the transform U is nothing but the area under the k–T curve, it is a monotonically increasing function of T, and the back-transform $T(U)$ is unique. The heat conduction equation then transforms to a Laplace equation for the transform parameter $U(T)$,

$$\nabla^2 U = 0 \tag{4}$$

and as long as the boundary conditions are of the first or second kind the problem is linear [16] as

$$\begin{gathered} T\,|_{\mathrm{r}_s} = T_s \rightarrow U\,|_{\mathrm{r}_s} = U(T_s) = U_s \\ -k\frac{\partial T}{\partial n}\bigg|_{\mathrm{r}_s} = q_s \rightarrow -k_o\frac{\partial U}{\partial n}\bigg|_{\mathrm{r}_s} = q_s \end{gathered} \tag{5}$$

Here r_s denotes a point on the surface. In the case of boundary conditions of the third kind, the problem is nonlinear in the boundary condition

$$-k\frac{\partial T}{\partial_n}\bigg|_{\mathrm{r}_s} = h_s[T|_{\mathrm{r}_s} - T_\infty] \rightarrow -k_o\frac{\partial U}{\partial n}\bigg|_{\mathrm{r}_s} = h_s[T(U|_{\mathrm{r}_s}) - T_\infty] \tag{6}$$

and iteration is required. This is accomplished by rewriting the convective boundary condition as

$$-k_o\frac{\partial U}{\partial n}\bigg|_{\mathrm{r}_s} = h_s[U|_{\mathrm{r}_s} - T_\infty] - T_\infty] + h_s[T(U|_{\mathrm{r}_s}) - U|_{\mathrm{r}_s}] \tag{7}$$

and first solving the problem with the linearized boundary condition

$$-k_o\frac{\partial U}{\partial n}\bigg|_{\mathrm{r}_s} = h_s[U|_{\mathrm{r}_s} - T_\infty] \tag{8}$$

to provide an initial guess for iteration.

The heat conduction equation thus reduces to the Laplace equation in any case. We will henceforth use T as the dependent variable with the understanding that when dealing with nonlinear problems, $T \rightarrow U$. The Laplace equation is readily solved by first converting into a boundary integral equation (BIE) [14, 15],

$$C(\xi)T(\xi) + \oint_S q(x)G(x,\xi)\, dS(x) = \oint_S T(x)H(x,\xi)\, dS(x) \tag{9}$$

where $S(x)$ is the surface bounding the domain of interest, ξ is the source point, x is the field point, $q(x) = -k\,\partial T(x)/\partial n$ is the heat flux, $G(x,\xi)$ is the fundamental solution, and $H(x, \xi) = -k\ \partial G(x, \xi)/\partial n$. The fundamental solution is the response of the adjoint governing differential operator at any field point x due to a Dirac delta function acting at the source point ξ, and is given by $G(x, \xi) = 1/4\pi T(x, \xi)$ in three dimensions, where $r(x, \xi)$ is the Euclidean distance from the source point ξ. The free term $C(\xi) = \oint_{S(x)} H(x, \xi)\ dS(x)$ can be shown analytically to be the internal angle subtended at source point divided by 4π when ξ is on the boundary and 1 when ξ is at the interior. In the standard BEM, polynomials are employed to discretize the boundary geometry and distribution of the temperature and heat flux on the boundary. The discretized BIE is usually collocated at the boundary points, leading to the algebraic analog of Eq. (9),

$$[H]\{T\} = [G]\{q\} \tag{10}$$

These equations are readily solved upon imposition of boundary conditions. We use subparametric constant, isoparametric bilinear, and superparametric biquadratic, discontinuous boundary elements as the basic elements in our 3-D BEM code; these are illustrated in Figure 1. Such elements avoid the so-called star-point issue and allow for discontinuous fluxes. Moreover, the biquadratic elements we use are superparametric quadratic elements, bilinear model of the geometry, and biquadratic model of the temperature and heat flux, because this provides compatibility of geometric models with grids generated by structured finite-volume grid generators.

EXPLICIT DOMAIN DECOMPOSITION

In a standard BEM solution process, just outlined, if N is the number of boundary nodes used to discretize the problem, the number of floating-point operations (FLOPS) required to generate the algebraic system is proportional to N^2. Direct memory allocation is also proportional to N^2. Enforcing imposed boundary conditions yields

$$[H]\{T\} = [G]\{q\} \quad \Rightarrow \quad [A]\{x\} = \{b\} \tag{11}$$

where $\{x\}$ contains nodal unknowns T or q, whichever is not specified in the boundary conditions. The solution of the algebraic system for the boundary unknowns can be performed using a direct solution method such as LU decomposition requiring FLOPS proportional to N^3 or an iterative method such as biconjugate gradient or general minimization of residuals that, in general, require FLOPS proportional to N^2 to achieve convergence. In 3-D problems of any appreciable size this approach is computationally prohibitive and leads to enormous memory demands.

If a domain decomposition solution process is adopted instead, the domain is decomposed by artificially subsectioning the single domain under interest into K subdomains. Each of these is independently discretized and solved by standard BEM while enforcing continuity of temperature and heat flux at the interfaces. It is worth mentioning that discretization of neighboring subdomains does not have to be coincident; that is, at the connecting interface, boundary elements and nodes from the two adjoining subdomains are not required to be structured following a sequence or particular position. The only requirement at the connecting interface is that it forms a closed boundary with the same path on both sides. The information between

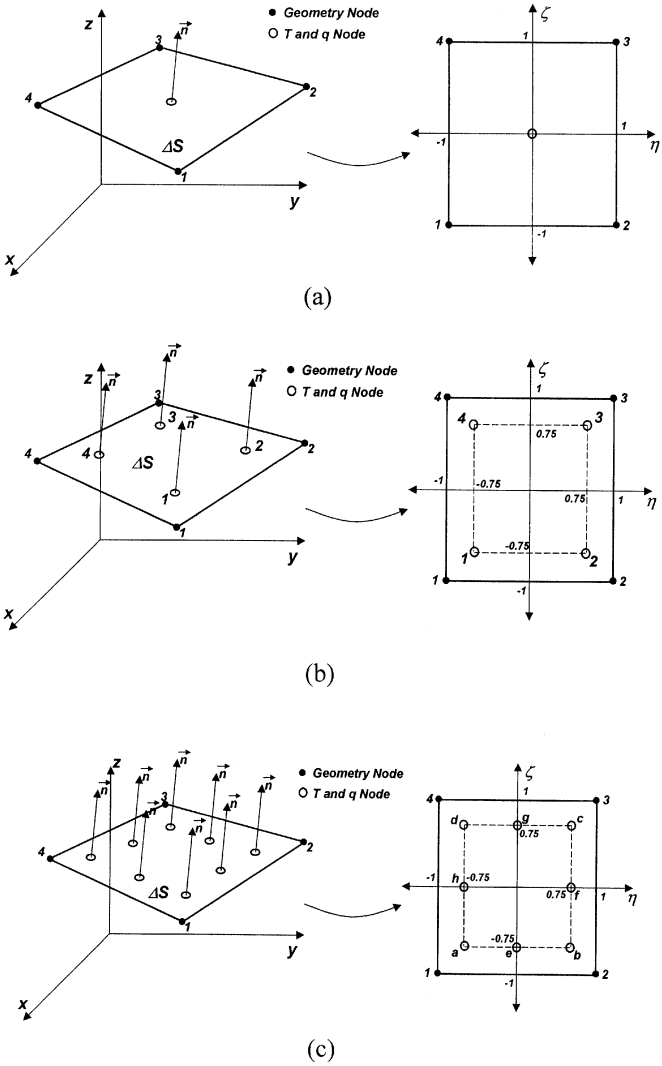


Figure 1. Constant, bilinear, and biquadratic discontinuous elements used for coarse and refined BEM solutions: (*a*) discontinuous subparametric constant element used for coarse solution; (*b*) discontinuous isoparametric bilinear element used for first level of refined solution; (*c*) discontinuous superparametric biquadratic element used for second level of refined solution.

neighboring subdomains separated by an interface can be effectively passed through an interpolation—for instance, by compactly supported radial-basis functions.

The process is illustrated in two dimensions in Figure 2, with a decomposition of four ($K = 4$) subdomains. The conduction problem is solved independently over each subdomain, where initially, a guessed boundary condition is imposed over the interfaces in order to ensure the well-posedness of each subproblem. We will address how to arrive at an efficient initial guess later. The problem in subdomain Ω_1 is transformed into

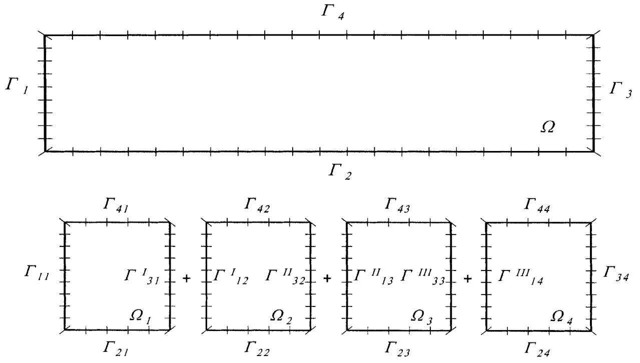


Figure 2. BEM single-region discretization and four-domain BEM decomposition.

$$\nabla^2 T_{\Omega_1}(x,y) = 0 \quad \Rightarrow \quad [H_{\Omega_1}]\{T_{\Omega_1}\} = [G_{\Omega_1}]\{q_{\Omega_1}\} \tag{12}$$

The composition of this algebraic system requires (n^2) FLOPS, where n is the number of boundary nodes in the subdomain, as well as (n^2) for direct memory allocation. This new proportionality number n is roughly equivalent to $n \approx 2N/(K+1)$, as long as the discretization along the interfaces has the same level of resolution as the discretization along the boundaries. Direct memory allocation requirement for later algebraic manipulation is now reduced to a proportion of n^2, as the influence coefficient matrices can easily be stored in ROM memory for later use after the boundary-value problems on remaining subdomains have been effectively solved. For the example shown here, where the number of subdomains is $K = 4$, the new proportionality value n is approximately equal to $n \approx 2N/5$. This simple multiregion example reduces the memory requirements to about $n^2/N^2 = (4/25) = 16\%$ of the standard BEM approach.

The algebraic system for subdomain Ω_1 is rearranged, with the aid of given and guessed boundary conditions, as

$$[H_{\Omega_1}]\{T_{\Omega_1}\} = [G_{\Omega_1}]\{q_{\Omega_1}\} \quad \Rightarrow \quad [A_{\Omega_1}]\{x_{\Omega_1}\} = \{b_{\Omega_1}\} \tag{13}$$

The solution of the new algebraic system of subdomain Ω_1 requires now a number of FLOPS proportional to $n^3/N^3 = (8/125) = 6.4\%$ of the standard BEM approach if a direct algebraic solution method is employed, or a number of floating-point operations proportional to $n^2/N^2 = (4/25) = 16\%$ of the standard BEM approach if an indirect algebraic solution method is employed. For both FLOPS count and direct memory requirement, the reduction is dramatic. However, as the first set of solutions for the subdomains was obtained using guessed boundary conditions along the interfaces, the global solution needs to follow an iteration process and satisfy a convergence criterion.

Globally, the FLOPS count for the formation of the algebraic setup for all K subdomains must be multiplied by K, so the total operation count for the coefficient matrices computation is given by:

$$K\frac{n^2}{N^2} \approx \frac{4K}{(K+1)^2}$$

For this particular case with $K = 4$, $Kn^2/N^2 = 16/25 = 64\%$ of the standard BEM approach. Moreover, the more significant reduction is revealed in the RAM memory requirements, as only the memory needs for one of the subdomains must be allocated at a time, and when a parallel strategy is adopted the matrices for each subdomain are stored by its assigned processor. Therefore, for this case of $K = 4$, the true memory reduction is $n^2/N^2 = 4/25 = 16\%$ of the standard BEM.

With respect to the algebraic solution of the system of Eq. (9), if a direct approach such as LU factorization is employed for all subdomains, the LU factors of the coefficient matrices for all subdomains can be computed only once at the first iteration step and stored on disk for later use during the iteration process, for which only a forward and a backward substitution will be required. This feature allows a significant reduction in the operational count through the iteration process until convergence is achieved, as only a number of floating-point operations proportional to n, as opposed to n^3, is required at each iteration step. To this computation time is added the access to ROM memory at each iteration step, which is usually larger than access to RAM. Alternatively, if the overall convergence of the problem requires few iterations, iterative solvers such as GMRES offer an efficient alternative.

ITERATIVE SOLUTION ALGORITHM

The initial guess is crucial to the success of any iteration. In order to provide an adequate initial guess, we first solve the problem using a coarse-grid, constant-element model, obtained by collapsing the nodes of the discontinuous bilinear element to the centroid, and supply that model with a physically based initial guess for interface temperatures. This converged solution then serves as the initial guess to a finer grid solution obtained using isoparametric bilinear elements, and this solution, in turn, may be used to provide the starting point to a superparametric biquadratic model (see Figure 1, where these three elements are illustrated).

An efficient initial guess can be made using a 1-D heat conduction argument for every node on the external surfaces to every node at the interface of each subdomain, and an area-over-distance argument for the contribution of an external temperature node to an interface node; see Figure 3.

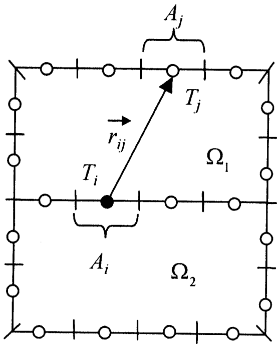


Figure 3. Initial guess at the interface node i illustrated in two dimensions for a two-region subdomain decomposition. A coarse constant-element surface mesh is used to provide the initial guess.

Relating any interface node i to any exterior node j, one can estimate

$$T_i = \frac{\sum_{j=1}^{Ne} A_j T_j / r_{ij}}{\sum_{j=1}^{Ne} A_j / r_{ij}} \tag{14}$$

where $r_{ij} = | \mathbf{r}_{ij} |$ is the magnitude of the position vector from interfacial node i to surface node j. There are Ne exterior nodes, which are imposed with boundary conditions such that N_T exterior nodes are imposed with temperatures, N_q exterior nodes subjected to heat flux conditions, and N_h exterior boundary nodes subjected to convective boundary conditions. Using a 1-D conduction argument for flux and convective nodes, an electric circuit analogy is shown in Figure 4.

Using these arguments, the following initial guess for any interfacial node can be readily obtained in terms of a simple algebraic expression:

$$T_i = \frac{\sum_{j=1}^{N_T} B_{ij} T_j - \sum_{j=1}^{N_q} B_{ij} R_{ij} q_j + \sum_{j=1}^{N_h} B_{ij} H_{ij} T_{\infty j} / H_{ij} + 1}{S_i - \sum_{j=1}^{N_T} B_{ij} + \sum_{j=1}^{N_h} B_{ij} H_{ij} / H_{ij} + 1} \tag{15}$$

$$B_{ij} = \frac{A_j}{r_{ij}}, \quad R_{ij} = \frac{r_{ij}}{k}, \quad H_{ij} = \frac{h_j}{k} r_{ij}, \quad \text{and} \quad S_i = \sum_{j=1}^{N} \frac{A_j}{r_{ij}} \tag{16}$$

with $N = N_T + N_q + N_h$, the thermal conductivity of the medium is k, the film coefficient at the jth convective surface is h_j. The area of element j denoted by A_j is readily computed as:

$$A_j = \oint_{\Gamma_j} d\Gamma(x, y, z) = \int_{-1}^{+1} \int_{-1}^{+1} |J_j(\eta, \zeta)| d\eta d\zeta \tag{17}$$

For a nonlinear problem, the conductivity of the medium is taken at a mean reference temperature. Once the initial temperatures are imposed as boundary conditions at the interfaces, a resulting set of normal heat fluxes along the interfaces will computed. These are then nonsymmetrically averaged in an effort to match the heat flux from neighboring subdomains.

Considering a two-domain substructure, the averaging at the interface is explicitly given as,

$$q^I_{\Omega_1} = q^I_{\Omega_1} - \frac{q^I_{\Omega_1} + q^I_{\Omega_2}}{2} \quad \text{and} \quad q^I_{\Omega_2} = q^I_{\Omega_2} - \frac{q^I_{\Omega_2} + q^I_{\Omega_1}}{2} \tag{18}$$

to ensure the flux continuity condition $q^I_{\Omega_1} = -q^I_{\Omega_2}$ after averaging. Compactly supported radial basis interpolation can be employed in the flux averaging process in

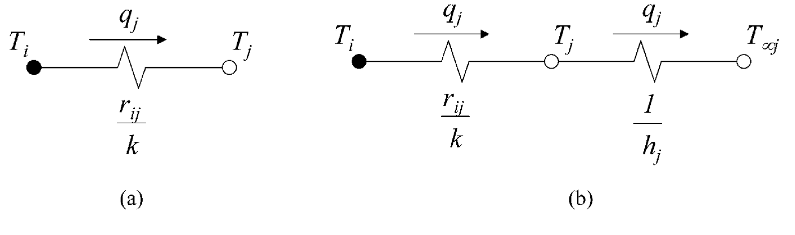


Figure 4. Electric circuit analogy to 1-D heat conduction from node i to node j: (a) heat flux node j; (b) convective node j.

order to account for unstructured grids along the interface from neighboring subdomains.

Using these fluxes the BEM equations are again solved leading to mismatched temperatures along the interfaces for neighboring subdomains. These temperatures are interpolated, if necessary, from one side of the interface to the other side using compactly supported radial basis functions to account for the possibility of interface mismatch between the adjoining substructure grids. Once this is accomplished, the temperature is averaged out at each interface. Illustrating this for a two-domain substructure, again we have for regions 1 and 2 interfaces,

$$\begin{aligned} T^I_{\Omega_1} &= \frac{T^I_{\Omega_1} + T^I_{\Omega_2}}{2} + \frac{R'' q^I_{\Omega_1}}{2} \\ T^I_{\Omega_2} &= \frac{T^I_{\Omega_1} + T^I_{\Omega_2}}{2} + \frac{R'' q^I_{\Omega_2}}{2} \end{aligned} \tag{19}$$

in general to account for a case where a physical interface exists and a thermal contact resistance is present between the connecting subdomains, where R'' is the thermal contact resistance imposing a jump on the interface temperature values. These now matched temperatures along the interfaces are used as the next set of boundary conditions.

It is important to note that when dealing with a nonlinear problem, the interfacial temperature update is performed in terms of temperatures T and not in terms of the Kirchhoff transform variable U. That is, given the current values of the transform variable from either side of the subdomain interface at the current iteration, these are both inverted to provide the actual temperatures and it is these temperatures that are averaged. This is an important point, as the Kirchhoff transform amplifies the jump in temperature at the interface, leading to the divergence of the iterative process reported previously in the literature [17–19]. Also, if a convective boundary condition is imposed at the exposed surface of a subdomain, a sublevel iteration is carried out for that subdomain. However, as the solution for such subdomain is part of the overall iterative process, the sublevel iterations are not carried out to convergence; rather, a few sublevel iterations are carried out. For such cases, we set the number of sublevel iterations to a default number of 5, with an option for the user to increase that number as needed.

The iteration is continued until a convergence criterion is satisfied. A measure of convergence may be defined as the L_2 norm of mismatched temperatures along all interfaces as

$$L_2 = \sqrt{\frac{1}{K \cdot N^I} \sum_{k=1}^{K} \sum_{i=1}^{N^I} \left(T^I - T^I_u\right)^2} \tag{20}$$

This norm measures the standard deviation of BEM computed interface temperatures T^I and averaged-out updated interface temperatures T^I_u. The iteration routine can be stopped once this standard deviation reaches a small fraction ϵ of $\Delta T_{\max,}$ where $\Delta T_{\max}$ is the maximum temperature span of the global field. It is noted that we refer to an iteration as the process by which an iterative sweep is carried out to update both the interfacial fluxes and temperatures such that the above norm may be computed. In all cases in this article, we set $\epsilon = 5 \times 10^{-4}$.

PARALLEL IMPLEMENTATION ON A PC CLUSTER

The domain decomposition BEM formulation detailed above is ideally suited to parallel computing. We have built a small Windows-XP-based P4 cluster consisting of 6 Intel-based P4 PCs (1.7–2 GHz), each equipped with RAMBUS memory ranging from 256 to 512 MB, and two additional PIII PCs, both of which are dual-processor PCs. Thus, in total, this modest cluster is comprised of 8 PCs and 10 processors, and these are connected through a local workgroup in a 100 base-T Ethernet network with full duplex Siemens switches. We also connect to an additional 12 PCs with single 2 GHz Intel P4 processors, each equipped with 1 GB SDRAM accessible through a separate workgroup in a 100 base-T Ethernet network. A parallel version of the code is implemented under MPICH libraries which conform to MPI and MPI2 standards [20–22] and using the COMPAQ Visual Fortran compiler. The parallel code collapses to serial computation if a single processor is assigned to the cluster.

Upon launching the code under MPI, the processors are identified and given a rank. A small BEM problem is solved on each processor to identify its relative performance. A load-balancing routine is then performed to assign domains optimally to each processor by minimizing an objective function that contains information with regard to subdomain sizes and relative computational capability. Specifically, the following objective function is minimized:

$$S = \sum_{N=1}^{N\mathrm{PRO}} [\mathrm{LOAD}(N) - \mathrm{FRA}(N)]^2 \tag{21}$$

where there are $N = 1, 2 \ldots, N\mathrm{PRO}$ processors available to the cluster. The fraction of the overall inverse time it took the Nth computer to solve the test problem is defined as

$$\mathrm{FRA}(N) = \frac{[1/t(N)]}{\sum_{N=1}^{N\mathrm{PRO}} [1/t(N)]} \tag{22}$$

where $t(N)$ is the running time it took the Nth processor in the solution of the test problem and $\sum_{N=1}^{N\mathrm{PRO}} \mathrm{FRA}(N) = 1$. The faster the Nth processor, the larger is $\mathrm{FRA}(N)$ assigned to the Nth processor.

The $\mathrm{LOAD}(N)$ vector is a function measuring the load to the Nth processor in solving the actual problem and is defined as

$$\mathrm{LOAD}(N) = \sum_{k=1}^{NR} \mathrm{ILOAD}(N, k) \times \left[\frac{NE(k)}{NE_{\mathrm{total}}}\right]^a \tag{23}$$

Here, there are $k = 1, 2 \ldots, NR$ regions or subdomains in the actual problem to be solved, region k $NE(k)$ boundary elements used to discretize that region, and there are NE_{total} number of elements in the model. The power $a = 3$ when a direct solver is used for each subdomain problem and $a = 2$ when GMRES isused to solve each subdomain problem. The terms ILOAD (N, k) come from a matrix [ILOAD] assigning loads of certain regions to certain processors. [ILOAD] is a Boolean matrix depending whether a region is assigned to a processor or not. For example, the [ILOAD] matrix shown in Eq. (24) shows a current configuration for an 8-region ($NR = 8$) and 5-processor ($N\text{PRO} = 5$) problem. In the displayed load configuration, processor 1 is assigned regions 2 and 8, processor 2 is assigned regions 3, 5, and 6, while the remaining processor are assigned each a region.

$$[\text{ILOAD}] = \begin{array}{c|cccccccc} & 1 & 2 & 3 & \cdots & & & & \text{NR} \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ \vdots & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ N\text{PRO} & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array} \tag{24}$$

This optimization is performed using a discrete genetic algorithm. A key step in the domain decomposition is to keep each subdomain discretization to a number of elements that allows the problem to be stored in available RAM memory, avoiding disk paging. An illustration of a sample cluster identification and benchmarking is shown in Figure 5 while its final load balancing is illustrated in Figure 6.

NUMERICAL VALIDATION AND EXAMPLES

All computations for the first three examples were performed on a cluster of 6 P4s and for the last two examples on a cluster of 12 P4s each with 1 GB of SDRAM. All geometric lengths are dimensionless, as are all temperatures. Thermophysical properties are scaled to the reference length and temperature. We first present a

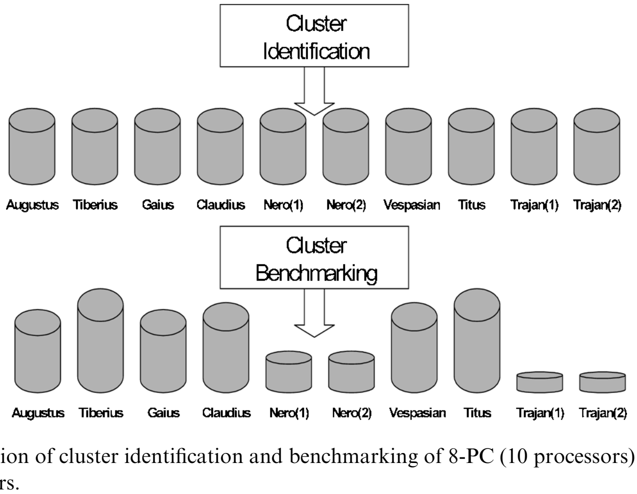


Figure 5. Illustration of cluster identification and benchmarking of 8-PC (10 processors) cluster. Two PCs are dual processors.

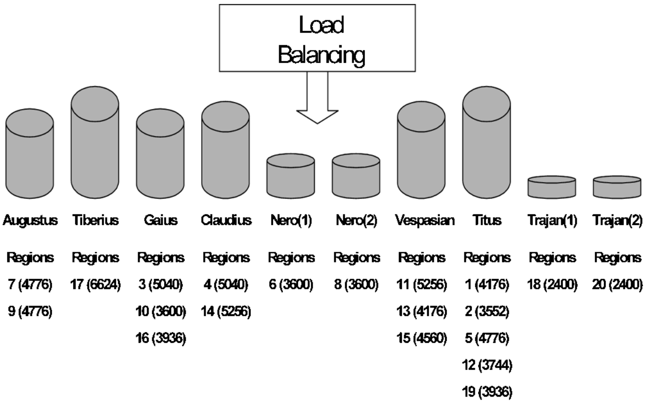


Figure 6. Load balancing for a film-cooled blade 20 region and 85,224 degrees of freedom problem using the 8-PC (10 processor) cluster.

validation example comparing numerical and exact solution for a nonlinear problem in a rectangular slab. Here, we take the nonlinear conductivity as $k(T) = 1 + T$, and use the following expression for temperature:

$$T(x, y, z) = -1 + \sqrt{1 + \frac{3}{19}(2 - 2x^2 + y^2 + z^2)} \tag{25}$$

which satisfies the heat conduction equation, Eq. (1), for the chosen thermal conductivity variation. We use the above to impose boundary conditions on a rectangular slab of length 4 and height and width 1. The slab is discretized into four equal subregions with a total of 600 elements, and this corresponds to 600 degrees of freedom (DOF) for the constant-element model, 2,400 DOF for the bilinear model, and 4,800 DOF for the biquadratic model. Using the exact solution, first-kind boundary conditions were imposed at the two end caps at $y = 0$ and $y = 4$, while second-kind boundary conditions were imposed on the barrel surfaces. The problem converged in 73 s taking 7 iterations for the constant-element model to converge to provide the initial guess for the bilinear case, which converged in one sweep of temperature and flux updates, and subsequently the biquadratic case, which also converged in one sweep of temperature and flux updates. The total time to solution reported here includes load balancing, generation of H and G matrices, and iteration to solution. The additional advantage of our approach is that we are also able to state that our solution is grid-independent, as we perform a grid refinement on our way to the final solution. Results are plotted in Figure 7 for the BEM computed and exact solutions as well as the absolute derivation, which is less than 0.0125 over $\Delta T_{\max} = 1$.

Next we consider a cylinder of radius 1 and length 10 decomposed into 10 equal subdomains corresponding to a discretization of 2,080 elements and 2,080 DOF for the constant-element discretization, 8,320 DOF for the bilinear

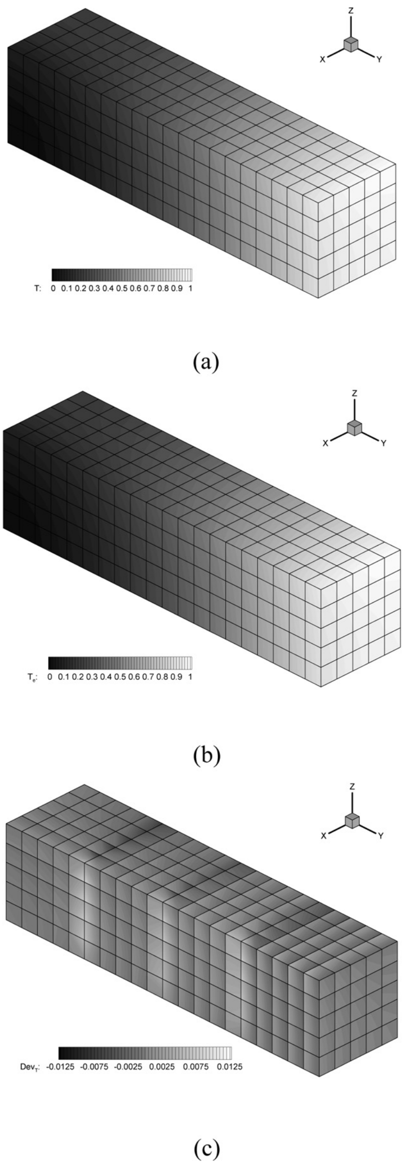


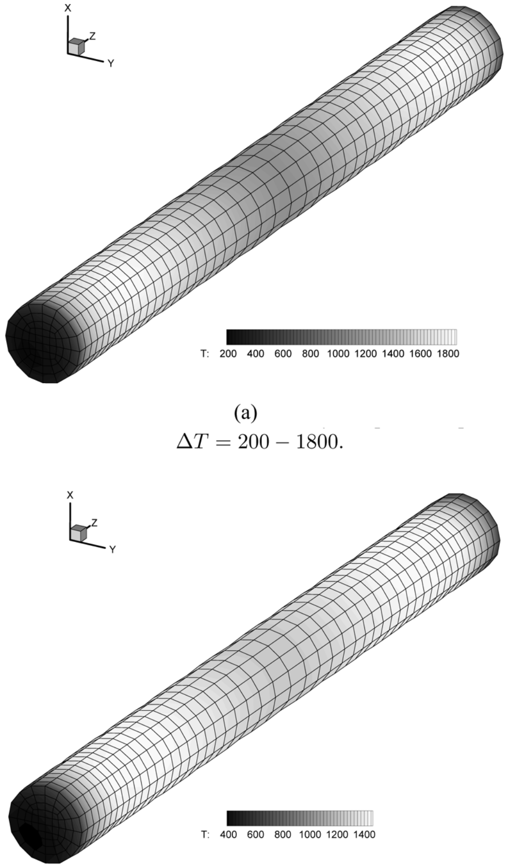
Figure 7. Comparison of BEM domain decomposition solution versus exact solution for a nonlinear conduction problem in a slab: (*a*) BEM-computed solution; (*b*) exact solution; (*c*) absolute deviation, less than 0.0125 over $\Delta T_{\max} = 1$.

discretization, and 16,640 DOF for the biquadratic discretization. Here we consider two cases: (1) a rod with nonlinear conductivity taken as $k(T) = 1.93[1 + 9.07 \times 10^{-4}\ (T - 720)]$, and (2) a composite rod with end caps comprising 10% of the geometry with a low nonlinear conductivity taken as $k(T) = 7.51[1 + 4.49 \times 10^{-4}(T - 1420)]$ while the remainder of the rod has the

Table 1. Number of iterations and timings for rod problem

6-P4 cluster—2,080 elements	Case 1 (homogeneous)	Case 2 (composite)
Constant elements (2,080 DOF)	5 iterations	9 iterations
Bilinear elements (8,320 DOF)	1 iteration	1 iteration
Biquadratic elements (16,640 DOF)	1 iteration	1 iteration
Total time to solution	284 s	292 s

same conductivity as in case (1) or $k(T) = 19.33\ [1 + 4.53 \times 10^{-4}\ (T - 1420)]$ over 80% of the interior. Convective boundary conditions were imposed everywhere on the cylinder walls, with the ends cooled by convection with $T_\infty = 0$ and $h = 10$,


X
Z
Y
T: 200 400 600 800 1000 1200 1400 1600 1800
X
Z
Y
T: 400 600 800 1000 1200 1400

(a)
$\Delta T = 200 - 1800.$

(b)
$\Delta T = 400 - 1400.$

Figure 8. Results from BEM solution in a rod: (*a*) homogeneous rod with nonlinear conductivity, temperature span over the rod is $\Delta T = 200$–1,800; (*b*) composite rod with nonlinear conductivity, temperature span over the rod is $\Delta T = 400$–1,400.

while the perimeter is heated by convection with $h = 1$ and T_∞ varying from 1,000 to 4,000. The timings and convergence of the solutions are provided in Table 1 while the plots of the isotherms for the homogeneous and composite cases are shown in Figure 8.

The next problem considers a more intricate geometry and subsequently a larger-scale problem. Here, a plenum-cooled turbine blade is decomposed into the six subdomains illustrated in Figure 9. There are a total of 5,014 elements, or 5,014 DOF for the constant-element discretization, 20,056 DOF for the bilinear element discretization, and 40,112 DOF for the biquadratic element discretization.

Again, we solve two cases: (1) assumes a constant conductivity $k = 1$ and (2) takes the nonlinear conductivity to vary as $k(T) = 1.93[1 + 9.07 \times 10^{-4}(T - 720)]$. A mixed set of boundary conditions is imposed on the surface of the blade. The plena are imposed with convective conditions, with T_∞ varying linearly from 300 to 500 along the planum depth and with $h = 5$. The end surfaces in the spanwise direction are insulated, while the remaining surfaces are imposed with convective conditions with $T_\infty = 1{,}000$ and $h = 10$. The timings and convergence of the solutions are provided in Table 2 while the plots of the isotherms for the linear and nonlinear cases are shown in Figure 10. Only constant and bilinear analyses were performed in this case.

Next, we consider a conduction problem in an intricate geometry of a film-cooled blade. The domain decomposition for this blade is shown in Figure 11. Here the discretization is comprised of 21,306 elements distributed over 20 subdomains. This corresponds to 21,306 DOF for the constant-element discretization and 85,224 DOF for the bilinear discretization. We consider two cases: (1) linear with a constant conductivity of $k = 1.34$, and (2) nonlinear with $k(T) = 1.09[1 + 4.29 \times 10^{-4}(T - 1{,}620)]$. The end wall surfaces in the spanwise direction are taken as

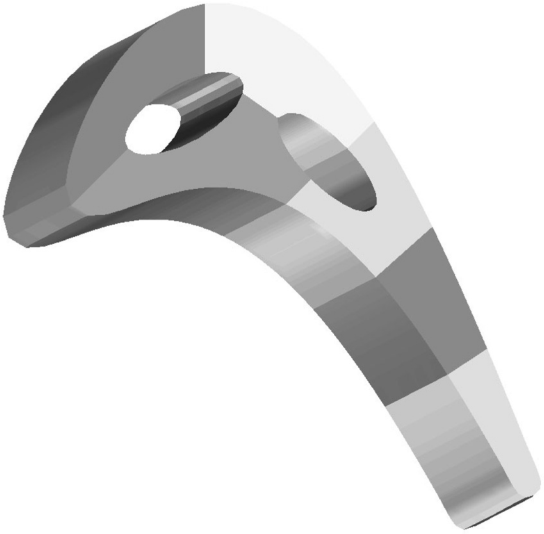
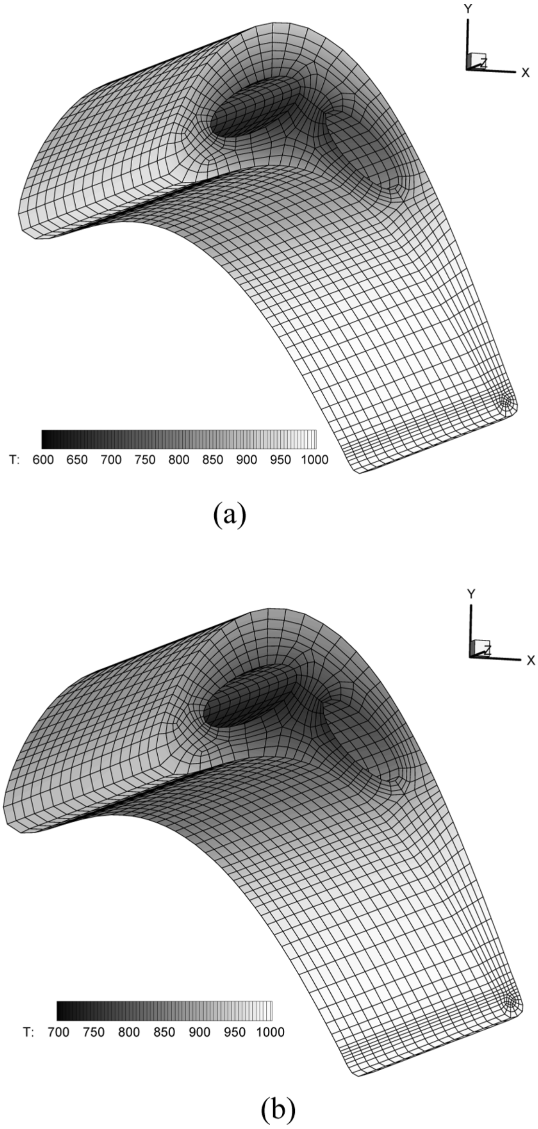


Figure 9. Domain decomposition of the plenum-cooled blade.

Table 2. Number of iterations and timings for blade problem

6 × P4 cluster—5,014 elements	Case 1 (linear)	Case 2 (nonlinear)
Constant elements (5,014 DOF)	3 iterations	9 iterations
Bilinear elements (20,056 DOF)	1 iteration	1 iteration
Total time to solution	765 s	781 s


Y
Z
X
T: 600 650 700 750 800 850 900 950 1000
(a)
Y
Z
X
T: 700 750 800 850 900 950 1000

(b)

Figure 10. Results from BEM solution in a plenum-cooled blade: (*a*) plot of the temperature over the plenum-cooled blade, temperature span over the blade is $\Delta T = 600$–1,000; (*b*) nonlinear conductivity case, $\Delta T = 700$–1,000.

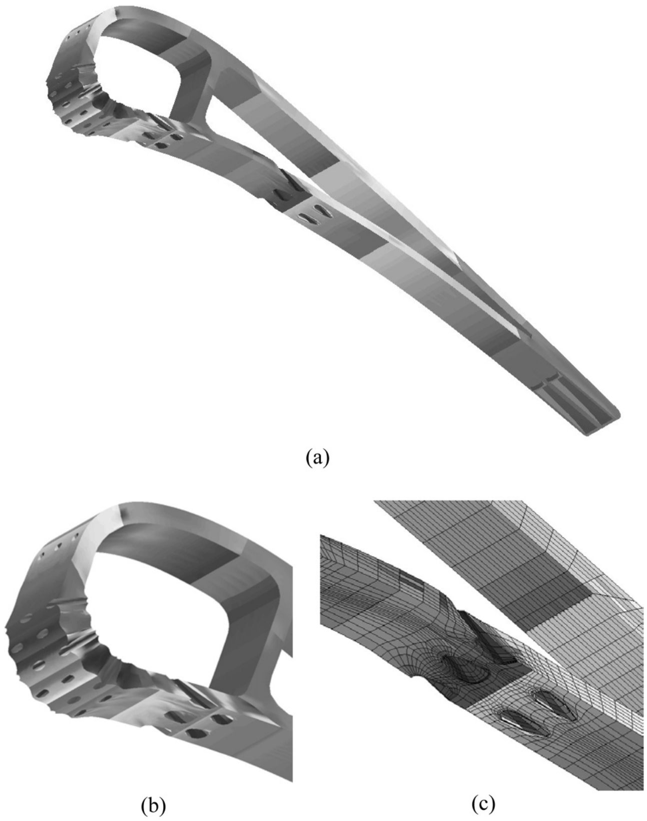


Figure 11. Domain decomposition of film-cooled blade, 21,306-element model and 20 subregions: (*a*) overall domain decomposition of film-cooled blade; (*b*) close-up of leading edge; (*c*) close-up of mesh and domain decomposition.

adiabatic, while temperature boundary conditions imposed on the blade surfaces were obtained from a conjugate analysis carried out on the blade coupling the 3-D BEM code for heat conduction to the Glenn-HT finite-volume code for the flow analysis [1, 2]. The temperatures varied from 1,600 to 3,100 over the surfaces exposed to film cooling, plenum air, and hot gas flowing over the external surfaces. The convergence and timings are reported in Table 3 while the plots of the isotherms

Table 3. Number of iterations and timings for film-cooled blade problem

10 × PC cluster—21,306 elements	Case 1 (linear)	Case 2 (nonlinear)
Constant elements (21,306 DOF)	10 iterations	10 iterations
Bilinear elements (85,224 DOF)	1 iteration	1 iteration
Total time to solution	3,222 s	3,230 s

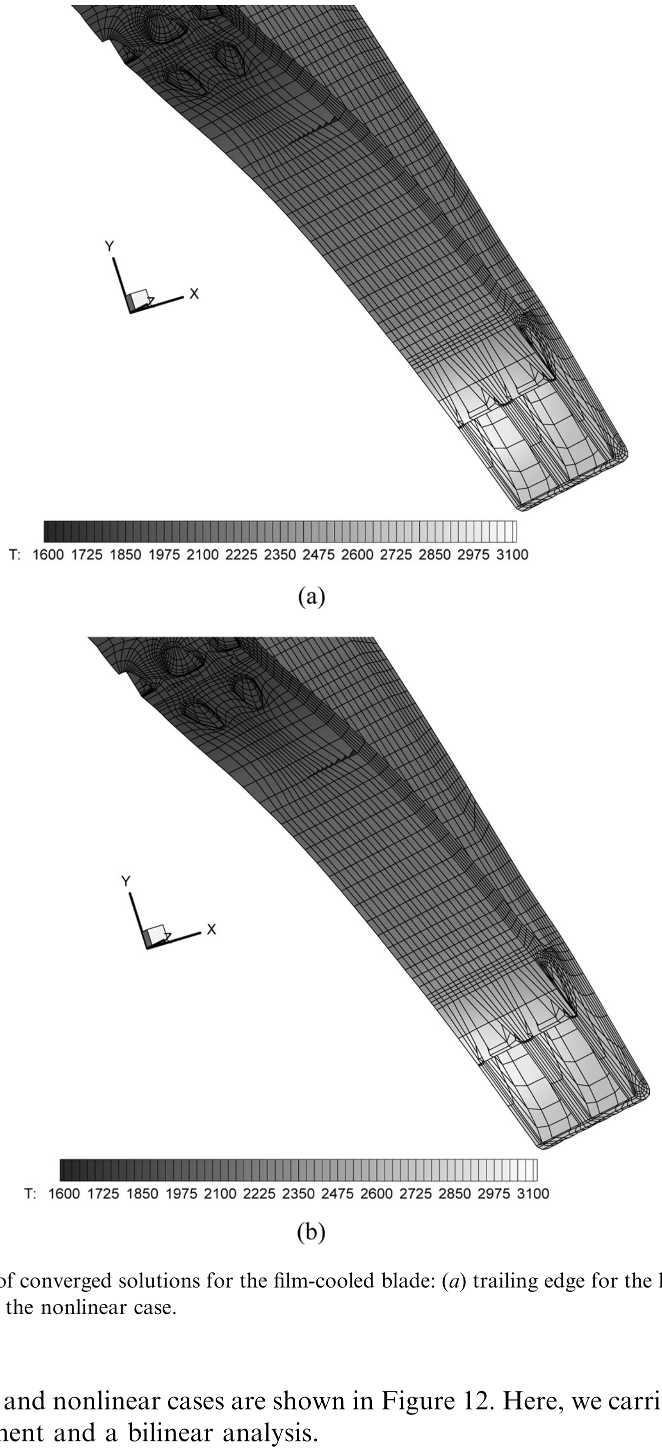

Y
X
T: 1600 1725 1850 1975 2100 2225 2350 2475 2600 2725 2850 2975 3100
(a)
Y
X
T: 1600 1725 1850 1975 2100 2225 2350 2475 2600 2725 2850 2975 3100
(b)

Figure 12. Plot of converged solutions for the film-cooled blade: (*a*) trailing edge for the linear model; (*b*) trailing edge for the nonlinear case.

for the linear and nonlinear cases are shown in Figure 12. Here, we carried out only a constant-element and a bilinear analysis.

Finally, we consider a large-scale conduction problem in a U-tube. The domain decomposition for this case is shown in Figure 13. Here the discretization

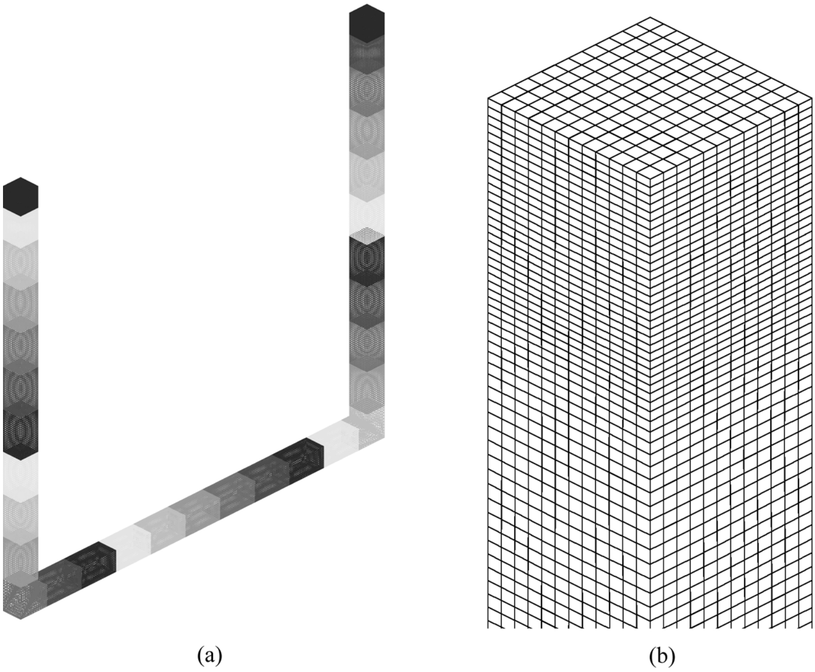


Figure 13. Domain decomposition of U-tube, 44,640-element model and 31 subregions: (*a*) overall domain decomposition of U-tube; (*b*) close-up of mesh.

is comprised of 44,640 elements distributed over 31 subdomains. This corresponds to 44,640 DOF for the constant-element discretization and 178,560 DOF for the bilinear discretization. We consider again two cases modeling stainless steel as the composing material: (1) uniform conductivity taken as $k = 14.9$, and (2) nonlinear conductivity taken as $k(T) = 14.9 \quad [1 + 4.7 \times 10^{-4}(T - 500)]$. The perimetric surface of the U-tube is kept insulated, while heat is added through one of the end caps at a rate of 1,000 and heat is removed through the other end cap by convection at 0° with a heat transfer coefficient of 100. The convergence and timings are reported in Table 4 while the plots of the isotherms for the linear cases are shown in Figure 14. Here, we carried out only a constant-element and a bilinear analysis.

Table 4. Number of iterations and timings for U-tube problem

12 × PC cluster—44,640 elements	Case 1 (linear)	Case 2 (nonlinear)
Constant elements (44,640 DOF)	11 iterations	12 iterations
Bilinear elements (178,560 DOF)	1 iteration	1 iteration
Total time to solution	4,307 s	4,353 s

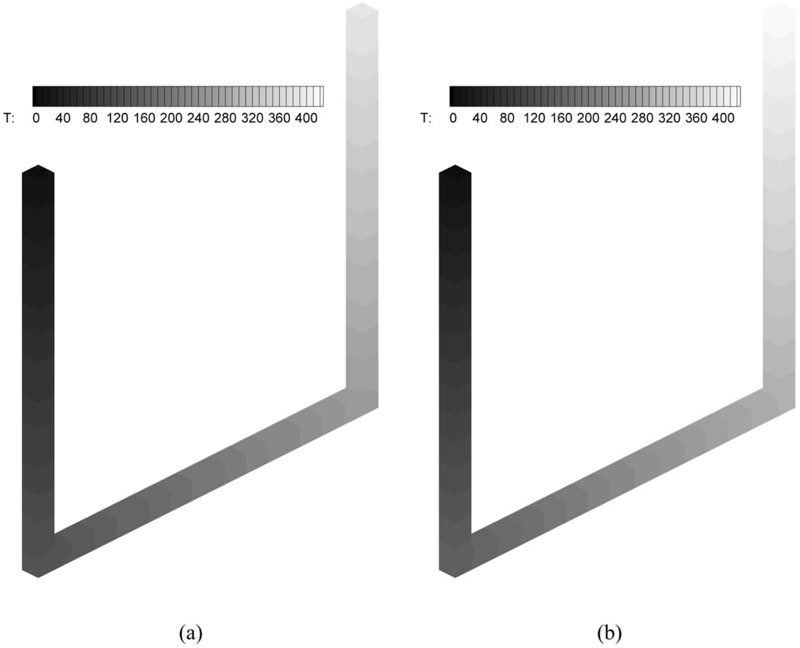


Figure 14. Plot of converged solutions for the U-tube: (*a*) uniform thermal conductivity; (*b*) variable thermal conductivity.

CONCLUSIONS

In this article we presented an efficient iterative domain decomposition method to solve large-scale 3-D BEM problems. The technique uses a coarse grid with a physically based initial guess to provide an effective initial guess for the fine-grid solution. The method lends itself ideally to parallel computation, and large-scale problems can readily solved on modest PC clusters. The approach is general and can be applied to other field problems such as elasticity or thermoelasticity.

REFERENCES

1. A. Kassab, E. Divo, J. Heidmann, E. Steinthorsson, and F. Rodriguez, BEM/FVM Conjugate Heat Transfer Analysis of a Three-Dimensional Film Cooled Turbine Blade, *Int. J. Numer. Meth. Heat Fluid Flow*, vol. 13, pp. 581–610, 2003.
2. J. D. Heidmann, A. J. Kassab, E. A. Divo, F. Rodriguez, and E. Steinthorsson, Conjugate Heat Transfer Effects on a Realistic Film-Cooled Turbine Vane, ASME Paper GT2003-38553, 2003.
3. F. Rizzo and D. J. Shippy, A Formulation and Solution Procedure for the General Non-homogeneous Elastic Inclusion Problem, *Int. J. Solids Struct.*, vol. 4, pp. 1161–1179, 1968.
4. R. A. Bialecki, M. Merkel, H. Mews, and G. Kuhn, In- and Out-of-Core BEM Equation Solver with Parallel and Nonlinear Options, *Int. J. Numer. Meth. Eng.*, vol. 39, pp. 4215–4242, 1996.
5. J. H. Kane, B. L. Kashava-Kumar, and S. Saigal, An Arbitrary Condensing, Non Condensing Strategy for Large Scale, Multi-Zone Boundary Element Analysis, *Comput. Meth. Appl. Mech. Eng.*, vol. 79, pp. 219–244, 1990.

6. B. Baltz and M. S. Ingber, A Parallel Implementation of the Boundary Element Method for Heat Conduction Analysis in Heterogenous Media, *Eng. Anal.*, vol. 19, pp. 3–11, 1997.
7. N. Kamiya, H. Iwase, and E. Kita, Parallel Implementation of Boundary Element Method with Domain Decomposition, *Eng. Anal.*, vol. 18, pp. 209–216, 1996.
8. A. J. Davies and J. Mushtaq, The Domain Decomposition Boundary Element Method, on a Network of Transputers, Ertekin, *BETECHXI: Proc. of 11th Conf. Boundary Element Technology*, Honolulu, Hawaii, pp. 397–406, Computational Mechanics Publications, Southampton, UK, 1996.
9. N. Mai-Duy, P. Nguyen-Hong, and T. Tran-Cong, A Fast Convergent Iterative Boundary Element Method on PVM Cluster, *Eng. Anal.*, vol. 22, pp. 307–316, 1998.
10. L. Greengard and J. Strain, A Fast Algorithm for the Evaluation of Heat Potentials, *Commun. Pure Appl. Math.*, vol. 43, pp. 949–963, 1990.
11. W. Hackbush and Z. P. Nowak, On the Fast Multiplication in the Boundary Element Method by Panel Clustering, *Numer. Math.*, vol. 54, pp. 463–491, 1989.
12. H. Bucher and L. C. Wrobel, A Novel Approach to Applying Wavelet Transforms in Boundary Element Method, *BETEQII: Advances in Boundary Element Techniques, II*, Piscatawy, NJ, pp. 3–13, Hogaar Press, Geneva, Switzerland, 2000.
13. F. Rodriguez, E. Divo, and A. J. Kassab, A Strategy for BEM Modeling of Large-Scale Three Dimensional Heat Transfer Problems, in A. J. Kassab, D. W. Nicholson, and I. Ionescu, (eds.), *Recent Advances in Theoretical and Applied Mechanics, Vol. XXI: Proc. of SECTAM XXI,* Orlando, FL, pp. 645–654, Rivercross Press, Orlando, FL, 2002.
14. C. A. Brebbia, J. C. F. Telles, and L. C. Wrobel, *Boundary Element Techniques*, Springer-Verlag, Berlin, 1984.
15. L. C. Wrobel, *The Boundary Element Method—Applications in Thermo-fluids and Acoustics*, Vol. 1, Wiley, New York, 2002.
16. N. M. Ozisik, *Heat Conduction*, Wiley, New York, 1980.
17. J. P. S. Azevedo and L. C. Wrobel, Non-linear Heat Conduction in Composite Bodies: A Boundary Element Formulation, *Int. J. Numer. Meth. Eng.*, vol. 26, pp. 19–38, 1988.
18. R. Bialecki and R. Nahlik, Solving Nonlinear Steady-State Potential Problems in Nonhomogeneous Bodies Using the Boundary Element Method, *Numer. Heat Transfer B*, vol. 16, pp. 79–96, 1989.
19. R. Bialecki and G. Kuhn, Boundary Element Solution of Heat Conduction Problems in Multi-zone Bodies of Non-linear Materials, *Int. J. Numer. Meth. Eng.*, vol. 36, pp. 799–809, 1993.
20. W. Gropp, E. Lusk, and R. Thakur, *Using MPI: Portable Parallel Programming with the Message-Passing Interface*, MIT Press, Cambridge, MA, 1999.
21. W. Gropp, E. Lusk, and R. Thakur, *Using MPI-2: Advanced Features of the Message-Passing Interface*, MIT Press, Cambridge, MA, 1999.
22. T. E. Sterling, *Beowulf Cluster Computing with Windows*, MIT Press, Cambridge, MA, 2001.