CHAPTER 5

Parallel domain decomposition boundary element method approach for large-scale transient and steady nonlinear heat conduction

E. Divo\textsuperscript{1}, A.J. Kassab\textsuperscript{2} & K. Erhart\textsuperscript{2}

\textsuperscript{1}Engineering Technology Department, University of Central Florida, Orlando, FL, USA.
\textsuperscript{2}Mechanical, Materials, and Aerospace Engineering Department, University of Central Florida, Orlando, FL, USA.

Abstract

It is well known that the boundary element method (BEM) requires only a surface mesh to solve linear, nonlinear, and transient heat conduction problems, but the resulting matrix is fully populated. This poses serious challenges for large-scale 3D problems due to storage requirements and solution times for large sets of non-symmetric equations. Therefore, it is often necessary to employ a domain decomposition approach. This is done herein using an 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 the fine surface grid model. The process converges very efficiently and offers substantial savings in memory. The iterative domain decomposition technique is ideally suited for parallel computation and its MPI implementation on clusters is discussed. Results from 3D BEM heat conduction models including models of upwards of 85,000 nodes arising from an intricate film-cooled vane are presented. We demonstrate that the BEM can be readily applied to solve large-scale linear, nonlinear, and transient heat conduction problems.

1 Introduction

Numerical solutions of engineering problems often require large complex systems of equations to be set up and solved. For any system of equations the amount of
computer memory required for storage is proportional to the square of the number of unknowns, which for large problems can exceed machine limitations. For this reason almost all computational software use some type of problem decomposition. For methods that result in sparse matrices the storage alone can be decomposed to save memory, but for techniques such as boundary element methods (BEM) fully populated matrices generally result and another approach is needed.

The 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). However, in implementing the BEM for intricate 3D 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 from the storage requirements as well as the need to solve a large set of non-symmetric equations.

The BEM community has generally approached this problem by: (1) artificial subsectioning of the 3D model into a multi-region model, an idea originated for piecewise non-homogeneous media [1–3], in conjunction with block-solvers reminiscent of FEM frontal solvers [4, 5] or iterative methods [6–9], and (2) fast-multipole methods adapted to BEM coupled to a GMRES non-symmetric 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 chapter, a domain decomposition or artificial multi-region subsectioning technique is developed along with a region-by-region iteration algorithm tailored for parallel computation [13]. Once the domain decomposition details are presented, two applications will be discussed and detailed. The first is a BEM algorithm to solve large-scale, 3D, steady, nonlinear heat conduction problems, which will allow for multiple regions of different nonlinear conductivities. A non-symmetric update of the interfacial fluxes to ensure equality of fluxes at the subdomain interfaces is formulated. The second application is a BEM algorithm to solve large-scale, transient heat conduction problems. The transient heat conduction equation is transformed into a modified Helmholtz equation using a Laplace transform. The time domain solution is retrieved with a Stehfest numerical inversion routine.

The domain decomposition technique described herein employs an iteration scheme, which is used to ensure the continuity of both temperature and heat flux at the region interfaces. In order to provide a sufficiently accurate initial guess for the iterative process, a physically based initial guess for the temperatures at the domain interfaces is derived, and a coarse grid solution obtained with constant elements is employed. The results of the constant grid model serve as an initial guess for finer discretizations obtained with linear and quadratic boundary element models. The process converges very efficiently, offers substantial savings
Parallel Domain Decomposition BEM Approach

in memory, and does not require complex data-structure preparation required by block-solvers or the multipole approaches. Moreover, the process is shown to converge for steady linear and nonlinear problems as well as for transient problems. The nonlinear problems are treated using the classical Kirchhoff transform. Results from a series of numerical examples are presented. The numerical examples range from validations of the approach against exact solutions in simple geometries to increasingly intricate BEM heat conduction models including a plenum-cooled and a film-cooled turbine blade. The transient cases are then verified in a similar fashion including an intermediate verification for the BEM solution of the modified Helmholtz equation.

2 Explicit domain decomposition

In a standard BEM solution process, outlined below for both cases discussed above, 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

\[
[H][T] = [G][q] \Rightarrow [A][x] = b
\]

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 bi-conjugate gradient or general minimization of residuals that, in general, require FLOPS proportional to \( N^2 \) to achieve convergence. In 3D problems of any appreciable size, the solution becomes computationally prohibitive and leads to enormous memory demands.

A domain decomposition solution process is adopted instead, where 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 the discretization of neighboring subdomains in this method of decomposition does not have to be coincident, i.e. 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 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 2D in Fig. 1, 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 create a well-posed problem for each subdomain. The details of an efficient initial guess technique will be addressed later. The problem in subdomain \( \Omega_1 \)
is transformed into
\[ \nabla^2 T_{\Omega_1}(x, y) = 0 \Rightarrow [H_{\Omega_1}] [T_{\Omega_1}] = [G_{\Omega_1}] [q_{\Omega_1}] \] (2)

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 \( n^2 \), as the influence coefficient matrices can easily be stored in ROM memory for later use after the boundary value problems on the 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 multi-region example reduces the memory requirements to about \( n^2/N^2 = (4/25) = 16\% \) of the standard BEM approach.

The algebraic system for subdomain \( \Omega_1 \) is rearranged, with the aid of given and guessed boundary conditions, as:
\[ [H_{\Omega_1}] [T_{\Omega_1}] = [G_{\Omega_1}] [q_{\Omega_1}] \Rightarrow [A_{\Omega_1}] [x_{\Omega_1}] = [b_{\Omega_1}] \] (3)

The solution of the new algebraic system of subdomain \( \Omega_1 \) now requires a number FLOPS proportional to \( n^2/N^2 = (8/125) = 6.4\% \) 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

---

Figure 1: BEM single region discretization and four domain BEM decompositions.
direct memory requirement, this reduction is dramatic. However, as the first set of solutions for the subdomains were obtained using guessed boundary conditions along the interfaces, the global solution needs to follow an iteration process and satisfy a convergence criteria.

Globally, the FLOPS count for the formation of the algebraic setup for all \( K \) subdomains must be multiplied by \( K \), therefore, the total operation count for the coefficient matrices computation is given by: \( Kn^2/N^2 \approx 4K/(K + 1)^2 \). For this particular case with \( K = 4 \), this corresponds to \( Kn^2/N^2 = 16/25 = 64\% \) 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, as the others can temporarily be stored into ROM, 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 memory requirements are reduced to only \( n^2/N^2 = 4/25 = 16\% \) of the standard single region case.

In order to reduce the computational efforts needed with respect to the algebraic solution of the system of eqn (1), a direct approach LU factorization is employed for all subdomains. The LU factors of the coefficient matrices for all subdomains are constant. As they are independent of the right-hand side vector, they can be computed only once at the first iteration step and stored on disc for later use during the iteration process. Therefore, at each iteration only a forward and a backward substitution will be required for the algebraic solution. This feature allows a significant reduction in the operational count through the iteration process, as only a number of floating point operations proportional to \( n \) as opposed to \( n^3 \) is required at each iteration step. The access to ROM memory at each iteration step must also be added to this computation time, which is usually larger than access to RAM. Typically, however, the overall convergence of the problem requires few iterations, and this ROM access is not a significant addition. Additionally, iterative solvers such as GMRES may offer a more efficient alternative.

### 3 Iterative solution algorithm

The initial guess is crucial to the success of any iteration scheme. In order to provide an adequate initial guess for the 3D case, the problem is first solved 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 iso-parametric bilinear elements, and this solution, in turn, may be used to provide the starting point to a super-parametric biquadratic model, see Fig. 6a–c where these three elements are illustrated.

While the constant element solution can be used as an initial guess for the later runs, an initial guess is still required in the solution of the constant element case. An efficient initial guess can be made using a 1D heat conduction argument for every node on the external surfaces to every node at the interface of each subdomain. An area over distance argument is then used to weight the contribution of an external temperature node to an interface node, see Fig. 2.
Relating any interface node $i$ to any exterior node $j$, one can estimate:

$$T_i = \frac{\sum_{j=1}^{N_c} A_j T_j}{\sum_{j=1}^{N_c} A_j}$$

(4)

where $r_{ij} = |\vec{r}_{ij}|$ is the magnitude of the position vector from interfacial node $i$ to surface node $j$. There are $N_c$ exterior nodes which are imposed with boundary conditions such that $N_T$ exterior nodes are imposed with temperatures, $N_q$ exterior nodes are subjected to heat flux conditions, and $N_h$ exterior boundary nodes are subjected to convective boundary conditions. Using a 1D conduction argument for the flux and convective nodes (see Fig. 3), the following initial guess for any interfacial node can be readily obtained in terms of a simple algebraic expression:

$$T_i = \frac{\sum_{j=1}^{N_T} B_{ij} T_j - \sum_{j=1}^{N_q} B_{ij} q_j + \sum_{j=1}^{N_h} \frac{B_{ij} H_0 T_{\infty}}{H_j + 1}}{S_i - \sum_{j=1}^{N_T} B_{ij} + \sum_{j=1}^{N_h} \frac{B_{ij} H_0}{H_j + 1}}$$

(5)
Parallel Domain Decomposition BEM Approach

and the coefficients are explicitly

\[ B_{ij} = \frac{A_j}{r_{ij}}, \quad R_{ij} = \frac{r_{ij}}{k}, \quad H_{ij} = \frac{h_j}{k}, \quad \text{and} \quad S_i = \sum_{j=1}^{N} \frac{A_j}{r_{ij}} \quad (6) \]

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

\[ A_j = \oint_{\Gamma_j} d\Gamma(x, y, z) = \int_{-1}^{+1} \int_{-1}^{+1} |J_j(\eta, \zeta)| d\eta d\zeta \quad (7) \]

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 be computed. These are then non-symmetrically 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,

\[ q^l_{\Omega_1} = q^l_{\Omega_1} - \frac{q^l_{\Omega_2} + q^l_{\Omega_2}}{2} \quad \text{and} \quad q^l_{\Omega_2} = q^l_{\Omega_2} - \frac{q^l_{\Omega_1} + q^l_{\Omega_1}}{2} \quad (8) \]

to ensure the flux continuity condition \( q^l_{\Omega_1} = -q^l_{\Omega_2} \) after averaging. Compactly supported radial basis interpolation can be employed in the flux averaging process in 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, the interface temperatures for regions 1 and 2 are:

\[ T^l_{\Omega_1} = \frac{T^l_{\Omega_1} + T^l_{\Omega_2}}{2} + \frac{\tilde{R} q^l_{\Omega_1}}{2} \]
\[ T^l_{\Omega_2} = \frac{T^l_{\Omega_1} + T^l_{\Omega_2}}{2} + \frac{\tilde{R} q^l_{\Omega_2}}{2} \quad (9) \]

in general to account for a case where a physical interface exists and a thermal contact resistance is present between the connecting subdomains, where \( \tilde{R} \) is the thermal contact resistance imposing a jump on the interface temperature values. These matched temperatures along the interfaces are used as the next set of boundary conditions.
It is important to note that when dealing with the 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 [14–16]. 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 a 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, the number of sublevel iterations is set to a default number of 5, with an option for the user to increase that number as needed.

The overall 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:

$$L_2 = \sqrt{\frac{1}{K \times N_I} \sum_{k=1}^{K} \sum_{i=1}^{N_I} (T_i^k - T_i^{u,k})^2}$$

This norm measures the standard deviation of BEM computed interface temperatures $T_i$ and averaged-out updated interface temperatures $T_i^{u,k}$. The iteration routine can be stopped once this standard deviation reaches a small fraction $\epsilon$ of $\Delta T_{\text{max}}$, where $\Delta T_{\text{max}}$ is the maximum temperature span of the global field. It is noted that an iteration is referred to 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. Here it is important to note that for the steady problems a value of $\epsilon = 5 \times 10^{-4}$ is sufficient for accurate solutions. However, due to the amplification effects of the Stehfest transform the transient cases require values as small as $\epsilon = 10^{-15}$.

### 4 Parallel implementation on a PC cluster

The domain decomposition BEM formulation detailed above is ideally suited to parallel computing. A small Windows XP-based cluster consisting of 10 Intel-based P3 and P4 CPUs (1.7 GHz–2 GHz) equipped with RAMBUS memory ranging from 768 MB to 1,024 MB was built to implement and test these algorithms. This small cluster is interconnected through a local workgroup in a 100 base-T Ethernet network with full duplex switches. An additional cluster comprised of 12 Intel P4 CPUs (2 GHz) each equipped with 1 GB SDRAM was built separately and made 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 with MPI and MPI2 standards [17–19] and using the COMPAQ Visual FORTRAN compiler. The parallel code collapses to serial computation if a single processor is assigned to the cluster.
Parallel Domain Decomposition BEM Approach

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 static load balancing routine is then performed to optimally assign domains to each processor by minimizing an objective function that contains information with regard to subdomain sizes and relative computational capabilities. Specifically, the following objective function is minimized:

$$S = \sum_{N=1}^{NPRO} [LOAD(N) - FRA(N)]^2$$  \hspace{1cm} (11)

where there are \( N = 1, 2, \ldots, NPRO \) processors available to the cluster. The fraction of the overall inverse time it took the \( N \)th computer to solve the test problem is defined as

$$FRA(N) = \frac{[1/t(N)]}{\sum_{N=1}^{NPRO} [1/t(N)]}$$  \hspace{1cm} (12)

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

The \( LOAD(N) \) vector is a function measuring the load to the \( N \)th processor in solving the actual problem and that is defined as

$$LOAD(N) = \sum_{k=1}^{NR} ILOAD(N,k) \times \left[ \frac{NE(k)}{NE_{total}} \right]^a$$  \hspace{1cm} (13)

Here, there are \( k = 1, 2, \ldots, NR \) regions or subdomains in the actual problem to be solved, region \( k \) has \( NE(k) \) 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 is used 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 on whether a region is assigned or not to a processor. For example, the \([ILOAD]\) matrix shown in eqn (14) shows a current configuration for an 8-region (\( NR = 8 \)) and 5-processor (\( NPRO = 5 \)) problem. In the displayed load configuration, processor 1 is assigned regions 2 and 8, processor 2 is assigned regions 3, 5, 6 while the remaining processors are each assigned a region.

$$[ILOAD] = \begin{bmatrix}
1 & 2 & 3 & \ldots & NR \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
2 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\
\vdots & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
NPRO & 0 & 0 & 0 & 1 & 0 & 0 & 0
\end{bmatrix}$$  \hspace{1cm} (14)

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 (accessing ROM).

A typical cluster identification and benchmarking is illustrated for the 10-node cluster in Fig. 4. The resulting load balancing for a 20-region model performed with the discrete genetic algorithm process described above is shown in Fig. 5.

Figure 4: Illustration of cluster identification and benchmarking of 8-PC (10 processors) cluster. Two PCs are dual processors.

Figure 5: Load balancing for a film-cooled blade 20 region and 85,224 degrees of freedom problem using the 8-PC (10 processors) cluster.
5 Applications in heat transfer

The following sections detail the BEM formulations for the two cases introduced above, steady 3D nonlinear heat conduction, and transient 2D linear conduction. The discussion begins with the derivations of each technique and is followed by verification examples and comparisons.

5.1 3D nonlinear heat conduction

The initial discussion shall focus primarily on nonlinear 3D heat transfer, where the governing equation under consideration is the steady-state nonlinear heat conduction equation, or:

$$\nabla \cdot [k(T) \nabla T] = 0$$  \hspace{1cm} (15)

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 as:

$$\nabla^2 T = 0$$  \hspace{1cm} (16)

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, \( U(T) \) \[20\], such that:

$$U(T) = \frac{1}{k_0} \int_{T_o}^{T} k(T) dT$$  \hspace{1cm} (17)

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 \( k \) vs. \( T \) 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 transformed parameter \( U(T) \), as:

$$\nabla^2 U = 0$$  \hspace{1cm} (18)

The boundary conditions are transformed linearly as long as they are of the first or second kind, such that:

$$T|_{r_s} = T_s \rightarrow U|_{r_s} = U(T_s) = U_s$$

$$-k \frac{\partial T}{\partial n}|_{r_s} = q_s \rightarrow -k_o \frac{\partial U}{\partial n}|_{r_s} = q_s$$  \hspace{1cm} (19)
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 as:

\[
-k \frac{\partial T}{\partial n} \bigg|_{r_s} = h_s [T|_{r_s} - T_\infty] \rightarrow -k_o \frac{\partial U}{\partial n} \bigg|_{r_s} = h_s [T(U|_{r_s}) - T_\infty]
\]  

(20)

In this case, iteration is required. This is accomplished by rewriting the convective boundary condition as:

\[
-k_o \frac{\partial U}{\partial n} \bigg|_{r_s} = h_s [U|_{r_s} - T_\infty] + h_s [T(U|_{r_s}) - U|_{r_s}]
\]  

(21)

and first solving the problem with the linearized boundary condition:

\[
-k_o \frac{\partial U}{\partial n} \bigg|_{r_s} = h_s [U|_{r_s} - T_\infty]
\]  

(22)

to provide an initial guess for iteration.

The heat conduction equation thus reduces to the Laplace equation in any case. The notation will be hereon changed back to \( T \) as the dependent variable with the understanding that when dealing with a nonlinear problem: \( T \rightarrow U \). The Laplace equation is readily solved by first converting it into a boundary integral equation (BIE), [20, 21], as:

\[
C(\xi) T(\xi) + \oint_S q(x) G(x, \xi) \, dS(x) = \oint_S T(x) H(x, \xi) \, dS(x)
\]  

(23)

where \( S(x) \) is the surface bounding the domain of interest, \( \xi \) 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 \( \xi \), and is given by \( G(x, \xi) = 1/4\pi k r(x, \xi) \) in 3D where \( r(x, \xi) \) is the Euclidean distance from the source point \( \xi \). The free term \( C(\xi) = \oint_S H(x, \xi) \, dS(x) \) can be shown analytically to be the internal angle subtended at source point divided by \( 4\pi \) when \( \xi \) is on the boundary and one when \( \xi \) 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 eqn (23), or

\[
[H][T] = [G][q]
\]  

(24)

These equations are readily solved upon imposition of boundary conditions. Subparametric constant, isoparametric bilinear, and superparametric biquadratic, discontinuous boundary elements are used as the basic elements in the 3D BEM codes developed to implement these algorithms. These are illustrated in Fig. 6a–c. Such elements avoid the so-called star-point issue and allow for discontinuous fluxes.
Figure 6: 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 the refined solution, and (c) discontinuous superparametric biquadratic element used for second level of the refined solution.
Moreover, the biquadratic elements used here are superparametric, with bilinear model of the geometry and biquadratic model of the temperature and heat flux. This type of element provides compatibility of geometric models with grids generated by structured finite volume grid generators.

5.2 Transient heat conduction

The BEM has been traditionally used to solve transient heat conduction problems through three different approaches: (1) using the convolution scheme where a time-dependent Green’s function is introduced to build a transient boundary integral equation model, (2) using the dual reciprocity method (DRM) to expand the spatial portion of the governing equation using radial basis functions and a finite difference scheme to march in time, and (3) using the Laplace transformation of the governing equation to eliminate the time derivative leading to a modified Helmholtz equation that can be solved using a steady BEM approach and further inverting the BEM solution back into real space–time using a numerical Laplace inversion scheme [20, 22–31]. The first approach will require the generation and storage of BEM influence coefficient matrices at every time step of the convolution scheme making the technique unfeasible for medium or large problems, particularly in 3D applications, as the computational and storage requirements become unrealistically high. The second approach poses a different issue as the global interpolation functions for the dual reciprocity technique, such as the widely used radial basis functions lack convergence and error estimation approximations and can at times lead to unwanted behavior and significantly increase the conditioning number of the resulting algebraic system. The third approach, originated in a BEM application by Rizzo and Shippy [23], does not require time marching or any type of interpolation, but it requires fine-tuning of the BEM solution of the modified Helmholtz equation and a numerical Laplace inversion of the results. Real variable-based numerical Laplace inversion techniques such as the Stehfest transformation [28–29] provide very accurate results for non-oscillative type of functions, such as the ones expected to result from transient heat conduction applications, as all poles of the transformed solution are real and distributed along the negative real axis. One type of parallelization has been discussed by Davies and Crann [30] where individual solutions, as required for the numerical Laplace inversion, are solved simultaneously in multiple processors. This type of parallelization does reduce the computational time, but does not aid in storage requirements as the entire domain must be handled by each processor, thus leaving room for efficiency improvements.

The following sections will formulate a Laplace-transformed BEM algorithm structured with the iterative parallel domain decomposition scheme detailed above, which acts to significantly reduce the computational and storage requirements of large-scale problems. Accurate results are reported for different cases in regular and irregular geometries imposed with different types of boundary conditions. The iteration algorithm is proven to efficiently converge for all the reported situations under different levels of domain decomposition and offers much promise for the efficient solution of 3D transient heat conduction problems using the BEM.
5.2.1 Governing equation and the Laplace transform

Transient heat conduction is governed by the well-known diffusion equation, which for a 2D rectangular coordinate system is given by:

\[ \nabla \cdot [k \nabla T(x, y, t)] = \rho c \frac{\partial T}{\partial t}(x, y, t) \]  

(25)

The purpose of applying the Laplace transform to the transient heat conduction equation is to eliminate the time-dependency of the equation. Taking the Laplace transform of the diffusion equation yields the following:

\[ \nabla \cdot [k \nabla \tilde{T}(x, y, s)] = \rho cs \tilde{T}(x, y, s) - \rho c T(x, y, 0) \]  

(26)

where, \( \tilde{T}(x, y, s) \) is the Laplace-transformed temperature and new dependent variable. The above expression can be further reduced by requiring the initial condition:

\[ T(x, y, 0) = 0 \]  

(27)

which is true for any case of uniform initial condition with a proper superposition. Equation (26) is also no longer a function of time, but now contains the Laplace transform parameter, \( s \). This parameter can now simply be treated as a constant in all remaining derivations. The dependency of the temperature field on \( s \) can now be eliminated, and the initial condition can be applied resulting in the following expression:

\[ \nabla \cdot [k \nabla \tilde{T}(x, y)] = \rho cs \tilde{T}(x, y) = 0 \]  

(28)

The thermal conductivity can be independent of temperature and the above expression will have the form of the modified Helmholtz equation:

\[ \nabla^2 \psi - m^2 \psi = 0 \]  

(29)

The solution to this equation is well known, since many other physical problems are governed by the modified Helmholtz equation, such as acoustic propagation (where \( \psi \), would be the acoustic potential) [31]. The BEM has been effectively implemented for the solution of such acoustic propagation problems. Therefore, the Laplace transformation of the diffusion equation yields a suitable form for the desired solution method.

Finally, the boundary conditions must be transformed in order to refer the entire problem in the proper Laplace transform space. The boundary conditions are transformed using the following relations:

\[ \tilde{T}(x, y, s) \big|_\Gamma = \left. \frac{T(x, y)}{s} \right|_\Gamma \]  

(30)

\[ \tilde{q}(x, y, s) \big|_\Gamma = \left. \frac{q(x, y)}{s} \right|_\Gamma \]  

(31)

where, \( \Gamma \), represents the boundary (control surface).
5.2.2 BEM for the modified Helmholtz equation

The development of a BEM solution begins by reducing the governing equation to a boundary-only integral equation. The current form of the Laplace-transformed transient heat conduction problem can be expressed in integral form by pre-multiplying the equation by a generalized function $G(x, y, \xi)$ and integrating over the domain of interest of the problem at hand ($\Omega$: control volume) to obtain:

$$\int_{\Omega} [G(x, y, \xi) \nabla \cdot (k \nabla \bar{T}(x, y))] d\Omega - \rho cs \int_{\Omega} G(x, y, \xi) \bar{T}(x, y) d\Omega = 0$$  \hspace{1cm} (32)

Integration by parts is applied twice to the above equation (Green’s second identity) and the equation below is obtained:

$$\oint_{\Gamma_1} Gk \partial \bar{T} \partial n d\Gamma_1 - \oint_{\Gamma_1} \bar{T} k \partial G \partial n d\Gamma_1 + \int_{\Omega} \bar{T} \nabla \cdot (k \nabla G) d\Omega - \rho cs \int_{\Omega} G \bar{T} d\Omega = 0$$  \hspace{1cm} (33)

where the dependencies have been omitted for clarity. The following two expressions can be defined as:

$$\bar{q}(x, y) = -k \frac{\partial \bar{T}}{\partial n}(x, y)$$  \hspace{1cm} (34)

$$H(x, y, \xi) = -k \frac{\partial G}{\partial n}(x, y, \xi)$$  \hspace{1cm} (35)

Here, $n$ is the outward-drawn normal. Introducing the above expressions into the integral equation yields:

$$\int_{\Omega} \bar{T} \nabla \cdot (k \nabla G) d\Omega - \rho csG \bar{T} d\Omega = \oint_{\Gamma} G \bar{q} d\Gamma - \oint_{\Gamma} H \bar{T} d\Gamma$$  \hspace{1cm} (36)

The Dirac delta function can now be used to perturb the adjoint operator over the function $G$ as:

$$\nabla \cdot [k \nabla G(x, y, \xi)] - \rho csG(x, y, \xi) = -\rho c \delta(x, y, \xi)$$  \hspace{1cm} (37)

This equation is then solved in free space to determine the fundamental solution, $G$ as:

$$G(x, y, \xi) = \frac{1}{2\pi \alpha} K_0 \left( \frac{s}{\alpha r} \right)$$  \hspace{1cm} (38)

where $\alpha$ is defined as $\alpha = k / \rho c$, $K_0$ is a modified Bessel function of the second kind of order zero, and $r = \sqrt{(x - x_i)^2 + (y - y_i)^2}$. Now that the function $G$ has been determined the function $H$ can be evaluated from its definition as:

$$H(x, y, \xi) = -\frac{\rho c}{2\pi} \left[ \frac{\partial K_0}{\partial x} \left( \frac{s}{\alpha r} \right) n_x + \frac{\partial K_0}{\partial y} \left( \frac{s}{\alpha r} \right) n_y \right]$$  \hspace{1cm} (39)
Parallel Domain Decomposition BEM Approach

where \( n_x \) and \( n_y \) are the \( x \)- and \( y \)-components of the outward-drawn normal, \( n \). Noting that the derivative of \( K_o(z) \) is \( -K_1(z) \), the above expression is simplified to the following:

\[
H(x, y, \xi) = -\frac{\rho c}{2\pi r} \sqrt{\frac{\alpha}{r}} \cdot [(x - x_i) n_x + (y - y_i) n_y]
\] (40)

Finally, the sifting property of the Dirac delta function is used to eliminate the only domain integral in the integral equation and obtain a boundary-only integral equation as:

\[
\rho c C(\xi) \tilde{T}(\xi) = \oint_{\Gamma} H(x, y, \xi) \tilde{T}(x, y) \, d\Gamma - \oint_{\Gamma} G(x, y, \xi) \tilde{q}(x, y) \, d\Gamma
\] (41)

The first step of the BEM implementation is to divide the boundary into \( N \)-discrete elements as:

\[
\Gamma = \sum_{j=1}^{N} \Delta \Gamma_j
\] (42)

The boundary integral equation is discretized along the boundary as shown below:

\[
\rho c C(\xi) \tilde{T}(\xi) = \sum_{j=1}^{N} \int_{\Delta \Gamma_j} \tilde{H}(x, y, \xi) \tilde{T}(x, y) \, d\Gamma - \sum_{j=1}^{N} \int_{\Delta \Gamma_j} G(x, y, \xi) \tilde{q}(x, y) \, d\Gamma
\] (43)

where \( C(\xi) \) is a geometry-dependent term calculated as the internal angle at point \( \xi \) divided over \( 2\pi \). Therefore, \( C(\xi) = 1 \) if \( \xi \in \Omega \) and \( C(\xi) = \frac{1}{2} \) when \( \xi \in \Gamma \) and \( \Gamma \) is a smooth boundary.

The functions \( C(\xi) \) and \( \tilde{T}(\xi) \) can now be evaluated at a specific collocation point \( \xi_i \), in order to obtain the expression below:

\[
\rho c C(\xi_i) \tilde{T}(\xi_i) = \sum_{j=1}^{N} \hat{H}_{ij} \tilde{T}_j - \sum_{j=1}^{N} G_{ij} \tilde{q}_j
\] (44)

where \( G_{ij} = \int_{\Delta \Gamma_j} G(x, y, \xi_i) \, d\Gamma \) and \( \hat{H}_{ij} = \int_{\Delta \Gamma_j} H(x, y, \xi_i) \, d\Gamma \) are known as the influence coefficients, and these are evaluated numerically. Adaptive Gauss–Kronrod quadratures are used for this purpose. The equation above can now be simplified and the following expression results:

\[
\sum_{j=1}^{N} H_{ij} \tilde{T}_j = \sum_{j=1}^{N} G_{ij} \tilde{q}_j
\] (45)

Here, \( H_{ij} = \hat{H}_{ij} - (\rho c/2) \delta_{ij} \), such that \( \delta_{ij} = 0 \) if \( i \neq j \), and \( \delta_{ij} = 1 \) if \( i = j \). Boundary conditions can be further applied to reduce the system of equations above to the standard algebraic form \([A][x] = [b]\).

Once the system is solved by standard linear algebra methods, the solution must be inverted numerically from the Laplace space to the real (transient space). The numerical inversion routine of choice is described in the following section.
5.2.3 Numerical inversion of the Laplace transformed solution

The final step of the overall numerical solution is the inversion of the Laplace transformed BEM solution. While many techniques exist for such an inversion, the Stehfest transform has the advantages of being quite stable, very accurate, and simple to implement. The Stehfest transform works by computing a sample of solutions at a specified number of times and predicting the solution based on this sample [32]. Due to the non-oscillative behavior of the transient heat conduction equation, the Stehfest transform works exceptionally well. The accuracy of the Stehfest transform as coded for this algorithm was verified by testing several equations with known analytical inversions, and satisfactory results were obtained. A function, \( f(t) \), which has a Laplace transform, \( \tilde{f}(s) \), can be inverted using the relation:

\[
f(t) \sim \lim_{n \to \infty} I_n(t)
\]

where the sample \( I_n(t) \) is defined on the basis of a delta sequence \( \delta_n(t, s) \) as:

\[
I_n(t) = \int_0^\infty \delta_n(t, s) \tilde{f}(s) ds
\]

A delta sequence [33] is defined such that as \( n \to \infty \) then, the delta sequence tends to the Dirac delta function \( \delta(t) \), i.e.:

\[
\lim_{n \to \infty} \delta_n(t, s) \to \delta(t)
\]

The Stehfest inversion is considered the best attempt at an improvement using extrapolation methods on the result of an asymptotic expansion for the difference \( I_n(t) - f(t) \) resulting from a specific delta sequence first proposed by Garver in 1966 [32]. The Stehfest inversion of the Laplace transform \( \tilde{f}(s) \) of a function of time \( f(t) \) is given by:

\[
f(t) = \ln \left( \frac{2}{t} \right) \sum_{n=1}^{N} K_n \tilde{f}(s_n)
\]

where the sequence of \( s \)-values is provided explicitly by:

\[
s_n = \frac{\ln 2}{t}
\]

and the series coefficients are:

\[
K_n = (-1)^{n+N/2} \sum_{k=\lceil(n+1)/2 \rceil}^{\min(n,N/2)} \frac{k^{N/2}(2k)!}{(N/2-k)!k!(k-1)!(n-k)!(2k-n)!}
\]

The coefficients \( K_n \) are computed once and stored. Double precision arithmetic is mandatory to obtain accurate solutions. This method has been shown to provide accurate inversion for heat conduction problems in the BEM literature and is adopted in this study as the method to invert Laplace-transformed BEM solutions. Typically, the upper limit in the series is taken as \( N = 12 \sim 14 \), as cited by
Stehfest [28]. However for these types of BEM solution inversions, Moridis and Reddell [34] reported little gains in accuracy for \( N = 6 \sim 10 \), and demonstrated accurate results using \( N = 6 \). Davies and Crann [35] also report accurate results using \( N = 8 \), for BEM problems with periodic boundary conditions. For this work \( N = 12 \) has been used following the original results of Stehfest, and for maximum accuracy. It is also notable that due to amplification effects of the large factorial coefficients, \( K_n \), on both round off and truncation errors, BEM solutions must be carried to very high levels of precision. For this reason very accurate integration, linear solver, and iteration routines are necessary in the BEM solution. This requirement acts to further increase the computational power and time needed for accurate transient results. The advantage of this inversion method still remains, however, due to its consistent requirements for any time solution. The computation is independent of the given time value, which is a major advantage over time-marching schemes that require much longer run times for large time solutions compared to small time solutions.

6 Numerical validation and examples

6.1 3D nonlinear heat conduction verification examples

Steady nonlinear heat conduction problems are now considered. All geometric lengths as well as temperatures are dimensionless. Thermophysical properties are scaled to the reference length and temperature. A validation example comparing a numerical and exact solution for a nonlinear problem in a rectangular slab is first presented. Here, the nonlinear conductivity is taken to vary as \( k(T) = 1 + T \), and use the following expression for temperature:

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

(52)

This expression satisfies the heat conduction equation, eqn (15), for the chosen thermal conductivity variation. The above is used to impose boundary conditions on a rectangular slab of length 4 and height and width of 1. The slab is discretized into four equal subregions with a total of 600 elements, and this corresponds to 600 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 BEM problem converged in a total time of 73 s including seven iterations for the constant element model to provide the initial guess for the bilinear case, one converged sweep of temperature and flux updates in the bilinear grid, and subsequently one converged sweep of temperature and flux updates in the biquadratic grid. The total time to solution reported here includes load balancing, generation of \( H \) and \( G \) matrices, and iteration to solution. The additional advantage of this approach is that it allows to state that the solution is grid independent as a grid refinement is performed on the
way toward the final solution. Results are plotted in Fig. 7a–c for the BEM computed and exact solutions as well as the absolute which is less than 0.0125 over $\Delta T_{\text{max}} = 1$.

Next, a cylinder of radius 1 and length 10 is considered. The cylinder is 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 discretization, and 16,640 DOF for the biquadratic discretization. Here, two cases are considered: (1) a rod with nonlinear conductivity taken as $k(T) = 1.93[1 + 9.07 \times 10^{-3}(T - 720)]$, and (2) a composite rod with endcaps 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 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 $h = 10$ while the perimeter is heated by convection with $h = 1$ and $T_\infty$ varying from 1000 to 4000. The timings and total iterations for convergence of the solutions are provided in Table 1. The results are illustrated in Fig. 8a and b in the form of surface temperature contour plots.
Table 1: Number of iterations and timings for the rod problem.

<table>
<thead>
<tr>
<th></th>
<th>Case 1 (Homogeneous)</th>
<th>Case 2 (Composite)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-P4 cluster ~ 2,080 elements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Constant elements (2,080 DOF)</td>
<td>5 iterations</td>
<td>9 iterations</td>
</tr>
<tr>
<td>Bilinear elements (8,320 DOF)</td>
<td>1 iteration</td>
<td>1 iteration</td>
</tr>
<tr>
<td>Biquadratic elements (16,640 DOF)</td>
<td>1 iteration</td>
<td>1 iteration</td>
</tr>
<tr>
<td>Total time to solution</td>
<td>284 s</td>
<td>292 s</td>
</tr>
</tbody>
</table>

Figure 8: Results from the BEM solution in a rod: (a) homogeneous rod with non-linear conductivity, temperature span over the rod is $\Delta T = 200 - 1800$, and (b) composite rod with nonlinear conductivity, temperature span over the rod is $\Delta T = 400 - 1400$.

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

Table 2: Number of iterations and timings for the blade problem.

<table>
<thead>
<tr>
<th>6 × P4 cluster ~ 5,014 elements</th>
<th>Case 1 (Linear)</th>
<th>Case 2 (Nonlinear)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant elements (5,014 DOF)</td>
<td>3 iterations</td>
<td>9 iterations</td>
</tr>
<tr>
<td>Bilinear elements (20,056 DOF)</td>
<td>1 iteration</td>
<td>1 iteration</td>
</tr>
<tr>
<td>Total time to solution</td>
<td>765 s</td>
<td>781 s</td>
</tr>
</tbody>
</table>

Again, two cases are solved: (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 are imposed on the surface of the blade. The plena are imposed with convective conditions with $T_\infty$ varying linearly from 300 to 500 along the plenum depth and $h = 5$. The end surfaces in the span-wise direction are insulated, while the remaining surfaces are imposed with convective conditions with $T_\infty = 1000$ and $h = 10$. The timings and number of iterations for convergence of the solutions are provided in Table 2. The results are illustrated in Fig. 10a and b as temperature contour plots. Only constant and bilinear analyses were performed in this case.

Next, a conduction problem in an intricate geometry of a film-cooled blade is considered. The domain decomposition and discretization for this blade are shown in Fig. 11a–c. Here the discretization is comprised of 21,306 elements distributed over 20 subdomains. This corresponds to 21,306 DOF for the constant elements discretization and 85,224 DOF for the bilinear discretization. Two cases are considered: (1) linear with a constant conductivity of $k = 1.34$, and (2) nonlinear with $k(T) = 1.09[1 + 4.29 \times 10^{-4}(T - 1620)]$. The endwall surfaces in the span-wise direction are taken as adiabatic, while temperature boundary conditions imposed on the blade surfaces were obtained from a conjugate analysis.
carried out on the blade coupling the 3D BEM code for heat conduction to the Glenn-HT finite volume code for the flow analysis [1, 2]. The temperatures varied from 1600–3100 over the surfaces exposed to film cooling, plenum air, and hot gas flowing over the external surfaces. The timings and number of iterations for convergence of the solutions are reported in Table 3. The solutions are shown in Fig. 12a and b as temperature contour plots. Here, only a constant element and a bilinear analysis were carried out.

Finally, a large-scale conduction problem in a U-tube is considered. The domain decomposition for this case is shown in Fig. 13. Here the discretization is comprised
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, and (c) close-up of mesh and domain decomposition.

<table>
<thead>
<tr>
<th></th>
<th>Case 1 (Linear)</th>
<th>Case 2 (Nonlinear)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant elements (21,306 DOF)</td>
<td>10 iterations</td>
<td>10 iterations</td>
</tr>
<tr>
<td>Bilinear elements (85,224 DOF)</td>
<td>1 iteration</td>
<td>1 iteration</td>
</tr>
<tr>
<td>Total time to solution</td>
<td>3,222 s</td>
<td>3,230 s</td>
</tr>
</tbody>
</table>
Figure 12: Plot of converged solutions for the film-cooled blade: (a) trailing edge linear model and (b) trailing edge for the nonlinear case.
Figure 13: Domain decomposition of U-tube, 44,640 element model and 31 subregions: (a) overall domain decomposition of U-tube and (b) close-up of mesh.

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

<table>
<thead>
<tr>
<th></th>
<th>12 × PC cluster ~ 44,640 elements</th>
<th>Case 1 (Linear)</th>
<th>Case 2 (Nonlinear)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant elements (44,640 DOF)</td>
<td>11 iterations</td>
<td>11 iterations</td>
<td>12 iterations</td>
</tr>
<tr>
<td>Bilinear elements (178,560 DOF)</td>
<td>1 iteration</td>
<td>1 iteration</td>
<td>1 iteration</td>
</tr>
<tr>
<td>Total time to solution</td>
<td>4,307 s</td>
<td>4,353 s</td>
<td></td>
</tr>
</tbody>
</table>

of 44,640 elements distributed over 31 subdomains. This corresponds to 44,640 DOF for the constant elements discretization and 178,560 DOF for the bilinear discretization. Again, two cases are considered modeling stainless steel as the composing material: (1) uniform conductivity taken as $k = 14.9$, and (2) nonlinear conductivity taken as $k(T) = 14.9[1 + 4.7 \times 10^{-4}(T - 500)]$. The perimetric surface of the U-tube is kept insulated while heat is added through one of the endcaps at a rate of 1000 and heat is removed through the other endcap by convection at 0 degrees with a heat transfer coefficient of 100. The timings and number of iterations for convergence of the solutions are reported in Table 4. The solutions are shown in Fig. 14a and b as temperature contour plots. Here, only a constant element and a bilinear analysis were carried out.
6.2 Transient heat conduction verification examples

The following section details the verification procedures used to test the transient domain decomposition Laplace transform BEM heat conduction solver described above. An initial test was performed to verify the accuracy of the BEM solution of the modified Helmholtz equation. A $L = 4$ by $l = 1$ rectangular region imposed with $\bar{T} = 0$ on the bottom and left walls and with $\bar{q} = -1$ on the top and right wall is considered. The selected parameters for the modified Helmholtz equation are: $s = 0.005$, $\rho = 1000$, $c = 2$, and $k = 1$. The analytical solution is given by the following series expansion:

$$\bar{T}(x, y) = \sum_{n=0}^{\infty} E_n \sinh(\delta_n x) \sin(\lambda_n y) + \sum_{n=0}^{\infty} F_n \sin(\gamma_n x) \sinh(\nu_n y)$$

(53)

where

$$E_n = \frac{2}{D_n \delta_n \cosh(\delta_n L)}$$

$$\lambda_n = \frac{\pi}{2l}(2n + 1) \quad \text{and} \quad \delta_n = \sqrt{\frac{\lambda_n^2}{4} + \rho cs}$$

$$F_n = \frac{2}{L \gamma_n \nu_n \cosh(\nu_n l)}$$

$$\gamma_n = \frac{\pi}{2L}(2n + 1) \quad \text{and} \quad \nu_n = \sqrt{\frac{\gamma_n^2}{4} + \rho cs}$$

(54)
The analytical solution followed by the BEM solutions for a varying number of regions are displayed in Fig. 15a–d in the form of contour plots. A total of 100 quadratic discontinuous boundary elements were employed in the single region case and 10 additional boundary elements were added at each interface for the multi-region cases. The accuracy of the BEM routine is clearly shown, as the iso-lines are virtually identical to the exact solution, including the case of multiple regions cases. The stability of the BEM solver is also readily recognized in the plot of residual vs. iteration for the multiple region cases shown in Fig. 16. Here, it is shown that the residual decays rapidly for both multi-region cases. The convergence criterion used was $\epsilon = 10^{-8}$ and achieved in less than 10 iterations for the four-region case.
It should be noted that the two-region model below corresponds to a storage reduction of 64% when compared to a single region, while the four-region case obtains an 84% reduction in RAM memory requirements. As mentioned in the multi-region discussion above, the number of FLOPS is also significantly dropped, with a 56% and 74% reduction in FLOPS for the two- and four-region cases, respectively.

Next, a transient test was performed to verify the accuracy of the BEM solution combined with the numerical inversion of the Laplace transform. Again, a \( L = 4 \) by \( l = 1 \) rectangular region imposed with \( T = 0 \) on the bottom and left walls and with \( q = -1 \) on the top and right walls is considered. The domain is initially at a temperature \( T(x, y, 0) = 0 \). The selected parameters for the diffusion equation are: \( \rho = 1000 \), \( c_p = 2 \), and \( k = 1 \). A total of 100 quadratic discontinuous boundary elements were employed in the single-region case and 10 additional boundary elements were added at each interface for the multi-region cases. An analytical solution was derived for the problem at hand yielding the following form:

\[
T(x, y, t) = T_{ss}(x, y) + \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} C_{nm} \sin(\beta_n x) \sin(\lambda_m y)e^{-\alpha \beta_n^2 t}  \tag{55}
\]

where

\[
T_{ss}(x, y) = \sum_{n=0}^{\infty} E_n \sinh(\delta_n x) \sin(\gamma_n y) + \sum_{n=0}^{\infty} F_n \sin(\gamma_n x) \sinh(\gamma_n y) \tag{56}
\]
with
\[
E_n = \frac{2}{l \beta_n^2 \cosh (\delta_n L)}
\]
\[
\delta_n = \frac{\pi}{2l} (2n + 1)
\]
\[
F_n = \frac{2}{L \gamma_n^2 \cosh (\gamma_n l)}
\]
\[
\gamma_n = \frac{\pi}{2l} (2n + 1)
\]
and
\[
C_{nm} = \frac{4}{L^2} \int_0^L \int_0^l T_0(x, y) \sin (\beta_n x) \sin (\lambda_m y) dy dx
\]
with
\[
T_0(x, y) = T(x, y, 0) - T_{ss}(x, y)
\]
\[
\beta_n = \frac{\pi}{2L} (2n + 1)
\]
\[
\lambda_m = \frac{\pi}{2l} (2m + 1)
\]
\[
\delta_{nm} = \sqrt{\lambda_m^2 + \beta_n^2}
\]

The analytical solutions followed by the BEM solutions for a varying number of regions are displayed in Fig. 17a–d in the form of contour plots. This time, the

Figure 17: Contour plots of the transient temperature solution: (a) exact solution at \(t = 25\) s, \(t = 100\) s, \(t = 200\) s, \(t = 500\) s; (b) one-region BEM at \(t = 25\) s, \(t = 100\) s, \(t = 200\) s, \(t = 500\) s; (c) two-region BEM at \(t = 25\) s, \(t = 100\) s, \(t = 200\) s, \(t = 500\) s; and (d) four-region BEM at \(t = 25\) s, \(t = 100\) s, \(t = 200\) s, \(t = 500\) s.
temperature contours are shown at different times ($t = 25\, \text{s}$, $t = 100\, \text{s}$, $t = 200\, \text{s}$, $t = 500\, \text{s}$) and excellent agreement is found between the exact and BEM solutions in both single and multi-region cases. In addition, a plot of the temperature evolution from $t = 0\, \text{s}$ to $t = 1000\, \text{s}$ is shown in Fig. 18 for the point $(x, y) = (3.8, 0.8)$, revealing virtually perfect agreement between exact and BEM solutions.
Figure 18: Exact and BEM temperature evolution and deviations at point \((x, y) = (3.8, 0.8)\).

Figure 19 depicts the residuals of the iterative BEM solution, for the region case for three of the twelve total inverse transform steps at \(t = 100\) s. Although the norm for the first case, \(s1\), takes the longest to decay, it is the least critical as the transform coefficient is small for the first steps. The convergence criterion for the multi-region BEM iteration process for the transient case was reduced to \(\epsilon = 10^{-15}\).
to account for the error magnification caused by the large factorials found in the Stehfest transform algorithm.

The final example presented here shows a transient heat transfer analysis performed on a typical guide vane under the estimated turbine conditions given below. The vane geometry is shown in Fig. 20, along with the applied convective boundary condition equations, noting that all temperatures are measured in degrees centigrade above ambient. The free-stream temperature, $T_\infty$, has been assumed constant at 2500 degrees, and the heat transfer coefficient is given an exponentially decaying value with the maximum at the vane tip. As this is only a showcase of the abilities of the conduction solver all values have been estimated and are not perfectly representative of turbine conditions. The results of the analysis show the smooth, rapid propagation of heat through the vane over time in Fig. 21a–d.

The same vane shape is then investigated with the addition of cooling passages using a coolant flow at 1800 degrees, see Fig. 22, two-passage case. The second
Figure 21: Temperature fields at various times for a solid turbine guide vane: 
(a) \( t = 0.2 \) s and \( t = 0.4 \) s, (b) \( t = 0.6 \) s and \( t = 0.8 \) s, (c) \( t = 1.0 \) s, and (d) temperature scale for all time values.

Figure 22: BEM discretization and BCs for the turbine guide vane problem with cooling passages.

The case contains a single, centrally located cooling passage. Figure 23a–d show the slow down in heat transfer caused by the addition of the single cooling passage. The final case shows how multiple passages may also be employed to produce similar effects, see Fig. 24a–d. It is noted that both cases increase the heat up time as desired, but neither significantly reduces the steady temperature values. The hottest point on the vane is the front tip and the temperature of the tip is tracked and plotted over time for each of the three cases above in Fig. 25.
Figure 23: Temperature fields for the turbine guide vane with cooling passage: (a) \( t = 0.2 \) s and \( t = 0.4 \) s, (b) \( t = 0.6 \) s and \( t = 0.8 \) s, (c) \( t = 1.0 \) s and (d) temperature scale for all time values.

Figure 24: Temperature fields for the turbine guide vane with two cooling passages: (a) \( t = 0.2 \) s and \( t = 0.4 \) s, (b) \( t = 0.6 \) s and \( t = 0.8 \) s, (c) \( t = 1.0 \) s and (d) temperature scale for all time values.
7 Conclusions

The BEM is often an efficient choice for the solution of various engineering field problems as it acts to decrease the dimensionality of the problem. However, the solution of large problems is still prohibitive as the BEM coefficient matrices are typically fully populated and difficult to subdivide or compress. This chapter has presented an efficient iterative domain decomposition method to reduce the storage requirement and allow the solution of such large-scale problems. The decomposition approach lends itself ideally to parallel message passing type computing due to the independence of each of the BEM subregion solutions. With this approach, large-scale problems can be readily solved on small PC clusters. The domain decomposition approach is setup in general and can be applied to any type of BEM problem. This method of domain decomposition was then applied to steady, 3D, linear, and nonlinear heat conduction as well as to transient, 2D, linear heat conduction problems. These formulations are general and can be directly applied to other field problems such as elasticity, thermoelasticity, and acoustics. Results have been presented and excellent agreement with analytical solutions was found.

Acknowledgment

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


