Efficient Large Scale Transient Heat Conduction Analysis Using A Parallelized Boundary Element Method

2006

Kevin Erhart
University of Central Florida

Find similar works at: https://stars.library.ucf.edu/etd

University of Central Florida Libraries http://library.ucf.edu

Part of the Mechanical Engineering Commons

STARS Citation

https://stars.library.ucf.edu/etd/974

This Masters Thesis (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of STARS. For more information, please contact lee.dotson@ucf.edu.
EFFICIENT LARGE SCALE TRANSIENT HEAT CONDUCTION ANALYSIS
USING A PARALLELIZED BOUNDARY ELEMENT METHOD

by

KEVIN JAMES ERHART
B.S. University of Central Florida, 2004

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Mechanical, Materials, and Aerospace Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2006
ABSTRACT

A parallel domain decomposition Laplace transform Boundary Element Method, BEM, algorithm for the solution of large-scale transient heat conduction problems will be developed. This is accomplished by building on previous work by the author and including several new additions (most note-worthy is the extension to 3-D) aimed at extending the scope and improving the efficiency of this technique for large-scale problems. A Laplace transform method is utilized to avoid time marching and a Proper Orthogonal Decomposition, POD, interpolation scheme is used to improve the efficiency of the numerical Laplace inversion process. A detailed analysis of the Stehfest Transform (numerical Laplace inversion) is performed to help optimize the procedure for heat transfer problems. A domain decomposition process is described in detail and is used to significantly reduce the size of any single problem for the BEM, which greatly reduces the storage and computational burden of the BEM. The procedure is readily implemented in parallel and renders the BEM applicable to large-scale transient conduction problems on even modest computational platforms.

A major benefit of the Laplace space approach described herein, is that it readily allows adaptation and integration of traditional BEM codes, as the resulting governing equations are time independent. This work includes the adaptation of two such traditional BEM codes for steady-state heat conduction, in both two and three dimensions. Verification and validation example problems are presented which show the accuracy and efficiency of the techniques. Additionally, comparisons to commercial Finite Volume Method results are shown to further prove the effectiveness.
# TABLE OF CONTENTS

LIST OF FIGURES............................................................................................................ iv

LIST OF TABLES.............................................................................................................. vi

LIST OF SYMBOLS......................................................................................................... vii

CHAPTER 1 INTRODUCTION......................................................................................... 1

CHAPTER 2 TRANSIENT HEAT CONDUCTION USING THE LAPLACE TRANSFORM............................................................................................................. 6

CHAPTER 3 BOUNDARY ELEMENT METHOD......................................................... 10
  3.1 Boundary Integral Development................................................................. 10
  3.2 Boundary Element Method Development............................................... 13

CHAPTER 4 DOMAIN DECOMPOSITION AND PARALLELIZATION.................... 15
  4.1 Domain Decomposition Procedures......................................................... 15
  4.2 Parallelization............................................................................................ 20

CHAPTER 5 NUMERICAL LAPLACE INVERSION.................................................... 22

CHAPTER 6 VERIFICATION EXAMPLES............................................................... 26

CHAPTER 7 PRACTICAL PROBLEMS......................................................................... 36

CHAPTER 8 PROPER ORTHOGONAL DECOMPOSITION........................................43
  8.1 Laplace Space Approach............................................................................. 44
  8.2 Time Domain Approach.............................................................................. 45
  8.3 Numerical Study of POD Approaches...................................................... 46

CHAPTER 9 CONCLUSIONS AND EXTENSIONS...................................................... 51

REFERENCES.................................................................................................................. 53
LIST OF FIGURES

Figure 1: BEM problem domain, boundary conditions, and single-region discretization.................................................................16

Figure 2: BEM multi-region decomposition and discretization.................................17

Figure 3: Graphical representation of Stehfest Transform errors.............................25

Figure 4: Verification example geometry and boundary conditions.........................26

Figure 5: Contour plots of the temperature solution for the transient verification problem ...............................................................................................................28

Figure 6: Multi-region BEM temperature accuracy comparisons..........................30

Figure 7: Temperature and Error in Temperature $\nabla t = 250 \text{sec}$ for several levels of grid refinement for 3-D cube.......................................................32

Figure 8: Temperature $\nabla t = 250 \text{sec}$ for several values inversion steps $N_\gamma$, for 3-D cube................................................................................................................33

Figure 9: Temperature throughout time at several positions for 3-D cube..................33

Figure 10: Contour plots of the temperature field at various times for 3-D cube........35

Figure 11: Laminar airfoil meshes (convective BC's imposed on all boundaries)...........37

Figure 12: BEM comparison to finite volume solver (Fluent 6.1)..............................38

Figure 13: BEM discretization and BC's for non-symmetric airfoil with cooling passages..................................................................................................................39

Figure 14: Airfoil leading and trailing edge region temperatures over time..............39

Figure 15: Temperature field comparison for non-symmetric airfoil with cooling passages..................................................................................................................40

Figure 16: Contour plots of the temperature field at various times for 3-D turbine blade..................................................................................................................41

Figure 17: One-Dimensional transient verification example problem....................46

Figure 18: Results at various time levels for 1-D conduction problem.....................48

Figure 19: Results at a location of $x = 100 \text{mm}$ throughout time.........................49
Figure 20: Non-symmetric airfoil problem temperatures throughout time at two discrete locations (one near the trailing edge and another near a cooling passage)........50
LIST OF TABLES

Table 1: Stehfest Transform sample size comparison....................................................... 23

Table 2: s-value comparison for two time value sets....................................................... 25
# LIST OF SYMBOLS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho$</td>
<td>Density</td>
</tr>
<tr>
<td>$k$</td>
<td>Thermal Conductivity</td>
</tr>
<tr>
<td>$c$</td>
<td>Specific Heat</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Thermal Diffusivity</td>
</tr>
<tr>
<td>$T$</td>
<td>Temperature</td>
</tr>
<tr>
<td>$q$</td>
<td>Normal Heat Flux</td>
</tr>
<tr>
<td>$s$</td>
<td>Laplace Transform Parameter</td>
</tr>
<tr>
<td>$x, y, z$</td>
<td>Spatial Coordinates</td>
</tr>
<tr>
<td>$t$</td>
<td>Time</td>
</tr>
<tr>
<td>$\Gamma$</td>
<td>Boundary of Integration (Control Surface)</td>
</tr>
<tr>
<td>$\Omega$</td>
<td>Domain of Integration (Control Volume)</td>
</tr>
<tr>
<td>$\xi$</td>
<td>Perturbation Point (Source Location)</td>
</tr>
<tr>
<td>$x_i, y_i, z_i$</td>
<td>Coordinates of Perturbation Point</td>
</tr>
<tr>
<td>$N_s$</td>
<td>Number of Stehfest Inversion Steps</td>
</tr>
<tr>
<td>$N$</td>
<td>Number of Boundary Elements</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

It is well known that most practical engineering problems, especially heat transfer and fluid mechanics, cannot be solved exactly using analytical means, which is most often due to the presence of irregular geometries, complex boundary conditions, or non-linearity in either material properties or the governing equation itself. For these types of problems, solutions are most often sought using numerical techniques such as the finite element method (FEM), the finite volume method (FVM), or the boundary element method (BEM). The FEM has quite wide-spread use in the area of mechanics, while the FVM is most often used in fluid mechanics and heat transfer applications. While both of these techniques cover a very wide range of applications and have been refined to a level such that they are very efficient in solving large problems, they still involve approximating the problem at hand using a mesh or grid. It is well known that meshing procedures are far from automated and the generation of a suitable mesh is typically the majority of the effort needed in the employment of such methods. Additionally, the results of these methods are often highly dependent upon the quality of the mesh used. The latter method, BEM, however, is often considered the most efficient overall, as only the boundary of a problem (not the interior) needs to be meshed. This feature arises due to the use of boundary-only integral equations, which eliminate domain integrals and reduce the dimensionality of the problem by one order. This creates great savings, mostly in the problem setup time, as the ease of meshing is greatly reduced, due to this reduction in dimensionality. Although the setup time is reduced, BEM generally requires the solution of fully populated matrix systems; typically these matrices are much smaller than FVM or FEM matrices, but are still much more computationally intensive to solve,
due to the lack of symmetry and bandedness that are generally present in FEM and FVM.
The work of this thesis attempts to increase the efficiency of the BEM in this area. The
boundary element method has been applied to numerous engineering problems including
solid mechanics, heat transfer, and acoustics, with details on the general formulations and
applications found in Brebbia, Telles, and Wrobel [1], and as such any improvements in
the general technique, such as those proposed herein, could have applications in any such
area. This work will show the effectiveness of these new developments for the specific
case of transient heat conduction.

The BEM has traditionally been used to solve transient heat conduction problems using
three different approaches: (1) A convolution scheme incorporating a time-dependent
Green's function to build a transient boundary-only integral equation model, (2) Dual
Reciprocity Method (DRM) to expand the spatial portion of the governing equation using
Radial-Basis functions and a finite difference approach to march in time, as discussed by
Partridge, Brebbia, and Wrobel [2], and (3) A Laplace transformation of the governing
equation to eliminate the time dependence, which then allows the use of standard steady-
state type BEM techniques. The first approach will require the generation and storage of
the BEM influence coefficient matrices at every time step of the convolution scheme.
These huge storage needs make this method unrealistic for medium- or large-scale
problems. The second approach poses different problems as the global interpolation
functions for the Dual Reciprocity technique, such as the widely used Radial-Basis
functions, lack convergence and error estimation, and often induce unwanted behavior
which increases the conditioning number of the resulting algebraic system. Additionally,
the DRM techniques require almost a full re-write of traditional BEM codes, making this
approach rather difficult to implement. The third approach, originated in BEM
application by Rizzo and Shippy [3], does not require time marching or any type of
interpolation and thus it has the greatest possibility for the development of a highly efficient transient BEM algorithm; one such technique is proposed by Narayanan and Beskos [4]. It does however, require fine-tuning of the BEM solution of the Modified Helmholtz equation and a numerical Laplace inversion of the results. Another advantage of this method is its ease of implementation, as traditional BEM codes can be employed with only slight modifications and only a few additional routines.

This thesis will formulate a Laplace-transformed BEM algorithm, structured with an iterative, parallel, domain decomposition scheme, which acts to significantly reduce the computational and storage requirements of large-scale problems. As the development of an efficient solver for large-scale transient heat conduction problems is the goal of this work, efficiency improvements for BEM algorithms will be discussed throughout the text, and a chapter has been devoted to a Proper Orthogonal Decomposition, POD, scheme which provides improvements within the Laplace inversion segment. This POD approach is shown to reduce the overall number of BEM Laplace space solutions required for inversion by up to 50% for many problems. The domain decomposition approach has been adopted to address the storage and computational issues imposed by large-scale problems. In this method the problem domain is artificially sectioned into sub-domains, each of which is independently tackled with standard BEM. This method of domain decomposition has proven effective in reducing the overall computational and memory requirements typically needed for the solution of large-scale problems. This technique was developed by Divo, Kassab, and Rodriguez [5] for steady-state applications, and a similar decomposition approach is employed by Ingber, et al [6] for 3-D diffusion problems utilizing a time-discretization approach. Parallelization (running multiple processes simultaneously on many computers) is used to achieve further computational savings. The sub-regions are distributed over the nodes of a parallel
computer cluster using Message Passage Interface (MPI) libraries for processor communications. An iteration routine is employed to satisfy continuity conditions at artificially created interfaces between sub-domains. Another type of parallelization has also been discussed by Davies and Crann [7], 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 solution is found with a single large matrix, as opposed to several small matrices, as in the domain decomposition procedure discussed herein. A chapter is devoted specifically to describing these types of parallelization procedures and the domain decomposition procedure, including a combination of the two methods that has proved quite effective.

This thesis will begin with the derivation of a boundary-only integral equation suitable for the BEM by applying the Laplace transform to the transient heat conduction equation. This development is shown for two-dimensional applications for brevity as it is easily extended to three-dimensions and the necessary fundamental solutions for 3-D applications are shown herein as well. Following this process, the general methods of the BEM are developed employing the ideas discussed above to complete the BEM solution technique. The processes of domain decomposition and parallelization are then addressed in further detail and then the final step of the solution process, the numerical Laplace inversion, is developed using the Stehfest transform [8-9]. A small optimization study is also presented for the Stehfest process as used in this approach. The following chapters are then devoted to providing numerical examples to demonstrate both the accuracy and efficiency of this work, in both 2-D and 3-D, including comparisons to commercially available software. This is followed by a chapter on Proper Orthogonal Decomposition, which is used for further efficiency improvements in the inversion process. And finally a
summary chapter is presented, along with concluding remarks and directions for future work are discussed.
Many methods exist to solve the transient heat conduction (diffusion) equation, either numerically or analytically. Methods such as superposition allow a single problem to be broken up into a set of less complex problems, which facilitates the solution of some complex problems. The reduction of complexity is also the goal of the Laplace transform approach, which attempts to reduce a differential equation to an algebraic expression or reduce the order of an equation. The Laplace transform, when applied to the diffusion equation, eliminates the time dependency of the equation and yields a "steady" type equation. This equation is a spatially dependent, constant coefficient, partial differential equation. The analytical or numerical methods used to solve this equation are now much less complicated in the Laplace space. Whichever method is used, the resulting solution is converted back to the time domain using an inverse Laplace transformation which completes the problem solution. The Laplace transform process is detailed below and marks the first step toward the development of a boundary integral equation for the expression of the diffusion equation.

The Laplace transform of a function, \( f(t) \), is defined by the following expression:

\[
\mathcal{L}\{f(t)\} = \overline{F}(s) = \int_{0}^{\infty} e^{-st} f(t) \, dt
\]  

(1)

where an over-bar is used to represent a function in the transformed space. As mentioned above, the formulation of this approach will be discussed for two dimensional heat conduction problems, but the necessary additions for three dimensions have been
included as well. The general differential form of the two-dimensional transient heat conduction equation is given below:

$$\frac{\partial}{\partial x} \left[ k \frac{\partial T}{\partial x}(x, y, t) \right] + \frac{\partial}{\partial y} \left[ k \frac{\partial T}{\partial y}(x, y, t) \right] = \rho c \frac{\partial T}{\partial t}(x, y, t)$$

(2)

The initial formulation of this project will deal only with isotropic materials of constant thermal conductivity, $k$, density, $\rho$, and specific heat, $c$. The above properties can be combined for simplicity by defining the thermal diffusivity, $\alpha$ as:

$$\alpha = \frac{k}{\rho c}$$

(3)

These constants can now be removed from the differentials and the diffusion equation reduces to:

$$\frac{\partial^2 T}{\partial x^2}(x, y, t) + \frac{\partial^2 T}{\partial y^2}(x, y, t) = \frac{1}{\alpha} \frac{\partial T}{\partial t}(x, y, t)$$

(4)

The above equation represents the general form of the diffusion equation that is used throughout this work.

Applying the definition of the Laplace transform, given above in Equation 1, to the heat conduction equation, Equation 4, yields the following expression:

$$\nabla^2 \hat{T}(x, y, s) = \frac{s}{\alpha} \hat{T}(x, y, s) - \frac{1}{\alpha} \hat{T}(x, y, 0)$$

(5)

where $\nabla^2$ represents the Laplace operator. The above expression can be further reduced by employing a shifted temperature definition as:

$$\theta(x, y, t) = T(x, y, t) - T(x, y, 0)$$

(6)

The solution procedure for $\theta$ will be identical to that for $T$ as long as the initial condition, $T(x, y, 0)$, is harmonic (satisfies the Laplace equation). If the initial condition does not satisfy this constraint, the shift to $\theta$ will cause the appearance of a generation-like term in
Equation 5, which means an approach such as Dual Reciprocity BEM or the method of particular solutions would be needed (these techniques are detailed extensively by Partridge et al. [2], and are not the focus of this work). Since the solution for $\theta$ and $T$ are identical, the discussion below will be left in terms of $T$, and reader is reminded that such a temperature shift is needed for non-zero initial condition problems. It is also noteworthy that almost all problems will begin from an initially steady-state, which means that the harmonic condition is satisfied for almost all problems.

Equation 5 is now no longer a function of time, but 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 $s$ is thus removed, and the temperature shift applied, resulting in the following expression:

$$\nabla^2 T(x, y) - \frac{s}{\alpha} T(x, y) = 0$$

(7)

As stated above, the material properties will be assumed independent of temperature and space, therefore the above expression has constant coefficients and has the form of the Modified Helmholtz Equation:

$$\nabla^2 \psi - m^2 \psi = 0$$

(8)

where $m^2 = s/\alpha$. The methods of solution to this equation are 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) [10]. The BEM has been effectively implemented for the solution of such acoustic propagation problems; therefore it is known that the Laplace transformation of the diffusion equation has yielded a suitable form for the desired solution method.

Additionally, the boundary conditions must be transformed in order to refer the entire problem to the proper Laplace Transform space. The boundary conditions are
transformed using the following relations (assuming the boundary conditions are constant in time):

\[
T(x, y, s) \bigg|_\Gamma = \frac{\hat{T}(x, y)}{s} \bigg|_\Gamma
\]  

(9)

\[
\bar{q}(x, y, s) \bigg|_\Gamma = \frac{\hat{q}(x, y)}{s} \bigg|_\Gamma
\]  

(10)

where, \(\Gamma\), represents the boundary (control surface). For mixed type (convective) boundary conditions the fluid temperature is transformed using Equation 9 above, while the convective coefficient remains unchanged. Equations 7, 9, and 10 above mark the completion of a properly-posed boundary value problem for the solution of transient heat conduction problems. The next step in the process will be obtaining a boundary-only integral equation from these relationships, which is detailed in the following chapter.
CHAPTER 3

BOUNDARY ELEMENT METHOD

Much of the reasoning behind the choice of using the BEM for this application has been discussed in previous chapters, but again, the major advantage is the reduction of dimensionality. The reduction of dimensionality most importantly creates time savings for the end-user of the program. The user is benefited by the reduced meshing requirements. For large complex problems, creating the mesh is often the most time consuming task, and BEM programs require far less meshing than FVM or FEM programs. For 3-D BEM problems only a 2-D boundary surface must be meshed, which eliminates the complication of interior volume meshing and avoids many of the areas that require very careful meshing in other techniques. The numerical implementation of the BEM is generally considered more difficult and the numerical solution itself is not as efficient, but the mesh savings often more than make up for these issues. The development of a boundary integral equation, which is used as the basis for the formulation of the BEM, is described in the following section.

3.1 Boundary Integral Development

The development of a boundary element method solution will begin by reducing the governing equation to a boundary-only integral equation. The current form of the Laplace transformed, transient heat conduction equation (Equation 7) can be converted from differential to integral form by multiplying through by a function $G$, yet to be determined, and integrating over the domain, as follows:
\[
\int_{\Omega} G(x, y, \xi) \left[ \nabla^2 T(x, y) - \frac{s}{\alpha} T(x, y) \right] d\Omega = 0 \tag{11}
\]

where \( \Omega \) represents the entire domain (control volume or area). The above equation is expanded to obtain the expression below:

\[
\int_{\Omega} \left[ G(x, y, \xi) \left( \nabla^2 T(x, y) \right) \right] d\Omega - \frac{s}{\alpha} \int_{\Omega} G(x, y, \xi) T(x, y) d\Omega = 0 \tag{12}
\]

Integration by parts is applied twice to the above equation (Green's Second Identity) and the following expression is obtained (the dependencies have been omitted for clarity):

\[
\oint_{\Gamma} G \frac{\partial T}{\partial n} d\Gamma - \oint_{\Gamma} T \frac{\partial G}{\partial n} d\Gamma + \int_{\Omega} \left[ T \left( \nabla^2 G \right) - \frac{s}{\alpha} G T \right] d\Omega = 0 \tag{13}
\]

where \( \Gamma \) represents the boundary of interest (control surface or edge). Some terms of the above expression can be combined to simplify the remaining manipulations by defining the following quantities:

\[
\bar{q}(x, y) = -k \frac{\partial T}{\partial n}(x, y) \tag{14}
\]

\[
H(x, y, \xi) = -k \frac{\partial G}{\partial n}(x, y, \xi) \tag{15}
\]

Using these definitions Equation 13 can be rewritten as:

\[
k \int_{\Omega} T \left[ \nabla^2 G - \frac{s}{\alpha} G \right] d\Omega = \oint_{\Gamma} G \bar{q} d\Gamma - \oint_{\Gamma} H T d\Gamma \tag{16}
\]

The unknown function \( G(x, y, \xi) \), is seen to be the free-space solution, and can now be determined as the solution of the adjoint equation perturbed by a Dirac Delta function, \( \delta(x, y, \xi) \) applied at the source point, \( \xi \). This free space solution, \( G(x, y, \xi) \), is found by letting the following expression hold true:

\[
\nabla^2 G(x, y, \xi) - \frac{s}{\alpha} G(x, y, \xi) = -\frac{1}{\alpha} \delta(x, y, \xi) \tag{17}
\]
This equation is then solved in free-space to determine the fundamental solution in 2-D applications as:

\[
G(x, y, \xi) = \frac{1}{2\pi \alpha} K_0\left(\sqrt{\frac{s}{\alpha}} r\right)
\]  

(18)

where \( K_0 \) is a Modified Bessel Function of the second kind of order zero, and \( r \) is the distance from the source point, \( \xi_i = (x_i, y_i) \), to the point of integration \( (x, y) \) defined as:

\[
r = \sqrt{(x - x_i)^2 + (y - y_i)^2}
\]

And for 3-D applications the solution is found to be:

\[
G(x, y, z, \xi) = \frac{1}{4\pi \alpha r} \exp\left(-\sqrt{\frac{s}{\alpha}} r\right)
\]

(19)

Now that the appropriate form of the function \( G(x, y, \xi) \) has been found, the definition of \( H(x, y, \xi) \), repeated below, can be used to determine its value (in 2-D) as:

\[
H(x, y, \xi) = -k \frac{\partial G}{\partial n}
\]

(20)

\[
H(x, y, \xi) = -\frac{k}{2\pi \alpha} \left[ \frac{\partial K_0}{\partial x} \left(\sqrt{\frac{s}{\alpha}} r\right) n_x + \frac{\partial K_0}{\partial y} \left(\sqrt{\frac{s}{\alpha}} r\right) n_y \right]
\]

(21)

Noting that the derivative of \( K_0(z) \) is \(- K_1(z)\), the above expression for 2-D is simplified to the following:

\[
H(x, y, \xi) = -\frac{k}{2\pi \alpha r} \sqrt{\frac{s}{\alpha}} K_1\left(\sqrt{\frac{s}{\alpha}} r\right) [(x - x_i) n_x + (y - y_i) n_y]
\]

(22)

And in 3-D as:

\[
H(x, y, z, \xi) = -\frac{\exp\left(-\sqrt{\frac{s}{\alpha}} r\right)}{4\pi \alpha r^2} \left(1 + \sqrt{\frac{s}{\alpha}}\right) k \frac{\partial r}{\partial n}
\]

(23)

Finally, the desired boundary-only integral equation is obtained (omitting spatial dependences for generality), using the appropriate forms of \( G \) and \( H \) found above in equations 18-23, as:
$
\frac{k}{\alpha} C(\xi) \bar{T}(\xi) = \int_{\Gamma} H \bar{T} \, d\Gamma - \int_{\Gamma} G \bar{q} \, d\Gamma \tag{24}$

where $C(\xi)$ is determined geometrically and for smooth boundaries is equal to $\frac{1}{2}$. Now that a boundary-only integral equation has been found, the development can continue by discussing more on the procedures needed for the implementation of the boundary element method.

### 3.2 Boundary Element Method Development

The typical formulation of the BEM begins by dividing the domain boundary into discrete elements. Much like FEM the choice of element order and type is important for obtaining accurate results. This work uses linear discontinuous boundary elements for 2-D and Bi-linear discontinuous elements for 3-D. It is important to note that this division of the boundary is the only approximation made by the BEM thus far as the above formulation was an exact procedure. This section will detail the BEM process of obtaining an algebraic system of equations from the discretized boundary field.

The first step of the BEM is to divide the boundary, $\Gamma$, into discrete elements such that:

$$
\Gamma = \sum_{j=1}^{N} \Delta \Gamma_j \tag{25}
$$

The final boundary integral equation (Equation 24) is now rewritten as the summation of the integral over each boundary element (with $x$ being a general coordinate set in either two or three dimensions), as shown below:

$$
\frac{k}{\alpha} C(\xi) T(\xi) = \sum_{j=1}^{N} \int_{\Delta \Gamma_j} \bar{H}(x, \xi) T(x) \, d\Gamma_j - \sum_{j=1}^{N} \int_{\Delta \Gamma_j} G(x, \xi) \bar{q}(x) \, d\Gamma_j \tag{26}
$$
This equation is now collocated at points $\xi_i$, where $i$ denotes the current boundary node and $j$ denotes the element of integration, in order to obtain the expression below (again omitting spatial dependencies):

$$\frac{k}{\alpha} C_i T_i = \sum_{j=1}^{N} \hat{H}_{ij} T_j - \sum_{j=1}^{N} G_{ij} q_j$$

(27)

where $G_{ij} = \int_{\Delta \Gamma_j} G(x, \xi_i) d\Gamma$ and $\hat{H}_{ij} = \int_{\Delta \Gamma_j} H(x, \xi_i) d\Gamma$. Equation 27 can now be simplified to the following expression:

$$\sum_{j=1}^{N} H_{ij} \bar{T}_j = \sum_{j=1}^{N} G_{ij} \bar{q}_j$$

(28)

where $H_{ij} = \hat{H}_{ij} - \frac{k}{2\alpha} \delta_{ij}$, such that $\delta_{ij} = 0$ if $i \neq j$, and $\delta_{ij} = 1$ if $i = j$, and noting that $C_i = \frac{1}{2}$ as the discontinuous elements used herein produce smooth boundaries at all collocation nodes. Once the known boundary conditions are applied this system can be rearranged to a set of equations in the standard algebraic form $[A]\{x\} = \{b\}$. This system of equations can than be solved by standard linear algebra methods, and the BEM solution is ready to be inverted numerically from the Laplace space domain to the real (time) domain. The numerical inversion routine is an integral part of the overall solution and Chapter 5 is devoted to this topic, which proceeds a discussion on parallelization and domain decomposition in the following chapter.
CHAPTER 4

DOMAIN DECOMPOSITION AND PARALLELIZATION

Domain decomposition is the process of splitting a problem domain into a series of smaller pieces, called sub-regions, and is the most critical aspect of rendering the BEM applicable to large-scale problems. Domain decomposition also allows the process to easily be parallelized, as each sub-region can be solved independently of the others. The following sections will detail the process of implementing domain decomposition (in 2-D) as well as two different methods of parallelization.

4.1 Domain Decomposition Procedures

First, the procedure of domain decomposition and the multi-region BEM iteration process will be described. Initially the problem domain, $\Omega$, is identified along with the corresponding boundary conditions over the entire boundary, $\Gamma$. A typical problem definition along with the corresponding boundary conditions and a sample single-region BEM discretization are depicted in Figure 1.

If a standard BEM solution process were to be adopted from the discretization of Figure 1, a system of influence coefficient matrices and boundary values of size $N$, where $N$ is the number of boundary nodes used to discretize the problem, will be formed as:

$$\nabla^2 \bar{T} - \frac{s}{\alpha} \bar{T} = 0 \Rightarrow [H]\{\bar{T}\} = [G]\{\bar{q}\}$$

(29)

The number of floating point operations required to arrive at the algebraic system shown above is proportional to $N^2$, the direct memory allocation is also proportional to $N^2$. 

15
Later, with the aid of the boundary condition distribution, the system is re-arranged into standard algebraic form as:

\[ [H]\{T\} = [G]\{\bar{q}\} \Rightarrow [A]\{x\} = \{b\} \tag{30} \]

where \(\{x\}\) represents the unknowns, either \(\{T\}\) or \(\{\bar{q}\}\) around the boundary. The solution to the algebraic system for the boundary unknowns can be performed using a direct solution method such as LU decomposition, requiring floating point operations proportional to \(N^3\) or an indirect method such as Bi-conjugate Gradient or Generalized Minimum Residual (GMRes) which, in general, require floating point operations proportional to \(N^2\) to achieve convergence. Since the domain decomposition procedure will reduce the size, \(N\), of each BEM problem, the amount of savings will be tremendous in both memory and computational time as these requirements grow with either the square or cube of \(N\).

\[ \begin{align*}
\nabla \cdot [k \nabla \bar{T}(x, y)] - \rho cs\bar{T}(x, y) &= 0 \\
\Gamma_1, \bar{T}_1, \bar{q}_1, or \bar{q}_1 &\propto \bar{T}_4 \\
\Gamma_2, \bar{T}_2, \bar{q}_2, or \bar{q}_2 &\propto \bar{T}_2 \\
\Gamma_3, \bar{T}_3, \bar{q}_3, or \bar{q}_3 &\propto \bar{T}_3 \\
\Gamma_4, \bar{T}_4, \bar{q}_4, or \bar{q}_4 &\propto \bar{T}_4
\end{align*} \]

Figure 1: BEM problem domain, boundary conditions, and single-region discretization
If a multi-region solution process were to be adopted instead, the domain is divided into \( K \) sub-domains and each one is independently discretized. Figure 2 shows the same problem depicted in Figure 1 with a multi-region BEM discretization of four (\( K = 4 \)) sub-domains. It is worth mentioning that the BEM discretizations of neighboring sub-domains do not have to be coincident, as information between neighboring sub-domains separated by an interface will be passed through an interpolation process as opposed to just a node to node connection.

![Figure 2: BEM multi-region decomposition and discretization](image)

The BEM can now be used to solve each sub-domain independently, where a guessed boundary condition is initially imposed at the interfaces in order to ensure that each sub-problem is well-posed. The resulting algebraic system for sub-domain \( \Omega_1 \) is rearranged, with the aid of given and guessed boundary conditions, as:

\[
[H_{\Omega_1}]{T_{\Omega_1}} = [G_{\Omega_1}]{\Omega_1} \Rightarrow [A_{\Omega_1}]{x_{\Omega_1}} = {b_{\Omega_1}}
\]  

(31)

The number of elements in each sub-domain, \( n \), follows the general rule that \( n \approx \frac{2N}{K+1} \), when the interface discretization level is similar to that of the boundary. The solution of the new algebraic system of sub-domain \( \Omega_1 \) will now require a number of floating point operations proportional to \( n^3/N^3 = (8/125) = 6.4\% \) of the standard BEM approach if a direct algebraic solution method is employed. And memory allocation is reduced to \( n^2/N^2 = (4/25) = 16\% \) of the original problem. The reduction in both floating point operation count and direct memory requirement is dramatic. However, as the first set of
solutions for the sub-domains were obtained using guessed boundary conditions along the interfaces, the global solution needs to follow an iteration process which decreases this operation savings, but only slightly.

Overall, the floating point count for the formation of the algebraic setup for all $K$ sub-domains must be multiplied by $K$ however, therefore the total operation count for the coefficient matrices computation is given by:

$$K \frac{n^2}{N^2} \approx \frac{4K}{(K+1)^2}$$  \hspace{1cm} (32)

For the particular case of $K = 4$, $K \frac{n^2}{N^2} = 16/25 = 64\%$ of the standard BEM approach. The more significant reduction however is revealed in the RAM memory requirements as only the memory needs for one of the sub-domains must be allocated at a time. Additionally, a direct approach to solving the algebraic equations, such as the LU decomposition method, can be employed for all sub-domains, and the LU factors of the coefficient matrices for all sub-domains can be computed only once at the first iteration step and stored in ROM memory for later use during the iteration process. Each iteration then only requires a forward and a backward substitution to arrive at a solution for the system in hand. This feature provides 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 iteration process begins by computing the initial guess temperatures (using a 1-D conduction scheme, see Divo, Kassab, and Rodriguez [5]) in the transform space and imposing these as boundary conditions at the interfaces. A resulting set of transformed normal heat fluxes, $\overline{q}$, along the interfaces are then computed. These are then non-symmetrically averaged in an effort to match the normal heat flux from neighboring sub-
domains to ensure the flux continuity condition $\bar{q}^I_{\Omega_i} = -\bar{q}^I_{\Omega_2}$ after averaging. Radial Basis interpolation can be employed in the averaging process in order to account for unstructured grids along the interface from neighboring sub-domains (a description for this procedure is given by Kassab, et al [11] for a similar interface problem, which arises in conjugate heat transfer analysis). Using these fluxes as boundary conditions at the interfaces the BEM equations are again solved, leading to mismatched temperatures along the interfaces for neighboring sub-domains. Again these temperatures values are interpolated, if necessary, from one side of the interface to the other using Radial Basis functions. Once this is accomplished, the temperature is averaged at each interface. Illustrating this for a two-domain substructure, we have for regions 1 and 2 interfaces:

\[
T^I_{\Omega_1} = \frac{T^I_{\Omega_1} + T^I_{\Omega_2}}{2} + \frac{R_w^I q^I_{\Omega_1}}{2} \\
T^I_{\Omega_2} = \frac{T^I_{\Omega_1} + T^I_{\Omega_2}}{2} + \frac{R_w^I q^I_{\Omega_2}}{2}
\]  

(33)

where $R_w^I$ is the thermal contact resistance imposing a jump on the interface temperature values to account for a case where a physical interface actually exists. These now matched temperatures along the interfaces are used as the next set of boundary conditions. The iteration process is continued until a convergence criterion is satisfied. A measure of convergence may be defined as the $L_2$ norm of mismatched transformed temperatures along all interfaces as:

\[
L^2 = \sqrt{\frac{1}{K \cdot N^I} \sum_{k=1}^{K} \sum_{i=1}^{N^I} (T^I - T^I_u)^2}
\]

(34)

This norm measures the standard deviation of BEM computed transformed interface temperatures $\bar{T}^I$ and averaged updated transformed interface temperatures $\bar{T}^I_u$. It is noted, that an iteration is referred to as the process by which a sweep is carried out to update both the interfacial fluxes and temperatures such that the above norm may be
computed. Other types of interface updating procedures are discussed by Ingber et al [6] and convergence of the procedure is detailed by Kamiya et al [12]. Also, the reader is referred to the following literature [6,13-15] for additional applications of such domain decomposition procedures in fluid mechanics and transient heat transfer.

4.2 Parallelization

Parallelization is the process of completing several tasks simultaneously using a group of computers, typically referred to as a computer cluster. The domain decomposition procedure described above lends itself quite easily to parallelization as each sub-region solution is independent of the others at each iteration. This type of parallelization will be referred to herein as spatial, as we are about to describe another method which arises in transient cases and will be referred to as temporal. Temporal parallelization is accomplished by solving a full problem at an individual time step, or in this case, Laplace Space parameter value. One additional concern which really only arises for the spatial case is that of processor load balancing. While many computer clusters may be homogeneous, it is typically impossible to obtain equally sized sub-domains for large realistic problems in the domain decomposition process. This causes imbalance in the work load sent to each processor and increases processor idle time and wastes computer resources. In the temporal case however, each temporal solution requires the same amount of effort as any other, so load balancing is much more straight-forward and can often be achieved almost perfectly. Thus for transient cases, temporal parallelization is generally more effective as communication among processors is minimized and load balancing is more exact. In the temporal scheme only the Laplace parameter and the final transformed temperatures and fluxes need to be communicated, whereas in the spatial scheme the temperatures and fluxes need to be communicated at each iteration in addition
to the final values. For the temporal case, the time savings scale almost completely linear with the number of processors used, but for the spatial case, imperfect load balancing and additional communications cause the scaling to reduce below linear. The most computational savings however can be achieved by combining both these procedures, but this is only useful when a large number of processors is available for computation, and can be quite cumbersome to distribute properly.
CHAPTER 5

NUMERICAL LAPLACE INVERSION

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, and is compared to other approaches by Davies and Martin [16]. The Stehfest transform works by computing a specified number of solutions (a sample) at certain values of the Laplace parameter, $s$, and predicting the solution based on this sample. Due to the non-oscillative behavior of typical transient heat conduction problems, the Stehfest transform works exceptionally well. The Stehfest inversion of the Laplace transform of a function of time $f(t)$ is explained thoroughly by Stehfest [8-9] and the final form is shown here for completeness:

$$f(t) = \ln\left(\frac{2}{t}\right)\sum_{n=1}^{N_s} K_n F(s_n)$$  \hspace{1cm} (35)

where the sequence of $s$-values is provided explicitly by:

$$s_n = n \frac{\ln 2}{t}$$  \hspace{1cm} (36)

And the expansion coefficients are determined by:

$$K_n = (-1)^{n+N/2} \sum_{k=(n+1)/2}^{\min(n,N/2)} \frac{k^{N/2}(2k)!}{(N/2-k)!k!(k-1)!(n-k)!(2k-n)!}$$  \hspace{1cm} (37)

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 the Laplace transformed BEM solutions. Typically, the upper limit in the series is taken as $N_s = 12 \sim 14$, as cited by Stehfest [8], however for these type of BEM solution
inversions, Moridis and Reddell [17] reported little gains in accuracy for $N_s = 6 \sim 10$, and demonstrated accurate results using $N_s = 6$. Davies and Crann [18] also report accurate results using $N_s = 8$, for BEM problems with periodic boundary conditions.

Due to the differences in transform step usage reported by various authors, and the difference in technique posed by the domain decomposition approach used herein, a detailed study has been performed in an attempt to determine the optimal number of transform steps. Table 1 below shows the BEM solution times and deviations from the exact solution for the range $N_s = 4 \sim 16$, on a rectangular bar (described in the later verification section). Multiple time values along with one, two, and four region cases have been considered, providing a range of situations for the optimization (averages are presented here). Taking into account both computational time and accuracy, it is decided that $N_s = 8$ is the best choice, as increasing to $N_s = 10$ offers little accuracy increase, but increases the computational burden by about 25%, from the two additional BEM solutions. It is also noticeable from the results below that $N_s = 4$ or $N_s = 6$ both offer acceptable solutions and should produce solutions within practical limits for many situations, especially where trends are more important than actual quantitative values.

Table 1. Stehfest Transform sample size comparison
(Computed using a single P3, dual processor, 700 MHz machine, $N = 160$)

<table>
<thead>
<tr>
<th>Stehfest Number ($N_s$)</th>
<th>Average % Deviation</th>
<th>Average Run Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>6.29</td>
<td>32.51</td>
</tr>
<tr>
<td>6</td>
<td>0.470</td>
<td>45.40</td>
</tr>
<tr>
<td>8</td>
<td>0.0611</td>
<td>57.83</td>
</tr>
<tr>
<td>10</td>
<td>0.0511</td>
<td>74.97</td>
</tr>
<tr>
<td>12</td>
<td>2.753</td>
<td>89.30</td>
</tr>
<tr>
<td>14</td>
<td>177.35</td>
<td>99.43</td>
</tr>
<tr>
<td>16</td>
<td>6354.80</td>
<td>111.96</td>
</tr>
</tbody>
</table>
It is also notable that due to amplification effects of the large factorial coefficients present in the Stehfest Transform, on both round-off and truncation errors, BEM solutions must be carried to very high levels of precision, which is especially true for large values of \( N_s \). This relationship between truncation error and error amplification is shown graphically in Figure 3. For this reason very accurate integration, linear solvers, 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 computational burden 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. Trends over time can be effectively found by performing multiple runs of this routine using time increments many orders of magnitude larger than those necessary for finite difference solutions. Computational savings can also be realized by analyzing the Laplace transform parameter values required for inversion by the Stehfest transform. It is seen that each time solution requires \( N_s \) BEM solutions in Laplace space, which are found at integer multiples of the Laplace transform parameter, \( s \). This shows that if the first desired time solution is chosen and the remaining are integer multiples of this time, numerous overlapping solutions will occur. The required Laplace transform solutions for selected times can be computed up-front and solutions at the unique Laplace transform parameter values can then be calculated and stored, where a simple indexing can be used to combine the correct solutions within the inversion routine upon completion of all required solutions. Significant computational savings are found and memory requirements are small as only the temperatures and fluxes (not the coefficient matrices) for each Laplace space solution must be saved. An illustration of this procedure is given in Table ** for a simple example problem where the required number of solutions is
reduced from 49 to 40 by adjusting the values slightly from those set by the user. Accurate trends are reported using this method and since heat diffusion generally does not involve oscillations, time interpolation is rather straightforward and quite accurate. This numerical inversion marks the last step of the overall solution process, and now the procedure is in a position to be tested and analyzed for accuracy and efficiency, which is the focus of the next two chapters.

Figure 3: Graphical representation of Stehfest Transform errors

Table 2: s-value comparison for two time value sets
(highlighted terms are repeated and don't need computation)

(a) User defined times, 49 total solutions required

<table>
<thead>
<tr>
<th>time value</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>8.99</td>
<td>13.86</td>
<td>20.79</td>
<td>27.73</td>
<td>34.66</td>
<td>41.59</td>
<td>48.52</td>
<td>55.45</td>
</tr>
<tr>
<td>0.2</td>
<td>3.47</td>
<td>5.93</td>
<td>10.40</td>
<td>15.86</td>
<td>17.33</td>
<td>20.79</td>
<td>24.26</td>
<td>27.73</td>
</tr>
<tr>
<td>0.5</td>
<td>1.98</td>
<td>2.77</td>
<td>4.16</td>
<td>5.55</td>
<td>5.91</td>
<td>5.32</td>
<td>9.70</td>
<td>11.09</td>
</tr>
<tr>
<td>2</td>
<td>0.95</td>
<td>0.69</td>
<td>1.04</td>
<td>1.39</td>
<td>1.75</td>
<td>2.08</td>
<td>2.43</td>
<td>2.77</td>
</tr>
<tr>
<td>10</td>
<td>0.17</td>
<td>0.14</td>
<td>0.21</td>
<td>0.28</td>
<td>0.36</td>
<td>0.42</td>
<td>0.48</td>
<td>0.55</td>
</tr>
<tr>
<td>15</td>
<td>0.06</td>
<td>0.09</td>
<td>0.14</td>
<td>0.18</td>
<td>0.22</td>
<td>0.26</td>
<td>0.30</td>
<td>0.37</td>
</tr>
<tr>
<td>20</td>
<td>0.05</td>
<td>0.07</td>
<td>0.10</td>
<td>0.14</td>
<td>0.17</td>
<td>0.21</td>
<td>0.24</td>
<td>0.28</td>
</tr>
<tr>
<td>25</td>
<td>0.03</td>
<td>0.06</td>
<td>0.09</td>
<td>0.11</td>
<td>0.14</td>
<td>0.17</td>
<td>0.19</td>
<td>0.22</td>
</tr>
</tbody>
</table>

(b) Shifted times, 40 total solutions required

<table>
<thead>
<tr>
<th>time value</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>8.39</td>
<td>13.86</td>
<td>20.79</td>
<td>27.73</td>
<td>34.66</td>
<td>41.59</td>
<td>48.52</td>
<td>55.45</td>
</tr>
<tr>
<td>0.2</td>
<td>3.47</td>
<td>5.93</td>
<td>10.40</td>
<td>15.86</td>
<td>17.33</td>
<td>20.79</td>
<td>24.26</td>
<td>27.73</td>
</tr>
<tr>
<td>0.5</td>
<td>1.98</td>
<td>2.77</td>
<td>4.16</td>
<td>5.55</td>
<td>5.91</td>
<td>5.32</td>
<td>9.70</td>
<td>11.09</td>
</tr>
<tr>
<td>2</td>
<td>0.95</td>
<td>0.69</td>
<td>1.04</td>
<td>1.39</td>
<td>1.75</td>
<td>2.08</td>
<td>2.43</td>
<td>2.77</td>
</tr>
<tr>
<td>10</td>
<td>0.17</td>
<td>0.14</td>
<td>0.21</td>
<td>0.28</td>
<td>0.36</td>
<td>0.42</td>
<td>0.48</td>
<td>0.55</td>
</tr>
<tr>
<td>15</td>
<td>0.06</td>
<td>0.09</td>
<td>0.14</td>
<td>0.18</td>
<td>0.22</td>
<td>0.26</td>
<td>0.30</td>
<td>0.37</td>
</tr>
<tr>
<td>20</td>
<td>0.05</td>
<td>0.07</td>
<td>0.10</td>
<td>0.14</td>
<td>0.17</td>
<td>0.21</td>
<td>0.24</td>
<td>0.28</td>
</tr>
<tr>
<td>25</td>
<td>0.03</td>
<td>0.06</td>
<td>0.09</td>
<td>0.11</td>
<td>0.14</td>
<td>0.17</td>
<td>0.19</td>
<td>0.22</td>
</tr>
</tbody>
</table>
CHAPTER 6
VERIFICATION EXAMPLES

This chapter details several example problems which are used as verification of the transient BEM solutions, in both two and three dimensions. The first case is a two dimensional problem which consists of an $L = 4m$ by $l = 1m$ rectangular region imposed with $T = 0$ on the bottom and left walls and with $q = -1$ on the top and right walls as depicted in Figure 4. The domain is initially at a temperature $T(x, y, 0) = 0$. The selected material properties are: $\rho = 1000$, $c = 2$, and $k = 1$, all in standard SI units. 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 found for the problem at hand using superposition and separation of variables. The analytical solution followed by the BEM solutions for a varying number of regions are displayed in Figure 5 in the form of contour plots. The temperature contours are shown at various times ($t = 25s$, $t = 100s$, $t = 200s$, $t = 500s$) and excellent agreement is found between the exact and BEM solutions in
both single and multi-region cases. The error between the exact and BEM solutions has been quantified using an $L_2$ norm (computed with 32 interior points) and is plotted at several time levels in Figure 6a below. In addition, a plot of the temperature evolution and deviation from $t = 0$ to $t = 1000s$ is shown in Figures 6c and 6d for the single point $(x, y) = (3.8, 0.8)$, revealing virtually perfect agreement between exact and BEM solutions. These plots clearly show that the high level of accuracy that is typical of BEM is maintained while using the numerical inversion and multi-region implementation, and as hoped, the domain decomposition procedure has not affected the accuracy of the results in any appreciable manner. This last statement is especially evident as we see that the difference in temperature error norms between the single and multi-region cases (shown in Figure 6b below) is about 5 orders of magnitude less than the value of the error norm itself (shown in Figure 6a below).
Figure 5: Contour plots of the temperature solution for the transient verification problem
a) BEM temperature error norms for 32 interior points at various time levels

b) Difference in temperature error norms between single region and multi-region BEM
c) Single point temperature over time

d) Single point temperature deviation over time

e) Interior point distribution for a and b, and single interior point location for c and d

Figure 6: Multi-region BEM temperature accuracy comparisons
The next case is a simple three dimensional problem in a 1m cube. The boundary conditions are \( T(0, y, z) = 0 \) and \( q(1, y, z) = -100 \) and insulated on all other faces. The exact solution is 1-D in the \( x \) direction and is easily calculated analytically. The figures below clearly show the accuracy of the technique for this simplistic 3-D problem. Figure 7 shows a brief demonstration of grid convergence as we see the error in the BEM solutions decrease as the grid spacing is reduced. A grid of 5 elements across (25 per face) is shown to have a maximum error of about \( 0.1^\circ C \), which is within the range of acceptability for almost all engineering applications. This 5x5x5 grid has thus been used for the remainder of this problem. Upon completion of this grid convergence check, a brief study of the affects of the number of Stehfest inversion steps was completed for this 3-D case. Again an optimal value of \( N_s = 8 \) was found (as discussed in Chapter 5 above) with these findings being displayed in Figure 8. Figure 9 shows the temperature history throughout time for three points within the domain as compared to the exact solutions at these locations (note these results are practically independent of the \( y \) and \( z \) coordinates as seen in Figure 10). And finally Figure 10 shows the entire temperature field at several instances in time. It is clearly visible in these plots that the solution remains 1-D (as expected) throughout the entire time of interest and a rise in temperature is seen at all points until finally reaching the steady-state, linear, solution around a time of \( t = 5000 \) sec.
Figure 7: Temperature and Error in Temperature @ $t = 250 \text{ sec}$ for several levels of grid refinement for 3-D cube
Figure 8: Temperature $t = 250$ sec for several values inversion steps $N_s$, for 3-D cube

Figure 9: Temperature throughout time at several positions for 3-D cube
a) Temperature fields $@ t = 100\, sec$ and $t = 500\, sec$

b) Temperature fields $@ t = 1000\, sec$ and $t = 2000\, sec$
c) Temperature field @ $t = 5000$ sec and colorbar for all cases
Figure 10: Contour plots of the temperature field at various times for 3-D cube
CHAPTER 7

PRACTICAL PROBLEMS

The following chapter shows several practical example problems which have been solved using the methodology developed within this thesis. Two 2-D example problems are presented and the results are compared to the commercial FVM software, Fluent 6.1 to give further validation to this methodology. A 3-D example is then presented for a large turbine blade to demonstrate the practical applications of this work in areas such as gas turbine engines.

In this first example a laminar airfoil with 3 cooling passages was designed to test the accuracy findings on a larger, more complex scale. The entire model consisted of about 1600 degrees of freedom, which were split into eight separate sub-domains. Constant convective boundary conditions were applied at all surfaces. The problem was modeled in Fluent 6.1 using an appropriate level of discretization, see Figure 11 below for each mesh. Since an analytical solution to this problem is not available, a time step convergence study was completed to ensure stable, accurate results for the finite difference analysis. At a time step of 0.04 seconds the change in solutions become negligible and the problem was found as time step converged. The temperature solutions at two points in each model were recorded and are displayed over time in Figure 12 below. Contour plots at a single time are also presented to show the agreement of the entire temperature field. These results show almost perfect agreement between the BEM and FVM solutions.
a) BEM Mesh

b) FVM Mesh

Figure 11: Laminar airfoil meshes (convective BC's imposed on all boundaries)

![Graph showing airfoil temperatures throughout time](image)

**Airfoil Temperatures Throughout Time**

- Fluent 6.1 Lead Edge
- BEM Lead Edge
- Fluent 6.1 Trail Edge
- BEM Trail Edge

a) Temperature between leading edge and first passage region and in trailing edge region
b) BEM temperature field at $t = 40 \text{ sec}$

c) Finite volume temperature field at $t = 40 \text{ sec}$

d) Temperature scale for both solution fields

Figure 12: BEM comparison to finite volume solver (Fluent 6.1)

The next example shows a transient heat transfer analysis performed on a non-symmetric airfoil under the estimated turbine conditions given below. The geometry is shown in Figure 13 along with the applied convective boundary conditions, noting that all temperatures are measured above ambient. The exterior 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 leading edge. The cooling passages use a coolant flow at 1800 degrees and a constant heat transfer coefficient. As this is only a test of the abilities of the conduction solver, all values have been assumed and are not representative of actual turbine conditions. The results of the analysis show the smooth, rapid propagation of heat through the airfoil over time in Figure 14. The temperature at two points, one near the leading edge and one near the trailing edge, are tracked and plotted over time in Figure 15, for both the FVM and BEM solutions. The results again show nearly perfect agreement between the two solution methods.
Figure 13: BEM discretization and BC’s for non-symmetric airfoil with cooling passages

Figure 14: Airfoil leading and trailing edge region temperatures over time
The final example problem is a 3-D turbine blade, again with two cooling passages. There are convective boundary conditions imposed on all exterior and interior surfaces and insulated conditions on the two outside flat edges. The problem is split into 6 unequal regions for implementation of the domain decomposition procedure described above. The temperature field solutions are shown for the blade at several instances of time, including a long time which approaches the steady-state solution in Figure 16. The smoothness of the temperature field is quite apparent and the results at large times compare well to steady-state code results, which instill further confidence in these values.
a) Temperature fields @ $t = 10\ sec$ and $t = 20\ sec$

b) Temperature fields @ $t = 40\ sec$ and $t = 250\ sec$ (steady-state)

c) Temperature colorbar for all cases

Figure 16: Contour plots of the temperature field at various times for 3-D turbine blade
This chapter has presented solutions to several practical example problems along with comparisons to proven methods of solution. These results provide good confidence that the Laplace transform BEM procedure can produce accurate results over a range of problem geometries and time levels, in both two and three dimensions. Since much of the work of this thesis is focused on application to large problems it is important to now discuss the computational times of some of these examples. For the 3-D turbine blade, six regions were used to break the problem into more manageable pieces. The total number of nodes used was just over 27,000, which if solved using a single region would require the storage of about 800,000,000 values for the influence coefficient matrices. However, using the six region model above the memory required was reduced to less than 6,000,000 values, or over 125 times less memory needed. Additionally, the problem was run using a temporal parallelization scheme using 20 nodes of a computer cluster. The use of this parallelization brought the total solution time from about 20 hours to just over 1 hour, a nearly 20 fold savings (note that the temporal parallelization scheme scales just about linearly with the number of processors used). These savings are quite dramatic and may help make the BEM an attractive alternative to traditional FVM and FEM solvers, and the next chapter will describe a POD interpolation process which can be used to achieve even further computational savings.
CHAPTER 8

PROPER ORTHOGONAL DECOMPOSITION

The aim of the Proper Orthogonal Decomposition, POD, approach is to increase the efficiency of Laplace Transform solution schemes by employing an interpolation process within the inversion phase. This is done by reducing the number of BEM field solutions required during the Stehfest Inversion process. It is worth pointing out that the majority of the computational time is spent computing the BEM field solutions, so any reduction in the number of BEM solutions required will equate to huge computational savings. As described above, every time domain solution requires the calculation of 8 individual BEM solutions in Laplace space in order to complete the inversion process accurately (recall \( N_a = 8 \) is the optimal number of Stehfest transform steps discussed above). This makes the Laplace transform method extremely slow if many time domain solutions are required. The advantage over time-marching is still appreciable, as we are free to choose any value of time for our solutions. Also the method of "time parallelization" detailed above is still applicable within the POD approach. Since the intent of the POD approach is to reduce the number of BEM solutions required while maintaining the accuracy of the standard approach, two schemes were developed: (1) apply the POD interpolation in the Laplace Space, before the inversion process and (2) apply the POD interpolation in the time domain after the inversion process. Both these methods are detailed in the first two sections below, and some numerical results and verifications are shown in the last section of the chapter to compare the two approaches, all of which precluded by a brief description of the POD method.

The Proper Orthogonal Decomposition was developed over 100 years ago and has received much attention in engineering literature. The POD has been used in the areas of
fluid mechanics, heat transfer, dynamics, and more recently inverse problems [20-23], and the technique is well described in many of these publications and as such is only described briefly here for clarity. The first step to implementing the POD is to obtain a set of so-called "snapshots". These snapshots are representations of the field of interest at some values of the independent or unknown parameters. In this application, a snapshot is a vector containing the temperatures, at a predefined set of points, within the body of interest for a single value of time, or Laplace space parameter. Several snapshots are then combined to form a rectangular matrix, which is the foundation of the POD technique. The POD is used as an efficient method of interpolation, as truncation can be used. The POD finds the most appropriate snapshots to use within the interpolation process, and truncates the unnecessary ones. In this formulation the interpolating (basis) functions are the simple Radial Basis functions (RBF):

$$\phi(Z) = 1 + |Z - Z_i|$$

(38)

Where $Z$ is taken as either the time value, $t$, or the Laplace parameter, $s$, depending on which of the two approaches described below is being implemented.

### 8.1 Laplace Space Approach

In this approach the solutions at various values of the Laplace transform parameter, $s$, are found up front, within the range of those needed for the specified time interval. The appropriate range for $s$ is easily found using the maximum and minimum time values requested as: $s_{min} = \frac{ln 2}{t_{max}}$ and $s_{max} = 8 \frac{ln 2}{t_{min}}$. The specific number of solutions used can be chosen by the user and varied for the specific problem and time interval at hand. It is hoped that few solutions are required to accurately represent the entire Laplace space solution. Since the Stehfest Transform requires solutions at specific values of $s$, these
solutions are generated using the POD interpolation with the RBF basis functions described above. Although it is hopeful that only a few solutions are needed, it is noted that an accurate interpolation is key, as error amplification occurs during the inversion process, due to the large coefficients in the series expansion of the Stehfest Transform. As a simple example, suppose the time of interest for a particular problem is from $t = 0$ to $t = 100$ seconds, and that the first time value needed will be at $t = 0.1$ seconds. The maximum and minimum values of $s$ are computed, and then an equal spacing discretization is applied over this range of $s$ values, using a set number of solution points, $np$. The BEM solution is then carried out for these $np$ values of $s$. Then the POD interpolation procedure described above is carried out to find the temperature at each point for each value of $s$ needed by the Stehfest Transform. Suppose 50 time solutions were generated using this procedure, this would typically require $50 \times N_s = 50 \times 8 = 400$ Laplace space BEM solutions using a standard approach, but this technique was able to reduce this to only $np$ required BEM solutions (where $np$ should be much smaller than this value, on the order of 50 for this case).

### 8.2 Time Domain Approach

The time domain approach is an attempt to gain efficiency, but avoid the error amplification effects of interpolating before inversion. More accurate results are expected since the interpolation is directly over the desired parameter, time, but efficiency gains will likely be decreased. This approach uses the following simple formula for determining the most efficient time values to use for interpolation:

$$t_i = 2^i t_{min} \quad \text{for } i = 1, 2...n$$  \hspace{1cm} (39)
where $t_{\text{min}}$ is the smallest given time value and $n$ is found as the smallest positive integer satisfying, $t_{\text{max}} < 2^n t_{\text{min}}$, with $t_{\text{max}}$ as the maximum time of interest. This ensures that the values of time used will be clustered toward smaller time values, where faster transients typically occur, and will be such that the required Laplace space solutions for inversion will overlap as much as possible. Again using the simple example described above, the value of $n = 11$ is found. Using a standard time discretization will require $nN_s = 11 \times 8 = 88$ Laplace space solutions, however using the discretization scheme shown above, the required number of solutions is reduced to only 36 due to the overlapping of the required Laplace space solutions. For a general case the required number of solutions can be calculated as: $3n + 3$, for this scheme. This type of discretization may not be optimal for high accuracy however, as the spacing between solutions becomes quite large at higher values of time, so for some cases a different discretization must be used. As a general rule, an efficient time discretization can reduce the number of required solutions by about 50%.

### 8.3 Numerical Study of POD Approaches

The first case used to test both the methods described above was the simple and classic case of one-dimensional heat conduction in a bar. The problem geometry along with the initial and boundary conditions are shown in Figure 17.

![Figure 17: One-Dimensional transient verification example problem](image-url)
The results at two instances of time are shown below for both POD methods (Laplace and time domain), along with the analytical solution in order to compare the accuracy of each method. These results were generated for the problem described above using 36 total Laplace space solutions for each of the two POD approaches, so that accuracy for the same computational effort could be visualized, see Figures 18-19. It is clear from the results that the time domain interpolation scheme is more accurate, even for the same level of computational effort. These errors are expected as any error in interpolation for the Laplace domain approach is greatly amplified during the inversion process, as described above, however the errors were not expected to be present for similar levels of computational effort.

Figure 19 shows results throughout time at a single point located at the center ($x = 0.1m$) of the bar. It is quite noticeable that the Laplace space scheme does not provide satisfactory results, as many points have unacceptably large errors. The gains in efficiency that were hopeful for this method were also not realized, as a very large number of Laplace space solutions were needed to get the results to the level shown in Figure 19. This scheme is able to provide accurate results if a very large number of points are used, but this actually causes an increase in computational requirements; this requirement is due to the level of precision needed for accurate numerical inversion. The results for the time domain scheme do provide accurate results however, and have yielded some reasonable gains in efficiency, which are discussed further in the next example below.
Figure 18: Results at various time levels for 1-D conduction problem
Since the above example was used to determine the feasibility and accuracy of the two proposed schemes, this example will be focused on the efficiency gains for the time domain interpolation approach (the Laplace space interpolation approach will not be used as it was proven not to be useful). This example will use the non-symmetric airfoil found previously in Chapter 7. The time of interest will be from startup ($t = 0$ sec) to warm up (taken here as $t = 10$ sec). Using the standard approach and gathering results at every $\frac{1}{2}$ second would require computing 160 Laplace space BEM solutions. This took a total run-time of 53 min on a single Pentium 4, 3.0 GHz machine. Employing the time domain POD interpolation scheme, the number of required solutions was reduced to 64, dropping the run-time to 21 min on the same machine, a reduction of about 60%. The standard approach results are known to be quite accurate from previous work [19], and since the results for the more efficient POD time interpolation are in good agreement, as shown in Figure 20, we have confidence that this method works well.
Figure 20: Non-symmetric airfoil problem temperatures throughout time at two discrete locations (one near the trailing edge and another near a cooling passage)
CHAPTER 9

CONCLUSIONS AND EXTENSIONS

This thesis has developed a Laplace transform BEM algorithm for the efficient solution of large-scale transient heat conduction problems. The goals for the application of this work to large-scale problems have been accomplished by using several techniques, including domain decomposition, parallel implementation, and the proper orthogonal decomposition. The focus of this work has been specific to heat conduction, but many of the techniques are general and can be applied to other areas of interest in engineering. The results of this work are quite promising and have given reason to pursue further development of the method and possibly the implementation of a user friendly BEM heat transfer package.

The BEM formulation for the solution of the Modified Helmholtz equation was detailed as the first step of this work and then several of the efficiency improvement schemes employed by this solver were discussed along with their effects on accuracy and performance. The accuracy of the combination of these methods was then proven through comparisons to analytical solutions as well as commercial software. Additionally, the practical applications of this work were demonstrated through example heat conduction problems in both two and three dimensions, which also showed the usefulness of this work in areas such as turbine development and design.

The extension of this work to 3-D and the inclusion of time parallelization and POD interpolation has established the effectiveness and accuracy of these techniques for realistic, large-scale problems. The advantage of the elimination of time dependence in
the governing equation, using the Laplace transformation, allowed the extension to 3-D in a very straight-forward manner, as only slight modifications to a steady-state 3-D code and the inclusion of the inversion routine were necessary for implementation. One future path for this work may be the development of an automated domain decomposition routine. Current domain decomposition is performed manually with interfaces being part of the user input, therefore this automation procedure would have profound effects, as the creation of boundary meshes would become much less complex. Certain rules must be followed for the domain decomposition techniques to be effective and care must be taken during this process. This increases setup time and takes away some of the allure of the BEM. The automation of this process would eliminate this additional setup time and restore the simplicity of generating a BEM mesh.

Further work is also necessary for the fine tuning of the POD interpolation scheme for the most possible efficiency gains. Additional equations may be tested for interpolation as well as switching to a standard interpolation scheme, as the performance gains of using a truncated POD are quite small when compared to the overall computational effort of the problem solution.
REFERENCES


