

Nonlinear, Transient Conduction Heat Transfer Using A Discontinuous Galerkin Hierarchical Finite Element Method

by

Jerome Charles Sanders
B.S. in Physics, May 2002
The College of New Jersey

A Thesis submitted to
The faculty of
The School of Engineering and Applied Science of
The George Washington University
in partial satisfaction of the requirements for the degree of
Master of Science

August 31, 2004

This research was conducted at NASA's Langley Research Center

Abstract

Hypersonic and reentry vehicles are exposed to high temperatures and large temperature gradients as they travel through the atmosphere. Previous launch vehicles relied on thermal protection systems (TPS) to handle thermal loads while the structure supported aerodynamic loads. Future launch vehicle designs will likely use non-insulated hot structures and multifunctional structures that sustain both thermal and aerodynamic loads simultaneously. Finite element methods are needed to predict accurately the thermal and structural response of these types of structures.

Due to the large temperature ranges encountered by hypersonic and reentry vehicles, the thermal conductivity of the material must be considered temperature-dependent. Current methods used to account for this temperature-dependence assume thermal conductivity is constant over each element or alternatively, the finite element integrals containing the thermal conductivity are integrated using Gaussian quadrature. These methods require a large number of elements or an excessive number of integration points, respectively, to achieve satisfactory accuracy.

A hierarchical p-version space-time finite element method with a structurally compatible mesh for nonlinear, transient problems is presented in this research. A hierarchical finite element scheme with structurally compatible elements is used in space. A discontinuous Galerkin time stepping scheme is used as the finite element method in time. An optimized interpolation of the temperature-dependent thermal conductivity eliminates the need for a large number of elements or an excessive number of integration points and provides increased computational efficiency by utilizing master matrices. The finite element formulation is tested and validated using sample problems. This finite element formulation is shown to have the same theoretical convergence rates as those expected for linear problems when a single source of error in the approximation can be isolated.

Acknowledgements

I would like to thank Dr. Kim Bey of the Metals and Thermal Structures Branch at NASA's Langley Research Center. Her patience, knowledge, and guidance were invaluable to my research. I would also like to thank Dr. Paul Cooper from the George Washington University for his unique perspective and for his helpful insights into my work. My thanks also go to Dr. Stephen Scotti for allowing me the opportunity to work in the Metals and Thermal Structures Branch. Finally, I would like to thank my family for their encouragement and support.

Contents

Abstract	i
Acknowledgements	ii
Nomenclature	xi
1 Introduction	1
1.1 Review of Previous Work	4
1.2 Purpose	5
1.3 Scope	6
2 Finite Element Method	7
2.1 Initial Boundary Value Problem	7
2.2 Weak Formulation	9
2.3 Finite Element Formulation	11
2.3.1 Discontinuous Galerkin Method in Time	14
2.3.2 Approximation of Solution	15
2.3.3 Temperature Approximation	17
2.3.4 Temperature-Dependent Thermal Conductivity	20
2.4 Element Matrices	21
2.5 Time-Stepping Solution Method	24
2.6 Applying Initial and Boundary Conditions	26

3	Basis Functions	29
3.1	Time Basis Functions	29
3.2	Through-Thickness Basis Functions	31
3.3	Hierarchical In-Plane Basis Functions	35
4	Interpolation of Thermal Conductivity	47
4.1	Lagrange Interpolation Functions	48
4.1.1	Time Interpolation Functions	48
4.1.2	Through-Thickness Interpolation Functions	50
4.1.3	In-Plane Interpolation Functions	50
4.2	Optimization of Sampling Points	54
4.3	Determination of Interpolation Coefficients	55
4.4	Convergence Criteria for Iteration	58
5	Computer Implementation	60
5.1	Integration	62
5.2	Master Element Matrices	63
5.3	Enforcing Essential Boundary Conditions	69
6	Numerical Results	72
6.1	Error Convergence Estimates	72
6.2	Error Due to Fixed-Point Iteration	76
6.3	Sample Problems	77
6.3.1	Example 1	79
6.3.2	Example 2	83
6.3.3	Example 3	87
6.3.4	Example 4	92

6.3.5	Example 5	97
6.3.6	Example 6	100
7	Concluding Remarks	105
	References	108

List of Figures

2.1	Three-dimensional spatial domain, Ω , with Dirichlet and Neumann boundary conditions	8
2.2	Space-time mesh	12
2.3	Discontinuous Solution in Time	15
2.4	Arbitrary Two-Dimensional Spatial Domain with a Uniform Thickness	17
2.5	Time-Stepping Solution Flowchart	25
3.1	Three-Dimensional Domain Collapsed onto a Two-Dimensional Mesh	32
3.2	One-dimensional hierarchical element	32
3.3	Local Basis Functions for $p_{tk} = 5$	33
3.4	Triangular Hierarchical Element Node Numbering Convention	35
3.5	Global Coordinates and Areas for a Triangle	36
3.6	Triangular Elements Sharing a Side	40
4.1	Sample Time Lagrange Element	49
4.2	In-Plane Lagrange Nodal Positions for $r_{ip} = 3$ (left) and $r_{ip} = 5$ (right)	52
4.3	Fekete Points for In-Plane Degrees of Interpolation $r_{ip} = 3, 6, 9$	55
5.1	Operational Order of ndgSCHTp	61
5.2	Extraction of Sub-Matrix from Master Time Mass Matrix	65
5.3	Extraction of Element Basis Functions from Master In-Plane Basis	66

5.4	Extraction of Sub-Matrix from Master In-Plane Matrix	67
5.5	Sample Element with Varying Degrees of Through-Thickness Approximation	68
5.6	Three-Dimensional Domain with Surface and Edge Boundary Conditions	69
6.1	Error Convergence Estimates on a Log-Log Scale	
	(a) H^1 Error vs. Element Size	
	(b) L^2 Error vs. Number of Time Steps	76
6.2	2, 8, 16, and 32 Element In-plane Meshes	78
6.3	Exact Solution and Finite Element Error for Example 1 at $t = 1s$	
	(a) Exact solution	
	(b) Point-wise error with $p_{ip} = r_{ip} = 5$ and $p_t = r_t = 1$	80
6.4	Convergence of the Error for Example 1	
	(a) H^1 Error for Different Values of p_{ip}	
	(b) L^2 Error for Different Values of p_{ip}	
	(c) H^1 Error for Different Values of r_{ip}	
	(d) L^2 Error for Different Values of r_{ip}	81
6.5	Exact Solution and Finite Element Error for Example 2 at $t = 2s$	
	(a) Exact solution	
	(b) Point-wise error with $p_{ip} = r_{ip} = 4$ and $p_t = r_t = 6$ using one time step	84
6.6	Example 2: H^1 Error as a Function of Time for $p_t = 1$ and $p_t = 4$	85
6.7	Convergence of the L^2 Error for Example 2	
	(a) The Effect of Different Values of p_t	
	(b) The Effect of Different Values of r_t	86
6.8	Exact Solution and Finite Element Error for Example 3 at $t = 10s$	
	(a) Exact solution	
	(b) Point-wise error with $p_{ip} = r_{ip} = 4$, $p_t = r_t = 1$, and $p_{tk} = r_{tk} = 6$. . .	89

6.9	Convergence of the Error for Example 3	
	(a) H^1 Error for Different Values of p_{ip}	
	(b) L^2 Error for Different Values of p_{ip}	
	(c) H^1 Error for Different Values of p_{tk}	
	(d) L^2 Error for Different Values of p_{tk}	91
6.10	Exact Solution and Finite Element Error for Example 4 at $t = 1s$	
	(a) Exact solution	
	(b) Point-wise error with $p_{ip} = 3$, $r_{ip} = 6$, $p_t = 2$ and $r_t = 4$	93
6.11	Convergence of the Error for Example 4	
	(a) H^1 Error for Different Values of p_{ip}	
	(b) L^2 Error for Different Values of p_{ip}	
	(c) H^1 Error for Different Values of r_{ip}	
	(d) L^2 Error for Different Values of r_{ip}	
	(e) H^1 Error for the Tabulation of Thermal Conductivity	
	(f) L^2 Error for the Tabulation of Thermal Conductivity	95
6.12	Exact Solution and Finite Element Error for Example 5 at $t = 1s$	
	(a) Exact solution	
	(b) Point-wise error with $p_{ip} = r_{ip} = 5$ and $p_t = r_t = 6$ using five time steps	98
6.13	Convergence of the L^2 Error for Example 5	
	(a) The Effect of Different Values of p_t	
	(b) The Effect of Different Values of r_t	99
6.14	Exact Solution and Finite Element Error for Example 6 at $t = 1s$	
	(a) Exact solution	
	(b) Point-wise error using a 16 element mesh and 5 time steps with $p_{ip} =$ $r_{ip} = 6$ and $p_t = r_t = 4$	101

6.15 Convergence of the Error for Example 6

- (a) H^1 Error as a Function of Time
- (b) L^2 Error for Different Values of p_t
- (c) H^1 Error for Different Values of p_{ip}
- (d) L^2 Error for Different Values of p_{ip} 102

List of Tables

3.1	Index Permutation for Edge Basis Functions	39
3.2	Pascal's Triangle for Formulation of Interior Basis Functions	42
4.1	Pascal's Triangle for the Number of In-Plane Interpolation Functions . . .	51
6.1	Number and Size of Elements Used to Analyze Example Problems	78

Nomenclature

<u>Symbol</u>	<u>Definition</u>
a	temperature approximation coefficient
$\mathbf{a}, \{\mathbf{a}\}$	column vector of temperature approximation coefficients
b	interpolation coefficient
\mathbf{b}	column vector of interpolation constants
c	y intercept
c_p	specific heat
$\{c\}, \{d\}$	sample column vectors
d	uniform thickness of the spatial domain
$e(\tilde{x}, t)$	error in the approximation at a point
$g(x, y, z)$	given initial condition
$g(x, y, z)_e$	initial condition over a spatial element
h	maximum distance between any two points on a triangular element (element size)
$k(u)$	isotropic temperature-dependent thermal conductivity
\hat{k}	approximate thermal conductivity
$\{\hat{k}\}$	column vector of the conductivity values at \tilde{N} sampling points
m	slope

n_t	number of time steps
$\hat{\mathbf{n}}$	unit vector normal to the boundary
p	degree of the temperature approximation
\bar{q}	temperature-dependent thermal conductivity multiplied by the gradient of the temperature, $-k(u)\nabla u$
q_s	prescribed heat flux along Γ_N
r	degree of the conductivity interpolation
$\{r\}$	initial condition source vector
t	global time coordinate
Δt	length of a time step
u	exact scalar temperature function
u_D	prescribed temperature on Γ_D
$[u_n]$	difference in the solution at a discontinuity
\hat{u}	finite element temperature approximation
v	scalar test function
$\{w\}$	initial condition coefficients for the spatial basis functions
x, y, z	global cartesian coordinates
A	in-plane element area
A_1, A_2, A_3	three triangular areas opposite the corresponding vertex
A_i	local element area
$[A], [B]$	sample square matrices
C	value of the convergence criterion
$[C]$	capacitance matrix
D_e	space-time element

$\{F\}$	vector created from the assembly of the load vector and the mass matrix multiplied by the unknowns
$\{H\}$	load vector
I	time interval
$[IC]$	initial condition matrix
$[J]_m$	2×2 Jacobian matrix
J	Jacobian
$[K]$	stiffness matrix
$[L]$	load vector
L_1, L_2, L_3	area coordinates for an arbitrary triangle
$[M]$	mass matrix
N	number of basis functions for the temperature approximation
\tilde{N}	number of conductivity interpolation functions, number of sampling points
$[P]$	square matrix of length \tilde{N}_b such that each row is the vector of interpolation functions evaluated at a single sampling point
P	Legendre polynomial
Q	scalar internal heat source function
$[S]$	global matrix formed from the assembled sum of the capacitance, stiffness, and mass matrices
T	end of the global time interval for the problem
$[V]$	convective matrix

α, β	local coordinates for the interior basis functions
η	local coordinate for through-thickness basis
θ	column vector of N_t time basis functions
$\tilde{\theta}$	column vector of \tilde{N}_t time interpolation functions
ξ	local in-plane coordinate for an element
ρ	density
τ	local time coordinate
ϕ	column vector of N_{ip} in-plane basis functions
$\tilde{\phi}$	column vector of \tilde{N}_{ip} in-plane interpolation functions
χ	column vector of N_b element basis functions
$\tilde{\chi}$	column vector of \tilde{N}_b interpolation functions
ψ	column vector of N_{tk} through-thickness basis functions
$\tilde{\psi}$	column vector of \tilde{N}_{tk} through-thickness interpolation functions
Γ	total boundary of the global domain
Γ_D, Γ_N	Dirichlet and Neumann boundaries
Υ_j	j^{th} global space-time basis function
Ω	spatial domain

Norms

$\| \cdot \|_{H^1(\Omega)}$

$\| \cdot \|_{L^2(I_n, L^2(\Omega))}$

$\| \cdot \|_{L^2(I, H^1(\Omega))}$

Definition

H^1 norm in space at a point in time

L^2 norm over a single time step with an L^2 norm in space

L^2 norm over I with an H^1 norm in space

Subscripts

0	first time step
b	total number on an element
e	element
f	finite number of terms
i, j, k, l, m	indices
ip	in-plane
n	n^{th} time step
t	time
tk	through-thickness
$[]_e$	element matrix
$[]_l$	local matrix
$[]_{master}$	master matrix

Definition

Superscripts

+	beginning of a time step
-	end of a time step
e	edge
i	interior
i, j, k	indices
m	iteration counter
sp	sampling point
v	vertex
T	transpose
$[]^{1,2,3,4}$	denote four different local matrices

Definition

Operators

∇

\otimes

Definition

column vector gradient operator, $[\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}]^T$

outer tensor product

Chapter 1

Introduction

Hypersonic and reentry vehicles are exposed to high temperatures and large temperature gradients as they travel through the atmosphere. Friction between the atmosphere and the vehicle creates excessive heat that could result in significant damage or destruction of the vehicle. The shuttle orbiter has a Thermal Protection System (TPS) designed to protect it during reentry. The TPS on the orbiter consists of ceramic tiles, reinforced carbon-carbon, and insulation blankets that protect the aluminum structure of the orbiter from extreme heat. The next generation crew return vehicle (CRV) will also need protection from the extreme heat of reentry. Future launch vehicle designs will likely use non-insulated hot structures and multifunctional structures. Both hot and multifunctional structures are designed to sustain thermal and aerodynamic loads simultaneously unlike the separate TPS and aluminum structure used for the orbiter. The use of a single structure to sustain thermal and aerodynamic loads can result in significant weight savings. Accurate methods of performing thermal and structural analyses for multifunctional and hot structures are needed.

The thermal response of a material can seriously impact its structural performance. In order to determine accurately the effect of thermal stresses on the material, the temperature distribution within the material must be determined. Finite element methods are used to perform structural and thermal analyses. Commercial finite element codes,

such as NASTRAN [1] and ABAQUS [2], are capable of performing both thermal and structural analyses. These codes rely on three-dimensional elements for the thermal analysis and two-dimensional elements for the structural analysis. In order to perform an accurate analysis including thermal stress, separate meshes are created for the structural and thermal analyses.

Conduction heat transfer problems are usually analyzed with traditional finite element methods in space. In traditional, or h-version, finite elements the temperature approximation is linear over an element. The solution is improved by increasing the number of elements in the mesh until the solution achieves the required accuracy. This type of refinement is computationally expensive since a new mesh must be created when the number of elements is increased. Higher-degree spatial elements, or p-version elements, increase the accuracy of the finite element method by allowing polynomial approximations of higher degree over an element. This allows a smaller number of elements to capture the temperature distribution. The problem with traditional p-version (or Lagrange) elements is that as the degree of the polynomial approximation over an element is increased, additional nodes must be added to each element. In addition to adding new nodes, the basis functions change as the degree of the polynomial approximation is increased. Hence, this method can become computationally expensive since a new mesh must be created when nodes are added to the mesh and the basis functions are redefined.

Commercial finite element codes normally analyze transient problems of conduction heat transfer using finite difference schemes in time. In order to provide an accurate approximation in time, finite difference methods usually require the use of small time steps. This often results in a large number of time steps, increasing the computational expense of the analysis. Finite difference methods also impose limits on the size of the time step needed for stability, with large time steps often resulting in unstable approxima-

tions. Finite difference methods become computationally expensive because the solution is calculated at each time step and numerous time steps are required.

Another difficulty encountered in accurately approximating conduction heat transfer is the variation in the thermal conductivity of a material with temperature. While the thermal conductivity of a material can be approximated as a constant for small temperature ranges, over large temperature ranges the thermal conductivity can vary greatly. This variation in the thermal conductivity makes the problem nonlinear. In order to obtain a solution, an iterative scheme must be used in combination with a method to account for this variation in the thermal conductivity. Current methods used to account for this variation assume that the thermal conductivity is constant over each element or alternatively, the finite element integrals containing the thermal conductivity are integrated using Gaussian quadrature. The first method requires numerous elements in order to achieve an accurate approximation and the second requires a large number of integration points for acceptable accuracy.

A hierarchical p-version space-time finite element method for nonlinear problems with a structurally-compatible mesh would eliminate the weaknesses of traditional finite element and finite difference methods. Hierarchical p-version finite elements allow higher degrees of approximation on each element. Unlike traditional p-version elements, hierarchical elements do not require additional nodes as the degree of the approximation is increased. Instead, existing nodes acquire additional degrees of freedom as the degree of the spatial approximation increases. This eliminates the need for new nodes to be added to the mesh as the degree of spatial approximation is increased. Also unlike traditional p-version basis functions, hierarchical basis functions do not change as the degree of approximation is increased. If solutions requiring higher-degrees of approximation are needed, the additional basis functions can simply be added to the original set of basis

functions. Hierarchical modelling also allows a three-dimensional domain to be collapsed onto a two-dimensional domain. Two-dimensional elements can then be used to create a structurally-compatible finite element mesh that can also perform a three-dimensional thermal analysis. This mesh consists of two-dimensional in-plane elements with an implied through-thickness. The need for separate meshes for structural and thermal analysis can be eliminated by using hierarchical modelling.

A p-version discontinuous Galerkin time-stepping method provides several advantages over the finite difference schemes normally used in time. This finite element method is more accurate than the traditional finite difference methods and allows the temperature over a time step to be represented by higher-degrees of approximation in time. This can greatly reduce the number of time steps necessary to represent accurately the temperature evolution by allowing larger time steps. The unconditional stability of the discontinuous Galerkin method eliminates concerns over the size of the time step.

To model the nonlinearity of the problem, the thermal conductivity is interpolated and an iterative solver is implemented. Higher-degree interpolants can then be used over each element to accurately capture the variation in the thermal conductivity. This method does not rely on a large number of elements or numerous integration points to obtain an accurate approximation.

1.1 Review of Previous Work

Since the development of hierarchical p-version finite elements by Babuška, Szabó, and Peano in 1970's and 1980's, several authors have utilized their benefits to solve a variety of heat transfer problems [3], [4]. Tamma and Saw developed an adaptive method for two-dimensional thermal analysis using p-version hierarchical elements [5]. Gould used these elements to approximate radiation heat transfer problems [6]. Hierarchical p-version

elements have also been utilized by Tomey [7], Lang [8], and Walker [9].

The discontinuous Galerkin method was developed in the 1970's as a method to discretize the neutron transport equation (see [10], [11], and references therein). Later the method was adapted and applied to parabolic partial differential equations [12]. *A-priori* error estimates for the discontinuous Galerkin method have been developed as well [10], [11], [12]. Tomey applied the method to linear, transient heat conduction problems [7].

Several methods of accounting for the variation in the thermal conductivity have been developed. One method assumes that the thermal conductivity is constant over an element. In another method, the finite element integrals containing the thermal conductivity are integrated using Gaussian quadrature [3]. Walker developed a method to interpolate the thermal conductivity for steady-state heat conduction problems using higher-degree interpolants that reduces calculation time by using master matrices [9].

1.2 Purpose

The purpose of this research is to develop, implement, and test a finite element formulation for nonlinear, transient conduction heat transfer. A hierarchical p-version finite element method is used in space and the discontinuous Galerkin finite element method is used in time. An approach for applying non-constant initial conditions is developed to account for initial conditions that vary over the spatial domain. The variation in the thermal conductivity of the material is captured using polynomial interpolants. Master matrices are used to improve computational efficiency. Sample problems with exact solutions are solved to validate the method and demonstrate error convergence.

1.3 Scope

The finite element formulation for nonlinear, transient conduction heat transfer is developed in Chapter 2. The initial boundary value problem is presented and the weak form is derived. The finite element formulation includes the approximation using finite elements, discontinuous Galerkin time-stepping, the Galerkin method in space, the temperature approximation, and temperature-dependent thermal conductivity. The element matrices resulting from the finite element formulation are then presented. The time-stepping solution method for the nonlinear problem is discussed and a section on applying initial and boundary conditions ends the chapter.

The basis functions used in the temperature approximation are discussed in Chapter 3. The basis functions for the time, through-thickness, and in-plane approximations are defined. The properties and advantages of the basis functions are also discussed.

Chapter 4 provides a description of the method used to account for the variation in the thermal conductivity. The functions used to interpolate the thermal conductivity in the time, through-thickness, and in-plane dimensions are defined. The optimization of sampling points and the calculation of the interpolation coefficients are then discussed. The chapter closes by providing the convergence criterion used for the iterative solver.

The implementation of the finite element method using computer code is discussed in Chapter 5. The master matrices and integration techniques are discussed. The method of accounting for the boundary conditions in the computer code ends the chapter.

In Chapter 6 the method is validated. Error convergence is discussed and sample problems are constructed to test the computer code. Results from the sample problems and error convergence plots show the validity of the method.

The final chapter contains a summary of this research, the conclusions drawn, and some suggestions for future work.

Chapter 2

Finite Element Method

The partial differential equation governing nonlinear, transient conduction heat transfer is difficult to solve analytically. The partial differential equation is nonlinear because the thermal conductivity varies as a function of temperature. The finite element method is used to obtain an approximate solution to the nonlinear partial differential equation.

A finite element method for solving the nonlinear, transient conduction heat transfer problem is discussed in this chapter. The method uses hierarchical p-version finite element methods in space and time. The Galerkin finite element method is used in space and a discontinuous Galerkin finite element method is used in time. To account for the temperature-dependent thermal conductivity, interpolation is used along with fixed point iteration. The notation presented here is based on the notation of Walker [9] and Tomey [7].

2.1 Initial Boundary Value Problem

Using the principle of the conservation of energy, the governing partial differential equation for nonlinear, transient conduction heat transfer in a three-dimensional domain is

$$\rho c_p \frac{\partial u}{\partial t} - \nabla^T (k(u) \nabla u) = Q \quad (2.1)$$

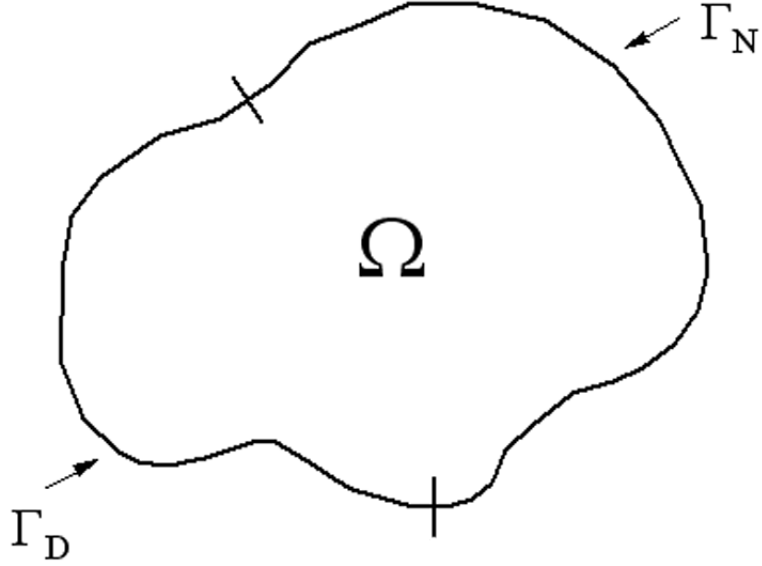


Figure 2.1: Three-dimensional spatial domain, Ω , with Dirichlet and Neumann boundary conditions

where u is a scalar function that represents temperature, ρ is the density of the material, c_p is the specific heat of the material, t is a scalar that represents time, ∇ is the gradient operator $[\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}]^T$, $k(u)$ is the temperature-dependent thermal conductivity of an isotropic material, and Q is the scalar internal heat generation function. By letting $\bar{q} = -k(u)\nabla u$ to simplify notation for the time being, (2.1) is simplified

$$\rho c_p \frac{\partial u}{\partial t} + \nabla^T \bar{q} = Q \quad (2.2)$$

The boundary of the three-dimensional spatial domain is partitioned according to the type of boundary conditions prescribed

$$\Gamma = \Gamma_D + \Gamma_N \quad (2.3)$$

where Γ_D represents portions of the domain with Dirichlet or essential boundary conditions and Γ_N represents portions of the domain with Neumann or natural boundary conditions. A sample domain and boundary is shown in Figure 2.1. The initial and

boundary conditions for the partial differential equation are

$$\begin{aligned} u &= u_D && \text{on } \Gamma_D \\ -(k(u)\nabla u)^T \hat{\mathbf{n}} &= q_s && \text{on } \Gamma_N \\ u(x, y, z, t_i) &= g(x, y, z) && x, y, z \in \Omega \end{aligned} \quad (2.4)$$

where $\hat{\mathbf{n}}$ is a unit vector normal to the boundary, u_D and q_s are the specified temperature and heat flux on the boundaries, and $g(x, y, z)$ is a function that represents the temperature distribution in the spatial domain, Ω , at some initial time, t_i .

A solution to (2.1) is sought that satisfies the partial differential equation subject to the given initial and boundary conditions.

2.2 Weak Formulation

To facilitate the development of the finite element method, the problem is cast into the weak, or variational form. First (2.2) is multiplied by a test function, v . Next, the term Qv is subtracted from both sides of the equation and the equation is integrated over the the spatial domain, Ω , and the time interval of interest, $(t_i, T]$, where without loss of generality, t_i is set equal to zero.

$$\int_0^T \int_{\Omega} \left[\rho c_p v \frac{\partial u}{\partial t} - v \nabla^T \bar{q} - Qv \right] d\Omega dt = 0 \quad (2.5)$$

The test function can be any integrable function and is subject to less stringent regularity requirements than the dependent variable, u . At this point in the formulation, this weighted integral statement is equivalent to the partial differential equation and does not include the boundary conditions [13].

The product rule and the divergence theorem are used to weaken the regularity requirement on u and include the boundary conditions in the finite element formulation.

By the product rule

$$\int_{\Omega} v \nabla^T \bar{q} d\Omega = \int_{\Omega} \nabla^T (\bar{q}v) - \int_{\Omega} (\nabla v)^T \bar{q} d\Omega \quad (2.6)$$

Using the divergence theorem to convert a volume integral into a surface integral [14, pg. 270], the first integral on the right hand side of (2.6) is broken into two parts

$$\int_{\Omega} \nabla^T (\bar{q}v) = \int_{\Gamma} v \bar{q}^T \hat{\mathbf{n}} \, d\Gamma = \int_{\Gamma_D} v \bar{q}^T \hat{\mathbf{n}} \, d\Gamma_D + \int_{\Gamma_N} v \bar{q}^T \hat{\mathbf{n}} \, d\Gamma_N \quad (2.7)$$

The test function is required to be zero on the essential boundaries. In other words, since $v = 0$ on Γ_D , (2.7) becomes

$$\int_{\Omega} \nabla^T (\bar{q}v) = \int_{\Gamma_N} v \bar{q}^T \hat{\mathbf{n}} \, d\Gamma_N \quad (2.8)$$

Substituting (2.8) into (2.6)

$$\int_{\Omega} v \nabla^T \bar{q} \, d\Omega = \int_{\Gamma_N} v \bar{q}^T \hat{\mathbf{n}} \, d\Gamma_N - \int_{\Omega} (\nabla v)^T \bar{q} \, d\Omega \quad (2.9)$$

Then substituting (2.9) into (2.5) and distributing the integral over the spatial domain yields

$$\int_0^T \left[\int_{\Omega} \rho c_p v \frac{\partial u}{\partial t} \, d\Omega - \int_{\Omega} (\nabla v)^T \bar{q} \, d\Omega + \int_{\Gamma_N} v \bar{q}^T \hat{\mathbf{n}} \, d\Gamma_N - \int_{\Omega} Qv \, d\Omega \right] dt = 0 \quad (2.10)$$

Next, the third and fourth terms on the left hand side of the (2.10) are subtracted from both sides of the equation. Finally, expanding the substitution made earlier, $\bar{q} = -k(u)\nabla u$, and defining $q_s = -(k(u)\nabla u)^T \hat{\mathbf{n}}$, the weak form is obtained

$$\begin{aligned} & \int_0^T \left[\int_{\Omega} \rho c_p v \frac{\partial u}{\partial t} \, d\Omega + \int_{\Omega} (\nabla v)^T (k(u)\nabla u) \, d\Omega \right] dt \\ &= \int_0^T \left(\int_{\Omega} Qv \, d\Omega + \int_{\Gamma_N} v q_s \, d\Gamma_N \right) dt \end{aligned} \quad (2.11)$$

The solution to the weak form of the partial differential equation is identical to the solution of the original partial differential equation assuming the solution is sufficiently smooth [15].

2.3 Finite Element Formulation

The weak form has a solution space that is infinite dimensional. The solution to the weak form, u , can be expressed as an infinite series

$$u(x, y, z, t) = \sum_{j=1}^{\infty} \Upsilon_j(x, y, z, t) a_j \quad (2.12)$$

where Υ_j are global basis functions defined over the space and time domain, Ω and $(0, T]$ respectively, and a_j are unknown constants. Since obtaining an exact solution to the weak form of the problem is often difficult or impossible, the finite element method is used to obtain an approximate solution, \hat{u} . The approximate solution is a truncation of the infinite series to a finite series with N_f terms

$$\hat{u}(x, y, z, t) = \sum_{j=1}^{N_f} \Upsilon_j(x, y, z, t) a_j \quad (2.13)$$

A projection of the infinite dimensional solution space onto a finite dimensional solution space is sought. This projection space is a subspace of the infinite dimensional solution space.

In the finite element method, the domain is partitioned into elements. For the problems discussed in this research, space-time elements are used since the finite element method is used in both space and time. The finite element mesh spans the entire spatial and temporal domain of the problem. The mesh is created by partitioning the domain into elements and the elements are assembled to create a global mesh.

In Figure 2.2, a sample space-time mesh for a one-dimensional transient problem is shown. The nodes in the mesh are represented by the symbols \bullet and \times . A space-time element is denoted by D_e , where e is the number of the sample element. The element D_2 is represented in the global, element, and local contexts. In the element context, the spatial component of D_2 is denoted as Ω_e and the time component of D_2 is denoted by

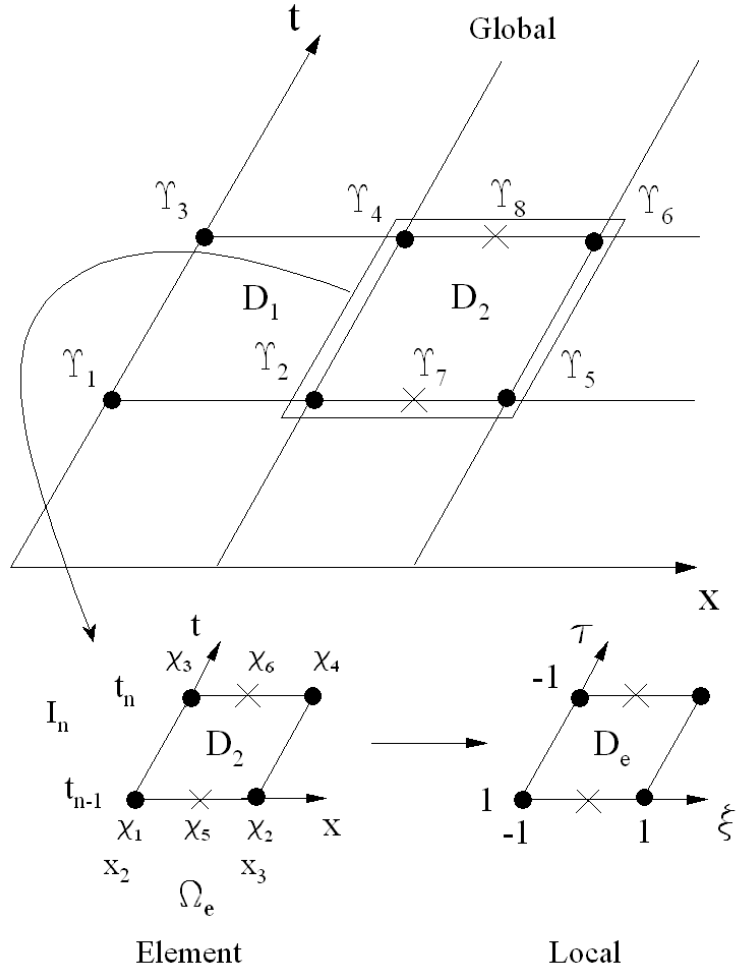


Figure 2.2: Space-time mesh

I_n . In the global and element contexts, D_2 is written in terms of the global variables x and t . In the local context, D_2 is expressed in terms of the local variables, ξ and τ . The local domain is defined from -1 to 1 since the basis functions used in this work are defined over that interval. The local domain for the basis functions is discussed in greater detail in Chapter 3.

To form the element basis, the local basis is created first. Then the local basis is mapped to the element's position in the global domain to create the element basis. The global basis is formed by a combination of the element basis and this process is illustrated using Figure 2.2. In Figure 2.2, the element and global basis functions are listed above

their corresponding nodes in the appropriate context. The element basis functions for D_1 are χ_1 , χ_2 , χ_3 , and χ_4 and the global basis functions, in the same order, are given above the global nodes in Figure 2.2. The global and local basis functions for D_2 are also shown. The global basis functions Υ_1 , Υ_3 , Υ_5 , Υ_6 , Υ_7 , and Υ_8 are identical to the element basis functions χ_1 , χ_3 , χ_2 , χ_4 , χ_5 , and χ_6 respectively since these global basis functions are not shared among the spatial elements. However, Υ_2 is a combination of the element basis functions χ_2 from element one and χ_1 from element two. Also, Υ_4 is a combination of the element basis functions χ_4 from element one and χ_3 from element two.

The approximate solution over the time step t_{n-1} to t_n and the entire spatial domain is given by scaling each one of the global basis functions by the corresponding global constant, a_j , and summing. These global constants are calculated by solving the finite element equations. The approximate solution over the time step t_{n-1} to t_n for the sample mesh shown in Figure 2.2 is

$$\hat{u}_n = \Upsilon_1 a_1 + \Upsilon_2 a_2 + \Upsilon_3 a_3 + \Upsilon_4 a_4 + \Upsilon_5 a_5 + \Upsilon_6 a_6 + \Upsilon_7 a_7 + \Upsilon_8 a_8 \quad (2.14)$$

The approximate solution can also be represented over each element in terms of the element basis functions. The approximate solution for D_2 over a single time step, $\hat{u}_{n,2}$ is given by multiplying each element basis function by the appropriate global constant and summing

$$\hat{u}_{n,2} = \chi_1 a_2 + \chi_2 a_5 + \chi_3 a_4 + \chi_4 a_6 + \chi_5 a_7 + \chi_6 a_8 \quad (2.15)$$

To represent two and three-dimensional spatial solutions using element basis functions, the process is the same. The general approximate solution for any element over a single time step, $\hat{u}_{n,e}$, is expressed as

$$\hat{u}_{n,e} = \sum_{m=1}^{N_b} \chi_m(\Omega, t) a_m \quad (2.16)$$

where χ_m are the element basis functions, N_b is the total number of basis functions for that element, and a_m are the global constants associated with the global basis functions that span the element. In a more compact notation

$$\hat{u}_{n,e} = \chi^T \mathbf{a}_{n,e} \quad (2.17)$$

where χ^T is a row vector of N_b element basis functions and $\mathbf{a}_{n,e}$ is a column vector of the global constants for that space-time element.

2.3.1 Discontinuous Galerkin Method in Time

The global time domain is specified from time zero to some later time, T . To apply the finite element method in time, the global time domain is partitioned into n_t elements so that the time interval of each time step is specified by $I_n = [t_{n-1}, t_n]$. Each time step has a length of $\Delta t = t_n - t_{n-1}$.

A major advantage of the discontinuous Galerkin method is that the method is unconditionally stable for linear parabolic problems [16]. This method allows for discontinuities or jumps in the solution between the time steps. An example of a discontinuous solution in time is shown in Figure 2.3. In Figure 2.3, the values of the solution at a discontinuity are given by u_n^- and u_n^+ such that

$$\begin{aligned} u_n^- &= \lim_{\epsilon \rightarrow 0} u(\cdot, t_n - \epsilon) \\ u_n^+ &= \lim_{\epsilon \rightarrow 0} u(\cdot, t_n + \epsilon) \end{aligned} \quad (2.18)$$

These represent the values of the solution at some time t_n for the left and right elements respectively. The difference in the solution at a discontinuity is given by

$$[u_n] = u_n^+ - u_n^- \quad (2.19)$$

Since the exact solution is continuous in time, continuity should be enforced in some way. Continuity is weakly enforced by multiplying the two temperature values at the

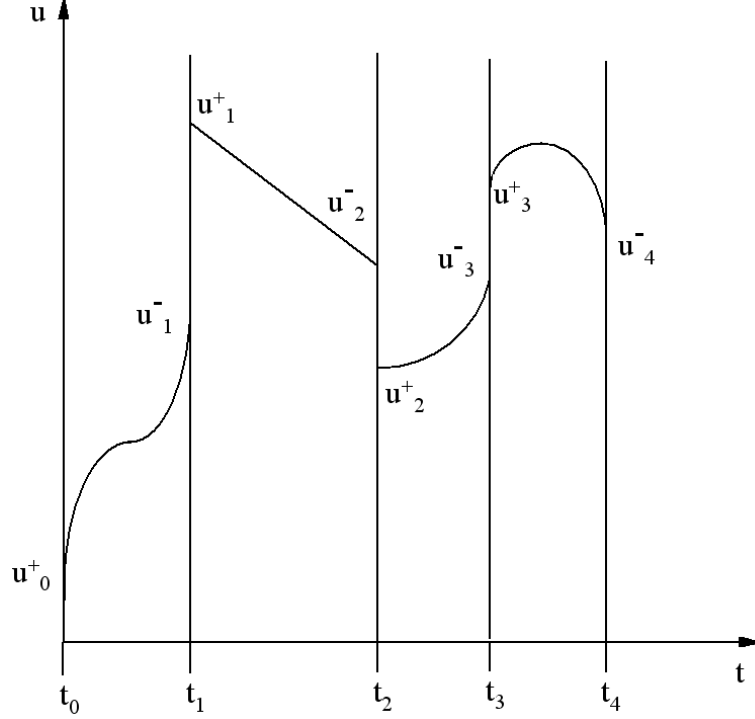


Figure 2.3: Discontinuous Solution in Time

discontinuous point by a test function evaluated at t_n , integrating over the space domain, summing the time steps, and equating [7, pg. 13].

$$\sum_{n=0}^{n_t} \int_{\Omega} \rho c_p u_n^+ v_n^+ d\Omega = \sum_{n=0}^{n_t} \int_{\Omega} \rho c_p u_n^- v_n^+ d\Omega \quad (2.20)$$

Now that a time stepping scheme has been chosen, the weak form must be modified to allow for discontinuous solutions. By adding the second line of (2.20) to (2.11), the discontinuous Galerkin formulation of the weak form is obtained.

$$\begin{aligned} & \sum_{n=0}^{n_t} \left[\int_{I_n} \left(\int_{\Omega} \rho c_p v_n \frac{\partial u_n}{\partial t} d\Omega + \int_{\Omega} (\nabla v_n)^T (k(u_n) \nabla u_n) d\Omega \right) dt + \int_{\Omega} \rho c_p u_n^+ v_n^+ d\Omega \right] \\ &= \sum_{n=0}^{n_t} \left[\int_{I_n} \left(\int_{\Omega} Q v_n d\Omega + \int_{\Gamma_{N_e}} v_n q_s d\Gamma_N \right) dt + \int_{\Omega} \rho c_p u_n^- v_n^+ d\Omega \right] \end{aligned} \quad (2.21)$$

2.3.2 Approximation of Solution

Now an approximate solution to the discontinuous Galerkin formulation of the weak form is sought. As discussed in Section 2.3, by projecting the infinite dimensional solution

space to a finite dimensional subspace, an approximate solution to the discontinuous Galerkin formulation of the weak form is obtained. The exact solution over a time step, u_n , becomes \hat{u}_n and the test function over a time step, v_n , becomes \hat{v}_n such that \hat{u}_n and \hat{v}_n will represent functions from the subspaces of the exact solution space when summed over all time steps. By approximating u_n and v_n (2.21) becomes

$$\begin{aligned} & \sum_{n=0}^{n_t} \left[\int_{I_n} \left(\int_{\Omega} \rho c_p \hat{v}_n \frac{\partial \hat{u}_n}{\partial t} d\Omega + \int_{\Omega} (\nabla \hat{v}_n)^T (k(\hat{u}_n) \nabla \hat{u}_n) d\Omega \right) dt + \int_{\Omega} \rho c_p \hat{u}_n^+ \hat{v}_n^+ d\Omega \right] \\ = & \sum_{n=0}^{n_t} \left[\int_{I_n} \left(\int_{\Omega} Q \hat{v}_n d\Omega + \int_{\Gamma_{N_e}} \hat{v}_n q_s d\Gamma_{N_e} \right) dt + \int_{\Omega} \rho c_p \hat{u}_n^- \hat{v}_n^+ d\Omega \right] \end{aligned} \quad (2.22)$$

The domain has been partitioned into space-time elements and in order to use the finite element method, the individual elements must be assembled to form the global domain in both space and time. The time domain has been broken into steps, but the spatial domain still needs to be partitioned. This is done by summing over the total number of elements at a particular time step, N_e . The approximate solution at a time step, \hat{u}_n , becomes $\hat{u}_{n,e}$ and the test function at a time step, \hat{v}_n , becomes $\hat{v}_{n,e}$ such that $\hat{u}_{n,e}$ and $\hat{v}_{n,e}$ represent the approximate solution over a space-time element at a specific time step. When these space-time elements are assembled, they form the entire spatial domain over that time step. By summing over the space-time elements at each time step (2.22) becomes

$$\begin{aligned} & \sum_{n=0}^{n_t} \sum_{e=1}^{N_e} \left[\int_{I_n} \left(\int_{\Omega_e} \rho c_p \hat{v}_{n,e} \frac{\partial \hat{u}_{n,e}}{\partial t} d\Omega_e + \int_{\Omega_e} (\nabla \hat{v}_{n,e})^T (k_e(\hat{u}_{n,e}) \nabla \hat{u}_{n,e}) d\Omega_e \right) dt \right. \\ & \quad \left. + \int_{\Omega_e} \rho c_p \hat{u}_{n,e}^+ \hat{v}_{n,e}^+ d\Omega_e \right] \quad (2.23) \\ = & \sum_{n=0}^{n_t} \sum_{e=1}^{N_e} \left[\int_{I_n} \left(\int_{\Omega_e} Q \hat{v}_{n,e} d\Omega_e + \int_{\Gamma_{N_e}} \hat{v}_{n,e} q_s d\Gamma_{N_e} \right) dt + \int_{\Omega_e} \rho c_p \hat{u}_{n,e}^- \hat{v}_{n,e}^+ d\Omega_e \right] \end{aligned}$$

For an initial boundary value problem, an initial condition, $u(x, y, z, t_i)$ is given. From this initial condition, the approximate temperature distribution over the mesh at the left end of the first time step, \hat{u}_0^+ , is known. Using this information, an explicit time

stepping scheme is used so that the temperature distribution for the current time step is determined using the solution from the previous time step. Modifying (2.23) to reflect time stepping and rearranging, the formulation for the time stepping scheme becomes

$$\begin{aligned} & \sum_{e=1}^{N_e} \left[\int_{\Omega_e} \left(\int_{I_n} \left[\rho c_p \hat{v}_{n,e} \frac{\partial \hat{u}_{n,e}}{\partial t} + (\nabla \hat{v}_{n,e})^T (k_e(\hat{u}_{n,e}) \nabla \hat{u}_{n,e}) \right] dt + \rho c_p \hat{u}_{n,e}^+ \hat{v}_{n,e}^+ \right) d\Omega_e \right] \\ = & \sum_{e=1}^{N_e} \left[\int_{\Omega_e} \left(\int_{I_n} Q \hat{v}_{n,e} dt + \rho c_p \hat{u}_{n-1,e}^- \hat{v}_{n-1,e}^+ \right) d\Omega_e + \int_{\Gamma_{N_e}} \int_{I_n} \hat{v}_{n,e} q_s dt d\Gamma_{N_e} \right] \quad (2.24) \end{aligned}$$

2.3.3 Temperature Approximation

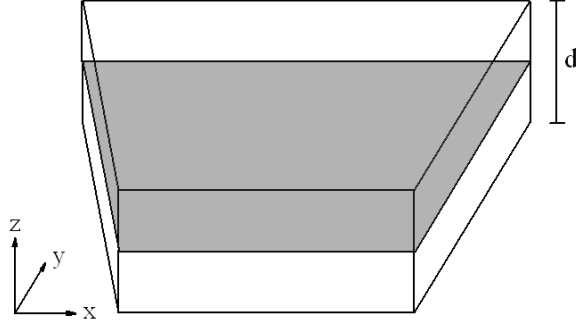


Figure 2.4: Arbitrary Two-Dimensional Spatial Domain with a Uniform Thickness

The approximate solution over a space-time element is given by (2.17). The space-time basis is defined to be a polynomial in space and time. This polynomial is created by a product of three separate polynomials, two in space and one in time. In this research, the domain is assumed to be an arbitrary two-dimensional domain with a uniform thickness d , as shown in Figure 2.4. By making this assumption, the m^{th} element basis function, χ_m , from (2.16), is defined as a product of the basis functions

$$\begin{aligned} \chi_m &= \phi_i(x, y) \theta_j(t) \psi_k(z) \\ i &= 1, \dots, N_{ip}; \quad j = 1, \dots, N_t; \quad k = 1, \dots, N_{tk} \\ m &= (i - 1) N_t N_{tk} + (j - 1) N_{tk} + k \end{aligned} \quad (2.25)$$

where $\phi_i(x, y)$ is a basis function in the spatial directions x and y , $\theta_j(t)$ is a basis function in time, $\psi_k(z)$ is a basis function in the z spatial direction, N_{ip} , N_t , and N_{tk} represent

the total number of basis functions in the in-plane (x - y), time, and through-thickness directions respectively such that $N_b = N_{ip}N_tN_{tk}$. These polynomials are described in greater detail in the next chapter. Then, χ^T is a row vector of all N_b basis functions from (2.26). Using this definition for χ^T , the approximate solution over an element from (2.17) becomes

$$\hat{u}_{n,e} = \sum_{i=1}^{N_{ip}} \sum_{j=1}^{N_t} \sum_{k=1}^{N_{tk}} \phi_i(x, y) \theta_j(t) \psi_k(z) a_m \quad (2.26)$$

This equation is written in a more elegant form using the outer tensor product operator, \otimes , and by writing ϕ , θ , and ψ as vectors. Also, note that the space-time dependence of the basis functions is dropped from the notation for simplicity.

$$\hat{u}_{n,e} = \chi^T \mathbf{a}_{n,e} = (\phi \otimes \theta \otimes \psi)^T \mathbf{a}_{n,e} \quad (2.27)$$

The Galerkin method is used in the spatial approximation so that the test function, $\hat{v}_{n,e}$ is defined using the same basis functions that were used for the approximate solution

$$\hat{v}_{n,e} = \chi_m = \phi_i \theta_j \psi_k \quad (2.28)$$

where $\hat{v}_{n,e}$ is a test function on an element for a single time step. By substituting every basis function for the test function, a set of equations can be written and the test function, $\hat{v}_{n,e}$, is replaced by a vector of the basis functions

$$\hat{v}_{n,e} = \chi = (\phi \otimes \theta \otimes \psi) \quad (2.29)$$

The outer tensor product operator, \otimes , applies to matrices and vectors. Below a sample is shown to illustrate how the operator is used. Given

$$[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad [B] = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \quad \{c\} = \begin{Bmatrix} c_1 \\ c_2 \end{Bmatrix} \quad \{d\} = \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix}$$

then

$$[A] \otimes [B] = \begin{bmatrix} a_{11}b_{11} & a_{11}b_{12} & a_{12}b_{11} & a_{12}b_{12} & a_{13}b_{11} & a_{13}b_{12} \\ a_{11}b_{21} & a_{11}b_{22} & a_{12}b_{21} & a_{12}b_{22} & a_{13}b_{21} & a_{13}b_{22} \\ a_{21}b_{11} & a_{21}b_{12} & a_{22}b_{11} & a_{22}b_{12} & a_{23}b_{11} & a_{23}b_{12} \\ a_{21}b_{21} & a_{21}b_{22} & a_{22}b_{21} & a_{22}b_{22} & a_{23}b_{21} & a_{23}b_{22} \\ a_{31}b_{11} & a_{31}b_{12} & a_{32}b_{11} & a_{32}b_{12} & a_{33}b_{11} & a_{33}b_{12} \\ a_{31}b_{21} & a_{31}b_{22} & a_{32}b_{21} & a_{32}b_{22} & a_{33}b_{21} & a_{33}b_{22} \end{bmatrix}$$

and

$$\{c\} \otimes \{d\} = \begin{Bmatrix} c_1d_1 \\ c_1d_2 \\ c_1d_3 \\ c_2d_1 \\ c_2d_2 \\ c_2d_3 \end{Bmatrix}$$

Now, substituting the first part from (2.27) and (2.29), and leaving the thermal conductivity as a function of temperature, (2.24) is written as

$$\begin{aligned} & \sum_{e=1}^{N_e} \left[\int_{\Omega_e} \left(\int_{I_n} \left[\rho c_p \chi \frac{\partial \chi^T}{\partial t} \mathbf{a}_{n,e} + k_e(\hat{u}_{n,e}) (\nabla \chi^T)^T \nabla \chi^T \mathbf{a}_{n,e} \right] dt + \rho c_p \chi^+ (\chi^+)^T \mathbf{a}_{n,e} \right) d\Omega_e \right] \\ &= \sum_{e=1}^{N_e} \left[\int_{\Omega_e} \left(\int_{I_n} Q \chi dt + \rho c_p \chi^- (\chi^+)^T \mathbf{a}_{n-1,e} \right) d\Omega_e + \int_{\Gamma_{N_e}} \int_{I_n} \chi q_s dt d\Gamma_{N_e} \right] \end{aligned} \quad (2.30)$$

The + and – superscripts denote evaluation of the basis functions at the beginning or end of a time element, respectively. The element matrices and load vectors in (2.30) are explicitly defined as

$$\begin{aligned} [C_e] &= \int_{\Omega_e} \int_{I_n} \rho c_p \chi \frac{\partial \chi^T}{\partial t} dt d\Omega_e \\ [K_e(\hat{u}_{n,e})] &= \int_{\Omega_e} \int_{I_n} k_e(\hat{u}_{n,e}) (\nabla \chi^T)^T \nabla \chi^T dt d\Omega_e \\ [M_e^{++}] &= \int_{\Omega_e} \rho c_p \chi^+ (\chi^+)^T d\Omega_e \\ [M_e^{-+}] &= \int_{\Omega_e} \rho c_p \chi^- (\chi^+)^T d\Omega_e \\ \{H_e\} &= \int_{\Omega_e} \int_{I_n} Q \chi dt d\Omega_e + \int_{\Gamma_{N_e}} \int_{I_n} \chi q_s dt d\Gamma_{N_e} \end{aligned} \quad (2.31)$$

where $[C_e]$ represents the element capacitance matrix, $[K_e(\hat{u}_{n,e})]$ represents the element stiffness (conductance) matrix, $[M_e^{++}]$ and $[M_e^{-+}]$ are element mass matrices, and $\{H_e\}$

is the element load vector. Using the matrix and vector notation from (2.31), (2.30) can be written as

$$\sum_{e=1}^{N_e} \left[[C_e] + [K_e(\hat{u}_{n,e})] + [M_e^{++}] \right] \{\mathbf{a}_{n,e}\} = \sum_{e=1}^{N_e} \{H_e\} + [M_e^{-+}] \{\mathbf{a}_{n-1,e}\} \quad (2.32)$$

These element matrices and vectors are assembled to form a global system of equations that is solved at each time step. This assembly of element equations is denoted by the summation over the elements. By using the explicit time stepping scheme, (2.32) is solved n_t times, once for each time step.

2.3.4 Temperature-Dependent Thermal Conductivity

The temperature dependence of the thermal conductivity must be taken into account in the finite element formulation. The approximate temperature is defined by multiplying each basis function by the appropriate unknown and summing. Each basis function is a function of the spatial dimensions and time. This means that the thermal conductivity is implicitly a function of the spatial dimensions, time, and the unknowns, $\mathbf{a}_{n,e}$. To approximate the thermal conductivity as a function of temperature, \hat{k}_e is defined over an element

$$k_e(\hat{u}_{n,e}) \approx \hat{k}_e = \sum_{i=1}^{\tilde{N}_{ip}} \sum_{j=1}^{\tilde{N}_t} \sum_{k=1}^{\tilde{N}_{tk}} \tilde{\phi}_i \tilde{\theta}_j \tilde{\psi}_k b_{ijk} = \tilde{\chi}^T \mathbf{b}_{n,e} = (\tilde{\phi} \otimes \tilde{\theta} \otimes \tilde{\psi})^T \mathbf{b}_{n,e} \quad (2.33)$$

where $\tilde{\phi}$, $\tilde{\theta}$, $\tilde{\psi}$ are vectors that represent the interpolation functions over the in-plane, time, and through-thickness dimensions respectively, \tilde{N}_{ip} , \tilde{N}_t , and \tilde{N}_{tk} are the number of interpolation functions in the in-plane, time, and through-thickness dimensions, and $\mathbf{b}_{n,e}$ is a vector of interpolation constants that is calculated from $\mathbf{a}_{n,e}$. There are a total of \tilde{N}_b interpolation functions such that

$$\tilde{N}_b = \tilde{N}_{ip} \tilde{N}_t \tilde{N}_{tk} \quad (2.34)$$

The interpolation functions and the determination of the interpolation constants, $\mathbf{b}_{n,e}$, are discussed in Chapter 4. Substituting (2.33) into $[K_e(\hat{u}(\mathbf{a}_{n,e}))]$ in (2.31) gives

$$[K_e(\mathbf{a}_{n,e})] = \int_{\Omega_e} \int_{I_n} \tilde{\chi}^T \mathbf{b}_{n,e} (\nabla \chi^T)^T \nabla \chi^T dt d\Omega_e \quad (2.35)$$

The stiffness matrix is no longer a function of the approximate temperature, rather it is now a function of the unknowns. By taking this new representation of the stiffness matrix into account, (2.32) becomes

$$\sum_{e=1}^{N_e} \left[[C_e] + [K_e(\mathbf{a}_{n,e})] + [M_e^{++}] \right] \{\mathbf{a}_{n,e}\} = \sum_{e=1}^{N_e} \{H_e\} + [M_e^{-+}] \{\mathbf{a}_{n-1,e}\} \quad (2.36)$$

2.4 Element Matrices

The element matrices as defined in (2.31) are inefficient for computational purposes. To make calculation of these matrices more efficient, some modification is required. First, the matrices and vectors without thermal conductivity will be modified. These matrices do not contain interpolation functions or interpolation constants. The definition of χ from (2.27) is inserted into (2.31) and the integral over the spatial domain is expanded into separate integrals for the $x - y$ plane (or in-plane), denoted by A (the area of an element in the $x - y$ plane), and z (or through-thickness) direction. Finally, the temperature approximation basis functions are rearranged using the properties of the outer tensor product and the constants are placed outside the integrals. Then, $[C_e]$ is written as

$$\begin{aligned} [C_e] &= \int_{\Omega_e} \int_{I_n} \rho c_p \chi \frac{\partial \chi^T}{\partial t} dt d\Omega_e \\ &= \int_A \int_{-\frac{d}{2}}^{\frac{d}{2}} \int_{t_{n-1}}^{t_n} \rho c_p (\phi \otimes \theta \otimes \psi) \left(\phi \otimes \frac{d\theta}{dt} \otimes \psi \right)^T dt dz dA \quad (2.37) \\ &= \rho c_p \int_A \phi \phi^T dA \otimes \int_{t_{n-1}}^{t_n} \theta \frac{d\theta^T}{dt} dt \otimes \int_{-\frac{d}{2}}^{\frac{d}{2}} \psi \psi^T dz \end{aligned}$$

Now, $[M_e^{++}]$ is

$$\begin{aligned}
[M_e^{++}] &= \int_{\Omega_e} \rho c_p \chi^+ (\chi^+)^T d\Omega_e \\
&= \int_A \int_{-\frac{d}{2}}^{\frac{d}{2}} \rho c_p (\phi \otimes \theta^+ \otimes \psi) (\phi \otimes \theta^+ \otimes \psi)^T dz dA \\
&= \rho c_p \int_A \phi \phi^T dA \otimes \theta^+ (\theta^+)^T \otimes \int_{-\frac{d}{2}}^{\frac{d}{2}} \psi \psi^T dz
\end{aligned} \tag{2.38}$$

Similarly, $[M_e^{-+}]$ becomes

$$\begin{aligned}
[M_e^{-+}] &= \int_{\Omega_e} \rho c_p \chi^- (\chi^+)^T d\Omega_e \\
&= \int_A \int_{-\frac{d}{2}}^{\frac{d}{2}} \rho c_p (\phi \otimes \theta^- \otimes \psi) (\phi \otimes \theta^+ \otimes \psi)^T dz dA \\
&= \rho c_p \int_A \phi \phi^T dA \otimes \theta^- (\theta^+)^T \otimes \int_{-\frac{d}{2}}^{\frac{d}{2}} \psi \psi^T dz
\end{aligned} \tag{2.39}$$

Then, $\{H_e\}$ is written as

$$\begin{aligned}
\{H_e\} &= \int_{\Omega_e} \int_{I_n} Q \chi dt d\Omega_e + \int_{\Gamma_{N_e}} \int_{I_n} \chi q_s dt d\Gamma_{N_e} \\
&= \int_A \int_{-\frac{d}{2}}^{\frac{d}{2}} \int_{t_{n-1}}^{t_n} Q (\phi \otimes \theta \otimes \psi) dt dz dA \\
&\quad + \int_{\Gamma_{N_e}} \int_{t_{n-1}}^{t_n} (\phi \otimes \theta \otimes \psi) q_s dt d\Gamma_{N_e}
\end{aligned} \tag{2.40}$$

To simplify $[K_e(\mathbf{a}_{n,e})]$, a few steps must be added to account for the thermal conductivity as a function of temperature. Starting from (2.35), the definitions of χ and $\tilde{\chi}$ are inserted from (2.27) and (2.33) and the integral over the spatial domain is expanded into separate integrals for the $x - y$ plane and z direction.

$$[K_e(\mathbf{a}_{n,e})] = \int_A \int_{-\frac{d}{2}}^{\frac{d}{2}} \int_{t_{n-1}}^{t_n} (\tilde{\phi} \otimes \tilde{\theta} \otimes \tilde{\psi})^T \mathbf{b}_{n,e} (\nabla (\phi \otimes \theta \otimes \psi)^T)^T \nabla (\phi \otimes \theta \otimes \psi)^T dt dz dA$$

Next, the thermal conductivity interpolation is written as a sum and the interpolation constant is placed outside the integrals.

$$[K_e(\mathbf{a}_{n,e})] = \sum_{m=1}^{\tilde{N}_b} b_m \int_A \int_{-\frac{d}{2}}^{\frac{d}{2}} \int_{t_{n-1}}^{t_n} \tilde{\phi}_i \tilde{\theta}_j \tilde{\psi}_k (\nabla (\phi \otimes \theta \otimes \psi)^T)^T \nabla (\phi \otimes \theta \otimes \psi)^T dt dz dA$$

The single summation represents the three part summation in (2.33). The index for b_m is given by

$$m = (i - 1)\tilde{N}_t\tilde{N}_{tk} + (j - 1)\tilde{N}_{tk} + k \quad (2.41)$$

such that b_m is the m^{th} interpolation constant on that element. Each piece of the summation represents one part of the stiffness matrix. Then, by considering one piece of the summation and distributing the ∇ operator

$$\begin{aligned} [K_e(\mathbf{a}_{n,e})]_m &= b_m \int_A \int_{-\frac{d}{2}}^{\frac{d}{2}} \int_{t_{n-1}}^{t_n} \tilde{\phi}_i \tilde{\theta}_j \tilde{\psi}_k \left[\left(\frac{\partial \phi}{\partial x} \otimes \theta \otimes \psi \right) \left(\frac{\partial \phi^T}{\partial x} \otimes \theta^T \otimes \psi^T \right) + \right. \\ &\left. \left(\frac{\partial \phi}{\partial y} \otimes \theta \otimes \psi \right) \left(\frac{\partial \phi^T}{\partial y} \otimes \theta^T \otimes \psi^T \right) + \left(\phi \otimes \theta \otimes \frac{d\phi}{dz} \right) \left(\phi^T \otimes \theta^T \otimes \frac{d\phi^T}{dz} \right) \right] dt dz dA \end{aligned}$$

where the m in $[K_e(\mathbf{a}_{n,e})]_m$ represents a single matrix in the summation over all of the matrices generated by the different interpolation functions and constants. In other words

$$[K_e(\mathbf{a}_{n,e})] = \sum_{m=1}^{\tilde{N}_b} [K_e(\mathbf{a}_{n,e})]_m \quad (2.42)$$

The temperature approximation basis functions are rearranged using the properties of the tensor product and the gradient operator. The final form of the m^{th} part of the stiffness matrix is given by

$$\begin{aligned} [K_e(\mathbf{a}_{n,e})]_m &= b_m \left[\int_A \tilde{\phi}_i \left[\frac{\partial \phi}{\partial x} \frac{\partial \phi^T}{\partial x} + \frac{\partial \phi}{\partial y} \frac{\partial \phi^T}{\partial y} \right] dA \otimes \int_{t_{n-1}}^{t_n} \tilde{\theta}_j \theta \theta^T dt \otimes \int_{-\frac{d}{2}}^{\frac{d}{2}} \tilde{\psi}_k \psi \psi^T dz \right. \\ &\left. + \int_A \tilde{\phi}_i \phi \phi^T dA \otimes \int_{t_{n-1}}^{t_n} \tilde{\theta}_j \theta \theta^T dt \otimes \int_{-\frac{d}{2}}^{\frac{d}{2}} \tilde{\psi}_k \frac{d\psi}{dz} \frac{d\psi^T}{dz} dz \right] \quad (2.43) \end{aligned}$$

The stiffness matrix is given by

$$\begin{aligned} [K_e(\mathbf{a}_{n,e})] &= \sum_{m=1}^{\tilde{N}_b} b_m \left[\int_A \tilde{\phi}_i \left[\frac{\partial \phi}{\partial x} \frac{\partial \phi^T}{\partial x} + \frac{\partial \phi}{\partial y} \frac{\partial \phi^T}{\partial y} \right] dA \otimes \int_{t_{n-1}}^{t_n} \tilde{\theta}_j \theta \theta^T dt \otimes \int_{-\frac{d}{2}}^{\frac{d}{2}} \tilde{\psi}_k \psi \psi^T dz \right. \\ &\left. + \int_A \tilde{\phi}_i \phi \phi^T dA \otimes \int_{t_{n-1}}^{t_n} \tilde{\theta}_j \theta \theta^T dt \otimes \int_{-\frac{d}{2}}^{\frac{d}{2}} \tilde{\psi}_k \frac{d\psi}{dz} \frac{d\psi^T}{dz} dz \right] \quad (2.44) \end{aligned}$$

2.5 Time-Stepping Solution Method

The global system of equations given in (2.36),

$$\sum_{e=1}^{N_e} \left[[C_e] + [K_e(\mathbf{a}_{n,e})] + [M_e^{++}] \right] \{\mathbf{a}_{n,e}\} = \sum_{e=1}^{N_e} \{H_e\} + [M_e^{-+}] \{\mathbf{a}_{n-1,e}\}$$

can be written as

$$[S(\mathbf{a}_n)] \{\mathbf{a}_n\} = \{F\}_{n-1} \quad (2.45)$$

where $[S(\mathbf{a}_n)]$ is the global matrix formed from the sum of the three matrices on the left hand side, \mathbf{a}_n is the global vector of unknowns, and $\{F\}_{n-1}$ is the vector on the right hand side. The n and $n - 1$ subscripts denote the time step, such that n is the current time step and $n - 1$ is the previous time step. Since the stiffness matrix, $[K_e(\mathbf{a}_{n,e})]$, in (2.45) is a function of the unknowns, the system of equations is nonlinear and must be solved iteratively. The fixed point method is used in this research [17].

In the fixed point iteration method, an initial guess is made for the values of the global unknowns. Since the problem considered in this work is transient, the solution from the previous time step is used as the initial guess. For the first time step, the initial condition is used. A flowchart for the time-stepping solution method including the fixed point iteration is shown in Figure 2.5. The time-stepping solution method depicted in Figure 2.5 is described in the following paragraphs.

Starting from the initial condition, the values of the global unknowns are calculated and are specified by \mathbf{a}_0 . The method for determining \mathbf{a}_0 from the initial condition is discussed in the next section. Once \mathbf{a}_0 is known, the interpolation constants, $\mathbf{b}_{n,e}$, can be calculated. Now that the interpolation constants are known, $[S(\mathbf{a}_0)]$ can be formulated. The iteration counter is denoted by m and the first step in the problem is written as

$$[S(\mathbf{a}_0)] \{\mathbf{a}_1^m\} = \{F\}_0 \quad (2.46)$$

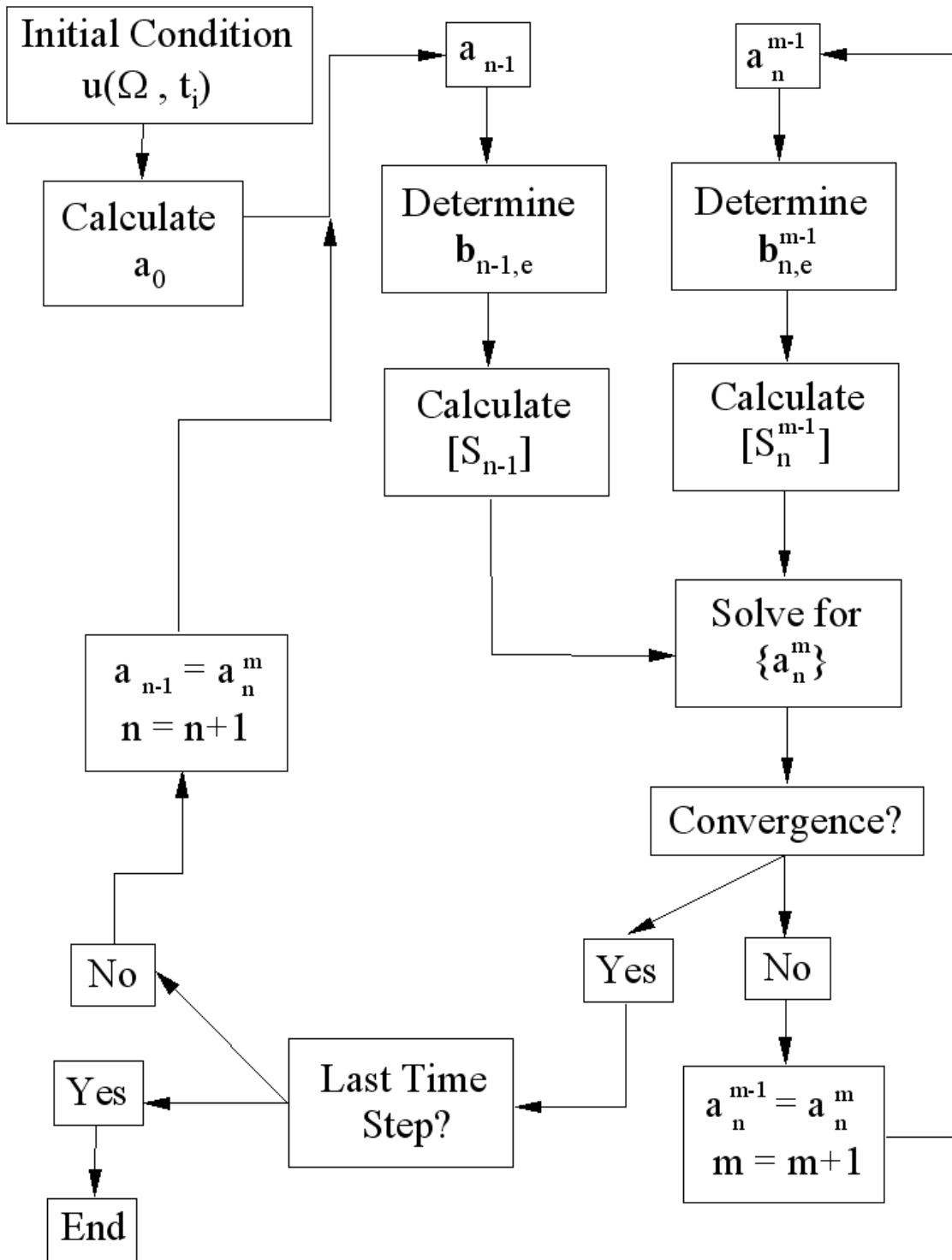


Figure 2.5: Time-Stepping Solution Flowchart

and is solved for the unknowns \mathbf{a}_1^m . Now that \mathbf{a}_1^m is known, the iteration counter, m is increased and these values are used to recalculate $\mathbf{b}_{n,e}$ and $[S]$. Now the problem is written as

$$\left[S \left(\mathbf{a}_1^{m-1} \right) \right] \{ \mathbf{a}_1^m \} = \{ F \}_0 \quad (2.47)$$

This equation is solved iteratively and $\mathbf{b}_{n,e}$ and $[S]$ are updated using the new unknowns each time until the method converges. The right hand side of the problem does not change with each iteration. If the difference, in some quantified manner, between \mathbf{a}^m and \mathbf{a}^{m-1} is less than a specified tolerance, then the iteration has converged and the final solution at the time step, \mathbf{a}_n , has been determined.

After the iteration converges, the time step counter, n , is increased to begin the next time step and the solution that was just found is now written as \mathbf{a}_{n-1} . The problem is now cast as

$$\left[S \left(\mathbf{a}_{n-1} \right) \right] \{ \mathbf{a}_n^m \} = \{ F \}_{n-1} \quad (2.48)$$

Again, this equation is solved for the unknowns and becomes

$$\left[S \left(\mathbf{a}_n^{m-1} \right) \right] \{ \mathbf{a}_n^m \} = \{ F \}_{n-1} \quad (2.49)$$

Then the problem is solved iteratively as before and $[S]$ is updated using the previous iterate until convergence is attained. The time step counter is then updated and the method repeats from (2.48) for the next time step. This process is repeated until the unknowns for the final time step have been determined.

2.6 Applying Initial and Boundary Conditions

The initial and boundary conditions for the problem are given by (2.4). Before (2.46), (2.47), (2.48), and (2.49) are solved for the unknowns, the Dirichlet boundary conditions must be enforced. For these boundary conditions, the temperature is specified on the

boundary and the values of the global unknowns are known. To enforce these boundary conditions, the penalty method is used. This method is described in detail in Chapter 5. The Neumann boundary conditions are included in the element load vector H_e from (2.40). For the Neumann boundary conditions, q_s is specified and these values are used in the calculation of the element load vector. The heat flux function, q_s , is contained in the second term on the right hand side of (2.40). The computer implementation for these boundary conditions is also discussed in Chapter 5.

To start the time-stepping solution method described, the initial condition vector, \mathbf{a}_0 , must be determined. If the initial condition is a constant, then the linear spatial degrees of freedom are set to the constant temperature specified by the initial condition and all other degrees of freedom are set to zero. Occasionally, the initial condition, $g(x, y, z)$, is specified as a function of the spatial domain. In this case, \mathbf{a}_0 must be determined using the basis functions. In order to find \mathbf{a}_0 , an approximation of the initial condition, \hat{u}_0 is sought such that

$$\int_{\Omega} \hat{v}_0 \hat{u}_0 \, d\Omega = \int_{\Omega} \hat{v}_0 g(x, y, z) \, d\Omega \quad (2.50)$$

By substituting (2.27) and (2.29) into (2.50)

$$\int_{\Omega} \chi_0 \chi_0^T \mathbf{a}_0 \, d\Omega = \int_{\Omega} \chi_0 g(x, y, z) \, d\Omega \quad (2.51)$$

Then by defining

$$[IC] = \int_{\Omega} \chi_0 \chi_0^T \, d\Omega \quad (2.52)$$

$$\{r\} = \int_{\Omega} \chi_0 g(x, y, z) \, d\Omega \quad (2.53)$$

and moving the constant vector \mathbf{a}_0 outside the integral, (2.51) is written as

$$[IC] \mathbf{a}_0 = \{r\} \quad (2.54)$$

The global matrix, $[IC]$, and the global vector $\{r\}$ from (2.54) are created by forming the element matrices and assembling

$$[IC]_e = \int_A \phi \phi^T dA \otimes \int_{-\frac{d}{2}}^{\frac{d}{2}} \psi \psi^T dz \quad (2.55)$$

$$[IC] = \sum_{e=1}^{N_e} [IC]_e \quad (2.56)$$

$$\{r\}_e = \int_A \int_{-\frac{d}{2}}^{\frac{d}{2}} g(x, y, z)_e (\phi \otimes \psi) dA dz \quad (2.57)$$

$$\{r\} = \sum_{e=1}^{N_e} \{r\}_e \quad (2.58)$$

where $g(x, y, z)_e$ represents the initial condition restricted to an element. These matrices do not contain the time basis, θ , since the initial condition does not vary over time. They also do not contain the interpolation for the thermal conductivity since the temperature distribution has been explicitly defined by the initial condition.

Since the time basis is not used to create the matrices and vectors in (2.55) and (2.57), (2.54) is rewritten without the time basis and solved for $\{w\}$, where $\{w\}$ is a vector of the coefficients that are multiplied by the appropriate basis functions to approximate the initial condition

$$[IC] \{w\} = \{r\} \quad (2.59)$$

The vector $\{w\}$ will have a length of $N_{ip}N_{tk}$ since the time basis was not included. To create \mathbf{a}_0 from $\{w\}$, the degrees of freedom used by the time basis must be included to ensure that \mathbf{a}_0 is the appropriate size. The time basis must be modified since there is no variation over time for the initial condition. To accomplish this, the constant time basis functions are set equal to one and all other time basis functions are set equal to zero. The degrees of freedom must be arranged in the appropriate order so that the order used earlier, $(\phi \otimes \theta \otimes \psi)$, is followed. Finally, \mathbf{a}_0 is created by multiplying the values of $\{w\}$ by the modified time basis functions and by tracking the order of the degrees of freedom.

Chapter 3

Basis Functions

The basis functions and the necessary mapping relationships to convert from a local element to a specific element in the global context are discussed in this chapter. The functions used in this research were chosen because of their desirable numerical properties and the ability to generalize the formulation of the local matrices.

3.1 Time Basis Functions

The time basis functions used in this work are the set of Legendre polynomials, P_n , that can be generated using a recurrence relation [18, pg. 208]

$$P_{n+1}(\tau) = \frac{(2n+1)\tau P_n(\tau) - nP_{n-1}(\tau)}{n+1} \quad (3.1)$$

where $P_0(\tau) = 1$ and $P_1(\tau) = \tau$. The first few Legendre polynomials are

$$\begin{aligned} P_0(\tau) &= 1 \\ P_1(\tau) &= \tau \\ P_2(\tau) &= \frac{1}{2}(3\tau^2 - 1) \\ P_3(\tau) &= \frac{1}{2}(5\tau^3 - 3\tau) \\ P_4(\tau) &= \frac{1}{8}(35\tau^4 - 30\tau^2 + 3) \end{aligned} \quad (3.2)$$

The time basis functions first introduced in (2.26) are defined as

$$\theta_j(\tau) = P_j(\tau) \quad \text{for } j = 0, \dots, p_t \quad (3.3)$$

where p_t is the degree of the polynomial approximation in time. For an approximation of degree p_t , there are N_t time basis functions where $N_t = p_t + 1$.

The Legendre polynomial basis presented here has advantages over the monomial basis used by Tomey [7, pg. 20-2]. The Legendre polynomials are orthogonal over the interval $(-1, 1)$ resulting in element matrices that are less susceptible to computer round off error. Matrix conditioning problems that occur when a monomial basis is used are eliminated by using an orthogonal basis.

By taking into account the relationship between t and τ , a mapping can be established to create the specific global time element from the local time element

$$t = t_{n-1} + \frac{\Delta t}{2} (\tau + 1) \quad (3.4)$$

$$\tau = \frac{2}{\Delta t} (t - t_{n-1}) - 1 \quad (3.5)$$

where t is the time, t_{n-1} is the time at the left end of the time element, Δt is the length of the time element in the global context, $\Delta t = t_n - t_{n-1}$, and τ is the local variable. By differentiating (3.4) and rearranging

$$dt = \frac{\Delta t}{2} d\tau \quad (3.6)$$

$$\frac{d\tau}{dt} = \frac{2}{\Delta t} \quad (3.7)$$

and by using the chain rule

$$\frac{d}{dt} = \frac{d}{d\tau} \frac{d\tau}{dt} = \frac{2}{\Delta t} \frac{d}{d\tau} \quad (3.8)$$

yields the necessary relationships for mapping from the local time element to the specific global time element.

The matrices and vectors that need to be calculated using the time basis are contained in (2.37) through (2.40) and (2.44). Using (3.6) and (3.8) to modify the time basis in (2.37) through (2.40) and (2.44), a local calculation is made and the element specific global time matrices and vectors are formed from the local matrices and vectors

$$[V_t]_e = \int_{t_{n-1}}^{t_n} \theta \frac{d\theta^T}{dt} dt = \int_{-1}^1 \theta \frac{d\theta^T}{d\tau} d\tau = [V_t]_l \quad (3.9)$$

$$\theta^+ (\theta^+)^T = \theta|_{-1} (\theta|_{-1})^T \quad (3.10)$$

$$\theta^- (\theta^+)^T = \theta|_1 (\theta|_{-1})^T \quad (3.11)$$

$$\{L_t\}_e = \int_{t_{n-1}}^{t_n} \theta dt = \frac{\Delta t}{2} \int_{-1}^1 \theta d\tau = \frac{\Delta t}{2} \{L_t\}_l \quad (3.12)$$

$$[M_t^j]_e = \int_{t_{n-1}}^{t_n} \tilde{\theta}_j \theta \theta^T dt = \frac{\Delta t}{2} \int_{-1}^1 \tilde{\theta}_j \theta \theta^T d\tau = \frac{\Delta t}{2} [M_t^j]_l \quad (3.13)$$

where $[V_t]$ is the convective time matrix, $\{L_t\}$ is the time load vector, and $[M_t^j]$ is the time mass matrix for the j^{th} time interpolation function. The subscripts e and l denote element specific global and local time element respectively. The local matrices and vectors on the right hand side of (3.9) through (3.13) are computed first. The element specific matrices and vectors are created by multiplying the local matrices and vectors by the appropriate scaling factor as shown above.

3.2 Through-Thickness Basis Functions

In this research, two-dimensional triangular elements are used to represent a three-dimensional spatial domain. The three-dimensional domain is collapsed onto a two-dimensional mesh with an implied thickness as illustrated in Figure 3.1. The approximation over the implied thickness is contained within each of the in-plane and time degrees of freedom. As shown in (2.27), the temperature variation in the z direction is included in every in-plane and time basis function. Since the variation in the z direction is included

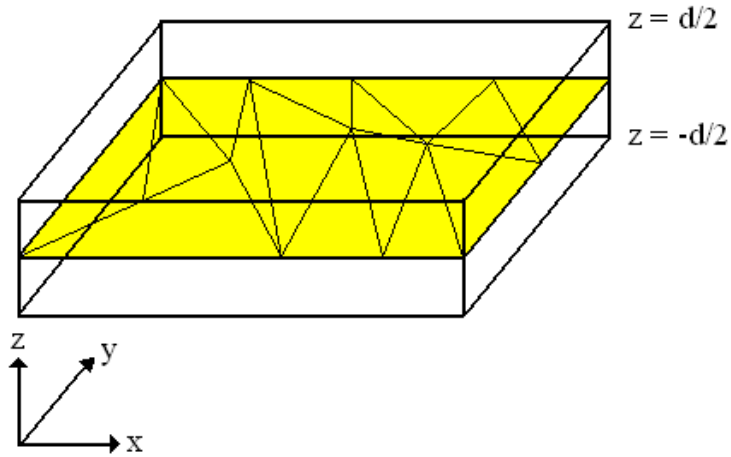


Figure 3.1: Three-Dimensional Domain Collapsed onto a Two-Dimensional Mesh

in every degree of freedom, two-dimensional spatial elements can be used to represent a three-dimensional spatial domain.

A set of one-dimensional hierarchical basis functions are used to represent the temperature distribution in the through-thickness (z) direction. A sample one-dimensional through-thickness element is shown in Figure 3.2. Each end node has one linear basis function associated with it and the middle node is added for higher degrees of approximation. As the degree of the polynomial approximation is increased, the degrees of freedom for the middle node increase accordingly.

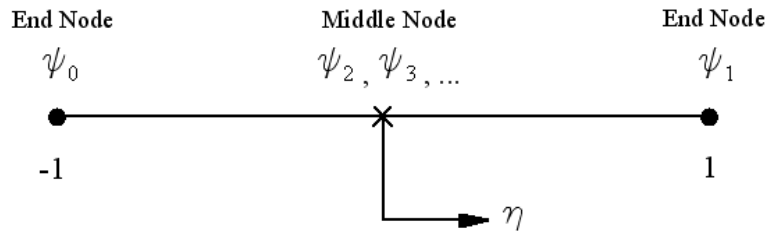


Figure 3.2: One-dimensional hierarchical element

The basis functions of degree two or higher are formed by integrating the Legendre polynomials and multiplying by a scaling factor. The derivatives of the integrated Leg-

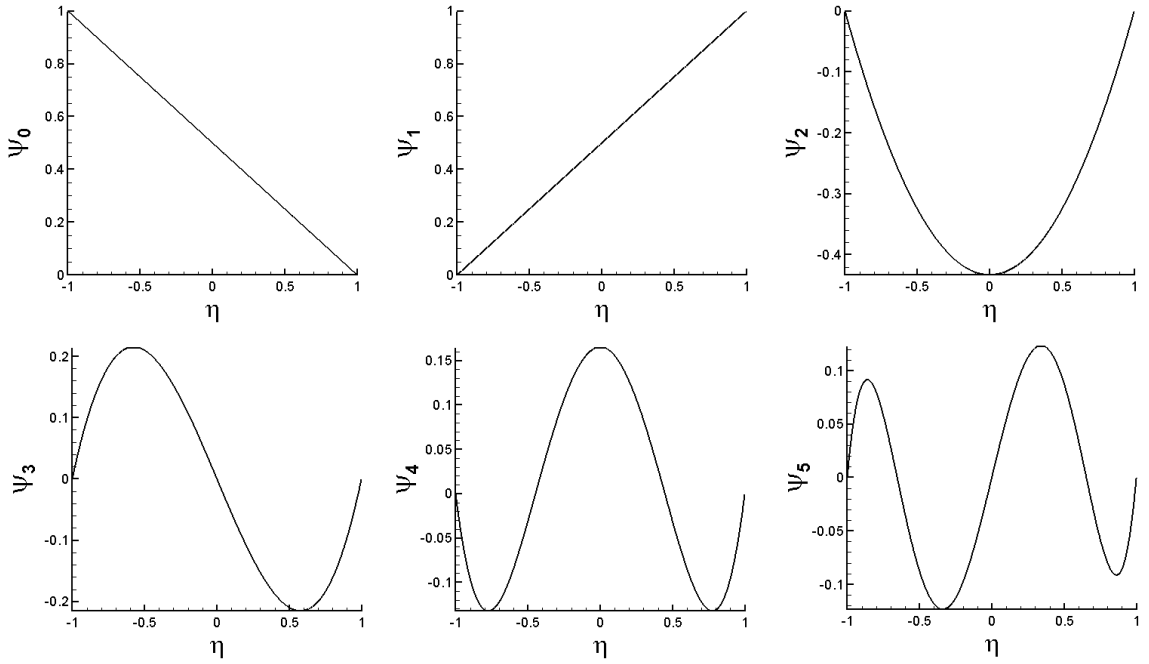


Figure 3.3: Local Basis Functions for $p_{tk} = 5$

endre polynomials are orthogonal over the interval $(-1, 1)$. The local through-thickness basis functions are given by

$$\begin{aligned}
 \psi_0(\eta) &= \frac{1}{2}(1 - \eta) \\
 \psi_1(\eta) &= \frac{1}{2}(1 + \eta) \\
 \psi_k(\eta) &= \frac{1}{\sqrt{2(2k-1)}} [P_k(\eta) - P_{k-2}(\eta)] \quad \text{for } k = 2, \dots, p_{tk} \quad (3.14)
 \end{aligned}$$

where p_{tk} is the degree of the polynomial approximation in the through-thickness direction and P_k are the Legendre polynomials from (3.1). A plot of the local basis functions for $p_{tk} = 5$ is shown in Figure 3.3. For a polynomial approximation of degree p_{tk} at an in-plane node there are $N_{tk} = p_{tk} + 1$ basis functions. If a two-dimensional spatial problem is considered, then $p_{tk} = 0$ and $\psi_0 = 1$. The p-enrichment of the solution is based on the fact that the higher-degree hierarchical basis functions are zero at the end nodes (See Figure 3.3). This allows the higher-degree polynomial terms to be added without creating a new mesh.

The global domain in the z-direction is $-\frac{d}{2} \leq z \leq \frac{d}{2}$. The local basis is used to create the element specific basis by utilizing the relationships between the local coordinate, η , and the global coordinate, z

$$\eta = \frac{2}{d} z \quad (3.15)$$

$$z = \frac{d}{2} \eta \quad (3.16)$$

By differentiating (3.16) and rearranging

$$dz = \frac{d}{2} d\eta \quad (3.17)$$

$$\frac{d\eta}{dz} = \frac{2}{d} \quad (3.18)$$

and by applying the chain rule

$$\frac{d}{dz} = \frac{d}{d\eta} \frac{d\eta}{dz} = \frac{2}{d} \frac{d}{d\eta} \quad (3.19)$$

the necessary relationships for mapping from the local through-thickness element to the specific through-thickness element in the global domain are obtained. The matrices and vectors that need to be calculated using the through-thickness basis are contained in (2.37) through (2.40) and (2.44). Using (3.17) and (3.19) to modify the through-thickness basis in (2.37) through (2.40) and (2.44), a local calculation is made and the element specific matrices and vectors are formed from the local matrices and vectors

$$[M_{tk}]_e = \int_{-\frac{d}{2}}^{\frac{d}{2}} \psi \psi^T dz = \frac{d}{2} \int_{-1}^1 \psi \psi^T d\eta = \frac{d}{2} [M_{tk}]_l \quad (3.20)$$

$$\{L_{tk}\}_e = \int_{-\frac{d}{2}}^{\frac{d}{2}} \psi dz = \frac{d}{2} \int_{-1}^1 \psi d\eta = \frac{d}{2} \{L_{tk}\}_l \quad (3.21)$$

$$[M_{tk}^k]_e = \int_{-\frac{d}{2}}^{\frac{d}{2}} \tilde{\psi}_k \psi \psi^T dz = \frac{d}{2} \int_{-1}^1 \tilde{\psi}_k \psi \psi^T d\eta = \frac{d}{2} [M_{tk}^k]_l \quad (3.22)$$

$$[K_{tk}^k]_e = \int_{-\frac{d}{2}}^{\frac{d}{2}} \tilde{\psi}_k \frac{d\psi}{dz} \frac{d\psi^T}{dz} dz = \frac{2}{d} \int_{-1}^1 \tilde{\psi}_k \frac{d\psi}{d\eta} \frac{d\psi^T}{d\eta} d\eta = \frac{2}{d} [K_{tk}^k]_l \quad (3.23)$$

where $[M_{tk}]$ is the through-thickness mass matrix, $\{L_{tk}\}$ is the through-thickness load vector, and $[M_{tk}^k]$ and $[K_{tk}^k]$ are the through-thickness mass and stiffness matrices for the k^{th} through-thickness interpolation function. The subscripts e and l denote element specific and local respectively. Again, the local matrices and vectors on the right hand side are computed first. The element specific matrices and vectors are created by multiplying the local matrices and vectors by the appropriate scaling factor.

3.3 Hierarchical In-Plane Basis Functions

In this research, two-dimensional triangular elements are used in the $x - y$ or in-plane directions. These triangular elements are hierarchical to allow the p-enrichment of the approximation without mesh refinement. A sample triangular hierarchical element is shown in Figure 3.4.

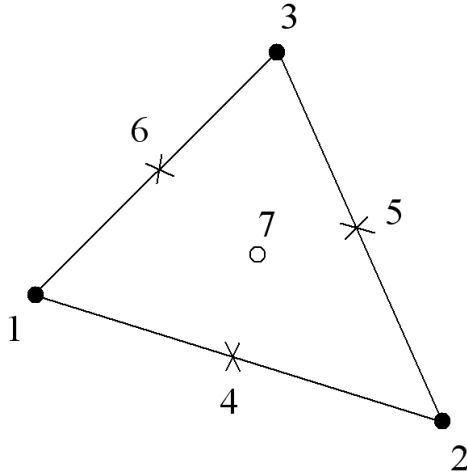


Figure 3.4: Triangular Hierarchical Element Node Numbering Convention

There are seven nodes in a triangular hierarchical element; three vertex nodes denoted by \bullet , three edge nodes denoted by \times , and one interior node denoted by \circ . The basis functions for the triangle are created specifically for each type of node. The convention for numbering the nodes is vertex nodes first, then edge nodes, and finally the interior

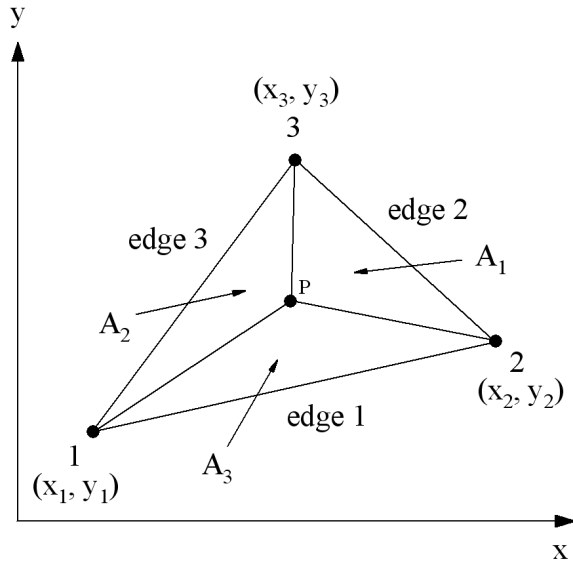


Figure 3.5: Global Coordinates and Areas for a Triangle

node. The vertex nodes on an element are numbered in a counterclockwise direction from the starting node. Edge nodes are numbered starting with the edge between vertex nodes one and two and proceeding counterclockwise.

Since the nodes on a side of the triangle generally do not line up along one cartesian axis, cartesian coordinates are not used to describe the basis functions for triangles. A better set of coordinates for triangular elements is barycentric or area coordinates. An arbitrary triangle with point P contained within the triangle is shown in Figure 3.5. The area coordinates for any point P can be expressed in terms of the areas of three triangles formed by connecting point P to each of the vertices [19, pg. 186]. These areas are labelled according to the opposite vertex node, not according to the edge numbers. Therefore, A_1 is opposite vertex node one, A_2 is opposite vertex node two, and A_3 is opposite vertex node three.

As point P is moved to any position within the triangle, the ratio of the three areas changes. The maximum value for each area will occur when point P is moved to the vertex node of the corresponding area. For example, if point P is moved to vertex node

1, $A_2 = A_3 = 0$ and A_1 is equal to the total area of the triangle. Also note that if point P is moved to anywhere along edge 2, $A_1 = 0$. Similar relationships also hold for the other two areas.

The area coordinates for a triangular element are given by

$$\begin{aligned} L_1 &= \frac{A_1}{A} \\ L_2 &= \frac{A_2}{A} \\ L_3 &= \frac{A_3}{A} \end{aligned} \tag{3.24}$$

where A is the total area of the entire triangle given by

$$A = A_1 + A_2 + A_3 \tag{3.25}$$

Since each area coordinate is a ratio of the total area, the maximum value for an area coordinate is one and the minimum value is zero. The area coordinates are unity when point P is located on their vertex node and zero along the edge opposite that vertex node. From (3.24) and (3.25)

$$L_1 + L_2 + L_3 = 1 \tag{3.26}$$

Since only two area coordinates are needed to locate a point in the $x - y$ plane, the three area coordinates specified by L_1 , L_2 , and L_3 are not linearly independent. Normally when using area coordinates, L_1 and L_2 are taken to be independent and L_3 is the dependent area coordinate according to the equation of constraint given in (3.26)

$$L_3 = 1 - L_1 - L_2 \tag{3.27}$$

Each vertex node of the triangle has one basis function associated with it. The vertex basis functions capture the linear variation in the temperature approximation over the element. The basis functions for the vertex nodes of the triangle are given by the area

coordinates

$$\begin{aligned}
\phi_1^v &= L_1 \\
\phi_2^v &= L_2 \\
\phi_3^v &= L_3
\end{aligned} \tag{3.28}$$

where the subscript denotes the node number and the superscript denotes that it is a vertex basis function. These vertex basis functions are included for any polynomial approximation of temperature in the in-plane direction of degree p_{ip} . For any triangular element, the number of vertex basis functions, N_{ip}^v , is three.

Now that the vertex basis functions have been established for a triangular element, the edge basis functions are defined. The edge basis functions are needed when the polynomial degree of the temperature approximation over an edge is two or higher, $p_{ip}^e \geq 2$. For $p_{ip}^e < 2$ there are no edge nodes and only the vertex basis functions are used. The polynomial degree of the in-plane temperature approximation can be different for each edge. For an in-plane temperature approximation over an edge of degree p_{ip}^e , there are $p_{ip}^e - 1$ edge basis functions for that edge node.

The edge basis functions are created by using the higher-degree one-dimensional hierarchical basis functions from (3.14). An edge coordinate is defined so that the one-dimensional hierarchical basis is evaluated over the appropriate interval of $(-1, 1)$

$$\xi_e = L_k - L_e \tag{3.29}$$

where ξ_e is the edge coordinate for edge e , L_e is the area coordinate for vertex node e , and L_k is the area coordinate of the next vertex node, k , in a counterclockwise direction from vertex node e . Table 3.1 [9, pg. 20] gives the values of e and k for any of the edges. A quadratic polynomial of the edge coordinate is also used in the formulation of the edge

Edge e	Vertex k	Vertex e
1	2	1
2	3	2
3	1	3

Table 3.1: Index Permutation for Edge Basis Functions

basis functions

$$\frac{1 - (\xi_e)^2}{4} \quad (3.30)$$

To form the hierarchical triangular edge basis functions, the one-dimensional higher-order basis functions are written in terms of the edge coordinate, multiplied by the scaling factor, $L_k L_e$, and divided by (3.30). For edge e , the hierarchical basis functions are given by

$$\phi_i^e = L_k L_e \left(\frac{4}{1 - (\xi_e)^2} \right) \psi_{i+1}(\xi_e) \quad \text{for } i = 1, 2, \dots, p_{ip}^e - 1 \quad (3.31)$$

where ϕ_i^e is the i^{th} edge basis function for edge e , ξ_e is the edge coordinate for edge e , and ψ_{i+1} is the $i^{\text{th}} + 1$ hierarchical basis function given by (3.14). To ensure that the edge basis functions are non-zero along their own edge and zero along the other edges, the scaling factor $L_k L_e$ is included. This scaling increases the polynomial degree of the edge basis functions by two. The edge basis is returned to the appropriate polynomial degree by dividing by (3.30). Substituting the definition of the edge coordinate from (3.29) into (3.31) yields

$$\phi_i^e = L_k L_e \left(\frac{4}{1 - (L_k - L_e)^2} \right) \psi_{i+1}(L_k - L_e) \quad \text{for } i = 1, 2, \dots, p_{ip}^e - 1 \quad (3.32)$$

In a two-dimensional mesh some triangles share sides, and therefore edge basis functions, with other triangles in the mesh. An example of two triangular elements sharing a side is shown in Figure 3.6. In Figure 3.6, the global node numbers are listed next to the appropriate nodes. Element one, e_1 , is formed by connecting global vertex nodes 1, 2, and 3 and element two, e_2 , is formed by connecting global vertex nodes 1, 3, and

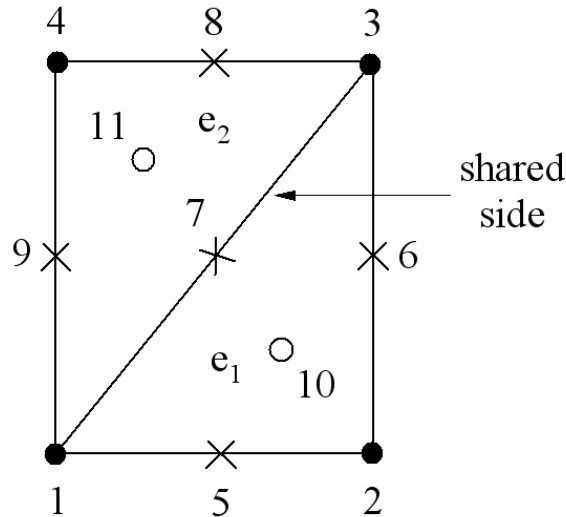


Figure 3.6: Triangular Elements Sharing a Side

4. The side shared by the two elements is located between global vertex nodes 1 and 3. When two triangular elements share a side, some of the basis functions along the shared side are discontinuous. This discontinuity occurs because the local edge coordinates, ξ_e , along the shared edge run from -1 to 1 in the opposite direction for each element. This concept is illustrated using Figure 3.6. For element one, the shared side starts at node 3 and ends at node 1. The edge coordinate on the shared side of element one varies from -1 at node 3 to 1 at node 1. For element two, the shared side runs in the opposite direction, starting at node 1 and ending at node 3. The edge coordinate on the shared side for element two varies for from -1 at node 1 to 1 at node 3. The contrasting values of the edge coordinates do not affect the even-degree edge basis functions, but odd-degree edge basis functions are affected.

To fix the discontinuity that occurs with the odd-degree edge basis functions, a simple remedy is applied. The elements are entered into the computer program via a connectivity table that specifies the element number and the global nodes that make up the element. If the global vertex node numbers spanning an edge go from a smaller number to a larger

number, then every odd-degree edge basis function along that edge is multiplied by -1 . In Figure 3.6 the odd-degree edge basis functions in element one along the sides from nodes 1 to 2 and 2 to 3 are multiplied by -1 . In element two, the odd-degree edge basis functions along the sides from nodes 1 to 3 and 3 to 4 are multiplied by -1 . Multiplying by -1 reverses the direction of the edge coordinate. By modifying the appropriate edge basis functions as described above, continuity has been restored to the basis functions for the elements in Figure 3.6. The elements have also been modified to allow sharing on any other edge with additional triangular elements.

The basis functions for the internal nodes create higher-degree variation within the interior of the element. The internal basis functions exist when the polynomial degree of the temperature approximation over an element is three or higher, $p_{ip} \geq 3$. For $p_{ip} < 3$ there are no internal nodes. Note that p_{ip} may have different values for each element in the mesh. The vertex and edge basis functions capture the variation over the edges of the triangle. In order to avoid redundancy, the interior basis functions are designed to vanish along the edges. A simple function that satisfies these requirements is given by

$$\phi_1^i = L_1 L_2 L_3 \quad (3.33)$$

such that ϕ_1^i is the interior basis function for an element with $p_{ip} = 3$. For in-plane approximations higher than degree three, additional functions are added to the set of the interior basis. In order to accomplish this, a set of local orthogonal coordinates, α and β , are defined over the interval $(-1, 1)$

$$\begin{aligned} \alpha &= L_2 - L_1 \\ \beta &= 2L_3 - 1 \end{aligned} \quad (3.34)$$

To construct the interior basis functions, consider Pascal's triangle shown in Table 3.2 where $P(\alpha)$ and $P(\beta)$ are the Legendre polynomials in terms of the interior coordinates.

where N_{ip}^v is the number of vertex basis functions, N_{ip}^{e1} is the number of edge basis functions on edge 1, N_{ip}^{e2} is the number of edge basis functions on edge 2, N_{ip}^{e3} is the number of edge basis functions on edge 3, and N_{ip}^i is the number of edge basis functions within the interior. If the same in-plane approximation degree, p_{ip} , is used for all of the edges and the interior, then the total number of in-plane basis functions for an element given by (3.37) simplifies to

$$N_{ip} = \frac{(p_{ip} + 1)(p_{ip} + 2)}{2} \quad (3.38)$$

The approximation in the in-plane direction captures the temperature distribution in the $x - y$ plane. The polynomial degree of approximation for the in-plane direction is the highest degree term needed in the approximation. The degree of each term is calculated by adding the degree in x and the degree in y for that term. For example, if the function $2x + 3y$ is to be represented exactly by the in-plane approximation, only the vertex basis functions are needed since the function is linear in x and linear in y . By contrast, if the function $2x + 4xy + 3y$ is to be represented exactly by the in-plane approximation, a second degree polynomial is needed. This is because the $4xy$ term is second degree, first degree in x plus first degree in y . A fifth degree approximation is needed to represent exactly a function such as $3x^2 + 10xy^4$, first degree in x plus fourth degree in y . A fifth degree approximation can also exactly represent the function $5x^3y^2 + 8x^5 + 7y^5$.

The three vertex nodes in Figure 3.5 have global coordinates (x_1, y_1) , (x_2, y_2) , and (x_3, y_3) . The local area coordinates are used to create local matrices and vectors. These matrices and vectors are then modified to create element specific matrices and vectors in the global context. The global coordinates are related to the local coordinates by

$$\begin{aligned} x &= x_1L_1 + x_2L_2 + x_3L_3 \\ y &= y_1L_1 + y_2L_2 + y_3L_3 \end{aligned} \quad (3.39)$$

The matrices and vectors that are calculated using the in-plane basis are contained in (2.37) through (2.40) and (2.44). The local in-plane integrals are calculated first by integrating over the local element area. These integrals are then mapped from the local domain to the global domain to create the element specific integrals. In order to accomplish this, a relation between the element specific differential area, dA , and the local element area, dA_L , is needed. The Jacobian is used to map the differential area from local to the global coordinates [14, pg. 220-1]

$$dA = | J | dA_L = | J | dL_1 dL_2 \quad (3.40)$$

where $| J |$ is the absolute value of the Jacobian, J . The Jacobian is given by

$$J = \frac{\partial(x, y)}{\partial(L_1, L_2)} = \det \begin{bmatrix} \frac{\partial x}{\partial L_1} & \frac{\partial y}{\partial L_1} \\ \frac{\partial x}{\partial L_2} & \frac{\partial y}{\partial L_2} \end{bmatrix} = \det \begin{bmatrix} x_1 - x_3 & y_1 - y_3 \\ x_2 - x_3 & y_2 - y_3 \end{bmatrix} \quad (3.41)$$

Only L_1 and L_2 are used since there are only two linearly independent area coordinates. The Jacobian is constant for any element because the transformation between local and global coordinates is linear [20]. By using the chain rule of partial differentiation a relationship between the partial derivatives with respect to local and global coordinates is established

$$\begin{Bmatrix} \frac{\partial \phi_i}{\partial L_1} \\ \frac{\partial \phi_i}{\partial L_2} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial L_1} & \frac{\partial y}{\partial L_1} \\ \frac{\partial x}{\partial L_2} & \frac{\partial y}{\partial L_2} \end{bmatrix} \begin{Bmatrix} \frac{\partial \phi_i}{\partial x} \\ \frac{\partial \phi_i}{\partial y} \end{Bmatrix} \quad (3.42)$$

The matrix in (3.42) is the same matrix from (3.41). This matrix is known as the Jacobian matrix, $[J_m]$. Solving (3.42) for the partial derivatives of the basis functions with respect to the global variables x and y yields

$$\begin{Bmatrix} \frac{\partial \phi_i}{\partial x} \\ \frac{\partial \phi_i}{\partial y} \end{Bmatrix} = [J_m]^{-1} \begin{Bmatrix} \frac{\partial \phi_i}{\partial L_1} \\ \frac{\partial \phi_i}{\partial L_2} \end{Bmatrix} \quad (3.43)$$

This relationship is used to convert the in-plane integrals in (2.44) to local integrals and allow a simple conversion from local to global coordinates.

Using (3.40) to modify the in-plane basis in (2.37) through (2.40) and (2.44) and by pulling the constant Jacobian outside the integrals, a local calculation can be made instead of an element specific calculation for several matrices

$$[M_{ip}]_e = \int_A \phi \phi^T dA = |J| \int_{A_L} \phi \phi^T dL_1 dL_2 = |J| [M_{ip}]_l \quad (3.44)$$

$$\{L_{ip}\}_e = \int_A \phi dA = |J| \int_{A_L} \phi dL_1 dL_2 = |J| \{L_{ip}\}_e \quad (3.45)$$

$$[M_{ip}^i]_e = \int_A \tilde{\phi} \phi \phi^T dA = |J| \int_{A_L} \tilde{\phi} \phi \phi^T dL_1 dL_2 = |J| [M_{ip}^i]_l \quad (3.46)$$

where $[M_{ip}]$ is the in-plane mass matrix, $\{L_{ip}\}$ is the in-plane load vector, and $[M_{ip}^i]$ is the in-plane mass matrix for the i^{th} in-plane interpolation function. Again, the subscripts e and l denote element specific and local respectively. The local matrices and vectors are computed first and the element specific matrices and vectors are created by multiplying the local matrices and vectors by the Jacobian.

The in-plane stiffness matrix for the i^{th} in-plane interpolation function from (2.44) is

$$[K_{ip}^i]_e = \int_A \tilde{\phi}_i \left[\frac{\partial \phi}{\partial x} \frac{\partial \phi^T}{\partial x} + \frac{\partial \phi}{\partial y} \frac{\partial \phi^T}{\partial y} \right] dA \quad (3.47)$$

A transformation from global coordinates to local coordinates for the i^{th} in-plane stiffness matrix is needed. First, the inverse Jacobian matrix, $[J_m]^{-1}$ from (3.43) is written in a convenient form as

$$[J_m]^{-1} = \begin{bmatrix} \alpha_x & \rho_x \\ \alpha_y & \rho_y \end{bmatrix} \quad (3.48)$$

Using (3.48) to transform from global to local coordinates, the two symmetric matrices from (3.47) are written as

$$\begin{aligned} \left[\frac{\partial \phi}{\partial x} \frac{\partial \phi^T}{\partial x} \right] &= \left[\left(\alpha_x \frac{\partial \phi}{\partial L_1} + \rho_x \frac{\partial \phi}{\partial L_2} \right) \left(\alpha_x \frac{\partial \phi^T}{\partial L_1} + \rho_x \frac{\partial \phi^T}{\partial L_2} \right) \right] \\ \left[\frac{\partial \phi}{\partial y} \frac{\partial \phi^T}{\partial y} \right] &= \left[\left(\alpha_y \frac{\partial \phi}{\partial L_1} + \rho_y \frac{\partial \phi}{\partial L_2} \right) \left(\alpha_y \frac{\partial \phi^T}{\partial L_1} + \rho_y \frac{\partial \phi^T}{\partial L_2} \right) \right] \end{aligned} \quad (3.49)$$

Expanding (3.49)

$$\begin{aligned} \left[\frac{\partial \phi}{\partial y} \frac{\partial \phi}{\partial y}^T \right] &= \left[\left(\alpha_x^2 + \alpha_y^2 \right) \frac{\partial \phi}{\partial L_1} \frac{\partial \phi}{\partial L_1}^T + (\alpha_x \rho_x + \alpha_y \rho_y) \frac{\partial \phi}{\partial L_1} \frac{\partial \phi}{\partial L_2}^T \right. \\ &\quad \left. + (\alpha_x \rho_x + \alpha_y \rho_y) \frac{\partial \phi}{\partial L_2} \frac{\partial \phi}{\partial L_1}^T + \left(\rho_x^2 + \rho_y^2 \right) \frac{\partial \phi}{\partial L_2} \frac{\partial \phi}{\partial L_2}^T \right] \end{aligned} \quad (3.50)$$

Substituting (3.50) and (3.40) into (3.47), the i^{th} in-plane stiffness matrix is written as [9, pg.53-54]

$$\begin{aligned} [K_{ip}^i]_e &= \int_{A_L} \left[\left(\alpha_x^2 + \alpha_y^2 \right) \tilde{\phi}_i \frac{\partial \phi}{\partial L_1} \frac{\partial \phi}{\partial L_1}^T + (\alpha_x \rho_x + \alpha_y \rho_y) \tilde{\phi}_i \frac{\partial \phi}{\partial L_1} \frac{\partial \phi}{\partial L_2}^T + \right. \\ &\quad \left. (\alpha_x \rho_x + \alpha_y \rho_y) \tilde{\phi}_i \frac{\partial \phi}{\partial L_2} \frac{\partial \phi}{\partial L_1}^T + \left(\rho_x^2 + \rho_y^2 \right) \tilde{\phi}_i \frac{\partial \phi}{\partial L_2} \frac{\partial \phi}{\partial L_2}^T \right] |J| dA_L \end{aligned} \quad (3.51)$$

The terms α_x , α_y , ρ_x , and ρ_y are taken from the inverse Jacobian matrix and are constant over each element. Since these terms and the Jacobian are constant over the element, they are placed outside the integral and (3.51) becomes

$$\begin{aligned} [K_{ip}^i]_e &= |J| \left[\left(\alpha_x^2 + \alpha_y^2 \right) \int_{A_L} \tilde{\phi}_i \frac{\partial \phi}{\partial L_1} \frac{\partial \phi}{\partial L_1}^T dA_L + (\alpha_x \rho_x + \alpha_y \rho_y) \int_{A_L} \tilde{\phi}_i \frac{\partial \phi}{\partial L_1} \frac{\partial \phi}{\partial L_2}^T dA_L \right. \\ &\quad \left. + (\alpha_x \rho_x + \alpha_y \rho_y) \int_{A_L} \tilde{\phi}_i \frac{\partial \phi}{\partial L_2} \frac{\partial \phi}{\partial L_1}^T dA_L + \left(\rho_x^2 + \rho_y^2 \right) \int_{A_L} \tilde{\phi}_i \frac{\partial \phi}{\partial L_2} \frac{\partial \phi}{\partial L_2}^T dA_L \right] \end{aligned} \quad (3.52)$$

The i^{th} local in-plane stiffness matrices from (3.52) are

$$[K_{ip}^i]_l^1 = \int_{A_L} \tilde{\phi}_i \frac{\partial \phi}{\partial L_1} \frac{\partial \phi}{\partial L_1}^T dA_L \quad (3.53)$$

$$[K_{ip}^i]_l^2 = \int_{A_L} \tilde{\phi}_i \frac{\partial \phi}{\partial L_1} \frac{\partial \phi}{\partial L_2}^T dA_L \quad (3.54)$$

$$[K_{ip}^i]_l^3 = \int_{A_L} \tilde{\phi}_i \frac{\partial \phi}{\partial L_2} \frac{\partial \phi}{\partial L_1}^T dA_L \quad (3.55)$$

$$[K_{ip}^i]_l^4 = \int_{A_L} \tilde{\phi}_i \frac{\partial \phi}{\partial L_2} \frac{\partial \phi}{\partial L_2}^T dA_L \quad (3.56)$$

where the superscripts 1, 2, 3, and 4 denote the four different local in-plane stiffness matrices. The local in-plane stiffness matrices from (3.53) through (3.56) are calculated first. The i^{th} element specific in-plane stiffness matrix, $[K_{ip}^i]_e$, is calculated by substituting the local in-plane stiffness matrices from (3.53) through (3.56) into (3.52).

Chapter 4

Interpolation of Thermal Conductivity

The thermal conductivity of a material may vary greatly with temperature. This is especially true when a material is exposed to the broad temperature ranges that are commonly encountered in aerospace applications. The thermal conductivity of real materials is usually experimentally determined and tabulated for different temperatures. This chapter explains the method used to account for the variation of the thermal conductivity as a function of temperature.

To account for the variation in the thermal conductivity, an iterative process is used. The approximate temperature distribution from the previous iteration is used to create a polynomial that mimics the behavior of the thermal conductivity over each element. A set of sampling points on each element are used to represent the temperature distribution over the element. Using the values of temperature at these sampling points and the tabulated values of thermal conductivity, a polynomial is created to represent the variation in the thermal conductivity over an element.

The polynomial representing the variation of the thermal conductivity over each element is created using interpolation. Interpolation is frequently used to fit a polynomial to data using a set number of points. These points are called sampling points and are used

to represent a larger set of data. The interpolant is equal to the values of the thermal conductivity at the sampling points and provides a smooth polynomial distribution in between the sampling points. By using an appropriate number of sampling points, the interpolant approximates the variation in the thermal conductivity over an element.

The interpolant is formed from a set of interpolation functions that are multiplied by scaling factors and then summed. The thermal conductivity over an element is defined by (2.33)

$$\hat{k}_e = \tilde{\chi}^T \mathbf{b}_{n,e} = \left(\tilde{\phi} \otimes \tilde{\theta} \otimes \tilde{\psi} \right)^T \mathbf{b}_{n,e} \quad (4.1)$$

where $\tilde{\phi}$, $\tilde{\theta}$, $\tilde{\psi}$ are vectors that represent the interpolation functions over the in-plane, time, and through-thickness dimensions respectively and $\mathbf{b}_{n,e}$ is a vector of interpolation constants that is calculated from $\mathbf{a}_{n,e}$.

4.1 Lagrange Interpolation Functions

The Lagrange polynomials are used as interpolation functions in this research. An important property of the Lagrange interpolation functions is that they are unity at their node and zero at all other nodes. The interpolation functions over time and in the through-thickness direction are each a set of one-dimensional Lagrange polynomials. The interpolation functions for the in-plane dimension are a two-dimensional set of Lagrange polynomials. The in-plane and through-thickness interpolation functions are the same functions used by Walker [9].

4.1.1 Time Interpolation Functions

One-dimensional Lagrange interpolation functions are used to approximate the variation in the thermal conductivity over time. The degree of polynomial approximation for the thermal conductivity over the time dimension is denoted by r_t . To create the time

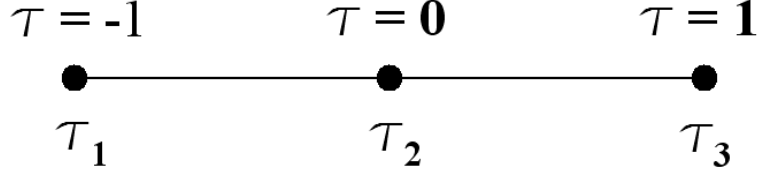


Figure 4.1: Sample Time Lagrange Element

interpolation functions, $\tilde{N}_t = r_t + 1$ evenly spaced nodes are placed over the time interval.

Then, the Lagrange interpolation functions for the time dimension are

$$\tilde{\theta}_j = \prod_{\substack{n=1 \\ n \neq j}}^{\tilde{N}_t} \left(\frac{\tau - \tau_n}{\tau_j - \tau_n} \right) \quad (4.2)$$

where $\tilde{\theta}_j$ is the j^{th} interpolation function corresponding to node j , \tilde{N}_t is the number of time Lagrange interpolation functions, τ is the localized time variable, τ_n is the location of the n^{th} node in terms of the local time coordinate, and τ_j is the location of the j^{th} node in terms of the local time coordinate. There is one interpolation function for each evenly-spaced node for a total of \tilde{N}_t time interpolation functions.

If a second degree polynomial approximation, $r_t = 2$, is used then there are three evenly-spaced nodes for the time Lagrange element such that $\tilde{N}_t = 3$. The Lagrange time element for this interpolation is shown in Figure 4.1. The time Lagrange interpolation functions for this sample element are given by

$$\begin{aligned} \tilde{\theta}_1 &= \left(\frac{\tau - \tau_2}{\tau_1 - \tau_2} \right) \left(\frac{\tau - \tau_3}{\tau_1 - \tau_3} \right) = \left(\frac{\tau - 0}{-1 - 0} \right) \left(\frac{\tau - 1}{-1 - 1} \right) = \frac{1}{2} \tau (\tau - 1) \\ \tilde{\theta}_2 &= \left(\frac{\tau - \tau_1}{\tau_2 - \tau_1} \right) \left(\frac{\tau - \tau_3}{\tau_2 - \tau_3} \right) = \left(\frac{\tau - (-1)}{0 - (-1)} \right) \left(\frac{\tau - 1}{0 - 1} \right) = -\tau^2 + 1 \\ \tilde{\theta}_3 &= \left(\frac{\tau - \tau_1}{\tau_3 - \tau_1} \right) \left(\frac{\tau - \tau_2}{\tau_3 - \tau_2} \right) = \left(\frac{\tau - (-1)}{1 - (-1)} \right) \left(\frac{\tau - 0}{1 - 0} \right) = \frac{1}{2} \tau (\tau + 1) \end{aligned} \quad (4.3)$$

Each time Lagrange interpolation function results in a separate time mass matrix by substituting one function at a time into (3.13). This means that there are \tilde{N}_t time mass matrices, one for each time Lagrange interpolation function.

4.1.2 Through-Thickness Interpolation Functions

The same interpolation functions used for the time dimension are used for the through-thickness dimension. The Lagrange interpolation functions for the through-thickness dimension are

$$\tilde{\psi}_k = \prod_{\substack{n=1 \\ n \neq k}}^{\tilde{N}_{tk}} \left(\frac{\eta - \eta_n}{\eta_k - \eta_n} \right) \quad (4.4)$$

where $\tilde{\psi}_k$ is the k^{th} interpolation function corresponding to node k , \tilde{N}_{tk} is the number of through-thickness Lagrange interpolation functions, η is the localized through-thickness variable, η_n is the location of the n^{th} node in terms of the local through-thickness coordinate, and η_k is the location of the k^{th} node in terms of the local through-thickness coordinate. There is one interpolation function at each evenly-spaced node for a total of \tilde{N}_{tk} through-thickness interpolation functions. Even though the same interpolation functions are used for the time and through-thickness interpolation, the degree of the polynomial interpolant may be different and the functions are completely independent from one another in the finite element formulation.

The through-thickness Lagrange interpolation functions each result in a separate mass and stiffness matrix by substituting one function at a time into (3.22) and (3.23). There are \tilde{N}_{tk} through-thickness mass matrices and \tilde{N}_{tk} through-thickness stiffness matrices, one for each through-thickness Lagrange interpolation function.

4.1.3 In-Plane Interpolation Functions

Two-dimensional Lagrange interpolation functions for a triangular element are used to approximate the variation in the thermal conductivity over the in-plane direction of an element. The in-plane degree of polynomial approximation for the thermal conductivity is denoted by r_{ip} . The number of terms needed to completely represent a polynomial of degree r_{ip} can be determined from Pascal's triangle shown in Table 4.1. To create

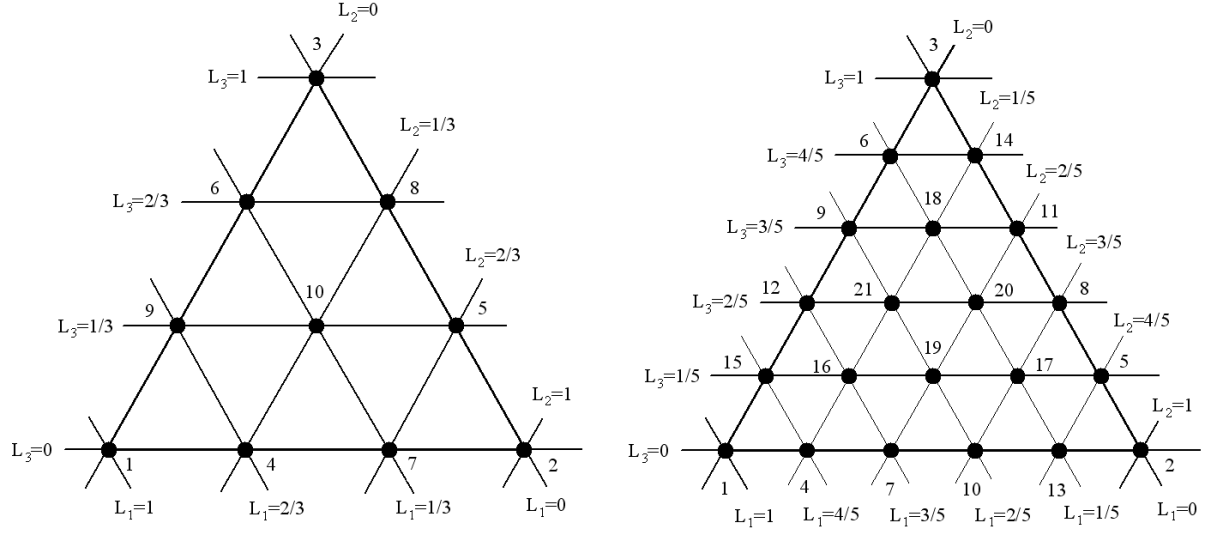


Figure 4.2: In-Plane Lagrange Nodal Positions for $r_{ip} = 3$ (left) and $r_{ip} = 5$ (right)

a product of r_{ip} terms. To determine the terms in the product, a set of lines, each representing a fixed value for one area coordinate, is drawn (as shown in Figure 4.2) to specify graphically the location of any in-plane Lagrange node. Every Lagrange node lies at the intersection of three lines specifying the values of the three area coordinates at that node. Using the values of the three area coordinates at the i^{th} node and the product operator, the in-plane Lagrange interpolation functions are given by

$$\tilde{\phi}_i = \prod_j^n \prod_{k=1}^m \left(\frac{L_j - L_{j_k}}{L_{j_p} - L_{j_k}} \right) \quad i = 1, \dots, \tilde{N}_{ip} \quad (4.6)$$

where j is the index of a nonzero area coordinate for the i^{th} node, n is the index of the last nonzero area coordinate, L_j is the j^{th} nonzero area coordinate, L_{j_p} is the value of L_j at the line passing through the i^{th} node, m is the number of values along a line for L_j that are less than L_{j_p} , and L_{j_k} is a value of L_j along a line that is less than L_{j_p} .

As an example, consider node ten for $r_{ip} = 3$ in Figure 4.2. The values of the three area coordinates at node ten are $L_1 = L_2 = L_3 = \frac{1}{3}$, therefore $j = 1$ and $n = 3$. Since the values of the three area coordinates are the same, $L_{1_p} = L_{2_p} = L_{3_p} = \frac{1}{3}$. Then, because

all of the area coordinates have the same value, $m = 1$ for $j = 1$, $j = 2$, and $j = 3$. Finally, $L_{j_k} = L_{j_1} = 0$ for all values of j since the only value of L_j along a line that is less than $L_{j_p} = \frac{1}{3}$ is 0. The in-plane Lagrange interpolation function for node ten is

$$\tilde{\phi}_{10} = \prod_{j=1}^3 \prod_{k=1}^1 \left(\frac{L_j - L_{j_k}}{L_{j_p} - L_{j_k}} \right) = \left(\frac{L_1 - 0}{\frac{1}{3} - 0} \right) \left(\frac{L_2 - 0}{\frac{1}{3} - 0} \right) \left(\frac{L_3 - 0}{\frac{1}{3} - 0} \right) \quad (4.7)$$

Consider nodes one, seven, and sixteen for $r_{ip} = 5$ in Figure 4.2. The in-plane Lagrange interpolation functions for these nodes are

$$\begin{aligned} \tilde{\phi}_1 &= \left(\frac{L_1 - 0}{1 - 0} \right) \left(\frac{L_1 - \frac{1}{5}}{1 - \frac{1}{5}} \right) \left(\frac{L_1 - \frac{2}{5}}{1 - \frac{2}{5}} \right) \left(\frac{L_1 - \frac{3}{5}}{1 - \frac{3}{5}} \right) \left(\frac{L_1 - \frac{4}{5}}{1 - \frac{4}{5}} \right) \\ \tilde{\phi}_7 &= \left(\frac{L_1 - 0}{\frac{3}{5} - 0} \right) \left(\frac{L_1 - \frac{1}{5}}{\frac{3}{5} - \frac{1}{5}} \right) \left(\frac{L_1 - \frac{2}{5}}{\frac{3}{5} - \frac{2}{5}} \right) \left(\frac{L_2 - 0}{\frac{2}{5} - 0} \right) \left(\frac{L_2 - \frac{1}{5}}{\frac{2}{5} - \frac{1}{5}} \right) \\ \tilde{\phi}_{16} &= \left(\frac{L_1 - 0}{\frac{3}{5} - 0} \right) \left(\frac{L_1 - \frac{1}{5}}{\frac{3}{5} - \frac{1}{5}} \right) \left(\frac{L_1 - \frac{2}{5}}{\frac{3}{5} - \frac{2}{5}} \right) \left(\frac{L_2 - 0}{\frac{1}{5} - 0} \right) \left(\frac{L_3 - 0}{\frac{1}{5} - 0} \right) \end{aligned} \quad (4.8)$$

By using symmetry, the interpolation functions in (4.8) are modified to create the interpolation functions for nodes 2, 8, and 17 by replacing L_1, L_2, L_3 with L_2, L_3, L_1 respectively [9, pg. 38]

$$\begin{aligned} \tilde{\phi}_2 &= \left(\frac{L_2 - 0}{1 - 0} \right) \left(\frac{L_2 - \frac{1}{5}}{1 - \frac{1}{5}} \right) \left(\frac{L_2 - \frac{2}{5}}{1 - \frac{2}{5}} \right) \left(\frac{L_2 - \frac{3}{5}}{1 - \frac{3}{5}} \right) \left(\frac{L_2 - \frac{4}{5}}{1 - \frac{4}{5}} \right) \\ \tilde{\phi}_8 &= \left(\frac{L_2 - 0}{\frac{3}{5} - 0} \right) \left(\frac{L_2 - \frac{1}{5}}{\frac{3}{5} - \frac{1}{5}} \right) \left(\frac{L_2 - \frac{2}{5}}{\frac{3}{5} - \frac{2}{5}} \right) \left(\frac{L_3 - 0}{\frac{2}{5} - 0} \right) \left(\frac{L_3 - \frac{1}{5}}{\frac{2}{5} - \frac{1}{5}} \right) \\ \tilde{\phi}_{17} &= \left(\frac{L_2 - 0}{\frac{3}{5} - 0} \right) \left(\frac{L_2 - \frac{1}{5}}{\frac{3}{5} - \frac{1}{5}} \right) \left(\frac{L_2 - \frac{2}{5}}{\frac{3}{5} - \frac{2}{5}} \right) \left(\frac{L_3 - 0}{\frac{1}{5} - 0} \right) \left(\frac{L_1 - 0}{\frac{1}{5} - 0} \right) \end{aligned} \quad (4.9)$$

By using symmetry again, the interpolation functions for nodes 2, 8, and 17 are modified to create the interpolation functions for nodes 3, 9, and 18 by replacing L_1, L_2, L_3 with L_2, L_3, L_1 respectively

$$\begin{aligned} \tilde{\phi}_3 &= \left(\frac{L_3 - 0}{1 - 0} \right) \left(\frac{L_3 - \frac{1}{5}}{1 - \frac{1}{5}} \right) \left(\frac{L_3 - \frac{2}{5}}{1 - \frac{2}{5}} \right) \left(\frac{L_3 - \frac{3}{5}}{1 - \frac{3}{5}} \right) \left(\frac{L_3 - \frac{4}{5}}{1 - \frac{4}{5}} \right) \\ \tilde{\phi}_9 &= \left(\frac{L_3 - 0}{\frac{3}{5} - 0} \right) \left(\frac{L_3 - \frac{1}{5}}{\frac{3}{5} - \frac{1}{5}} \right) \left(\frac{L_3 - \frac{2}{5}}{\frac{3}{5} - \frac{2}{5}} \right) \left(\frac{L_1 - 0}{\frac{2}{5} - 0} \right) \left(\frac{L_1 - \frac{1}{5}}{\frac{2}{5} - \frac{1}{5}} \right) \\ \tilde{\phi}_{18} &= \left(\frac{L_3 - 0}{\frac{3}{5} - 0} \right) \left(\frac{L_3 - \frac{1}{5}}{\frac{3}{5} - \frac{1}{5}} \right) \left(\frac{L_3 - \frac{2}{5}}{\frac{3}{5} - \frac{2}{5}} \right) \left(\frac{L_1 - 0}{\frac{1}{5} - 0} \right) \left(\frac{L_2 - 0}{\frac{1}{5} - 0} \right) \end{aligned} \quad (4.10)$$

Each in-plane Lagrange interpolation function results in a separate mass and stiffness matrix by substituting one function at a time into (3.46) and (3.52). Therefore there are \tilde{N}_{ip} in-plane mass matrices and \tilde{N}_{ip} in-plane stiffness matrices for each element, one for each in-plane Lagrange interpolation function.

4.2 Optimization of Sampling Points

A sampling point is defined by a set of local coordinates specifying the location of the in-plane, time, and through-thickness coordinates in an element. The choice of sampling points has a substantial effect on the quality of the interpolant that approximates the variation in the thermal conductivity over an element. Using evenly spaced sampling points often results in Runge's Phenomenon, where the resulting interpolant oscillates wildly near the ends of the interval [21, pg. 295]. To minimize this oscillation, the Chebyshev points are biased toward the ends of the one-dimensional element [9, pg. 42] and the Fekete points are biased towards the vertices of the triangle [23]. To reduce Runge's Phenomenon, the Chebyshev and Fekete sampling points used by Walker [9] are also used in this research.

The zeros of the Chebyshev polynomial are the optimal set of sampling points for a one-dimensional element in the sense that they minimize the error of the interpolant in the least squares sense. The coordinate of the k^{th} Chebyshev point for an interpolant of degree r_{tk} is determined from

$$\eta_k = \cos \left(\frac{\left(k - \frac{1}{2}\right) \pi}{r_{tk} - 1} \right) \quad k = 1, 2, \dots, r_{tk} + 1 \quad (4.11)$$

such that each η_k is a zero of the Chebyshev polynomial of degree $r_{tk} + 1$ [22]. The Chebyshev points are used for the time interpolation in addition to the through-thickness interpolation.

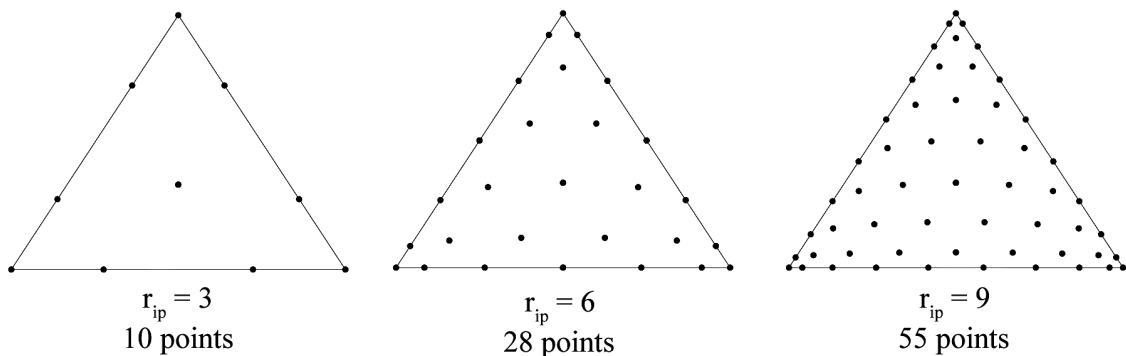


Figure 4.3: Fekete Points for In-Plane Degrees of Interpolation $r_{ip} = 3, 6, 9$

A proven optimal arrangement of sampling points for a two-dimensional triangle does not exist, so the nearly optimal Fekete point distribution is used. There is no analytic formula for the location of the Fekete points in a triangle, so they are calculated numerically [23, pg. 1707]. The number of Fekete points needed for an interpolation of degree r_{ip} is equal to the number of in-plane interpolation functions given by (4.5). Figure 4.3 shows the Fekete points for in-plane degrees of interpolation of 3, 6, and 9.

4.3 Determination of Interpolation Coefficients

In order to calculate the total stiffness matrix for an element, the vector of interpolation coefficients for that element, $\mathbf{b}_{n,e}$, must be determined. These interpolation coefficients were first used in (2.33) shown below.

$$k_e(\hat{u}_{n,e}) \approx \hat{k}_e = \sum_{i=1}^{\tilde{N}_{ip}} \sum_{j=1}^{\tilde{N}_t} \sum_{k=1}^{\tilde{N}_{tk}} \tilde{\phi}_i \tilde{\theta}_j \tilde{\psi}_k b_{ijk} = \tilde{\chi}^T \mathbf{b}_{n,e} = (\tilde{\phi} \otimes \tilde{\theta} \otimes \tilde{\psi})^T \mathbf{b}_{n,e}$$

To determine the interpolation coefficients, the temperature distribution over the element must be known. In the time-stepping solution method presented in Section 2.5, the approximate temperature distribution over the finite element mesh is known for the previous time step. For the first time step, the temperature distribution for the previous time step is given by the initial condition. The approximate temperature distribution

over an element is given in (2.27) and the approximate temperature distribution over an element at the previous time step is given by

$$\hat{u}_{n-1,e} = \chi^T \mathbf{a}_{n-1,e} = (\phi \otimes \theta \otimes \psi)^T \mathbf{a}_{n-1,e} \quad (4.12)$$

By evaluating the basis functions at the chosen sampling point in local coordinates, the value of the temperature approximation at a sampling point is calculated

$$\hat{u}_l^{sp} = \chi^T |_l^{sp} \mathbf{a}_{n-1,e} = \left(\phi |_{l_{ip}}^{sp} \otimes \theta |_{l_t}^{sp} \otimes \psi |_{l_{tk}}^{sp} \right)^T \mathbf{a}_{n-1,e} \quad (4.13)$$

where \hat{u}_l^{sp} is the value of the temperature approximation at the l^{th} sampling point, $|_l^{sp}$ denotes the evaluation of each of the basis functions at the location of the sampling point in their dimension, $|_{l_{ip}}^{sp}$ denotes evaluation at the l^{th} sampling point coordinates in the in-plane direction, $|_{l_t}^{sp}$ denotes evaluation at the l^{th} sampling point coordinate in the time direction, and $|_{l_{tk}}^{sp}$ denotes evaluation at the l^{th} sampling point coordinate in the through-thickness direction.

Once the values of the temperature for all of the sampling points on an element have been calculated, the values of the thermal conductivity are then determined. The thermal conductivity is tabulated as a function of temperature. If the temperature for a sampling point is given in the table, the thermal conductivity is simply the value of the thermal conductivity from the table. When the temperature lies between two entries in the table, the thermal conductivity is linearly interpolated

$$\hat{k}_l = \frac{\hat{u}_l^{sp} - u_2}{u_1 - u_2} k_1 + \frac{\hat{u}_l^{sp} - u_1}{u_2 - u_1} k_2 \quad (4.14)$$

where \hat{k}_l is the value of thermal conductivity at the l^{th} sampling point, \hat{u}_l^{sp} is the value of the temperature approximation at the l^{th} sampling point, u_1 and u_2 are the two nearest values to \hat{u}_{sp} from the table, and k_1 and k_2 are the corresponding table entries for the thermal conductivity. By using the values of the temperature approximation at each of

the sampling points, tabulated values of thermal conductivity, and (4.14), the values of the thermal conductivity are determined for every sampling point [9, pg. 40-41].

The interpolant approximating the thermal conductivity is required to be equal to the values of the thermal conductivity at every sampling point. The thermal conductivity at the l^{th} sampling point is written as

$$\hat{k}_l = \tilde{\chi}^T |_l^{sp} \mathbf{b}_{n,e} = \left(\tilde{\phi} |_{l_{ip}}^{sp} \otimes \tilde{\theta} |_{l_t}^{sp} \otimes \tilde{\psi} |_{l_{tk}}^{sp} \right)^T \mathbf{b}_{n,e} \quad l = 1, 2, \dots, \tilde{N}_b \quad (4.15)$$

where \hat{k}_l is the value of the thermal conductivity at the sampling point l , $\tilde{\chi} |_l^{sp}$ is a column vector of the interpolation functions evaluated at the l^{th} sampling point, and $\mathbf{b}_{n,e}$ is the column vector of interpolation coefficients for that element. The number of interpolation functions and therefore the number of sampling points for an element is \tilde{N}_b from (2.34). Evaluating (4.15) at every sampling point results in a set of \tilde{N}_b equations. In matrix form, these equations are written

$$\{\hat{k}\} = [P] \mathbf{b}_{n,e} \quad (4.16)$$

where $\{\hat{k}\}$ is a column vector of the thermal conductivity values at the sampling points and $[P]$ is a square matrix of length \tilde{N}_b such that each row is a vector of interpolation functions evaluated at a single sampling point. The thermal conductivity values, $\{\hat{k}\}$, and the square matrix, $[P]$, are known and the interpolation coefficients are solved for

$$\mathbf{b}_{n,e} = [P]^{-1} \{\hat{k}\} \quad (4.17)$$

By solving (4.17) for every element in the mesh, all of the interpolation coefficients are calculated. The same procedure is used to calculate the interpolation coefficients regardless of whether the temperature approximation used is from the previous time step or the previous iteration in the iterative solver.

The Chebyshev and Fekete sampling points become tightly packed for higher degrees of interpolation. This may cause $[P]$ to become ill-conditioned and highly susceptible to

computer round-off error. To remedy this problem, $[P]$ is reconditioned by multiplying both sides of (4.16) by $[P]^T$ [9, pg. 42-43]

$$[P]^T \{\hat{k}\} = [P]^T [P] \mathbf{b}_{n,e} \quad (4.18)$$

For higher degrees of interpolation, the interpolation coefficients are calculated by solving (4.18) for $\mathbf{b}_{n,e}$

$$\mathbf{b}_{n,e} = \left[[P]^T [P] \right]^{-1} [P]^T \{\hat{k}\} \quad (4.19)$$

4.4 Convergence Criteria for Iteration

In order to use the time-stepping solution method discussed in Section 2.5, an appropriate method of determining whether the solution has converged is needed for the iterative solver. The convergence criterion used by Walker [9] is extended here to include transient problems. The normalized difference of the $L^2(I_n, L^2(\Omega))$ norms in space and time of two successive iterations, \hat{u}_n^m and \hat{u}_n^{m-1} , are used to assess convergence

$$C = \frac{\| \hat{u}_n^m \|_{L^2(I_n, L^2(\Omega))} - \| \hat{u}_n^{m-1} \|_{L^2(I_n, L^2(\Omega))}}{\| \hat{u}_n^m \|_{L^2(I_n, L^2(\Omega))}} \quad (4.20)$$

where C is the value of the convergence criterion and \hat{u}_n^m and \hat{u}_n^{m-1} are determined from \mathbf{a}_n^m and \mathbf{a}_n^{m-1} respectively. The first time through the iterative solver, \hat{u}_{n-1} is used instead of \hat{u}_n^{m-1} and is determined from \mathbf{a}_{n-1} . This convergence criterion measures the difference in the magnitudes of the temperature approximations from one iteration to the next. A user defined tolerance is set and when C is less than the tolerance, convergence is achieved and the solution method proceeds.

The $L^2(I_n, L^2(\Omega))$ norm of the temperature approximation in space and time is written as

$$\| \hat{u}_n \|_{L^2(I_n, L^2(\Omega))} = \sqrt{\int_{t_{n-1}}^{t_n} \int_{\Omega} (\hat{u}_n)^2 d\Omega dt} \quad (4.21)$$

Substituting for \hat{u}_n defined in (2.17) and considering all of the elements at a time step

$$\| \hat{u}_n \|_{L^2(I_n, L^2(\Omega))} = \sum_{e=1}^{N_e} \sqrt{\mathbf{a}_{n,e}^T \left[\int_{t_{n-1}}^{t_n} \int_{\Omega_e} \chi \chi^T d\Omega dt \right] \mathbf{a}_{n,e}} \quad (4.22)$$

then by applying (2.27) and reorganizing,

$$\| \hat{u}_n \|_{L^2(I_n, L^2(\Omega))} = \sum_{e=1}^{N_e} \sqrt{\mathbf{a}_{n,e}^T \left[\int_{A_e} \phi \phi^T dA_e \otimes \int_{t_{n-1}}^{t_n} \theta \theta^T dt \int_{-\frac{d}{2}}^{\frac{d}{2}} \psi \psi^T dz \right] \mathbf{a}_{n,e}} \quad (4.23)$$

Note that the in-plane and through-thickness matrices shown here have already been calculated to form $[C_e]$. The time mass matrix can be calculated by using a property of the Lagrange interpolation functions. The sum of the Lagrange interpolation functions at any point on the domain is one. This allows the time mass matrix to be calculated by summing over all of the time mass matrices from (3.13)

$$[M_t]_e = \int_{t_{n-1}}^{t_n} \theta \theta^T dt = \frac{\Delta t}{2} \sum_{j=1}^{\tilde{N}_t} \int_{-1}^1 \tilde{\theta}_j \theta \theta^T d\tau = \frac{\Delta t}{2} \sum_{j=1}^{\tilde{N}_t} [M_t^j]_l \quad (4.24)$$

This concept also applies to the in-plane and through-thickness matrices. Any element matrix can be created by summing over all of the corresponding matrices containing Lagrange interpolation functions.

Finally, by using the element matrices from (3.20), (3.44), and (4.24), the two different terms in (4.20) are calculated using matrices that have already been determined

$$\| \hat{u}_n^m \|_{L^2(I_n, L^2(\Omega))} = \sum_{e=1}^{N_e} \sqrt{(\mathbf{a}_{n,e}^m)^T \left[[M_{ip}]_e \otimes [M_t]_e \otimes [M_{tk}]_e \right] \mathbf{a}_{n,e}^m} \quad (4.25)$$

$$\| \hat{u}_n^{m-1} \|_{L^2(I_n, L^2(\Omega))} = \sum_{e=1}^{N_e} \sqrt{(\mathbf{a}_{n,e}^{m-1})^T \left[[M_{ip}]_e \otimes [M_t]_e \otimes [M_{tk}]_e \right] \mathbf{a}_{n,e}^{m-1}} \quad (4.26)$$

Chapter 5

Computer Implementation

The finite element formulation presented in Chapter 2 is implemented in a FORTRAN program to validate the method. The program is entitled p-Version Nonlinear Discontinuous Galerkin Structurally Compatible Heat Transfer (ndgSCHTp). The original program was created by Dr. Kim S. Bey at NASA's Langley Research Center and was entitled Structurally Compatible Heat Transfer (SCHT) [24]. SCHT used traditional linear triangular elements in the in-plane direction combined with hierarchical elements in the through thickness direction. SCHT was capable of performing transient or steady-state conduction heat transfer in two or three-dimensions. Tomey modified SCHT to include hierarchical p-version functions in the in-plane direction. This version was entitled dgSCHTp and could be used for layered materials with constant anisotropic properties [7]. Walker modified dgSCHTp to analyze nonlinear steady-state problems with thermal conductivity varying as a function of temperature. Walker's version was entitled nSCHTp, for p-version Nonlinear Structurally Compatible Heat Transfer, and did not allow layered or anisotropic materials [9].

The current version (ndgSCHTp) uses p-version elements in space and time and accounts for variations in the thermal conductivity as a function of temperature. The discontinuous Galerkin method is used in time. The operational order for ndgSCHTp with the modifications made for this research is shown in Figure 5.1. Currently the code

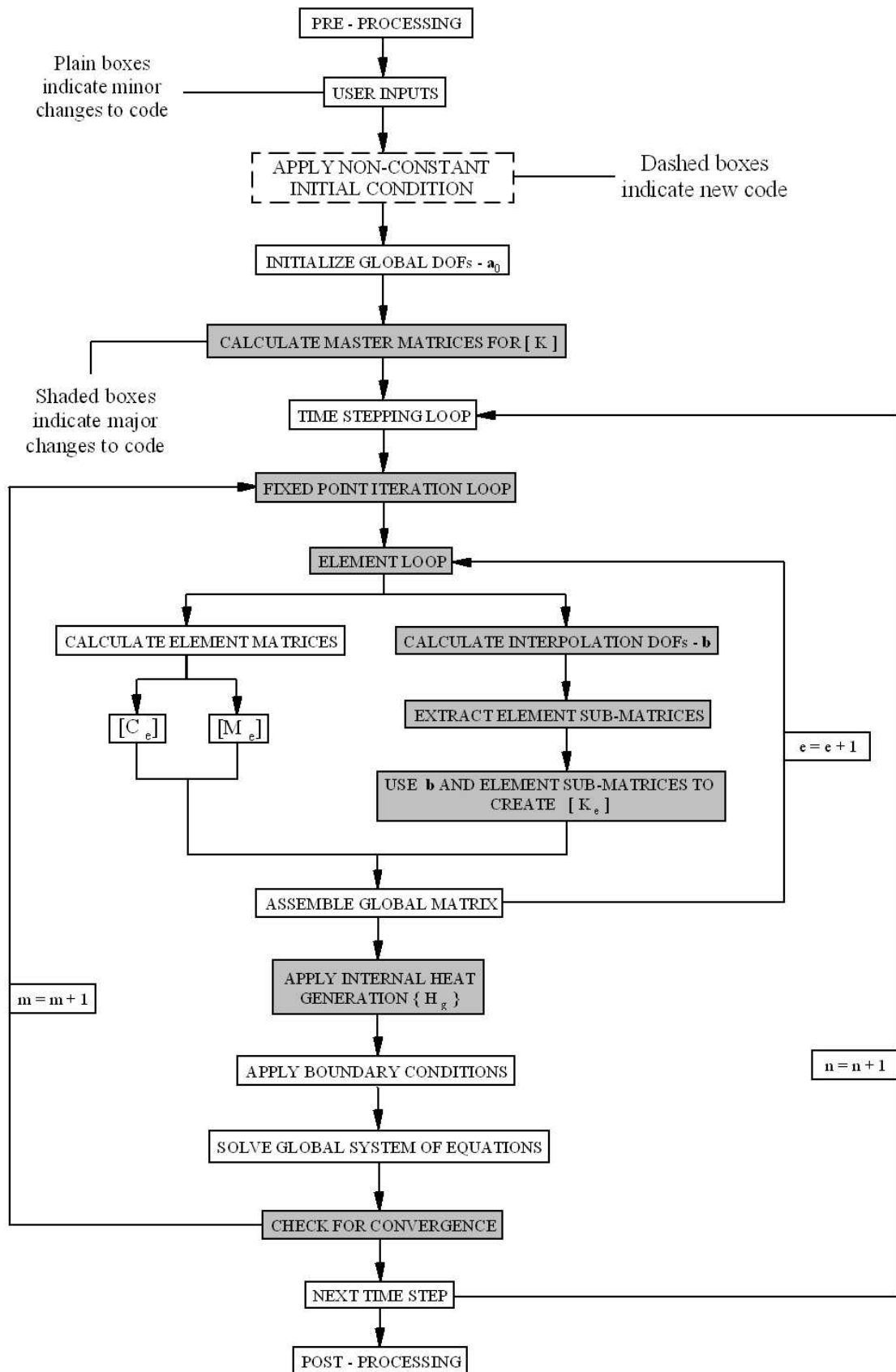


Figure 5.1: Operational Order of ndgSCHTp

only allows single isotropic materials, but the formulation could be extended to layered anisotropic materials.

5.1 Integration

Numerous integrals must be evaluated to calculate the matrices and vectors from the finite element formulation. Several techniques are used to evaluate the integrals as efficiently and accurately as possible.

The capacitance matrix, $[C_e]$, and mass matrices, $[M_e^{++}]$ and $[M_e^{-+}]$, in (2.37) through (2.39) contain integrals involving the in-plane, time, and through-thickness basis functions. The in-plane mass matrix is integrated using the symmetrical Gaussian quadrature method that utilizes area coordinates and is more efficient than the Gaussian product method [25]. The time convective matrix, $[V_t]_l$ in (3.9), is integrated using one-dimensional Gauss-Legendre numerical quadrature [13, pg. 254-8]. The through-thickness mass matrix, $[M_{tk}]_l$ in (3.20), is integrated analytically since the integrand is a simple polynomial. The exact integrals for the through-thickness mass matrices were hard coded into ndgSCHTp.

The stiffness matrix, $[K_e(\mathbf{a}_{n,e})]$, in (2.44) contains in-plane, time, and through-thickness integrations involving interpolation and basis functions. Numerical quadrature is used to calculate all of the necessary integrals for the stiffness matrix. For the in-plane integrals in the stiffness matrix, the Fekete sampling points are used as integration points [23]. The numerical integration using Fekete points is computationally expensive because it uses more integration points than gaussian quadrature. The time mass matrices, through-thickness mass, and through-thickness stiffness matrices are integrated using one-dimensional Gauss-Legendre numerical quadrature.

The load vector, $\{H_e\}$, from (2.40) is made up of the source term component contain-

ing Q and the heat flux boundary component containing q_s . The source term component is integrated over the in-plane direction using Fekete points. The integration over the time and through-thickness directions is accomplished using one-dimensional Gauss-Legendre numerical quadrature. For the heat flux boundary component, if q_s is constant, in-plane flux contributions along the element edges are determined exactly using [13]

$$q_s \int_a^b L_m L_n ds = \frac{q_s m! n!}{m+n+1} (b-a) \quad (5.1)$$

where L_m and L_n are the non-zero area coordinates along an element edge.

The surface flux boundary conditions occur along the top and bottom of the implied through-thickness. These boundary condition contributions to the load vector are integrated numerically. The finite element formulation presented in this research is robust enough to account for non-uniform edge and surface heat flux boundary conditions, however, the ndgSCHTp code can only handle uniform heat flux boundary conditions.

5.2 Master Element Matrices

Calculating the finite element matrices can become computationally expensive for higher-degree approximations and meshes with a large number of elements. This is especially true for the stiffness matrix, $[K_e(\mathbf{a}_{n,e})]$, which is a sum of \tilde{N}_b matrices for each element, one for every interpolation function. To overcome the excessive number of terms that have to be integrated, a set of master matrices are created for the in-plane, time and through-thickness matrices needed to create the element stiffness matrices. By interpolating the thermal conductivity, the stiffness matrix does not have to be calculated at every iteration. Instead, the matrices for each interpolation function are created only once, but are weighted and summed at each iteration. The master matrix routines created by Walker are used for the in-plane and through-thickness basis functions [9]. The methods

of storage and calculation were extended in this work to include the new time master matrices.

The properties of hierarchical basis functions allow the creation of master matrices. For a given degree of temperature approximation, the hierarchical basis contains all of the basis functions needed for that degree of approximation and any lower degree of approximation. As a result, the matrices for the highest degree of approximation contain the necessary matrices for lower degrees of approximation that can then be extracted for elements of lower degree. The highest degree matrices are therefore called master matrices. The master matrices are computed in local coordinates and the element matrices are created by extracting the proper sub-matrix from the master matrices and mapping the sub-matrix to global coordinates. The integrals necessary to create the master matrices are performed once at the beginning of ndgSCHTp. As a result, there is a reduction in the number of integrals that need to be calculated.

Master matrices are created using the local matrices from (3.13), (3.22), (3.23), (3.46), and (3.53) through (3.56) for a total of eight different master matrices. The master matrices are formed using the matrices from a single equation above. A master matrix is created for each interpolation function. As an example, the j^{th} master time mass matrix is the j^{th} local time mass matrix given by

$$[M_t^j]_l = [M_t^j]_{master} = \int_{-1}^1 \tilde{\theta}_j \theta \theta^T d\tau \quad (5.2)$$

The local matrices are converted to global matrices by scaling with the appropriate factor. For (5.2), the appropriate scaling factor from (3.13) is $\frac{\Delta t}{2}$. The master time mass matrices for $p_t = 6$ and $r_t = 2$ are shown in Figure 5.2. The master matrices are stored in two-dimensional arrays with the master matrix for each interpolation function stored beneath the master matrix for the previous interpolation function. There are three 7×7 master matrices, one for each time interpolation function. The sub-matrices extracted for $p_t = 2$

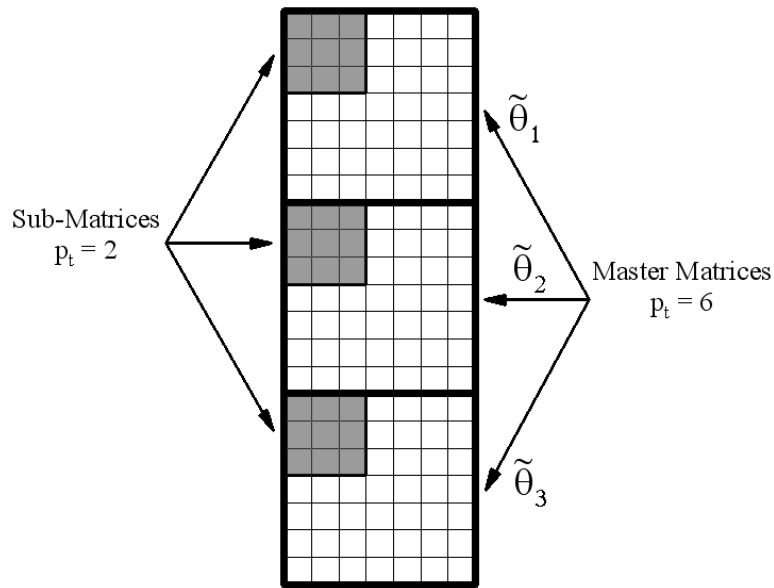


Figure 5.2: Extraction of Sub-Matrix from Master Time Mass Matrix

are 3×3 . The master matrices for the one-dimensional time and through-thickness basis from (3.13), (3.22), and (3.23) are stored and extracted in the same manner.

Since the Lagrange interpolation functions change for different degrees of approximation, separate master matrices must be created for each degree of interpolation function. If the degree of interpolation varies for elements in the mesh, new master matrices must be created for each degree of interpolation. To reduce the number of master matrices needed, the interpolation degree is required to be the same for every element. Any degree can be chosen for each of the time, through-thickness, and in-plane interpolation functions. However, once a degree is chosen for each, this degree of interpolation is used for every element in the mesh.

The master matrices for the two-dimensional in-plane basis are stored according to the type of basis function. The vertex basis functions are stored first, followed by the edge basis functions, and finally the interior basis functions. The order of storage for a master in-plane basis for an element of degree 6 is shown in Figure 5.3. The element basis extracted from the master basis shown in Figure 5.3 has degrees of approximation of 1

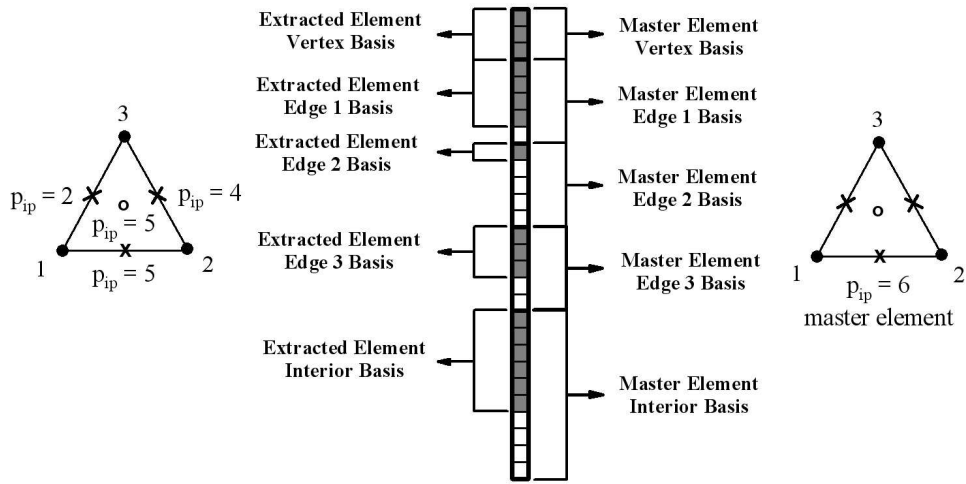


Figure 5.3: Extraction of Element Basis Functions from Master In-Plane Basis

for each of the three vertices, 5, 2, and 4 for the three edges, and 5 for the interior. The square in-plane matrices from (3.46) and (3.53) through (3.56) each have their own set of master matrices, one for each in-plane interpolation function. The sub-matrices for the appropriate degree of temperature approximation are extracted from the master matrix by keeping track of the locations of the necessary basis functions for the vertex, edge, and interior in the master basis as shown in Figure 5.3. The extraction of the in-plane sub-matrix for the same element used in Figure 5.3 is shown in Figure 5.4. The shaded boxes represent the parts of the master matrix that are extracted to form the element matrix. The master matrix corresponding to a temperature approximation of degree six is a 28×28 array. The sub-matrix that is extracted has varying degrees of approximation along the edges and interior of the element and is 17×17 . All of the in-plane master matrices are stored in the same order and the sub-matrices are extracted in the same manner as shown in Figure 5.4. Once the proper sub-matrices (local matrices) have been extracted, the in-plane element mass and stiffness matrices are formed by mapping the extracted local matrices to global coordinates as shown in (3.46) and (3.52).

For the collapsed elements in the finite element formulation, the through-thickness

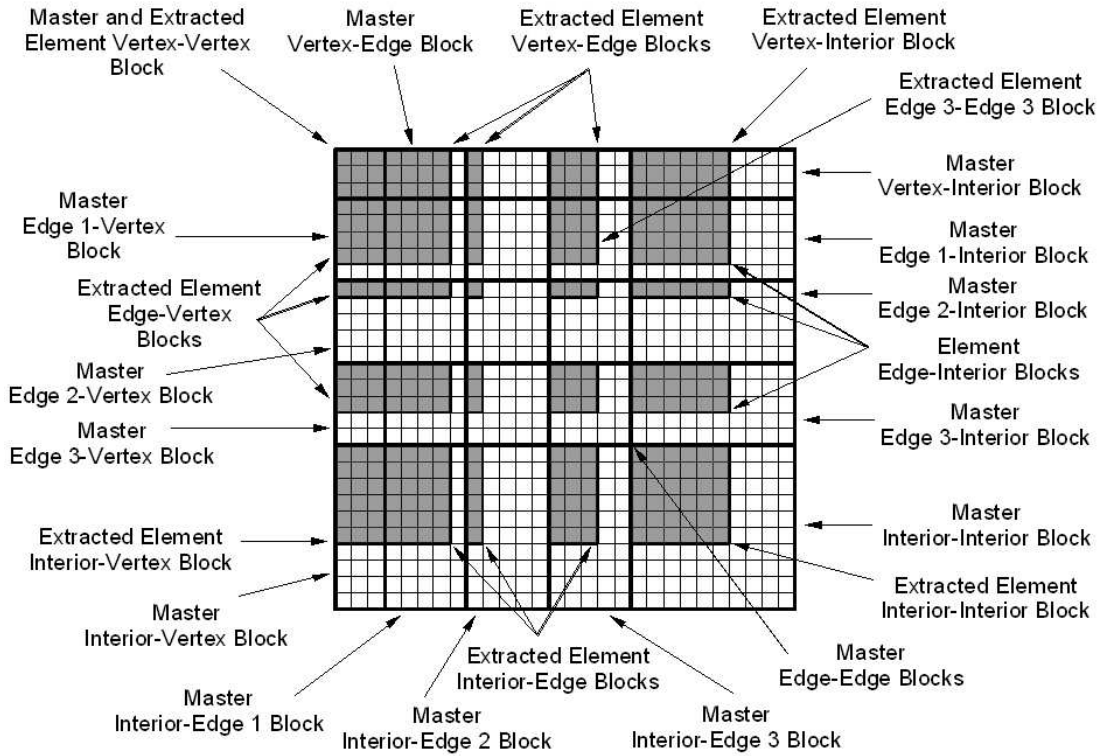


Figure 5.4: Extraction of Sub-Matrix from Master In-Plane Matrix

basis functions are associated with the in-plane nodes on the two-dimensional element. The finite element formulation presented in Chapter 2 allows the degree of approximation in the through-thickness direction to vary at each node. A sample element with varying degrees of approximation in the through-thickness direction at each node is shown in Figure 5.5. For varying through-thickness degrees of approximation, the tensor product must be performed with the proper through-thickness degree of approximation. This ensures that the correct through-thickness basis functions are associated with each in-plane node. The in-plane and time degrees of freedom that form an entry in the matrix are multiplied by the appropriate through-thickness matrix using the tensor product. As an example, suppose $\phi_1\theta_2$ and $\phi_4\theta_3$ are the two degrees of freedom multiplied together to form an entry in this matrix. This entry is one term in a matrix formed by a tensor product on the in-plane and time matrices. Also assume that the through-thickness degree of

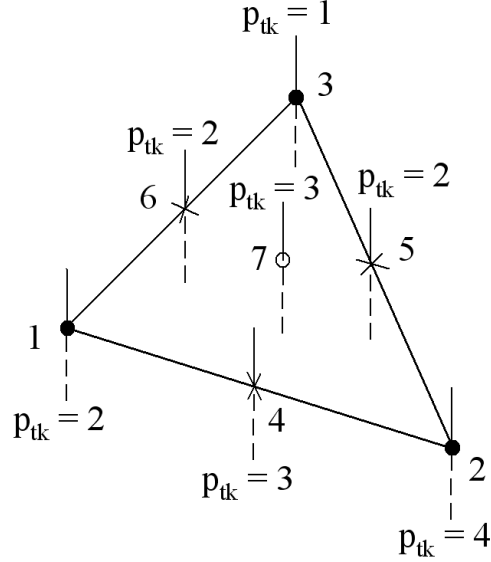


Figure 5.5: Sample Element with Varying Degrees of Through-Thickness Approximation

approximation for each of these in-plane nodes is $p_{tk} = 2$ and $p_{tk} = 3$ respectively. The through-thickness matrix multiplied by this term using the tensor product would have a dimension of 3×4 . The tensor product operation for this example is shown below

$$\phi_1\phi_4\theta_2\theta_3 \otimes [\psi\psi^T] = \phi_1\phi_4\theta_2\theta_3 \begin{bmatrix} \psi_0\psi_0 & \psi_0\psi_1 & \psi_0\psi_2 & \psi_0\psi_3 \\ \psi_1\psi_0 & \psi_1\psi_1 & \psi_1\psi_2 & \psi_1\psi_3 \\ \psi_2\psi_0 & \psi_2\psi_1 & \psi_2\psi_2 & \psi_2\psi_3 \end{bmatrix} \quad (5.3)$$

The tensor product is performed on a matrix level in ndgSCHTp for all of the matrices. Currently, the degree of the through-thickness approximation must be the same over the entire mesh. The code could easily be modified to allow for variation in the degree of the through-thickness approximation. This is accomplished by creating a more robust tensor product operator that keeps track of the degree of through-thickness approximation at each in-plane node. Previous versions of the ndgSCHTp allowed the degree of the through-thickness approximation to vary at each in-plane node, but as capability was added to the code this feature was not retained.

5.3 Enforcing Essential Boundary Conditions

The boundary conditions must be accounted for by the finite element code. The natural boundary conditions have already been included in the finite element formulation, but the essential boundary conditions need to be satisfied. The two different types of essential boundary conditions used by ndgSCHTp are specified edge and surface temperatures. A sample three-dimensional domain with surface and edge boundary conditions is shown in Figure 5.6. Edge boundary conditions are specified using the linear in-plane unknowns on the element edges coincident with the boundaries of the in-plane domain and may occur for two or three-dimensional problems. For two-dimensional problems, the edge boundary conditions are simply the outer edges of the domain with essential boundary conditions. For three-dimensional problems, the edge boundary conditions also account for the essential boundary conditions on faces that are perpendicular to the in-plane direction. These boundary conditions run in the z direction along the boundary of the in-plane domain. Surface boundary conditions are usually used in three-dimensional problems. The surface boundary conditions occur at the top and bottom of the through-thickness approximation and are specified using the linear through-thickness degrees of freedom.

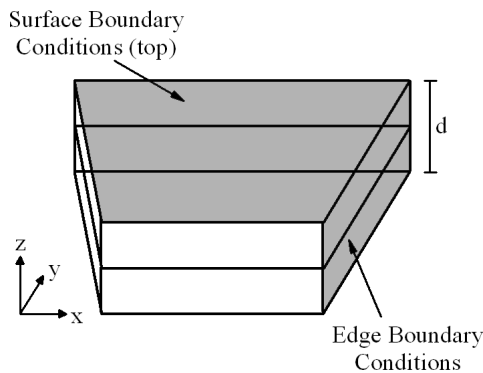


Figure 5.6: Three-Dimensional Domain with Surface and Edge Boundary Conditions

To capture the edge boundary conditions, the appropriate in-plane unknowns are modified. For a constant temperature on the edges, all of the higher-degree in-plane unknowns on the boundary are set to zero. The linear in-plane unknowns on the boundary are set equal to the boundary condition temperature. For the surface boundary conditions, the appropriate through-thickness unknowns are assigned values. For a constant or linear surface temperature, all of the higher-degree through-thickness unknowns are set to zero. The linear through-thickness degrees of freedom are set equal to the constant boundary condition temperature. If the surface temperature has different values on the top and bottom, then the linear through-thickness unknowns are set to the corresponding temperature. The through-thickness unknowns corresponding to ψ_0 are set to the surface temperature of the bottom surface. The degrees of freedom corresponding to ψ_1 are set to the surface temperature of the top surface.

If the boundary condition is a polynomial in space, the unknowns corresponding to the polynomial are used to account for the boundary condition. The unknowns are defined so that the polynomial distribution on the boundary is reflected in the solution. Unknowns corresponding to basis functions with degrees higher than that of the polynomial are set to zero.

To enforce essential boundary conditions, ndgSCHTp uses the penalty method. After the global matrix and source vector are assembled at each time step, the penalty method is applied to ensure that the known degrees of freedom attain the proper values. The penalty parameter is a very large number used to implement the method and in ndgSCHTp, the penalty parameter is 10^{35} . As an example, suppose that the global matrix is 5×5 , the essential boundary condition has a constant value of T_{bc} , and the essential boundary condition occurs at unknowns a_1 and a_4 . The penalty method for this example is shown

below

$$\begin{bmatrix} S_{11}(10^{35}) & S_{12} & S_{13} & S_{14} & S_{15} \\ S_{21} & S_{22} & S_{23} & S_{24} & S_{25} \\ S_{31} & S_{32} & S_{33} & S_{34} & S_{35} \\ S_{41} & S_{42} & S_{43} & S_{44}(10^{35}) & S_{45} \\ S_{51} & S_{52} & S_{53} & S_{54} & S_{55} \end{bmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{pmatrix} = \begin{pmatrix} F_1 + T_{bc}(10^{35}) \\ F_2 \\ F_3 \\ F_4 + T_{bc}(10^{35}) \\ F_5 \end{pmatrix} \quad (5.4)$$

After the penalty method is applied, (5.4) is solved for the unknowns and the solution satisfies the essential boundary conditions. In general, the diagonal entry in the global matrix corresponding to an unknown with an essential boundary condition is multiplied by the penalty. The penalty is then multiplied by the value of the constant boundary condition temperature and added to the terms in the source vector corresponding to the unknowns with essential boundary conditions.

Chapter 6

Numerical Results

After the finite element approximation is calculated, the accuracy of the approximation must be determined. In this chapter, the methods used to quantify the error are discussed. This discussion is followed by several sample problems and an error analysis of each problem.

6.1 Error Convergence Estimates

In order to determine the accuracy of the finite element method presented, an effective way of quantifying the error in the approximation is needed. There are two types of error estimates used to quantify the error, *a priori* and *a posteriori*. *A priori* error estimates give the theoretical rate of convergence of the error and are based on the exact solution to a problem. *A posteriori* error estimates use the finite element solution to a problem since the exact solution is usually not known. In this research, the rate of error convergence for the approximation is compared to *a priori* error convergence estimates.

The error in the approximation at any point is defined as

$$e(x, y, z, t) = u(x, y, z, t) - \hat{u}(x, y, z, t) \quad (6.1)$$

where u is the exact solution and \hat{u} is the approximate solution. The error analysis in this research utilizes H^1 norms in space and L^2 norms in time. The H^1 , or energy norm,

quantifies the spatial error in the finite element approximation

$$\| e(t) \|_{H^1(\Omega)} = \left(\int_{\Omega} \left[e^2 + \left(\frac{\partial e}{\partial x} \right)^2 + \left(\frac{\partial e}{\partial y} \right)^2 + \left(\frac{\partial e}{\partial z} \right)^2 \right] dx dy dz \right)^{\frac{1}{2}} \quad (6.2)$$

This error norm is a function of time. By evaluating (6.2) at a particular point in time, the global error in space at any point in time can be determined. The global error in space and time is calculated by taking the L^2 norm of (6.2) with respect to time

$$\begin{aligned} \| e \|_{L^2(I, H^1(\Omega))} &= \left[\int_0^T \| e(t) \|_{H^1(\Omega)}^2 dt \right]^{\frac{1}{2}} \\ &= \left[\int_0^T \left(\int_{\Omega} \left[e^2 + \left(\frac{\partial e}{\partial x} \right)^2 + \left(\frac{\partial e}{\partial y} \right)^2 + \left(\frac{\partial e}{\partial z} \right)^2 \right] dx dy dz \right) dt \right]^{\frac{1}{2}} \end{aligned} \quad (6.3)$$

where I denotes the entire time interval, $(0, T]$, and $H^1(\Omega)$ represents the spatial error norm. The L^2 norm is also known as the root mean square norm because it is analogous to the root mean square error at a point. The difference is that the L^2 norm is used for the error between two functions, not two points. The global error over the entire time interval is given by (6.3). For multiple time steps, the error norm is calculated at each time step

$$\| e \|_{L^2(I_n, H^1(\Omega))} = \left[\int_{t_{n-1}}^{t_n} \left(\int_{\Omega} \left[e^2 + \left(\frac{\partial e}{\partial x} \right)^2 + \left(\frac{\partial e}{\partial y} \right)^2 + \left(\frac{\partial e}{\partial z} \right)^2 \right] dx dy dz \right) dt \right]^{\frac{1}{2}} \quad (6.4)$$

and the total global error norm is the sum of the error norms from each time step

$$\| e \|_{L^2(I, H^1(\Omega))} = \left[\sum_1^{n_t} \| e \|_{L^2(I_n, H^1(\Omega))}^2 \right]^{\frac{1}{2}} \quad (6.5)$$

One issue with the H^1 and L^2 norms is that the magnitude depends on the magnitude of the exact solution. This means that the value of the global error has little meaning without the magnitude of the exact solution. In order to remedy this, the magnitude of the H^1 and L^2 norms of the exact solution are calculated for each problem. Using the magnitude of the exact solution norms, the H^1 and L^2 error norms can be normalized

and the percent error in the H^1 and L^2 norms calculated

$$\% \text{ error in } H^1(\Omega) = 100 \left(\frac{\| e(t) \|_{H^1(\Omega)}}{\| u(t) \|_{H^1(\Omega)}} \right) \quad (6.6)$$

$$\% \text{ error in } L^2(I, H^1(\Omega)) = 100 \left(\frac{\| e \|_{L^2(I, H^1(\Omega))}}{\| u \|_{L^2(I, H^1(\Omega))}} \right) \quad (6.7)$$

For *a priori* error estimates, the rate convergence of the error in an approximation is known prior to the calculation of the approximation. The error for the in-plane, through-thickness, and time approximations are dependent on the degree of the approximation and the mesh size [27]. Work by Schötzau and Schwab [16] and Werder, et al. [26] established the error bounds for linear, transient heat conduction problems. The error bounds assume a uniform mesh size and uniform degrees of approximation for all of the elements. For an approximation where the only source of error is the in-plane approximation, the H^1 norm of the error is bounded by

$$\| e(t) \|_{H^1(\Omega)} \leq C_1 h^{p_{ip}} \quad (6.8)$$

where C_1 is a constant, h is the size of an in-plane element, and p_{ip} is the degree of the in-plane approximation. As the size of an element is decreased, the H^1 error converges to zero at a rate of p_{ip} . For an approximation where the only source of error is the approximation in time, the L^2 norm of the error is bounded by

$$\| e \|_{L^2(I, H^1(\Omega))} \leq C_2 \Delta t^{p_t+1} \quad (6.9)$$

where C_2 is a constant, Δt is size of the uniform time step, and p_t is the degree of the time approximation. As the size of the time step is decreased, the L^2 error converges to zero at a rate of $p_t + 1$. Since the size of the time step is inversely proportional to the number of time steps, this means that as the number of time steps increases, the L^2 error converges to zero at a rate of $p_t + 1$. In (6.8) and (6.9), the in-plane and time error,

respectively, must each be the sole source of error in the approximation in order for the bounds to apply. Taking the logarithm of both sides of (6.8) and (6.9) yields

$$\log \| e(t) \|_{H^1(\Omega)} \leq p_{ip} \log h + \log C_1 \quad (6.10)$$

$$\log \| e \|_{L^2(I, H^1(\Omega))} \leq (p_t + 1) \log \Delta t + \log C_2 \quad (6.11)$$

Recall slope-intercept form of an equation of a line

$$y = mx + c \quad (6.12)$$

where m is the slope of the line and c is the y intercept. By comparing (6.10) and (6.11) to (6.12) and by treating the log of the error norms as the dependent variable, y , and the log of the size of the element or the log of the size of the time step as the independent variable, x , the slope of the lines on a log-log plot can be determined. From (6.10), the logarithms of the H^1 error as a function of mesh size should decrease at a rate equal to the degree of the in-plane approximation, p_{ip} . In other words, on a log-log plot of the H^1 error, the slope of the line should equal the degree of the in-plane approximation. From (6.11), the logarithms of the L^2 error should decrease at a rate equal to the degree of approximation in time, p_t , plus one. This means that on a log-log plot of the L^2 error, the slope of the line should equal the degree of approximation in time plus one. The *a priori* convergence estimates for the H^1 and L^2 norms are illustrated in Figure 6.1 (a) and (b). In order to compare convergence rates, uniform elements are used in all of the sample problems and the plots are presented on a log-log scale. This allows the slope of the lines to be used to determine the rate of error convergence.

These error norms and convergence rates are valid for transient linear problems. Walker showed that the H^1 error convergence rates given above are valid for nonlinear steady-state problems when the in-plane temperature approximation is the dominant source of error [9]. The applicability of these convergence rates to transient, nonlinear

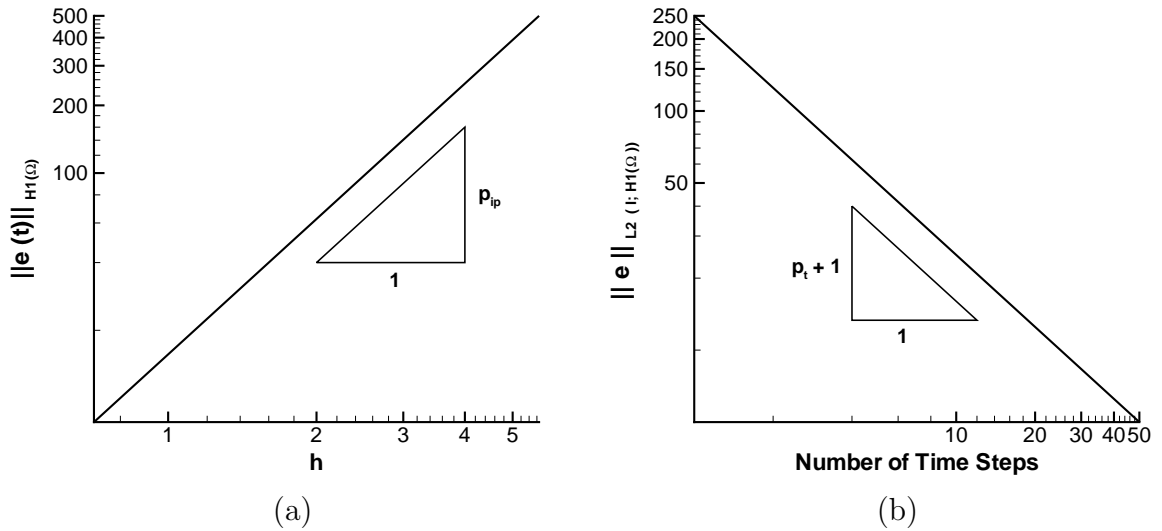


Figure 6.1: Error Convergence Estimates on a Log-Log Scale

(a) H^1 Error vs. Element Size

(b) L^2 Error vs. Number of Time Steps

problems has not been previously studied. The results presented later in this chapter include a discussion of the validity of these error convergence rates for nonlinear, transient problems.

6.2 Error Due to Fixed-Point Iteration

To determine the convergence rates for a degree of approximation as the mesh or time step size is refined, all other sources of error must be eliminated. The error due to the fixed-point iteration is eliminated by setting the convergence tolerance to a sufficiently small number. In this research 10^{-12} is used as the convergence tolerance for all of the examples presented here. To limit the amount of time spent in the iteration, the maximum number of iterations is also specified for each run. Unless otherwise specified in the following sample problems, the maximum number of iterations allowed is 15. All of the results presented in this research either converged to the predefined tolerance or the error resulting from the 15 iterations was several orders of magnitude lower than the

error in the solution.

For real world applications, an iteration tolerance of 10^{-12} is not practical or necessary. For practical purposes, convergence to a tenth or hundredth of a degree is adequate. This accuracy is normally obtained by ndgSCHTp within three or four iterations.

6.3 Sample Problems

For most nonlinear, transient problems of conduction heat transfer, an exact solution to the initial boundary value problem is impossible to determine. In order to precisely analyze the accuracy of the method, problems with exact solutions are created so that the exact error can be computed. Since the initial and boundary conditions are determined directly from the exact solution, an exact solution with appropriate initial and boundary conditions should be chosen. To construct a problem with an exact solution, u , the source term for the problem, Q , is obtained by substituting u into (2.1)

$$Q = \rho c_p \frac{\partial u}{\partial t} - k(u) \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) - \frac{\partial k(u)}{\partial x} \frac{\partial u}{\partial x} - \frac{\partial k(u)}{\partial y} \frac{\partial u}{\partial y} - \frac{\partial k(u)}{\partial z} \frac{\partial u}{\partial z} \quad (6.13)$$

A function for the thermal conductivity is also chosen. In the examples presented in this research, the thermal conductivity is restricted to be a polynomial function of temperature of the form

$$k(u) = C_k u^m + k_0 \quad (6.14)$$

where C_k and k_0 are constants and m is an integer exponent. Substituting (6.14) into (6.13) and simplifying yields

$$Q = \rho c_p \frac{\partial u}{\partial t} - (C_k u^m + k_0) \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) - m C_k u^{m-1} \left(\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial y} \right)^2 + \left(\frac{\partial u}{\partial z} \right)^2 \right) \quad (6.15)$$

If the exact solution and thermal conductivity are chosen to be simple polynomials, then ndgSCHTp should produce the exact solution when the polynomial degree of the

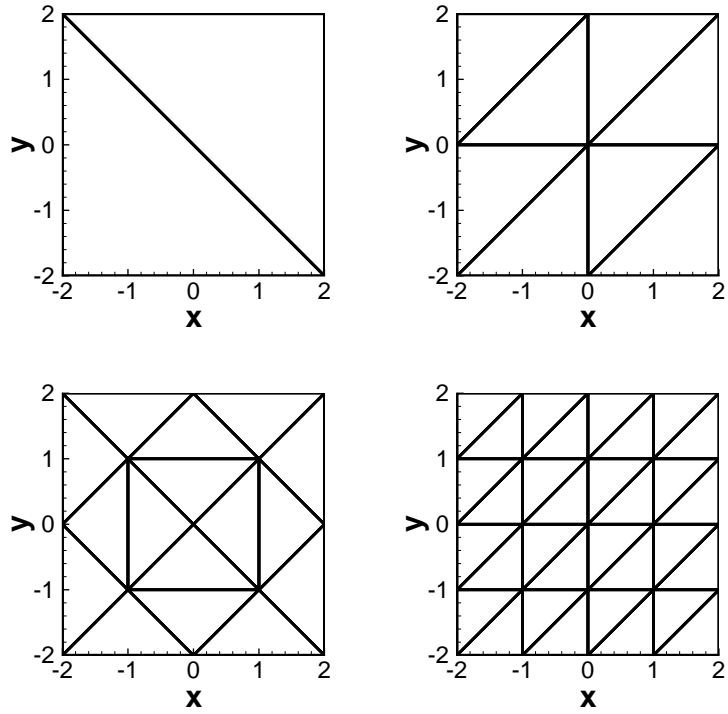


Figure 6.2: 2, 8, 16, and 32 Element In-plane Meshes

temperature approximation equals that of the exact solution and the degree of the conductivity interpolant captures the actual variation in the thermal conductivity over an element.

The in-plane uniform meshes used in this research are shown in Figure 6.2. A table of the number of elements with their corresponding element size, h , is shown in Table 6.1.

Number of Elements	h
2	5.657
8	2.828
16	2.000
32	1.414

Table 6.1: Number and Size of Elements Used to Analyze Example Problems

6.3.1 Example 1

The first example is used to validate the in-plane temperature approximation and the in-plane interpolation of the thermal conductivity. The exact solution for example one is chosen to be

$$u = 100 \left(1 - \frac{x}{2}\right)^2 \left(1 + \frac{x}{2}\right) \left(1 - \frac{y}{2}\right) \left(1 + \frac{y}{2}\right) t + 50 \quad (6.16)$$

The domain for this problem is a 4×4 sheet lying in the $x - y$ plane with its center at the origin. The thermal conductivity for this problem is chosen to be

$$k(u) = 0.005u + 1.0 \quad (6.17)$$

Since the exact solution was chosen ahead of time, the value of the thermal capacitance is not an important parameter for the problem and is chosen to be

$$\rho c_p = 175 \quad (6.18)$$

The time interval is chosen to run from $t = 0s$ until $t = 1s$. The initial and boundary conditions are

$$\begin{aligned} u(-2, y, t) &= u(2, y, t) = 50 \\ u(x, -2, t) &= u(x, 2, t) = 50 \\ u(x, y, 0) &= 50 \end{aligned} \quad (6.19)$$

The source term for example one is calculated by substituting (6.16) and (6.17) into (6.15). The temperature distribution for the exact solution at $t = 1s$ is shown in Figure 6.3 (a).

As input for ndgSCHTp, the thermal conductivity is tabulated for several values of temperature including temperature values above and below the maximum and minimum values for the exact solution. Since the thermal conductivity is a linear function of

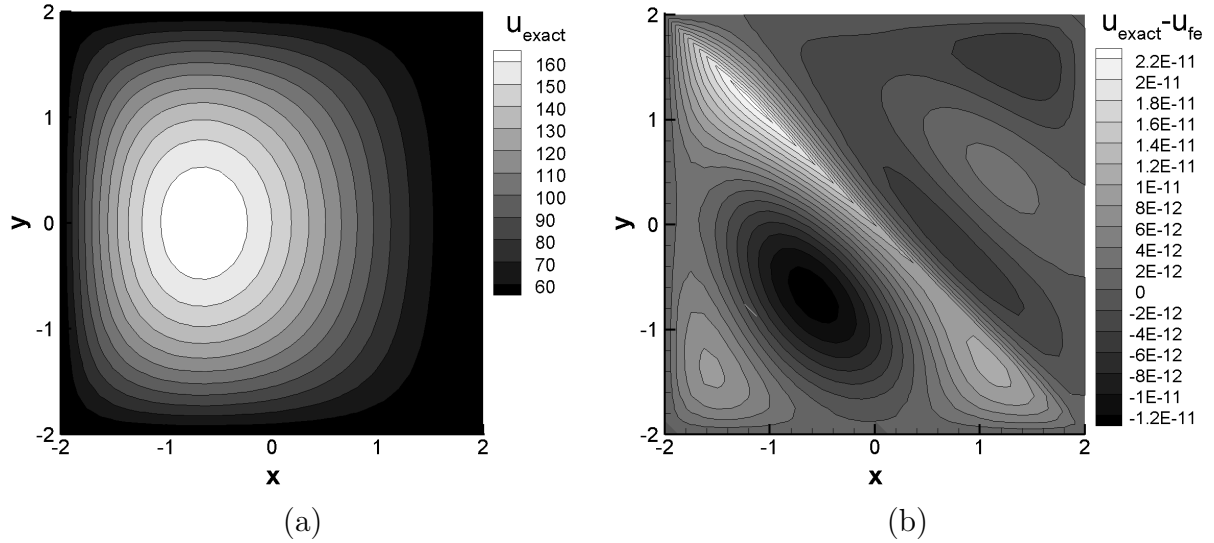


Figure 6.3: Exact Solution and Finite Element Error for Example 1 at $t = 1s$

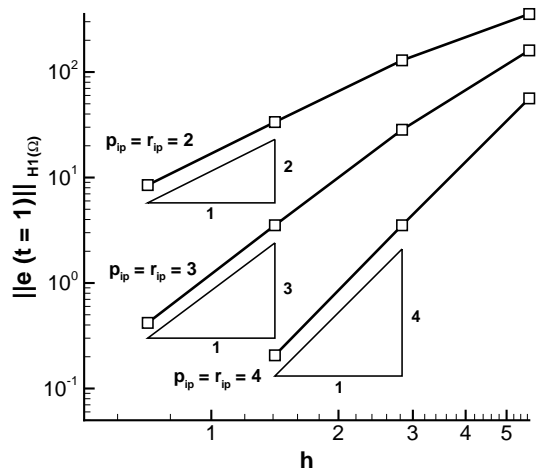
(a) Exact solution

(b) Point-wise error with $p_{ip} = r_{ip} = 5$ and $p_t = r_t = 1$

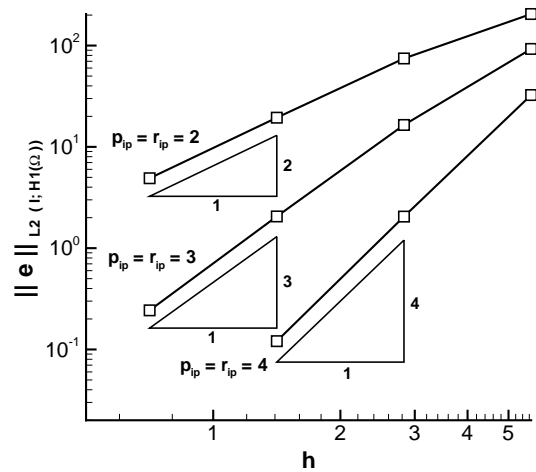
temperature and ndgSCHTp linearly interpolates between the tabulated conductivity values, no error is introduced due to the tabulation of the thermal conductivity.

The exact solution is a product of a fifth degree polynomial in the in-plane direction and a first degree polynomial in time. Therefore, ndgSCHTp should reproduce the exact solution when $p_{ip} = r_{ip} = 5$ and $p_t = r_t = 1$ using just one time step. The exact solution was successfully reproduced by ndgSCHTp to within machine precision as shown in the point-wise error contours in Figure 6.3 (b).

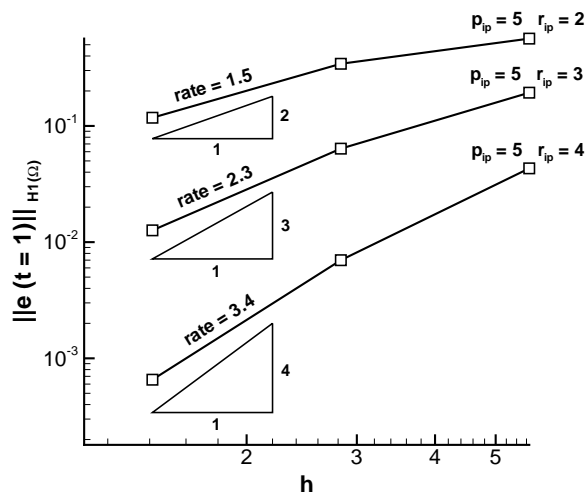
The error convergence results for example one are shown in Figure 6.4. The x axis is h , the size of the in-plane elements in the meshes. The y axis is the H^1 error at $t = 1s$ (left side of figure) or the L^2 error (right side of figure). Each point on the plots represents the error in a finite element approximation obtained with a mesh composed of elements of a particular size. The finite element approximation is produced using ndgSCHTp and the error is calculated using an error routine written specifically for ndgSCHTp. In order to isolate the error due to the in-plane approximation, the degree of the time approximation



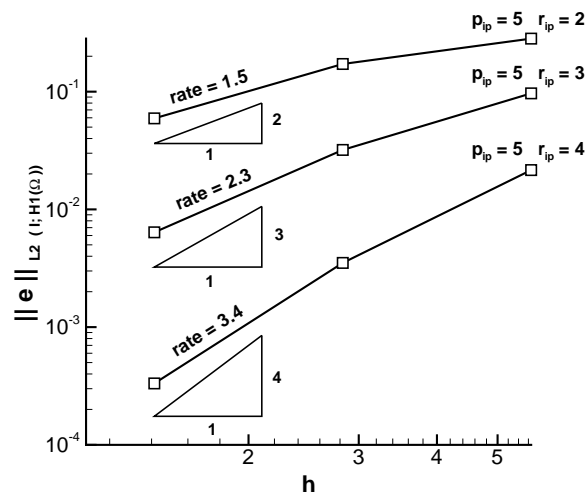
(a)



(b)



(c)



(d)

Figure 6.4: Convergence of the Error for Example 1

$$\|u(t=1)\|_{H^1(\Omega)} \approx 694.5 \quad \|u\|_{L^2(I, H^1(\Omega))} \approx 487.6$$

(a) H^1 Error for Different Values of p_{ip} (b) L^2 Error for Different Values of p_{ip}

(c) H^1 Error for Different Values of r_{ip} (d) L^2 Error for Different Values of r_{ip}

and interpolation was set equal to that of the exact solution, $p_t = r_t = 1$. This makes the finite element approximation exact in time. Since there is no error in time, the in-plane temperature approximation and conductivity interpolation are the only sources of error in the finite element approximation.

In Figure 6.4, each line represents a finite element approximation with a particular value for the degree of in-plane temperature approximation, p_{ip} , and the degree of in-plane conductivity interpolation, r_{ip} . Each point on a line represents the error for a given size mesh. As the number of elements in the mesh is increased, the size of the element, h , decreases. The error convergence rates for linear problems, discussed earlier in this chapter, are compared to the convergence rates obtained for this nonlinear problem.

In Figure 6.4 (a), the H^1 error versus mesh size at $t = 1s$ is shown on a log-log scale. Each line represents a different value for p_{ip} . The in-plane interpolation degree, r_{ip} , is set equal to the in-plane temperature approximation degree, p_{ip} . The triangles show the theoretical rates of convergence for linear problems. The convergence rates for this nonlinear example are identical to the convergence rates expected for a linear problem. In Figure 6.4 (b), the L^2 error versus mesh size is shown on a log-log scale for varying in-plane degrees of temperature approximation. The L^2 error is a representation of the total error including the secondary variables in space over the entire time interval for the problem. The convergence rates for the L^2 error should be the same as the convergence rates for the H^1 error since the L^2 error takes into account the H^1 error in space. Again, the expected convergence rates for the linear problem are achieved for this nonlinear problem. The triangles in the figure have the same slope as the lines.

In Figure 6.4 (c) and Figure 6.4 (d), the H^1 error at $t = 1s$ and the L^2 error versus mesh size are shown on a log-log scale. Each line represents a different value for r_{ip} with $p_{ip} = 5$, the degree of the exact solution. This isolates the in-plane interpolation of the

thermal conductivity so that it is the only source of error in the approximation. The rate of convergence for the in-plane interpolation is written above each line in the plots. These convergence rates show that in order to obtain the convergence rates expected for linear problems, the degree of the temperature approximation must be equal to the degree of interpolation. The triangles serve as a reference to compare with the slopes of the lines.

As a reference, the H^1 error at $t = 1s$ and the L^2 error of the exact solution for example one are given in the caption of Figure 6.4. Using these values, (6.6), and (6.7), a percent error can easily be calculated that gives a relative measure of error rather than an absolute error that is dependent on the magnitude of the solution to the problem.

6.3.2 Example 2

Example two is used to validate the temperature approximation and interpolation of the thermal conductivity in time. The exact solution for example two is chosen to be

$$u = \left(1 - \frac{x}{2}\right) \left(1 + \frac{x}{2}\right) \left(1 - \frac{y}{2}\right) \left(1 + \frac{y}{2}\right) t^6 \quad (6.20)$$

The domain for this problem is the same as that of example one. The thermal conductivity for this problem is chosen to be

$$k(u) = 0.005u + 1.0 \quad (6.21)$$

and the thermal capacitance for example two is given by

$$\rho c_p = 175 \quad (6.22)$$

The time interval is chosen to run from $t = 0s$ until $t = 2s$. The initial and boundary

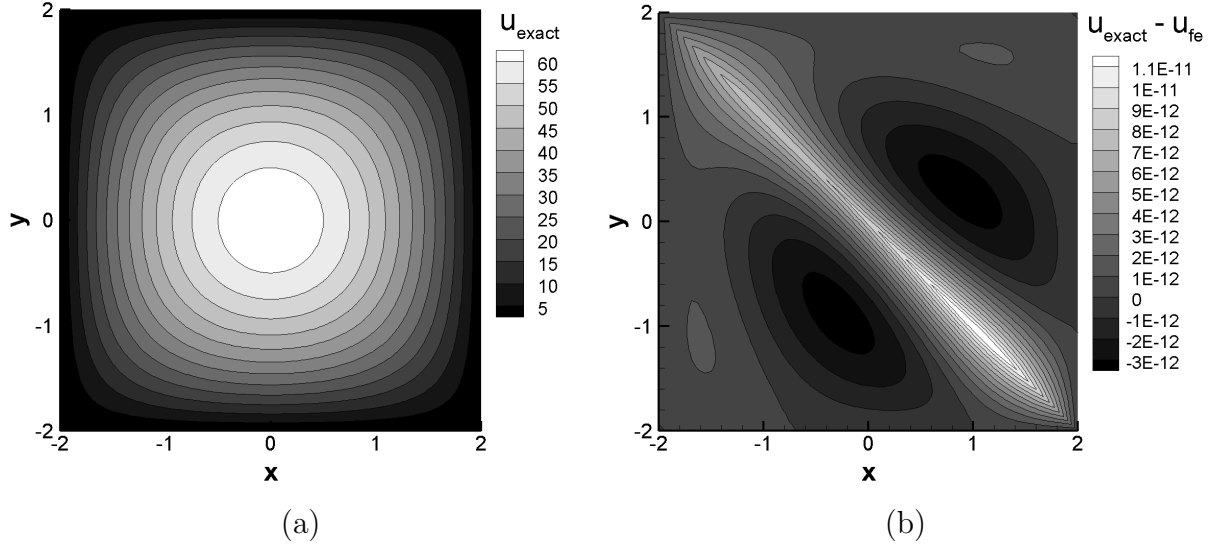


Figure 6.5: Exact Solution and Finite Element Error for Example 2 at $t = 2s$

(a) Exact solution

(b) Point-wise error with $p_{ip} = r_{ip} = 4$ and $p_t = r_t = 6$ using one time step

conditions are given by

$$\begin{aligned}
 u(-2, y, t) &= u(2, y, t) = 0 \\
 u(x, -2, t) &= u(x, 2, t) = 0 \\
 u(x, y, 0) &= 0
 \end{aligned} \tag{6.23}$$

The source term for example two is calculated by in the same manner as before, this time by substituting (6.20) and (6.21) into (6.15). The temperature distribution for the exact solution at $t = 2s$ is shown in Figure 6.5 (a).

Again, since the thermal conductivity is a linear function of temperature and ndgSCHTp uses linear interpolation, there is no error in the solution due to the tabulation of the thermal conductivity.

For example two, the exact solution is a product of a fourth degree polynomial in the in-plane direction and a sixth degree polynomial in time. Therefore, ndgSCHTp should reproduce the exact solution when $p_{ip} = r_{ip} = 4$ and $p_t = r_t = 6$ using just one time

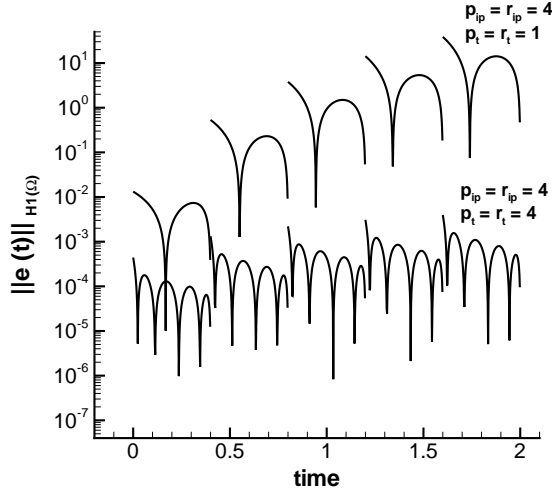


Figure 6.6: Example 2: H^1 Error as a Function of Time for $p_t = 1$ and $p_t = 4$

step. The exact solution was successfully reproduced by ndgSCHTp to within machine precision as shown in the point-wise error plot in Figure 6.5 (b).

The H^1 error versus time on a linear-log scale is shown in Figure 6.6. This figure shows two different approximations in time, $p_t = 1$ and $p_t = 4$, with both using five time steps. Some characteristics of discontinuous Galerkin time stepping appear in the figure. At each time step, there are $p_t + 1$ dips and discontinuities in the approximations occur between each of the five time steps. The discontinuous Galerkin formulation creates the discontinuities seen in the figure. It is also worthwhile to note that the shape of the error is the same for every time step and that the H^1 error grows in time. This growth of the error in time is a result of the error propagating from one time step to the next.

The error convergence results for example two are shown in Figure 6.7. The x axis is the number of time steps and the y axis is the L^2 error norm. The number of time steps can be thought of as the number of elements in the time dimension. In order to isolate the error due to the time approximation, the in-plane degree of approximation and interpolation is set equal to that of the exact solution, $p_{ip} = r_{ip} = 4$. This makes the finite element approximation exact in the in-plane direction. Since there is no error

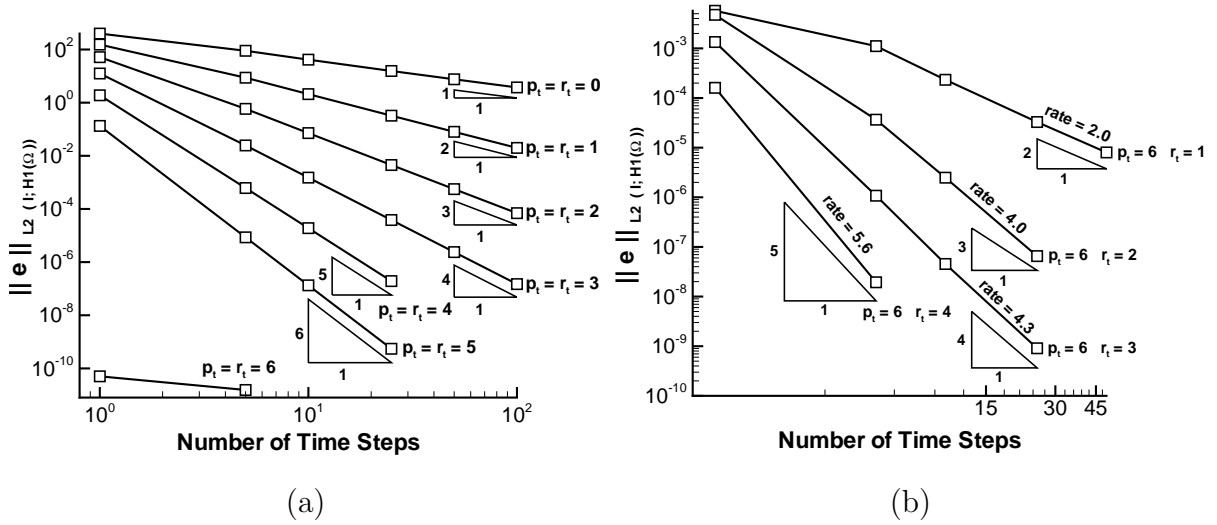


Figure 6.7: Convergence of the L^2 Error for Example 2 $\|u\|_{L^2(I; H^1(\Omega))} \approx 113.6$
(a) The Effect of Different Values of p_t (b) The Effect of Different Values of r_t

in the in-plane approximation, the temperature approximation and interpolation in time are the only sources of error in the finite element approximation.

In Figure 6.7, all points on a line represent a finite element approximation with a defined value for the degree of temperature approximation in time, p_t , and the degree of interpolation in time, r_t . Each point on a line represents the error in the approximation for a given number of time steps over the same time interval. As the number of time steps is increased, the size of the time step, Δt , decreases. Again, the plots are presented on a log-log scale allowing the slope of the line to be used to determine the rate of convergence of the error.

In Figure 6.7 (a), the L^2 error versus the number of time steps is shown. Each line represents a different value for p_t . The interpolation degree in time, r_t , is set equal to the temperature approximation degree in time, p_t . The triangles show the slope of the lines on the log-log plot and hence the theoretical rates of convergence for linear problems. The convergence rates for this nonlinear example are identical to the convergence rates

expected for a linear problem since the slope of the triangles matches the slope of the lines.

In Figure 6.7 (b), the L^2 error versus the number of time steps is shown. Each line represents different value for r_t with $p_t = 6$, the degree of the exact solution, so that the interpolation of the thermal conductivity in time is the only source of error. The actual rate of convergence in time is written above each line in the plot. These convergence rates show that in order to obtain the convergence rates expected for linear problems, the degree of the temperature approximation must be equal to the degree of interpolation. The triangles serve as a reference to compare with the slopes of the lines.

Again, as a reference, the L^2 error norm of the exact solution is given in the caption of Figure 6.7. Using this value and (6.7), a percent error can be calculated for example two.

6.3.3 Example 3

This example is used to validate the through-thickness temperature approximation and thermal conductivity interpolation method. The exact solution for example three is chosen to be

$$u = 100 \left(1 - \frac{x}{2}\right) \left(1 + \frac{x}{2}\right) \left(1 - \frac{y}{2}\right) \left(1 + \frac{y}{2}\right) (1 - z)^4 (1 + z)^2 t \quad (6.24)$$

The domain for this problem is a $4 \times 4 \times 2$ rectangular prism in the x , y , and z directions respectively, with its center at the origin. The thermal conductivity for this problem is chosen to be the same as in example one and two

$$k(u) = 0.005u + 1.0 \quad (6.25)$$

The thermal capacitance for example three is chosen to be

$$\rho c_p = 1 \quad (6.26)$$

The time interval is chosen to run from $t = 0s$ until $t = 10s$. The initial and boundary conditions for $u(x, y, z, t)$ are

$$\begin{aligned}
u(-2, y, z, t) &= u(2, y, z, t) = 0 \\
u(x, -2, z, t) &= u(x, 2, z, t) = 0 \\
u(x, y, -1, t) &= u(x, y, 1, t) = 0 \\
u(x, y, z, 0) &= 0
\end{aligned} \tag{6.27}$$

The source term is calculated in the same manner as before, by substituting (6.24) and (6.25) into (6.15). The temperature distribution for the exact solution at $t = 10s$ is shown in Figure 6.8 (a).

Again, since the thermal conductivity is a linear function of temperature and ndgSCHTp uses linear interpolation between tabulated values, there is no error in the solution due to the tabulation of the thermal conductivity.

The exact solution is a product of a fourth degree polynomial in the in-plane direction, a first degree polynomial in time, and a sixth degree polynomial in the through-thickness direction. This means that ndgSCHTp should reproduce the exact solution when $p_{ip} = r_{ip} = 4$, $p_t = r_t = 1$, and $p_{tk} = r_{tk} = 6$ using just one time step. The exact solution was successfully reproduced by ndgSCHTp to within machine precision as shown in the point-wise error plot in Figure 6.8 (b).

The iteration tolerance for example three is 10^{-12} and the maximum number of iterations allowed is 40. The iteration convergence was slower for this problem because the approximation is a product of three separate terms: the in-plane, time, and through-thickness basis functions. In the first two problems, the approximation was a product of only the in-plane and time basis functions. The inclusion of the through-thickness basis function results in matrices that are more susceptible to computer round off error. For

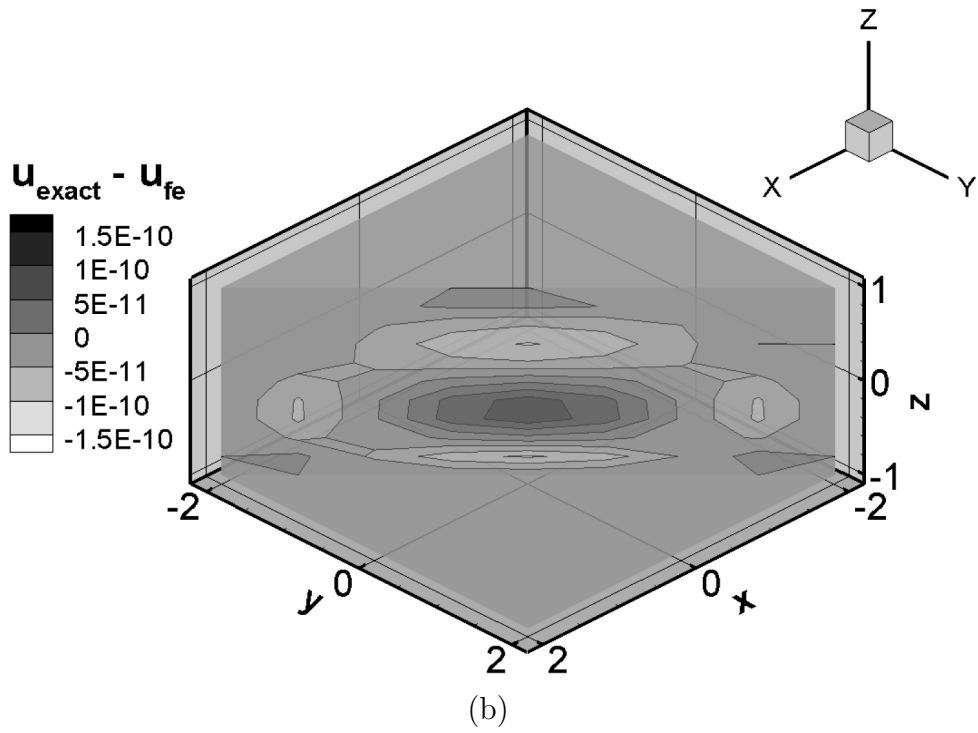
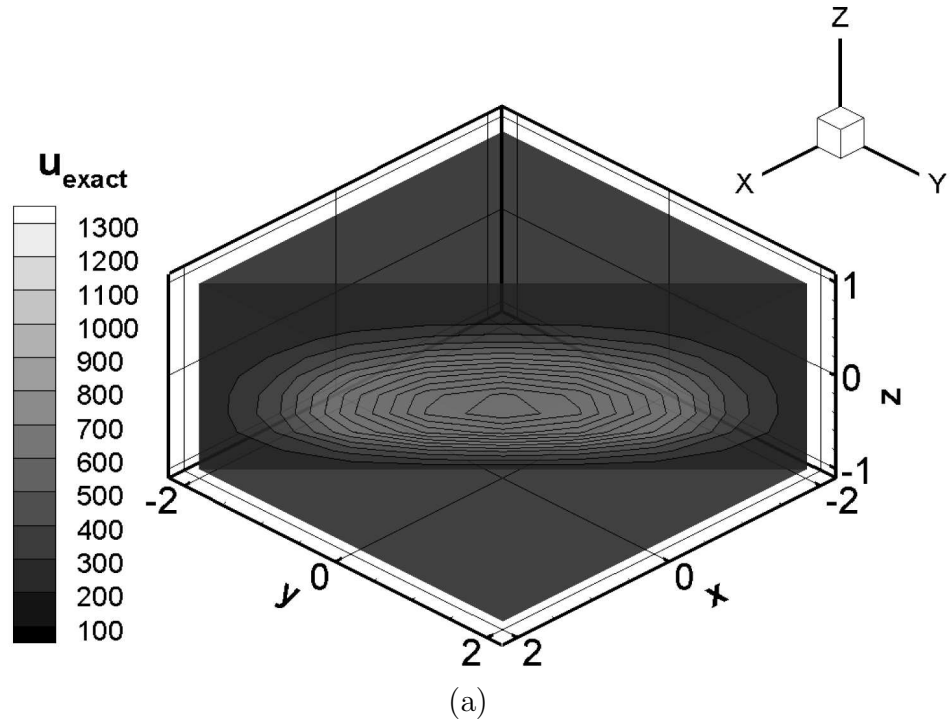


Figure 6.8: Exact Solution and Finite Element Error for Example 3 at $t = 10s$
 (a) Exact solution
 (b) Point-wise error with $p_{ip} = r_{ip} = 4$, $p_t = r_t = 1$, and $p_{tk} = r_{tk} = 6$

this example, iteration convergence to less than a tenth of a degree is achieved within five or six iterations.

The error convergence results for example three are shown in Figure 6.9. In (a), (b), (c), and (d) the x axis is h , the size of an in-plane element, and the y axis is the H^1 error norm at $t = 10s$ or the L^2 error norm. In order to isolate the error due to the in-plane and through-thickness approximations, the degree of the time approximation and interpolation was set equal to that of the exact solution, $p_t = r_t = 1$. The finite element approximation is then exact in time. Since there is no error in time, the in-plane and through-thickness temperature approximations and interpolations are the only sources of error in the finite element approximation.

In Figure 6.9 (a), the H^1 error versus the size of an in-plane element at $t = 10s$ is shown on a log-log scale. Each line represents different value for p_{ip} . In order to isolate the error due to the in-plane approximation, the degree of through-thickness approximation and interpolation was set equal to that of the exact solution, $p_{tk} = r_{tk} = 6$. The in-plane interpolation degree, r_{ip} , is set equal to the in-plane temperature approximation degree, p_{ip} . The convergence rates for the in-plane approximation are either equal to or slightly better than the convergence rates expected for a linear problem. In Figure 6.9 (b), the L^2 error versus h on a log-log scale is shown for varying in-plane degrees of temperature approximation. Again, the achieved convergence rates are either equal to or slightly better than the convergence rates expected for a linear problem.

In Figure 6.9 (c) and Figure 6.9 (d), the H^1 error at $t = 10s$ and the L^2 error versus h are shown on a log-log scale. The in-plane degree of temperature approximation and interpolation are set equal to three in both plots. Each line represents a different value for p_{tk} with $r_{tk} = p_{tk}$. There is no mesh refinement in the through-thickness direction since the finite element method collapsed the through-thickness approximation onto the

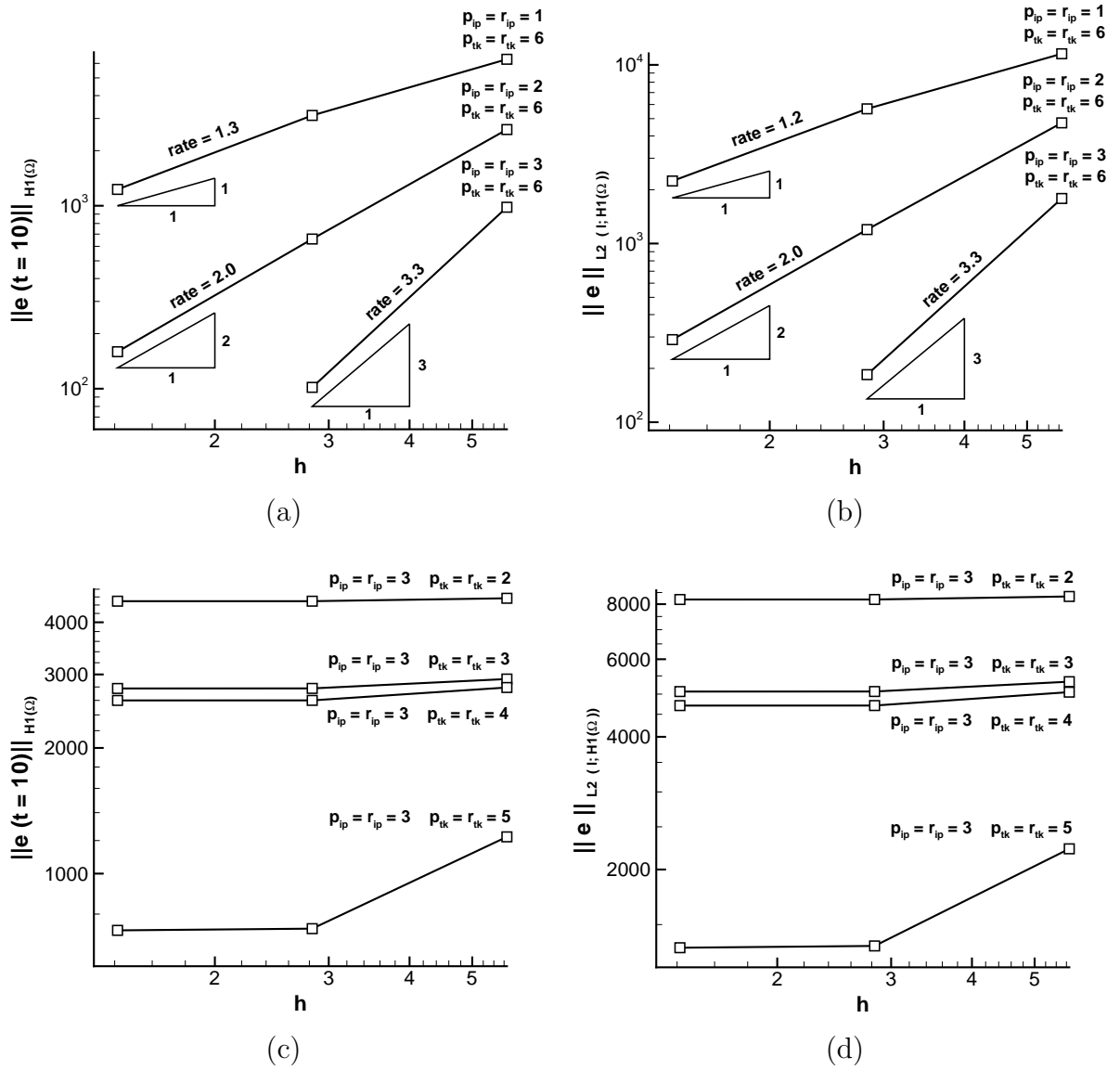


Figure 6.9: Convergence of the Error for Example 3

$$\|u(t=10)\|_{H^1(\Omega)} \approx 6509.0 \quad \|u\|_{L^2(I; H^1(\Omega))} \approx 11883.7$$

(a) H^1 Error for Different Values of p_{ip} (b) L^2 Error for Different Values of p_{ip}

(c) H^1 Error for Different Values of p_{tk} (d) L^2 Error for Different Values of p_{tk}

in-plane mesh. This means that refining the mesh does not improve the through-thickness approximation. Since the exact solution for example three is a fourth degree polynomial in the in-plane direction, the in-plane approximation and interpolation are sources of error in these plots. As the mesh is refined, h gets smaller and the error in the in-plane approximation decreases. The lines for $p_{tk} = 2, 3,$ and 4 are nearly flat. Since the in-plane approximation is being refined, this means that the through-thickness approximation is the dominate source of error for these lines. For $p_{tk} = 5$, note the decrease in the error as h is decreased from 5.67 to 2.83. Over this interval, the in-plane approximation is the dominate source of error. As h is decreased from 2.83 to 1.41 for $p_{tk} = 5$, the dominate source of error is the through-thickness approximation.

Again, the H^1 norm at $t = 10s$ and the L^2 norm of the exact solution are given in the caption of Figure 6.4. Using these values, (6.6), and (6.7), a percent error can easily be calculated.

6.3.4 Example 4

This example explores the error when the thermal conductivity is a quadratic function of the temperature. The exact solution for example four is chosen to be

$$u = 100 \left(1 - \frac{x}{2}\right)^2 \left(1 + \frac{x}{2}\right) t^2 \quad (6.28)$$

The domain for this problem is a 4×4 sheet lying in the $x - y$ plane with its center at the origin. The thermal conductivity for this problem is chosen to be

$$k(u) = 0.00005u^2 + 1.0 \quad (6.29)$$

The thermal capacitance is chosen to be

$$\rho c_p = 175 \quad (6.30)$$

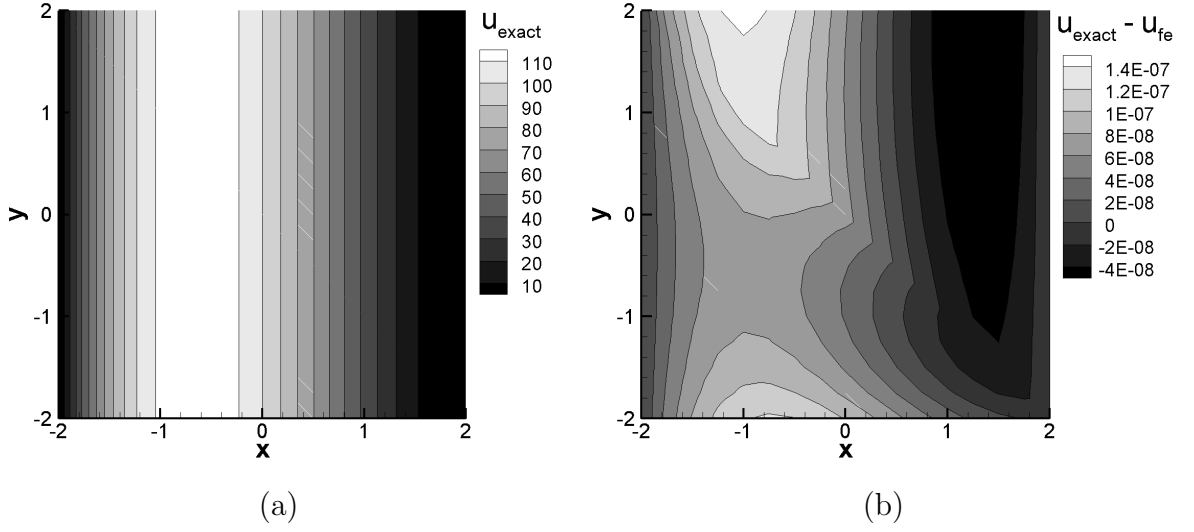


Figure 6.10: Exact Solution and Finite Element Error for Example 4 at $t = 1s$

(a) Exact solution

(b) Point-wise error with $p_{ip} = 3$, $r_{ip} = 6$, $p_t = 2$ and $r_t = 4$

The time interval is chosen to run from $t = 0s$ until $t = 1s$. The initial and boundary conditions for $u(x, y, t)$ are

$$\begin{aligned}
 u(-2, y, t) &= u(2, y, t) = 0 \\
 q_s(x, -2, t) &= q_s(x, 2, t) = 0 \\
 u(x, y, 0) &= 0
 \end{aligned} \tag{6.31}$$

The source term for example four is calculated by substituting (6.28) and (6.29) into (6.15). The temperature distribution for the exact solution at $t = 1s$ is shown in Figure 6.10 (a).

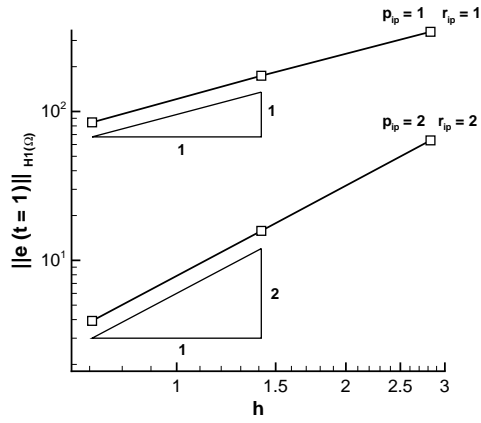
For this example, the thermal conductivity is a quadratic function of the temperature. Since ndgSCHTp uses linear interpolation between the tabulated values, error associated with the linear interpolation between tabulated values is introduced into the finite element approximation.

The exact solution is a product of a third degree polynomial in the in-plane direction

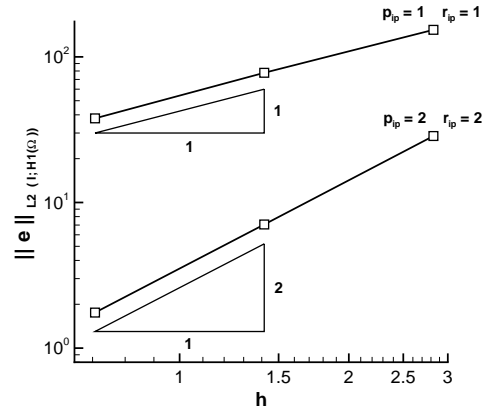
and a second degree polynomial in time. Since the thermal conductivity is a quadratic function of temperature, the interpolation used to account for the temperature-dependent thermal conductivity must reflect its quadratic dependence on the temperature. To accurately capture the quadratic variation in the thermal conductivity as a function of temperature, the degrees of interpolation should be twice the degrees of the exact temperature approximation. Then the only error in the finite element approximation is the error associated with the linear interpolation between the tabulated values of the thermal conductivity. By tabulating the values of thermal conductivity every quarter of a degree, this error can be minimized. Therefore, ndgSCHTp should nearly reproduce the exact solution when $p_{ip} = 3$, $r_{ip} = 6$, $p_t = 2$, and $r_t = 4$ using just one time step as shown in the point wise error plot in Figure 6.10 (b).

The error convergence results for example four are shown in Figure 6.11. In (a), (b), (c), and (d) the x axis is h and the y axis is the H^1 error at $t = 1s$ or the L^2 error. In order to isolate the error due to the in-plane approximation, the degree of time approximation and interpolation was set equal to that of the exact solution, $p_t = 2$ and $r_t = 4$. The finite element approximation is then exact in time. Since there is no error in time, the in-plane temperature approximation and interpolation are the only sources of error besides the tabulation of the thermal conductivity.

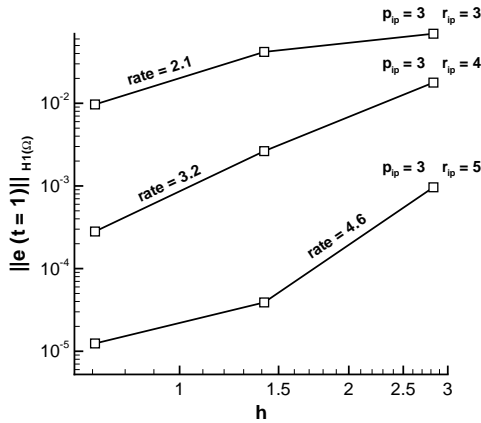
In Figure 6.11 (a), the H^1 error at $t = 1s$ versus h is shown on a log-log scale. Each line represents a different value for p_{ip} . The in-plane interpolation degree, r_{ip} , is set equal to the in-plane temperature approximation degree, p_{ip} . Even though this problem has a quadratic function for the thermal conductivity, r_{ip} is set equal to p_{ip} and not double p_{ip} because the temperature approximation will be the dominant source of error. The triangles show the slope of the lines on the log-log plot and hence the theoretical rates of convergence for linear problems. The convergence rates for this nonlinear example



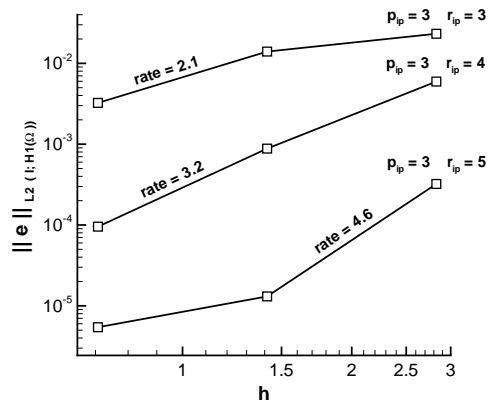
(a)



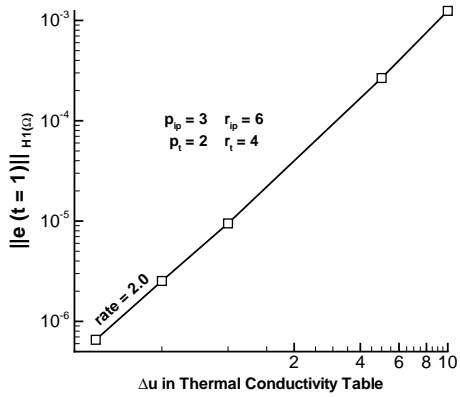
(b)



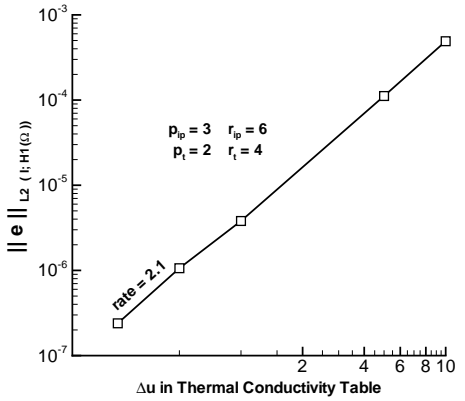
(c)



(d)



(e)



(f)

Figure 6.11: Convergence of the Error for Example 4

$$\|u(t=1)\|_{H^1(\Omega)} \approx 604.7 \quad \|u\|_{L^2(I; H^1(\Omega))} \approx 270.4$$

- (a) H^1 Error for Different Values of p_{ip}
- (b) L^2 Error for Different Values of p_{ip}
- (c) H^1 Error for Different Values of r_{ip}
- (d) L^2 Error for Different Values of r_{ip}
- (e) H^1 Error for the Tabulation of Thermal Conductivity
- (f) L^2 Error for the Tabulation of Thermal Conductivity

are identical to the convergence rates expected for a linear problem. In Figure 6.11 (b), the L^2 error versus h on a log-log scale is shown for varying in-plane degrees of temperature approximation. The convergence rates for the L^2 error should be the same as the convergence rates for the H^1 error since the L^2 error has an H^1 error norm in space. Again, the achieved convergence rates are equal to the convergence rates for linear problems.

In Figure 6.11 (c) and 6.11 (d), the H^1 error at $t = 1s$ and the L^2 error versus h , respectively, are shown on log-log scales. Each line represents different value for r_{ip} with p_{ip} always equal to three, the degree of the exact solution. This isolates the in-plane interpolation so that it is the main source of error in the approximation. The rate of convergence for the in-plane interpolation is written above each line in the plots. Note that for the lines in each plot representing $r_{ip} = 5$, the error due to the tabulation in the thermal conductivity starts to dominate the approximation as the mesh is refined resulting in a slower convergence rate.

In Figure 6.11 (e) and (f), the H^1 error at $t = 1s$ and the L^2 error, respectively, are shown on log-log scales. The x axis is the space between tabulated values of temperature in the thermal conductivity table. The degree of the in-plane temperature approximation and interpolation are set so that they are exact, $p_{ip} = 3$ and $r_{ip} = 6$. As a result, these plots represent the error in the finite element approximation due solely to the tabulation of the thermal conductivity. The line shows the convergence of the error as the space between temperature values in the thermal conductivity table is decreased. The rate of convergence, written above the line in the plots, is equal to the rate of convergence expected for a quadratic function approximated by a piecewise linear function.

Again, the H^1 error at $t = 1s$ and the L^2 error of the exact solution are given in the caption of Figure 6.11. Using these values, (6.6), and (6.7), a percent error can easily be

calculated.

6.3.5 Example 5

Since real problems of transient heat conduction do not usually have simple polynomial solutions, example five is used to validate the temperature approximation and interpolation in time for a more complicated and realistic solution. This example also provides validation of the approach for applying the non-constant initial condition. The exact solution for example five is chosen to be

$$u = 4000 \left(1 - \frac{x}{2}\right) \left(1 + \frac{x}{2}\right) \left(1 - \frac{y}{2}\right)^2 \left(1 + \frac{y}{2}\right) e^{-2t} \quad (6.32)$$

The domain for this problem is a 4×4 sheet lying in the $x - y$ plane with its center at the origin. The thermal conductivity for this problem is chosen to be

$$k(u) = 0.005u + 1.0 \quad (6.33)$$

The thermal capacitance for example five is chosen to be

$$\rho c_p = 175 \quad (6.34)$$

The time interval is chosen to run from $t = 0s$ until $t = 1s$. The initial and boundary conditions for $u(x, y, t)$ are

$$\begin{aligned} u(-2, y, t) &= u(2, y, t) = 0 \\ u(x, -2, t) &= u(x, 2, t) = 0 \end{aligned} \quad (6.35)$$

$$u(x, y, 0) = 4000 \left(1 - \frac{x}{2}\right) \left(1 + \frac{x}{2}\right) \left(1 - \frac{y}{2}\right)^2 \left(1 + \frac{y}{2}\right)$$

The source term for example five is calculated in the same manner as before, this time by substituting (6.32) and (6.33) into (6.15). The temperature distribution for the exact solution at $t = 1s$ is shown in Figure 6.12 (a).

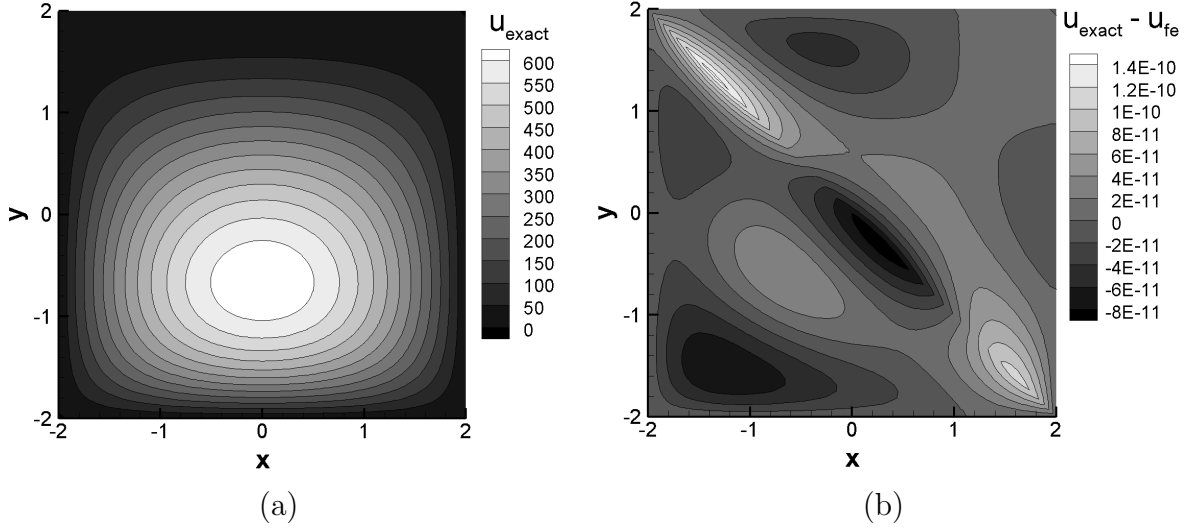


Figure 6.12: Exact Solution and Finite Element Error for Example 5 at $t = 1s$

(a) Exact solution

(b) Point-wise error with $p_{ip} = r_{ip} = 5$ and $p_t = r_t = 6$ using five time steps

Again, since the thermal conductivity is a linear function of temperature and ndgSCHTp uses linear interpolation, there is no error in the solution due to the tabulation of the thermal conductivity.

For example five, the exact solution is a product of a fifth degree polynomial in the in-plane direction and a exponential function in time. An exponential function can be represented by an infinite polynomial series. This means that any finite polynomial series is an approximation of the infinite series. However, by combining a sufficiently high degree of approximation and several time steps, an accurate approximation should be obtained. The point-wise error for a sixth degree approximation in time using five time steps is shown Figure 6.12 (b). The minimal error in Figure 6.12 (b) shows the accuracy of the approximation.

The error convergence results for example five are shown in Figure 6.13. The x axis is the number of time steps and the y axis is the L^2 error norm. In order to isolate the error in the approximation in time, the in-plane degree of approximation and interpolation was

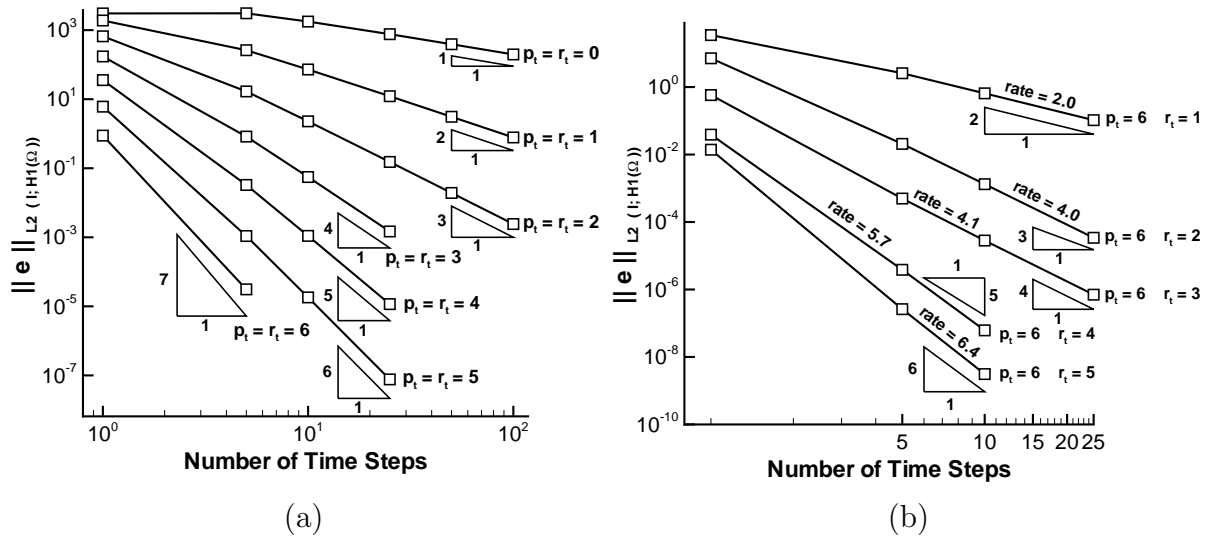


Figure 6.13: Convergence of the L^2 Error for Example 5 $\|u\|_{L^2(I; H^1(\Omega))} \approx 10105.4$
(a) The Effect of Different Values of p_t (b) The Effect of Different Values of r_t

set equal to that of the exact solution, $p_{ip} = r_{ip} = 5$. The finite element approximation is then exact in the in-plane direction. Therefore the temperature approximation and interpolation in time are the only sources of error in the finite element approximation.

In Figure 6.13 (a), the L^2 error versus the number of time steps is shown on a log-log scale. Each line represents different value for p_t . The interpolation degree in time, r_t , is set equal to the temperature approximation degree in time, p_t . The triangles show the slope of the lines on a log-log plot for the theoretical rates of convergence for linear problems. The convergence rates for this realistic, nonlinear example are identical to the convergence rates expected for a linear problem.

In Figure 6.13 (b), the L^2 error versus the number of time steps is shown on a log-log scale. Each line represents a different value for r_t with p_t always equal to six. This minimizes the temperature approximation error in time so that the interpolation in time is the main source of error in the approximation. The rate of convergence is written above each line in the plot. This figure shows the error due to the interpolation of the

thermal conductivity in time. The triangles serve as a reference to compare with the slopes of the lines.

Again, the L^2 error norm of the exact solution is given in the caption of Figure 6.13. Using this value and (6.7), a percent error can be calculated.

6.3.6 Example 6

This example is used to validate the in-plane temperature approximation and interpolation methods for a more complicated and realistic solution. This example also provides validation of the approach for applying the non-constant initial condition. The exact solution for example six is chosen to be

$$u = 68 \left(1 + \cos \frac{\pi x}{2}\right) \left(1 + \cos \frac{\pi y}{2}\right) e^{-t} \quad (6.36)$$

The domain for this problem is a 4×4 sheet lying in the $x - y$ plane with its center at the origin. The thermal conductivity for this problem is chosen to be

$$k(u) = 0.005u + 1.0 \quad (6.37)$$

The thermal capacitance for example six is chosen to be

$$\rho c_p = 175 \quad (6.38)$$

The time interval is chosen to run from $t = 0s$ until $t = 1s$. The initial and boundary conditions for $u(x, y, t)$ are

$$\begin{aligned} u(-2, y, t) &= u(2, y, t) = 0 \\ u(x, -2, t) &= u(x, 2, t) = 0 \end{aligned} \quad (6.39)$$

$$u(x, y, 0) = 68 \left(1 + \cos \frac{\pi x}{2}\right) \left(1 + \cos \frac{\pi y}{2}\right)$$

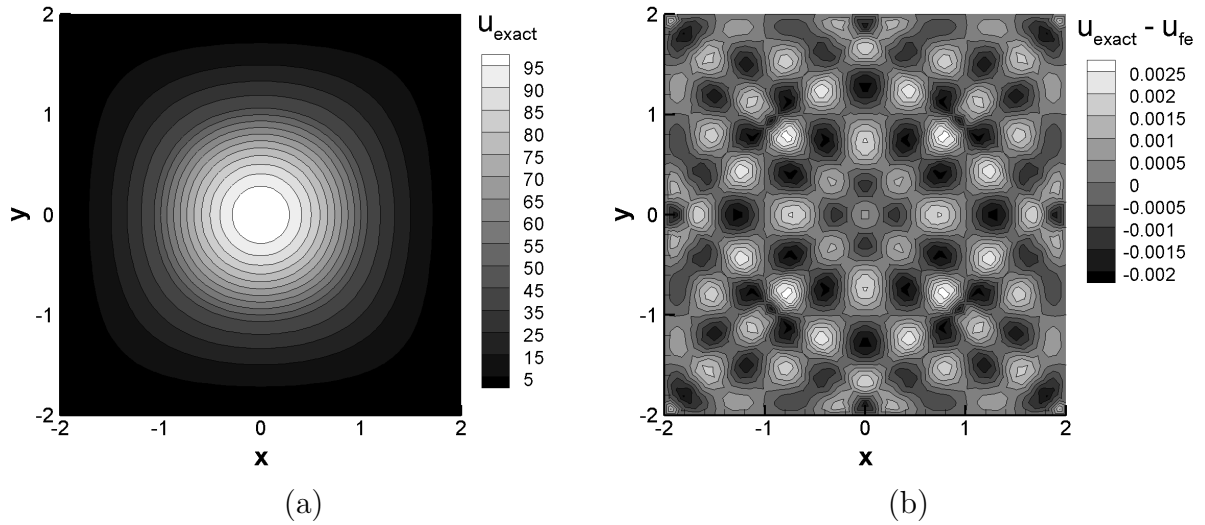


Figure 6.14: Exact Solution and Finite Element Error for Example 6 at $t = 1s$

(a) Exact solution

(b) Point-wise error using a 16 element mesh and 5 time steps with $p_{ip} = r_{ip} = 6$ and $p_t = r_t = 4$

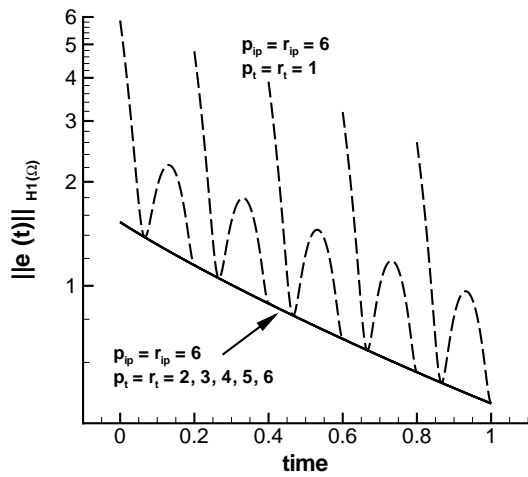
The source term for example six is calculated by substituting (6.36) and (6.37) into (6.15).

The temperature distribution for the exact solution at $t = 1s$ is shown in Figure 6.14

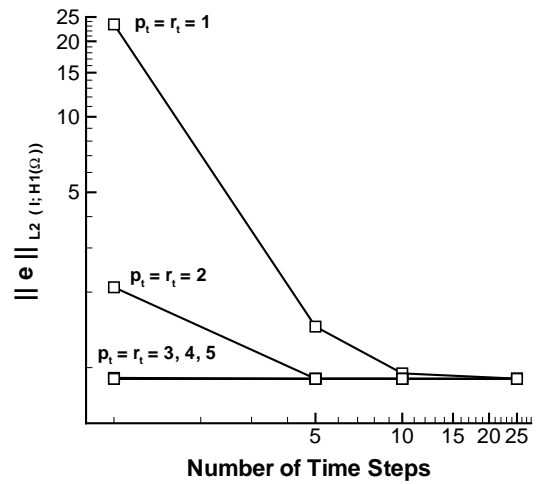
(a).

For example six, the exact solution is a product of a trigonometric function in the in-plane direction and an exponential function in the time dimension. Both trigonometric and exponential functions can be represented by an infinite polynomial series. This means that any finite polynomial series is an approximation of the infinite series. However, by combining a sufficiently high degree of approximation with several elements, an accurate approximation should be obtained. The point-wise error for a sixth degree in-plane approximation and a fourth degree time approximation using 16 elements and 5 time steps is shown Figure 6.14 (b). The point-wise error in this approximation is larger than in previous examples, but certainly within engineering accuracy.

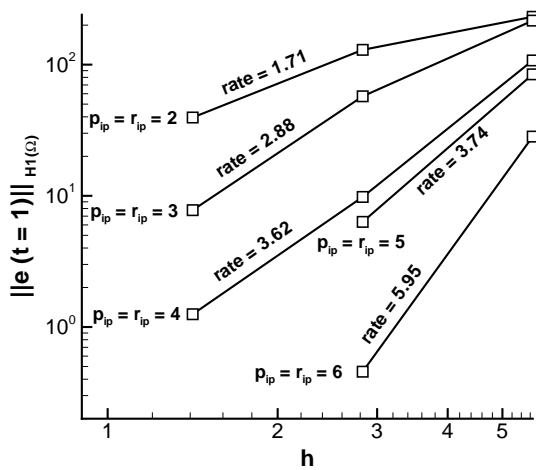
The H^1 error versus time on a linear-log scale is shown in Figure 6.15 (a). This



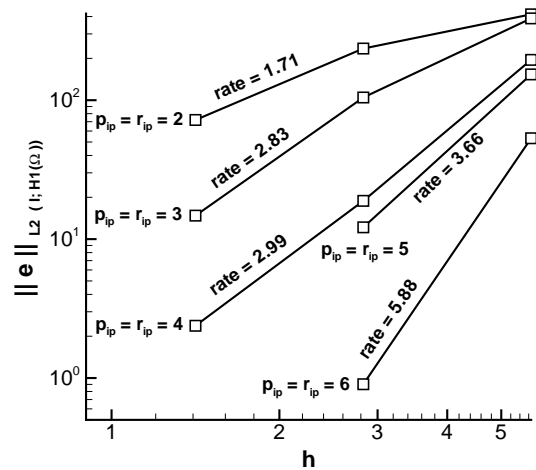
(a)



(b)



(c)



(d)

Figure 6.15: Convergence of the Error for Example 6

$$\|u(t=1)\|_{H^1(\Omega)} \approx 345.2 \quad \|u\|_{L^2(I, H^1(\Omega))} \approx 617.0$$

(a) H^1 Error as a Function of Time(b) L^2 Error for Different Values of p_t (c) H^1 Error for Different Values of p_{ip} (d) L^2 Error for Different Values of p_{ip}

figure shows several different approximations in time, $p_t = 1$ through $p_t = 6$, with all approximations using five time steps. An eight element mesh with $p_{ip} = r_{ip} = 6$ was used for the in-plane approximation. For $p_t = 1$ at each time step, there are two dips and the discontinuity in the approximation occurs between each of the five time steps showing the characteristics of the discontinuous Galerkin method. Unlike the results presented in Figure 6.6 from example two, the error is decreasing in time. This is a result of the dominance of the error in the in-plane approximation. Since the exact solution for this problem is continuous in time, the continuous line in the figure for higher-degrees of approximation in time shows that the error in time is minimal compared to the in-plane error. For an approximation in time of degree two or higher, the in-plane approximation is totally dominating the error and the characteristics of the discontinuous Galerkin method disappear. Since the error in time approximation is insignificant compared to the error in the in-plane approximation, the characteristics of the time error are not evident.

The L^2 error versus the number of time steps for several different degrees of approximation in time is shown in Figure 6.15 (b) on a log-log scale. An eight element mesh with $p_{ip} = r_{ip} = 6$ was used for the in-plane approximation. Note that the lines for $p_t = 3$, 4, and 5 are nearly flat. Since the approximation in time is being refined as the number of time steps is increased, this means that the in-plane approximation is the dominate source of error for these lines. Note that the lines for $p_t = 1$ and $p_t = 2$ from one time step to five time steps show a significant decrease in the L^2 error. A significant decrease in the L^2 error is also evident for $p_t = 1$ from five time steps to ten time steps. For these approximations, the time approximation is the dominant source of error, not the in-plane approximation. The results shown in Figure 6.15 (a) are supported by Figure 6.15 (b). The characteristics of the discontinuous Galerkin method using five time steps were evident for $p_t = 1$ only in Figure 6.15 (a). From Figure 6.15 (b), it is evident that

for approximations in time higher than degree one using five time steps, the in-plane approximation is dominating the error.

In Figure 6.15 (c), the H^1 error versus h at $t = 1s$ is shown on a log-log scale. Each line represents different value for $p_{ip} = r_{ip}$. For this plot, $p_t = r_t = 6$ and five time steps were used for the approximation in time. The convergence rates for the in-plane approximation are all less than the convergence rates expected for a linear problem. These reduced convergence rates are a result of the error in the approximation in time. Although the in-plane approximation is clearly dominating the error as shown in Figure 6.15 (b), the approximation in time is contributing to the error and slowing the rate of convergence for the in-plane approximations.

The slowing of the in-plane convergence rates due to the error in the approximation in time is supported by Figure 6.15 (d). In this figure, the L^2 error versus the mesh size is shown on a log-log scale. The same degrees of approximation in time and space along with the same mesh and number of time steps from Figure 6.15 (c) were used in this figure. Note the decreased rates of convergence in Figure 6.15 (d) compared to Figure 6.15 (c). These decreased rates are a result of the integration over the error in time for the L^2 error as compared to the H^1 error which is just evaluated at a point in time.

From the sample problems presented in this research, the linear convergence rates are valid for nonlinear problems only when a single source of error in the approximation can be isolated. As shown in the first five examples, when the in-plane or time approximation is isolated, the linear convergence rates apply to these nonlinear problems. In example six, the error convergence rates are less than those expected for linear problems because there is error in both the in-plane and time approximations.

Chapter 7

Concluding Remarks

Finite element methods are used to accurately predict the thermal response of a structure during reentry and hypersonic travel through the atmosphere. Many commercial finite element codes currently rely on separate meshes for structural and thermal analyses. These codes also rely upon mesh regeneration to improve the solution using h-version or traditional p-version elements. The commercial codes use finite element methods in space and finite difference schemes in time. Finite difference schemes in time require small time steps for accuracy and stability. The conduction heat transfer problem is nonlinear due to the variation in the thermal conductivity with temperature. Current finite element methods for nonlinear problems utilize one of two approaches. Either the thermal conductivity is assumed to be constant over each element or the finite element integrals containing the thermal conductivity are integrated using Gaussian quadrature.

A hierarchical p-version space-time finite element method with a structurally compatible mesh for nonlinear, transient problems was developed and tested in this research. Hierarchical, p-version finite elements allowed higher degrees of approximation on each element and eliminated the need for new meshes as the degree of approximation is increased. Three different sets of hierarchical basis functions, one each for the in-plane, time, and through-thickness dimensions, were used to approximate the temperature distribution over an element. The hierarchical modelling also allowed a three-dimensional

domain to be collapsed onto a two-dimensional, structurally-compatible mesh. The p-version discontinuous Galerkin method allows higher degrees of approximation in time and provides stability regardless of the size of the time step. Higher-degree interpolants were used over each element to accurately capture the variation in the thermal conductivity. The interpolants utilized three different sets of interpolation functions, one each for the in-plane, time, and through-thickness dimensions, to account for the temperature-dependent thermal conductivity. The finite element method presented in this research eliminates many of the limitations of commercial finite element codes. The finite element formulation was tested in a FORTRAN code called ndgSCHTp. To increase the efficiency of ndgSCHTp, master matrices were utilized. The FORTRAN code was validated for nonlinear problems constructed to have exact solutions so the error between the finite element approximation and the exact solution could be quantified.

The finite element method presented in this research provided accurate solutions to nonlinear, transient problems of conduction heat transfer. For simple polynomial exact solutions in space and time, ndgSCHTp reproduces the exact solutions to within machine precision. Accurate solutions were also obtained for non-polynomial exact solutions. The error due to the tabulation of the thermal conductivity was explored and its effect on the finite element approximation was shown. The H^1 and L^2 error convergence rates for these nonlinear problems were shown to agree with the theoretical convergence rates for linear problems when a single source of error could be isolated. The finite element method developed was shown to be very accurate for transient, nonlinear problems of conduction heat transfer.

There are several areas of future research that should be explored relating to this work. Problems with complicated domains should be explored. These sample problems would show the robustness of the method presented here and expose any possible problems. The

finite element formulation presented in this work has been expanded to include nonlinear specific heat as a function of temperature, $c_p(u)$, using the same interpolation method used for the thermal conductivity. This formulation should be implemented and tested using a computer code. Layered structures consisting of several different materials are commonly used in thermal protection systems. The current finite element formulation should be modified to account for layered structures. The method presented by Lang [8] for layered structures would be difficult to modify for temperature-dependent thermal conductivity, so a new method to handle layered materials should be developed. The finite element formulation should also be extended to include materials with anisotropic conductivity.

References

- [1] The MacNeal Schwendler Corporation, *NASTRAN User's Manual*, 1995.
- [2] Abaqus Inc., *ABAQUS / Standard User's Manual*.
- [3] Surana, K.S. and Orth, N.J., "Completely Hierarchical p-Version Axisymmetric Shell Element for Nonlinear Heat Conduction in Laminated Composites," *Computers and Structures*, Vol.46, No.5, 1993, pp. 777-789.
- [4] Szabó, B. and Babuška, I., *Finite Element Analysis*, John Wiley & Sons Inc., New York, 1981.
- [5] Tamma, K. and Saw, K., "Hierarchical p-Version Finite Elements and Adaptive a Posteriori Computational Formulations for Two-Dimensional Thermal Analysis," *AIAA Paper 89-0518*, 1989.
- [6] Gould, D., "Hierarchical p-Version Finite Elements for Radiation Heat Transfer," *Ph.D. Dissertation*, University of Virginia, 1999.
- [7] Tomey, J. P., "p-Version Discontinuous Galerkin Method for Heat Transfer in Built-up Structures," *M.S. Thesis*, The George Washington University, 2001.
- [8] Lang, C. G., "Finite Element *A Posteriori* Error Estimation for Heat Conduction," *NASA CR-2002-211958*, November 2002.

- [9] Walker, D. T., “Nonlinear Conduction Heat Transfer Using a Hierarchical Finite Element Method,” M.S. Thesis, The George Washington University, 2003.
- [10] Schotzau, D., Scwab, C., “An hp A-Priori Error Analysis of the DG Time-Stepping Method for Initial Value Problems,” IMA Research Report 1652, University of Minnesota, 1999.
- [11] Schotzau, D., Scwab, C., “Time Discretization of Parabolic Problems by the hp-Version of the Discontinuous Galerkin Finite Element Method,” SIAM Journal of Numerical Analysis, Vol. 38, No. 3, 2000, pp. 837-875.
- [12] Eriksson, K., Johnson, C., Larsson, S., “Adaptive Finite Elements for Parabolic Problems VI: Analytic Semigroups,” SIAM Journal of Numerical Analysis, Vol. 35, No. 4, 1998, pp. 1315-1325.
- [13] Reddy, J. N., *An Introduction to the Finite Element Method, Second Edition*, McGraw-Hill, Boston, 1993.
- [14] Boas, Mary L., *Mathematical Methods in the Physical Sciences, Second Edition*, John Wiley & Sons, New York, 1983.
- [15] Hughes, Thomas J.R., *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Prentice-Hall, Englewood Cliffs, New Jersey, 1987.
- [16] Schötzaue, D. and Schwab, C., “Time Discretization of Parabolic Problems by the HP-Version of the Discontinuous Galerkin Finite Element Method,” SIAM Journal on Numerical Analysis, Vol.38, No.3, 2000, pp. 837-875.
- [17] Chapra, S. C. and Canale, R. P., *Numerical Methods for Engineers*, WCB McGraw-Hill, Boston, 1998.

- [18] Myint-U, T., Debnath, L., *Partial Differential Equations for Scientists and Engineers*, Third Edition, P T R Prentice Hall, Englewood Cliffs, New Jersey, 1987.
- [19] Zienkiewicz, O. C., Morgan, K., *Finite Elements and Approximation*, John Wiley & Sons, New York, 1983.
- [20] Mish, K., “Two-Dimensional Finite Elements,” U.C.Davis Course Notes ECI212A, <http://cee.engr.ucdavis.edu/Faculty/mish/212A/2dfems.pdf> ¹
- [21] Young, D. M., Gregory, R. T., *A Survey of Numerical Mathematics In Two Volumes*, Dover Publications, Inc., New York, 1972.
- [22] Rivlin, T. J., *Chebyshev Polynomials: From Approximation Theory to Algebra and Number Theory*, John Wiley & Sons, New York, 1990.
- [23] Taylor, M. A., Wingate, B. A. and Vincent, R. E., “An Algorithm for computing Fekete Points In The Triangle,” *SIAM Journal*, Vol. 38 No. 5, pp 1707-1720, 2000.
- [24] Bey, K. S., “The SCHAT Code, A User and Programmer’s Manual,” Metals and Thermal Structures Branch internal document, NASA Langley Research Center, 2002.
- [25] Dunavant, D. A., “High Degree Efficient Symmetrical Gaussian Quadrature Rules for the Triangle,” *International Journal for Numerical Methods in Engineering*, Vol. 21, 1989, pp. 1129-1148.
- [26] Werder, T., Gerdes, K., Schötzau, D., Schwab, C., “hp Discontinuous Galerkin Time Stepping for Parabolic Problems,” Research Report No. 2000-01 April 2000, Seminar for Applied Mathematics, ETH Zurich, 2000.

[27] Becker, E., Carey, G., Oden, J., *Finite Elements: An Introduction, Volume I*,
Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1981.

¹ Contact the author for copies of World Wide Web references: jcsand@copper.net,
jeromecsanders@yahoo.com