A MULTI-SCALE GENERALIZED FINITE ELEMENT METHOD FOR SHARP, TRANSIENT THERMAL GRADIENTS

BY

PATRICK JAMES O’HARA

DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Civil Engineering in the Graduate College of the University of Illinois at Urbana-Champaign, 2010

Urbana, Illinois

Doctoral Committee:

Assistant Professor C. Armando Duarte, Chair and Director of Research
Dr. Thomas G. Eason III, Wright-Patterson Air Force Base
Professor Philippe H. Geubelle
Professor Narayana R. Aluru
Abstract

In this research, heat transfer problems exhibiting sharp thermal gradients are analyzed using the generalized finite element method. Convergence studies show that low order (linear and quadratic) elements require strongly refined meshes for acceptable accuracy. The high mesh density leads to small allowable time-step sizes, and significant increase in the computational cost. When mesh refinement and unrefinement is required between time-steps the mapping of solution vectors and state-dependent variables becomes difficult.

A generalized FEM with global-local enrichments is proposed for the class of problems investigated in this research. In this procedure, a global solution space defined on a coarse mesh is enriched through the partition of unity framework of the generalized FEM with solutions of local boundary value problems. The local problems are defined using the same procedure as in the global-local FEM, where boundary conditions are provided by a coarse-scale global solution. Coarse, uniform, global meshes are acceptable even at regions with thermal spikes that are orders of magnitude smaller than the element size. Convergence on these discretizations is achieved even when no or limited convergence is observed in the local problems.

The two-way information transfer provided by the proposed generalized FEM is appealing to several classes of problems, especially those involving multiple spatial scales. The proposed methodology brings the benefits of generalized FEM to problems where limited or no information about the solution is known a-priori.

The proposed methodology is formulated for, and applied to transient problems, where local domains at time $t^{n+1}$ obtain their boundary conditions from the global domain at $t^n$. No transient effects need to be considered in the local domain. The method has shown the ability to produce
accurate and efficient transient simulations in situations where traditional FEM analyses would lead to difficult re-meshing, and solution mapping issues.

With the proposed methodology, the enrichment functions are added hierarchically to the stiffness matrix. As such, large portions of the coarse, global meshes may be assembled and factorized only once. The factorizations can then be re-used for multiple loading scenarios, or multiple time-steps so as to significantly improve the computational efficiency of the simulations.

The issue of prohibitively small time-step sizes dictated by high mesh density in traditional FEM analyses is also addressed. With the use of appropriate shape functions, sufficient accuracy is obtained without the requirement of highly refined meshes. The resulting critical time-steps are less restrictive, making transient analyses more computationally feasible.
To my family, Meredith, Brigid and Seamus
Acknowledgments

I would like to thank my advisor, Professor C. Armando Duarte for providing me with the opportunity to seek my Ph.D. under his guidance at the University of Illinois at Urbana-Champaign. I would also like to thank my committee, Professor Philippe H. Geubelle, Professor Narayana R. Aluru, and in particular, Dr. Tom Eason from AFRL; for the feedback and suggestions that they have given me.

I would like to thank my family and friends for all of the support that they have given me. In particular, I would like to thank Jerônimo P. A. Pereira for his keen insight, and endless willingness to provide assistance wherever possible.

Finally, I gratefully acknowledge the contributions of the Midwest Structural Sciences Center (MSSC). The Center is supported by the U.S. Air Force Research Laboratory Air Vehicles Directorate under contract number FA8650-06-2-3620.
# Table of Contents

List of Tables ................................................................. ix

List of Figures .............................................................. xi

Chapter 1 Introduction ...................................................... 1
  1.1 Motivation and Applications ........................................ 1
  1.2 Related Methods ...................................................... 6
    1.2.1 Multi-Scale Techniques ........................................ 6
    1.2.2 Transient Analysis and Time-Dependent Shape Functions .... 15
  1.3 Research Goals ....................................................... 29

Chapter 2 Steady-State Heat Conduction ............................... 31
  2.1 Problem Formulation .................................................. 31
  2.2 Generalized FEM Approximations ................................... 32
  2.3 Model Problem ........................................................ 34
    2.3.1 Convergence Analysis .......................................... 36
    2.3.2 GFEM with Special Enrichment Functions ................. 43
  2.4 Generalized Finite Element Analysis with Global-Local Enrichments .... 44
    2.4.1 Formulation of Global Problem ............................... 45
    2.4.2 Local Problems .................................................. 45
    2.4.3 Global-Local Enrichment Functions ......................... 46
  2.5 Analysis of Model Problem Using the $GFEM^g$ .................. 50
    2.5.1 $H$-extensions in the Local Problem ....................... 53
    2.5.2 A Two-Step Approach to Improve Local Solutions ....... 60
    2.5.3 Spike Absent from Initial Global Problem ................ 63
    2.5.4 Effect of Enriching the Global Problem .................... 65
  2.6 Summary .............................................................. 67

Chapter 3 Transient Heat Conduction .................................. 71
  3.1 Problem Formulation ................................................ 71
List of Tables

2.1 Summary of Output for Exponential Enrichment Functions. . . . . . . . . . . . . . . 43
3.1 Summary of Output for Polynomial Elements. . . . . . . . . . . . . . . . . . . . . 84
3.2 Comparison of Output for Discretizations With and Without Exponential Enrich-
ment Functions. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 87
3.3 Effect of Volumetric Heat Capacity Magnitude. . . . . . . . . . . . . . . . . . . . 89
3.4 Output for Elements with Time-Dependent Exponential Enrichment Functions. . . 91
3.5 Material Parameters. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 101
3.6 Output for Al Beam. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 109
3.7 Material Composition for Each Case. . . . . . . . . . . . . . . . . . . . . . . . . . 115
3.8 $L_2^{error}$ Values for Each Trial Investigated. . . . . . . . . . . . . . . . . . . . 118
4.1 Summary of $hp$-GFEM Simulations. . . . . . . . . . . . . . . . . . . . . . . . . . 137
4.2 Summary of Computational Effort for $GFEM^{gl}$ Simulations with ReAnalysis. . 137
4.3 Comparison of Energy Values. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 138
4.4 Comparison of CPU Time for 3 Solution Strategies. . . . . . . . . . . . . . . . . . 138
4.5 Direct Comparison of Transient Solution Strategies. . . . . . . . . . . . . . . . . . 152
4.6 Summary of Transient $hp$-GFEM Simulations. . . . . . . . . . . . . . . . . . . . 153
4.7 Summary of Computational Effort for Transient $GFEM^{gl}$ Simulations with Re-
Analysis. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 153
5.1 Summary of Output for Calculation of $\Delta t_{cr}$. . . . . . . . . . . . . . . . . . 161
5.2 Summary of Output for Linear Elements. . . . . . . . . . . . . . . . . . . . . . . 162
5.3 Comparison of $L_2^{error}$ for Different $t_f$. (Linear Elements) . . . . . . . . . . . 164
5.4 Summary of Output for Quadratic Elements. . . . . . . . . . . . . . . . . . . . . . 164
5.5 Summary of Output for Quartic Elements. . . . . . . . . . . . . . . . . . . . . . . 164
5.6 Summary of Output for Exponential Elements. . . . . . . . . . . . . . . . . . . . . 167
5.7 Effect of Spike Location on $\Delta t_{cr}$. . . . . . . . . . . . . . . . . . . . . . . . . 169
5.8 Simulation Data for Varying Magnitudes of $\rho c$. . . . . . . . . . . . . . . . . . 171
5.9 Summary of Simulation Data. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 173
6.1 $\Delta t_{cr}$ bounds for explicit $GFEM^{gl}$ simulations. . . . . . . . . . . . . . . . . . . . . . . . 181
List of Figures

1.1 Shock waves in the flow field of an X-15 at Mach 3.5. (Figure taken from NASA website.) ................................................................. 2

1.2 Potentially representative normal flux on the leading edge of a wing. Distance indicates the distance in inches from the point of maximum flux. ............... 2

2.1 Construction of a generalized FEM shape function using a polynomial (a) and a non-polynomial enrichment (b). Here, $\varphi_\alpha$ is the function at the top, the enrichment function, $L_{\alpha i}$, is the function in the middle, and the generalized FE shape function, $\phi_{\alpha i}$, is shown at the bottom. .......................................................... 33

2.2 Temperature profile of model problem. The solution is smooth everywhere in the domain except in a small neighborhood of $x_0 = 125 \text{ mm}$, where a sharp temperature spike develops. ................................................................. 35

2.3 Typical structure of the locally refined meshes used in the 1-D model. Element size in subsequent plots refers to $h_i$. .................................................. 37

2.4 Locally refined 3-D mesh using a bounding box to define the region of local mesh refinement. ................................................................. 37

2.5 Convergence in energy norm for low order 1-D generalized and $p$-hierarchical finite elements. Quadratic GFEM and $p$-hierarchical FEM deliver the same level of accuracy. Thus, only two curves can be seen in the plot. ............................... 38

2.6 Convergence in energy norm for discretizations with 1-D and 3-D GFEM elements; 1-D $p$-hierarchical elements; and 2-D quadratic Serendipity elements. The curves for 1-D elements coincide. .................................................. 39

2.7 Convergence in energy norm for 3-D linear and quadratic tetrahedral elements. Sequences of meshes locally refined around the thermal spike are used. ........ 40

2.8 Example of a highly graded mesh and the associated well-resolved spike in the temperature field. ................................................................. 40

2.9 Temperature distribution computed on a mesh where the line along the thermal peak cuts the domain at a 45 degree angle. Roughness parameter $\gamma = 0.05$. ......... 41

2.10 Convergence in the energy norm for the two domain orientations considered. In one case, the internal layer is aligned with element edges while in the second case it is not. The roughness parameter $\gamma$ is taken as 0.05. .................. 42
2.11 One-Dimensional GFEM solution computed on a uniform mesh with five quadratic elements. The element containing the thermal spike is enriched with exponential enrichment function (2.10).

2.12 The generalized FEM with global-local enrichment functions. (a) Initial and enriched global problems discretized with a coarse mesh. The initial global problem provides boundary conditions for local problems containing sharp thermal spikes. (b) Local problem used to compute global-local enrichment functions.

2.13 Error of solution computed on a uniform global mesh with 50 elements in the $x-$direction and one element in the $y-$direction. Standard eight-node quadratic finite elements are used. The location of the thermal spike is indicated in the figure. We can observe that the error in the computed temperature is relatively large even far from the thermal spike. Error when the exact solution is given by $u(x) = \sin(\pi x/L)$, i.e., the thermal spike $e^{-\gamma(x-x_0)^2}$ is removed is also provided. In this case, the error of the finite element solution is very small.

2.14 Error of solution computed on a uniform global mesh with 50 elements in the $x-$direction and one element in the $y-$direction. Standard eight-node quadratic finite elements are used. In this case we can see that as $\gamma$ becomes smaller, the error decreases, and also becomes more localized.

2.15 Error of solution computed on a uniform global mesh with 50 elements in the $x-$direction and one element in the $y-$direction. Illustrates that the actual error values are highly dependent upon the description of the loading function.

2.16 Extraction of initial local mesh from global Mesh 1x. The bounding box used for selection of seed nodes is shown (rectangle) along with the seed nodes (solid circles). Nine seed nodes are shown, but there are eighteen in total: Nine on top surface (shown) and nine on the bottom surface of mesh.

2.17 Local mesh extracted from Mesh 1x, and 9 levels of local refinement are used. The rectangle in the figure represents the bounding box used for refinement. (Not to scale)

2.18 Relative error in energy norm for enriched global problems associated with meshes 0x, 1x and 2x. Each curve corresponds to a fixed global mesh and $h$-extensions in the local problems. The CPU time includes the CPU time taken for assembly and solution of the enriched global domain as well as the assembly and solution time taken in the corresponding local domain.

2.19 Relative error in energy norm for local problems subjected to boundary conditions provided by global solutions computed on meshes 0x, 1x and 2x. The CPU time considers the time taken for assembly and solution of the local problem.

2.20 Data of Figure 2.18 plotted against the number of DOFs in the local problems. The plot includes data points that could not be included in Figure 2.18 due to a limitation of the function we use to measure CPU time.
Temperature distributions computed in the initial global, local, and enriched global problems corresponding to global Mesh 0x and 13 levels of refinement in the local problem. The IGw/S methodology is used. The thermal spike is well resolved in the enriched global problem even though it falls within quite large elements. Enriched nodes in global domain are denoted in Figure 2.21(c) by red glyphs.

Temperature distributions computed in the initial global, local, and enriched global problems corresponding to global Mesh 1x and 13 levels of refinement in the local problem. The IGw/S methodology is used. Enriched nodes in global domain are denoted in Figure 2.22(c) by red glyphs.

Temperature distributions computed in the initial global, local, and enriched global problems corresponding to global Mesh 2x and 13 levels of refinement in the local problem. The IGw/S methodology is used. Enriched nodes in global domain are denoted in Figure 2.23(c) by red glyphs.

Figures of local domains with 9 levels of local refinement, and 1 and 6 layers of elements.

Shows the lack of convergence in the energy norm when the local refinement is kept constant, but the number of layers used in creating the local problem is increased. Circed point indicates the data point from Figure 2.20.

Temperature distributions computed in the initial global, local, and enriched global problems corresponding to global Mesh 2x and 13 levels of refinement in the local problem. The IGw/S-II strategy is used. The thermal spike is well resolved in the initial global problem and thus the local problem can also capture well this behavior.

Relative error in energy norm for local problems in strategies IGw/S and IGw/S-II. The only difference in the local problems is the boundary conditions used.

Relative error in energy norm for enriched global problems in strategies IGw/S and IGw/S-II.

Temperature distributions for initial global, local, and enriched global problems for IGw/oS strategy corresponding to global Mesh 2x, 13 levels of refinement in the local problem. Meshes 0x and 1x provide qualitatively similar results. The thermal spike is well resolved in both the local and enriched global problems.

Relative error in energy norm for local problems in strategies IGw/S and IGw/oS. The only difference in the local problems is the boundary conditions used. In the case of IGw/oS strategy, The thermal spike was removed from the initial global domain.

Relative error in energy norm for enriched global problems in strategies IGw/S and IGw/oS.

Comparison of convergence in energy norm in local and enriched global domains for IGw/S strategy.

Comparison of convergence in energy norm in local and enriched global domains for IGw/S-II strategy.
2.34 Data of Figure 2.33 plotted against the number of DOFs in the local problems. The plot includes data points that could not be included in Figure 2.33 due to a limitation of the function we use to measure CPU time.

2.35 Comparison of convergence in energy norm in local and enriched global domains for IGw/oS strategy.

3.1 Temperature field as described in (3.53).

3.2 Initial conditions as described in (3.55).

3.3 Energy versus time plots for quadratic elements.

3.4 Error versus element size for 1D, 2D and 3D simulations.

3.5 Comparison of element sizes for 1-D meshes with and without exponential enrichment.

3.6 Plot of internal energy versus time for the 1-, 2-, and 3-D discretizations with exponential enrichments.

3.7 Plot of internal energy versus time for the special, exponential element. Only a very small portion of the plot is focused at a high level of zooming in order to see a discernible difference in the plots.

3.8 Internal energy versus time curves generated with exponential enrichment functions and increasing values of volumetric heat capacity.

3.9 Internal energy versus time curves generated with 1-D and 3-D meshes enriched with time-dependent exponential enrichment functions and Formulation 1.

3.10 Internal energy versus time curves generated with 1-D time-dependent exponential enrichment functions, large values of \( \rho c \), and Transient Formulation 2.

3.11 Internal energy versus time curves generated with 1- and 3-D time-dependent exponential enrichment functions.

3.12 Illustration of the GFEM\textsubscript{gl} for time-dependent problems. The global domain is discretized with a coarse mesh, even if the solution is not smooth. The coarse-scale problem at \( t^n \) provides boundary conditions for local problems containing sharp thermal spikes. The solution of these problems are used to build approximation spaces for the coarse-scale problem at time \( t^{n+1} \).

3.13 Convergence of \( L_2^{\text{error}}(U(t)) \) with respect to the number of local dofs.

3.14 Plots of internal energy versus time for GFEM\textsubscript{gl} solutions with increasing levels of mesh refinement in local problem. The volumetric heat capacity is taken as \( \rho c = 3.9e-5 \). All curves computed with Transient Formulation 1.

3.15 Plot of internal energy versus time for GFEM\textsubscript{gl} solutions in 3-D for \( \rho c = 3.9e-5 \) and \( \rho c = 10 \). For \( \rho c = 10 \) both transient formulations are investigated. The curve for the exact solution is given by (3.53) and (3.57), and is independent of \( \rho c \).

3.16 Spatial and temporal variation of the Gaussian laser (\( \gamma = 10 \)).

3.17 Global mesh used for the beam model. Flux boundary conditions are denoted by red arrows.
3.18 Relative difference in energy between any two successive \( hp \)-GFEM analyzes. Analysis performed to determine reference value for steady-state internal energy. 103

3.19 Relative difference in energy between any two successive \( hp \)-GFEM solutions, \( U^{\text{Diff},m} \), versus problem size of discretization \( m \). Analysis performed to determine reference value for steady-state internal energy. 103

3.20 Convergence of the relative error in the energy norm. Convergence rates of \( \beta = 3.07 \) and \( \beta = 2.61 \) are obtained for \( hp \)-GFEM and \( GFEM^{gl} \) analyzes, respectively. 104

3.21 Convergence to reference internal energy versus time curve for \( hp \)-GFEM simulations. 105

3.22 Relative difference in the discrete \( L_2 ^2 \) norm of the internal energy, \( L_{\text{Diff},m}^2(U(t)) \), versus problem size of discretization \( m \). Solutions are computed using the \( hp \)-GFEM. 106

3.23 Local mesh used for \( GFEM^{gl} \) simulations, nine levels of local refinement are used. 107

3.24 Convergence of the internal energy versus time curves for \( GFEM^{gl} \) solutions using Formulation 1. 107

3.25 Illustrates how the \( L_2 ^{\text{error}}(U(t)) \) value is effected by \( \rho c \). 108

3.26 Through-the-thickness temperature distribution for the Al beam (\( \rho c = 18.3 \)). 110

3.27 Internal energy versus time curves for Al. Solutions computed with \( GFEM^{gl} \) on a fixed global mesh and \( h \)-extensions in the local domain. 111

3.28 Convergence in \( L_2 ^{\text{error}}(U(t)) \) for internal energy versus time curves generated with \( hp \)-GFEM and \( GFEM^{gl} \). 112

3.29 Convergence in energy norm for \( hp \)-GFEM and \( GFEM^{gl} \) solutions at \( t = 0.5 \text{ sec} \). 113

3.30 Convergence in energy norm for \( hp \)-GFEM and \( GFEM^{gl} \) solutions at \( t = 1.0 \text{ sec} \). 113

3.31 Illustrates the through-the-thickness variation of \( V_{\text{SiC}} \) for the values of \( q \) used in the simulations. 114

3.32 Convergence in \( L_2 ^{\text{error}}(U(t)) \) norm of the internal energy for the \( GFEM^{gl} \) as applied to Case 1 (Al) and Case 2 (SiC). 116

3.33 Internal energy versus time curves for Case 1 (Al) and Case 2 (SiC) computed with \( GFEM^{gl} \) and \( hp \)-GFEM. Reference curves generated using \( hp \)-GFEM are shown using solid lines while the \( GFEM^{gl} \) data are shown using glyphs. 116

3.34 Convergence in the \( L_2 \) norm of through-the-thickness temperature distributions for the \( GFEM^{gl} \) as applied to Case 1 (Al) and Case 2 (SiC). 117

3.35 Through-the-thickness temperature distributions at \( t_{\text{final}} \) for each of the 5 Cases. For each of the plots, solid lines indicate reference curves generated with the \( hp \)-GFEM, and glyphs indicate \( GFEM^{gl} \) data. 117

3.36 Plot of \( L_2 ^{\text{Diff}} \) for \( hp \)-GFEM and \( GFEM^{gl} \) simulations with different size \( \Delta t \). 119

3.37 Spatial and temporal variation of the Gaussian laser flux, (\( \gamma = 50 \)). 120

3.38 Internal energy vs time curves for solutions obtained with time-rise constant \( \gamma = 10 \) and \( \gamma = 50 \). Reference curves generated using \( hp \)-GFEM are shown using solid lines while the \( GFEM^{gl} \) data are shown using glyphs. 120
3.39 Convergence in the $L^2_{\text{error}}$ norm of the internal energy for $GFEM^{gl}$ as a function of element size in local problem. ......................................................... 121
3.40 Convergence in the $L^2_{\text{error}}$ norm of through-the-thickness temperature distribution at $t_{\text{final}}$ as a function of element size in local problem. ................................. 122
3.41 Evolution of the through-the-thickness temperature distribution for the beam with $\gamma = 10$. ............................................................ 122
3.42 Evolution of the through-the-thickness temperature distribution for the beam with $\gamma = 50$. ............................................................ 123
3.43 Moving laser flux applied to the beam model. ................................. 123
3.44 Internal energy vs time curves for solutions obtained with a moving source. . 125
3.45 Snap-shots in time for the transient $GFEM^{gl}$ simulation for the beam with a moving laser flux. ......................................................... 125
3.46 Snap-shots in time for the transient $hp$-GFEM simulation for the beam with a moving laser flux. ......................................................... 125

4.1 Mesh 0x, local solution applied at nodes denoted by red glyphs. ................ 130
4.2 Mesh 1x, local solution applied at nodes denoted by red glyphs. ................ 131
4.3 Mesh 2x, local solution applied at nodes denoted by red glyphs. ................ 131
4.4 Comparison of CPU times for various solution strategies for Mesh 0x. ........ 132
4.5 Comparison of CPU times for various solution strategies for Mesh 1x. ........ 133
4.6 Comparison of CPU times for various solution strategies for Mesh 2x. ........ 133
4.7 Model of panel used for analysis. The locations of the three load cases to be considered are also indicated. ......................................................... 134
4.8 Resulting temperature distribution for load Case 1. ............................... 135
4.9 Resulting temperature distribution for load Case 2. ............................... 136
4.10 Resulting temperature distribution for load Case 3. ............................... 136
4.11 CPU time requirements versus number of spike locations considered. .... 137
4.12 Point-wise error in the solution, $e(u)$, for $a = 0.025$. The flux function itself is also plotted to illustrate that the solution is affected even outside of the region subjected to the sharp loading. ..................................................... 139
4.13 Point-wise error in the solution, $e(u)$, for various values of $a$. As $a$ decreases, the flux function becomes sharper, and the error in the solution becomes significantly worse. ......................................................... 140
4.14 Global mesh used for the beam model. Flux boundary conditions are denoted by red arrows. ............................................................ 144
4.15 Convergence in the $L^2_{\text{error}}$ value for internal energy versus time curves generated with the transient $GFEM^{gl}$ as well as the transient $GFEM^{gl}_{ReAn}$. ................................. 145
4.16 Internal energy versus time curves generated with the transient $GFEM^{gl}$ as well as the transient $GFEM^{gl}_{ReAn}$. Data generated with $GFEM^{gl}$ is shown with using solid lines, while data generated with $GFEM^{gl}_{ReAn}$ is shown using glyphs. ......................... 145
4.17 CPU time requirements versus time for \( hp \)-GFEM as well as transient \( GFEM_{ReAn}^{gl} \) simulations. In (a) both the factorization and \((f/b_{sub})\), and assembly times are included; (b) only considers the factorization and \((f/b_{sub})\) time. .......................... 147

4.18 Internal energy versus time curves for \( hp \)-GFEM as well as transient \( GFEM_{ReAn}^{gl} \) simulations. ................................................................. 148

4.19 CPU time requirements versus time for \( hp \)-GFEM as well as transient \( GFEM_{ReAn}^{gl} \) simulations. In (a) both the factorization and \((f/b_{sub})\), and assembly times are included; (b) only considers the factorization and \((f/b_{sub})\) time. .......................... 148

4.20 Internal energy versus time curves for \( hp \)-GFEM as well as transient \( GFEM_{ReAn}^{gl} \) simulations. ................................................................. 149

4.21 CPU time requirements versus time for \( hp \)-GFEM as well as transient \( GFEM_{ReAn}^{gl} \) simulations. In (a) both the factorization and \((f/b_{sub})\), and assembly times are included; (b) only considers the factorization and \((f/b_{sub})\) time. .......................... 150

4.22 Internal energy versus time curves for \( hp \)-GFEM as well as transient \( GFEM_{ReAn}^{gl} \) simulations. ................................................................. 151

4.23 Internal energy versus time curve for transient \( GFEM_{ReAn}^{gl} \) simulation utilizing two more levels of local refinement than used for the curve shown in Figure 4.22. . . . 151

4.24 CPU time requirements versus time for \( hp \)-GFEM as well as transient \( GFEM_{ReAn}^{gl} \) simulations. In (a) both the factorization and \((f/b_{sub})\), and assembly times are included; (b) only considers the factorization and \((f/b_{sub})\) time. .......................... 153

5.1 Temperature field as described in (5.5). ................................. 158

5.2 Initial conditions as described in (5.8). ................................. 158

5.3 Internal energy as a function of time for linear elements. ................................................................. 163

5.4 Internal energy as a function of time for linear elements, using least squares to smooth out data. ................................................................. 163

5.5 Internal energy as a function of time for quadratic elements. ................................. 165

5.6 Internal energy as a function of time for quadratic elements, using least squares to smooth out data. ................................................................. 165

5.7 Internal energy as a function of time for quartic elements. ................................. 166

5.8 Internal energy as a function of time for quartic elements, using least squares to smooth out data. ................................................................. 166

5.9 Internal energy as a function of time for exponential elements. ................................. 168

5.10 Internal energy as a function of time for exponential elements. Zoomed in to see a difference in the curves. ................................................................. 168

5.11 Reference solution for simulations involving larger values of \( \rho c \), described by (5.19). 170

5.12 Simulation results generated using the enrichment in (5.21). ................................................................. 171

5.13 Simulation results generated using only quadratic, polynomial shape functions. ................................................................. 172

5.14 Comparison of element sizes for exponential and polynomial meshes. ................................................................. 173
6.1 Difference in CPU time requirements for the $GFEM_{ReAn}^{gl}$ when the assembly time is either considered or neglected. .............................................. 176

6.2 Master/Sub-Local problem approach for the parallelization of the $GFEM_{gl}$ algorithm. Master-local problem domains are decomposed into sub-local problems, which are then solved in parallel. .............................................. 178
Chapter 1

Introduction

1.1 Motivation and Applications

Many applications in engineering practice involve the analysis of structural behavior with multiple spatial scales of interest. One such case is seen in structures subjected to intense thermal loadings of a localized nature. The motivation for this particular investigation is the analysis of localized heat sources on the skin of hypersonic flight vehicles.

Vehicles in hypersonic flight are subjected to very severe thermo-mechanical loadings (only the thermal loadings are considered here). At very high speeds, there are significant changes in the properties of the compressed air, such as the density and temperature [95]. Changes in the pressure distributions on a hypersonic flight vehicle can cause the formation of shock waves in specific locations (Figure 1.1). Possible interactions of these shock waves can cause very intense thermal loadings, which are very localized and exhibit sharp gradients. A potentially representative loading was obtained through private communications, and is shown in Figure 1.2. For the loading shown in the figure, the maximum flux is $126 \frac{Btu}{s \cdot ft^2}$, and it is seen to drop off rapidly. In some instances it is estimated that the maximum flux may act over a distance on the order of as little as several microns. It should be noted that this distribution may be taken only as potentially representative, as the values are not considered typical. Rather, they correspond to one particular mission profile.

The characterization of the resulting thermal loadings and pressure distributions, as well as the effect of these loadings on the aeroelastic behavior of the vehicle itself has been the focus of many research investigations [21, 50, 51, 76, 99, 100, 104]. The most severe of these loadings is the so-called Edney Type IV shock wave interaction, which is a bow shock/cowl shock interaction which
Figure 1.1: Shock waves in the flow field of an X-15 at Mach 3.5. (Figure taken from NASA website.)

Figure 1.2: Potentially representative normal flux on the leading edge of a wing. Distance indicates the distance in inches from the point of maximum flux.
may occur on the leading edge of a wing.

Attempts have also been made to take into account the chemistry of the high-speed flow field itself [55] and to develop a fully analytical solution for the Type IV interaction [47]. A comprehensive summary of the research performed in the area of high-speed air-vehicles can be found in [95]. While the methodology proposed here is largely motivated by hypersonic flight vehicles, it may find application in any of a number of practical engineering situations involving intense, and highly localized heat sources.

One such application area is in Computational Weld Mechanics (CWM). A summary of the current state-of-the-art in CWM is presented in [70]. The field is concerned with the accurate thermo-mechanical analysis of the weld material subjected to thermal cycles. A common approach is to break down each time-step into a thermo-mechanical problem with only a one-way coupling. First, a thermal analysis is run, followed by a mechanical analysis based on the generated thermal field. For the thermal analysis it is recommended by the author of [70] to use the backward Euler method, or the $\alpha$-method with $\alpha = 1.0$ for the time-stepping algorithm, along with a lumped capacity matrix. With this setup one will have unconditional stability in the algorithm, as well as avoid negative temperatures which sometimes result in the regions of high gradients with the use of consistent capacity matrices.

Due to the transient, highly localized meshing requirements of CWM, computational efficiency is often of great concern. A commonly used strategy is to utilize dynamic, adaptive meshes. Typically the mesh contains regions of high refinement only locally near the moving heat source, and then the regions are unrefined subsequently. Another technique is substructuring, in which it is assumed that the region near the weld is the only important region, as the rest of the structure is assumed to have a smooth solution, and is thus condensed out of the system of equations. Again, this process is transient, so dynamic substructuring is required in which the local region is changing throughout the course of the simulation. It is required, however, to obtain a thermal loading applied by the condensed out global structure, so recovery steps are necessary which limit the efficiency. Attempts have also been made to combine the dynamic meshing and substructuring techniques,
where the global structure is modeled with shells, and the substructure is dynamically meshed with solid elements. Examples of the meshing strategy can be found in Figure 5 of [70]. This method requires a mapping of the local thermal field to the global model at each time-step, again limiting the computational efficiency and accuracy.

Another common method to reduce problem size involves dimensional reduction, in which 2-D elements, or shell elements are used, even though the assumptions for shell elements are not valid in the regions close to the weld. Yaghi et al. [106] use 2-D elements due to computational considerations, even though they note that welding is inherently a 3D process. A 2-D mesh with extremely high levels of localized refinement used for a welding simulation can be found in Figure 1 of [106].

The highly localized nature of the welding process, combined with high spatial gradients, the requirement of transient analyses as well as problems arising from the lack of efficient solution methodologies make CWM a good candidate application area for this research. Another aspect of CWM which can be addressed by methods similar to the one proposed in this work is the necessity to accurately represent phase transitions, and bi-material interfaces. While not addressed in this work, these types of problems have been studied in [15, 17, 75], and can be accurately accounted for through the use of the special enrichment functions which can represent well the moving interface between the material phases.

A similar application to welding involves laser line heating/laser forming of metal plates. Thermo-mechanical forming of doubly curved plates involves the induction of plastic deformation through the heating and subsequent cooling of the plate. A discussion of both the physical processes involved, as well as current computational methods are provided in [107, 108]. Line heating by oxyacetylene torch, or lasers are both commonly used practices, particularly in shipyards. In order to accurately predict the final shape, the thermal fields must be accurately resolved, before the accompanying mechanical analysis is performed. It is noted by the authors that the FEM has been used for simulations with some success, but the main drawback is the required computational time. Again, in this instance, high mesh density is only required in a small local region near the applied
heat source, and large elements suffice elsewhere. Typically, 3D solid elements are required for analysis because the through-the-thickness temperature gradient is the sought-after result, and shell elements are therefore, not appropriate. The standard technique involves using a single, fixed mesh which has a highly graded region, yielding high refinement along the line where the heat source is to be applied. An example of this type of mesh is can be found in Figure 2 of [108].

As mentioned previously, this leads to a mesh which is overly refined because the localized source will only be in a given location briefly. The authors describe their ‘rezoning’ technique, in which the mesh refinement is changed between subsequent time-steps to have high refinement in only a small region. The rezoning strategy is illustrated in Figure 1 of [108]. While this technique can show significant improvement in the CPU time taken to run the simulation, there is a projection of solutions and variables required between subsequent meshes. It is unclear how this projection impacts the quality of the simulation.

The highly localized nature of this application once again makes it a good application area for the proposed methodology. The goals of this research, to be further discussed subsequently, will be to address the issue with mesh refinement, and will not require any type of projection of solution vectors or state variables between subsequent time-steps. As such, if coupled (one-way) with a mechanical solver, the proposed methodology would make for a good candidate for the simulation of laser line heating of plates.

While the previously discussed application areas are of interest to the engineering community, they are by no means an exhaustive list. Several investigations have been performed into the numerical solutions of more generalized heat transfer problems with localized effects, similar to the type of problems of interest in this work. To briefly name a few: Tamma and Saw [98] offer a local, hierarchical $p$-enrichment strategy for thermal problems in which a-posteriori error estimates are used to drive the local $p$-enrichment in elements whose error level is deemed unacceptable. The effects of localized, intense laser irradiation on a functionally graded composite plate is investigated in [18], using a Meshless Local Petrov-Galerkin Method (MLPG). The effects of heat generated due to dynamic fracture in an elastic-plastic medium is investigated in [69]. The authors are able to obtain
good results with the use of the Streamline Upwind Petrov-Galerkin Method, which eliminates the spurious oscillations seen in results generated by traditional FEM and finite difference methods. In [101] the authors investigate crack initiation in the regions near localized heat sources, such as thermal shocks.

1.2 Related Methods

In this section, a review of existing methods related to the method proposed in this work is presented. The review is not intended to be an all-inclusive summary of the literature in the field. For further discussion, or more details for a specific method covered, the reader is referred to the specific reference of interest and the references therein.

It is also noted that some of the methods are applied to engineering problems which also pose possible applications for the method proposed in this work. While there are possible applications presented in this section, they are included here, and not in the previous section because the solution methods which they present are of primary interest, and due to their similarity with the method proposed in this work, there is some natural cross-over of potential target application areas.

1.2.1 Multi-Scale Techniques

The goal of this research is to numerically solve a partial differential equation (PDE) which exhibits both a smooth, coarse-scale solution component, as well as a highly localized, fine-scale component. This type of analysis technique is not unique, and there have been numerous previous investigations into similar techniques which are also geared towards solving PDEs with multiple scales of interest exhibited by the solution.

In the first such method discussed, Fish [43] proposes the $s$-version of the FEM. In this version the accuracy of the solution is increased by superimposing a mesh of high-order hierarchical $p$-elements in regions of unacceptable error. The solution is broken down into the sum of the base
mesh \((u^0)\) and the overlay mesh \((u^H)\) as \(u = u^0 + u^H\). Dirichlet boundary conditions are applied on the superimposed mesh so as to guarantee \(C^0\) continuity, as shown in (1.1). Homogeneous BCs are applied to portions of the superimposed mesh which do not intersect with the boundary of the global mesh (1.2).

\[
\begin{align*}
  u &= u^0 + u^H \quad \text{on } \Omega_H \\
  u^H &= 0 \quad \text{on } \Gamma_{OH} \\
  u^0 + u^H &= u^P \quad \text{on } \Gamma_u
\end{align*}
\]

Homogeneous BCs could also be applied to portions of the superimposed mesh which intersect the boundary of the domain, but the BCs in (1.3) are applied in these regions so as to increase the resolution of the solution on the displacement boundary of the global mesh. The method yields favorable results in terms of error obtained, but it does so at a significant increase in the number of degrees-of-freedom (dofs) added to the system of equations to be solved. In addition to the increase in computational requirements, special quadrature rules are also required to numerically integrate the products of shape functions representing the \(u^0\) and \(u^H\) portions of the solutions when superimposed elements intersect more than one original element.

It is worth noting that the s-FEM concept is extremely similar to the earlier methodology proposed in [8] by Belytschko and Fish, in which the overlay consists of a spectral approximation, using a Rayleigh-Ritz approximation. The spectral approximations are used only in regions of high gradients, where the additional degrees of freedom provided by the spectral overlay yield much better solution accuracy.

Fish extends the s-FEM to applications involving multilayer composite laminates [44], where high order polynomials are used for both the global approximation, as well as the superimposed overlay.
elements. In [41], Fish uses low order elements in the overlay mesh to provide a partition of unity through which the insertion of special enrichment functions is possible. In this work, eigenmodes are used as enrichment in the overlay approximation to simulate the propagation of discontinuities, in what is termed the reduced order $s$-FEM.

In a similar work to that of Fish, Rank [86] proposes an $h$-$p$ domain decomposition to analyze a reaction-diffusion problem involving an interface between the two species involved. For this application, the solution is smooth from a global perspective, so a $p$-version of the FEM captures the solution accurately. Locally, however; near the interface of the two species the solution exhibits sharp gradients, which require high levels of refinement. This requirement is satisfied by the addition of $h$- and $p$-version FEM overlay meshes, with Dirichlet BCs guaranteeing $C^0$ continuity, and the overall solution taken as the sum of the global and local solutions. In this work the author implements his methodology in an existing $p$-version FEM code. He first solves the problem with the $p$-code, and marks refinement zones, where the errors are unacceptably large. Daughter meshes are extracted from these regions, and used to generate the overlay meshes. In this way compatibility of the meshes is preserved, but the resulting mesh is non-conforming, requiring the Dirichlet BCs to guarantee $C^0$ continuity.

Krause and Rank [68] then extend the previous method for solid mechanics applications. They note that their method is very similar the $s$-FEM put forth by Fish. Schematically the two methods look very similar. The main difference is that the $s$-FEM [43] uses low-order global approximations, with high-order hierarchical superimposed meshes; while the so-called $hp$-$d$ method uses a $p$-mesh globally, and only a linear $h$-version mesh locally. The authors formulate a Gauss-Seidel iterative solution strategy which allows for an efficient solution for the system of equations. In their implementation they iteratively refine the overlay mesh, and generate solutions for the corresponding overlay mesh until acceptable error levels are achieved.

The method is then further extended by Duster, et al. [36] to incorporate the enhancement of dimensionally reduced models. They use the same basic methodology as the previous, and again rely on the BCs in order to guarantee $C^0$ continuity. The Gauss-Seidel solution strategy is utilized
as well, with an emphasis on the ability to separate out the coarse-scale and fine scale problems and solve them separately. In this way they do not need to compute the coupling matrices directly, and only exchange information in the form of pseudo-loads and pseudo-solutions between the different scales. The authors also note the possibility of using more than one overlay mesh, without the need to re-formulate the solution methodology. They demonstrate the methodology by coupling a 1D beam model with 2D elasticity in the regions of interest in the form of a beam with holes in it and show a good ability to capture the global and local responses with high accuracy. They also solve a 3D problem, which consists of a 3D plate with a cut-out in it.

A method similar in nature is proposed by Wyart, et al. [105] and applied to a stiffened panel with a crack; a problem considered to be of industrial level interest. In their methodology, the so-called substructured finite element/extended finite element method ($S-\text{FE}_{\text{shell}}/\text{XFE}_{3\text{D}}$ method) uses shell elements to represent the safe, uncracked portion of the panel, and 3D elasticity models in the local, cracked regions of interest. They then use this method to extract stress intensity factors (SIFs) in complex thin-walled structures, i.e. a stiffened panel with a crack in it. While the application is different, the methodology is similar to that of Duster, et. al in their use of dimensionally reduced models globally, and full-order models in the local regions of interest.

Chahine, et al [11] propose the so-called Spider XFEM. In this methodology the asymptotic displacement at the crack tip is approximated with the addition of an adaptive patch in the region. The authors note that only some of the information about the solution is required a-priori, as opposed to the standard XFEM when the asymptotic expansion is assumed to represent the displacement field at the crack tip. A Lagrangian triangulation defined in the region of the crack tip is geometrically transformed into polar coordinates, yielding a circular mesh, somewhat resembling a spider web. This overlay mesh is what leads the name Spider XFEM. An opening mode displacement field is prescribed as the Dirichlet BCs on the spider mesh. A cut-off function is then required to obtain $C^0$ continuity of the approximation field. The methodology is applied to a cracked, isotropic, homogeneous material, with marginal success. The authors stress that the requirement of only partial knowledge of the displacement fields a-priori (the dependence in $r$ of the asymptotic displacement)
is a benefit. The method performs marginally well because the convergence order is as good as that of the standard XFEM, but the actual error values are not. Hou and Wu [58], develop a method similar to the one proposed in this research for problems dealing with porous media. Their aim is also to capture fine-scale effects on the large scale model without the need for explicitly resolving the fine-scale features on the global mesh. The authors aim to do this through the construction of FE shape functions which are tailored to reflect the underlying physics of the fine-scale features. They do this through the solution of boundary value problems (BVP) defined on the elements from a coarse discretization. An elliptic problem is solved of the form

$$-\nabla a(x) \nabla u = f \quad \text{in} \quad \Omega$$  \hspace{1cm} (1.4)

where $a(x) = a_{ij}(x)$, which is assumed to be a non-smooth function, but yields both a symmetric and positive definite matrix. The special shape functions, $\phi^i$ are then defined through the solution of

$$-\nabla \cdot a(x) \nabla \phi^i = 0 \quad \text{in element} \quad \kappa$$  \hspace{1cm} (1.5)

The question remains as to what BCs should be applied. One possibility is the application of the trace of standard, bi-linear FE shape functions on $\partial \kappa$, i.e. $\phi^i(x) = N^i(x)$ for $x \in \partial \kappa$. With this BC type, the authors did not obtain satisfactory results. A second possibility is to solve a 1D problem along each element edge, defined as

$$\frac{\partial}{\partial \xi} a_\mu(x) \frac{\partial \mu_1(x)}{\partial \xi} = 0$$  \hspace{1cm} (1.6)

where the assumption has been made that $a_\mu(x)$ is separable and can be written as $a_\mu(x) = a_1(x)a_2(y)$. Better results are obtained with this type of BC. The authors also had to deal with the effect of resonance, which occurs when the element size scale, $h$, is similar in magnitude to
some characteristic length scale, $\epsilon$, of the problem to be solved. To overcome this issue the authors used oversampling, in which they sample from a domain larger than that of an element in the construction of the shape function, $\phi_i$. While this technique overcomes the problem of resonance, it leads to a non-conforming method.

The work of Hou is extended by Efendiev, et al in [35, 37, 38] in which boundary conditions used to generate the shape functions are provided in the form of coarse or fine-scale global solves at the initial time-step. The shape functions are then able to deliver favorable results throughout the simulation, provided that the global boundary conditions do not change during the simulation. If global boundary conditions are changed frequently during the simulation, the shape functions must be re-computed in order to maintain accuracy. As such, the method may not be appropriate for these situations, when the global boundary conditions do not remain fixed. This drawback has been addressed to an extent in [35] in which thresholding is used so as to only require small portions of the global information to be re-computed as global boundary conditions change in time.

In the work of Fish and Yau [45], a method is introduced which employs both mathematical homogenization theory with PoU concepts. For more background, a survey on homogenization techniques is presented in [109]. Fish and Yau are attempting to insert information pertaining to the fine-scale in terms of material composition and properties, and insert this information into the coarse-scale through the PoU concept. They successfully apply their technique to the problem of continuum level fracture mechanics.

**Stabilized Methods** Stabilized finite element methods are in general very similar to standard Galerkin methods, in which a mesh-dependent stabilization term is added to the weighted residual equation. These terms tend to be functions of the residuals of the strong form equations on an element-by-element basis. Their inception was spurred by a desire to control the spurious oscillations and numerical pollution effects exhibited by standard FE simulations of advection-diffusion and Navier-Stokes equations. Franca et al. [49] propose a stabilized method for the advection-diffusion equation (1.7) subjected to homogeneous boundary conditions.
\begin{equation}
\mathbf{a} \cdot \nabla \mathbf{u} - \kappa \Delta \mathbf{u} = f \quad \text{in} \quad \Omega
\end{equation}

where \(a(x)\) is the flow velocity defined such that \(\nabla \cdot a = 0\) in \(\Omega\), \(\kappa\) is the diffusivity, and \(f(x)\) is the prescribed source. The bi-linear and linear forms, reflecting the stabilization terms are

\begin{equation}
B(u,v) = (a \cdot \nabla u, v) + (\kappa \nabla u, \nabla v) + \sum_{K \in \mathcal{T}_h} (a \cdot \nabla u - \kappa \Delta u, \tau(a \cdot \nabla v + \kappa \Delta v))\kappa
\end{equation}

\begin{equation}
F(v) = (f,v) + \sum_{K \in \mathcal{T}_h} (f, \tau(a \cdot \nabla v + \kappa \Delta v))\kappa
\end{equation}

where the stabilization term is defined as

\begin{equation}
\tau(x, Pe_K(x)) = \frac{h_K}{2 |a(x)|_p} \xi(Pe_K(x))
\end{equation}

For details on the variable definitions, see [49]. The authors then compare their formulation to that of the Streamline Upwind/Petrov-Galerkin (SUPG) method as presented in [10], in which the linear and bi-linear forms are defined as

\begin{equation}
B_{SUPG}(u,v) = (a \cdot \nabla u, v) + (\kappa \nabla u, \nabla v) + \sum_{K \in \mathcal{T}_h} (a \cdot \nabla u - \kappa \Delta u, \tau a \cdot \nabla v)\kappa
\end{equation}

\begin{equation}
F_{SUPG}(v) = (f,v) + \sum_{K \in \mathcal{T}_h} (f, \tau a \cdot \nabla v)\kappa
\end{equation}

The difference in the two formulations being the presence/absence of the \(\kappa \Delta v\) in the stabilization term. It is noted that with the use of linear elements, \(\Delta v = 0\) and the two methods coincide. The authors prove stability of the method, and obtain satisfactory results with their stabilized weighted residual form.

Franca et al. [48], propose a very similar method for the linearized, as well as fully incompressible
Navier-Stokes equations, and once again compare their formulation with that obtained via the SUPG method. For the case of the Navier-Stokes equations, the stability parameters, $\tau$ are a function of the Reynolds number, $Re$. For more details on the formulations and stability parameters the reader is referred to [48].

Hughes [59, 62], extends the ideas presented previously to multiscale problems in which the solution is thought to be composed of a large scale and small scale component. Hughes notes that the length scale of the element mesh is significantly larger than the characteristic length scale for the small or subgrid scale, but much smaller than the characteristic scale of the large scales. The small and large scales are referred to as 'unresolvable scales' and 'resolvable scales', respectively. In this work Hughes adopts the technique of solving for the large scale using finite elements, and solving for the small scales using the stabilization terms. In this case, Green’s functions are used, and more precisely element Green’s functions, approximated by 'bubbles' are used for the subgrid scale. With the use of 'bubbles’ there is the built-in, restrictive assumption that the small scale has a non-local effect on the large scale within an element, but this effect vanishes on the element boundaries, as is the nature of the 'bubble' function.

In the formulation of the so-called 'subgrid models' Hughes assumes that the solution can be written as a sum of the small scale and large scale components as $u = \bar{u} + u'$ where $\bar{u}$ represents the large scale and $u'$ represents the unresolvable, subgrid scale. The subgrid scales are solved for using Green’s functions

$$\begin{align*}
\int_{\Omega} g(x,y) (\mathcal{L}\bar{u} - f)(x) \, d\Omega_x &= M(\mathcal{L}\bar{u} - f) \\
\end{align*}$$

(1.13)

Further details of the derivation of the equations presented here can be found in [59]. The above can be substituted back in to the weak form, yielding the following equation

$$\begin{align*}
\mathbf{a}(\bar{\mathbf{w}},\bar{u}) + (\mathcal{L}^{\ast} \bar{\mathbf{w}}, M(\mathcal{L}\bar{u} - f)) = (\bar{\mathbf{w}}, f)
\end{align*}$$

(1.14)
The essential idea is that $u'$ can be solved for analytically on each element with Green’s functions, then (1.14) can be solved for numerically to yield $\bar{u}$. The author gives a brief discussion on the use of 'bubble’ functions to approximate the Green’s function on each individual element. More in-depth discussion on the use of 'bubble’ functions for stabilization terms can be found in [91].

After the presentation of subgrid scale models, Hughes includes a discussion of stabilized methods, and their relation to the previously described subgrid scale models. The general form for a stabilized FE method is given as

$$a(\bar{w}^h, \bar{u}^h) + \left( L\bar{w}^h, \tau \left( \mathcal{L} \bar{u}^h - f \right) \right)_\Omega = (\bar{w}^h, f)$$

(1.15)

where $L$ is some differential operator, and the weighted residual equation is modified from the standard FE form to include a stabilization term, $\tau$, which is often times an algebraic operator. For certain choices of the operator, i.e. the integral operator, $M$, in (1.13) the two methods coincide. The choice of $\tau$ given by Hughes defines $\tau$ as a function of the Green’s function associated with the problem as

$$\tau = \frac{1}{\text{meas}(\Omega^e)} \int_{\Omega^e} \int_{\Omega^e} g(x,y) d\Omega_x d\Omega_y$$

(1.16)

With the proposed stabilized method the essential idea remains the same as that proposed for the subgrid scale model: first solve for $u'$ analytically with a Green’s function, and then substitute this results in to the weighted residual equation and solve for $\bar{u}$ numerically. It is further noted [62] that stabilized methods are residual-driven, as the residual of the Euler-Langrange equations appears in the stabilization term, and as such the small scale solution is equivalent to the error in the coarse-scale solution, i.e. $u' = u - \bar{u}$.

A potential drawback to low-order stabilized methods is presented by Jansen et al. [65]. The authors refer to low-order methods as finite element methods which use shape functions that are of a lower order than the order of the highest derivative present in the governing equations. In
these situations, it is unclear as to whether the formulations remain consistent. The authors note
the method is in general inconsistent, except when the mesh-dependent stabilization parameter,
\( \tau \) tends to zero (at this point the standard FE formulation is recovered). That is to say that as
the element size tends to zero, so too does \( \tau \), and the method recovers consistency. This is not a
desirable property since the aim of the methodology is to resolve the small scale features without
the use of high levels of mesh refinement. As such, care must be taken when low-order finite
elements are used for stabilized methods so as to gain the increase in efficiency and resolution of
the small scale features without the loss of consistency of the method.

1.2.2 Transient Analysis and Time-Dependent Shape Functions

The proposed method in this research is not unique in the use of time-dependent shape functions, as
several other methods also allow for the evolution of shape functions in time. Fries and Zillian [52]
offer a mathematically rigorous investigation into time integration within the XFEM framework.
In the XFEM the solution is defined as

\[
\mathbf{u}(x,t) = \sum_{i \in \mathcal{I}} \mathbf{N}_i(x,t) \mathbf{u}_i + \sum_{i \in \mathcal{I}^*} \mathbf{M}_i(x,t) \mathbf{a}_i
\]

where the first sum is the standard FE approximation and the second sum represents the extended
doFs. \( \mathcal{I}^* \) is the set of nodes with extended enrichments.

Fries and Zillian analyze the diffusion, advection-diffusion, and quasi-linear Burgers equations.
In their discussion the authors note that it is important to discretize in time before discretizing in
space when time-dependent shape functions are to be used. In their investigation, they suggest the
use of the Crank-Nicolson, or generalized trapezoidal rule (\( \alpha = 0.5 \)) for time-stepping purposes.
The generalized mid-point rule is not recommended in general. It is noted that both methods are
equivalent for the linear diffusion and advection-diffusion equations.

The authors investigate both the Discontinuous Galerkin (space-time) FEM, as well as finite differ-
encing in time. They generate reference solutions utilizing space-time Lagrangian finite elements
with a $p$-order of 7 or 8, with elements matching the moving discontinuity. With the DG method, 3rd order accuracy in time is achieved. While this is very good convergence, the method does not extend to 3D in a straightforward manner, as a 4th axis would be required for the time variable.

For the spatial approximation, the $\text{abs}$ (1.18), (1.19) enrichment is utilized to capture the moving discontinuity, and the mesh no longer needs to align with the discontinuity. In (1.19), $\phi(x,t)$ is a level set function.

$$
M_i(x,t) = N_i(x,t) \cdot [\psi(x,t) - \psi(x_i,t_i)] \quad \forall i \in I^* 
$$ (1.18)

$$
\psi(x,t) = \text{abs}(\phi(x,t))
$$ (1.19)

With the use of finite differencing for time-integration, the generalized trapezoidal rule is recommended in order to obtain second order accuracy from a temporal standpoint. To this end, it is important to have sufficient spatial accuracy, so the error is controlled by the time-step size and not the spatial resolution. The formulation for the finite differencing in time accounts for the unsymmetrical capacity and stiffness matrices which are encountered during the discretization process (discussed further in subsequent paragraph). Special consideration is given to the integration of these matrices since the integration mesh changes from one time-step to the next due to the moving discontinuity. The authors show that the integration scheme used can impact the convergence of the solution, so care must be taken when defining the integration scheme to be used.

Van der Meer, et al. [102] use time-dependent shape functions in order to model transient geothermal systems. Their application area deals with high-gradient, conductive heat transfer associated with geothermal heating systems. The aspect ratio of the analysis domain as well as the slenderness of the pipe leads to very computationally expensive analyses. They propose a formulation for the generalized trapezoidal rule in time, and select ($\theta = 1.0$) to yield the unconditionally stable, implicit backward Euler method (1.20). The formulation also accounts for time-dependencies in the shape functions in its incorporation of unsymmetrical capacity and stiffness matrices in the
evaluation of the inertial contributions to the right hand side (rhs) vector (1.21).

\[ [C + \Delta tK]T_{j+1} = [C_b]T_j + F_{j+1}^{\text{ext}} \] (1.20)

\[ C_b = \int_{\Omega} \rho c_p N_{j+1}^T N_j \, d\Omega \] (1.21)

For the spatial approximation they use specially designed enrichment functions in order to permit the use of coarse meshes, and somewhat alleviate the computational demand for a meaningful analysis. The first special enrichment function, \( f(\zeta, a) \) is based on a-priori knowledge of the shape of the solution

\[ f(\zeta, a) = \exp \left( \frac{-(1 + \zeta)}{a} \right) \] (1.22)

where \( \zeta \) is the generalized coordinate and the parameter \( a \) is optimized iteratively. The authors also provide a second option in which the enrichment function is analytically derived to represent well a simplified, but similar physical process (1.23). A 2D formulation is also offered for the second enrichment function, while the first strategy is only 1D.

\[ f(\zeta) = \sum_n c_n \sin \left( \frac{n\pi(1 + \zeta)}{2} \right) \] (1.23)

Merle and Dolbow [75], apply the XFEM to problems of thermal phase change and moving heat spikes. The proposed methodology utilizes the X-FEM for the spatial discretization, using specially tailored enrichment functions to drastically improve efficiency. The time-stepping algorithm uses a standard trapezoidal method for most of the nodes, and a special \( L^2 \) projection to estimate the time derivative at nodes with special enrichment functions. The shape functions used to solve the problem involving a moving heat spike evolve in time to follow the thermal spike as it propagates throughout the domain (1.24).
\[ g(x, t) = e^{-[x - x_{\text{front}}(t)]^2} \]  

(1.24)

The second application is the problem with a moving, solidification interface between two materials. In this instance, the Heaviside enrichment (1.25) is used in order to generate a function with a discontinuous derivative in order to model the behavior near the interface.

\[ g(x) = H(x - x_f) \]  

(1.25)

A local, iterative procedure is required to ensure the continuity of the temperature field locally, as well as to determine the jump in the interfacial flux. The authors also note the importance of selecting a proper enrichment function, as they solve the same problem using a ‘tent’ function \( g(x) = 1 - \| x - x_f \| \), and show that the results are significantly worse than those obtained with the Heaviside enrichment.

Chessa and Belytschko [15] analyze the problem of modeling axisymmetric, two-phase flows. In their case, they are also dealing with a moving discontinuity, in the form of the changing interface between the two fluids. They are using time-dependent shape functions, where the time parameter shows up in the level set functions, \( \phi(x, t) \) used to implicitly define the interface. Their formulation is for the Navier Stokes equations, which is not so relevant to this work. Their formulation does also account for the unsymmetrical capacity and stiffness matrices (1.21) that contribute to the inertial terms in the right-hand-side vector. Time-dependent level-sets are again used by Chessa and Belytschko [16] within a space-time finite element context in order to solve the linear wave equation as well as Burger’s equation.

Nakonieczny and Sadowski [77] propose a meshless method in order to analyze the effect of thermal shocks on cylindrical functionally graded (FGM) plates. They use a semi-discrete partition of unity (PoU) finite element method in space in which they approximate the temperature field as
\[ T(x,t) = \sum_j \psi_j(x) \sum_k p_k(x) \tilde{T}_k(t) \]  

(1.26)

In the previous equation, \( \tilde{T}_k \) is the nodal temperature value, and the PoU functions, \( \psi_j \), are defined as:

\[ \psi_j(x) = \frac{w_j(x)}{\sum_k w_k(x)} \]  

(1.27)

The authors use splines (1.28) and Schwartz functions (1.29) to generate the partition of unity. Standard finite element (FE) shape functions are used for the basis functions, \( p_k(x) \). The authors acknowledge that non-standard, application specific shape functions are permissible, but they cite the geometrical simplicity of their domain as the reason for selecting standard FE shape functions.

\[ w_j = 1 - 6 \left( \frac{d_j}{r_j} \right)^2 + 8 \left( \frac{d_j}{r_j} \right)^3 - 3 \left( \frac{d_j}{r_j} \right)^4 \]  

(1.28)

\[ w_j = \exp \left[ \frac{d_j^2}{(d_j^2 - 1)} \right] \]  

(1.29)

The proposed method accounts for nonlinearity in terms of temperature-dependent material properties. The properties are determined from statistical data, also obtained from experiments. The time-marching scheme used amounts to the explicit, forward Euler finite difference method. When comparing their results to a reference obtained from an analytical solution, they achieve high levels of accuracy. When compared to experimental data obtained for a cylindrical, FGM plate subjected to a 'thermal shock', their accuracy is not as high.

Ching and Chen [18], apply the MLPG method to thermo-mechanical simulations of an FGM plate subjected to laser heating. In this method the authors use moving least squares to form their basis functions. The authors use the following weight functions.
where $d_i = |x - x_i|$, $c_i$ is the distance between node $i$ and its third nearest neighboring node, and $r_i$ is the size of the radial support of the weight function, outside of which $W = 0$. For the transient thermal analysis, the generalized trapezoidal rule is used with $\beta = \frac{1}{2}$ yielding the unconditionally stable, second order accurate Crank-Nicolson method. For the FGM modeling, effective moduli are estimated as follows: the heat capacity $\rho c$ is determined using the rule of mixtures; the bulk modulus, $K$, shear modulus, $\mu$, thermal conductivity, $\kappa$ and thermal expansion coefficient $\alpha$ are approximated using the Mori-Tonaka method as well as the self-consistent field method. The authors perform a parametric analysis varying the volume fractions of the different phases, and are able to obtain good results spatially through the use of the Gaussian function which they are using in their moving least squares approximation.

Waisman and Belytschko [103] use an iterative technique similar to the one used in [102] to develop special shape functions which represent well the underlying physics of the problems which they are attempting to solve. They apply the special shape functions (1.31) and (1.32) to capture boundary layers, such as those that show up in the solution of the reaction-diffusion boundary layer problem. Local residuals are derived and the parameter, $\beta$, is iterated upon until the local residual is sufficiently small.

$$\Phi(x, \beta) = \exp[-\beta x]$$  \hspace{1cm} (1.31)

$$\Phi(x, \beta) = \frac{1}{(x+1)^\beta}$$  \hspace{1cm} (1.32)

The authors also solve a fracture mechanics problem using enrichment functions of the form $\Phi_i(\beta) = r^\beta f_n(\theta)$. In this instance they once again iteratively solve for $\beta$ and are in effect solv-
Rethore, et al. [88] propose an FEM for dynamic fracture utilizing remeshing. The authors note the inherent complications with crack propagation, in terms of remeshing requirements after each propagation step. As such, the authors note that one needs to rewrite Newmarks update formulae to reflect the use of two different discretizations. The result is a system of five equations in eight unknowns. The change in the discretization between two subsequent steps requires a projection of the state vectors (displacement, velocity, and acceleration) from time-step $t^n$ to the discretization at $t^{n+1}$, in order to reduce the system down to 5 equations in 5 unknowns. For the projection operation, a simple linear interpolation is used between time-steps, since remeshing algorithms and projection techniques are not their focus, rather the energy balance itself is under investigation.

The main contribution of the work is what the authors have termed the balance recovery method. The first step is equilibrating the state vector from $t^n$ on the discretization at $t^{n+1}$, by recognizing that this is only possible if crack closing forces are applied along the new crack surface, ($\Delta a$). This crack closing force is taken as the energy required to generate new crack surface, and as a result the methodology is able to minimize the energy which is artificially introduced by the change of discretization. This particular property is not possible using only remeshing and projection of state vectors between subsequent discretizations.

Figure 8 of [88] illustrates the benefit of the balance recovery method over a more standard (according to the authors most FEM software does not use the recovery method) FEM using only remeshing and projection techniques. The end result is that the cumulative energy out of balance is significantly reduced with the use of the balance recovery method.

Rethore, et al. [89] then expand the previous methodology for use within the XFEM framework. The same basic procedure is described, but the XFEM offers significant simplification due to the lack of remeshing requirements associated with the XFEM and similar methodologies. With the advancement of the crack front, the authors are continuously adding enrichments (in the form of
discontinuous Heaviside and branch function enrichments), so the size of the state vector is always increasing. The newly added dofs are initialized to zero, so as to avoid the artificial addition of energy. Due to that lack of remeshing, the authors note that the state vector at \( t^n \) satisfies equilibrium on the discretization at \( t^{n+1} \), which further simplifies the balance recovery method applied to the XFEM. The only requirement, is that enrichments are always added, and never removed, that is to say that the solution space at \( t^{n-1} \) is a subspace of the solution space at \( t^n \).

The topic of energy conservation is again revisited by Combescure, et al [19] in the context of dynamic crack propagation within the XFEM framework. The authors present a comprehensive discussion of various methods which they do not consider robust, such as remeshing, cohesive zone models and element deletion, along with supporting evidence for their claims. Proof of energy conservation within the XFEM is presented, along with strong language about the inability to ensure energy conservation with remeshing techniques. The authors once again provide results comparing the energy unbalance for standard FEM with that obtained by FEM with balance recovery, and XFEM, where in this instance cohesive zone models are used with the XFEM simulation.

There is a growing body of literature which deals with explicit time-integration for dynamic fracture, once again within the XFEM framework. Menouillard, et al [73] propose a mass lumping technique (1.33) which enhances the allowable time-step size in dynamic XFEM simulations. The authors note the problem with the allowable time-step size tending to zero as the crack approaches a support boundary when a consistent mass matrix is used. They pose the following lumping strategy (1.33) which allows for feasible allowable time-steps. The allowable step size does not tend to zero as the crack approaches a support boundary, and remains on the order of magnitude of allowable time-steps using standard FEM, noted by the authors to be related by \( \frac{1}{\sqrt{2}} \).

\[
m_{diag} = \frac{m}{n_{nodes}} \frac{1}{mes(\Omega_{el})} \int_{\Omega_{el}} \phi^2 s \Omega
\]

(1.33)

Menouillard, et al. [74] further extend the idea of mass lumping for the XFEM, by first investigating lumping techniques posed by Rozycki et al [90] which involves first a simple row-sum
technique. This leads to the time-step tending to zero as the element size tends to zero, as noted in
the previous paper. Rozycki et al then pose the idea of distributing the mass evenly to each node
and thus eliminating the problem of the time-step tending toward zero.

Menouillard et al start with the first proposed mass-lumping method, along with the resulting shape
functions for an element enriched with the Heaviside function. They also develop the formulation
for the Hansbo and Hansbo basis functions (shadow node version) and note that the two sets of
basis functions are related by a simple rotation. They use the simple row-sum technique for the
Hansbo and Hansbo element, and note that this lumped mass matrix tends to zero as the crack
crosses over a node. They then use the simple rotation matrix that they define relating the Hansbo
and Hansbo basis with the Heaviside shape functions. The result is a block-diagonal mass matrix
which no longer yields allowable time-step sizes which tend toward zero as the crack approaches
a node.

Elguejdj, et al. [40] extend the mass-lumping work of Menouillard, et al. [74] to be used with
arbitrary enrichments, and not solely for enrichments used for fracture mechanics applications.
They adopt the same basic lumping strategy

$$m_L = \frac{1}{\sum_{i=1}^{\text{nodes}} \psi^2(x_i)} \int_{\Omega_e} \rho \psi^2 s \Omega_e$$

(1.34)

Even though they propose the technique for arbitrary enrichments, they only investigate its use
for the Heaviside enrichment. With the proposed lumping strategy, they are able to obtain critical
time-steps that are once again on the order of those obtained with standard FEM, depending on
where the discontinuity is located relative to support boundaries. They note that the most stringent
that the allowable time-step can be is one half of that obtained for standard FEM and as such put
forth this rule of thumb to be used for dynamic XFEM simulations utilizing arbitrary enrichments.
In part 2 [56] of the previous paper the authors then propose a stable-explicit/explicit dynamic
scheme, based on a stable-explicit scheme proposed by Chang [12–14]. In this method the system
of equations is as follows
\[ M\ddot{U}_{n+1} + K\dot{U}_{n+1} = F_{n+1} \] (1.35)

\[ U_{n+1} = U_n + \beta_1^{-1}\Delta t\dot{U}_n + \beta_2^{-1}\Delta t^2\ddot{U}_n \] (1.36)

\[ \dot{U}_{n+1} = \dot{U}_n + \frac{1}{2}\Delta t (\ddot{U}_n + \ddot{U}_{n+1}) \] (1.37)

where \( \beta_1 = (I + \frac{1}{4}\Delta t^2M^{-1}K) \) and \( \beta_2 = \frac{1}{2}\beta_1^{-1} \). The method as posed by Chang is unconditionally stable, and second order accurate. While the method is explicit, it does require a linear system solver due to the use of the operators \( \beta_1 \) and \( \beta_2 \).

The method then requires the use of a more sophisticated analysis on an element-by-element basis which produces the end result of allowable time-steps which are identical to those obtained for standard FEM, regardless of the crack front location relative to a nodal support boundary. To this end, the authors tie the scheme proposed by Chang with the standard central difference method by incorporating techniques used in [6, 60, 61], in which the element-by-element time-integration strategies are utilized. The system of equations is modified to the form

\[ M\ddot{U}_{n+1} + K\dot{U}_{n+1} = F_{n+1} \] (1.38)

\[ U^E_{n+1} = U^E_n + \Delta t\dot{U}^E_n + \Delta t^2\ddot{U}^E_n \] (1.39)

\[ U^{SE}_{n+1} = U^{SE}_n + \beta_1^{-1}\Delta t\dot{U}^{SE}_n + \beta_2^{-1}\Delta t^2\ddot{U}^{SE}_n \] (1.40)

\[ \dot{U}_{n+1} = \dot{U}_n + \frac{1}{2}\Delta t (\ddot{U}_n + \ddot{U}_{n+1}) \] (1.41)
where \( M = M^{SE} + M^{E}, \ K = K^{SE} + K^{E} \) and \( F = F^{SE} + F^{E} \), in which the superscript ‘SE’ refers to the stable-explicit elements, and ‘E’ refers to standard explicit element. Due to the requirement for a linear solver on the SE terms, it is desireable to keep the SE components as small as possible. It is noted that \( M^{SE} \) is a lumped matrix using (1.34) and \( M^{E} \) is a standard lumped mass matrix. With the previously described modifications to the system of equations, the authors are able to obtain lumped mass matrices for arbitrary enrichments which yield allowable time-step sizes which are identical to those obtained using standard finite elements.

As noted, the use of partitioning the mesh into subgroups and performing the time-integration using different integration schemes in different partitions is also investigated in [6]. The motivation for this work was based on the observation that for applications involving structure-media (fluid, soil, etc) interaction it would be possible to integrate the stiff, high frequency structural domain implicitly and the softer, low frequency media domain explicitly. The authors provide the update formulae for the two portions of the displacement vector as

\[
d_{n+1}^E = 2d_n - d_{n-1} - \Delta t^2 (M^E)^{-1} (P_n^E - K^{EE} d_n^E - K^{EI} d_n^I)
\]

\[
d_{n+1}^I = \left( K^I + \frac{4}{\Delta t^2} M^I \right)^{-1} \left\{ P_{n+1}^I - K^{IE} d_{n+1}^E - M^I \left( d_{n+1}^I - \frac{4}{\Delta t} \gamma_n^I - \frac{4}{\Delta t^2} d_n^I \right) \right\}
\]

where ‘E’ denotes explicit and ‘I’ denotes implicit integration. For the explicit integration scheme the central difference method is used and for the implicit integration the unconditionally stable trapezoidal rule is used. It is noted from the update equations (1.42), (1.43) that the explicit integration must be done first, followed by the implicit, due to the implicit updates dependency upon \( d_{n+1}^I \). The method is then extended in [7] to support the use of explicit-explicit, implicit-implicit, implicit-explicit and explicit-explicit\(^m\) integration techniques. The fourth method utilizes explicit integration in each partition but uses different time-steps in each partition. The time-step sizes are related through a scalar, \( m \). In [78] a subcycling method is presented in which it is no longer necessary to have the time-step sizes as integer-multiples of each other, further extending the flexibility.
of the mixed time integration method.

A similar methodology, involving the partitioning of the mesh into subdomains, and using different
time-step sizes in each subdomain is presented by Gates et al [53, 54]. The authors present an
asynchronous, multi-domain variational time integrator appropriate for both linear and nonlinear
mechanics problems. In this work, the major difference is that the time-integration scheme is
derived from discretizing directly the Langrangian, and using variational calculus. The authors
note that within this framework they obtain symplectic integrators which are capable of preserving
both momentum as well as energy.

The authors define the Langrangian of a mechanical system to be equal to the kinetic minus poten-
tial energy

\[ L(q, \dot{q}) = \frac{1}{2} \dot{q}^T M \dot{q} - V(q) \]  

(1.44)

where \( M \) is a positive definite mass matrix, and \( \dot{q} \) is a velocity. The authors then define the action
as

\[ A = \int_{t_0}^{t_f} L(q, \dot{q}) dt \]  

(1.45)

and invoke Hamilton’s principle stating that the motion of the systems make the action stationary
with respect to \( q(t) \) in order to derive the Euler-Lagrange equations for the differential equation.
The following systems of equations are derived for the discrete Euler-Lagrange equations within
the Langrangian framework as

\[ \frac{1}{2} \Delta t \left( -\alpha \nabla V (q_{n-1} + \alpha) - (1 - \alpha) \nabla V (q_{n-1}) - (1 - \alpha) \nabla V (q_{n+\alpha}) - \alpha \nabla V (q_{n+1 - \alpha}) \right) = 
\]

\[ M \left( \frac{q_{n+1} - 2q_n + q_{n-1}}{\Delta t} \right) \]  

(1.46)
and within the Hamiltonian framework as

\[ p_n = \frac{1}{2} \Delta t (1 - \alpha) \nabla V (q_{n+\alpha}) + \frac{1}{2} \Delta t \alpha \nabla V (q_{n+1-\alpha}) + M \left( \frac{q_{n+1} - q_n}{\Delta t} \right) \]  

(1.47)

\[ p_{n+1} = -\frac{1}{2} \Delta t \alpha \nabla V (q_{n+\alpha}) - \frac{1}{2} \Delta t (1 - \alpha) \nabla V (q_{n+1-\alpha}) + M \left( \frac{q_{n+1} - q_n}{\Delta t} \right) \]  

(1.48)

in which \( q_{n+\alpha} = (1 - \alpha)q_n + \alpha q_{n+1} \). The authors present a domain decomposition which allows the mesh to be broken down into subdomains, each using its own time-step size, with no requirement of particular ratios between any of the subdomain time-steps. Continuity is enforced with the use of an interface, \( \psi \), to which adjacent subdomains are constrained. The constraint involves the use of local Lagrange multipliers, \( \lambda_n \). The resulting methodology yields a method which conserves energy and can be used to efficiently solve large scale problems due to the efficiency with which the integration strategy can be parallelized. While the use of variational time integrators is worthy of brief discussion, it is not of direct interest to the method proposed in this work. As such, for further details the reader is referred to [54] and the references therein. A thorough survey of variational integrators as applied to time-dependent discrete mechanics is presented by Marsden and West in [71].

The majority of the previous references have dealt with problems in which the multi-scale nature arises from the presence of more than one spatial scale of interest, whether it comes from a crack, or localized heating. Solin et. al [94], on the contrary, propose a method utilizing hp-FEM along with dynamic meshes for problems which are multiscale from a temporal standpoint. The authors apply their method to heat and moisture transfer, and note that heat often propagates much faster than moisture. As such they would like to use not only adaptive meshes with refinement only where needed, but also an adaptive time step size. The authors are interested in simulating the life of a nuclear reactor vessel, with a 30 year time history. Due to the required refinement levels, along with requirements of very small time steps, and considering that the material models and coupling yield unsymmetric matrices, the required problem size becomes very computationally
The proposed method employs automated creation of small, low-order elements in the neighborhood of the local, moving fronts, and large high-order elements in regions where the solution is sufficiently smooth.

The solution strategy used is Rothe’s Method, which can be seen as a reverse version of the method of lines. Rothe’s Method requires the system to be discretized in time, but continuous with respect to spatial variables. The authors note that this provides a better setting for spatially adaptive algorithms. The solution at a given time-step starts with a mesh, $\tau_0$, assumed to be a uniform, coarse mesh discretizing the domain, $\Omega$. The solution, $u^{n+1}$ is then computed adaptively in $k_{n+1}$ steps starting with $\tau_0$ and then computing intermediate solutions $u^{n+1,1}, u^{n+1,2}, \ldots, u^{n+1,k_{n+1}}$ on meshes $\tau_{n+1,1}, \tau_{n+1,2}, \ldots, \tau_{n+1,k_{n+1}}$ where the intermediate meshes are obtained through a series of mutually independent refinement steps. The authors note that their refinement scheme allows for hanging nodes, and is therefore more computationally efficient. The time-step itself is adapted using an estimate of the local truncation error

$$
\tau_{k+1} = \left( \frac{e_{k-1}}{e_k} \right)^{k_p} \left( \frac{TOL_t}{e_k} \right)^{k_t} \left( \frac{e_{k-1}^2}{e_k e_{k_2}} \right)^{k_D} \tau_k
$$

(1.49)

The authors then compare three variations of their proposed methodology. In the first case they use $h$-adaptivity with quadratic elements, the second case uses an adaptive $hp$-FEM in which both fields are approximated on the same mesh, and the third case uses an adaptive $hp$-FEM where the temperature and humidity are approximated on separate meshes, referred to as the adaptive multi-mesh $hp$-FEM. The third variation is allowed in this case because the fields behave significantly different enough that they may be approximated independently. The results obtained by the study indicate that the two, $hp$-FEM versions utilized are more efficient in terms of dofs, CPU time, and adaptive time-step size.
1.3 Research Goals

The main focus of this research is based on the fact that the accurate analysis of sharp, localized gradients can require very high levels of local refinement when standard finite elements are used. In the 3D case, this translates into an excessively large number of degrees-of-freedom [84], making efficient analyses very difficult. These difficulties are amplified when transient simulations involving many time-steps are required. The high refinement levels required by the FEM to achieve suitable accuracy often leads to excessively small time step requirements. Adaptive meshing in the transient setting can also prove to be computationally expensive, and the mapping of time-dependent solutions between successive meshes is a non-trivial process. Even in a parallel computing environment, effective dynamic load balancing, and thus good parallel efficiency is also non-trivial to achieve.

In this work we propose to use the Generalized Finite Element Method with global-local enrichment functions ($GFEM^{gl}$) to circumvent the need for highly refined meshes. The methodology must first be applied to steady-state heat transfer problems in order to assess its efficacy in terms of resolving the localized spikes with a coarse mesh. The coarse mesh to be used will lead to a smaller system of equations to be solved for, which pays the largest dividend in the transient case. When transient analyses are required, the savings obtained by using a coarse mesh will be amplified as the number of time-steps is increased. With the proposed methodology we will be able to alleviate the high mesh density required by traditional FEM and resolve the sharp localized gradients on fixed, coarse meshes, thus making the 3D transient analysis of such scenarios more computationally feasible. With the proposed method, the mesh will remain fixed, and only the enrichments will change, allowing for the potential to re-use the large, global matrices throughout a simulation. The required modification is the hierarchical addition of the global-local enrichment functions, which will be updated at each time-step, and as such, they are time-dependent.

In summary, this work aims to accomplish the following goals:

- Develop a methodology utilizing numerically generated enrichments, allowing for the accu-
rate and efficient resolution of fine-scale features on a coarse mesh.

- Perform rigorous verification of the proposed method.

- Avoid the need for high levels of refinement which lead to prohibitively large problem sizes.

- Apply proposed methodology to transient simulations and study its convergence and robustness.

- Avoid the necessity to perform mesh refinement/unrefinement cycles, mapping of solutions and state-dependent variables between meshes.

- Re-use large, global matrices so as to maximize efficiency; and perform a detailed computational cost study.

- Alleviate the prohibitively small time-step sizes required for accurate/stable simulations due to high mesh density in explicit FEM simulations.
Chapter 2

Steady-State Heat Conduction

2.1 Problem Formulation

This chapter investigates steady-state heat transfer problems with solutions exhibiting highly localized sharp thermal gradients. Consider a domain $\Omega \subset \mathbb{R}^3$ with boundary $\partial \Omega$ decomposed as $\partial \Omega = \Gamma_u \cup \Gamma_f$ with $\Gamma_u \cap \Gamma_f = \emptyset$. The strong form of the governing equation is given by Poisson’s equation

$$-\nabla (\kappa \nabla u) = Q(x) \text{ in } \Omega \quad (2.1)$$

where $u(x)$ is the temperature field, $\kappa$ is the thermal conductivity tensor and $Q(x)$ is the internal heat source.

The following boundary conditions are prescribed on $\partial \Omega$

$$u = \bar{u} \text{ on } \Gamma_u \quad (2.2)$$

$$-\kappa \nabla u \cdot n = \bar{f} \text{ on } \Gamma_f \quad (2.3)$$

where $n$ is the outward unit normal vector to $\Gamma_f$ and $\bar{f}$ and $\bar{u}$ are prescribed normal heat flux and temperature, respectively.

In order for (2.1) through (2.3) to be used in a finite element implementation, the strong form of the governing equations must be cast in the weak, or variational form. The weak form of the steady-state heat equation is easily derived as found in [63, 87], and shown below
\[
\int_{\Omega} (\nabla u)^T \kappa \nabla w d\Omega = \int_{\Omega} Q w d\Omega + \int_{\Gamma_f} \tilde{f} w d\Gamma_f \tag{2.4}
\]

The Galerkin form of the previous problem is stated as:

Find \( u^{FE} \in X^{hp}_{FE}(\Omega) \subset H^1_0(\Omega) \) such that, \( \forall w^{FE} \in H^1_0(\Omega) \)

\[
\int_{\Omega} (\nabla u^{FE})^T \kappa \nabla w^{FE} d\Omega = \int_{\Omega} Q w^{FE} d\Omega + \int_{\Gamma_f} \tilde{f} w^{FE} d\Gamma_f \tag{2.5}
\]

where \( H^1_0(\Omega) \subset H^1(\Omega) \) whose functions satisfy the essential boundary conditions (2.2). Plugging finite element shape functions into (2.5), we arrive at the fully discretized form of (2.1)

\[
\int_{\Omega} [B]_T [\kappa][B] dV \{d\} = \int_{\Omega} [N]_T Q dV + \int_{\Gamma_f} [N]_T \tilde{f} ds \tag{2.6}
\]

where \([N]_{(1 \times ndofs)}\) is the row matrix of finite element shape functions, and \([B]_{(3 \times ndofs)}\) contains the gradients of the finite element shape functions. The same boundary conditions as seen in the strong form of the governing equations will be enforced also in the weak form. Using the terms from (2.6) the stiffness matrices and force vectors can be calculated element by element, and assembled appropriately to construct the global system of equations.

### 2.2 Generalized FEM Approximations

The generalized FEM [2, 3, 29, 81, 96] is one instance of the partition of unity method. This method has its origins in the works of Babuška et al. [2, 3, 72] (under the names “special finite element methods”, “generalized finite element method” and “finite element partition of unity method”) and Duarte and Oden [26, 32–34, 81] (under the names “\(hp\) clouds” and ”cloud-based \(hp\) finite element method”). Several meshfree methods proposed in recent years can also be viewed as special cases of the partition of unity method. In the GFEM, discretization spaces for a Galerkin method are defined using the concept of a partition of unity and local spaces that are built based on
a-priori knowledge about the solution of a problem. A shape function, $\phi_{ai}$, in the GFEM is computed from the product of a linear finite element shape function, $\varphi_{\alpha}$, and an enrichment function, $L_{ai}$.

$$\phi_{ai}(x) = \varphi_{\alpha}(x)L_{ai}(x) \quad \text{(no summation on } \alpha),$$

(2.7)

where $\alpha$ is a node in the finite element mesh. Figure 2.1 illustrates the construction of GFEM shape functions.

The linear finite element shape functions $\varphi_{\alpha}$, $\alpha = 1, \ldots, N$, in a finite element mesh with $N$ nodes constitute a partition of unity, i.e., $\sum_{\alpha=1}^{N} \varphi_{\alpha}(x) = 1$ for all $x$ in a domain $\Omega$ covered by the finite element mesh. This is a key property used in partition of unity methods. An a-priori error estimate for partition of unity approximations and, in particular, for the generalized finite element method, was proved by Babuška et al. [2, 3, 72].

![Figure 2.1](image)

Figure 2.1: Construction of a generalized FEM shape function using a polynomial (a) and a non-polynomial enrichment (b). Here, $\varphi_{\alpha}$ is the function at the top, the enrichment function, $L_{ai}$, is the function in the middle, and the generalized FE shape function, $\phi_{ai}$, is shown at the bottom.

**Enrichment functions** The GFEM has been successfully applied to the simulation of boundary layers [27], dynamic propagating fractures [30], line singularities [29], acoustic problems with
high wave number [4], polycrystalline microstructures [93], porous materials [96], plastic fracture mechanics [39], etc. A thorough summary of recent advances in the GFEM/XFEM is presented in [9].

All these applications have relied on closed-form enrichment functions that are known to approximate well the physics of the problem. These custom or special enrichment functions are able to provide more accurate and robust simulations than the polynomial functions traditionally used in the standard FEM, while relaxing some meshing requirements of the FEM. However, for many classes of problems—like those involving multiscale phenomena or non-linearities—enrichment functions with good approximation properties are, in general, not available analytically. In Section 2.4, a procedure to numerically build enrichment functions is presented for problems exhibiting highly localized sharp thermal gradients. The approach is based on the solution of local boundary value problems and can be used when no or limited a-priori knowledge about the solution is available.

### 2.3 Model Problem

A model problem representative of thermal loads experienced by a hypersonic vehicle subjected to a Type IV interaction (Cf. Section 1) is defined in this section. This problem is used to assess the performance of the FEM and the GFEM when solving problems with solutions exhibiting highly localized sharp thermal gradients. The solution of the model problem is given by

$$u(x) = \exp^{-\gamma(x-x_0)^2} + \sin\left(\frac{\pi x}{L}\right)$$

(2.8)

where $x_0 = 125\ mm$, $L = 500\ mm$ and $\gamma$ is a parameter controlling the roughness of the solution. Unless otherwise indicated, the value of $\gamma$ is taken as 1.0. The temperature profile (2.8) is shown in Figure 2.2. The temperature distribution on a plate $\Omega$ has a sharp localized spike in a small neighborhood of $x_0$, similar to the types of distributions described in [21, 104]. This model problem was originally proposed by Merle and Dolbow [75] and was also analyzed by O’Hara [84].
The domain is taken as \( \Omega = \{ \mathbf{x} \in \mathbb{R}^3 : 0 < x < 500, \ 0 < y < 250, \ 0 < z < 30 \} \) where all dimensions are in \( mm \). Homogeneous Dirichlet boundary conditions are applied on faces \( x = 0 \) and \( x = 500 \) and homogeneous Neumann boundary conditions are prescribed on all other faces. A heat source given by

\[
Q(x) = -\nabla^2 u(x),
\]

with \( u(x) \) defined in (2.8), is prescribed in \( \Omega \).

![Temperature profile of model problem.](image)

Figure 2.2: Temperature profile of model problem. The solution is smooth everywhere in the domain except in a small neighborhood of \( x_0 = 125 \ mm \), where a sharp temperature spike develops.

The energy norm associated with the problem defined in Section 2.1 is given by

\[
\|u\|_E = \sqrt{B(u,u)} = \sqrt{\int_{\Omega} (\nabla u \cdot \nabla u) \, d\Omega}
\]

where \( B(u,u) \) is the bilinear form associated with the Laplace operator.

In the numerical experiments presented below, the accuracy of a numerical approximation \( u_h \) of \( u \) is measured using the relative error in the energy norm, i.e.,

\[
e^E_E = \frac{\|u - u_h\|_E}{\|u\|_E} = \sqrt{\frac{B(u,u) - B(u_h,u_h)}{B(u,u)}}
\]
With the temperature field varying only in the $x$ direction, the exact energy of the solution can be obtained from (2.9), where $A$ is the cross-section area (in 3-D) in the $yz$ plane. MATLAB was used for the symbolic integration of (2.9). The reference value for the energy corresponding to (2.8) is

$$B(u,u) = 9474.62$$

$$B(u,u) = \left| \int_0^l -\kappa \left( \frac{du}{dx} \right)^2 dx \cdot A \right|$$

(2.9)

### 2.3.1 Convergence Analysis

In this section, the model problem described above is solved using the FEM and the GFEM. 1-, 2- and 3-D discretizations are used. This is possible due to the 1-D nature of the exact solution. In all the numerical experiments presented below, a high order Gaussian quadrature rule was used to compute the load vector over elements near the thermal spike at $x_0$. This is required due to the non-polynomial nature of heat source $Q(x)$. In the case of 3-D discretizations, a tensor-product Gaussian rule with 729 points is used. This rule was selected such that the convergence studies presented below are not affected by integration errors. Details on the numerical experiments used in the selection of this rule can be found in [84].

For 1-D analyses, the domain is a 1-D bar, 500 $mm$ in length, discretized with either 3-node $p$-hierarchical elements, or 2-node GFEM elements. Figure 2.3 shows how the 1-D meshes are broken up into three regions. The left- and right-most regions have fixed element sizes $h_L$ and $h_R$, respectively. The middle region, containing the spike ($120 \; mm \leq x \leq 130 \; mm$), is the only region which is refined, with element size $h_i$. For 2-D analyses, the domain is 500 $mm$ in length, 2 $mm$ in width, and discretized with 8-node (quadratic) or 4-node (linear) quadrilateral finite elements. Uniform meshes are used in the 2-D case. Results are also presented using quadratic, GFEM triangular elements on a mesh which is 500 $\times$ 250 $mm$. For the 3-D analyses, the domain is discretized using 4-node GFEM tetrahedral elements. In the 3-D case the mesh is again locally
refined, as shown in Figure 2.4, where refinement is done only in the portion of the domain which contains the peak. The element size in corresponding plots refers to the length in the $x$-direction of the smallest elements in the refined region.

Figure 2.3: Typical structure of the locally refined meshes used in the 1-D model. Element size in subsequent plots refers to $h_i$.

Figure 2.4: Locally refined 3-D mesh using a bounding box to define the region of local mesh refinement.

Convergence in energy norm of 1-D FEM and GFEM discretizations is shown Figure 2.5. Linear and quadratic $p$-hierarchical FEM [97] and two-node quadratic GFEM [29, 81] elements are used. The convergence rates are denoted in the plot as $B'$. From the plot, we can observe that quadratic GFEM and $p$-hierarchical FEM deliver the same level of accuracy. Thus, the curves for these elements coincide. The curves also show that there is a delay in reaching the optimal rate of convergence due to the rough nature of the solution, and the difficulty in resolving the localized
thermal spike. The asymptotic convergence rates obtained are very close to the optimal rates of 1.0 for linear elements ($B = 0.97$), and 2.0 for quadratic elements ($B = 1.96$).

Figure 2.5: Convergence in energy norm for low order 1-D generalized and $p$-hierarchical finite elements. Quadratic GFEM and $p$-hierarchical FEM deliver the same level of accuracy. Thus, only two curves can be seen in the plot.

Figure 2.6 shows convergence in energy norm for quadratic discretizations with 1-D and 3-D GFEM elements; 1-D $p$-hierarchical elements; and 2-D Serendipity elements. The relative error in energy norm is plotted against element size in the x-direction. The convergence behavior is similar in each of the four discretization sequences used, achieving near the theoretical convergence rate of 2.0 ($B = 1.96$).

Figure 2.7 shows the convergence in energy norm of 3-D GFEM discretizations. The data for the quadratic element is the same as in Figure 2.6, but here the relative error in energy norm is plotted versus the number of degrees of freedom instead of element size. It is quite apparent that in 3-D, the required element size to achieve acceptable error values translates into a very large number of degrees of freedom on highly graded meshes, as illustrated in Figure 2.8. In the case of linear elements, nearly $10^6$ degrees of freedom are required to achieve an error level below 10%. With this in mind, and considering that the geometry of the domains of interest are much more complex than in our model problem and that time dependent effects must be considered, a more efficient
Figure 2.6: Convergence in energy norm for discretizations with 1-D and 3-D GFEM elements; 1-D $p$-hierarchical elements; and 2-D quadratic Serendipity elements. The curves for 1-D elements coincide.

solution methodology is required. One point to take note of is that in Figure 2.7, the relative error values are cut off at 1.0. As can be seen in the plot, there is a pre-convergent zone, where the error is 100 percent or higher, before a minimal level of refinement is reached. The pre-convergent regions on the curves are due to the mesh being too coarse to capture the localized behavior of the solution. A similar phenomenon is observed in [64] in which the capability of the finite element method to solve Helmholtz’s equation is investigated.

**Thermal gradient not aligned with mesh**

In all the discretizations used previously, the spike in the loading was favorably oriented with respect to the mesh. In the previous analyses, the gradient in the temperature profile was in the global $x$-direction, as shown in Figure 2.2. The discretizations used for the analyses are favorable in that they have element edges which are perpendicular to the global $x$-direction, or the direction of $\nabla u$. This may not always be the case in practice, however; because the orientation of shock waves and thermal loadings may not line up with a primary axis of the coordinate system used to create the mesh or if an unstructured mesh is used. The effect of the orientation of the elements with respect to the gradient in the temperature profile is investigated in this section.
Figure 2.7: Convergence in energy norm for 3-D linear and quadratic tetrahedral elements. Sequences of meshes locally refined around the thermal spike are used.

Figure 2.8: Example of a highly graded mesh and the associated well-resolved spike in the temperature field.
The model problem with roughness parameter $\gamma = 0.05$ is solved on domains with different orientations with respect to the gradient in the temperature profile. In the first case, the domain is as defined in Section 2.3 while in the second one the domain is rotated 45 degrees clockwise. In this case, the line along the thermal spike cuts the elements at a 45 degree angle. Figure 2.9 shows one mesh with this orientation. Neumann boundary conditions derived from the analytic solution (2.8) are applied to all faces of this domain. Quadratic tetrahedral GFEM elements are used in both cases. The reference value for the exact strain energy for the case of the domain oriented as in the previous section is taken as $B(u,u) = 2179.21$. In the second case, the reference value is taken as $B(u,u) = 2992.80$, and was obtained using a mesh with 23 levels of local refinement.

Figure 2.10 shows the convergence in the energy norm for the two domain orientations considered. From this plot it can be seen that a significant increase in the number of degrees of freedom, in some instances up to 100 times more, is required to solve for the situation when the peak does not line up with the mesh, and in fact with this situation there is a pre-convergent zone which does not show up for the case where the peak is aligned with the mesh. While we are not solving the same problem in both cases, the smoothness of the solution is the same. Thus it is reasonable to attest that the difference in convergence between the two cases is mainly due to the change of orientation of the thermal layer with respect to the mesh.

Figure 2.9: Temperature distribution computed on a mesh where the line along the thermal peak cuts the domain at a 45 degree angle. Roughness parameter $\gamma = 0.05$. 

41
From the numerical experiments presented above, it is clear that the approximation of functions exhibiting highly localized gradients requires strongly refined meshes. High order elements can reduce the need for mesh refinement but not eliminate it [84]. \(H^p\) discretizations in which both element size and polynomial order are optimally distributed in the domain [22–24, 80, 85, 92] are able to deliver exponential convergence for problems like the one analyzed here. Optimal \(h^p\) discretizations use strongly refined meshes around regions with sharp thermal gradients. This creates some difficulties in the case of, e.g., time-dependent problems. The refinement/unrefinement must follow a moving internal layer and thus the problem must be solved from scratch after each mesh update even in the case of linear problems. The adaptive construction of the \(h^p\)-discretization also requires several adaptive cycles on the large, global discretization. In the next sections, the possibility of exploring the flexibility provided by the generalized FEM is investigated to avoid mesh refinement/unrefinement cycles and instead use customized enrichment functions able to approximate well the behavior of the solution on a fixed, coarse mesh. Avoiding mesh refinement/unrefinement will be important in the consideration of transient problems due to the energy conserving nature of avoiding the re-meshing process, as proven in [20].
2.3.2 GFEM with Special Enrichment Functions

In all numerical experiments presented in previous sections, only polynomial enrichment functions are used. As a result, a high level of mesh refinement is required in order for acceptable error levels to be obtained. Merle and Dolbow [75], demonstrated that far greater efficiency can be achieved when a-priori knowledge of the solution is used, and an exponential enrichment function of the form

\[ L_{\alpha i}(x) = \exp\left(-(x-x_0)^2\right) \]  

is used to create GFEM shape functions specifically tailored to solve the model problem previously described.

For the purpose of comparison, a 1-D mesh consisting of five, equally-sized, quadratic GFEM elements and 12 degrees of freedom is used to solve the model problem, yielding a relative error in the energy norm of 0.996. When the element containing the thermal spike is enriched with the exponential enrichment function (2.10) the relative error in the energy norm drops to 1.58e-3, a three-orders of magnitude reduction by adding two degrees of freedom to the discretization. Figure 2.11 shows the solution obtained with this discretization. Table 2.1 shows the effect of adding the exponential enrichment functions for 2 and 3D simulations as well.

Table 2.1: Summary of Output for Exponential Enrichment Functions.

<table>
<thead>
<tr>
<th>Exponential Enrichment</th>
<th>(e_e^{\text{energy norm}})</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>0.996</td>
<td>1</td>
</tr>
<tr>
<td>Yes</td>
<td>1.58e-3</td>
<td>1</td>
</tr>
<tr>
<td>No</td>
<td>0.8894</td>
<td>2</td>
</tr>
<tr>
<td>Yes</td>
<td>0.0019</td>
<td>2</td>
</tr>
<tr>
<td>No</td>
<td>0.8617</td>
<td>3</td>
</tr>
<tr>
<td>Yes</td>
<td>0.0105</td>
<td>3</td>
</tr>
</tbody>
</table>

This numerical experiment demonstrates that it is possible to achieve a high level of accuracy using coarse meshes provided appropriate enrichment functions are used. Nonetheless, enrichments able to approximate well small scale behavior like the one exhibited in our model problem are,
in general, not known. Thus, a more general approach to building special enrichment functions is needed. The proposed approach is based on the generalized finite element method with global-local enrichments ($GFEM^{gl}$) presented in [28, 31]. As demonstrated in the next sections, the so-called global-local enrichments can be defined even when limited or no a-priori information about the solution of a problem is available while enabling the use of coarse macro-scale meshes.

2.4 Generalized Finite Element Analysis with Global-Local Enrichments

In this section, we present a procedure to build enrichment functions for the class of problems governed by (2.1) and subjected to boundary conditions (2.2) and (2.3). The formulation and application of the $GFEM^{gl}$ to 3-D elasticity equations can be found in [28, 31, 66]
2.4.1 Formulation of Global Problem

Consider a domain $\Omega_G = \Omega_G \cup \partial \Omega_G$ as illustrated in Figure 2.12(a). The boundary is decomposed as $\partial \Omega_G = \Gamma_G^u \cup \Gamma_G^f$ with $\Gamma_G^u \cap \Gamma_G^f = \emptyset$. The solution $u$ of the global or macroscale problem obeys Poisson’s equation (2.1) on $\Omega_G$ and the boundary conditions prescribed on $\partial \Omega_G$ and given by (2.2) and (2.3). A generalized FEM approximation, $u_G^0$, of the solution $u$ can be found solving the following problem:

Find $u_G^0 \in X_{hp}^G(\Omega_G) \subset H^1(\Omega_G)$ such that, $\forall w_G^0 \in X_{hp}^G(\Omega_G)$

$$\int_{\Omega_G} (\nabla u_G^0)^T \kappa \nabla w_G^0 d\Omega + \eta \int_{\Gamma_G} u_G^0 w_G^0 d\Gamma = \int_{\Omega_G} Q w_G^0 d\Omega + \int_{\Gamma_G^f} \bar{f} w_G^0 d\Gamma + \eta \int_{\Gamma_G^u} \bar{u} w_G^0 d\Gamma$$

(2.11)

where, $X_{hp}^G(\Omega_G)$ is a discretization of $H^1(\Omega_G)$ built with generalized FEM shape functions, and $\eta$ is a penalty parameter. The enforcement of the Dirichlet boundary condition could also be done using, e.g., the Nitsche method or the Characteristic function method. Details on these methods, as well as their theoretical analysis within the framework of the GFEM, are presented in the survey paper by Babuska et al. [5]. In this research, the penalty method is used due to its simplicity of implementation.

Problem (2.11) leads to a system of linear equations for the unknown degrees of freedom of $u_G^0$.

The mesh used to solve problem (2.11) is typically a coarse quasi-uniform mesh. This global or macroscale problem (2.11) is denoted hereafter as initial global problem for convenience.

2.4.2 Local Problems

Let $\Omega_L$ denote a subdomain of $\Omega_G$ as illustrated in Figure 2.12(b). In this research, we consider the case in which the solution $u$ exhibits a strong internal layer, in the form of a sharp spike, in the local domain $\Omega_L$.

The following local problem is solved on $\Omega_L$ after the global solution $u_G^0$ is computed as described above:
Find $u_L \in X_L^{hp}(\Omega_L) \subset H^1(\Omega_L)$ such that, $\forall \, w_L \in X_L^{hp}(\Omega_L)$

$$
\int_{\Omega_L} (\nabla u_L)^T \kappa \nabla w_L \, d\Omega + \eta \int_{\partial \Omega_L \setminus (\partial \Omega_L \cap \Gamma^f_G)} u_L w_L \, d\Gamma = \\
\eta \int_{\partial \Omega_L \setminus (\partial \Omega_L \cap \partial \Omega_G)} u^0_G w_L \, d\Gamma + \eta \int_{\partial \Omega_L \cap \Gamma^u_G} \bar{u} w_L \, d\Gamma + \int_{\Omega_L} Q w_L \, d\Omega + \int_{\partial \Omega_L \cap \Gamma^f_G} \bar{f} w_L \, d\Gamma
$$

(2.12)

where, $X_L^{hp}(\Omega_L)$ is a discretization of $H^1(\Omega_L)$ using GFEM shape functions.

A key aspect of problem (2.12) is the use of the generalized FEM solution of the global problem, $u^0_G$, as boundary condition on $\partial \Omega_L \setminus (\partial \Omega_L \cap \partial \Omega_G)$. Exact boundary conditions are prescribed on portions of $\partial \Omega_L$ that intersect either $\Gamma^u_G$ or $\Gamma^f_G$. Problem (2.12) is named hereafter *local problem* for convenience.

### 2.4.3 Global-Local Enrichment Functions

The procedure described above to compute the local solution $u_L$ is the well known global-local analysis [25, 42, 79]. This procedure enables the computation of local quantities of interest while not requiring modifications on the usually large and complex global mesh. It is also computationally efficient since a single global analysis needs to be performed, even when local quantities
must be computed at several subdomains $\Omega_L \subset \Omega_G$. However, the error of the local solution, $u_L$, depends not only on the discretization used in local domain $\Omega_L$, but also on the quality of boundary conditions used on $\partial \Omega_L \setminus (\partial \Omega_L \cap \partial \Omega_G)$, which are provided by the global solution $u_G^0$. One approach to address the poor accuracy of these boundary conditions is to use a sufficiently large local domain. Nonetheless, the minimum size of $\Omega_L$ for acceptable results is problem dependent. In particular, for the class of problems we are interested, the error of the global solution $u_G^0$ may be large even far from the thermal spike. This is illustrated in Figure 2.13. Figure 2.14 illustrates the same idea, but with decreasing values of $\gamma$. As can be seen from the figure, as $\gamma$ decreases and the solution becomes smoother, the point-wise error decreases and becomes more localized. It may be argued from Figure 2.14 that the largest value of $\gamma$ still yields error of less than 0.01 °F, and is thus negligible. While this is a valid point, the actual values of the error are not of interest so much as the non-localization of the error itself, which is clearly demonstrated. The actual values themselves are not of much interest because they are significantly dependent upon the description of the loading function itself. This property is illustrated in Figure 2.15, where $\gamma$ is increased from 1.0 to 1.2, and the point-wise error values are significantly larger with this small change in the loading function. Thus, it can be concluded that the local solution $u_L$ will have in general a large error, even when very fine meshes are used in the local domain.

In the GFEM with global-local enrichments (GFEM$^{gl}$) the poor accuracy of $u_L$ is addressed by going one step further in the analysis and using $u_L$ as an enrichment function for the global discretization. Generalized FEM shape functions for the global problem are defined as

$$\phi_\alpha = \varphi_\alpha u_L$$  \hspace{1cm} (2.13)

where $\varphi_\alpha$ denotes a partition of unity function of the coarse global mesh and $u_L$ is called a global-local enrichment function. The function defined in (2.13) is used at nodes $x_\alpha$ of the global mesh whose support, $\omega_\alpha$, is contained in the local domain $\Omega_L$. The global problem enriched with these functions is solved and quantities of interest computed. The solution of this enriched global prob-
Figure 2.13: Error of solution computed on a uniform global mesh with 50 elements in the \( x \)--direction and one element in the \( y \)--direction. Standard eight-node quadratic finite elements are used. The location of the thermal spike is indicated in the figure. We can observe that the error in the computed temperature is relatively large even far from the thermal spike. Error when the exact solution is given by \( u(x) = \sin(\pi x/L) \), i.e., the thermal spike \( e^{-\gamma(x-x_0)^2} \) is removed is also provided. In this case, the error of the finite element solution is very small.

Figure 2.14: Error of solution computed on a uniform global mesh with 50 elements in the \( x \)--direction and one element in the \( y \)--direction. Standard eight-node quadratic finite elements are used. In this case we can see that as \( \gamma \) becomes smaller, the error decreases, and also becomes more localized.
Figure 2.15: Error of solution computed on a uniform global mesh with 50 elements in the x-direction and one element in the y-direction. Illustrates that the actual error values are highly dependent upon the description of the loading function.

lem is hereafter denoted by $u^E_G$. The $GFEM^{gl}$ approach is illustrated in Figure 2.12. The global solution provides boundary conditions for local problems while local solutions are used as enrichment functions for the global problem through the partition of unity framework of the GFEM. The procedure described above may be repeated. The solution $u^E_G$ is used as boundary conditions for the local problem and so on. This strategy is investigated in Section 2.5.2.

The enriched global problems do not have, in general, to be solved from scratch since the shape function (2.13) is hierarchically added to the global space and only a small number of nodes in the global problem is enriched. This is demonstrated in [28]. The relation of the $GFEM^{gl}$ with other methods is also discussed in [28].

The performance of the $GFEM^{gl}$ when solving steady-state heat transfer problems with solutions exhibiting highly localized sharp thermal gradients is investigated in the next sections.
2.5 Analysis of Model Problem Using the $GFEM^{gl}$

The generalized FEM with global-local enrichment functions ($GFEM^{gl}$) described above is used in this section to solve the model problem defined in Section 2.3. The global, $\Omega_G$, and local, $\Omega_L$, domains are discretized with four node tetrahedral GFEM elements [29]. Quadratic and quartic ($p = 4$) elements are used in global and local domains, respectively. Uniform meshes in $x$-, $y$- and $z$-direction are used in the global domain. The meshes are created by first generating a mesh of hexahedral elements and then dividing each element into 6 tetrahedral elements. Hereafter, meshes are defined based on the number of hexahedral elements used in their generation, not the number of tetrahedral elements. Each global mesh has 2 elements in the $y$-direction, and 1 element in the $z$-direction. Mesh 0x has 10 elements in the $x$-direction, Mesh 1x has 20 elements in the $x$-direction, and Mesh 2x has 40 elements in the $x$-direction. The global domains show increasing levels of refinement in the $x$-direction only because the solution only exhibits a gradient in the $x$-direction, and is constant in $y$ and $z$. These meshes are shown in Figures 2.21(c), 2.22(c) and 2.23(c), respectively.

Creation of Local Problems  Local domains and their corresponding initial discretizations are defined by copying elements from the global mesh around the thermal spike. This is done with the aid of global seed nodes which are selected via a bounding box containing the temperature spike. For the analyses presented here, the same bounding box is used for each of the three global meshes–Meshes 0x, 1x and 2x. As such, the smallest possible bounding box size is determined by the coarsest global mesh, Mesh 0x. The bounding box is defined from min = $[100,0,0]$ to max = $[150,250,30]$. Figure 2.16 illustrates this procedure. Let $I_{seed}$ denote the indices of all global seed nodes in the bounding box. A local domain corresponding to a mesh with one layer of elements around the seed nodes is given by

$$\Omega_L^{\text{lay}=1} := \bigcup_{\beta \in I_{\text{seed}}} \omega_\beta$$
Figure 2.16: Extraction of initial local mesh from global Mesh 1x. The bounding box used for selection of seed nodes is shown (rectangle) along with the seed nodes (solid circles). Nine seed nodes are shown, but there are eighteen in total: Nine on top surface (shown) and nine on the bottom surface of mesh.

where $\omega_\beta$ is the union of (copy of) global elements sharing node $x_\beta$, $\beta \in I_{seed}$. Local domains with additional layers of elements around the seed nodes are defined analogously. The mesh corresponding to a local domain with $m$ layers of elements around a given set $I_{seed}$ is given by the union of (copy of) the mesh with $m-1$ layers and the global elements sharing a vertex node in the mesh with $m-1$ layers.

The size of the local domains are also kept constant for each global mesh used, and once again are determined by Mesh 0x. One layer of elements in Mesh 0x is selected, resulting in the local domain $\Omega_L = \{x \in \mathbb{R}^3 : 50 < x < 200, 0 < y < 250, 0 < z < 30\}$. Two and four layers of elements around the seed nodes are used for Meshes 1x and 2x, respectively. Again, this was selected to maintain a constant size in the local domain.

The local meshes themselves are then refined by bisecting all tetrahedral elements inside of the bounding box defined by $\text{min} = [122.5, 0, 0]$ and $\text{max} = [127.5, 250, 30]$ for the case when $\gamma = 1.0$. A local mesh is shown in Figure 2.17 corresponding to Mesh 1x with 9 levels of local refinement.
The seed nodes used to create the local domains are the same nodes which are then enriched with the local solution (global-local enrichments). Twelve nodes are enriched on Mesh 0x; eighteen nodes are enriched on Mesh 1x and thirty nodes are enriched on Mesh 2x. This is illustrated in Figures 2.21(c), 2.22(c) and 2.23(c).

The number of degrees of freedom (DOFs) in the global problem remains almost constant when global-local enrichments are used. In addition, the number of DOFs in the enriched global problem does not depend on the number of DOFs in the local problem. Therefore, the number of DOFs in the enriched global problem is not a good measure for the computational cost of $\nu_E^G$. In the convergence analyses presented hereafter, the computational cost for both local and global problems are measured with respect to CPU time, not number of DOFs. All measures are in seconds. For plots dealing with local problems, the CPU time considers the time taken for assembly and solution of the local problem. Plots dealing with the enriched global domain consider the CPU time taken for assembly and solution of the enriched global domain as well as the assembly and solution time.
taken in the corresponding local domain. The CPU times are meant to reflect the total computational effort required to generate the solution of interest, which is the underlying reason for the selection of each component considered in each case.

2.5.1 H-extensions in the Local Problem

The convergence of the enriched global problem when $h$-extensions are performed in the local problem is investigated in this section. The local problems are solved using Dirichlet boundary conditions on $\partial \Omega_L \setminus (\partial \Omega_L \cap \partial \Omega_G)$ provided by the initial global problem as discussed in Section 2.4.2. For reasons which will become more clear in the subsequent sections, the methodology used in this section will be referred to as Initial Global Problem with Spike, or IGw/S.

Figure 2.18 shows the relative error in energy norm in the enriched global problems associated with meshes 0x, 1x and 2x. All three cases show convergence of the enriched global problem as the local problems are refined. The global mesh is kept fixed for each curve shown, only the global-local enrichments (solution of local problems) are updated. The CPU time on the horizontal axis includes the CPU time taken for assembly and solution of the enriched global domain as well as the assembly and solution time taken in the corresponding local domain. Thus, as the local domains are refined, the reported CPU time increases. From the figure, we can observe that the pre-asymptotic range reduces as finer global meshes are used. In addition, for a given computational effort the accuracy of the enriched global solution computed on Mesh 2x can be up to one order of magnitude higher than on the other two meshes.

Figure 2.19 shows the relative error in energy norm in the local problems subjected to boundary conditions provided by global solutions computed on meshes 0x, 1x and 2x. Very large errors and poor or no convergence can be observed. Local problems subjected to boundary conditions from global meshes 1x and 2x initially show convergence but then the error levels off. This shows that the poor quality of the boundary conditions is controlling the error. This is confirmed in Sections 2.5.2 and 2.5.3 where we present two approaches to improve the quality of the boundary
conditions for the local problems. Interestingly, the global problems enriched with these poor local solutions show convergence as discussed above, attesting the robustness of the proposed $GFEM_{gl}$.

However, the convergence of the enriched global problem will at some point level off since the local solutions do not converge to the solution of the global problem due to errors in boundary conditions applied to local domains. The leveling off of the enriched global errors can be observed in Figure 2.20 which shows more data points than in Figure 2.18. The convergence for Meshes 1x and 2x level off due to poor quality of local solutions. The convergence for other meshes are also expected to eventually level off. In Sections 2.5.2 and 2.5.3 we propose two approaches to extend the convergent range of the enriched global problem.

Figures 2.21, 2.22 and 2.23 show temperature distributions computed in each phase of the $GFEM_{gl}$–initial global, local, and enriched global problems–corresponding to global meshes 0x, 1x and 2x, respectively, and 13 levels of refinement in the local problems. Solutions of initial global problems solved with meshes 0x and 1x completely miss the thermal spike and as a result the local solutions are of poor quality. The thermal spike, however, is clearly captured in all three enriched global
Figure 2.19: Relative error in energy norm for local problems subjected to boundary conditions provided by global solutions computed on meshes 0x, 1x and 2x. The CPU time considers the time taken for assembly and solution of the local problem.

Figure 2.20: Data of Figure 2.18 plotted against the number of DOFs in the local problems. The plot includes data points that could not be included in Figure 2.18 due to a limitation of the function we use to measure CPU time.
problems, attesting the importance of the extra step in the proposed $GFEM^{31}$. This is in agreement with the convergence plots shown in Figures 2.18 and 2.19. One other point of interest is the resolution of the spike in Figure 2.21(c) where a well-resolved spike can be seen in the temperature field even with the use of very large elements. In fact, the spike in the temperature field falls within elements, and not along a line of nodes, reflecting the shape of the specially-tailored enrichment functions from the local problem.

Figure 2.21: Temperature distributions computed in the initial global, local, and enriched global problems corresponding to global Mesh 0x and 13 levels of refinement in the local problem. The $IGw/S$ methodology is used. The thermal spike is well resolved in the enriched global problem even though it falls within quite large elements. Enriched nodes in global domain are denoted in Figure 2.21(c) by red glyphs.
(a) Solution initial global problem.

(b) Solution of local problem.

(c) Solution of enriched global problem.

Figure 2.22: Temperature distributions computed in the initial global, local, and enriched global problems corresponding to global Mesh 1x and 13 levels of refinement in the local problem. The IGw/S methodology is used. Enriched nodes in global domain are denoted in Figure 2.22(c) by red glyphs.
(a) Solution of initial global problem.

(b) Solution of local problem.

(c) Solution of enriched global problem.

Figure 2.23: Temperature distributions computed in the initial global, local, and enriched global problems corresponding to global Mesh 2x and 13 levels of refinement in the local problem. The IGw/S methodology is used. Enriched nodes in global domain are denoted in Figure 2.23(c) by red glyphs.
Expanding the Size of Local Domain to Improve BCs

In this section, a technique commonly used in the GL-FEM, where the number of layers used to define the local domain is increased so as to obtain boundary conditions further from the region of localized interest, is examined. In this case the size of the local problem increases slightly, but the added dofs are not near the local region, so they are not relied upon to improve the solution. The extra layers are used based upon the assumption that if the boundary conditions are obtained from further away from the local region, they will be of better quality. Figure 2.24 illustrates the difference between the meshes used for the simulations in this section. As can be seen, the meshes are identical locally, but there are layers of coarse elements added in order to give the local domain access to theoretically better boundary conditions.

![Figures of local domains with 9 levels of local refinement, and 1 and 6 layers of elements.](image)

Figure 2.24: Figures of local domains with 9 levels of local refinement, and 1 and 6 layers of elements.

Figure 2.25 shows the convergence in the energy norm for a series of simulations run, each with 9 levels of local refinement, and increasing numbers of element layers used to define the local domain. As can be seen from the plot, very poor convergence behavior is seen. This behavior is not necessarily unexpected, since as was shown in Figure 2.13 the error in the solution may be large even far from the local region of interest. As a result, simply obtaining boundary conditions from further away from the spike will not necessarily provide better quality boundary conditions, and as a results will not yield better convergence in the local domain.
Figure 2.25: Shows the lack of convergence in the energy norm when the local refinement is kept constant, but the number of layers used in creating the local problem is increased. Circled point indicates the data point from Figure 2.20.

2.5.2 A Two-Step Approach to Improve Local Solutions

As discussed in Section 2.4.3, the error of the global solution $u^0_G$ may be large even far from the thermal spike (Cf. Figure 2.13). Thus, local problems may be subjected, for the class of problems we are interested, to poor boundary conditions. As a result, the error of the local solutions can not be controlled simply by mesh refinement or element enrichment (Cf. Figure 2.19). In this section, we address this issue by performing one additional global-local cycle. The solution $u^E_G$ is used as boundary conditions for the local problems and then we proceed as before–Solve the local problems and enrich the global discretization with local solutions. For simplicity this approach is hereafter referred to as $IGw/S-II$–the two-step version of $IGw/S$. This particular approach is investigated because it has potential to be exploited in a transient solution to a time-dependent problem when the enriched global solution from time step $t^n$ can be used as boundary conditions for the local problem at time step $t^{n+1}$.

Figure 2.26 shows the temperature distributions computed in the initial global, local, and enriched global problems of the $IGw/S-II$ strategy. Here, the initial global problem corresponds to the enriched global problem in the $IGw/S$ strategy. The thermal spike is well resolved in this problem.
and thus improved BCs are imposed on the local problem which can also capture well this behavior and, in turn, provide good enrichment functions for the enriched global problem (Cf. Figure 2.26(c)).

Figure 2.26: Temperature distributions computed in the initial global, local, and enriched global problems corresponding to global Mesh 2x and 13 levels of refinement in the local problem. The IGw/S-II strategy is used. The thermal spike is well resolved in the initial global problem and thus the local problem can also capture well this behavior.

Figure 2.27 shows the relative error in energy norm in the local problems in strategies IGw/S and IGw/S-II. Local boundary conditions are provided by global solutions computed on meshes 0x, 1x and 2x. We can observe a dramatic difference between the two strategies. As expected, the local
problems in the second step of the *IGw/S-II* strategy are subjected to much improved boundary conditions than in the first step which lead to the improvement seen in convergence.

![Graph](image_url)

**Figure 2.27:** Relative error in energy norm for local problems in strategies *IGw/S* and *IGw/S-II*. The only difference in the local problems is the boundary conditions used.

Figures 2.28 shows the relative error in energy norm for enriched global problems in strategies *IGw/S* and *IGw/S-II*. All three cases, Meshes 0x, 1x and 2x, show convergence of the enriched global problem as the local problems are refined. The behavior of the energy norm is not as dramatically different between the two strategies, which is evidence of the robustness of the GFEM\textsuperscript{gl} to take local solutions which may be very poor and still deliver reasonable convergence in global domain. Nonetheless, some differences do exist at low error levels. The enriched global solution in strategy *IGw/S-II* does not level off as in the *IGw/S* strategy. Thus, the *IGw/S-II* strategy extends the range of target error level for the enriched global problem. Of course the convergence of the enriched global *IGw/S-II* may eventually level off but at a lower error level than in the *IGw/S* strategy. This behavior has not, however, been experienced in any numerical experiments performed so far.
2.5.3 Spike Absent from Initial Global Problem

In this section, we investigate another approach to improve the boundary conditions for the local problems and thus extend the range of target error level for the enriched global problem. In the strategy investigated here, the rough portion of the thermal loading applied to the initial global problem is removed. This idea is based on the fact that the spike in the temperature profile is very localized, and it has virtually no effect on the exact solution outside of a small neighborhood of the thermal spike. Therefore, as long as the local problem boundaries are not within a few millimeters of the temperature peak, the correct boundary conditions are essentially those from the smooth portion of the loading and the exponential portion will have no appreciable effect. For simplicity, this approach will be referred to as Initial Global problem without Spike, or IGw/oS.

Figure 2.29 shows the temperature distributions computed in the initial global, local, and enriched global problems of the IGw/oS strategy. The thermal spike is well resolved in both the local and enriched global problems even though it is absent in the initial global problem.

Figures 2.30 shows the relative error in energy norm in the local problems in strategies IGw/S and IGw/S-II. As in the case of strategy IGw/S-II, we can observe a dramatic improvement on the
(a) Solution of initial global problem in $IGw/oS$ strategy. The thermal spike is absent from this solution.

(b) Solution of local problem.

(c) Solution of enriched global problem.

Figure 2.29: Temperature distributions for initial global, local, and enriched global problems for $IGw/oS$ strategy corresponding to global Mesh 2x, 13 levels of refinement in the local problem. Meshes 0x and 1x provide qualitatively similar results. The thermal spike is well resolved in both the local and enriched global problems.
convergence behavior of the local solutions when strategy \( IGw/oS \) is used. In fact, the performance of strategies \( IGw/oS \) and \( IGw/S-II \) is very similar (Cf. Figure 2.27).

The only difference in the local problems is the boundary conditions used. In the case of \( IGw/oS \) strategy, the thermal spike was removed from the initial global domain.

Figure 2.31 shows the relative error in energy norm for enriched global problems in strategies \( IGw/S \) and \( IGw/oS \). We can observe convergence of the enriched global solution computed with strategy \( IGw/oS \) over a larger range of target error level than in the case of \( IGw/S \) strategy.

### 2.5.4 Effect of Enriching the Global Problem

The \( GFEM^{gl} \), as noted previously, differs from the traditional global-local FEM in that there is the added step of enriching the global domain with the local solution and re-solving the global problem. In this section, we compare the performance of these two methods for each of the strategies proposed in previous sections, i.e., \( IGw/S \), \( IGw/S-II \) and \( IGw/oS \).

Figure 2.32 compares the convergence in energy norm in the local and enriched global domains for strategy \( IGw/S \). In these plots, there is a significant difference in the convergence rates as well as the error values between local and enriched global solutions. In some cases, the enrichment of the global domain can take local solutions which show no convergence behavior, and a large relative
error, and convert this local information into a global solution which shows good convergence behavior, as well as significantly lower error values.

Figure 2.31: Relative error in energy norm for enriched global problems in strategies $IGw/S$ and $IGw/oS$.

Figures 2.32 and 2.34 compare the convergence in energy norm in the local and enriched global domains for strategy $IGw/S-II$. The local domains, in this case, are provided with good boundary conditions, and thus the local domains themselves do provide accurate solutions. As a result, the
improvement is not as drastic as that seen in the IGw/S case. The only noticeable improvement is in the case of Mesh 2x and at low error levels (Cf. Figure 2.34). As was mentioned earlier, the scenario where a well-resolved spike is used in the initial global problem is of particular interest because it will be relied upon particularly in the transient setting, where the enriched global problem of one time step may be used to provide accurate boundary conditions for the local problem in the next time step. This methodology, if successful, will provide us with the ability to resolve very fine local features using a fixed, coarse global mesh throughout the entire transient analysis.

![Figure 2.33: Comparison of convergence in energy norm in local and enriched global domains for IGw/S-II strategy.](image)

Figure 2.33 compares the convergence in energy norm in the local and enriched global domains for IGw/oS strategy. As in the IGw/S-II case, the local domains are provided with good boundary conditions, so the local domains are able to generate accurate solutions. As a result, the improvement is once again not as drastic as that seen in the IGw/S case.

### 2.6 Summary

In this chapter, the generalized FEM with global-local enrichments (GFEM<sup>gl</sup>) [28, 31, 66] is formulated for steady-state heat transfer problems with solutions exhibiting highly localized sharp
Figure 2.34: Data of Figure 2.33 plotted against the number of DOFs in the local problems. The plot includes data points that could not be included in Figure 2.33 due to a limitation of the function we use to measure CPU time.

Figure 2.35: Comparison of convergence in energy norm in local and enriched global domains for IGw/oS strategy.
thermal gradients.

The proposed method is related to the classical global-local FEM (GL-FEM) \cite{25, 42, 79} which is broadly used in the industry. They share several attractive features like

(i) the possibility of capturing localized solution features using uniform, coarse, global meshes. This removes, for example, the need to refine global meshes that are usually complex and very large. A single global mesh can be used to analyze the effect of localized thermal loads at different parts of a structure. All that is needed is the computation of local solutions and the hierarchical enrichment of the global solution space. Additional computational implications of this feature of the $GFEM^{gl}$ are discussed in Section 2.3.1 and in \cite{28};

(ii) the size of the enriched global problem is about the same as the initial global problem and it does not depend on the size or discretization used in the local problems;

(iii) while not explored in this research, it is conceivable to use in the $GFEM^{gl}$ different approximation methods to solve the global and local problems, like in the GL-FEM. $Hp$ adaptive finite elements methods \cite{22–24, 80, 85, 92}, for example, would be an excellent option for solving the local problems;

(iv) the solution of multiple local problems can be parallelized without difficulty allowing the solution of large problems very efficiently;

(v) the GL-FEM uses the same variational principle as the original problem and thus no stability issues are introduced by the method.

While the $GFEM^{gl}$ share many of the attractive features of the GL-FEM, the numerical experiments presented here and in \cite{28, 66}, demonstrate that the $GFEM^{gl}$ is much more robust than the GL-FEM. The former is able to deliver accurate global solutions even when limited or no convergence is observed in the local problems. The errors in the enriched global problem are, in some cases, orders of magnitude smaller than in the local problems. The difficulties of the GL-FEM with the class of problems investigated here is due to the large errors of global solutions computed on
coarse meshes. This is illustrated in Figure 2.13 which shows that the discretization error may be large even far from the thermal spike.

The numerical experiments presented here also demonstrate that the information transfer between local (fine) and global (coarse) scales using the partition of unity framework is very effective (Cf. Section 2.5.4). It is shown that the global problem converges at least as fast as the local problems and in many cases the enriched global problem can deliver much more accurate solutions than the local ones.

The $GFEM^{gl}$ brings the benefits of GFEM to problems were limited or no information about the solution is known a-priori. The only information used to obtain the global solutions shown on the coarse global meshes 2.21(c), 2.22(c) and 2.23(c) was that the solution has a localized behavior.

From the numerical experiments, it is also found that:

Coarse, uniform, global meshes are acceptable even at regions with thermal spikes that are orders of magnitude smaller than the element size. The element size in that region depends on the target error level. Finer global meshes reduce or eliminate the pre-asymptotic region in the convergence. Global convergence is achieved even when no convergence is observed in the local problems. Two approaches are proposed to improve the boundary conditions of the local problems and their convergence. The use of the corresponding improved local solutions as enrichment for the global problem is beneficial by extending the range of target error level for the enriched global problem.

As noted previously, the first method for improving the boundary conditions is of particular interest due to the potential to use the solution of the enriched global problem at $t^n$ as boundary conditions for the local domain at $t^{n+1}$ in transient simulations. This methodology is further developed in the next chapter, where transient effects are considered.
Chapter 3

Transient Heat Conduction

3.1 Problem Formulation

This chapter investigates transient heat transfer problems with solutions exhibiting highly localized sharp thermal gradients. Consider a domain $\Omega \subset \mathbb{R}^3$ with boundary $\partial \Omega$ decomposed as $\partial \Omega = \Gamma_c \cup \Gamma_f$ with $\Gamma_c \cap \Gamma_f = \emptyset$. The strong form of the governing equation is given by the 3D heat equation

$$\rho c \frac{\partial u}{\partial t} = \nabla (\kappa \nabla u) + Q(x, t) \quad \text{in} \quad \Omega \quad (3.1)$$

where $u(x,t)$ is the temperature field, $\rho c$ is the volumetric heat capacity, and $Q(x,t)$ is the internal heat source. In the general case, $\kappa$ is the thermal conductivity tensor, but in this instance, only isotropic materials are considered, therefore the thermal conductivity is merely a scalar, $\kappa = \kappa(x)$, and the material need not be homogeneous.

Both convective and Neumann boundary conditions prescribed on $\Gamma_c$ and $\Gamma_f$, respectively are considered. The boundary $\partial \Omega = \Gamma_c \cup \Gamma_f$ and $\Gamma_c \cap \Gamma_f = \emptyset$. At any time $t$, the normal flux is prescribed as

$$-\kappa \frac{\partial u}{\partial n} = \eta (\bar{u} - u) \quad \text{on} \quad \Gamma_c \quad (3.2)$$

$$-\kappa \frac{\partial u}{\partial n} = \bar{f} \quad \text{on} \quad \Gamma_f \quad (3.3)$$

where $\bar{u}$ and $\eta$ are the prescribed fluid temperature and convective coefficient, respectively.

Dirichlet boundary conditions can be treated as a limiting case of convective boundary conditions by selecting a large value for the convective coefficient $\eta$. This leads to the well known penalty
method [1].

The initial conditions must also be satisfied

\[ u(x, 0) = u^0(x) \quad \text{at} \quad t^0 \]  

(3.4)

where \( u^0(x) \) is the prescribed temperature field at time \( t = t^0 \).

3.2 Time Integration and Discrete Equations

In this section (3.1) is discretized in a finite element context. In the first formulation (3.1) is discretized first in space, and then in time. With this formulation strategy, the algorithm is appropriate for analyses which do not include time-dependencies in the shape functions. For the case with time-dependent shape-functions it is important to discretize the equations first in time, and then in space, as is discussed by Fries and Zilian [52]. The formulation for time-dependent shape functions is subsequently presented in 3.2.2.

3.2.1 Formulation 1: Discretizing Heat Equation Spatially, then Temporally

In this section two different, widely used formulations for time-stepping algorithms for the transient heat equation are presented. Both methods are equivalent, with slightly different formulations yielding systems of equations which are not form-equivalent, but both amount to the generalized trapezoidal rule, or the \( \alpha \)-method. The first formulation is presented by Reddy [87], and the second is presented by Hughes [63].
Formulation for $\alpha$-method (Reddy)

A standard formulation for time-integration of first order, parabolic equations can be found in many FEM books, one such instance is [87]. The formulation in [87], given by Reddy starts with a system of spatially discretized equations as in (3.5), and utilizes the finite difference assumption in (3.6).

\[
M\dot{u}^{n+1} + Ku^{n+1} = f^{n+1} \tag{3.5}
\]

\[
u^{n+1} = u^n + \Delta t \left[ \alpha \dot{u}^{n+1} + (1 - \alpha) \dot{u}^n \right] \tag{3.6}
\]

Equation (3.6) is used to eliminate $\dot{u}^{n+1}$ from (3.5), yielding the discrete system of equations used for time-integration:

\[
[M + \alpha \Delta t K] u^{n+1} = [M - (1 - \alpha) \Delta t K] u^n + \Delta t \left[ \alpha f^{n+1} + (1 - \alpha) f^n \right] \tag{3.7}
\]

The potential draw-back of this particular formulation is that it starts from a fully, spatially-discretized system, in which the quantities have previously been defined as

\[
M_{ij}^e = \int_{\Omega^el} \rho c \phi_i \phi_j d\Omega^el \tag{3.8}
\]

\[
K_{ij}^{el} = \int_{\Omega^el} \kappa \frac{\partial \phi_i}{\partial x_p} \frac{\partial \phi_j}{\partial x_q} d\Omega^el \quad \text{(summation on p,q = 1,2,3),} \tag{3.9}
\]

\[
F_i^e = \int_{\Omega^el} Q(x,t) \phi_i d\Omega^el + \int_{\Gamma_f^e} \bar{f}(x,t) \phi_i d\Gamma_f^el \tag{3.10}
\]

Where $\phi$ is the vector of finite element shape functions, $\Omega^el$ is the domain, $\Gamma_f^e$ is the boundary of the domain with prescribed, normal fluxes, $Q(x,t)$ is the internal source, $\rho c$ is the volumetric heat
capacity, \( \tilde{f}(\mathbf{x}, t) \) is the prescribed heat flux, and \( \kappa_{pq} \) is the thermal conductivity of the material. Again, we are assuming an isotropic material, therefore, \( \kappa_{pq} = \kappa \quad \forall p, q \). \( \{ u^{n+1} \} \) is the solution vector at \( t^{n+1} \) and \( \{ \dot{u}^{n+1} \} \) is the derivative with respect to time. (3.7) is then computed on an element-by-element basis and assembled to form the global system of equations. The previous formulation is the widely used \( \alpha \)-method, suitable for transient heat transfer simulations, but the formulation is in no way modified to incorporate the use of time-dependent shape functions.

**Formulation for \( \alpha \)-method (Hughes)**

The following formulation is presented in [63]. In this case Hughes starts from a spatially discretized system of equations. The spatially-discrete governing equations of are:

\[
\mathbf{M} \mathbf{v}^{n+1} + \mathbf{K} \mathbf{u}^{n+1} = \mathbf{f}^{n+1} \tag{3.11}
\]

\[
\mathbf{u}^{n+1} = \mathbf{u}^{n} + \Delta t \mathbf{v}^{n+\alpha} \tag{3.12}
\]

\[
\mathbf{v}^{n+\alpha} = (1 - \alpha) \mathbf{v}^{n} + \alpha \mathbf{v}^{n+1} \tag{3.13}
\]

where \( \alpha \in [0, 1] \). The following predictor (3.14) and corrector (3.15) are used to define \( \mathbf{u}^{n+1} \). One can then solve for \( \mathbf{v}^{n+1} \) in terms of \( \mathbf{u}^{n+1} \) and other known quantities, and effectively eliminate \( \mathbf{v}^{n+1} \) from (3.5).

\[
\hat{\mathbf{u}}^{n+1} = \mathbf{u}^{n} + (1 - \alpha) \Delta t \mathbf{v}^{n} \tag{3.14}
\]

\[
\mathbf{u}^{n+1} = \hat{\mathbf{u}}^{n+1} + \alpha \Delta t \mathbf{v}^{n+1} \tag{3.15}
\]
\begin{equation}
\mathbf{v}^{n+1} = \frac{\mathbf{u}^{n+1} - \hat{\mathbf{u}}^{n+1}}{\alpha \Delta t} \tag{3.16}
\end{equation}

(3.16) is now plugged into (3.5) and the following equation is posed which can be algebraically manipulated and solved for \( \mathbf{u}^{n+1} \).

\begin{equation}
M \frac{\mathbf{u}^{n+1} - \hat{\mathbf{u}}^{n+1}}{\alpha \Delta t} + \mathbf{K} \mathbf{u}^{n+1} = \mathbf{f}^{n+1} \tag{3.17}
\end{equation}

Moving known quantities to the right-hand-side and quantities dependent upon \( \mathbf{u}^{n+1} \) to the left-hand-side yields

\begin{equation}
M \frac{\mathbf{u}^{n+1}}{\alpha \Delta t} + \mathbf{K} \mathbf{u}^{n+1} = \mathbf{f}^{n+1} + M \frac{\hat{\mathbf{u}}^{n+1}}{\alpha \Delta t} \tag{3.18}
\end{equation}

(3.14) is then plugged in to (3.18) and simple algebra can be used to arrive at the following equations yielding \( \mathbf{u}^{n+1} \) and \( \mathbf{v}^{n+1} \), respectively.

\begin{equation}
\frac{1}{\alpha \Delta t} [M + \alpha \Delta t \mathbf{K}] \mathbf{u}^{n+1} = \mathbf{f}^{n+1} + \frac{M}{\alpha \Delta t} [\mathbf{u}^{n} + (1 - \alpha) \Delta t \mathbf{v}^{n}] \tag{3.19}
\end{equation}

\begin{equation}
\mathbf{v}^{n+1} = \frac{\mathbf{u}^{n+1} - \hat{\mathbf{u}}^{n+1}}{\alpha \Delta t} = \frac{\mathbf{u}^{n+1} - (\mathbf{u}^{n} + (1 - \alpha) \Delta t \mathbf{v}^{n})}{\alpha \Delta t} \tag{3.20}
\end{equation}

At this point it is reasonable to verify that (3.19) and (3.7) are equivalent. First (3.7) is divided by \( \alpha \Delta t \) and some terms are rearranged as follows

\begin{equation}
\frac{1}{\alpha \Delta t} [M + \alpha \Delta t \mathbf{K}] \mathbf{u}^{n+1} = \frac{\alpha \Delta t \mathbf{f}^{n+1}}{\alpha \Delta t} + \frac{M}{\alpha \Delta t} \mathbf{u}^{n} - \frac{(1 - \alpha) \Delta t \mathbf{K} \mathbf{u}^{n}}{\alpha \Delta t} + \frac{(1 - \alpha) \Delta t \mathbf{f}^{n}}{\alpha \Delta t} \tag{3.21}
\end{equation}

From (3.21) it can be deduced that the left-hand-side is equivalent for both equations, as well as the first two terms on the right-hand-side. The question is now to verify that the following is true:
\[
\frac{M}{\alpha \Delta t} (1 - \alpha) \Delta v^n = -\frac{(1 - \alpha) \Delta t K u^n}{\alpha \Delta t} + \frac{(1 - \alpha) \Delta t f^n}{\alpha \Delta t} \quad (3.22)
\]

which can be rearranged as follows:

\[
\frac{(1 - \alpha) \Delta t}{\alpha \Delta t} M v^n = \frac{(1 - \alpha) \Delta t}{\alpha \Delta t} \{ f^n - K u^n \} \quad (3.23)
\]

Dropping the coefficient common to both sides of the equation yields the following equality, which is known to be true from (3.7).

\[
M v^n = \{ f^n - K u^n \} \quad (3.24)
\]

From the previous it is apparent that the two formulations are not form-equivalent, but are in fact mathematically equivalent as would be expected.

### 3.2.2 Formulation 2: Discretizing Heat Equation Temporally, then Spatially

In this section the system of equations is discretized first in time, then in space [15, 52, 102]. With this formulation, the algorithm is appropriate for the use of time-dependent shape functions. The formulation starts with the strong form of the governing equation:

\[
\rho c \frac{\partial u}{\partial t} = \nabla \cdot \kappa \nabla u + Q \quad (3.25)
\]

The equation is multiplied by a weighting function, \( w \), and integrated over the domain, \( \Omega \).

\[
\int_{\Omega} w \rho c \frac{\partial u}{\partial t} d\Omega = \int_{\Omega} (w \nabla \cdot \kappa \nabla u + w Q) d\Omega \quad (3.26)
\]

Integration by parts is performed on the first term of the right-hand-side, and the domain integral is
moved to the left-hand-side of the equation. The boundary term is left on the right-hand-side with the applied source term.

\[
\int_{\Omega} \left( w \rho c \frac{\partial u}{\partial t} + \nabla w \cdot \kappa \nabla u \right) \, d\Omega = \int_{\partial \Omega} w \kappa \nabla u \cdot \hat{n} \, d\Gamma + \int_{\Omega} w Q \, d\Omega \tag{3.27}
\]

The term \( \nabla u \cdot \hat{n} \) is equivalent to \( \frac{\partial u}{\partial n} \), and when multiplied by the conductivity, \( \kappa \), yields the applied normal flux. The weak form of the system of equations, which will subsequently be discretized first in time, then in space is posed as follows

\[
\int_{\Omega} \left( w \rho c \frac{\partial u}{\partial t} + \nabla w \cdot \kappa \nabla u \right) \, d\Omega = \int_{\partial \Omega} w \kappa \frac{\partial u}{\partial n} \, d\Gamma + \int_{\Omega} w Q \, d\Omega \tag{3.28}
\]

The system of equations is first discretized in time. To this end the following finite difference approximations are used, yielding the generalized trapezoidal rule, or \( \alpha \)-method, used for the time-marching scheme.

\[
\frac{\partial u}{\partial t} = \frac{u^{n+1} - u^n}{\Delta t} \tag{3.29}
\]

\[
u^{n+1} = (1 - \alpha)\nu^n + \alpha u^{n+1} \tag{3.30}
\]

Plugging (3.29) and (3.30) into (3.28) yields the temporally discretized system of equations.

\[
\int_{\Omega} \left( w \rho c \frac{u^{n+1} - u^n}{\Delta t} + \nabla w \cdot \kappa \left[ (1 - \alpha)\nabla u^n + \alpha \nabla u^{n+1} \right] \right) \, d\Omega = \int_{\Omega} w \left[ \alpha Q^{n+1} + (1 - \alpha)Q^n \right] \, d\Omega + \int_{\partial \Omega} w \kappa \left[ \alpha \frac{\partial u^{n+1}}{\partial n} + (1 - \alpha)\frac{\partial u^n}{\partial n} \right] \, d\Gamma \tag{3.31}
\]

The boundary terms on \( \partial \Omega = \Gamma_c \cup \Gamma_f \) are considered first in detail, as follows:
\[
\int_{\partial \Omega} w \kappa \left[ \alpha \frac{\partial u^{n+1}}{\partial n} + (1 - \alpha) \frac{\partial u^n}{\partial n} \right] d\Gamma = \alpha \int_{\Gamma_f} w \bar{f}^{n+1} d\Gamma + (1 - \alpha) \int_{\Gamma_f} w \bar{f}^n d\Gamma + \\
\alpha \int_{\Gamma_c} w \gamma \bar{u}^{n+1} d\Gamma - \alpha \int_{\Gamma_c} w \gamma u^n d\Gamma + (1 - \alpha) \int_{\Gamma_c} w \gamma \bar{u}^n d\Gamma - (1 - \alpha) \int_{\Gamma_c} w \gamma u^n d\Gamma
\]

(3.32)

(3.31) is rearranged, with the proper boundary terms (3.32), such that terms involving \(u^{n+1}\) are moved to the left-hand-side, and all known terms (those not dependent upon \(u^{n+1}\)) are moved to the right-hand-side.

\[
\frac{1}{\Delta t} \int_{\Omega} \rho c u^{n+1} d\Omega + \alpha \int_{\Omega} \nabla \cdot \kappa \nabla u^{n+1} d\Omega + \alpha \int_{\Gamma_c} w \gamma u^{n+1} d\Gamma = \frac{1}{\Delta t} \int_{\Omega} \rho c u^n d\Omega + \\
- (1 - \alpha) \int_{\Omega} \nabla \cdot \kappa \nabla u^n d\Omega + \alpha \int_{\Omega} w Q^n d\Omega + (1 - \alpha) \int_{\Omega} w Q^n d\Omega + \alpha \int_{\Gamma_f} w \bar{f}^{n+1} d\Gamma + \\
+ (1 - \alpha) \int_{\Gamma_f} w \bar{f}^n d\Gamma + \alpha \int_{\Gamma_c} w \gamma \bar{u}^{n+1} d\Gamma + (1 - \alpha) \int_{\Gamma_c} w \gamma \bar{u}^n d\Gamma - (1 - \alpha) \int_{\Gamma_c} w \gamma u^n d\Gamma
\]

(3.33)

In the previous equation, the loading terms are defined as \(Q^n = Q(x, t^n)\) and \(\bar{f}^n = \bar{f}(x, t^n)\).

At this point, the system of equations is fully discretized in time. For the spatial discretization, generalized finite element shape functions are used, which may have time-dependencies. At any given time, \(t^n\), one defines \(u^n (x, t^n) = \Phi^n (x, t^n) \cdot u^n (t^n)\), where \(u^n (t^n)\) is the vector of degrees of freedom, and \(\Phi^n (x, t^n)\) is the vector of finite element shape functions at \(t^n\). Due to the potential time-dependent nature of the shape functions, it is very important to properly select the discretization for the weight function, \(w\). In the current implementation, \(w\) is required to be consistent across each term of (3.33). To this end, the weighting function, \(w\), is discretized using finite element shape functions at time \(t^{n+1}\). In the aforementioned equations, \(w = w^{n+1} (x, t^{n+1}) = \Phi^{n+1} (x, t^{n+1}) \cdot w^{n+1} (t^{n+1})\). (3.33) is now discretized on a term-by-term basis using the previously defined discretizations.
\[
\int_{\Omega} \rho c w^{n+1} u^{n+1} d\Omega = (w^{n+1})^T \int_{\Omega} \phi^{n+1} \rho c (\phi^{n+1})^T d\Omega u^{n+1} = (w^{n+1})^T M^{n+1} u^{n+1} 
\] (3.34)

\[
\int_{\Omega} \rho c w^{n+1} u^n d\Omega = (w^{n+1})^T \int_{\Omega} \phi^{n+1} \rho c (\phi^n)^T d\Omega u^n = (w^{n+1})^T M^{n+1} u^n 
\] (3.35)

\[
\int_{\Omega} \nabla w^{n+1} \cdot \kappa \nabla u^{n+1} d\Omega = (w^{n+1})^T \int_{\Omega} \nabla \phi^{n+1} \kappa (\nabla \phi^{n+1})^T d\Omega u^{n+1} = (w^{n+1})^T K^{n+1} u^{n+1} 
\] (3.36)

\[
\int_{\Omega} \nabla w^n \cdot \kappa \nabla u^n d\Omega = (w^{n+1})^T \int_{\Omega} \nabla \phi^n \kappa (\nabla \phi^n)^T d\Omega u^n = (w^{n+1})^T K^{n+1} u^n 
\] (3.37)

\[
\int_{\Omega} w^{n+1} Q^{n+1} d\Omega = (w^{n+1})^T \int_{\Omega} \phi^{n+1} Q^{n+1} d\Omega = (w^{n+1})^T f^{n+1} 
\] (3.38)

\[
\int_{\Omega} w^n Q^n d\Omega = (w^{n+1})^T \int_{\Omega} \phi^n Q^n d\Omega = (w^{n+1})^T f^{n+1,n} 
\] (3.39)

\[
\int_{\Gamma_f} w^{n+1} \vec{f}^{n+1} d\Gamma = (w^{n+1})^T \int_{\Gamma_f} \phi^{n+1} \vec{f}^{n+1} d\Gamma = (w^{n+1})^T f^{n+1} 
\] (3.40)

\[
\int_{\Gamma_f} w^n \vec{f}^n d\Gamma = (w^{n+1})^T \int_{\Gamma_f} \phi^n \vec{f}^n d\Gamma = (w^{n+1})^T f^{n+1,n} 
\] (3.41)

\[
\int_{\Gamma_c} \gamma w^{n+1} u^{n+1} d\Gamma = (w^{n+1})^T \int_{\Gamma_c} \phi^{n+1} \gamma (\phi^{n+1})^T d\Gamma u^{n+1} = (w^{n+1})^T M^{n+1} u^{n+1} 
\] (3.42)
\[
\int_{\Gamma_c} \gamma w^{n+1} u^n d\Gamma = (w^{n+1})^T \int_{\Gamma_c} \phi^{n+1} \gamma (\phi^n)^T d\Gamma u^n = (w^{n+1})^T M_c^{n+1,n} u^n
\]  \(3.43\)

\[
\int_{\Gamma_c} \gamma w^{n+1} \bar{u}^{n+1} d\Gamma = (w^{n+1})^T \int_{\Gamma_c} \phi^{n+1} \gamma \bar{u}^{n+1} d\Gamma = (w^{n+1})^T f_c^{n+1}
\]  \(3.44\)

\[
\int_{\Gamma_c} \gamma w^{n+1} \bar{u}^n d\Gamma = (w^{n+1})^T \int_{\Gamma_c} \phi^{n+1} \gamma \bar{u}^n d\Gamma = (w^{n+1})^T f_c^{n+1,n}
\]  \(3.45\)

Since equation (3.33) must hold for any admissible weight function \(w\), it must hold also for any \(w^{n+1}\). As such, one can pose the fully discretized system of equations as

\[
\begin{bmatrix}
\frac{1}{\Delta t} M^{n+1} + \alpha K^{n+1} + \alpha M_c^{n+1} \\
\end{bmatrix} u^{n+1} = \begin{bmatrix}
\frac{1}{\Delta t} M^{n+1,n} - (1 - \alpha) K^{n+1,n} - (1 - \alpha) M_c^{n+1,n} \\
\end{bmatrix} u^n + \alpha f_Q^{n+1} + (1 - \alpha) f_Q^{n+1,n} + \alpha f_N^{n+1} + (1 - \alpha) f_N^{n+1,n} + \alpha f_c^{n+1} + (1 - \alpha) f_c^{n+1,n}
\]

More concisely, the above equation is re-written as:

\[
\begin{bmatrix}
\frac{1}{\Delta t} M^{n+1} + \alpha \hat{K}^{n+1} \\
\end{bmatrix} u^{n+1} = \begin{bmatrix}
\frac{1}{\Delta t} M^{n+1,n} - (1 - \alpha) \hat{K}^{n+1,n} \\
\end{bmatrix} u^n + \alpha f^{n+1} + (1 - \alpha) f^{n+1,n}
\]  \(3.47\)

where

\[
\hat{K}^{n+1} = K^{n+1} + M_c^{n+1}
\]  \(3.48\)

\[
\hat{K}^{n+1,n} = K^{n+1,n} + M_c^{n+1,n}
\]  \(3.49\)

\[
f^{n+1} = f_Q^{n+1} + f_N^{n+1} + f_c^{n+1}
\]  \(3.50\)
\[ f^{n+1,n} = f_Q^{n+1,n} + f_N^{n+1,n} + f_c^{n+1,n} \]  

(3.51)

In all of the above equations, \( u^n \) are known values obtained from the solution at \( t^n \). It is noted that if the shape functions are not time-dependent, \( M^{n+1}_n = M^{n+1,n} = M, \ K^{n+1}_n = K^{n+1,n} = K, \ f_Q^{n+1,n} = f_Q^n, \ f_N^{n+1,n} = f_N^n \) and \( f_c^{n+1,n} = f_c^n \). It is further noted that if the convective boundary terms are neglected, (3.47) is equivalent to (3.7).

For the analyses presented in the subsequent sections, the value of \( \alpha \) is taken as \( \alpha = 1 \), yielding the unconditionally stable, Backward Euler algorithm. As such, only the non-symmetric capacity matrix, \( M^{n+1,n} \) is required, and it need not be assembled directly. The vector term \( M^{n+1,n}u^n \) can be computed as

\[ M^{n+1,n}u^n = \int_{\Omega} \phi^{n+1}_n \rho c (\phi^n)^T u^n d\Omega = \int_{\Omega} \rho c \phi^{n+1}_n u^n d\Omega \]  

(3.52)

where \( u^n = (\phi^n)^T u^n \), is the GFEM solution from time step \( t^n \).

### 3.3 Model Problem

The problem selected for verification involves a sharp spatial gradient in the temperature field (3.53), as well as in the resulting source term (3.54). There is also a temporal gradient, but it is smooth in nature. This particular problem is taken from [75], with the modification that we are assuming a stationary spike, i.e. velocity, \( V_{\text{spike}} = 0 \).

\[ u(x,t) = \left( \exp^{-\gamma(x-x_0)^2} + \sin \left( \frac{\pi x}{L} \right) \right) * \exp(-t) \]  

(3.53)

\[ Q(x,t) = \rho c \frac{\partial u}{\partial t}(x,t) - \kappa \frac{\partial^2 u}{\partial x^2}(x,t), \]  

(3.54)

The initial and boundary conditions are given in (3.55) and (3.56), respectively.
\[ u(x, 0) = \exp^{-\gamma(x-x_0)^2} + \sin\left(\frac{\pi x}{L}\right), \quad (3.55) \]

\[ u(0, t) = u(L, t) = 0, \quad (3.56) \]

In the above equations, \( x_0 = 125 \text{mm} \), \( L = 500 \text{mm} \) and \( \gamma \) is a parameter controlling the roughness of the solution. Unless otherwise indicated, the value of \( \gamma \) is taken as 1.0. The material properties are taken as thermal conductivity, \( \kappa = 1 \) and volumetric heat capacity, \( \rho c = \left(\frac{\pi}{L}\right)^2 \). The reference solution (3.53) is plotted in Figure 3.1 and the initial condition (3.55) is plotted in Figure 3.2. The value of \( x_0 \) indicates the location of the thermal spike. From the temporal standpoint, the solution undergoes a smooth, exponential decay in time.

![Reference solution in space and time.](image)

![Time slices of reference solution.](image)

Figure 3.1: Temperature field as described in (3.53).

Below, the heat equation with \( Q \) given by (3.54), initial and boundary conditions given in (3.55) and (3.56), respectively, is solved using 1-, 2- and 3-D GFEM discretizations.

The exact internal energy, \( U(t) \), in the solution domain \( \Omega \) is given by

\[ U(t) = \int_{\Omega} (\kappa \nabla u) \cdot (\nabla u) \, d\Omega, \quad (3.57) \]
while the internal energy of the GFEM solution at time $t^n$ is given by

$$U_{hp}(t^n) = \int_{\Omega} (\kappa \nabla u^n) \cdot (\nabla u^n) \, d\Omega$$  \hspace{1cm} (3.58)

The discrete $L_2$ norm of $U(t), t \in [0,t_{\text{final}}]$, is defined as

$$\|U(t)\|_2 = \left\{ \sum_n (U(t^n))^2 \right\}^{1/2}$$  \hspace{1cm} (3.59)

where the summation is performed over each time step $t^n, n \in [0,t_{\text{final}}]$. The relative error of $U_{hp}(t)$ in the discrete $L_2$ norm is given by

$$L^2_{\text{Error}}(U(t)) = \frac{\|U_{\text{exact}}(t) - U_{hp}(t)\|_2}{\|U_{\text{exact}}(t)\|_2} = \left\{ \frac{\sum_n (U_{\text{exact}}(t^n) - U_{hp}(t^n))^2}{\sum_n (U_{\text{exact}}(t^n))^2} \right\}^{1/2}$$  \hspace{1cm} (3.60)

This quantity can serve to tell how well the GFEM and exact curves for internal energy versus time match up.

Table 3.1 contains a summary of the results for each case analyzed using only polynomial enrichments. For CPU Times recorded in subsequent tables and figures, only the forward/backward
substitution and factorization time are taken into account. The reason being, is that the assembly process, which also has a large impact on the CPU Time, can be very easily parallelized. With this parallelization potential, it is likely that the assembly time can be made insignificant when compared to the forward/backward substitution and factorization times. As a result, only the forward/backward substitution and factorization times are considered here.

Table 3.1: Summary of Output for Polynomial Elements.

<table>
<thead>
<tr>
<th>p-order</th>
<th>$h_x$</th>
<th>$L_2^{error}$</th>
<th>CPU Time (sec)</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.000</td>
<td>0.6752</td>
<td>0.1409</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>5.000</td>
<td>0.6721</td>
<td>1.025</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>2.500</td>
<td>0.3612</td>
<td>0.7092</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2.500</td>
<td>0.3604</td>
<td>3.075</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1.250</td>
<td>0.1463</td>
<td>4.3320</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1.250</td>
<td>0.1460</td>
<td>12.300</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>0.625</td>
<td>0.0908</td>
<td>27.5504</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>5.000</td>
<td>0.2797</td>
<td>0.6719</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>5.000</td>
<td>0.2792</td>
<td>4.7150</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>6.250</td>
<td>0.3132</td>
<td>2.0500</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>2.500</td>
<td>0.1979</td>
<td>4.5097</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2.500</td>
<td>0.1973</td>
<td>15.580</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3.125</td>
<td>0.1979</td>
<td>10.455</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>1.250</td>
<td>0.0688</td>
<td>30.1524</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1.250</td>
<td>0.0686</td>
<td>62.525</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1.563</td>
<td>0.0834</td>
<td>106.600</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>0.625</td>
<td>0.0029</td>
<td>203.679</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>5.000</td>
<td>0.1871</td>
<td>11.1642</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>5.000</td>
<td>0.1863</td>
<td>51.6600</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>2.500</td>
<td>0.0054</td>
<td>914.860</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>2.500</td>
<td>0.0053</td>
<td>1008.40</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>1.250</td>
<td>0.0002</td>
<td>1436.27</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0.625</td>
<td>1.79e-6</td>
<td>1551.21</td>
<td>1</td>
</tr>
</tbody>
</table>

Energy versus time curves for quadratic elements are shown in Figure 3.3. As can be seen the curves approach the reference curve as the refinement is increased, as would be expected. In order to plot 1-, 2-, and 3-D results on the same plot, the value reported is actually internal energy per unit area. In 2-D this is taken to be the internal energy divided by $W$, the width of the domain; and in 3-D it is taken as internal energy divided by $A$, the cross-sectional area on the $yz$-plane. The
convergence in the $L_2^{\text{Error}}$ value as a function of element size is plotted in Figure 3.4. Uniform meshes are used for 1- and 2-D simulations while graded meshes are used for the 3-D simulations.

Figure 3.3: Energy versus time plots for quadratic elements.

Figure 3.4: Error versus element size for 1D, 2D and 3D simulations.

### 3.3.1 GFEM Simulations Using Special Enrichment Functions

In this section results are presented for simulations of the model problem using special, exponential enrichment functions. The set of enrichment functions applied to nodes whose support intersect
the spike at \( x = x_0 \) is

\[
L_{\alpha i} = \left\{ 1, \frac{x - x_\alpha}{h_\alpha}, \exp\left(-\frac{(x-x_0)^2}{h_\alpha}\right) \right\}
\]  (3.61)

where \( x_\alpha \) is the \( x \)-coordinate of the node and \( h_\alpha \) is a scaling parameter equal to the size of the largest element sharing the node [29, 81]. The resulting GFEM shape functions built using (3.61) are not time-dependent. Nodes whose support do not intersect the spike are enriched with

\[
L_{\alpha i} = \left\{ 1, \frac{x - x_\alpha}{h_\alpha} \right\}
\]  (3.62)

which leads to quadratic GFEM shape functions in the \( x \)-direction.

For the 1-D discretizations, a coarse mesh consisting of 5, 100 mm long elements and a fine mesh with 200, 2.5 mm long elements are used. A comparison of element sizes for these meshes is shown in Figure 3.5. The coarse mesh uses enrichments (3.61) and has only 14 degrees of freedom while the fine mesh uses (only) enrichments (3.62) and has 402 degrees of freedom.

For the 2-D discretizations, a coarse and a fine mesh with elements of length 6 mm and 2.5 mm in the \( x \)-direction, respectively, are used. In the 3-D case, the coarse (fine) mesh has elements 6 mm (3.125 mm) in the \( x \)-direction near the spike, but 20 mm in the \( x \)-direction in regions far from the spike in order to save some computational effort. Nodes of the 2-D and 3-D coarse meshes whose support intersect the thermal spike are enriched with functions (3.61) while nodes of the fine meshes use (only) enrichments (3.62).

The simulation results are compared in Table 3.2 in order to illustrate the benefit of the special enrichments in terms of accuracy and efficiency. In the table, \( h_x \) stands for element size in the
As can be seen in the table, there is a significant reduction in CPU Time, as well as in the relative error, $L^2_{\text{error}}(U(t))$, for the meshes that use exponential enrichments, even though the elements are larger than in the meshes without exponential enrichments. It should also be noted that the size of the elements in the 2- and 3-D discretizations with exponential enrichments is restricted mainly due to our ability to accurately integrate the sharply varying source terms, as well as the exponential term in the shape functions. Internal energy versus time curves are plotted in Figure 3.6 for 1-, 2- and 3-D simulations. A zoomed in view for 1-D results is provided in Figure 3.7 to show a discernable difference in the two curves.

Table 3.2: Comparison of Output for Discretizations With and Without Exponential Enrichment Functions.

<table>
<thead>
<tr>
<th>Exponential Enrich.</th>
<th>$h_x (mm)$</th>
<th>$L^2_{\text{error}}(U(t))$</th>
<th>CPU Time (sec)</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>100</td>
<td>5.58e-5</td>
<td>0.0203</td>
<td>1-D</td>
</tr>
<tr>
<td>No</td>
<td>2.5</td>
<td>0.1979</td>
<td>4.5097</td>
<td>1-D</td>
</tr>
<tr>
<td>Yes</td>
<td>6</td>
<td>0.0019</td>
<td>1.640</td>
<td>2-D</td>
</tr>
<tr>
<td>No</td>
<td>2.5</td>
<td>0.1974</td>
<td>15.580</td>
<td>2-D</td>
</tr>
<tr>
<td>Yes</td>
<td>6</td>
<td>0.0076</td>
<td>1.230</td>
<td>3-D</td>
</tr>
<tr>
<td>No</td>
<td>3.125</td>
<td>0.1874</td>
<td>10.455</td>
<td>3-D</td>
</tr>
</tbody>
</table>

Figure 3.6: Plot of internal energy versus time for the 1-, 2-, and 3-D discretizations with exponential enrichments.
Figure 3.7: Plot of internal energy versus time for the special, exponential element. Only a very small portion of the plot is focused at a high level of zooming in order to see a discernible difference in the plots.

**Effect of Volumetric Heat Capacity Magnitude**

The previous results correspond to a volumetric heat capacity $\rho c = (\pi/L)^2$, yielding a value of $\rho c = 3.9e - 5$ for the model used here. With a very small value of $\rho c$ the transient effects are kept to a minimum. In this section, the effect of larger values of $\rho c$ on the accuracy of the internal energy is investigated. Figure 3.8 shows the internal energy versus time curves for discretizations using exponential enrichments, for larger values of $\rho c$. Table 3.3 summarizes the relative error, $L^2_\text{error}(U(t))$, obtained for larger values of $\rho c$ for discretizations with and without the use of the exponential enrichment functions. For each value of $\rho c$ investigated it is seen that the addition of the special, exponential enrichment function greatly improves the error levels by inserting the necessary information into the solution space, enabling a high degree of accuracy on a relatively coarse mesh. The accuracy, however, decreases as $\rho c$, and the transient effects, increase.
Figure 3.8: Internal energy versus time curves generated with exponential enrichment functions and increasing values of volumetric heat capacity.

Table 3.3: Effect of Volumetric Heat Capacity Magnitude.

<table>
<thead>
<tr>
<th>Exponential Enrich.</th>
<th>$L_2^{\text{err}}(U(t))$, $\rho c = \frac{\pi^2}{T^2}$</th>
<th>$L_2^{\text{err}}(U(t))$, $\rho c = 10$</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>5.58e-5</td>
<td>0.0115</td>
<td>1-D</td>
</tr>
<tr>
<td>No</td>
<td>0.9920</td>
<td>0.9920</td>
<td>1-D</td>
</tr>
<tr>
<td>Yes</td>
<td>0.0076</td>
<td>0.0250</td>
<td>3-D</td>
</tr>
<tr>
<td>No</td>
<td>0.8772</td>
<td>0.8654</td>
<td>3-D</td>
</tr>
</tbody>
</table>

89
3.3.2  GFEM Simulations Using Special, Time-Dependent Enrichment Functions

In this section, the same problem as in the previous section is solved, but now a time-dependency is inserted into the exponential GFEM shape functions through the use of the following enrichment basis

\[ L_{\alpha i} = \left\{ 1, \frac{x-x_\alpha}{h_\alpha}, \exp^{-\left(\frac{x-x_0}{h_\alpha}\right)^2}, \exp^{-t} \right\} \]  

(3.63)

With time-dependency inserted in the shape functions, it becomes important to distinguish which formulation, described in Section 3.2, is being used. We first investigate the use of time-dependent shape functions with Formulation 1 in which the heat equation is first discretized spatially, and then temporally (cf. Section 3.2.1). In other words, we seek to investigate the effect on solution accuracy of using the standard \( \alpha \)-method as commonly formulated with no specific modifications to accommodate for shape functions evolving in time.

Figure 3.9 shows the internal energy versus time curves for various values of \( \rho_c \) using Formulation 1. As can be seen, as the value of \( \rho_c \) is increased, the changes in the capacity matrix due to the changing enrichment functions become more significant. With very small values of \( \rho_c \) the problem behaves similarly to solving a series of steady-state problems, with no real thermal inertial effects being evident. It is seen from the plots that the time-dependency causes a deterioration in the behavior of the algorithm for large values of \( \rho_c \). Therefore, Formulation 1 is not appropriate for time-dependent shape functions.

With the previous results in mind, the performance of Formulation 2 (cf. Section 3.2.2) is investigated. In order to account for the time-dependency of the shape functions with \( \alpha = 1 \), one needs only modify the right-hand-side of (3.47) as shown in (3.64) to reflect the time-dependency of the capacity matrix, since the other modified terms in (3.47) drop out.

\[ RHS^{n+1} = F^{n+1} + \frac{M^{(n+1)}}{\Delta t} u^n \]  

(3.64)
Figure 3.9: Internal energy versus time curves generated with 1-D and 3-D meshes enriched with time-dependent exponential enrichment functions and Formulation 1.

where $\tilde{M}^{(n+1)}$ in (3.65) again, represents the coupling of time-steps $t^n$ and $t^{n+1}$ as per the formulations presented in [52, 102], and Section 3.2.2.

\[
\tilde{M}_{ij}^{el(n+1)} = \int_{\Omega_{el}} \rho c \phi_i^{n+1} \phi_j^n d\Omega_{el}
\] (3.65)

With the Transient Formulation 2, significantly improved behavior is obtained in the internal energy versus time curves for 1-D simulations with large values of $\rho c$, as shown in Figure 3.10. Table 3.4 shows the significant improvement of the $L_2^{error}(U(t))$ for each value of $\rho c$ investigated. Figure 3.11 shows the dramatic improvement obtained using Formulation 2 in 1- and 3-D simulations.

Table 3.4: Output for Elements with Time-Dependent Exponential Enrichment Functions.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>$\rho c$</th>
<th>$L_2^{error}(U(t))$ Form. 2</th>
<th>$L_2^{error}(U(t))$ Form. 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-D</td>
<td>10</td>
<td>0.0115</td>
<td>0.5537</td>
</tr>
<tr>
<td>1-D</td>
<td>50</td>
<td>0.0121</td>
<td>0.5584</td>
</tr>
<tr>
<td>1-D</td>
<td>100</td>
<td>0.0122</td>
<td>0.5590</td>
</tr>
<tr>
<td>3-D</td>
<td>10</td>
<td>0.0250</td>
<td>0.4139</td>
</tr>
</tbody>
</table>

From the previous analysis it is convincing that accurate results can be generated for transient
Figure 3.10: Internal energy versus time curves generated with 1-D time-dependent exponential enrichment functions, large values of $\rho c$, and Transient Formulation 2.

Figure 3.11: Internal energy versus time curves generated with 1- and 3-D time-dependent exponential enrichment functions.
simulations involving localized heat sources on coarse meshes. In general, this is only possible provided that an enrichment function with good approximation properties is available, and the proper transient formulation is used. In the general case, no such enrichment functions are known a priori. As such, the generation of appropriate enrichment functions on the fly via the solution of local boundary value problems is proposed, as in the GFEM with global-local enrichments (GFEM$^{\text{gl}}$). The transient formulation for the GFEM$^{\text{gl}}$ is provided in the next section.

3.4 GFEM$^{\text{gl}}$ for Time-Dependent Problems

In this section, a procedure is presented to build enrichment functions for the class of problems governed by (3.1) and subjected to boundary conditions (3.2) and (3.3). A formulation for the GFEM$^{\text{gl}}$ for steady-state heat transfer, along with applications can be found in Chapter 2. The formulation and application of the GFEM$^{\text{gl}}$ to three-dimensional elasticity equations can be found in [28, 31, 66].

3.4.1 Formulation of Transient Global Problem

Consider a domain $\hat{\Omega}_G = \Omega_G \cup \partial \Omega_G$ as illustrated in Figure 3.12(a). The boundary is decomposed as $\partial \Omega_G = \Gamma^u_G \cup \Gamma^f_G$ with $\Gamma^u_G \cap \Gamma^f_G = \emptyset$. The solution $u^n$ of the global or macroscale problem obeys the heat equation (3.1) on $\Omega_G$ and the boundary conditions prescribed on $\partial \Omega_G$ and given by (3.2) and (3.3) at time $t^n$. In this section, the $\alpha$-method with $\alpha = 1.0$ is used. As such we use a backward difference approximation in time to approximate the time derivative in (3.1).

A generalized FEM approximation, $u^n_G$, of the solution $u^n$ can be found solving the following problem:
Find $u^n_G \in S^{\text{GFEM},n}_{G}(\Omega) \subset H^1(\Omega)$ such that, $\forall \, w^n_G \in S^{\text{GFEM},n}_{G}(\Omega)$

$$
\frac{\rho c}{\Delta t} \int_{\Omega} w^n_G u^n_G d\Omega + \int_{\Omega} (\nabla w^n_G)^T \kappa \nabla u^n_G d\Omega + \eta \int_{\Gamma_c} u^n_G w^n_G d\Gamma = \frac{\rho c}{\Delta t} \int_{\Omega} w^n_G u^{n-1}_G d\Omega + \int_{\Gamma_f} f^n w^n_G d\Gamma + \eta \int_{\Gamma_c} \bar{u}^n w^n_G d\Gamma + \int_{\Omega} w^n_G Q^n d\Omega
$$

(3.66)

where $S^{\text{GFEM},n}_{G}(\Omega) \subset H^1(\Omega)$ is the generalized FEM space at time step $n$. The enrichment functions in $S^{\text{GFEM},n}_{G}(\Omega)$ are defined in local spaces and have to be computed; a fine-scale problem is presented in the next subsection to achieve this goal. Note that the same approximation space is used for the GFEM solution $u^n_G$ and the weight function $w^n_G$ as discussed in Section 3.2.2. The mesh used to solve problem (3.66) is typically a coarse quasi-uniform mesh, even when the solution is not smooth. Problem (3.66) leads to a system of linear equations for the unknown degrees of freedom of $u^n_G$.

### 3.4.2 Fine-Scale Problems at Time $t^n$

The proposed GFEM gl involves the solution of a fine-scale boundary value problem defined in a neighborhood $\Omega_L$ of thermal spikes, where strong solution gradients develop. The local domain $\Omega_L$ is composed of the union of clouds $\omega_\alpha$ of the open cover $\{\omega_\alpha\}_{\alpha=1}^N$ of $\Omega$ that intersect or are close to a thermal spike.

Having the global approximation $u^n_G$ at time $t^n$, one computes the following fine-scale problem on $\Omega_L$ to find enrichment functions for the space $S^{\text{GFEM},n+1}_G(\Omega)$:

Find $u^n_L \in S^{\text{GFEM},n}_{L}(\Omega_L) \subset H^1(\Omega_L)$ such that, $\forall \, w^n_L \in S^{\text{GFEM},n}_{L}(\Omega_L)$

$$
\int_{\Omega_L} (\nabla u^n_L)^T \kappa \nabla w^n_L d\Omega + \eta \int_{\partial \Omega_L \setminus (\partial \Omega_L \cap \Gamma_f)} u^n_L w^n_L d\Gamma = \eta \int_{\partial \Omega_L \setminus (\partial \Omega_L \cap \partial \Omega)} u^n_L w^n_L d\Gamma + \eta \int_{\partial \Omega_L \cap \Gamma_c} \bar{u}^{n+1} w^n_L d\Gamma + \int_{\Omega_L} Q^{n+1} w^n_L d\Omega + \int_{\partial \Omega_L \cap \Gamma_f} f^{n+1} w^n_L d\Gamma
$$

(3.67)
where, $S^{GFEM,n}_L(\Omega_L)$ is a discretization of $H^1(\Omega_L)$ using, e.g., standard GFEM shape functions. It is possible, however, to use other methods, like the standard FEM or the Boundary Element Method, to solve the fine-scale problems. The proposed methodology enables one to select the most effective method for the particular class of fine scale problem considered. Thus, the methodology is highly flexible and general.

A key aspect of problem (3.67) is the use of the coarse-scale solution at time step $t^n$, $u^n_G$, as boundary condition on $\partial\Omega_L \setminus (\partial\Omega_L \cap \partial\Omega)$. In the numerical experiments presented in Section 3.5, the parameter $\eta$ is taken as a penalty number. Thus Dirichlet boundary conditions are enforced on $\partial\Omega_L \setminus (\partial\Omega_L \cap \Gamma_f)$. Exact boundary conditions are prescribed on portions of $\partial\Omega_L$ that intersect either $\Gamma_c$ or $\Gamma_f$. Another key point of problem (3.67) is that no transient effects are considered. However, the source function, $Q$, and the boundary conditions on $\partial\Omega_L \cap \Gamma_c$ and $\partial\Omega_L \cap \Gamma_f$ are computed at time step $t^{n+1}$. The rationale for this is that $u^n_L$ is used to define the global solution space at time $t^{n+1}$, as described in the section below.

### 3.4.3 Scale-Bridging with Global-Local Enrichment Functions

The solution, $u^n_L$, of the fine-scale problem defined above is used to build generalized FEM shape functions defined on a coarse global mesh:

$$
\phi^{n+1}_{\alpha i}(x) := \varphi_\alpha(x) u^n_L(x) \quad (3.68)
$$

where the partition of unity function, $\varphi_\alpha$, is provided by a global, coarse, FE mesh and $u^n_L$ has the role of an enrichment or basis function for the patch space $\chi_\alpha(\omega_\alpha)$. Hereafter, $u^n_L$ is denoted a global-local enrichment function. The global GFEM space containing shape functions $\phi^{n+1}_{\alpha i}$ is denoted $S^{GFEM,n+1}_G(\Omega)$. The coarse scale problem (3.66) is solved for $u^{n+1}_G \in S^{GFEM,n+1}_G(\Omega)$ and the procedure is repeated at each time step. The $GFEM^{gl}$ for time-dependent problems is illustrated in Figure 3.12. The global solution provides boundary conditions for fine-scale problems while local solutions are used as enrichment functions for the coarse-scale problem through the partition
of unity framework of the GFEM.

Figure 3.12: Illustration of the GFEM for time-dependent problems. The global domain is discretized with a coarse mesh, even if the solution is not smooth. The coarse-scale problem at \( t^n \) provides boundary conditions for local problems containing sharp thermal spikes. The solution of these problems are used to build approximation spaces for the coarse-scale problem at time \( t^{n+1} \).

Once again, it is important to note that only a few degrees of freedom are added to the global (coarse-scale) discretization even if the computation of the fine-scale solution requires several thousands of degrees of freedom since \( u^n_L \) is a known function at time step \( t^{n+1} \).

The global problem is solved on the coarse global mesh enriched with the shape functions defined in (3.68). These functions are hierarchically added to the FE discretization, and thus, a few entries are added to element matrices while keeping existing ones associated with standard FE shape functions. The hierarchical nature of the global-local enrichments can be exploited in the solution of the global problem and avoid the solution of the problem from scratch at every time step. This is in contrast with available adaptive finite element methods; as well as the focus of Chapter 4.

**Iterative Improvement of Global-Local Enrichment Functions** A key feature of the methodology described above is the use of available information at a simulation step \( t^n \) to build the solution space for the next time step, i.e., the GFEM space \( S_{G}^{GFEM,n+1}(\Omega) \) containing the GFEM solution \( u_G^{n+1} \). The coarse-scale solution at time step \( t^n \), \( u_G^n \), is used as boundary condition on \( \partial\Omega_L \setminus (\partial\Omega_L \cap \partial\Omega_G) \) for the fine-scale problem (3.67) instead of the unknown exact solution at time \( t^{n+1} \). As a result, the error of \( u_L^n \) depends not only on the discretization used in the local domain.
\( \Omega_L \), but, also on how much the solution of the problem changes at \( \partial \Omega_L \setminus (\partial \Omega_L \cap \partial \Omega_G) \) between time steps. The effect of the inexact boundary conditions on the accuracy of \( u^n_L \) can be addressed by repeating the above procedure at each time step:

(i) Use the solution of the global problem \( u_{G}^{n+1} \in S^{GFEM,n+1}_{G}(\Omega) \) as boundary conditions for the fine-scale problem (3.67) at time \( t^n \);

(ii) update global shape functions (3.68) and global solution space \( S^{GFEM,n+1}_{G}(\Omega) \).

(iii) solve the coarse scale problem (3.66) for \( u_{G}^{n+1} \in S^{GFEM,n+1}_{G}(\Omega) \).

(iv) Go to step (i) if the accuracy of \( u_{G}^{n+1} \) is not acceptable; proceed to the next time step otherwise.

In Section 3.5.2, the effect of time-step size on the accuracy of the \( GFEM^{gl} \) is investigated.

The performance of the \( GFEM^{gl} \) when solving transient heat transfer problems with solutions exhibiting highly localized sharp thermal gradients is investigated in the next section. As a note, a quasi-static solution is obtained at time \( t^0 \) to enforce the initial conditions (3.55). This is simply the solution of Poisson’s equation since no time-dependency is required. Again, more details of the formulation for steady-state \( GFEM^{gl} \) analysis can be found in Chapter 2.

3.5 Numerical Experiments Using \( GFEM^{gl} \)

3.5.1 Numerical Experiment 1: Model Problem Utilizing \( GFEM^{gl} \) in 3D

In this section the transient \( GFEM^{gl} \) is applied to the model problem as posed in Section 3.3. The same mesh is used as was used for the simulations with exponential enrichment functions, i.e. elements with a width of 6 mm in the \( x \)-direction. With this approach it can be assured that the only difference between the two analyses is the actual shape function itself, whether it be analytic or numerically generated. The local domain, in this instance, is selected to be the entire domain, a very poor choice in the general case, but it ensures the use of exact boundary conditions in the local
domain, free of any potential numerical pollution. The goal of this example is to verify Transient Formulation 2 with the $GFEM^3$. The local domains are subjected to $h$-extensions, in which high levels of refinement are used only locally in the region of the spike, resulting in highly graded local meshes with a uniform, orthotropic polynomial order of $(p_x, p_y, p_z) = (4, 1, 1)$.

It is again noted that no transient effects are considered in the local problem, and as the results indicate no transient effects need be considered to generate accurate results. Due to the nature of this particular problem, i.e., manufacturing the internal source function from a known solution, an internal source in the local domain as would be derived from the steady-state heat conduction, or Poisson’s equation is applied as

$$Q(x) = -\kappa \frac{\partial^2 u(x)}{\partial x^2} \text{ at time } t^{n+1} \tag{3.69}$$

The heat source in (3.69) yields the desired behavior in the local domain since the source reflects that lack of transient considerations. In the more general case, in which the loading is a prescribed surface flux, the same loading is applied to the global and local domains, without any type of modification. Figure 3.13 shows the convergence of the internal energy versus time curve for $\rho c = 3.9e-5$. Good results are obtained in this case, again because with very small values of $\rho c$ the transient, and therefore time-dependency effects are at a minimum. As such, the discretization formulation used is also not important because similar results are obtained using either.

Figure 3.14 shows global internal energy, $U(t)$, versus time curves. The volumetric heat capacity, $\rho c$, is taken as $3.9e-5$. Transient Formulations 1 and 2 provide nearly identical results in this case and only curves computed with Formulation 1 are shown. The effect of adding global-local enrichment functions is significant. The figure shows that the error in $U(t)$ can be controlled through mesh refinement in the local problem, thus avoiding refinement of the global mesh even when no a-priori knowledge about the exact solution is used.

As was the case in Section 3.3.2, the value of $\rho c$ is increased and its impact on the behavior of the solution is analyzed. As is shown in Figure 3.15, it is again seen that as the value of $\rho c$ is
Figure 3.13: Convergence of $L_2^{error}(U(t))$ with respect to the number of local dofs.

Figure 3.14: Plots of internal energy versus time for $GFEM^{el}$ solutions with increasing levels of mesh refinement in local problem. The volumetric heat capacity is taken as $\rho c = 3.9e-5$. All curves computed with Transient Formulation 1.
increased, the quality of results provided by Transient Formulation 1 deteriorates because of the
time-dependency of the shape functions used in the \( GFEM^{gl} \). Results obtained from \( GFEM^{gl} \) sim-
ulations and Transient Formulation 2 are also shown. From the curve corresponding to Formulation
2, it can be seen that the time-dependency of the shape functions are once again properly accounted
for, yielding much more accurate results. The relative error in internal energy for Formulations 1
and 2 are \( L_2^{Error}(U(t)) = 0.4266 \) and \( L_2^{Error}(U(t)) = 0.0271 \), respectively, when \( \rho_c = 10. \)

\[ \text{Figure 3.15: Plot of internal energy versus time for } GFEM^{gl} \text{ solutions in 3-D for } \rho_c = 3.9e^{-5}, \ L_2 \text{ Error } = 0.0193 \ \text{and } \rho_c = 10, \ L_2 \text{ Error } = 0.4266 \ (\text{Formulation 1}) \ \text{and } \rho_c = 10, \ L_2 \text{ Error } = 0.0272 \ (\text{Formulation 2}). \]

\[ \text{Exact} \]

3.5.2 Numerical Experiment 2: Beam Subjected to Stationary Laser Flux

In this section, the transient \( GFEM^{gl} \) is applied to a beam subjected to a normal, surface flux. The
methodology is first applied to an Aluminum (Al) beam, and then to a Silicon Carbide (SiC) beam,
both subjected to a Gaussian laser flux (3.70), the shape of which is shown in Figure 3.16.

The applied Gaussian laser flux is modeled as in [18], taking the form:

\[ \bar{f}(x,t) = I_0 * f(t) * \frac{1}{2\pi a^2} * G(x, b, a) \]  

(3.70)
Figure 3.16: Spatial and temporal variation of the Gaussian laser ($\gamma = 10$).

$$f(t) = 1 - \exp(-\gamma t)$$

(3.71)

$$G(x,b,a) = \exp\left(-\frac{(x-b)^2}{2a^2}\right)$$

(3.72)

In the above equations, the constants take the values: $I_0 = 295 \frac{\text{ft-lb}}{\text{s}}$, $a = 0.025 \text{in}$, $\gamma = 10.0 \text{s}^{-1}$, $b = 9.3 \text{in}$.

From this analysis one can determine the effect of the value of $\rho c$ for simulations in which the beam has material parameters similar to those of an actual engineering material. Table 3.5 shows the values used for the material parameters in the numerical simulations.

<table>
<thead>
<tr>
<th>Material</th>
<th>$\kappa$ ($\frac{\text{ft-lb}}{\text{s-in}^3\cdot\text{F}}$)</th>
<th>$\rho c$ ($\frac{\text{ft-lb}}{\text{in}^3\cdot\text{F}}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Al</td>
<td>2.92</td>
<td>18.3</td>
</tr>
<tr>
<td>SiC</td>
<td>1.32</td>
<td>15.6</td>
</tr>
</tbody>
</table>

The beam itself is of dimension $12 \times 0.5 \times 0.24$ inches, in the $x$-, $y$- and $z$-direction, respectively, and the global mesh is shown in Figure 3.17. Flux boundary conditions given by (3.70) are applied to a portion of the top surface of the beam at $8.0 \leq x \leq 10.0$, $0 \leq y \leq 0.5$, $z = 0.24$, as noted in
the figure. The rest of the boundary is subjected to convection boundary conditions (3.2), with 
\[ \eta = 11 \frac{ft-lb}{in^2\cdot\circ F} \] and \( \bar{u} = 70 \circ F \). In this example a situation is simulated in which the applied loading

evolves to a steady-state localized, sharp, surface flux as time progresses.

Figure 3.17: Global mesh used for the beam model. Flux boundary conditions are denoted by red arrows.

**Steady-State Convergence Analysis** The steady-state version of the problem described above is analyzed first. In this case the maximum value of the laser flux (i.e., the value of \( \bar{f}(x,t) \) when \( t \to \infty \)) is applied, and Poisson’s equation for steady-state heat conduction is solved.

A reference internal energy value is obtained using a sequence of six \( hp-GFEM \) discretizations. The \( hp-GFEM \) is based on polynomial enrichments only and mesh refinement. Let the relative difference in internal energy between two successive solutions, say \( u_{hp}^{m-1} \) and \( u_{hp}^m \), be given by

\[ U_{Diff,m} := \frac{|U_{hp}^m - U_{hp}^{m-1}|}{|U_{hp}^m|} \]

where \( U_{hp}^{m-1} \) and \( U_{hp}^m \) are the internal energy of the \( hp-GFEM \) solutions \( u_{hp}^{m-1} \) and \( u_{hp}^m \), respectively.

Figure 3.18 shows the relative difference in internal energy, \( U_{Diff,m} \), versus problem size of discretization \( m \). The last discretization in the sequence, \( u_{hp}^{m=5} \), has 821,412 dofs, a uniform polynomial order of \( p = 3 \), and \( U_{Diff,m=5} = O(10^{-6}) \). Based on these results, the reference internal energy is taken as \( U_{ref} = U_{hp}^{m=5} = 2.8575e6 \). The same data is plotted in Figure 3.19 on a log-log scale so the actual values of the relative difference between any two successive analyzes can be clearly seen.

Figure 3.20 shows the relative error in the energy norm for \( hp-GFEM \) and \( GFEM^{gl} \) solutions. In the case of the \( GFEM^{gl} \) the horizontal axis shows the element size in the local domain. The global
Figure 3.18: Relative difference in energy between any two successive $hp$-GFEM analyzes. Analysis performed to determine reference value for steady-state internal energy.

Figure 3.19: Relative difference in energy between any two successive $hp$-GFEM solutions, $U^{\text{Diff},m}$, versus problem size of discretization $m$. Analysis performed to determine reference value for steady-state internal energy.
mesh is the one shown in Figure 3.17. As a result, the $GFEM^{gl}$ discretization has 1,020 dofs, regardless of the local problem size. In contrast, the number of dofs in the $hp$-$GFEM$ is in the range $[1,000 - 190,000]$. For this problem, only one global-local iteration is required, as the boundary conditions in the local domain are sufficiently accurate, and a second iteration did not appreciably improve the error level achieved. As was the case with the $hp$-$GFEM$, a uniform polynomial order of $p = 3$ is used in the $GFEM^{gl}$ analysis. As can be seen from the plot, the $hp$-$GFEM$ achieves an asymptotic convergence rate of $\beta = 3.07$, as compared to the optimum convergence rate of $\beta_{opt} = 3.0$. The $GFEM^{gl}$ achieves a slightly lower, yet comparable convergence rate of $\beta = 2.61$. In both cases, a sufficiently refined mesh must be used in order to approach optimal convergence rates. This is due to the roughness of the solution. The figure also shows that not only the convergence rate but also the error in the energy norm of the $GFEM^{gl}$ is comparable to the $hp$-$GFEM$ when the same element size and polynomial order are used in the $hp$-$GFEM$ and in the local problem for the $GFEM^{gl}$.

![Figure 3.20: Convergence of the relative error in the energy norm. Convergence rates of $\beta = 3.07$ and $\beta = 2.61$ are obtained for $hp$-$GFEM$ and $GFEM^{gl}$ analyzes, respectively.](image)

**Transient Analysis: Determination of Reference Solution**  
A reference solution for the transient case is obtained using the same approach as in the steady-state case. Here, the $hp$-$GFEM$
discretizations use high levels of local refinement and a non-uniform, non-isotropic $p$-enrichment strategy in which the entire global domain has a $p$-order of $(p_x, p_y, p_z) = (3, 3, 3)$ with a local region around the laser flux with $(p_x, p_y, p_z) = (4, 3, 4)$. The relative difference in the $L_2$ norm of the internal energy between two successive solutions, say $u^m_{hp} - 1(t)$ and $u^m_{hp}(t)$, is computed using

$$L^{\text{Diff},m}_{2}(U(t)) = \frac{\|U^m_{hp}(t) - U^{m-1}_{hp}(t)\|_2}{\|U^m_{hp}(t)\|_2} \tag{3.73}$$

where $U^m_{hp} - 1(t)$ and $U^m_{hp}(t)$ are the internal energy of the $hp$-GFEM solutions $u^m_{hp} - 1(t)$ and $u^m_{hp}(t)$, respectively, and the discrete $L_2$ norm is defined in (3.59).

Figure 3.21 shows the relative difference in the $L_2$ norm of the internal energy, $L^{\text{Diff},m}_{2}(U(t))$, versus problem size of discretization $m$. The last discretization in the sequence, $u^m_{hp} = 7$, has 359,003 dofs and $L^{\text{Diff},m=7}_{2}(U(t)) = O(10^{-7})$. Based on these results, the reference internal energy is taken as $U_{\text{ref}}(t) = U^m_{hp} = 7(t)$. The plot in Figure 3.22 shows the same data, but on a log-log scale in order to show the actual values of the $L^{\text{Diff},m}_{2}$.

![Figure 3.21: Convergence to reference internal energy versus time curve for $hp$-GFEM simulations.](image)

With a reference curve defined, the Al beam is analyzed first using Formulation 1 and the effect of the time-dependency of the shape functions on the convergence of the solution can be analyzed.
We also investigate how the convergence behavior is improved if the value of $\rho_c$ is decreased, as would be expected from the previous analyses.

Results obtained using the $GFEM^{gl}$ along with Formulation 1 are provided in Figure 3.24. The $GFEM^{gl}$ solutions are generated using meshes with only 1,020 dofs and solution accuracy is improved through the use of $h$-extensions in the local domain. Figure 3.23 shows a picture of the local mesh used, corresponding to nine levels of local refinement. The local domains in this section are generated as described in Chapter 2, with seed nodes selected from a bounding box from $\text{min} = [8.2, 0.0, 0.0]$ to $\text{max} = [10.6, 0.5, 0.24]$, and localized refinement performed in a bounding box from $\text{min} = [8.7, 0.0, 0.0]$ to $\text{max} = [9.7, 0.5, 0.24]$.

As can be seen from Figure 3.24, the $GFEM^{gl}$ simulations are converging to a particular internal energy versus time curve which is significantly different than the reference curve generated using $hp$-$GFEM$, and no time-dependency in the shape functions. From the figure it can also be seen that the steady-state portions of the curves generated by both the $GFEM^{gl}$ as well as the $hp$-$GFEM$ are very similar, but the transient portions are significantly different.

The same model is again analyzed, but with different values of $\rho_c$, using 0.1 and 0.01 of the value used for $Al$. Figure 3.25 shows how the $L_2^{\text{error}}(U(t))$ value for the internal energy versus time curve
Figure 3.23: Local mesh used for $GFEM^g_l$ simulations, nine levels of local refinement are used.

Figure 3.24: Convergence of the internal energy versus time curves for $GFEM^g_l$ solutions using Formulation 1.
is effected by the value of $\rho c$.

![Graph](image)

Figure 3.25: Illustrates how the $L_2^{error}(U(t))$ value is effected by $\rho c$.

As would be expected, for larger values of $\rho c$, the transient effects become more pronounced. As such, the change in the capacity matrix from one time-step to the next, due to the time-dependency of the shape functions becomes more important. As a result, without the modifications to the right-hand-side as in (3.64), we see a deterioration in the convergence behavior for the internal energy versus time curves as $\rho c$ increases.

While the convergence of $L_2^{error}(U(t))$ is shown to deteriorate with an increase in $\rho c$ as previously mentioned, we can still find highly accurate results when looking at the steady-state value which the transient $GFEM_{gl}$ simulation is evolving to. The reference values are obtained by running a steady-state analysis; in other words applying the maximum laser flux value to the same domain and solving Poisson’s equation using a highly refined mesh and the $hp$-$GFEM$. In the case of the steady-state evaluation, there are two relative error parameters of interest. The first is a global parameter: the relative error in the internal energy value obtained from the steady-state portion of the transient $GFEM_{gl}$ curve with respect to the internal energy value obtained from the reference solution of Poisson’s equation (i.e. steady-state simulation). The second parameter of interest is the $L_2^{error_{temp}}(u(x))$ value for the curve of the through-the-thickness temperature distribution in
the beam, again comparing the results from the steady-state portion of the $GFEM^{gl}$ curve with the results obtained from the solution of Poisson’s equation. The value of $L_2^{error, temp}(u(x))$ is computed as

$$L_2^{error, temp}(u(x)) = \frac{\|u(x,y,z^n) - u_{hp}(x,y,z^n)\|_2}{\|u(x,y,z^n)\|_2}$$

where the $n$ temperature values are taken at $(x,y,z) = (9.3, 0.25, z)$, with $z^n = [0.0, 0.2, ..., 0.24]$, and $\| . \|_2$ is the discrete $L_2$ norm defined in (3.59).

Table 3.6 summarizes the error values for the two parameters of interest for each of the three values of $\rho_c$ investigated. As can be seen, very high levels of accuracy is obtained if only the final portion of the transient $GFEM^{gl}$ curves is considered.

Table 3.6: Output for Al Beam.

<table>
<thead>
<tr>
<th>$\rho_c$</th>
<th>$L_2^{error}(U(t))$</th>
<th>$L_2^{error, temp}(u(x))$</th>
<th>$U_{Steady-State}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>18.3</td>
<td>0.0886</td>
<td>9.26e-4</td>
<td>0.0018</td>
</tr>
<tr>
<td>1.83</td>
<td>0.0114</td>
<td>0.0017</td>
<td>8.9e-4</td>
</tr>
<tr>
<td>0.183</td>
<td>0.0012</td>
<td>0.0012</td>
<td>8.9e-4</td>
</tr>
</tbody>
</table>

Figure 3.26 shows a plot of the through-the-thickness temperature generated at the final time-step of a $GFEM^{gl}$ simulation as well as the reference curve generated from the steady-state $hp-GFEM$ simulation. As can be seen, the two curves are in very good agreement.

The main conclusion to be made from Table 3.6 and Figure 3.26 is that the values generated along the steady-state portion of the transient $GFEM^{gl}$ simulation match up very well with the reference values, despite the poor ability to $GFEM^{gl}$ to simulate the transient portion of the curve with Formulation 1.

For applications in which the steady-state values are more important than the actual time-evolution, the $GFEM^{gl}$ provides very reliable results using Formulation 1. Due to the nature of the heat equation, the steady-state values correspond to the maximum loading values as well as maximum temperature gradient through the thickness, due to the fact that if the load is removed, the heat
will dissipate, and thus lower the local thermal loading. As such, the main interests in terms of design are the through-the-thickness temperature distribution, as well as the maximum temperature reached. As noted, both values are obtained once steady-state is reached, and the $GFEM^{gl}$ can provide accurate results for both of these values. In general, we don’t want to require that the system has reached steady-state in order to obtain reliable results. With this in mind, the time-dependency of the shape functions must be accounted for, and the results at each time-step, regardless of whether steady-state is reached or not, need to be reliable for the simulations to be of practical use for design considerations. As such, the $Al$ beam is now analyzed using Formulation 2. All of the results presented from this point forward are generated using Transient Formulation 2 since the shape functions of the $GFEM^{gl}$ are time-dependent, and Formulation 2 is appropriate to account for this time-dependency.

The $GFEM^{gl}$ solutions are once again generated using meshes with only 1,020 $dofs$ and solution accuracy is improved through the use of $h$-extensions in the local domain. Results obtained for the $Al$ beam generated with the $GFEM^{gl}$ are plotted in Figure 3.27. This figure shows that the internal energy versus time curves converge to the proper reference curve as the global-local enrichments are improved through mesh refinement in the local domain. From the figure, it is also apparent that
the time-dependency of the enrichment functions is properly accounted for.

![Figure 3.27: Internal energy versus time curves for Al. Solutions computed with GFEM$^\text{gl}$ on a fixed global mesh and $h$-extensions in the local domain.](image)

Figure 3.27 shows the convergence in the $L_2^{error}$ norm of the internal energy for the GFEM$^\text{gl}$ and $hp$-GFEM. The error norm $L_2^{error}$ is computed using (3.60) and the reference internal energy $U_{ref}(t)$.

It should be noted that in the case of the $hp$-GFEM, the shape functions are not time-dependent. The measure of computational effort is taken as the “element size”, referring to the width in the $x$-direction, $h_x$, of the elements in the region of high refinement. For the $hp$-GFEM “element size” of course refers to elements in the global domain, while for the GFEM$^\text{gl}$ “element size” refers to the width of the elements used in the local problems, because the width of the elements in the global domain remains constant, $h_x = 0.5\text{in}$. As can be seen from the plot, at a given element size, the $hp$-GFEM produces solutions with slightly better accuracy, but it does so at the increase in the number of dofs used in the simulation. The number of dofs in the $hp$-GFEM is in the range $[10,000 – 30,000]$ while the GFEM$^\text{gl}$ discretization has 1,020 dofs. The convergence rates, $\beta$, and error levels are very comparable for both methods. This behavior is very similar to the steady-state case shown in Figure 3.20. The $p$-enrichment strategy for the $hp$-GFEM is $(p_x, p_y, p_z) = (3, 3, 3)$ globally, with a local region around $x = b$ where $(p_x, p_y, p_z) = (4, 3, 3)$. For
the $GFEM^gl$ simulations $(p_x, p_y, p_z) = (3, 3, 3)$ in the global domain, and $(p_x, p_y, p_z) = (4, 2, 2)$ in the local domain.

![Graph showing convergence](image)

Figure 3.28: Convergence in $L_2^\text{error}(U(t))$ for internal energy versus time curves generated with $hp$-$GFEM$ and $GFEM^gl$.

Figures 3.29 and 3.30 show the convergence in the relative error in the energy norm (3.75) for a single point along the curve $U_{hp}(t)$, at times $t = 0.5 \, sec$ and $t = 1.0 \, sec$, respectively. Again, we see that in the lower error ranges, the $hp$-$GFEM$ delivers slightly better accuracy at a given element size. At time $t = 0.5 \, sec$ we see that both the $hp$-$GFEM$ and $GFEM^gl$ deliver the same convergence rates, $\beta$, and at time $t = 1.0 \, sec$ the $hp$-$GFEM$ has a higher convergence rate, but the rate is still comparable with that obtained with the $GFEM^gl$. This behavior is, again, very similar to the steady-state case shown in Figure 3.20.

\[
\frac{\|u_{hp}(t) - u_{\text{ref}}(t)\|_E}{\|u_{\text{ref}}(t)\|_E} = \left( \frac{U_{hp}(t) - U_{\text{ref}}(t)}{U_{\text{ref}}(t)} \right)^{1/2}
\]  

(3.75)

**Analysis of Beam with Material Heterogeneity** The transient $GFEM^gl$ in now applied using Transient Formulation 2 to simulations involving material heterogeneity. Several different beam materials are investigated for the same beam model. In the first case, the beam is assumed to be
Figure 3.29: Convergence in energy norm for $hp$-GFEM and $GFEM^{gl}$ solutions at $t = 0.5\,sec$.

Figure 3.30: Convergence in energy norm for $hp$-GFEM and $GFEM^{gl}$ solutions at $t = 1.0\,sec$. 
made entirely of Al. In the second case, the beam is assumed to be made entirely of SiC. In the third case, the beam is assumed to be a composite made up of Al and SiC, with the volume fraction of SiC, $V_{SiC}$, taken to be constant through the thickness of the beam, with $V_{SiC} = 0.5$. In this instance there is no through-the-thickness gradation of the material properties. Cases 4 and 5 assume a variation of $V_{SiC}$ through the thickness of the beam, according to the following power law

$$V_{SiC} = V_{SiC}^{bottom} + \left( V_{SiC}^{top} - V_{SiC}^{bottom} \right) \left( \frac{y}{h} \right)^q$$

(3.76)

where $V_{SiC}^{bottom}$ and $V_{SiC}^{top}$ are the volume fraction of silicon carbide at the top and bottom faces of the beam, respectively, taken as $V_{SiC}^{top} = 1.0, V_{SiC}^{bottom} = 0.0$; $y$ is the $y$-coordinate of the material point, and $h$ is the height of the beam, $h = 0.24\ \text{in}$. Simulations are run using $q = 1, 3$, corresponding to through-the-thickness variations of $V_{SiC}$ shown in Figure 3.31. A summary of the material composition of each case analyzed is provided in Table 3.7.

![Figure 3.31: Illustrates the through-the-thickness variation of $V_{SiC}$ for the values of $q$ used in the simulations.](image)

For the cases in which the beam is assumed to be a Functionally Graded Material (FGM) (cases 4 and 5), the values of the effective material properties, $(\rho c)_{eff}$ and $\kappa_{eff}$ are homogenized using the
Table 3.7: Material Composition for Each Case.

<table>
<thead>
<tr>
<th>Case</th>
<th>Material</th>
<th>$V_{SiC}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Al</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>SiC</td>
<td>1.0</td>
</tr>
<tr>
<td>3</td>
<td>Al-SiC</td>
<td>0.5</td>
</tr>
<tr>
<td>4</td>
<td>Al-SiC</td>
<td>Power Law (q = 1)</td>
</tr>
<tr>
<td>5</td>
<td>Al-SiC</td>
<td>Power Law (q = 3)</td>
</tr>
</tbody>
</table>

rule-of-mixtures (3.77) and the Mori-Tanaka method (3.78), respectively [18, 57].

$$
(rho_{c})_{eff} = V_{SiC} * (rho_{c})_{SiC} + (1 - V_{SiC}) * (rho_{c})_{Al}
$$

(3.77)

$$
K_{eff} = K_{SiC} + \frac{(K_{Al} - K_{SiC}) * (1 - V_{SiC})}{1 + V_{SiC} \frac{K_{Al} - K_{SiC}}{3K_{SiC}}}
$$

(3.78)

In all cases, reference solutions are computed using the same procedure described earlier for the Al beam. Results obtained using the GFEM$^{bl}$ for Cases 1 and 2 (Al and SiC), the two base materials selected for analysis are plotted in Figure 3.32. The figure shows the convergence in the $L^2_{error}$ norm of the internal energy for the GFEM$^{bl}$. As can be seen from the figure, good convergence behavior is obtained for Cases 1 and 2.

Figure 3.33 shows the internal energy versus time curves for the first two cases. As we would expect, the material with a smaller thermal conductivity has steeper gradients in the solution, and thus more internal energy. In the figure, the GFEM$^{bl}$ curves are not visible in the plot because they fall on top of hp-GFEM reference curves.

Results for the through-the-thickness temperature distributions are provided in Figure 3.34. The figure shows the convergence in the $L^2_{error,temp}(u(x))$ value for the through-the-thickness temperature distributions at $t_{final}$ for Cases 1 and 2, computed as in (3.74). From the figure, it is apparent that good convergence behavior is obtained for this error parameter.

Figure 3.35 shows the actual through-the-thickness temperature variations at $t_{final}$ for each of the 5 cases. For each of the plots provided, solid lines indicate reference curves generated with the hp-
Figure 3.32: Convergence in $L_2^2(U(t))$ norm of the internal energy for the $GFEM^{gl}$ as applied to Case 1 (Al) and Case 2 (SiC).

Figure 3.33: Internal energy versus time curves for Case 1 (Al) and Case 2 (SiC) computed with $GFEM^{gl}$ and $hp$-$GFEM$. Reference curves generated using $hp$-$GFEM$ are shown using solid lines while the $GFEM^{gl}$ data are shown using glyphs.
GFEM, and glyphs indicate $GFEM^{gl}$ data. The $GFEM^{gl}$ data fall on top of the reference $hp$-$GFEM$ curves.

Figure 3.34: Convergence in the $L_2$ norm of through-the-thickness temperature distributions for the $GFEM^{gl}$ as applied to Case 1 ($Al$) and Case 2 ($SiC$).

Figure 3.35: Through-the-thickness temperature distributions at $t_{final}$ for each of the 5 Cases. For each of the plots, solid lines indicate reference curves generated with the $hp$-$GFEM$, and glyphs indicate $GFEM^{gl}$ data.

Table 3.8 summarizes the $L^2_{error}$ norm of the internal energy, $U_{hp}(t)$, as well as the through-the-thickness temperature distributions at $t_{final}$ for each of the 5 Cases. As can be seen from the
data, $GFEM^g$ results are in good agreement with the reference $hp$-$GFEM$ results for each of the five cases in terms of internal energy versus time as well as for through-the-thickness temperature curves at $t_{final}$.

Table 3.8: $L_2^{error}$ Values for Each Trial Investigated.

<table>
<thead>
<tr>
<th>Case</th>
<th>Material Type</th>
<th>$L_2^{error}$ (Temp)</th>
<th>$L_2^{error}$ (Internal Energy)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Al</td>
<td>1.15e-3</td>
<td>4.18e-4</td>
</tr>
<tr>
<td>2</td>
<td>SiC</td>
<td>1.34e-3</td>
<td>7.31e-4</td>
</tr>
<tr>
<td>3</td>
<td>$V_{SiC} = 0.5$</td>
<td>1.26e-3</td>
<td>5.07e-4</td>
</tr>
<tr>
<td>4</td>
<td>$q = 1$</td>
<td>1.76e-3</td>
<td>1.12e-2</td>
</tr>
<tr>
<td>5</td>
<td>$q = 3$</td>
<td>2.42e-3</td>
<td>1.53e-2</td>
</tr>
</tbody>
</table>

Effect of Time-Step Size

In this section the beam model is analyzed with material properties corresponding to Al, or Case 1, and varying time step sizes, $\Delta t$. For the $GFEM^g$ simulations, there is a lag in the boundary conditions applied to the local domain due to the fact that in the generation of the enrichment function used at the global domain at time $t^{n+1}$, the Dirichlet BCs are taken from the global solution from time $t^n$, whereas the Neumann BCs and heat source are taken from the prescribed data at time $t^{n+1}$. As such it is reasonable to investigate the accuracy of the $GFEM^g$ results relative to those generated with $hp$-$GFEM$ for different time-step sizes. Figure 3.36 shows the value of $L_2^{Diff}$ for the $GFEM^g$ and $hp$-$GFEM$ simulations.

In this instance, we define the value of $L_2^{Diff}$ as

$$L_2^{Diff} = \frac{\|U_{hp}(t) - U_{gl}(t)\|_2}{\|U_{hp}(t)\|_2}$$

where $U_{hp}(t)$ and $U_{gl}(t)$ are the internal energy computed with the $hp$-$GFEM$ and $GFEM^g$, respectively, and $\|\cdot\|_2$ is the discrete $L_2$ norm defined in (3.59). Both the $hp$-$GFEM$ and $GFEM^g$ curves are generated on meshes with 7 levels of refinement around the laser flux location. From the plot it can be seen that the $L_2^{Diff}$ value is, at least for this problem, relatively insensitive to
the time step size used, and the lag in the boundary conditions applied to the local problem from the global solution at \( t^n \) do not cause a problem in terms of the quality of the local solution, even when a larger \( \Delta t \) is used. In this instance, the insensitivity is likely due to the fact that the solution outside of the local region remains relatively constant, and as such the boundary conditions for the local domain are not changing significantly between subsequent time-steps. The end result is that the quality of the local solution is not significantly impacted by the lag due to \( \Delta t \). It is quite feasible that other problems may be impacted more severely by the lag in the boundary conditions. In this case, a second global-local iteration, as was used in the steady-state analysis, may be used to update the boundary conditions and yield more accurate results.

**Effect of Time-Rise Constant, \( \gamma \)**

The effect of the time-rise constant, \( \gamma \) on the ability of the transient \( GFEM^{gl} \) to deliver accurate results is now investigated. The material properties used in this section correspond to Al, referred to previously as Case 1. Figure 3.37 shows the spatial and temporal variations for the Gaussian flux when \( \gamma = 50 \).

In this instance, the intensity of the flux ramps up much more quickly in time as compared to Figure 3.16, resulting in a larger gradient in the internal energy versus time reference curve for the model.
problem, as shown in Figure 3.38. In the figure, a reference curve generated using \textit{hp-GFEM} is shown using a solid line while the \textit{GFEM}^{gl} data are shown using glyphs. The \textit{GFEM}^{gl} data for $\gamma = 50$ once again falls on top of the reference curve. The curves corresponding to $\gamma = 10$ are also provided to illustrate the difference in the evolution of the internal energy with respect to an increase in $\gamma$. In both instances, $\gamma = 10$ and $\gamma = 50$, 9 levels of localized refinement are used.

![Figure 3.37: Spatial and temporal variation of the Gaussian laser flux, ($\gamma = 50$).](image1)

![Figure 3.38: Internal energy vs time curves for solutions obtained with time-rise constant $\gamma = 10$ and $\gamma = 50$. Reference curves generated using \textit{hp-GFEM} are shown using solid lines while the \textit{GFEM}^{gl} data are shown using glyphs.](image2)
The convergence in the $L^2_{error}$ norm of the internal energy is shown in Figure 3.39 for $\gamma = 10$ and $\gamma = 50$. From the plots it can be seen that in both cases good convergence behavior is obtained, with differences only at very low error values. In both cases the $L^2_{error}$ value is driven below one percent relative difference.

![Graph showing convergence in the $L^2_{error}$ norm of the internal energy for $\gamma = 10$ and $\gamma = 50$](image)

Figure 3.39: Convergence in the $L^2_{error}$ norm of the internal energy for $GFEM^{gl}$ as a function of element size in local problem.

The convergence in the $L^2_{error}$ norm of the through-the-thickness temperature distribution is shown in Figure 3.40 for $\gamma = 10$ and $\gamma = 50$. The temperatures are taken from the same location as previously noted. From the plots it is again seen that good convergence behavior is obtained, with differences only at very low error values. In both cases the $L^2_{error}$ value is once again driven below one percent.

The evolution of the through-the-thickness temperature distributions in time are shown in Figures 3.41 and 3.42 for $\gamma = 10, 50$, respectively. From the figures it can be seen that for $\gamma = 50$ the temperature distributions at $t = 0.3, 0.65$ seconds are closer to the steady-state curves than those obtained with $\gamma = 10$, as would be expected.
Figure 3.40: Convergence in the $L^2_{\text{error}}$ norm of through-the-thickness temperature distribution at $t_{\text{final}}$ as a function of element size in local problem.

Figure 3.41: Evolution of the through-the-thickness temperature distribution for the beam with $\gamma = 10$. 
3.5.3 Numerical Experiment 3: Beam Subjected to Moving Laser Flux

In this section the same beam model as in the previous section is analyzed, with material properties corresponding to Al. In this example, the applied laser flux increases in time, as well as moves in space, as shown in Figure 3.43.

The applied Gaussian laser flux function now takes the form:
\[
\tilde{f}(x,t) = I_0 f(t) * \frac{1}{2\pi a^2} * G(x,b(t),a)
\]

(3.80)

\[
f(t) = 1 - \exp(-\gamma t)
\]

(3.81)

\[
G(x,b(t),a) = \exp\left(\frac{-(x - b(t))^2}{2a^2}\right)
\]

(3.82)

\[
b(t) = b_0 + V t
\]

(3.83)

In the above equations, the constants take the values: \(I_0 = 295 \text{ ft-lb/s}, \ a = 0.025 \text{ in}, \ \gamma = 10.0 \text{ s}^{-1}, \ b_0 = 9.25 \text{ in} \) and \(V = 0.5 \text{ in/sec}\). The reference solution in this section is generated using \(hp\)-GFEM with high levels of local refinement and cubic shape functions, resulting in a model with 433,635 dofs. The \(GFEM^g\) solutions are once again generated using meshes with only 1,020 dofs and solution accuracy is again improved through the use of \(h\)-extensions in the local problem.

Figure 3.44 shows the internal energy versus time curves for the \(GFEM^g\) simulation as well as the \(hp\)-GFEM reference curve. The \(GFEM^g\) curve is again difficult to see because it falls on top of the reference curve. Nine levels of mesh refinement are applied to the local problem used with the \(GFEM^g\). The \(L_2^{error}\) error norm of the internal energy for the \(GFEM^g\) simulation is \(6.01e-4\), indicating excellent agreement between the \(GFEM^g\) solution and the reference \(hp\)-GFEM solution.

Figures 3.45 and 3.46 show snap-shots of the solution at different times throughout the simulation for \(GFEM^g\) solutions and \(hp\)-GFEM solutions, respectively. It is of greatest interest to highlight the ability of the \(GFEM^g\) to resolve the transient, moving, thermal spike on elements which are significantly larger than the width of the spike itself, as shown in Figure 3.45. By contrast, Figure 3.46 illustrates the ability of the \(hp\)-GFEM to resolve the moving spike as well, but with the requirement of a significant increase in the mesh density. As such, there is great potential for a significant increase in the computational efficiency for transient simulations using the \(GFEM^g\).
Figure 3.44: Internal energy vs time curves for solutions obtained with a moving source.

Figure 3.45: Snap-shots in time for the transient $GFEM^{nl}$ simulation for the beam with a moving laser flux.

Figure 3.46: Snap-shots in time for the transient $hp$-$GFEM$ simulation for the beam with a moving laser flux.
3.6 Summary

In this chapter, the generalized FEM with global-local enrichments ($GFEM^{gl}$) is formulated for transient heat transfer problems with solutions exhibiting highly localized sharp thermal gradients. The proposed method enables the analysis of this class of problems using uniform, coarse, global meshes. This has several computational implications as discussed in Section 3.5. From the analyses performed in this chapter we have seen:

(i) the possibility of capturing localized, transient solution features using uniform, coarse, global meshes. This removes, for example, the need to refine global meshes that are usually complex and very large, the re-meshing of which is non-trivial between subsequent time-steps;

(ii) no transient effects need to be considered in the local domains;

(iii) with the proper discretization order (temporal, spatial) the $GFEM^{gl}$ produces results which are in very good agreement with the reference curves generated using $hp$-$GFEM$ and significantly more degrees of freedom;

(iv) the $GFEM^{gl}$ delivers accurate results in terms of the evolution of the internal energy as a function of time, as well as in the resulting through-the-thickness temperature distributions. The latter of which is important for design considerations, particularly if thermo-mechanical coupling is considered. Thermo-mechanical coupling, while not addressed in this work, is a topic to be investigated in future work;

(v) the size of the enriched global problem does not depend on the size or discretization used in the local problems;

(vi) the accuracy of the $GFEM^{gl}$ is relatively insensitive to the time-rise constant of the applied surface flux;

(vii) the $GFEM^{gl}$ uses a large amount of information which can be calculated once and re-used at each subsequent time-step, yielding the potential for significant improvement in the com-
putational efficiency. This potential increase in efficiency is the current focus upon the completion of this work;

(viii) the accuracy of the $GFEM^g_l$ is relatively insensitive to the size of $\Delta t$, and the resulting lag in Dirichlet boundary conditions applied in the local problems;

In the next chapter, to further investigate the potential benefits of $GFEM^g_l$ for transient simulations, the hierarchical nature of the enrichment functions is explored. In the present case, the vast majority of the global matrices remain constant, and do not change from time-step to time-step. As such it is possible to re-use this information by factorizing the large, global matrices only once at the beginning of a simulation, and then re-use the factorization at each subsequent time-step, yielding a much more efficient transient solution algorithm. This algorithm is formulated and applied in the next chapter.
Chapter 4

ReAnalysis for Steady-State and Transient Heat Transfer

4.1 Algorithmic Formulation for Steady-State ReAnalysis

In GFEM simulations there is a significant amount of reusable information, that if utilized, can achieve much better efficiency than has been seen in the previous chapter. In particular, the initial global stiffness matrix of the coarse-scale mesh needs only to be assembled and factorized once throughout the course of an entire simulation. The factorized matrix can then be used for subsequent forward/backward substitutions. In order to make this work properly, it is essential that the coarse-scale mesh remain fixed. The hierarchical nature of the fine-scale enrichments enables one to effectively separate out the coarse-scale and fine-scale problems. In order to implement this algorithm we assume that we can partition the enriched global matrix as follows:

- a portion that depends only on coarse-scale degrees of freedom
- a portion that depends only on the added fine-scale degrees of freedom
- a portion representing the coupling between the coarse- and fine-scale degrees of freedom

We similarly assume that we can partition the vector of degrees of freedom (load vector) into fine-scale and coarse-scale degrees of freedom (loads). With the previous assumptions in mind, the global system of equations is written as

\[ \mathbf{K}_G^{\text{partitioned}} \mathbf{u}_G^{\text{partitioned}} = \mathbf{f}_G^{\text{partitioned}} \]  \hspace{1cm} (4.1)

in which we define the following
\[ \mathbf{K}_{G}^{\text{partitioned}} = \begin{bmatrix} \mathbf{K}_{G}^{0} & \mathbf{K}_{G}^{0,gl} \\ \mathbf{K}_{G}^{gl,0} & \mathbf{K}_{G}^{gl} \end{bmatrix} \] 

\[ \tilde{\mathbf{u}}_{G}^{\text{partitioned}} = \begin{bmatrix} \tilde{\mathbf{u}}_{G}^{0} \\ \tilde{\mathbf{u}}_{G}^{gl} \end{bmatrix}^{T} \] 

\[ \mathbf{f}_{G}^{\text{partitioned}} = \begin{bmatrix} \mathbf{f}_{G}^{0} \\ \mathbf{f}_{G}^{gl} \end{bmatrix}^{T} \] 

where \((\cdot)^{0}\) indicates a quantity associated with the coarse-scale, and \((\cdot)^{gl}\) indicates a quantity associated with the fine-scale.

**Static Condensation**  With each of the matrices (vectors) defined, we can follow the static condensation scheme presented in [28]. The first equation obtained from (4.1) yields

\[ \tilde{\mathbf{u}}_{G}^{0} = (\mathbf{K}_{G}^{0})^{-1} \left[ \mathbf{f}_{G}^{0} - \mathbf{K}_{G}^{0,gl} \tilde{\mathbf{u}}_{G}^{gl} \right] \] 

Substituting this back into the second equation from (4.1), and rearranging so that the known quantities are in the right-hand-side, yields

\[ \left[ \mathbf{K}_{G}^{gl} - \mathbf{K}_{G}^{gl,0} \left( \mathbf{K}_{G}^{0} \right)^{-1} \mathbf{K}_{G}^{0,gl} \right] \tilde{\mathbf{u}}_{G}^{gl} = \mathbf{f}_{G}^{gl} - \mathbf{K}_{G}^{gl,0} \left( \mathbf{K}_{G}^{0} \right)^{-1} \mathbf{f}_{G}^{0}. \] 

For convenience (4.6) can be written as

\[ \hat{\mathbf{K}}_{G}^{gl} \tilde{\mathbf{u}}_{G}^{gl} = \hat{\mathbf{f}}_{G}^{gl} \] 

where

\[ \hat{\mathbf{K}}_{G}^{gl} = \left[ \mathbf{K}_{G}^{gl} - \mathbf{K}_{G}^{gl,0} \left( \mathbf{K}_{G}^{0} \right)^{-1} \mathbf{K}_{G}^{0,gl} \right] \]
\[ \mathbf{f}_G^{gl} = \left\{ \mathbf{f}_G^{gl} - \mathbf{K}_G^{gl,0} (\mathbf{K}_G^0)^{-1} \mathbf{f}_G^0 \right\} \] (4.9)

The solution to (4.7) is computationally inexpensive, since the dimension of the additional fine-scale dofs is very small compared to the dimension of the fixed, coarse-scale system. The calculation of both \( \hat{\mathbf{K}}_G^{gl} \) and \( \hat{\mathbf{f}}_G^{gl} \) require only the forward/backward substitution of the factorization of \( \mathbf{K}_G^0 \), which is only computed once, and then re-used for each loading condition. Once \( \bar{\mathbf{u}}_G^{gl} \) is calculated, the computation of \( \bar{\mathbf{u}}_G^0 \) also requires only the forward/backward substitution of the factorization of \( \mathbf{K}_G^0 \), as shown in (4.5).

### 4.2 Verification of Steady-State Implementation

In this section the ReAnalysis algorithm formulated for steady-state heat transfer is verified using output from the \( GFEM^{gl} \) analyses performed in Chapter 2. The plate is 500x250x30 mm, and \( GFEM^{gl} \) simulations are run on three different global meshes. Meshes 0x, 1x and 2x have global elements with widths \( h_x = 50, 25 \) and 12.5 mm, respectively. Figures 4.1, 4.2 and 4.3 show the enrichment strategies for Mesh 0x, 1x and 2x, respectively.

![Figure 4.1: Mesh 0x, local solution applied at nodes denoted by red glyphs.](image)

Curves for the relative error in the energy norm versus CPU time required to run the simulation are provided in Figures 4.4, 4.5 and 4.6 for Mesh 0x, 1x and 2x, respectively. In each of the figures, the label scratch indicates a simulation run using the standard \( GFEM^{gl} \) with no ReAnalysis, while
Figure 4.2: Mesh 1x, local solution applied at nodes denoted by red glyphs.

Figure 4.3: Mesh 2x, local solution applied at nodes denoted by red glyphs.
the label reAnalysis indicates use of the ReAnalysis algorithm. In this problem, the local domain is a relatively large portion of the global domain, and as such we would not expect much speed-up, if any at all, with the use of the ReAnalysis algorithm. In fact, in many cases the ReAnalysis actually requires more CPU time. The values of interest in this case are merely the error values, which illustrate that the results are essentially the same as those obtained with the standard GFEM\textsuperscript{gl}, as we would expect.

The second label for each curve indicates the integration rule used for the simulation. The label HO indicates the use of a high order rule in the entire domain, as is used in [82]. The label optimized indicates a more optimized integration rule, in which the high order integration rule is used locally in the region of sharp gradients, and a standard, lower order integration rule is used in the majority of the domain where the solution is smooth. With the more optimized integration rule we see that we can reduce the CPU time requirements in a fairly significant manner.

![Figure 4.4: Comparison of CPU times for various solution strategies for Mesh 0x.](image)

All of the results presented from this point forward reflect the use of optimized integration rules, as it is clear that the optimized integration rules lead to much more efficient simulations while not adversely impacting the GFEM solution.
Figure 4.5: Comparison of CPU times for various solution strategies for Mesh 1x.

Figure 4.6: Comparison of CPU times for various solution strategies for Mesh 2x.
4.3 Application of Steady-State Implementation: Large Panel

In this section the \( GFEM^{gl} \) with ReAnalysis is applied to a larger problem of more industrial interest in the form of a large panel subjected to a Gaussian flux applied at various locations on the panel surface. The panel itself is of dimension \( 48 \times 24 \times 0.24 \text{ in} \), and shown in Figure 4.7. The thermal conductivity is taken as \( \kappa = 2.92 \). The Gaussian surface flux is modeled as (4.10), with constants taking the values: \( I_0 = 295 \text{ ft-lb/s} \), \( a = 0.025 \text{ in} \) and the value of \( b \) varies, depending on the location. There is also a constant flux with the magnitude \( \text{flux}_{constant} = 15 \text{ ft-lb/s} \) applied to the entire top surface in addition to the sharp, Gaussian flux. All other boundaries are subjected to convective boundary conditions (3.2) with \( \eta = 11 \text{ ft-lb/in}^2 \cdot \circ F \) and \( \bar{u} = 70 \circ F \).

\[
\bar{f}(x) = I_0 \frac{1}{2\pi a^2} G(x, b, a) \tag{4.10}
\]

\[
G(x, b, a) = \exp\left(\frac{-(x - b)^2}{2a^2}\right) \tag{4.11}
\]

Figure 4.7: Model of panel used for analysis. The locations of the three load cases to be considered are also indicated.

The Gaussian fluxes are applied in three different potential locations of interest on the surface of the panel. The first location is along the lower edge of the panel, with flux centered at \( b = 9.3 \text{ in} \).
and a length of 3\textit{in}, i.e. the Gaussian flux is applied only for $0.0 \leq y \leq 3.0\textit{in}$. The second location is near the center of the panel with $b = 24.2\textit{in}$ and a length of 2\textit{in}; with the Gaussian flux applied only for $11.0 \leq y \leq 13.0\textit{in}$. The third location is near the top corner of the panel with $b = 32.5\textit{in}$ and a length of 1.5\textit{in}; with the Gaussian flux applied only for $22.5 \leq y \leq 24.0\textit{in}$. Output, in the form of the resulting temperature fields from $GFEM^{gl}$ analyses are provided in Figures 4.8, 4.9 and 4.10 for Gaussian flux locations 1, 2 and 3, respectively.

![Figure 4.8: Resulting temperature distribution for load Case 1.](image)

Tables 4.1 and 4.2 summarize the results of the simulations in terms of the CPU time requirements for the $hp$-GFEM and $GFEM^{gl}$ with ReAnalysis, respectively. The $GFEM^{gl}$ with ReAnalysis offers a significant speed-up in terms of CPU time required when the entire simulation is considered. This behavior is again due to the fact that the factorization from the coarse global matrix may be re-used for each load case, yielding the most significant speed-up.

Figure 4.11 is a plot of the total CPU time required to run an analysis versus the number of thermal spikes considered. The plot illustrates the previously discussed benefit that the gains in CPU time grow significantly with the number of spike locations considered. This potential can be further exploited when transient simulations are run using multiple time-steps at each location of interest,
Figure 4.9: Resulting temperature distribution for load Case 2.

Figure 4.10: Resulting temperature distribution for load Case 3.
Table 4.1: Summary of $hp$-GFEM Simulations.

<table>
<thead>
<tr>
<th>Case</th>
<th>ndofs</th>
<th>CPU Time</th>
<th>Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>140540</td>
<td>277.19</td>
<td>2.031e7</td>
</tr>
<tr>
<td>2</td>
<td>138006</td>
<td>270.11</td>
<td>1.344e7</td>
</tr>
<tr>
<td>3</td>
<td>110398</td>
<td>173.54</td>
<td>9.961e6</td>
</tr>
<tr>
<td><strong>Total CPU Time</strong></td>
<td><strong>720.84</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.2: Summary of Computational Effort for $GFEM^{gl}$ Simulations with ReAnalysis.

<table>
<thead>
<tr>
<th>Case</th>
<th>ndofs Local</th>
<th>CPU Time Local</th>
<th>ndofs Enr Glob</th>
<th>CPU Time Enr Glob</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12084</td>
<td>9.45</td>
<td>95130</td>
<td>72.81</td>
</tr>
<tr>
<td>2</td>
<td>9300</td>
<td>9.26</td>
<td>95110</td>
<td>65.02</td>
</tr>
<tr>
<td>3</td>
<td>10500</td>
<td>9.14</td>
<td>95100</td>
<td>64.85</td>
</tr>
</tbody>
</table>

ndofs Init Global | 95060
CPU Time Init Global | 120.83
**Total CPU Time** | **351.36**

which will be explored in Section 4.6.

![Figure 4.11: CPU time requirements versus number of spike locations considered.](image)

Figure 4.11: CPU time requirements versus number of spike locations considered.

While the internal energy values are not the main focus of this chapter, it is still worthwhile to compare the results obtained with both methods to ensure that they are comparable. Table 4.3 shows a comparison of the internal energy values obtained from both simulation methods. As can
be seen, there is good agreement between the internal energy values obtained in both cases, with a relative error on the order of one percent, or less.

Table 4.3: Comparison of Energy Values.

<table>
<thead>
<tr>
<th>Case</th>
<th>Energy Enriched Global</th>
<th>Energy hp-GFEM</th>
<th>Relative Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.019e7</td>
<td>2.031e7</td>
<td>0.0062</td>
</tr>
<tr>
<td>2</td>
<td>1.334e7</td>
<td>1.344e7</td>
<td>0.0074</td>
</tr>
<tr>
<td>3</td>
<td>9.907e6</td>
<td>9.961e6</td>
<td>0.0054</td>
</tr>
</tbody>
</table>

Table 4.4: Comparison of Energy Values.

| Energy Initial Global | 2.105e4 |

Table 4.4 now summarizes the CPU time requirements for solving for each of the three load cases separately, providing CPU time requirements for hp-GFEM, standard GFEM
gl analysis without ReAnalysis, and GFEM
gl with ReAnalysis. As can be seen from this table, as we would expect, the GFEM
gl without the ReAnalysis algorithm always provides the least computationally efficient simulations due the requirement of assembling and factorizing the large global matrix for the initial global solve, and then again for the enriched global solve. Small speed-ups are obtained using the GFEM
gl with ReAnalysis versus hp-GFEM when load case 1 or 2 are considered individually, whereas the hp-GFEM provides a more efficient simulation when only load case 3 is considered.

Table 4.4: Comparison of CPU Time for 3 Solution Strategies.

| Case | hp-GFEM | GFEM
gl w/ReAnalysis | GFEM
gl from Scratch |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>277.19</td>
<td>203.09</td>
<td>301.30</td>
</tr>
<tr>
<td>2</td>
<td>270.11</td>
<td>195.11</td>
<td>294.28</td>
</tr>
<tr>
<td>3</td>
<td>173.54</td>
<td>194.82</td>
<td>290.61</td>
</tr>
</tbody>
</table>

Effect of Numerical Pollution As has been discussed in detail in Chapter 2, the quality of the local solutions is significantly affected by the quality of the boundary conditions provided by the initial global solution on a coarse mesh. Due to the highly localized nature of the loading function, the coarse elements may not be able to properly resolve the localized behavior, and the errors in the solution may be significant even at locations far from the spike in the loading. This concept
is addressed in Chapter 2 for a small, model problem; and is now investigated for a much larger problem, which may be of more interest to the engineering community.

The error in the temperature is taken as the point-wise difference in the solutions generated on a highly refined (reference) mesh, $u_{\text{ref}}$, and the solution generated on the coarse, global mesh used for the $GFEM_{gl}$ analyses, $u_h$. It is important to note that no local enrichment functions are used here, merely the coarse global elements and their corresponding polynomial enrichment functions.

The temperature values are taken along the length of the panel, on the center of the top surface, i.e., $(x, y, z) = (x, 12.0, 0.24)$, for $0.0 \leq x \leq 48.0$ at 0.5 in intervals.

The point-wise error in the temperature field, $e(u) = u_{\text{ref}} - u_h$, is plotted in Figure 4.12 for load case 2 (center of panel). The loading function itself is also plotted along the length of the panel. As can be seen, even in a much larger problem of more general interest, the error in the solution is still impacted in a region which is much larger than the width of the loading function itself.

![Figure 4.12: Point-wise error in the solution, $e(u)$, for $a = 0.025$. The flux function itself is also plotted to illustrate that the solution is affected even outside of the region subjected to the sharp loading.](image)

The point-wise error in the temperature is plotted in Figure 4.13 for various values of $a$, the Gaussian radius. As would be expected, as the value of $a$ decreases and the loading becomes sharper, the error in the temperature field becomes worse, and less localized. As a result, the boundary...
conditions for the local problems may become polluted for small values of $a$, and either larger local domains must be used, or two iterations may be required, as is discussed in Section 2.5.2. An alternative approach, as investigated in Section 2.5.3, is to only apply the smooth loading in the coarse, global domain; and then apply the sharp Gaussian flux in the highly refined local domain as well as in the enriched global domain.

![Figure 4.13: Point-wise error in the solution, $e(u)$, for various values of $a$. As $a$ decreases, the flux function becomes sharper, and the error in the solution becomes significantly worse.](image)

### 4.4 Algorithmic Formulation for Transient ReAnalysis

In the $GFEM^g_l$ as applied to transient simulations, there is once again a significant amount of reusable information, that if utilized, can achieve much better efficiency than has been seen in Chapter 3. The speed-up in terms of CPU time will be significantly amplified as the number of time-steps is increased, as noted in the previous section. We start the formulation for the transient analysis by first assuming that we can partition the global matrices and vectors into portions representing the fixed coarse-scale dofs and the evolving fine-scale dofs.

\[
K_G^{\text{partitioned}} u_G^{\text{partitioned}} = f_G^{\text{partitioned}} \tag{4.12}
\]
in which we again define the following

\[
K^\text{partitioned}_G = \begin{bmatrix}
K^0_G & K^{0,gl}_G \\
K^{gl,0}_G & K^g_G
\end{bmatrix}
\]  

(4.13)

For transient simulations, taking into account inertial effects as discussed in detail in Chapter 3, we define the following at time \( t^{n+1} \):

\[
K^0_G = \left[ M^0_{CS} + \alpha \Delta t K^0_{CS} \right]
\]  

(4.14)

\[
K^{gl(n+1)}_G = \left[ M^{gl(n+1)}_{FS} + \alpha \Delta t K^{gl(n+1)}_{FS} \right]
\]  

(4.15)

\[
K^{gl,0(n+1)}_G = \left[ M^{gl,0(n+1)}_{CS-FS} + \alpha \Delta t K^{gl,0(n+1)}_{CS-FS} \right]
\]  

(4.16)

\[
K^{0,gl(n+1)}_G = \left[ M^{0,gl(n+1)}_{CS-FS} + \alpha \Delta t K^{0,gl(n+1)}_{CS-FS} \right]
\]  

(4.17)

In the previous equations, \( K^0_G \) is the effective stiffness matrix associated with the fixed, coarse-scale mesh. It is computed as a combination of the fixed, coarse-scale stiffness matrix, \( K^0_{CS} \), and the fixed, coarse-scale capacity matrix, \( M^0_{CS} \). This effective stiffness matrix may be assembled and factorized once, and this information can then be saved for use at each subsequent time-step.

\( K^{gl(n+1)}_G \) is the effective stiffness matrix for the fine-scale dofs. Similarly, it is a combination of the fine-scale stiffness matrix, \( K^{gl(n+1)}_{FS} \), and the fine-scale capacity matrix, \( M^{gl(n+1)}_{FS} \) at time \( t^{n+1} \). Since the fine-scale is changing at each time-step, this matrix must be assembled and factorized at each time step. It is, however; much smaller in dimension than \( K^0_G \), and thus less expensive to assemble and factorize. \( K^{gl,0(n+1)}_G \) and \( K^{0,gl(n+1)}_G \) represent the coupling between the dofs of the coarse and fine-scales. These will also need to be assembled and factorized at each time-
step. These two matrices are related as $K_{g}^{0,0(n+1)} = \left( K_{g}^{0,0(n+1)} \right) ^{T}$. As would be expected, they are combinations of the fine/coarse-scale coupling stiffness, $K_{cFS}^{0,gl(n+1)}$, and capacity, $M_{cFS}^{0,gl(n+1)}$, matrices at time $t^{n+1}$.

Next, we define

$$\tilde{u}_{G}^{(n+1)} = \left\{ \tilde{u}_{G}^{(0(n+1))}, \tilde{u}_{G}^{(gl(n+1))} \right\} ^{T} \quad (4.18)$$

where $\tilde{u}_{G}^{(0(n+1))}$ and $\tilde{u}_{G}^{(gl(n+1))}$ are again, degrees of freedom associated with the coarse- and fine-scales, respectively.

The right-hand-side of (4.12) is defined as

$$f_{G}^{(n+1)} = \left\{ f_{G}^{(0(n+1))}, f_{G}^{(gl(n+1))} \right\} ^{T} \quad (4.19)$$

where $f_{G}^{(0(n+1))}$ and $f_{G}^{(gl(n+1))}$ are the effective load vectors associated with the coarse- and fine-scales, respectively. To define the effective load vectors, further analysis is required to properly account for the inertial effects. The right-hand-side of (4.12) is written as

$$\begin{bmatrix} f_{G}^{(0(n+1))} \\ f_{G}^{(gl(n+1))} \end{bmatrix} = \Delta t \begin{bmatrix} \alpha f_{CS}^{(0(n+1))} + (1 - \alpha) f_{CS}^{(0(n))} \\ \alpha f_{FS}^{(gl(n+1))} + (1 - \alpha) f_{FS}^{(gl(n+1),n)} \end{bmatrix} + \begin{bmatrix} M_{CS}^{0} \\ M_{CSFS}^{0,gl(n+1),n} \end{bmatrix} \tilde{u}_{G}^{(0(n))} - \begin{bmatrix} K_{CS}^{0} \\ K_{CSFS}^{0,gl(n+1),n} \end{bmatrix} \tilde{u}_{G}^{(0(n))}$$

For simplicity we define
\[ \Psi_{CS}^0 = M_{CS}^0 - (1 - \alpha) \Delta t K_{CS}^0 \] (4.21)

\[ \Psi_{CS-FS}^{0,g(n+1,n)} = M_{CS-FS}^{0,g(n+1,n)} - (1 - \alpha) \Delta t K_{CS-FS}^{0,g(n+1,n)} \] (4.22)

\[ \Psi_{FS-CS}^{g(n+1,n)} = M_{FS-CS}^{g(n+1,n)} - (1 - \alpha) \Delta t K_{FS-CS}^{g(n+1,n)} \] (4.23)

\[ \Psi_{FS}^{g(n+1,n)} = M_{FS}^{g(n+1,n)} - (1 - \alpha) \Delta t K_{FS}^{g(n+1,n)} \] (4.24)

It should be noted, in the previous equations, \( \Psi \) indicates a quantity which takes into account the time-dependent nature of the shape functions as described in Chapter 3, and requires finite element shape functions from time steps \( t^n \) and \( t^{n+1} \). With the simplifications provided by (4.21) - (4.24), the components of the effective, partitioned load vector are written as

\[ f_{G}^{0(n+1)} = \alpha \Delta t f_{CS}^{0(n+1)} + (1 - \alpha) \Delta t f_{CS}^0 + \Psi_{CS}^0 u_{G}^0 + \Psi_{CS-FS}^{0,g(n+1,n)} u_{G}^{g(n+1)} \] (4.25)

\[ f_{G}^{g(n+1)} = \alpha \Delta t f_{FS}^{g(n+1)} + (1 - \alpha) \Delta t f_{FS}^{g(n+1)} + \Psi_{FS-CS}^{g(n+1,n)} u_{G}^0 + \Psi_{FS}^{g(n+1,n)} u_{G}^{g(n+1)} \] (4.26)

With each of the effective matrices and vectors defined, we can follow the static condensation scheme presented in Section 4.1 to efficiently solve the partitioned system of equations at any time \( t^{n+1} \).
4.5 Verification of Transient Implementation

In this section the implementation of the transient ReAnalysis algorithm is tested by analyzing the same flat beam model as is used in Chapter 3. The material parameters are taken as those of Al (κ = 2.92, ρc = 18.3), and the beam is once again subjected to a Gaussian surface flux, modeled as

\[
\bar{f}(x,t) = I_0 \ast f(t) \ast \frac{1}{2\pi a^2} \ast G(x,b,a)
\]

(4.27)

\[
f(t) = 1 - \exp(-\gamma \ast t)
\]

(4.28)

\[
G(x,b,a) = \exp\left(\frac{-(x-b)^2}{2a^2}\right)
\]

(4.29)

where the constants take the values: \(I_0 = 295 \frac{ft-lbf}{s}\), \(a = 0.025in\), \(\gamma = 10.0s^{-1}\), \(b = 9.3in\). For completeness, the mesh and associated boundary conditions are shown in Figure 4.14.

Figure 4.14: Global mesh used for the beam model. Flux boundary conditions are denoted by red arrows.

Figure 4.15 shows the convergence in the \(L_2^{error}\) value for the internal energy versus time curves generated using both the standard transient GFEM\textsuperscript{gl} as well as for the transient GFEM\textsuperscript{gl} with Re-Analysis \(\text{GFEM}_{\text{ReAn}}^\text{gl}\). As can be seen from the plot, the \(\text{GFEM}_{\text{ReAn}}^\text{gl}\) provides virtually identical error levels to those generated with the transient GFEM\textsuperscript{gl}.

Figure 4.16 shows the internal energy versus time curves for simulation data generated with both the transient GFEM\textsuperscript{gl} as well as the transient GFEM\textsuperscript{gl}_{ReAn}. It is again seen from the plot that both methods generate virtually identical results.

It should be noted that there is no mention of CPU time requirements in this section, because the
Figure 4.15: Convergence in the $L_2^{error}$ value for internal energy versus time curves generated with the transient GFEM$^g_l$ as well as the transient GFEM$^{g_l}_{ReAn}$.

Figure 4.16: Internal energy versus time curves generated with the transient GFEM$^g_l$ as well as the transient GFEM$^{g_l}_{ReAn}$. Data generated with GFEM$^g_l$ is shown with using solid lines, while data generated with GFEM$^{g_l}_{ReAn}$ is shown using glyphs.
aim is merely to illustrate that virtually identical results can be generated using either algorithm. The subsequent section provides analysis of a much larger problem, and as such, the CPU time requirements will be the main focus of the next section.

4.6 Application of Transient Implementation: Large Panel

The transient ReAnalysis algorithm is now applied to a large flat panel, taken to have dimensions $48 \times 24 \times 0.24$ in, (Figure 4.7) with material properties taken as $(\kappa = 2.92, \rho c = 18.3)$. The top surface is subjected to a normal flux defined as in (4.27), (4.28) and (4.29), with $I_0 = 295 \frac{ft-lbf}{s}$ and $a = 0.025$ in. The values of $\gamma$ and $b$ will be given in the appropriate subsection; and the remainder of the panel boundary is subjected to convective boundary conditions (3.2) with $\eta = 11 \frac{ft-lbf}{s \cdot in^2 \cdot ^\circ F}$ and $\bar{u} = 70 ^\circ F$. In addition to the Gaussian surface flux, a constant flux with magnitude $flux_{constant} = 5 \frac{ft-lbf}{s}$ is also applied to the top surface of the panel.

4.6.1 Load Case 1: Thermal Spike Along Edge of Panel

In the first loading case of interest, the loading is applied along the edge of the panel. The focus of the applied Gaussian flux is at $f_{Gauss} = (9.3, 1.5, 0.24)$, with a length of 3 in, i.e. the Gaussian flux is applied only for $0.0 \leq y \leq 3.0$ in. The value of $b$ is taken as the $x$-coordinate of the focus, in this case $b = 9.3$, and the time-rise constant is taken as $\gamma = 10.0 \text{s}^{-1}$.

Output in the form of CPU time requirements as a function of the number of time-steps is provided in Figure 4.17 for transient $hp$-GFEM and $GFEM_{ReAn}^gl$ simulations. From Figure 4.17(a) we see that there is a speed-up in terms of the CPU time requirements for the case of the $GFEM_{ReAn}^gl$ when both the factorization and forward/backward substitutions ($f/b_{sub}$) as well as assembly times are all considered. Since the ReAnalysis algorithm does not directly address the assembly process, and considering the high parallelization potential of the assembly, it is appropriate to compare both methods taking only the factorization and ($f/b_{sub}$) times into account. This data is illustrated
in Figure 4.17(b) where it can be seen that the $GFEM^g_{ReAn}$ offers a dramatic speed-up when only the factorization and $(f/b_{sub})$ is considered. The ratio of computational requirements when the assembly time is included is $\frac{CPU^{hp}}{CPU^{ReAn}} = 1.27$ whereas if the assembly time is not included $\frac{CPU^{hp}}{CPU^{ReAn}} = 4.59$. This result also offers much incentive to seriously consider the parallelization of the assembly process so as to approach the curves in Figure 4.17(b) as closely as possible.

Figure 4.17: CPU time requirements versus time for $hp$-GFEM as well as transient $GFEM^g_{ReAn}$ simulations. In (a) both the factorization and $(f/b_{sub})$, and assembly times are included; (b) only considers the factorization and $(f/b_{sub})$ time.

In this section, the error levels are of much less concern than the CPU time requirements, but it is worth mentioning that the $L^2_{Diff}$ value between the $hp$-GFEM and $GFEM^g_{ReAn}$ results generated with similar levels of local refinement is $L^2_{Diff} = 6.9e-3$, indicating a very good agreement between the results. The curves for internal energy versus time generated with both $hp$-GFEM and $GFEM^g_{ReAn}$ are provided in Figure 4.18 where we see that the two curves are in good agreement.

4.6.2 Load Case 2: Thermal Spike in Center of Panel

In the second loading case of interest, the rough loading is applied near the center of the panel. The focus of the applied Gaussian flux is at $f_{Gauss} = (24.2, 12.0, 0.24)$, with a length of 2 in, i.e. the Gaussian flux is applied only for $11.0 \leq y \leq 13.0$ in. The value of $b$ is again taken as the $x$-coordinate of the focus, in this case $b = 24.2$, and the time-rise constant is taken as $\gamma = 20.0$ s$^{-1}$.
Figure 4.18: Internal energy versus time curves for \( hp \)-GFEM as well as transient \( GFEM^{gl}_{ReAn} \) simulations.

Output in the form of CPU time requirements as a function of the number of time-steps are provided in Figures 4.19(a) and 4.19(b) when considering and neglecting assembly times, respectively. Once again it is seen that significant speed-up is obtained in both plots, with a dramatic difference seen in 4.19(b). The ratio of computational requirements when the assembly time is included is 
\[
\frac{CPU^{hp}_{CPU^{ReAn}}}{CPU^{ReAn}} = 1.78
\]
whereas if the assembly time is not included 
\[
\frac{CPU^{hp}_{CPU^{ReAn}}}{CPU^{ReAn}} = 5.25.
\]

Figure 4.19: CPU time requirements versus time for \( hp \)-GFEM as well as transient \( GFEM^{gl}_{ReAn} \) simulations. In (a) both the factorization and \((f/b_{sub})\), and assembly times are included; (b) only considers the factorization and \((f/b_{sub})\) time.
The $L^2_{Diff}$ value between the $hp$-GFEM and $GFEM^{gl}_{ReAn}$ results generated with similar levels of local refinement is $L^2_{Diff} = 8.1e - 3$, again indicating a very good agreement between the results. The curves for internal energy versus time generated with both $hp$-GFEM and $GFEM^{gl}_{ReAn}$ are provided in Figure 4.20 where we see that the two curves are in good agreement.

![Figure 4.20: Internal energy versus time curves for $hp$-GFEM as well as transient $GFEM^{gl}_{ReAn}$ simulations.](image)

**4.6.3 Load Case 3: Thermal Spike Moving Along Edge of Panel**

In the third loading case of interest, the rough loading is allowed to move along the edge of the panel. The focus of the applied Gaussian flux is initially at $f^0_{Gauss} = (9.3, 1.5, 0.24)$, with a length of 3 in, i.e. the Gaussian flux is applied only for $0.0 \leq y \leq 3.0 \text{ in}$. The value of $b_0$ is taken as the $x$-coordinate of the initial focus, in this case $b_0 = 9.3$, and $b(t) = b_0 + V_{Gauss} t$ where $V_{Gauss} = 1 \text{ in sec}^{-1}$. The time-rise constant is taken as $\gamma = 40.0 \text{s}^{-1}$.

This load case is of particular interest because the thermal spike is moving within the domain. For a standard finite element simulation, either adaptive meshing is required, which is a non-trivial process and may introduce additional error into the solution from the projection of information.
between subsequent, non-nested meshes; or very high levels of refinement will be required over much larger portions of the domain. The two readily available options either pose additional computational complexity, or lead to simulations with prohibitively large numbers of degrees of freedom, neither being desirable. For the $GFEM_{ReAn}^{gl}$ the coarse, global mesh need not be altered in any way as only the enrichments must be updated as the thermal spike moves.

Output in the form of CPU time requirements as a function of the number of time-steps are provided in Figures 4.21(a) and 4.21(b) when considering and neglecting assembly times, respectively. Once again it is seen that significant speed-up is obtained in both plots, with a more dramatic difference in 4.21(b). The ratio of computational requirements when the assembly time is included is $\frac{CPU_{hp}}{CPU_{ReAn}} = 1.27$ whereas if the assembly time is not included $\frac{CPU_{hp}}{CPU_{ReAn}} = 6.39$.

![Figure 4.21](image-url)

Figure 4.21: CPU time requirements versus time for $hp$-GFEM as well as transient $GFEM_{ReAn}^{gl}$ simulations. In (a) both the factorization and $(f/b_{sub})$, and assembly times are included; (b) only considers the factorization and $(f/b_{sub})$ time.

The $L_2^{Diff}$ value between the $hp$-GFEM and $GFEM_{ReAn}^{gl}$ results generated with similar levels of local refinement is $L_2^{Diff} = 3.8e-3$, again indicating a very good agreement between the results. The curves for internal energy versus time generated with both $hp$-GFEM and $GFEM_{ReAn}^{gl}$ are provided in Figure 4.22 where we see that the two curves are in good agreement. Both plots show oscillations in the internal energy versus time curves, due to the level of refinement used for the simulations. Figure 4.23 shows the internal energy versus time curve generated with $GFEM_{ReAn}^{gl}$ and two additional levels of localized refinement. As can be seen in this plot, small oscillations are
somewhat evident in the tail end of the simulation, but the behavior is significantly improved.

Figure 4.22: Internal energy versus time curves for $hp$-GFEM as well as transient $GFEM^gl_{ReAn}$ simulations.

Figure 4.23: Internal energy versus time curve for transient $GFEM^gl_{ReAn}$ simulation utilizing two more levels of local refinement than used for the curve shown in Figure 4.22.
4.6.4 Simulation Summary of Load Cases 1, 2 and 3

The pertinent simulation data for comparing the two methods (\(hp\)-GFEM and \(GFEM^{gl}_{ReAn}\)) is summarized in Table 4.5. The CPU time data in the table, as well as in subsequent tables does not consider assembly time. It should be noted that the third column in the table provides the \(L_2^{Diff}(U(t))\) values for each load case, to again point out the good agreement between the results generated with both methods. As can be seen from the CPU time data, and in contrast to the results presented for the steady-state analysis, the \(GFEM^{gl}_{ReAn}\) offers a significant speed-up in each of the three load cases, even when each load case is considered individually. This is primarily due to the fact that each individual transient load case is roughly equivalent to 40 different steady-state load cases, and as would be expected the method offers its largest gains in efficiency as the number of load cases (time-steps) is increased. With this in mind, we would expect the largest gains in computational efficiency when considering the entire simulation (load cases 1, 2 and 3).

Table 4.5: Direct Comparison of Transient Solution Strategies.

<table>
<thead>
<tr>
<th>Case</th>
<th>CPU Time (hp)-GFEM</th>
<th>CPU Time (GFEM^{gl}_{w/ReAnalysis})</th>
<th>(L_2^{Diff}(U(t)))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9 532.56</td>
<td>2 078.82</td>
<td>6.9e-3</td>
</tr>
<tr>
<td>2</td>
<td>9 193.14</td>
<td>1 750.58</td>
<td>8.1e-3</td>
</tr>
<tr>
<td>3</td>
<td>13 807.09</td>
<td>2 160.83</td>
<td>3.8e-3</td>
</tr>
</tbody>
</table>

Tables 4.6 and 4.7 summarize the CPU time requirements for the transient \(hp\)-GFEM and \(GFEM^{gl}_{ReAn}\), respectively, when the entire simulation is taken into account. As is, the largest individual CPU time requirement is for the initial solve, which is now considered once for the entire simulation, and not once for each load case. Figure 4.24 shows the CPU time versus number of time-steps for the entire simulation. Data is once again plotted with and without the time required for assembly. The ratio of computational requirements when the assembly time is included is \(\frac{CPU_{hp}}{CPU_{ReAn}} = 1.40\) whereas if the assembly time is not included \(\frac{CPU_{hp}}{CPU_{ReAn}} = 5.90\). As can be seen from the plots, the difference between the CPU time requirements is significantly increased with an increase in the number of time-steps, as would be expected.
Table 4.6: Summary of Transient $hp$-GFEM Simulations.

<table>
<thead>
<tr>
<th>Case</th>
<th>ndofs</th>
<th>CPU Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>140540</td>
<td>9532.56</td>
</tr>
<tr>
<td>2</td>
<td>138006</td>
<td>9193.14</td>
</tr>
<tr>
<td>3</td>
<td>158140</td>
<td>13807.09</td>
</tr>
</tbody>
</table>

**Total CPU Time** 32532.79

Table 4.7: Summary of Computational Effort for Transient $GFEM_{gl}$ Simulations with ReAnalysis.

<table>
<thead>
<tr>
<th>Case</th>
<th>ndofs Local</th>
<th>CPU Time Local</th>
<th>ndofs Enr Glob</th>
<th>CPU Time Enr Glob</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12084</td>
<td>356.80</td>
<td>95130</td>
<td>1485.78</td>
</tr>
<tr>
<td>2</td>
<td>9300</td>
<td>278.40</td>
<td>95110</td>
<td>1235.94</td>
</tr>
<tr>
<td>3</td>
<td>16152</td>
<td>445.27</td>
<td>95130</td>
<td>1479.32</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ndofs Init Global</th>
<th>95060</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Time Init Global</td>
<td>236.24</td>
</tr>
</tbody>
</table>

**Total CPU Time** 5517.75

Figure 4.24: CPU time requirements versus time for $hp$-GFEM as well as transient $GFEM_{gl} \text{ReAn}$ simulations. In (a) both the factorization and $(f/b_{sub})$, and assembly times are included; (b) only considers the factorization and $(f/b_{sub})$ time.
The previous results indicate that the largest gains in computational efficiency are obtained as the number of time-steps required for analysis is increased. As such, the transient ReAnalysis with the GFEM$^g_l$ is a good candidate for design scenarios in which the same mesh needs to be analyzed for multiple loading configurations. Again, this is due to the ability to assemble and factorize the global matrices once, and re-use the factorizations at each subsequent time-step. The same holds for multiple loading cases, as the global mesh need not be altered in any way regardless of the location of the loadings. As a result, significant gains are also available in terms of user time, in that only one mesh needs to be created for all of the loading cases to be considered. The proposed method may also prove to be a good candidate for explicit time-integration in which very large numbers of time-steps are required, but that application is beyond the scope of this work.

4.7 Summary

In this chapter, the ReAnalysis algorithm for the GFEM$^g_l$ (GFEM$^g_l_{ReAn}$) is formulated for steady-state and transient applications. The proposed method enables the re-use of large portions of the global matrices associated with the initial uniform, coarse, global mesh.

This has several computational implications as discussed in the previous sections. From the analyses performed in this chapter we have seen:

(i) the GFEM$^g_l_{ReAn}$ can deliver results with the same accuracy as the GFEM$^g_l$;

(ii) good computational efficiency can be achieved when the dimension of $K^g_l$ is small compared to the dimension of $K^0_G$;

(iii) the same global mesh may be used for multiple loading scenarios;

(iv) the gains in CPU time is amplified as the number of loading scenarios, or the number of time-steps increases;

In the final chapter, to further investigate the potential benefits of using coarse elements in GFEM
simulations, explicit analysis of 1-D heat transfer problems with localized thermal gradients is investigated. With larger elements, larger allowable time-steps are obtained, while still maintaining stable simulations. The impact on the critical time-step size, as well as how this translate into computational effort for a simulation is analyzed in the next chapter. It should be noted that the analyses are in 1-D, and as such analytical enrichments are used. Neither the $GFEM^{gl}$ nor the ReAnalysis algorithms are investigated in the next chapter.
5.1 Problem Formulation

Consider a domain \( \Omega \subset \mathbb{R}^3 \) with boundary \( \partial \Omega \) decomposed as \( \partial \Omega = \Gamma_u \cup \Gamma_f \) with \( \Gamma_u \cap \Gamma_f = \emptyset \). The strong form of the governing equation is given by the 3D heat equation

\[
\rho c \frac{\partial u}{\partial t} = \nabla (\mathbf{k} \nabla u) + Q(x, t) \quad \text{in} \quad \Omega
\]

(5.1)

where \( u(x, t) \) is the temperature field, \( \mathbf{k} \) is the thermal conductivity tensor, \( \rho c \) is the volumetric heat capacity, and \( Q(x, t) \) is the internal heat source.

The following boundary conditions are prescribed on \( \partial \Omega \)

\[
u = \bar{u} \quad \text{on} \quad \Gamma_u
\]

(5.2)

\[-\mathbf{k} \nabla u \cdot \mathbf{n} = \bar{f} \quad \text{on} \quad \Gamma_f
\]

(5.3)

where \( \mathbf{n} \) is the outward unit normal vector to \( \Gamma_f \) and \( \bar{f} \) and \( \bar{u} \) are prescribed normal heat flux and temperature, respectively.

The initial conditions must also be satisfied

\[
u(x, 0) = u^0(x) \quad \text{at} \quad t^0
\]

(5.4)

where \( u^0(x) \) is the prescribed temperature field at time \( t = t^0 \).
5.2 Model Problem

The problem selected for analysis is taken from [75], and it involves a sharp spatial gradient in the temperature field (5.5), as well as in the resulting source term (5.6). There is also a temporal gradient, but it is smooth in nature.

\[ u(x,t) = \left( \exp^{-\gamma(x-x_{\text{front}}(t))^2} + \sin \left( \frac{\pi x}{L} \right) \right) * \exp(-t) \]  (5.5)

\[ Q(x,t) = \rho c \frac{\partial u}{\partial t}(x,t) - \kappa \frac{\partial^2 u}{\partial x^2}(x,t), \]  (5.6)

\[ x_{\text{front}}(t) = x_0 + V t \]  (5.7)

The initial and boundary conditions are given in (5.8) and (5.9), respectively.

\[ u(x,0) = \exp^{-\gamma(x-x_0)^2} + \sin \left( \frac{\pi x}{L} \right), \]  (5.8)

\[ u(0,t) = u(L,t) = 0, \]  (5.9)

In the above equations, \( x_0 = 125\text{mm}, L = 500\text{mm}, V = 250\text{mm/sec} \) and \( \gamma \) is a parameter controlling the roughness of the solution. Unless otherwise indicated, the value of \( \gamma \) is taken as 1.0. The material properties are taken as thermal conductivity, \( \kappa = 1 \) and volumetric heat capacity, \( \rho c = \left( \frac{\pi}{L} \right)^2 \). The reference solution (5.5) is plotted in Figure 5.1 and the initial condition (5.8) is plotted in Figure 5.2. As can be seen, from the spatial standpoint, the thermal spike moves from \( \frac{L}{4} \) to \( \frac{3L}{4} \) in 1 sec \( (t^0 = 0\text{sec}, t^f = 1\text{sec}) \). From the temporal standpoint, the solution undergoes a smooth, exponential decay in time.
(a) Reference solution in space and time.

(b) Time slices of reference solution.

Figure 5.1: Temperature field as described in (5.5).

Figure 5.2: Initial conditions as described in (5.8).
5.3 Discrete Equations and Time-Integration

In this chapter, Transient Formulation 2 is used, as derived in detail in Chapter 3, with the time-stepping parameter $\alpha = 0.0$, yielding the conditionally stable Forward Euler algorithm. Substituting $\alpha = 0.0$ into (3.47) leads to the following linear system of equations

$$\frac{1}{\Delta t} M^{n+1} u^{n+1} = \left[ \frac{1}{\Delta t} M^{n+1,n} - K^{n+1,n} \right] u^n + f^{n+1,n} \quad (5.10)$$

where again,

$$M^{n+1} = \int_L \rho c \phi^{n+1} (\phi^{n+1})^T \, dx \quad (5.11)$$

$$M^{n+1,n} = \int_L \rho c \phi^{n+1} (\phi^n)^T \, dx \quad (5.12)$$

$$K^{n+1,n} = \int_L \nabla \phi^{n+1} \kappa (\nabla \phi^n)^T \, dx \quad (5.13)$$

$f^{n+1,n} = \int_L Q^n \phi^{n+1} \, dx \quad (5.14)$

In the previous equations $\phi$ is the vector of finite element shape functions, $\Omega$ is the domain, $Q$ is the internal source, $\rho c$ is the volumetric heat capacity, and $\kappa$ is the thermal conductivity of the material. Vector $u^{n+1}$ is the solution at $t^{n+1}$, and $u^n$ is a known quantity from time $t^n$. 

159
5.4 Generalized Eigenvalue Analysis to Determine Stability Requirements

For the conditionally stable Forward Euler algorithm, special care must be taken in selection of the time-step size (\( \Delta t \)) such that stability is maintained throughout the simulation. We solve the generalized eigenvalue problem, as posed in (5.15) to determine the magnitude of the dominant eigenvalue, \( \lambda_{\text{max}} \). The critical time-step size (\( \Delta t_{\text{cr}} \)) is then related to the dominant eigenvalue as in (5.16). More discussion of this stability criterion can be found in [87].

\[
(K - \lambda M)x = 0 \tag{5.15}
\]

\[
\Delta t_{\text{cr}} = \frac{2}{\lambda_{\text{max}}} \tag{5.16}
\]

In (5.15), when time-dependent shape functions are used, \( M = M^{n+1} \) as defined in (5.11) and \( K = K^{n+1} \), defined as

\[
K^{n+1} = \int_L \nabla \phi^{n+1} \kappa \left( \nabla \phi^{n+1} \right)^T dx \tag{5.17}
\]

When time-dependent shape functions are not used, the distinction is inconsequential because \( M = M^{n+1} = M^{n+1,n} \), and similarly for \( K \).

5.5 GFEM Analysis of Model Problem

5.5.1 Calculation of Critical Time-Step Sizes for Stable Simulations

In this section, (5.15) is solved for a series of uniform meshes, with different element sizes and polynomial orders. Table 5.1 provides the details (polynomial order, element size) for each dis-
cretization to be used for analysis, as well as a summary of the stability criterion for each discretization. The results for each type of element are investigated in more detail in subsequent sections. For the exponential elements, the eigenvalues need to be calculated at each time-step due to the changing discretization, and then the most stringent $\Delta t_{cr}$ is selected to yield a stable simulation.

Table 5.1: Summary of Output for Calculation of $\Delta t_{cr}$.

<table>
<thead>
<tr>
<th>Type</th>
<th>p-order</th>
<th>h (width)</th>
<th>$\lambda_{max}$</th>
<th>$\Delta t_{cr}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>1</td>
<td>1.250</td>
<td>196910</td>
<td>1.02e-5</td>
</tr>
<tr>
<td>Linear</td>
<td>1</td>
<td>0.625</td>
<td>787402</td>
<td>2.54e-6</td>
</tr>
<tr>
<td>Polynomial</td>
<td>2</td>
<td>2.500</td>
<td>246130</td>
<td>8.13e-6</td>
</tr>
<tr>
<td>Polynomial</td>
<td>2</td>
<td>1.250</td>
<td>985222</td>
<td>2.03e-6</td>
</tr>
<tr>
<td>Polynomial</td>
<td>2</td>
<td>0.625</td>
<td>3938400</td>
<td>5.08e-7</td>
</tr>
<tr>
<td>Polynomial</td>
<td>4</td>
<td>5.000</td>
<td>389910</td>
<td>5.13e-6</td>
</tr>
<tr>
<td>Polynomial</td>
<td>4</td>
<td>2.500</td>
<td>1559300</td>
<td>1.28e-6</td>
</tr>
<tr>
<td>Polynomial</td>
<td>4</td>
<td>1.250</td>
<td>6242700</td>
<td>3.20e-7</td>
</tr>
<tr>
<td>Exponential</td>
<td>2</td>
<td>100</td>
<td>128205</td>
<td>1.56e-5</td>
</tr>
<tr>
<td>Exponential</td>
<td>3</td>
<td>100</td>
<td>125786</td>
<td>1.59e-5</td>
</tr>
</tbody>
</table>

5.5.2 Analysis of Model Problem Using Polynomial Elements

**Linear Elements** In this section, simulation data generated with linear elements is investigated in detail. The accuracy of the solution, as well as CPU time required to produce the solution are examined. For the plots that deal with internal energy, the internal energy at time-step $n$ ($U_n$), is defined as the inner product of the flux and temperature gradient vectors, as shown in (3.58). Since the analytical solution is known, an exact curve for the internal energy as a function of time can be generated and plotted as a means for comparison. To put a single number which can serve to tell how well the curves match up, a discrete $L_2$-norm for the error is calculated as in (3.60) where the summation is performed over each data point along the curve.

For the CPU time shown in the subsequent tables, both the assembly and solution times are considered. For polynomial elements, the stiffness matrix is only assembled once, whereas for the exponential elements, with time-dependent shape functions, the assembly is performed at each
time-step. Since the exponential elements do cause this slight inconvenience, it seems appropriate that the CPU times recorded are reflective of this drawback.

The output for linear elements is summarized in Table 5.2. As expected, with a more refined mesh as well as a smaller time step, greater accuracy is obtained, but at a severe cost in CPU time. The internal energy is plotted as a function of time in Figure 5.3. As can be seen, there is a lot of noise in the solution due to the poor quality. As the mesh is refined, and the time-step size is reduced to maintain stability, it can be seen that the noise in the solution seems to damp out and the oscillations become smaller. Not surprisingly, greater accuracy is also achieved. The same data is then plotted in Figure 5.4 where a least squares fit is used to smooth out the noise in the data. It should also be noted that there is good agreement between the values of $\Delta t$ used for a stable analysis and the predicted values obtained from the generalized eigenvalue analysis.

Table 5.2: Summary of Output for Linear Elements.

<table>
<thead>
<tr>
<th>$\Delta t$</th>
<th>Stability</th>
<th>$h$ (width)</th>
<th>NumberTimeSteps</th>
<th>$L_2^{Error}$</th>
<th>CPU Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.015e-5</td>
<td>Yes</td>
<td>1.250</td>
<td>98 500</td>
<td>0.3231</td>
<td>1.842e5</td>
</tr>
<tr>
<td>1.03e-5</td>
<td>No</td>
<td>1.250</td>
<td>97 087</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2.53e-6</td>
<td>Yes</td>
<td>0.625</td>
<td>396 000</td>
<td>0.0909</td>
<td>1.611e6</td>
</tr>
<tr>
<td>2.56e-6</td>
<td>No</td>
<td>0.625</td>
<td>390 000</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

**High Order Polynomial Elements** This section contains output for simulations run using high order polynomial elements. Elements with quadratic ($p = 2$) and quartic ($p = 4$) shape functions are selected for analysis. Due to the excessive number of time steps required for stability, only a percentage of the entire simulation is run. For quadratic elements the simulation is run to 0.5 seconds, and for quartic elements the simulation is only run to 0.25 seconds. The projected CPU time is then taken as the time to run a percentage of the simulation, divided by that percentage. While this CPU time will not be exact, it is representative of the time that would be required to run the entire simulation. In order to determine if the relative error, $L_2^{Error}$, values are valid for comparison, Table 5.3 shows the values obtained for linear elements when only certain percentages of the
Figure 5.3: Internal energy as a function of time for linear elements.

Figure 5.4: Internal energy as a function of time for linear elements, using least squares to smooth out data.
simulation are considered. From this table it is apparent that the relative error values are relatively insensitive to the percentage of the simulation data considered. As such it is not unreasonable to directly compare the values obtained for the high order elements and shorter simulations with those obtained for linear and exponential elements corresponding to the entire simulation.

Table 5.3: Comparison of $L_2^{\text{Error}}$ for Different $t^f$. (Linear Elements)

<table>
<thead>
<tr>
<th>$t^f$</th>
<th>$h$ (width)</th>
<th>$L_2^{\text{Error}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>1.25</td>
<td>0.3228</td>
</tr>
<tr>
<td>0.50</td>
<td>1.25</td>
<td>0.3230</td>
</tr>
<tr>
<td>1.00</td>
<td>1.25</td>
<td>0.3231</td>
</tr>
</tbody>
</table>

Tables 5.4 and 5.5 summarize the data for quadratic and quartic elements, respectively. The internal energy is then plotted as a function of time in Figures 5.5 and 5.7. In both cases, we again see noise in the data which is damped out as the element size and time-step size are both reduced. Once again we also obtain better accuracy in the solution, but still at a severe cost in CPU time. Figures 5.6 and 5.8 have plots of the data for which a least squares fit has been used in order to smooth out the data. It should again be noted that the values of $\Delta t$ shown in Tables 5.4 and 5.5 are in good agreement with the predicted values in Table 5.1.

Table 5.4: Summary of Output for Quadratic Elements.

<table>
<thead>
<tr>
<th>$\Delta t$</th>
<th>Stability</th>
<th>$h$ (width)</th>
<th>NumberTimeSteps</th>
<th>$L_2^{\text{Error}}$</th>
<th>CPU Time (projected)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.10e-6</td>
<td>Yes</td>
<td>2.50</td>
<td>123 400</td>
<td>0.2851</td>
<td>1.213e5</td>
</tr>
<tr>
<td>8.15e-6</td>
<td>No</td>
<td>2.50</td>
<td>122 700</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1.99e-6</td>
<td>Yes</td>
<td>1.25</td>
<td>502 000</td>
<td>0.0437</td>
<td>1.237e6</td>
</tr>
<tr>
<td>2.05e-6</td>
<td>No</td>
<td>1.25</td>
<td>487 800</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 5.5: Summary of Output for Quartic Elements.

<table>
<thead>
<tr>
<th>$\Delta t$</th>
<th>Stability</th>
<th>$h$ (width)</th>
<th>NumberTimeSteps</th>
<th>$L_2^{\text{Error}}$</th>
<th>CPU Time (projected)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.10e-6</td>
<td>Yes</td>
<td>5.00</td>
<td>196 000</td>
<td>0.2978</td>
<td>2.049e6</td>
</tr>
<tr>
<td>5.15e-6</td>
<td>No</td>
<td>5.00</td>
<td>194 000</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Figure 5.5: Internal energy as a function of time for quadratic elements.

Figure 5.6: Internal energy as a function of time for quadratic elements, using least squares to smooth out data.
Figure 5.7: Internal energy as a function of time for quartic elements.

Figure 5.8: Internal energy as a function of time for quartic elements, using least squares to smooth out data.
5.5.3 Analysis of Model Problem Using Special Enrichment Functions

This section contains output for simulations using elements with special enrichment functions. We use knowledge of the solution to select our special enrichment function (5.18). The special, exponential enrichment functions are only applied to the nodes whose supports contain the thermal spike. All other nodes have only polynomial enrichments.

\[ L_{αi}(x, t) = \exp^{-(x-x_{\text{front}}(t))^2} \]  

Table 5.6 summarizes the data for each run. The internal energy is then plotted as a function of time in Figure 5.9. In the figure, Linear Exponential refers to elements with only a linear shape function, and an exponential shape function; whereas Quadratic Exponential refers to elements with linear and quadratic shape functions, as well as an exponential shape function.

The output for the exponential elements is summarized in Table 5.6. As can be seen, very good accuracy is obtained with the use of the special enrichment functions. The internal energy is plotted as a function of time in Figure 5.9. From looking at the plot we can see that there is no noise in the data, and that there is no discernable difference between the curves. In order to see a difference in the data, Figure 5.10 shows a significantly zoomed-in view of the curves. Again, no least squares fit was required for the special elements since the quality of the solution is very good, and there is no noise in the data.

<table>
<thead>
<tr>
<th>Δt</th>
<th>Stability</th>
<th>h (width)</th>
<th>NumberTimeSteps</th>
<th>L_2^{Trop}</th>
<th>CPU Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Exponential</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.54e-5</td>
<td>Yes</td>
<td>100</td>
<td>65 000</td>
<td>2.90e-4</td>
<td>2.756e4</td>
</tr>
<tr>
<td>1.62e-5</td>
<td>No</td>
<td>100</td>
<td>61 800</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Quadratic Exponential</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.54e-5</td>
<td>Yes</td>
<td>100</td>
<td>65 000</td>
<td>1.92e-4</td>
<td>2.789e4</td>
</tr>
<tr>
<td>1.62e-5</td>
<td>No</td>
<td>100</td>
<td>61 800</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Due to the fact that the spike is moving throughout the course of the simulation, the Δt_{cr} may be
Figure 5.9: Internal energy as a function of time for exponential elements.

Figure 5.10: Internal energy as a function of time for exponential elements. Zoomed in to see a difference in the curves.
different at each time-step, depending on the location of the spike with respect to a given node. It is noted in [73, 74], for fracture applications with standard FEA, a term in the stiffness matrix is \( \sim \frac{1}{h_x} \) (\( h_x \) is the element size) whereas the corresponding term in the mass matrix is \( \sim h_x \), yielding an infinitely small time-step requirement to maintain stability. For fracture applications with X-FEM’s use of the discontinuous Heaviside enrichment, the mass matrix becomes singular as the crack front approaches the boundary of \( \omega_\alpha \), the support of the partition of unity (PoU) function, again yielding an infinitely small time-step requirement to maintain stability [73, 74]. For the present application, the enrichment functions used do not show this behavior as the spike location nears the boundary of a PoU function’s support, so the stability requirement does not become infeasible. There is some dependency upon the location of the spike with respect to the location within the nodal support, but it is not as dramatic as that seen in the application to fracture. Table 5.7 shows the value for \( \Delta t_{cr} \) as it is affected by the location of the spike with respect to a nodal support. Distance refers to the distance between the node with the special enrichment, and the location of the moving, thermal front.

Table 5.7: Effect of Spike Location on \( \Delta t_{cr} \).

<table>
<thead>
<tr>
<th>Distance</th>
<th>( \Delta t_{cr} )</th>
<th>( \frac{\Delta t_{cr}}{\Delta t_{max}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>7.01e-5</td>
<td>1</td>
</tr>
<tr>
<td>( \frac{h_x}{2} )</td>
<td>2.60e-5</td>
<td>2.70</td>
</tr>
<tr>
<td>( \approx h_x )</td>
<td>1.56e-5</td>
<td>4.50</td>
</tr>
</tbody>
</table>

5.5.4 Effect of Volumetric Heat Capacity Magnitude

In this section the effect of the magnitude of the volumetric heat capacity, \( \rho c \) is investigated. A problem very similar to the model problem is analyzed, but in this instance the spike remains stationary, as indicated by the reference solution

\[
\begin{align*}
&u(x, t) = \left( \exp^{-\gamma(x-x_0)^2} + \sin \left( \frac{\pi x}{L} \right) \right) \ast \exp \left( \frac{1}{\gamma t} \right) \quad 0 < x < L \\
&\quad (5.19)
\end{align*}
\]
where again, the internal source is derived as

\[ Q(x,t) = \rho_c \frac{du}{dt}(x,t) - \kappa \frac{d^2u}{dx^2}(x,t), \]  

(5.20)

The initial and boundary conditions are the same as those applied to the original model problem, (5.8) and (5.9). A plot of the reference solution is shown in Figure 5.11, where the solution is seen to undergo the same decay in time, with the spike remaining stationary in space, with a fixed \( x_0 = 125 \text{mm} \). For larger values of \( \rho_c \), the critical time-steps become larger, as such longer simulations are run, but the reference solution is now parameterized by \( t_f \), so only one reference curve is required.

![Figure 5.11: Reference solution for simulations involving larger values of \( \rho_c \), described by (5.19).](image)

Simulation data is presented in Table 5.8 for simulations with and without the time-dependent exponential enrichment (5.21) applied to nodes which contain the thermal spike. For analyses utilizing (5.21), simulations are run with 110 time-steps, and elements of size \( h_x = 100 \text{mm} \). Simulations run without (5.21) have 1400 time steps, and quadratic elements with \( h_x = 1.25 \text{mm} \).

\[ L_{\alpha_i} = \left\{ \exp^{-\left(x-x_0\right)^2} \ast \exp\left(\frac{t}{t_f}\right) \right\} \]  

(5.21)
Table 5.8: Simulation Data for Varying Magnitudes of $\rho_c$.

<table>
<thead>
<tr>
<th>$\rho_c$</th>
<th>$t_f$</th>
<th>$\Delta t_{cr}$</th>
<th>$\Delta t$</th>
<th>$L_2^{error}$</th>
<th>CPU Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exponential Basis Functions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>7</td>
<td>0.0667</td>
<td>0.0636</td>
<td>3.63e-3</td>
<td>3.672</td>
</tr>
<tr>
<td>0.5</td>
<td>35</td>
<td>0.3333</td>
<td>0.3182</td>
<td>3.63e-3</td>
<td>3.701</td>
</tr>
<tr>
<td>1.0</td>
<td>70</td>
<td>0.6667</td>
<td>0.6364</td>
<td>3.63e-3</td>
<td>3.764</td>
</tr>
<tr>
<td>5.0</td>
<td>350</td>
<td>3.3333</td>
<td>3.1818</td>
<td>3.64e-3</td>
<td>3.731</td>
</tr>
<tr>
<td>10</td>
<td>700</td>
<td>6.6667</td>
<td>6.3636</td>
<td>3.67e-3</td>
<td>3.814</td>
</tr>
<tr>
<td>Polynomial Basis Functions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>7</td>
<td>0.0052</td>
<td>0.0050</td>
<td>0.0678</td>
<td>1.03e3</td>
</tr>
<tr>
<td>0.5</td>
<td>35</td>
<td>0.0260</td>
<td>0.0250</td>
<td>0.0678</td>
<td>1.06e3</td>
</tr>
<tr>
<td>1.0</td>
<td>70</td>
<td>0.0521</td>
<td>0.0500</td>
<td>0.0678</td>
<td>9.87e2</td>
</tr>
<tr>
<td>5.0</td>
<td>350</td>
<td>0.2604</td>
<td>0.2500</td>
<td>0.0679</td>
<td>1.01e3</td>
</tr>
<tr>
<td>10</td>
<td>700</td>
<td>0.5208</td>
<td>0.5000</td>
<td>0.0681</td>
<td>1.03e3</td>
</tr>
</tbody>
</table>

As can be seen from the table, the exponential enrichment functions once again deliver much higher levels of accuracy, with smaller requirements in terms of CPU time. Figure 5.12 shows time-slices of the solutions generated using the exponential enrichment, and as would be expected, the solutions generated match up very nicely with the reference solution shown in Figure 5.11. Figure 5.13 shows time-slices of the solutions generated using quadratic elements. The solutions are also in good agreement with the reference solution, but in this case there is a much greater cost in terms of CPU time required to generate the solutions.

![Simulation results](image-url)

Figure 5.12: Simulation results generated using the enrichment in (5.21).
5.6 Summary

Before comparing the different element types, several general conclusions can be made. Not surprisingly, increasing mesh refinement for a given polynomial order yields a better $L^2$ Error value, but does so at a significant increase in CPU time. Likewise, for the exponential elements, raising the polynomial order of the elements at a fixed level of refinement also yielded better $L^2$ Error values, but at only a slightly higher requirement for CPU time. In regards to the accuracy of the $\Delta t_{cr}$ produced from the generalized eigenvalue problem: for each discretization analyzed, the values of $\Delta t_{cr}$ proved to be very reliable.

For the sake of an easy comparison, the pertinent data collected is summarized in Table 5.9. With the data summarized here we can better determine if the exponential elements offer a significant reduction in CPU time spent in order to achieve a given level of error. Just to note, in the column for element Type, Exp1 = linear exponential elements, and Exp2 = quadratic exponential elements. From comparing the data in the table, it is quite clear that the exponential elements offer superior performance when compared to polynomial elements. The exponential elements offer extreme reductions in both the error values, as well as the CPU time required to generate the data. Not only are fewer time-steps required, but the system of equations is also significantly smaller, as illustrated by the dramatic difference in the mesh densities, shown in Figure 5.14. The exponential
elements also yield results which do not show any noise in the data. The overall conclusion to be made is that for the case of explicit time-stepping, the exponential elements do in fact offer far superior behavior in regards to: time-step size required for stability, accuracy of solution, and CPU time required.

Table 5.9: Summary of Simulation Data.

<table>
<thead>
<tr>
<th>Type</th>
<th>$h$ (width)</th>
<th>$L_2^{error}$</th>
<th>CPU Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>1.250</td>
<td>0.3231</td>
<td>1.842e5</td>
</tr>
<tr>
<td>Linear</td>
<td>0.625</td>
<td>0.0909</td>
<td>1.611e6</td>
</tr>
<tr>
<td>Quadratic</td>
<td>2.50</td>
<td>0.2851</td>
<td>1.213e5</td>
</tr>
<tr>
<td>Quadratic</td>
<td>1.25</td>
<td>0.0437</td>
<td>1.237e6</td>
</tr>
<tr>
<td>Quartic</td>
<td>5.00</td>
<td>0.2978</td>
<td>2.049e6</td>
</tr>
<tr>
<td>Exp1</td>
<td>100</td>
<td>2.90e-4</td>
<td>2.756e4</td>
</tr>
<tr>
<td>Exp2</td>
<td>100</td>
<td>1.92e-4</td>
<td>2.789e4</td>
</tr>
</tbody>
</table>

Figure 5.14: Comparison of element sizes for exponential and polynomial meshes.
Chapter 6
Contributions and Future Work

6.1 Contributions

In this research, the generalized FEM with global-local enrichments ($GFEM^{gl}$) [28, 31, 66, 82, 83] is formulated for, and applied to steady-state and transient heat transfer problems with solutions exhibiting highly localized, sharp thermal gradients.

The proposed method allows for:

(i) the possibility of capturing localized solution features using uniform, coarse, global meshes. This removes, for example, the need to refine global meshes that are usually complex and very large. A single global mesh can be used to analyze the effect of localized thermal loads at different parts of a structure. All that is needed is the computation of local solutions and the hierarchical enrichment of the global solution space. Additional computational implications of this feature of the $GFEM^{gl}$ are discussed in Section 2.3.1 and in [28];

(ii) the size of the enriched global problem is about the same as the initial global problem and it does not depend on the size or discretization used in the local problems;

(iii) the solution of multiple local problems can be parallelized without difficulty allowing the solution of large problems very efficiently.

(iv) the $GFEM^{gl}$ brings the benefits of GFEM to a broader class of problems: problems where limited or no information about the solution is known a-priori in order to provide analytical enrichments, e.g., multiscale problems, localized non-linearities, etc.
(v) the $GFEM^{gl}$ and $hp$-GFEM may be used with traditional time-stepping methods for parabolic partial differential equations with minor modifications. In this work the generalized trapezoidal rule, or $\alpha$-method is re-formulated and applied to transient $GFEM^{gl}$ simulations involving time-dependent enrichment functions.

(vi) the GFEM with special enrichment functions can provide accurate solutions, on relatively coarse meshes in transient analyses.

(vii) the $GFEM^{gl}$, taken as a generalization of the previous case, also produces accurate and efficient solutions in transient analyses when very limited information is known about the solution a-priori.

(viii) the $GFEM^{gl}$ enables the re-use of the factorization of the large, global matrices at subsequent load configurations and time-steps, yielding significant increases in the computational efficiency of the simulations run.

(ix) the GFEM with special enrichment functions allows for larger allowable time-step sizes in conditionally stable simulations. Simulations require fewer time-steps, significantly less CPU time, and yield high levels of accuracy.

6.2 Future Work

The $GFEM^{gl}$ has shown the potential to resolve localized, transient thermal spikes on relatively coarse finite element meshes, through the use of the numerically-generated enrichment functions. The algorithm has also shown the potential to offer significant speed-up in the CPU time required to run a transient simulation. While the previous results are very encouraging, and the focus of the current research project, there are still many areas which need to be actively investigated so as to further increase the efficiency of the algorithm and make it more appealing and applicable for use in practical engineering design situations.
6.2.1 Parallelization of Assembly Algorithm

The $GFEM^{gl}_{ReAn}$ has shown the ability to offer computational speed-up when compared to simulations run using standard finite element analyses. But, as the results have shown, there is great potential to significantly increase the savings in CPU time requirements if the algorithm is extended to better utilize the ever-growing computational resources which are becoming more available. Unlocking the parallelization potential of the $GFEM^{gl}_{ReAn}$ is a natural extension of this work, and the benefits could prove to be of great use to the scientific community. The algorithm has several areas in which parallelization should be explored.

One major result from the analyses in Chapter 4 is that the assembly algorithm is currently the bottleneck for the $GFEM^{gl}_{ReAn}$. The plot in Figure 6.1 shows the impact of the assembly algorithm on the CPU time requirements for the $GFEM^{gl}_{ReAn}$ simulation. The ratio of CPU time requirements is $\frac{CPU_{w/Assembly}}{CPU_{w/o Assembly}} = 4.91$.

![Figure 6.1: Difference in CPU time requirements for the $GFEM^{gl}_{ReAn}$ when the assembly time is either considered or neglected.](image)

In general, the assembly process is a readily parallelizable algorithm, as each element may be assembled separately and the contributions then added to the global matrices. However, there are potential speed-bumps, in that several elements contribute to the same entries in the global matrices, and the so-called race condition may occur when multiple elements sharing a connectivity are
assembled simultaneously. With this in mind, special consideration should be given the the proper sorting of elements such that the assembly may be performed in parallel, while avoiding the race condition as much as possible. If the assembly can be significantly sped-up, the two curves in Figure 6.1 can approach one-another as closely as possible, yielding significant gains in computational efficiency.

### 6.2.2 Parallelization of Local Solution: Master/Sub-Local Approach

The parallel solution of a local problem is another potential area of interest. As the structures to be analyzed become larger, it is likely that so to may the local domain sizes. It is quite feasible that a sharp thermal load may be applied over a long length of the structure, yielding a local problem which is itself very large. In this instance it is possible the break down the local domain (referred to now as master-local problem) into sub-local problems, each corresponding to a nodal support from the master-local domain. Each of these sub-local problems may obtain boundary conditions from the master-local domain, and they may then be solved in parallel. Each sub-local solution provides a portion of the over-all local enrichment function, which is applied back to the global domain to resolve the localized features on the coarse, global mesh. The method is shown schematically in Figure 6.2. While this algorithm has been formulated already, the parallelization potential, as well as the effects of the master-local domain boundary conditions on the quality of the sub-local domain enrichment functions has not been studied in any great detail.

The algorithm can be readily extended to multiple local domains originating from different locations in the global domain. Each of these master-local domains can be broken up into corresponding sub-local domains, and the whole lot of sub-local problems can be solved in parallel to generate enrichment functions for the global domain, regardless of which master-local domain they originated from.
6.2.3 Effect of Boundary Condition Type on Local Solution Quality

The use different boundary condition types in the local domain is another potential area of interest for future investigation. Of greatest interest is the potential use of Cauchy boundary conditions instead of Dirichlet boundary conditions, as was investigated for linear elastic fracture mechanics in [67]. In [67], the spring boundary conditions yield slightly better results than the Dirichlet boundary conditions. The motivation for Cauchy boundary conditions is that the coarse global solution yielding the Dirichlet boundary conditions is in general of poor quality, which may lead to correspondingly poor enrichment functions. With the current implementation, Dirichlet boundary conditions are applied using the penalty method, which amounts to the limiting case of Cauchy boundary conditions, with a very high spring stiffness (solid mechanics) or convective coefficient (heat transfer). Applying Cauchy boundary conditions instead amounts to a smaller spring stiffness (convective coefficient), and in essence enforces the potentially poor Dirichlet boundary conditions generated in the coarse global solve in a less stringent manner. As such, the quality of the enrichment function is not as significantly impacted by the quality of the boundary conditions obtained from the global domain.
The use of Cauchy boundary conditions may be of greatest use with the master/sub-local approach, where the local domains become smaller, and boundary conditions are applied to element faces which are closer to the region of interest. As the faces become closer to the region of interest, we will in general expect the quality of the boundary conditions to deteriorate. In the case of a localized thermal spike, it is very likely that the coarse global solution will produce boundary conditions which are locally very smooth, when in reality the enrichment function should be rough along the element faces. In this case, enforcing the smooth boundary condition less strictly, i.e. using a smaller convective coefficient, may yield sufficiently good results, which is more computationally efficient than the alternative of requiring multiple iterations before reasonable accuracy is obtained.

6.2.4 Development of A-Posteriori Error Estimates for the GFEM\textsuperscript{gl}

It has been well-documented that the quality of local boundary conditions is of great concern for GFEM\textsuperscript{gl} simulations. As such, it would be very beneficial to develop a-posteriori error estimates for the quality of the boundary conditions applied to the local domain. Development of error estimates are very important in this methodology due to the fact that error is introduced not only through the discretization used, as is seen in standard FEM, but also from the local boundary conditions, which have the potential to severely restrict the accuracy of the enrichment functions. With this in mind, it is essential to be able to estimate the levels of error arising from the local boundary conditions if the methodology is to be used in the industry. With the development of an a-posteriori error estimate, the code can internally determine if multiple iterations are required for good convergence, or if the boundary condition error is sufficiently small already. In this way, the methodology will be more user-friendly, and require less expertise from the user to determine if a given solution is of good quality, or if further iterations are required.
6.2.5 Thermo-Mechanical Coupling

In order to be of the greatest use in engineering design, it is desirable to compute the stress/strain fields resulting from the localized thermal loadings. As such, a one-way thermo-mechanical coupling can be done, linking the thermal $GFEM^{gl}$ solver to a 3D elasticity, FEM solver. The impact on the strain fields can provide useful insight into the resulting stresses induced in the materials, as well as in the potential for warping due to non-constant through-the-thickness temperature distributions. The latter may have great impact on the dynamic behavior, as well as the stability of structural elements. Both of which are of acute interest to design engineers.

6.2.6 Explicit Time-Stepping with the $GFEM^{gl}$

As has been discussed in Chapter 4, the $GFEM^{gl}$ provides relatively larger gains in CPU time as the number of time-steps is increased. With this in mind, explicit time-stepping algorithms, in which very small time-steps are often required for stability, make for an appropriate application area for the $GFEM^{gl}$. On top of the savings in CPU time arising from the efficient re-use of global matrices, the $GFEM^{gl}$ may also allow for larger time-steps, as was seen in Chapter 5. In Chapter 5, an analytical enrichment function is used in the 1-D case along with coarse elements, yielding larger allowable time-steps to go along with fewer equations to be solved at each time-step. The use of an analytical enrichment function is the 1-D analog of the more general $GFEM^{gl}$ used in 3-D.

In order to better understand the potential gains in the allowable time-step size for $GFEM^{gl}$ simulations, it is useful to consider a single $h$-extension performed in the local domain, along with the conceptual bounds on the allowable time-steps offered by the 1-D analog (Table 6.1).

With zero levels of refinement, the enrichment function will behave very similarly to a standard polynomial shape function defined on a coarse mesh. In this instance the allowable time-step will be relatively large, as seen in the first line of the table, but the accuracy delivered by the discretization is likely to be very poor. At the other end of the spectrum, as increased levels of
refinement are used in the local domain, the shape of the enrichment function will approach the shape of the analytical enrichment used in 1-D, yielding a more stringent time-step, as shown in the second line of the table, but the accuracy delivered will also be significantly improved. Whereas the critical time-step will become smaller as the quality of the enrichment function improves, it is still very likely that the allowable time-step will be larger than the associated $\Delta t_{cr}$ for a polynomial mesh with high levels of local refinement, capable of delivering high levels or accuracy, as was the case in Chapter 5. Explicit GFEM$^3_l$ simulations will provide an interesting investigation into the trade-off that is likely to be seen between the quality of the enrichment function, and the size of the associated $\Delta t_{cr}$.

### 6.2.7 Fluid-Structure Load Transfer

An area of potentially great interest is in the use of fluid-structure interaction for the purpose of applying the thermal loadings to the solid domain. The first reason being that the loads likely to be seen by a structural element in hypersonic flight are not well-known, so the loadings themselves are likely to be generated with computational fluid dynamics (CFD). As such, it is appropriate to apply the loads generated by the CFD simulations directly to the thermal model. Of potentially greater use, is the ability to analyze the actual interaction between the two domains; i.e. how the resulting temperature distributions, as well as the deflections in the structural elements (after thermo-mechanical coupling is done) effect the flow field itself. With the fully-integrated solver, great insight into not only how the fluxes generated by the fluid impact the structural response, but also how the structural response will then impact and alter the fluid flow, will be made available to the engineers working on the actual design of the hypersonic aircraft. The fluid-structural coupling
will itself pose difficulties which will need to be dealt with if an accurate coupling is to be done. In general, the meshes for the fluid and structural domains will not match up at the fluid-structure interface, requiring the use of a common refinement scheme along the interface to minimize the potential information loss due to the information transfer between the fluid and structural domains. The overall fluid-structural coupling scheme could be done using *Rocstar* [46], a code developed and maintained by the Center for Simulation of Advanced Rockets at the University of Illinois at Urbana-Champaign. The code has multi-physics capabilities, along with the required capability of generating a common refinement scheme at the fluid-structure interface. With this in mind, it may be a worthwhile endeavor to incorporate the capabilities developed for the *GFEM* into the *Rocstar* platform.
References


