

Investigation of Allowable Time-Step Sizes for Generalized Finite Element Analysis of the Transient Heat Equation

P. O'Hara[†], C.A. Duarte^{†*} and T. Eason[‡]

[†]Department of Civil and Environmental Engr., University of Illinois at Urbana-Champaign,

Newmark Laboratory, 205 North Mathews Avenue, Urbana, Illinois 61801, USA

e-mails: {pohara,caduarte}@illinois.edu

[‡]Air Force Research Laboratory

Air Vehicles Directorate, WPAFB, Ohio, USA,

e-mail: Thomas.Eason@wpafb.af.mil

September 14, 2010

Abstract

This paper investigates the heat equation for domains subjected to an internal source with a sharp spatial gradient. The solution is first approximated using linear finite elements, and sufficiently small time-step sizes to yield stable simulations. The main area of interest is then in the ability to approximate the solution using Generalized Finite Elements, and again explore the time-step limitations required for stable simulations. Both high order elements, as well as elements with special enrichments are used to generate solutions. When compared to linear finite elements, the high order elements deliver better accuracy at a given level of mesh refinement, but do not offer an increase in critical time-step size. When special enrichment functions are used, the solution can be approximated accurately on very coarse meshes, while yielding solutions which are both accurate and computationally efficient. The major conclusion of interest is that the significantly larger element size yields larger allowable time-step sizes while still maintaining stability of the time-stepping algorithm.

*Corresponding author. E-mail: caduarte@illinois.edu

1 Introduction

There are many application areas in engineering practice which involve the analysis of structure which have multiple spatial scales of interest. In this work, the desired application area is in the transient analysis of structures which are subjected to localized thermal loadings. Welding [28, 50] and laser forming of metal plates [51, 52] are two possible applications areas exhibiting localized heating events. The motivation for this particular study is the localized heating experienced by hypersonic flight vehicles due to the effects of shock impingement on the skin of the aircraft [10, 20, 21, 34, 42, 46, 47, 49].

The analysis of this type of problem has been the emphasis of many previous investigations [9, 27, 32, 45, 48], and in the 3D case, using standard finite elements, an excessively large number of degrees-of-freedom is required [39], making efficient analyses very difficult. These difficulties are further amplified when transient simulations involving many time-steps are required, and the excessively large system of equations needs to be solved at each time step.

Adaptive meshing in the transient setting is possible, but it 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.

The high refinement levels required by the FEM to achieve suitable accuracy further complicates the situation through the requirement of excessively small time steps. There have been many investigations similar to the one presented in this paper which have focused on the alleviation of the infeasibly small time step requirements.

There is a growing body of literature addressing explicit time-integration for dynamic fracture within the X-FEM framework. Menouillard, et al. [30] propose a mass lumping technique which enhances the allowable time-step size in dynamic X-FEM 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 develop their lumping strategy 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 obtained using standard finite elements of the same size. The authors note that the X-FEM and FEM critical time-step sizes are related by $\frac{1}{\sqrt{2}}$.

Menouillard, et al. further extend the idea of mass lumping for the X-FEM in [31]. In this work the mass-lumping technique combined with a straight-forward transformation of crack-front coordinates results in a block-diagonal mass matrix yielding allowable time-step sizes which again do not tend toward zero

as the crack approaches a node, and remain on the order of allowable time-steps using standard finite elements of the same size.

Elguedj, et al. [19] extend the mass-lumping work of Menouillard, et al. [31] to be used with arbitrary enrichments, and not solely for enrichments used for fracture mechanics applications. They adopt the same basic lumping strategy for arbitrary enrichments and are once again able to obtain critical time-steps that are on the order of those obtained with standard FEM. The authors note that the allowable time step size is dependent on where the discontinuity is located relative to support boundaries. They note that the most stringent allowable time-step can be one half of that obtained for standard FEM and as such put forth this rule of thumb to be used for dynamic X-FEM simulations utilizing arbitrary enrichments. In part two [23] of the previous paper the authors then propose a stable-explicit/explicit dynamic scheme, based on a stable-explicit scheme proposed by Chang [6–8]. The method as posed by Chang is unconditionally stable, and second order accurate. The method 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.

Mesh partitioning methods are similar to the stable-explicit/explicit dynamic scheme proposed by Chang and utilized by Elguedj, et al. in that they allow for different time-integration schemes in different portions of the mesh [4, 5, 25, 26, 35]. This class of methods often uses implicit-explicit mesh partitioning so as to allow for the use of implicit time-integration with significantly larger time-step sizes for a portion of the mesh, and do not have the entire simulation bound by the more stringent time step size required for stability in the explicit partition. While this class of methods is aimed at more efficient time-integration, they are of less interest to the research presented in this paper as they do not directly seek to *increase* the allowable time step size.

In this work we aim to relax meshing requirements while still achieving a high level of accuracy through the use of special enrichment functions. To this end, the GFEM relies on a-priori knowledge of the solution to obtain analytically defined enrichment functions. While these enrichments are able to deliver accurate solutions on relatively coarse meshes, we seek in this work to investigate how the larger elements used will impact the magnitude of the allowable time-step size.

The outline of this paper is as follows. In the next section we present the formulation for 3D heat transfer, followed by a description of the model problem to be used for analysis in Section 3. A brief description of GFEM approximations is presented in Section 4. Section 5 discusses the discretization in both space

and time, along with the stability requirements for numerically approximating the heat equation. Section 6 presents an analysis of the model problem with the use of both polynomial, and exponential enrichment functions, and compares the methods in terms of computational cost and relative accuracy. Conclusions are then provided in Section 7.

2 Problem Formulation

Consider a domain $\Omega \subset \mathbf{R}^3$ 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

$$\rho c \frac{\partial u}{\partial t} = \nabla(\boldsymbol{\kappa} \nabla u) + Q \quad \text{in } \Omega \quad (1)$$

where $u(\mathbf{x}, t)$ is the temperature field, $\boldsymbol{\kappa}$ is the thermal conductivity tensor, ρc is the volumetric heat capacity, and $Q(\mathbf{x}, t)$ is the internal heat source.

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

$$u = \bar{u} \quad \text{on } \Gamma^u \quad (2)$$

$$-\boldsymbol{\kappa} \nabla u \cdot \mathbf{n} = \bar{f} \quad \text{on } \Gamma^f \quad (3)$$

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

The initial conditions must also be satisfied

$$u(\mathbf{x}, 0) = u^0(\mathbf{x}) \quad \text{at } t^0 \quad (4)$$

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

3 Model Problem

The problem selected for analysis is taken from [32], and it involves a sharp spatial gradient in the temperature field (5), as well as in the resulting source term (6). There is also a temporal gradient, but it

is smooth in nature. This model problem is selected because it is similar to the type of severe, localized thermal loading which can be experienced by hypersonic flight vehicles.

$$u(x,t) = \left(\exp^{-\gamma(x-x_{front}(t))^2} + \sin\left(\frac{\pi x}{L}\right) \right) * \exp^{(-t)} \quad (5)$$

$$Q(x,t) = \rho c \frac{\partial u}{\partial t}(x,t) - \kappa \frac{\partial^2 u}{\partial x^2}(x,t), \quad (6)$$

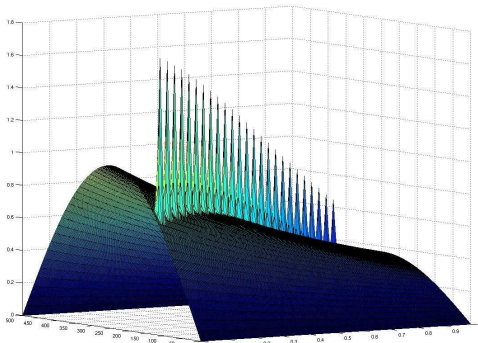
$$x_{front}(t) = x_0 + Vt \quad (7)$$

The initial and boundary conditions are given in (8) and (9), respectively.

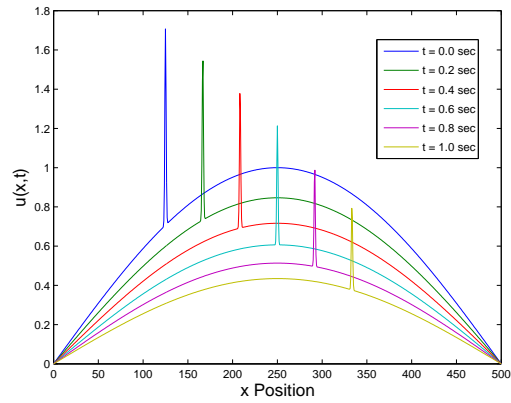
$$u(x,0) = \exp^{-\gamma(x-x_0)^2} + \sin\left(\frac{\pi x}{L}\right), \quad (8)$$

$$u(0,t) = u(L,t) = 0, \quad (9)$$

In the above equations, $x_0 = 125 \text{ mm}$, $L = 500 \text{ mm}$, $V = 250 \frac{\text{mm}}{\text{sec}}$ and γ is a parameter controlling the roughness of the solution. Unless otherwise indicated, the value of γ 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) is plotted in Figure 1 and the initial condition (8) is plotted in Figure 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 1: Shows the temperature field described in (5).

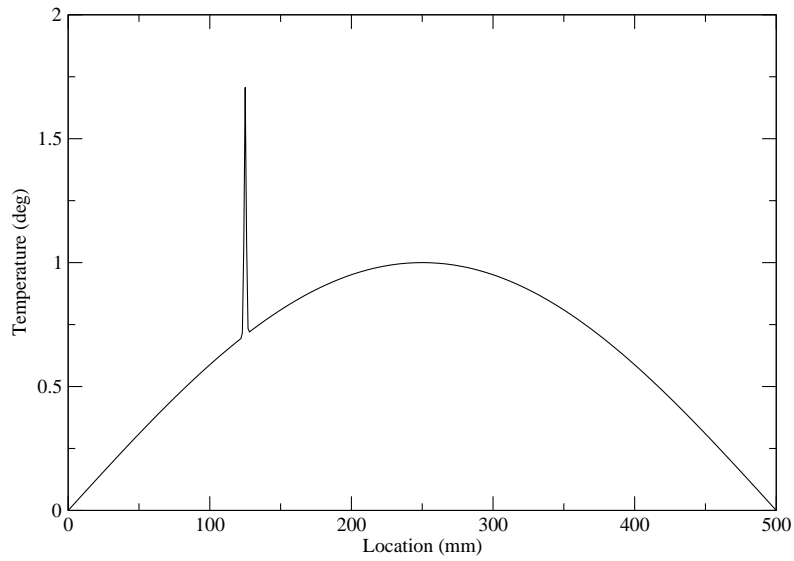


Figure 2: Initial conditions as described in (8).

4 Generalized FEM Approximations

The generalized FEM [1, 2, 14, 36, 44] is an instance of the partition of unity method, which has its origins in the works of Babuška et al. [1, 2, 29] and Duarte and Oden [12, 16–18, 36]. Many meshfree methods recently proposed can also be viewed as special cases of the partition of unity method.

The key feature of these methods is the use of a partition of unity (POU), which is a set of functions whose values sum to the one at any point in a domain. Additional methods based on the partition of unity concept are, for example, [11, 24, 33, 43].

In the GFEM, a discretization space is built using the concept of a partition of unity along with local spaces that are generated based on a-priori knowledge about the solution of a problem. A GFEM shape function, $\phi_{\alpha i}$, is computed as the product of a linear finite element shape function, φ_{α} , and an enrichment function, $L_{\alpha i}$,

$$\phi_{\alpha i}(\mathbf{x}) = \varphi_{\alpha}(\mathbf{x})L_{\alpha i}(\mathbf{x}) \quad (\text{no summation on } \alpha), \quad (10)$$

where α is a node in the finite element mesh. The linear finite element shape functions φ_{α} , $\alpha = 1, \dots, N$, in a finite element mesh with N nodes constitute a partition of unity, i.e., $\sum_{\alpha=1}^N \varphi_{\alpha}(\mathbf{x}) = 1$ for all \mathbf{x} in a domain Ω covered by the finite element mesh. Figure 3 illustrates the construction of a GFEM shape function.

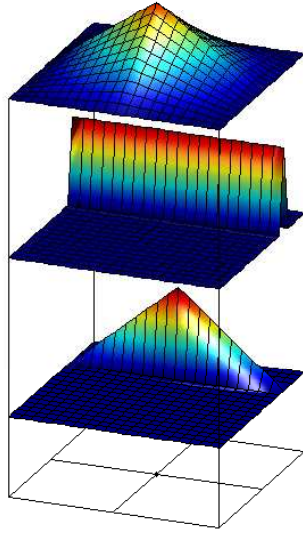


Figure 3: Construction of a generalized FEM shape function using a non-polynomial enrichment. Here, φ_α is the function at the top, the enrichment function, $L_{\alpha i}$, is the function in the middle, and the generalized FE shape function, $\phi_{\alpha i}$, is shown at the bottom.

Enrichment functions The GFEM has been successfully applied to the simulation of boundary layers [13], dynamic propagating fractures [15], line singularities [14], acoustic problems with high wave number [3], polycrystalline microstructures [41], porous materials [44], etc. These applications have all relied on closed-form enrichment functions that are known to approximate well the underlying physics of the problem investigated. These special enrichment functions are able to provide more accurate and robust simulations than the polynomial functions traditionally used in the standard FEM, while at the same time relaxing some meshing requirements of the FEM.

In this particular work we use knowledge of the solution to select our special enrichment function (11). The special, exponential enrichment functions are only applied to the nodes whose supports contain the thermal spike. All other nodes have only polynomial enrichments. The GFEM framework is what allows us to use (11) as our enrichment function, and we note the enrichment function is time dependent.

$$L_{\alpha i}(x, t) = \exp^{-\gamma(x-x_{front}(t))^2} \quad (11)$$

where $x_{front}(t) = x_0 + Vt$. The variables x_0 and V are defined as in Section 3. It should be again noted that the location of the exponential spike in the enrichment function is known beforehand from the analytical solution. If this was not the case, one would need to use information from the gradients in the loading function, or from a global error indicator in order to properly center the exponential enrichment function; both of which are beyond the scope of this paper.

5 Discrete Equations and Time-Integration

In this paper, we use the transient Formulation 2, as derived in detail in [38], with the time-stepping parameter $\alpha = 0.0$, yielding the conditionally stable, Forward Euler algorithm. Substituting $\alpha = 0.0$ into (33) from [38] leads to the following linear system of equations

$$\frac{1}{\Delta t} \mathbf{M}^{n+1} \mathbf{u}^{n+1} = \left[\frac{1}{\Delta t} \mathbf{M}^{n+1,n} - \mathbf{K}^{n+1,n} \right] \mathbf{u}^n + \mathbf{f}^{n+1,n} \quad (12)$$

where

$$\mathbf{M}^{n+1} = \int_L \rho c \boldsymbol{\phi}^{n+1} (\boldsymbol{\phi}^{n+1})^T dx \quad (13)$$

$$\mathbf{M}^{n+1,n} = \int_L \rho c \boldsymbol{\phi}^{n+1} (\boldsymbol{\phi}^n)^T dx \quad (14)$$

$$\mathbf{K}^{n+1,n} = \int_L \nabla \boldsymbol{\phi}^{n+1} \kappa (\nabla \boldsymbol{\phi}^n)^T dx \quad (15)$$

$$\mathbf{f}^{n+1,n} = \int_L Q^n \boldsymbol{\phi}^{n+1} dx \quad (16)$$

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

5.1 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 (Δt) such that stability is maintained throughout the simulation. We solve the generalized eigenvalue problem arising from the global system of equations, as posed in (17) to determine the magnitude of the dominant eigenvalue, λ_{max} . The critical time-step size (Δt_{cr}) is then related to the dominant

eigenvalue as in (18). More discussion of this stability criterion can be found in [40].

$$(\mathbf{K} - \lambda \mathbf{M}) \mathbf{x} = 0 \quad (17)$$

$$\Delta t_{cr} = \frac{2}{\lambda_{max}} \quad (18)$$

In (17), when time-dependent shape functions are used, $\mathbf{M} = \mathbf{M}^{n+1}$ as defined in (13) and $\mathbf{K} = \mathbf{K}^{n+1}$, defined as

$$\mathbf{K}^{n+1} = \int_L \nabla \phi^{n+1} \kappa (\nabla \phi^{n+1})^T dx \quad (19)$$

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

6 GFEM Analysis of Model Problem

6.1 Calculation of Critical Time-Step Sizes for Stable Simulations

In this section, (17) is solved for a series of uniform meshes, with different element sizes and polynomial orders. Table 1 summarizes the results of the generalized eigenvalue problem resulting from each discretization. The results for each type of element are investigated in more detail in subsequent sections. For the elements containing exponential enrichments (11), the eigenvalues need to be calculated at multiple time-steps due to the changing discretization, and then the most stringent Δt_{cr} is selected to yield a stable simulation. The elements with exponential enrichments will henceforth be referred to as exponential elements. In the table, Linear Exponential refers to elements with only a linear shape function, and an exponential enrichment; whereas Quadratic Exponential refers to elements with both a quadratic, as well as an exponential enrichment. As a result, a Linear Exponential element has only a linear and an exponential shape function; whereas a Quadratic Exponential element has linear, quadratic and exponential shape functions. It may be noted that the value of Δt_{cr} is very similar for both the Linear and Quadratic Exponential elements as the magnitude of the lead eigenvalue λ_{max} is essentially governed by the presence of the sharp, exponential enrichment function. The calculation of the critical time-steps for each discretization used is performed separately from the simulations run in subsequent sections. As

such, CPU times reported in the following sections do not reflect the time required for the calculation of the Δt_{cr} .

Table 1: Summary of Output for Calculation of Δt_{cr} .

<i>Shape Function Type</i>	<i>p-order</i>	<i>h_x (width)</i>	<i>λ_{max}</i>	<i>Δt_{cr}</i>
Linear	1	1.250	196910	1.02e-5
Linear	1	0.625	787402	2.54e-6
Polynomial	2	2.500	246130	8.13e-6
Polynomial	2	1.250	985222	2.03e-6
Polynomial	2	0.625	3938400	5.08e-7
Polynomial	4	5.000	389910	5.13e-6
Polynomial	4	2.500	1559300	1.28e-6
Polynomial	4	1.250	6242700	3.20e-7
Linear Exponential	-	100	128205	1.56e-5
Quadratic Exponential	-	100	125786	1.59e-5

6.2 Analysis of Model Problem Using Polynomial Enrichment Functions

Linear Elements In this section, we further investigate the results obtained for linear elements. In this, and subsequent sections, the accuracy of the solution, as well as CPU time required to produce the solution are examined. 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 is appropriate that the CPU Times recorded are reflective of the minor drawback.

For the plots provided dealing 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 (20). 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 (21) where the summation is performed over each time-step along the curve.

$$U_n = \int_{\Omega} (\kappa \nabla u^n) \cdot (\nabla u^n) d\Omega \quad (20)$$

$$L_2^{Error} = \sqrt{\frac{\sum \{ (U^{exact} - U^{FE})_n^2 \}}{\sum \{ (U^{exact})_n^2 \}}} \quad (21)$$

The output for linear elements is summarized in Table 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 4. As can be seen, there is a lot of noise in the solution due to its 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 where a least squares fit is used to smooth out the noise in the data.

Table 2: Summary of Output for Linear Elements.

Δt	Stability	h_x (width)	NumberTimeSteps	L_2^{Error}	CPU Time
1.015e-5	Yes	1.250	98500	0.3231	1.842e5
1.03e-5	No	1.250	97087	-	-
2.53e-6	Yes	0.625	396000	0.0909	1.611e6
2.56e-6	No	0.625	390000	-	-

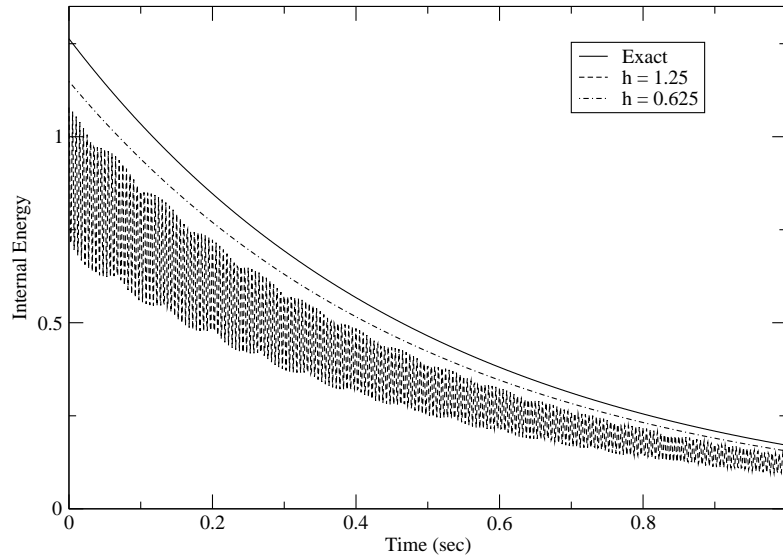


Figure 4: Internal Energy as a function of time for linear elements.

High Order Polynomial Elements This section contains output for simulations run using higher order polynomial elements. Elements with quadratic ($p = 2$) and quartic ($p = 4$) are selected for analysis. Due to the excessive number of time steps required for stability, only a percentage of the entire simulation is

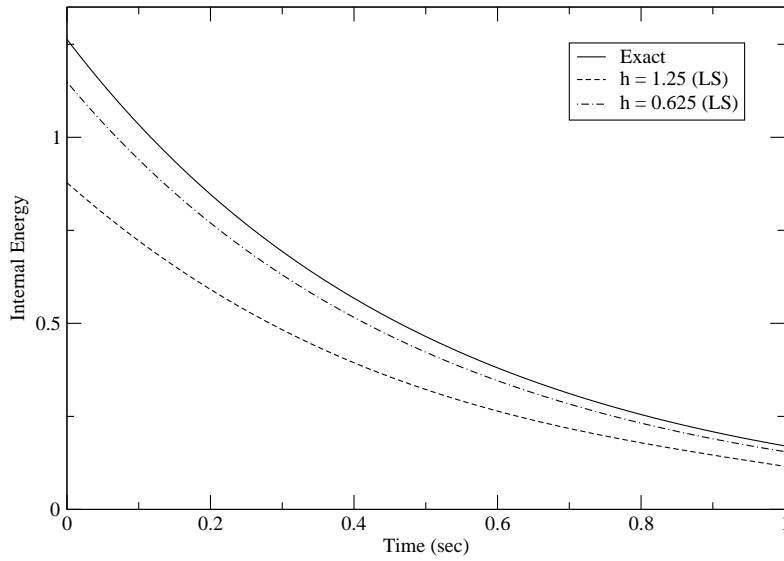


Figure 5: Internal Energy as a function of time for linear elements, using Least Squares to smooth out data.

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, then 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 L_2^{Error} values are valid for comparison, Table 3 shows the values obtained for linear elements when only certain percentages of the simulation are considered. From this table it is apparent that the L_2^{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 shape functions corresponding to the entire simulation.

Table 3: Comparison of L_2^{Error} for different t_f . (Linear elements)

t_f	h_x (width)	L_2^{Error}
0.25	1.25	0.3228
0.50	1.25	0.3230
1.00	1.25	0.3231

Tables 4 and 5 summarize the data for quadratic and quartic elements, respectively. The internal energy is then plotted as a function of time in Figures 6 and 8. 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 7 and 9 have plots of the data for which a least squares fit has been used in order to smooth out the data.

Table 4: Summary of Output for Quadratic Elements.

Δt	Stability	h_x (width)	NumberTimeSteps	L_2^{Error}	CPU Time (projected)
8.10e-6	Yes	2.50	123400	0.2851	1.213e5
8.15e-6	No	2.50	122700	-	-
1.99e-6	Yes	1.25	502000	0.0437	1.237e6
2.05e-6	No	1.25	487800	-	-

Table 5: Summary of Output for Quartic Elements.

Δt	Stability	h_x (width)	NumberTimeSteps	L_2^{Error}	CPU Time (projected)
5.10e-6	Yes	5.00	196000	0.2978	2.049e6
5.15e-6	No	5.00	194000	-	-

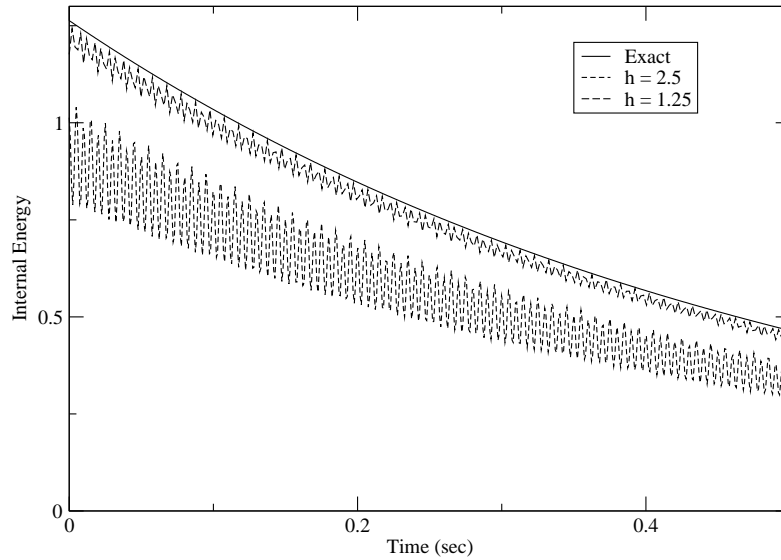


Figure 6: Internal energy as a function of time for quadratic elements.

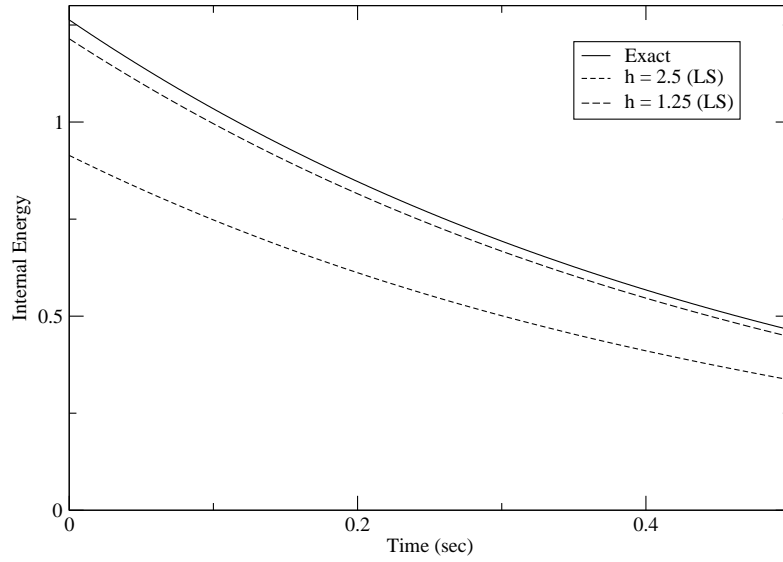


Figure 7: Internal energy as a function of time for quadratic elements, using least squares to smooth out data.

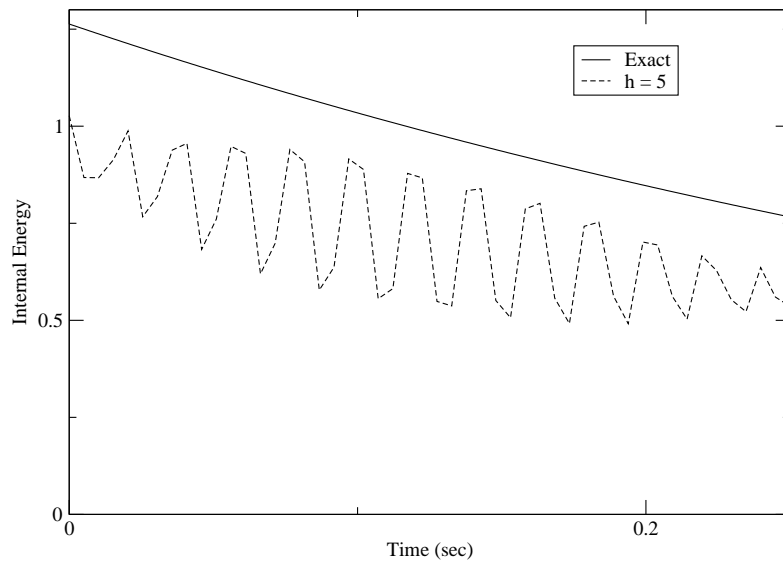


Figure 8: Internal energy as a function of time for quartic elements.

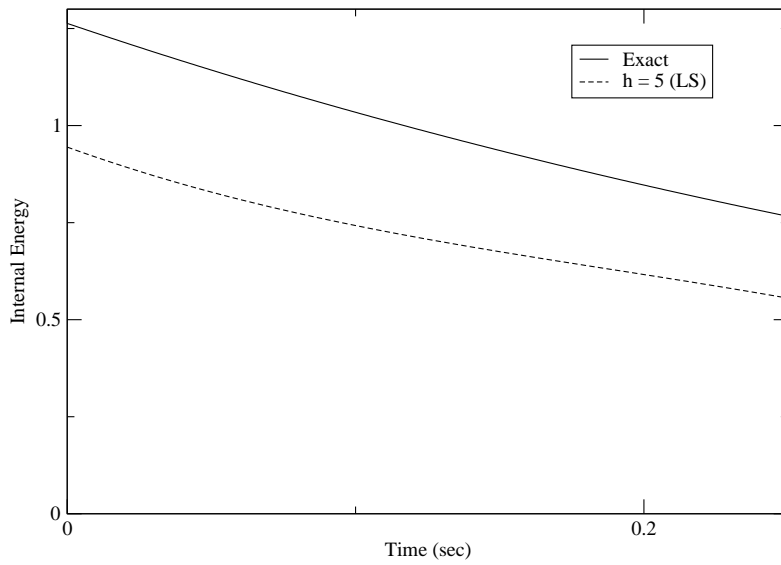


Figure 9: Internal energy as a function of time for quartic elements, using least squares to smooth out data.

6.3 Analysis of Model Problem Using Special Enrichment Functions

This section contains output for simulations for elements enriched with exponential functions (11). Table 6 summarizes the data for each run. The internal energy is then plotted as a function of time in Figure 10. Again, 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 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 10. 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 curves, Figure 11 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 6: Summary of Output for Elements Enriched with Exponential Functions.

Δt	Stability	h_x (width)	NumberTimeSteps	L_2^{Error}	CPU Time
Linear Exponential					
1.54e-5	Yes	100	65000	2.90e-4	2.756e4
1.62e-5	No	100	61800	-	-
Quadratic Exponential					
1.54e-5	Yes	100	65000	1.92e-4	2.789e4
1.62e-5	No	100	61800	-	-

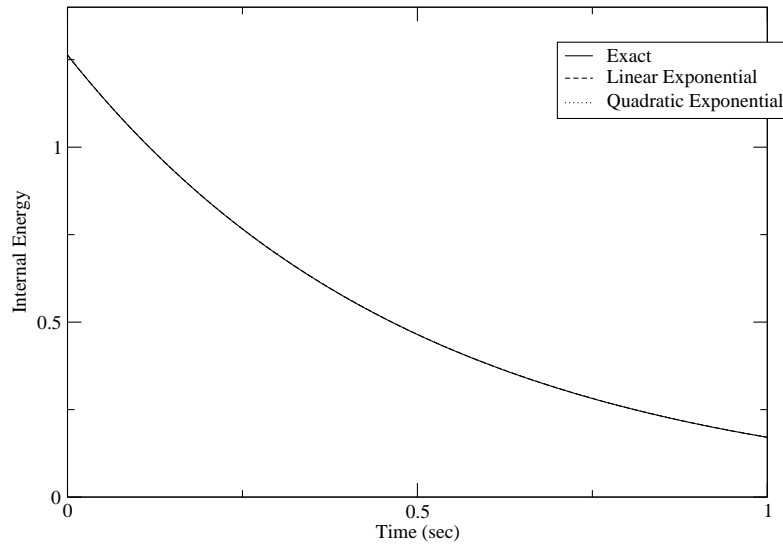


Figure 10: Internal energy as a function of time for exponential elements.

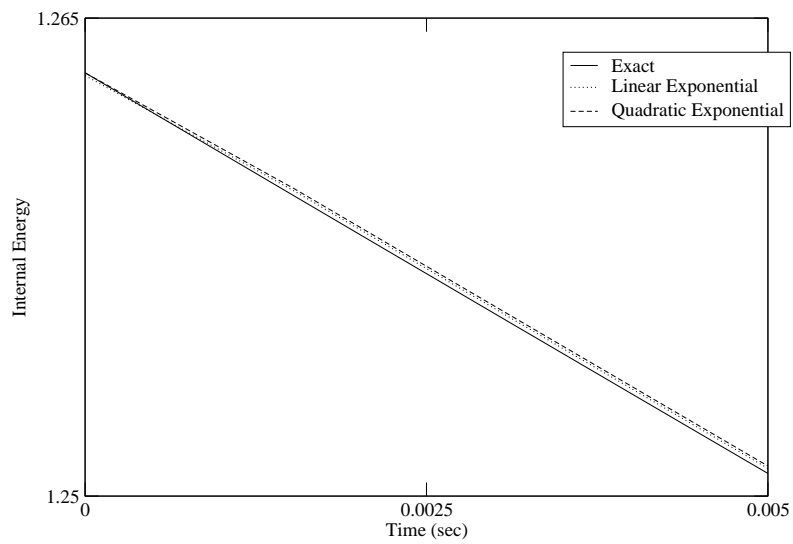


Figure 11: Internal energy as a function of time for exponential elements. Zoomed in to see a difference in the curves.

Due to the fact that the spike is moving throughout the course of the simulation, the Δt_{cr} may be different at each time-step, depending on the location of the spike with respect to a given node. It is noted in [30, 31], 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 edge of an elemental support, again yielding an infinitely small time-step requirement to maintain stability [30, 31]. For the present application, the enrichment functions used do not have this property as the spike location nears the edge of an elemental 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 7 shows the value for Δt_{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 7: Effect of Spike Location on Δt_{cr} .

<i>Distance</i>	Δt_{cr}	$\frac{\Delta t_{cr}^{max}}{\Delta t_{cr}}$
0	7.01e-5	1
$\frac{h_x}{2}$	2.60e-5	2.70
$\approx h_x$	1.56e-5	4.50

6.4 Comparison of Polynomial and Exponential Enrichments

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 Δt_{cr} produced from the generalized eigenvalue problem: for each discretization analyzed, the values of Δt_{cr} proved to be very reliable.

For the sake of an easy comparison, the pertinent data collected is summarized in Table 8. 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. From comparing the data in the table, it is quite clear that the exponential enrichments offer superior performance when compared to polynomial

enrichments. The exponential enrichments offer extreme reductions in both the error values, as well as the CPU time required to generate the data. The exponential enrichments 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 with a parabolic PDE, the exponential enrichments do in fact offer far superior behavior in regards to: time-step size required for stability, accuracy of solution, and CPU Time required.

Table 8: Summary of Simulation Data.

<i>Shape Function Type</i>	h_x (width)	L_2^{Error}	<i>CPU Time</i>
Linear	1.250	0.3231	1.842e5
Linear	0.625	0.0909	1.611e6
Quadratic	2.50	0.2851	1.213e5
Quadratic	1.25	0.0437	1.237e6
Quartic	5.00	0.2978	2.049e6
Linear Exponential	100	2.90e-4	2.756e4
Quadratic Exponential	100	1.92e-4	2.789e4

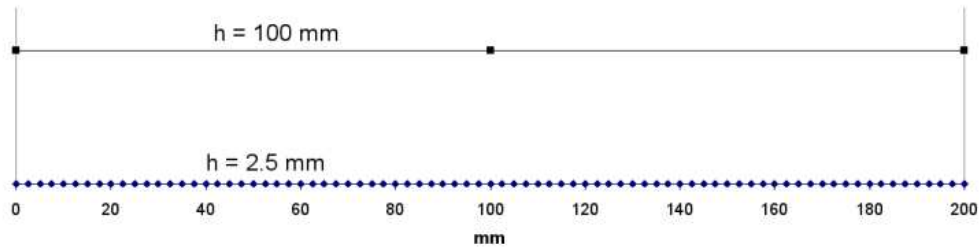


Figure 12: Comparison of element sizes for exponential and polynomial discretizations.

6.5 Effect of Volumetric Heat Capacity Magnitude

In this section we investigate the effect of the magnitude of the volumetric heat capacity, ρc . We analyze a problem very similar to the model problem, but in this instance the spike remains stationary, as indicated by the reference solution

$$u(x, t) = \left(\exp^{-\gamma(x-x_0)^2} + \sin\left(\frac{\pi x}{L}\right) \right) * \exp\left(\frac{-t}{t_{final}}\right) \quad 0 < x < L \quad (22)$$

where again, the internal source is derived as

$$Q(x, t) = \rho c \frac{du}{dt}(x, t) - \kappa \frac{d^2u}{dx^2}(x, t), \quad (23)$$

The initial and boundary conditions are the same as those applied to the original model problem, (8) and (9). A plot of the reference solution is shown in Figure 13, 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 ρc , the critical time-steps become larger. As such, longer simulations are run, but the reference solution is now parameterized by t_{final} , so only one reference curve is required.

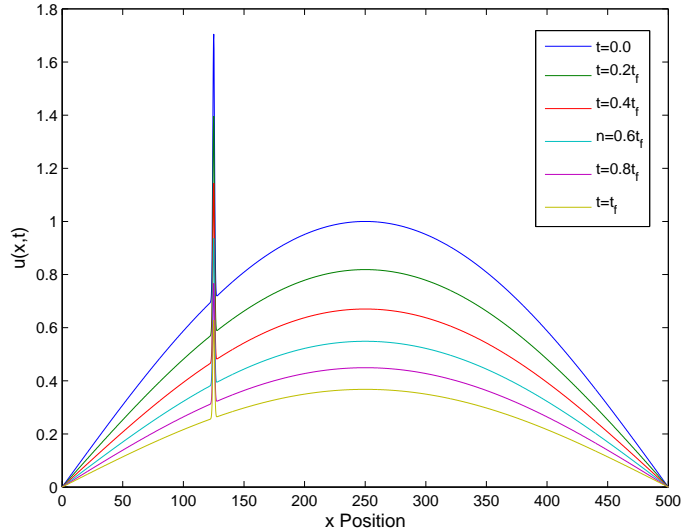


Figure 13: Reference solution for simulations involving larger values of ρc , described by (22).

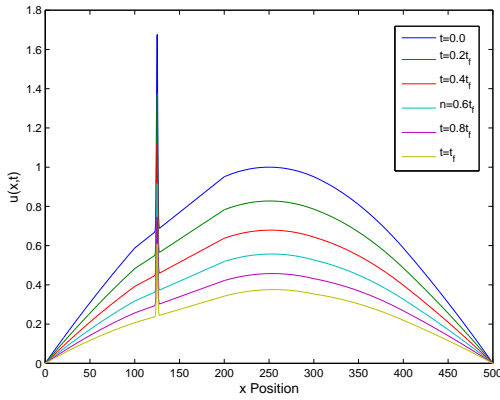
Simulation data is presented in Table 9 for simulations with and without the time-dependent exponential enrichment (24) applied to nodes which contain the thermal spike. For analyses utilizing (24), simulations are run with 110 time-steps, and elements of size $h_x = 100\text{ mm}$. Simulations run without (24) have 1400 time steps, and quadratic elements with $h_x = 1.25\text{ mm}$.

$$L_{\alpha i} = \left\{ 1, \exp^{-(x-x_0)^2} * \exp^{\frac{-t}{t_{final}}} \right\} \quad (24)$$

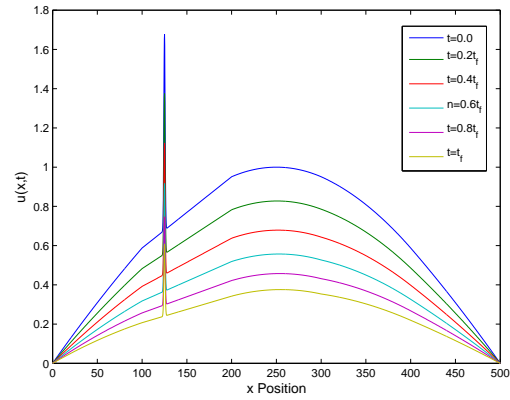
As can be seen from the table, the exponential enrichment functions once again deliver much higher levels of accuracy, with smaller requirements in terms of CPU time. Figure 14 shows time-slices of the solutions generated using the exponential enrichment, and as would be expected, the solutions generated match up very nicely with the reference solution shown in Figure 13. Figure 15 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.

Table 9: Simulation Data for Varying Magnitudes of ρc .

ρc	t_{final}	Δt_{cr}	Δt	L_2^{error}	CPU Time
Exponential Basis Functions					
0.1	7	0.0667	0.0636	3.63e-3	3.672
0.5	35	0.3333	0.3182	3.63e-3	3.701
1.0	70	0.6667	0.6364	3.63e-3	3.764
5.0	350	3.3333	3.1818	3.64e-3	3.731
10	700	6.6667	6.3636	3.67e-3	3.814
Polynomial Basis Functions					
0.1	7	0.0052	0.0050	0.0678	1.03e3
0.5	35	0.0260	0.0250	0.0678	1.06e3
1.0	70	0.0521	0.0500	0.0678	9.87e2
5.0	350	0.2604	0.2500	0.0679	1.01e3
10	700	0.5208	0.5000	0.0681	1.03e3

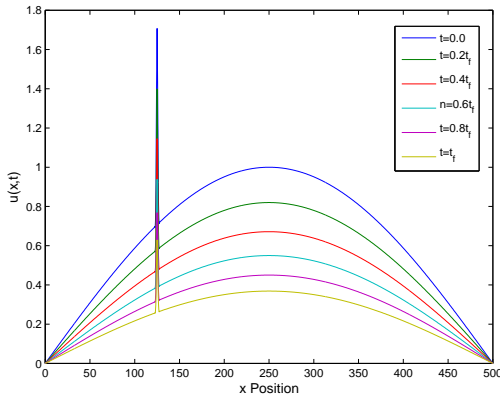


(a) $\rho c = 0.5$

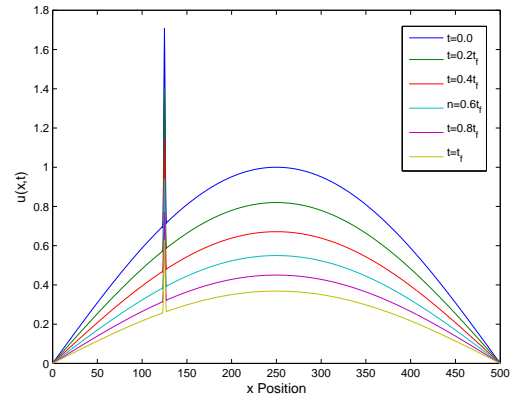


(b) $\rho c = 5$

Figure 14: Simulation results generated using the enrichment in (24).



(a) $\rho c = 0.5$



(b) $\rho c = 5$

Figure 15: Simulation results generated using only quadratic, polynomial shape functions.

7 Conclusions

In this work we investigate the potential gains in computational efficiency that the GFEM with special enrichments can provide for transient simulations due to its ability to produce accurate results on coarse meshes. The alleviation in mesh density leads to less stringent stability requirements on the time-step size. The end result is that fewer time-steps are required for a simulation, and the system of equations is also smaller at each time-step. As such, the CPU time requirements are greatly reduced when compared to simulations run with the use of standard finite elements. In the parabolic case, the use of the exponential enrichment functions led to not only more efficient simulations, but also to more accurate simulations as well, since the enrichment was selected specifically to capture the fine-scale portion of the solution.

The results presented here suggest that if the proper enrichment functions are available, larger time-step sizes may be used without negatively impacting the accuracy of the results. However, since this is not the case for most problems of interest involving multi-scale or non-linear phenomena, we are currently developing numerically generated enrichment functions, as a transient extension of [37].

Acknowledgments: The authors gratefully acknowledge the contributions of the Midwest Structural Sciences Center (MSSC) at the University of Illinois at Urbana-Champaign. The Center is supported by the U.S. Air Force Research Laboratory Air Vehicles Directorate under contract number FA8650-06-2-3620.

Appendix A. Discrete Equations

A brief summary of the temporal and spatial discretization processes is presented here. For more details, and a more in-depth discussion of the discretization process the reader is referred to [38].

We start with the strong form of the heat equation

$$\rho c \frac{\partial u}{\partial t} = \nabla \cdot \kappa \nabla u + Q \quad (25)$$

Equation (25) is multiplied by a weighting function, w , and integrated over the domain, Ω , yielding the variational form

$$\int_{\Omega} w \rho c \frac{\partial u}{\partial t} d\Omega = \int_{\Omega} (w \nabla \cdot \kappa \nabla u + w Q) d\Omega \quad (26)$$

Integration by parts is used 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.

$$\int_{\Omega} \left(w \rho c \frac{\partial u}{\partial t} + \nabla w \cdot \kappa \nabla u \right) d\Omega = \int_{\partial\Omega} w \kappa \frac{\partial u}{\partial n} d\Gamma + \int_{\Omega} w Q d\Omega \quad (27)$$

The temporal discretization is performed first, and standard finite differencing is used, in the form of the α -method, in which the following approximations are used

$$\frac{\partial u}{\partial t} = \frac{u^{n+1} - u^n}{\Delta t} \quad (28)$$

$$u^{n+\alpha} = (1 - \alpha)u^n + \alpha u^{n+1} \quad (29)$$

Plugging (28) and (29) into (27) yields the temporally discretized equation

$$\begin{aligned} & \int_{\Omega} \left(w \rho c \frac{u^{n+1} - u^n}{\Delta t} + \nabla w \cdot \kappa [\alpha \nabla u^{n+1} + (1 - \alpha) \nabla u^n] \right) d\Omega \\ &= \int_{\Omega} w [\alpha Q^{n+1} + (1 - \alpha) Q^n] d\Omega + \int_{\partial\Omega} w \kappa \left[\alpha \frac{\partial u^{n+1}}{\partial n} + (1 - \alpha) \frac{\partial u^n}{\partial n} \right] d\Gamma \end{aligned} \quad (30)$$

Boundary terms must be expanded out, and then (30) can be rearranged 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.

$$\begin{aligned}
& \frac{1}{\Delta t} \int_{\Omega} w \rho c u^{n+1} d\Omega + \alpha \int_{\Omega} \nabla w \cdot \kappa \nabla u^{n+1} d\Omega = \frac{1}{\Delta t} \int_{\Omega} \rho c w u^n d\Omega - \\
& (1 - \alpha) \int_{\Omega} \nabla w \cdot \kappa \nabla u^n d\Omega + \alpha \int_{\Omega} w \mathcal{Q}^{n+1} d\Omega + (1 - \alpha) \int_{\Omega} w \mathcal{Q}^n d\Omega \\
& + \alpha \int_{\Gamma_f} w \bar{f}^{n+1} d\Gamma + (1 - \alpha) \int_{\Gamma_f} w \bar{f}^n d\Gamma
\end{aligned} \tag{31}$$

At this point, we have our system of equations fully discretized in time. Generalized finite element shape functions are used for the spatial discretization, with special care being taken for the discretization of w . The weighting function, w , is discretized using finite element shape functions at time t^{n+1} , as is discussed by Fries and Zilian [22].

With the spatial discretization performed, we can pose the fully discretized system of equations as

$$\begin{aligned}
& \left[\frac{1}{\Delta t} \mathbf{M}^{n+1} + \alpha \mathbf{K}^{n+1} \right] \mathbf{u}^{n+1} = \left[\frac{1}{\Delta t} \mathbf{M}^{n+1,n} - (1 - \alpha) \mathbf{K}^{n+1,n} \right] \mathbf{u}^n \\
& + \alpha \mathbf{f}_Q^{n+1} + (1 - \alpha) \mathbf{f}_Q^{n+1,n} + \alpha \mathbf{f}_N^{n+1} + (1 - \alpha) \mathbf{f}_N^{n+1,n}
\end{aligned} \tag{32}$$

Plugging in $\alpha = 0.0$, and dropping the terms related to applied fluxes (in this work the model problem is 1-D with Dirichlet boundary conditions at both ends), we obtain

$$\left[\frac{1}{\Delta t} \mathbf{M}^{n+1} \right] \mathbf{u}^{n+1} = \left[\frac{1}{\Delta t} \mathbf{M}^{n+1,n} - \mathbf{K}^{n+1,n} \right] \mathbf{u}^n + \mathbf{f}_Q^{n+1,n}$$

which is the same as (12), and where $(\cdot)^{(n+1,n)}$ indicates a quantity which requires information from time steps t^n and t^{n+1} .

References

- [1] I. Babuška and J.M. Melenk. The partition of unity finite element method. *International Journal for Numerical Methods in Engineering*, 40:727–758, 1997.
- [2] I. Babuška, G. Caloz, and J.E. Osborn. Special finite element methods for a class of second order elliptic problems with rough coefficients. *SIAM Journal on Numerical Analysis*, 31(4):945–981, 1994.
- [3] I. Babuška, F. Ihlenburg, E. Paik, and S. Sauter. A generalized finite element method for solving

- the Helmholtz equation in two dimensions with minimal pollution. *Computer Methods in Applied Mechanics and Engineering*, 128(3–4):325–360, 1995.
- [4] T. Belytschko and R. Mullen. Stability of explicit-implicit mesh partitions in time integration. *International Journal for Numerical Methods in Engineering*, 12(10):1575–1586, 1978.
- [5] T. Belytschko, H. J. Yen, and R. Mullen. Mixed methods for time integration. *Computer Methods in Applied Mechanics and Engineering*, 17/18:259–275, 1979.
- [6] S. Y. Chang. Explicit pseudodynamic algorithm with unconditional stability. *Journal of Engineering Mechanics*, 128:935–947, 2002.
- [7] S. Y. Chang. Enhanced, unconditionally stable, explicit pseudodynamic algorithm. *Journal of Engineering Mechanics*, 133:541–554, 2007.
- [8] S. Y. Chang. An explicit method with improved stability. *International Journal for Numerical Methods in Engineering*, 77(8):1100–1120, 2008.
- [9] H. K. Ching and J. K. Chen. Thermomechanical analysis of functionally graded composites under laser heating by the MLPG method. *Computer Modeling in Engineering and Sciences*, 13(3):199–217, 2006.
- [10] D. D’Ambrosio. Numerical prediction of laminar shock/shock interactions in hypersonic flow. *Journal of Spacecraft and Rockets*, 40(2):153–161, 2003.
- [11] S. De and K.J. Bathe. The method of finite spheres. *Computational Mechanics*, 25:329–345, 2000.
- [12] C.A. Duarte. *The hp Cloud Method*. PhD dissertation, The University of Texas at Austin, December 1996. Austin, TX, USA.
- [13] C.A. Duarte and I. Babuška. Mesh-independent directional p -enrichment using the generalized finite element method. *International Journal for Numerical Methods in Engineering*, 55(12):1477–1492, 2002. <http://dx.doi.org/10.1002/nme.557>.
- [14] C.A. Duarte, I. Babuška, and J.T. Oden. Generalized finite element methods for three dimensional structural mechanics problems. *Computers and Structures*, 77:215–232, 2000.
- [15] C.A. Duarte, O.N. Hamzeh, T.J. Liszka, and W.W. Tworzydło. A generalized finite element method for the simulation of three-dimensional dynamic crack propagation. *Computer Methods in Ap-*

- plied Mechanics and Engineering*, 190(15-17):2227–2262, 2001. [http://dx.doi.org/10.1016/S0045-7825\(00\)00233-4](http://dx.doi.org/10.1016/S0045-7825(00)00233-4).
- [16] C.A.M. Duarte and J.T. Oden. Hp clouds—A meshless method to solve boundary-value problems. Technical Report 95-05, TICAM, The University of Texas at Austin, May 1995.
- [17] C.A.M. Duarte and J.T. Oden. Hp clouds – An hp meshless method. *Numerical Methods for Partial Differential Equations*, 12:673–705, 1996.
- [18] C.A.M. Duarte and J.T. Oden. An hp adaptive method using clouds. *Computer Methods in Applied Mechanics and Engineering*, 139:237–262, 1996.
- [19] T. Elguedj, A. Gravouil, and H. Maigre. An explicit dynamics extended finite element method. part 1: Mass lumping for arbitrary enrichment functions. *Computer Methods in Applied Mechanics and Engineering*, 198:2297–2317, 2009.
- [20] P. P. Friedmann, K. G. Powell, J. J. Mcnamara, B. J. Thuruthimattam, and R. Bartels. Hypersonic aerothermoelastic studies for reusable launch vehicles, 2004. AIAA 2004-1590.
- [21] P. P. Friedmann, K. G. Powell, J. J. Mcnamara, B. J. Thuruthimattam, and R. Bartels. Three-dimensional aeroelastic and aerothermoelastic behavior in hypersonic flow, 2005. AIAA 2005-2175.
- [22] T. P. Fries and A. Zilian. On time integration in the xfem. *International Journal for Numerical Methods in Engineering*, 79:69–93, 2009.
- [23] A. Gravouil, T. Elguedj, and H. Maigre. An explicit dynamics extended finite element method. part 2: Element-by-element stable-explicit/explicit dynamic scheme. *Computer Methods in Applied Mechanics and Engineering*, 198:2318–2328, 2009.
- [24] M. Griebel and M.A. Schweitzer. A particle-partition of unity method for the solution of elliptic, parabolic and hyperbolic PDEs. *SIAM Journal on Scientific Computing*, 22(3):853–890, 2000.
- [25] T. J. R. Hughes and W. K. Liu. Implicit-explicit finite elements in transient analysis: implementation and numerical examples. *Journal of Applied Mechanics*, 45:375–378, 1978.
- [26] T. J. R. Hughes, K. S. Pister, and R. L. Taylor. Implicit-explicit finite elements in nonlinear transient analysis. *Computer Methods in Applied Mechanics and Engineering*, 17/18:159–182, 1979.

- [27] W. Li, X. Deng, and A. J. Rosakis. Determination of temperature field around a rapidly moving crack-tip in an elastic-plastic solid. *International Journal of Heat and Mass Transfer*, 39(4):677–690, 1996.
- [28] L. E. Lindgren. Numerical modeling of welding. *Computer Methods in Applied Mechanics and Engineering*, 195:6710–6736, 2006.
- [29] J.M. Melenk and I. Babuška. The partition of unity finite element method: Basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 139:289–314, 1996.
- [30] T. Menouillard, J. Rethore, A. Combescure, and H. Bung. Efficient explicit time stepping for the extended finite element method (x-fem). *International Journal for Numerical Methods in Engineering*, 68:911–939, 2006.
- [31] T. Menouillard, J. Rethore, A. Combescure, and H. Bung. Mass lumping strategies for x-fem explicit dynamics: Applications to crack propagation. *International Journal for Numerical Methods in Engineering*, 74:447–474, 2008.
- [32] R. Merle and J. Dolbow. Solving thermal and phase change problem with the eXtended finite element method. *Computational Mechanics*, 28:339–350, 2002.
- [33] N. Moës, J. Dolbow, and T. Belytschko. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering*, 46:131–150, 1999.
- [34] J. R. Moselle, A. R. Wieting, M.S. Holden, and C. Glass. Studies of aerothermal loads generated in regions of shock/shock interaction in hypersonic flow, 1988. AIAA-88-0477.
- [35] M. O. Neal and T. Belytschko. Explicit-explicit subcycling with non-integer time step ratios for structural dynamic systems. *Computers and Structures*, 31(6):871–880, 1989.
- [36] J.T. Oden, C.A. Duarte, and O.C. Zienkiewicz. A new cloud-based *hp* finite element method. *Computer Methods in Applied Mechanics and Engineering*, 153:117–126, 1998.
- [37] P. O’Hara, C.A. Duarte, and T. Eason. Generalized finite element analysis of three-dimensional heat transfer problems exhibiting sharp thermal gradients. *Computer Methods in Applied Mechanics and Engineering*, 198(21-26):1857–1871, 2009. <http://dx.doi.org/10.1016/j.cma.2008.12.024>.

- [38] P. O'Hara, C.A. Duarte, and T. Eason. Transient analysis of sharp thermal gradients using coarse finite element meshes. *Computer Methods in Applied Mechanics and Engineering*, 2010. Submitted for publication.
- [39] Patrick James O'Hara. Finite element analysis of three-dimensional heat transfer for problems involving sharp thermal gradients. Master's thesis, University of Illinois at Urbana-Champaign, 2007.
- [40] J. N. Reddy and D. K. Gartling. *The Finite Element Method in Heat Transfer and Fluid Dynamics*. CRC Press LLC, Boca Raton, Florida, 2001.
- [41] A. Simone, C.A. Duarte, and E. van der Giessen. A generalized finite element method for polycrystals with discontinuous grain boundaries. *International Journal for Numerical Methods in Engineering*, 67(8):1122–1145, 2006. <http://dx.doi.org/10.1002/nme.1658>.
- [42] M. L. Spearman. Lessons learned in the high-speed aerodynamic research programs of the NACA/NASA, 2005. AIAA 2005-0327.
- [43] T. Strouboulis, I. Babuška, and K. Copps. The design and analysis of the generalized finite element method. *Computer Methods in Applied Mechanics and Engineering*, 81(1–3):43–69, 2000.
- [44] T. Strouboulis, K. Copps, and I. Babuška. The generalized finite element method. *Computer Methods in Applied Mechanics and Engineering*, 190:4081–4193, 2001.
- [45] K. K. Tamma and K. C. Saw. Hierarchical p -version finite elements and adaptive *a posteriori* computational formulations for two-dimensional thermal analysis. *Computers and Structures*, 32(5):1183–1194, 1989.
- [46] E. A. Thornton, A. R. Wieting, and K. Morgan. Application of integrated fluid-thermal-structural analysis methods. *Journal of Thin-Walled Structures*, 11:1–23, 1991.
- [47] T. L. Turner and R. L. Ash. Analysis of the thermal environment and thermal response associated with thermal-acoustic testing, 1990. AIAA-90-0975-CP.
- [48] D. Y. Tzou. Fracture path emanating from a rapidly moving heat source - The effect of thermal shock waves under high rate response. *Engineering Fracture Mechanics*, 41(1):111–125, 1992.
- [49] A. R. Wieting. Experimental study of shock wave interference heating on a cylindrical leading edge, 1987. NASA Technical Memorandum 100484.

- [50] A. H. Yaghi, T. H. Hyde, A. A. Becker, and W. Sun. Finite element simulation of welding and residual stresses in a p91 steel pipe incorporating solid-state phase transformation and post-weld heat treatment. *Journal of Strain Analysis for Engineering Design*, 43:275–293, 2008.
- [51] G. Yu, K. Masubichi, T. Maekawa, and N. M. Patrikalakis. A finite element model for metal forming by laser line heating. Massachusetts Institute of Technology, Cambridge, MA.
- [52] G. Yu, K. Masubichi, T. Maekawa, and N. M. Patrikalakis. Fem simulation of laser forming of metal plates. *Journal of Manufacturing Science and Engineering*, 123:405–410, 2001.