Submitted to International Journal of Protective Structures

Application of Computational Modeling for Explosive Hazard Assessments

J. Keith Clutter

SciRisq, Houston TX, USA

Abstract: Computational modeling is routinely used in the design and development of explosive systems but is not typically applied in performing hazard assessments related to explosives. Typically simplified models are used to perform analysis related to safety and security applications involving explosives. These models can misrepresent the destructive output from explosive devices because they do not resolve many of the aspects of the scenarios such as wave reflection, channeling and focusing of blast waves. Here a computational model developed specifically for explosive hazard assessments is reviewed and the advantages as compared to simple models are highlighted. Some of the obstacle to applying computational modeling to routine explosive hazard assessment problems is the ability to perform analysis in a timely manner and address the wide range of explosives encountered. Here techniques being developed to increase flexibility in application and increase efficiency are presented.

Keywords: hazard assessment; CFD; explosives; reaction kinetics

1 Introduction

Computational modeling has been an essential tool in the development and evaluation of explosive systems. Applications have ranged from understanding the detonation phenomena of various types of energetic materials to designing warheads. There has been less use of modeling in addressing safety and hazard assessments related to explosive systems. The needs exist to be able to analyze scenarios such as the response of explosives in a fire, the consequences of explosives in storage accidentally functioning and even the consequence of malicious attacks using explosives. This paper discusses current modeling capabilities useful in analyzing some of these scenario and presents developments underway to address the others.

Some efforts have been made in transitioning the advantages of computational modeling to the safety and hazard analysis communities. One of the challenges is that the practitioners in this area are not experts in computational analysis and really do not have the option to become such. Those tasked to perform safety and hazard assessments have relied on simple quantity-distance (QD) models for years. Therefore the first step is to work with the community to introduce them to the advantages of computational modeling. The author has developed a code titled the Computational Explosion and Blast Assessment Model (CEBAM) that is currently used by analyst addressing safety and security issues. A brief introduction to the advantages of using the more sophisticated modeling approaches for explosive hazard assessments is provided here.

Though used to some level, there remains a challenge in making computational modeling more readily available to this community. To do this there is a need to address the areas that typically make applying such models to problems difficult. These areas include:

- Geometry definition
- Ability to represent full range of possible explosion processes
- Numerical Techniques to improve efficiency

This paper presents methods being developed that address these various areas.

The geometries involved are typically complex such as the interior of buildings or a collection of buildings that compose a facility. There are even incidences where a shape such as a vehicle or ship needs to be represented. This is very common when one is trying to analyze a scenario such as a terrorist attack. Most times there exists a definition of the geometry in the form of a CAD model. There are a variety of options when defining complex geometries for analysis. Here an approach using adapting Cartesian meshing is presented.

Those tasked to analyze explosive hazards have to address scenarios ranging from fires initiating the explosion to devices functioning as designed and detonating. In the case of the response of explosives to fires or sympathetic detonation, it is not known a priori what will occur. Therefore there is a need for intelligent algorithms available to the safety analyst that determines what will occur based on scenario conditions. This can be a challenge since the typical models used for the reactions in explosives either represent the deflagration event or the detonation event but do not represent both in a seamless manner. An approach to address this issue is presented here.

One of the obstacles to the wider use of computational modeling for hazard assessment is the complaint of the time required to perform simulations. Of course when applying such models, the required time is dependent on the equation set being solved, the mesh size being used and the numerical techniques being applied. Here the equation set size is at a

minimum those required to represent the fluid dynamics processes plus what is required to capture the reaction processes. Some discussion is presented here on how to represent the reactions with a minimum set of equations. However, the need to represent a wide range of reaction processes requires more equations than the single reaction used in many codes. Also shown is grid adaptation to minimize the mesh size while capturing the essential physics. Finally a numerical solution technique to minimize the computational time to solve the additional reaction equations is presented.

2 Current Computational Model for Hazard Assessment

To address explosive hazards for safety and security issues, various organizations in both the government and private sectors require explosion effects modeling. The ultimate goal is typically the determination of the effects on people and other resources if an accident or attack were to occur. This requires engineering analysis into the structural response and the blast (pressure) loading caused by the explosive event. The structural response can be analyzed with a variety of tools to include FEA methods which is commonly used. FEA methods are the most sophisticated approach and perform the analysis using a true 3D representation of the structure. A full set of partial differential equations which govern the dynamics of the structure are solved.

The other aspect of addressing these problems is the determination of the pressure loading on the structure caused by the explosive and drives the response of the structure. As in the case of analyzing the structural aspects, there is a range of tools that can be used. The most common approach uses simple blast curve data that gives the pressure and impulse as a function of explosive charge size and distance. Though many times this method is packaged into computer programs and may give the appearance of being sophisticated they do not incorporate the specifics of the actual event. The data on which the load is determined is essentially a charge setting in an open field. Some of the models that include the blast-curve based load prediction are ConWep, AT Planner, VAPO and BlastFX. Some of these incorporate algorithms that account for more complex scenarios but since they are based on empirical data and simplified analytical adjustments, many times they are not accurate.

An alternative approach is to use blast load modeling to match the sophistication of the FEA structural analysis typically performed. In fact one has to wonder about the merits of using sophisticated structural response analysis with such simplified load predictions. To match the quality of the FEA type analysis a hydrocode / computational fluid dynamics (CFD) approach to obtain the blast loading information is needed. In theory general purpose CFD codes such as Fluent and CFX can be used for blast modeling but some models have been developed specifically for this problem. These include the following: CTH, Epic, SHARC, SHAMRC, AUTODYN, AIR 3D, CEBAM

The majority of these are government codes and not available to the general public but can be used on government projects if approval is granted. CEBAM was developed specifically for use by the safety and security community for hazard assessments [1,2]. Here its capabilities in addressing high-explosive events are briefly reviewed. It can also be used for vapor cloud explosion (VCE) scenarios [3].

What is the advantage of using the more sophisticated blast load modeling? In short, you get the actual load the structural will be exposed to, not an over simplified estimate. The response of structural elements is dependent on the loading magnitude, duration and shape. All of these parameters are affected by the specifics of the event such as what surrounds the location of the explosive charge. For instance, an explosion that occurs in an urban setting does not behave like one initiated in an open field which is the scenario assumed in the blast-curve based modeling.

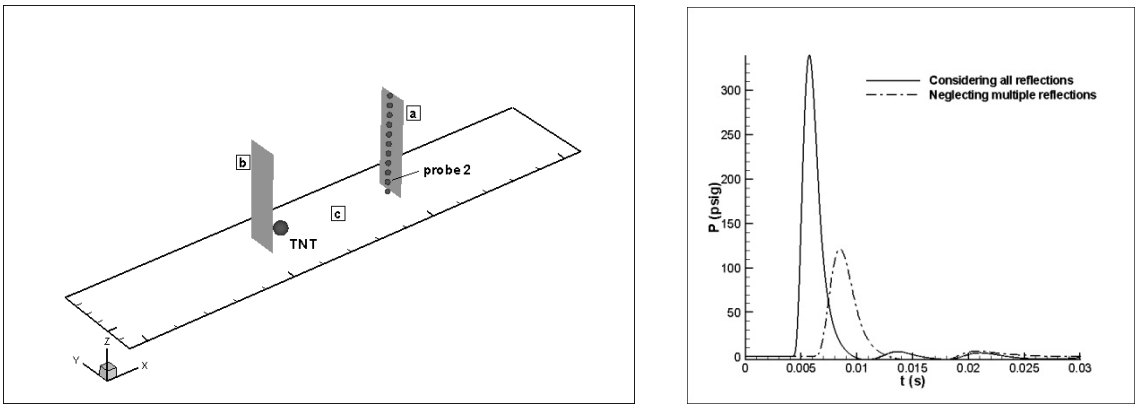

Figure 1. Effect of reflecting surfaces on the blast load magnitude.

Just the presence of a few reflecting surfaces can significantly increase the magnitude of the load. Figure 1 shows a simple example of the effect of reflecting surfaces. The scenario is taken from earlier studies evaluation the use of computational model in assessing explosive events in urban settings [4]. The pressure time histories as predicted using CEBAM are made at point 2 in the geometry. Walls “a” and “b” were 3 m wide and 10 m tall and spaced 14 m apart.

The charge was 300kg of TNT and located 2 m from wall “b” and 1 m from surface “c”. The comparison shows the effect of the presence of surfaces “b” and “c”. The environment can also dramatically affect the loading history. Figure 2 shows how the incident blast wave produced by a charge reflects off the buildings bordering a street and causing the load on the structures to be not a single pulse as assumed by the simple methods but a collection of pulses. This scenario has been used by code such as Air3D to benchmark their accuracy [5]. Therefore that data is shown along with results from CEBAM.

The configurations in Figure 1and Figure 2 are rather simple but are found to change the blast loading from the simple, single pulse assumed in the simple methods. These effects are even more exacerbated in complex situations such as the example in Figure 3. It shows that the load on a single building cannot be accurately predicted if surrounding objects are not included. For instance, the backside of buildings (the surface away from the blast) is typically considered to be exposed to a side-on, non-reflected load when using the simple methods. In fact, the side can be loaded much higher due to the channeling and focusing of blast off of neighboring buildings.

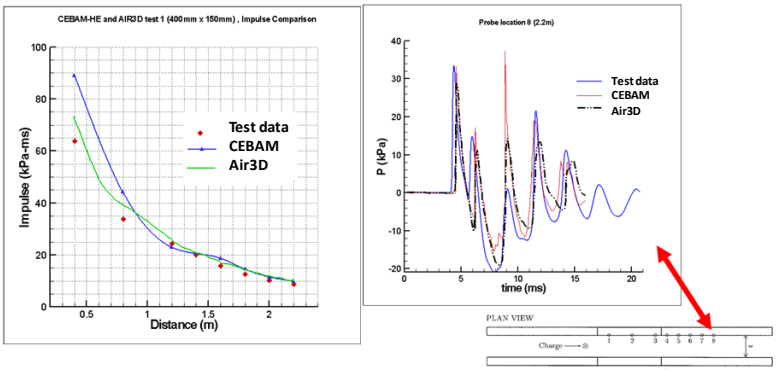

Figure 2. Blast channeling down a street between buildings.

Another phenomenon that occurs is the collision of the multiple blast fronts causing an increase in loading on areas of the structures that may be unexpected. The complex environment also changes the load produced on the faces of the structure. Instead multiple pulses are produced due to blast waves reflecting off the various objects. Such phenomenon is not captured by the simple methods.

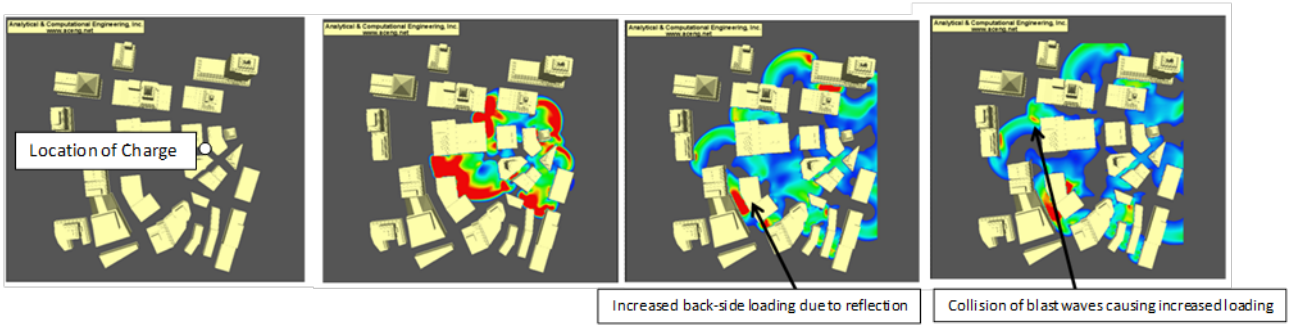

Figure 3. Example blast load analysis in an urban setting.

The brief examples here show the advantages of using computational modeling when analyzing explosive hazards that could occur in safety and security applications. CEBAM is one example of a code specifically developed for these applications. As in any field there remains the need for improvements. The next section discusses improvements underway to make the use of this type modeling more easily available to those needing to perform explosive hazard assessments.

3 Modeling Techniques for Hazard Assessments

Modeling tools such as CEBAM are useful in performing hazard assessments related to explosives. As noted they are required to capture much of the physics that influence the consequence on structures and people. However there remains the need to improve various aspects of the modeling. This is necessary to increase the use of the models by the community tasked to perform hazard assessments. Ease of use and efficiency are key requirements. Here developments that will produce such improvements are reviewed.

3.1 Grid Adaptation for Geometry Definition and Solution

When performing analysis for explosion safety and hazards the first requirement is the definition of the geometry involved. An example of the geometries that need to be represented is shown in Figure 4. There exist a variety of

meshing frameworks for use in computational modeling. The main classes are (1) body-fitted curvilinear meshing, (2) unstructured tetrahedral meshing, (3) Chimera techniques, and (4) Cartesian meshing and all have pros and cons. The body-fitted approach requires that the mesh be constructed around each body. In the case of the scenarios of interest here there are simply too many bodies with too many orientation possibilities for this to be a viable and efficient method.

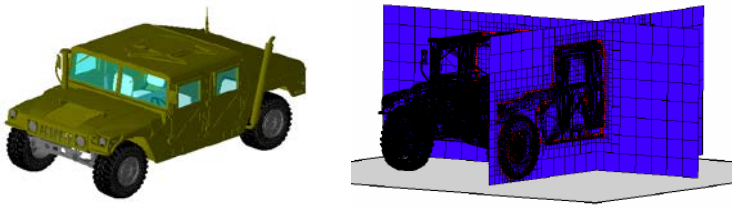

Figure 4. Example CAD model of a HUMVEE and the results of geometry definition using the adaptive Cartesian mesh.

The unstructured tetrahedral approach is a viable option and has been successfully used for problems such as airflow and pollutant in urban settings [6]. The construction of the mesh for the scenario of an explosion in a can be much more involved since the mesh construction begins with the surfaces of the geometry and for the problems of interest here, there are many and they cover a large range of sizes. The Cartesian meshing approach offers a very efficient option when building a scenario that contains several bodies such as in the scenarios of interest here. The baseline mesh is refined only where there exist surfaces of the geometry. The method employed here uses the basic principles found in previous work approaches [7,8,9].

Using the Cartesian approach, the domain of interest is defined as any rectangular shape. Also defined is the size of the coarsest grid desired which is referred to as the *Level 0* cells. The cells need not be cubic but larger aspect ratio cells can introduce numerical error depending on the particular flow solver used. Also defined is the number of adaptation levels desired, referred to here as *n levels*. The two parameters, *Level 0* cell size and *n levels* will influence both solution accuracy and efficiency.

The Cartesian mesh used here will be adapted as needed to define either geometries or solutions. The domain of interest is decomposed into a set of *Level 0* cells. If increased resolution is needed these cells are systematically divided into 8 children. The requirement for refinement is dependent on a local feature of interest in the geometry or solution. In the current implementation, the discontinuity between neighboring cells is never greater than a 2:1 ratio. This restriction does require that some additional cells be refined than just those meeting specified criteria. However, it simplifies the flow solver and difference equation solution algorithms and more than adequately accounts for any additional cost.

To facilitate the management of the computational mesh and the solution process, there are some basic grid accounting parameters that need to be associated with each cell. These include the following:

- level = level of the cell
- idiv = 0 if cell is not divided, = 1 if cell is divided
- Parent = pointer to a parent cell
- C2[i][j][k] = pointer to children cells: i=0,1; j=0,1; k=0,1
- xN = pointer to a neighboring cell: x=L(eft), R(ight), B(ottom), T(op), U(nder), O(ver)

A cell at level *n* is divided into 8 children at level *n+1*. The first three parameters are self-evident. The third parameter defines the children cells if any cell at any level is divide. The pointers are stored in a three-dimensional array making it easy to define algorithms such as flow solvers.

The adaptation of the mesh is first required to define the geometry of interest. The actual construction of the geometry is typically done in a CAD package and can be exported as a collection of triangular plates. Most of the common CAD programs allow such output. The adaptation to the geometry uses the plate information long with the defined grid domain, Level 0 cell size and the parameter n levels to construct a representation of the geometry within the Cartesian mesh.

The adaptation is based on the method developed by Akenine-Moller for graphics programming and the algorithm is derived from the separating axis theorem (SAT) [10]. The model cycles through the set of plates, refining only the cells which it intersects. The algorithm begins with the level 0 cells and whether the particular level 0 cell has been divided or not, it does not search lower if the plate does not intersect that cell. If the intersection check returns a *true* then the algorithm steps to the finer levels. If the cell has not been divided earlier, it is divided at that time. This recursive process continues until the volume the plate occupies has been divided to the finest level permitted. At that point, these finest of cells is defined to contain a geometric surface.

Shown in Figure 4 is the application of this process. Cut planes have been used to show how the structures are represented in the model. Those cells at the finest spacing that are colored red have been defined to contain a plate. The presence of these solid surfaces is represented in the computation by applying the correct boundary conditions.

In addition to refining the grid to represent the geometry, refinement is needed in areas of flow property gradients to ensure accuracy while maintaining efficiency. The two primary approaches to setting the criteria for adaptation are (1) using a measure of convergence of a solution or (2) using the local gradient information. Here the second method, as demonstrated in earlier studies [11], will be used to determine an adaptation criteria parameter labeled ε. This parameter will be set based on a series of tests evaluating whether the gradient in the flow parameters meets a set condition. The parameter assigned to the cell and used in the adaptation decision will be the largest of the set corresponding to the gradient between the cell and its six neighbors,

$$\varepsilon = \max\left(\varepsilon^L, \varepsilon^R, \varepsilon^B, \varepsilon^T, \varepsilon^U, \varepsilon^O\right) \tag{1}$$

where the parameter ε^N corresponds to the largest value as determined by each flow parameter.

The local pressure gradient will be evaluated using

$$\varepsilon_S^N = \left\{ \begin{array}{c} 1, \; if \; \dfrac{|P^N - P|}{\min(P^N, P)} > \xi_S \\ 0, \quad otherwise \end{array} \right\} \tag{2}$$

where P is the pressure in the cell being evaluated and P^N is the pressure in the neighboring cells. ξ_S is a set value that is used to determine if a cells needs to be refined or can be coarsened. For instance, if ξ_S is set to .1 then the cell will be refined until the maximum relative increase in pressure between the cell under evaluation and any neighbor is not greater than 10%. The refinement can be set as a function of other parameters but since the interest here is on blast loadings pressure will be used. Figure 5 shows a simple example of a steady-state solution using this adaptation method.

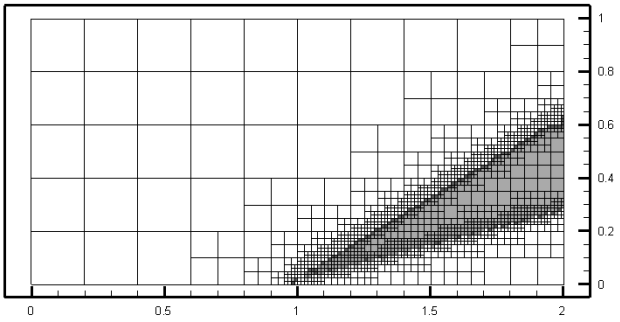

Figure 5. Solution of a Mach 3 ramp flow problem using the adaptation method with 5 levels of refinement and a gradient criterion of 10%.

For the steady-state problem it is rather straightforward to refine the mesh in the vicinity of gradients. However, the main objective of this current work is the solution of problems involving unsteady wave motion (i.e. blast waves from explosions). For the cases involving moving fronts, if refinement is restricted only to the immediate area where the front sets then refinement will have to occur frequently to ensure adequate adaptation is always present. If not, critical results such as the maximum blast pressure produced during the explosion will be under predicted. The requirement may be that adaptation be performed essentially at every time step. This can introduce excessive computational time and counter the goal of the adaptation in the first place. Here an alternative approach is suggested.

A method is proposed that will map out the current location where key fronts are located and project their motion. This additional area will be refined along with the current location of the front. Such an approach will require that the grid be refined only at discrete times during the time integration. By balancing the mapping and the interval between adaptation then both increased accuracy and efficiency can be achieved.

To develop an algorithm to map out areas in the vicinity of the fronts for refinement, the adaptation criteria already discussed is used. However it is treated as a domain property similar to temperature and a conduction-diffusion equation is solved to perpetuate the property through the domain. At the intervals when grid refinement occurs the conduction-diffusion equation is integrated in pseudo time, off-line from the time integration of the governing equations of the flow problem.

This is similar to the method used in earlier work [11]. However, in this earlier work a reaction-diffusion equation was employed. This was tested in the current study and it was found that the mapping was directly dependent on the time over which the reaction-diffusion equation was integrated. A steady-state solution did not exist. That is why in the current study this equation was replaced with the convection-diffusion equation which will converge to a steady-state solution. This makes the mapping independent of the time over which the integration is performed. Therefore the perpetuation of the adaptation property will depend solely on the nature of the flow field and the constants used in the equations.

The approach to map out regions around fronts in unsteady problems that need adaptation begins with the same process using in the steady-state situation. Those regions where the gradient warrants that the adaptation parameter, ε, be set to a value of 1 are defined. These cells are maintained at a ε = 1 state. This parameter is then treated as a property and perpetuated through the domain using the governing equation

$$\frac{\partial \varepsilon}{\partial t} = K\nabla^2\varepsilon - h(\varepsilon - \varepsilon_o) \tag{3}$$

patterned after the heat conduction-diffusion equation. Here, ε_o is set to 0. To present the use of this approach to map out the regions near fronts in need of refinement, a simple discontinuity in pressure and within a straight duct was used. Then the parameters K and h were adjusted to determine the characteristics of the mapping procedure. What was of interest was the size of the region over which ε perpetuated. Recall those cells where ε is greater than some specified value will be refined. All others will be coarsened. The governing partial differential equation is integrated using a finite volume approach with explicit time integration.

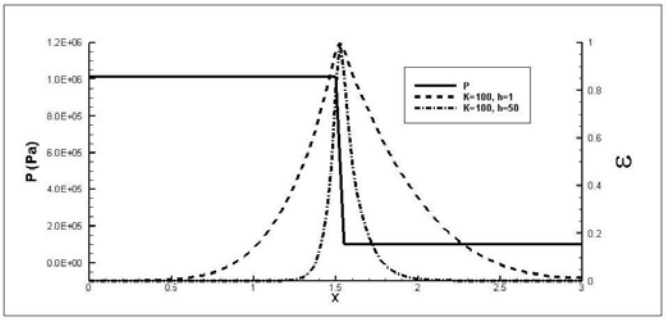

Figure 6. Steady state mapping of ε for h=1 and 50 when K=100.

Figure 6 shows the effect of the parameters K and h on the mapping of the parameter ε in a 1-D configuration with a discontinuity in pressure. There is some tradeoff between the time required to map the parameter and the size of the region defined for refinement with more time required for the parameter settings that maps a larger region. Because the interest is in unsteady problems, the region refined needs to be large enough to accommodate the movement of the fronts over the interval between when adaptation occurs. The setting of K and h will be evaluated using a representative vehicle explosion problem. Note the region over which the property ε spreads is dependent on the ratio between K and h and not necessarily the numerical values.

The adaptive meshing approach pays dividends in making the definition of geometries a much easier process for the user. It also helps to reduce the required time to run simulations by using refined mesh cells only where needed. To demonstrate the use of the method for explosion analysis in a facility, consider the scenario in Figure 7. Also shown is the pressure field produced by the explosion after it has progressed past the buildings as modeled using an unadapted mesh with spacing of 1 meter in all directions.

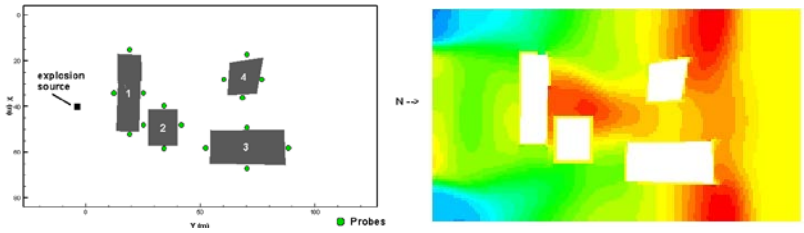

Figure 7. Explosion scenario involving a collection of buildings.

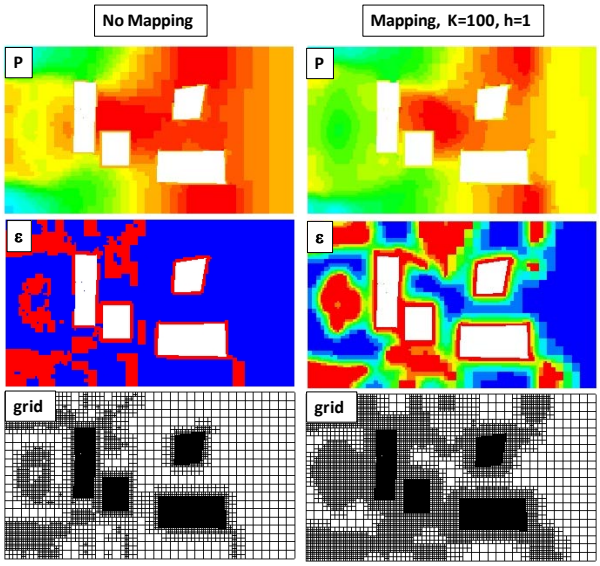

Figure 8. Comparison of solution using the adaptive method with and without mapping the criteria function.

The scenario is now simulated using the adaptive method and results are shown in Figure 8. The left set of images shows the predicted pressure field as well as the refinement parameter and the mesh if the mapping to introduce more cells in front of the blast is not used. The right set of images shows the same information when mapping is used. In this case K=100 and h=1. The resulting pressure field is much closer to that predicted when a fine mesh is used everywhere. This can be contributed to the fact that if additions cells are not introduced in front of the waves causing pressure discontinuity there is numerical dissipation in the solution.

3.2 Reaction Mechanism

Addressing explosive hazards may require the consideration of a wide range of scenarios. Therefore any tools applied to this area most accurately resolve the response of the energetic component to thermal (i.e. cook-off scenario) or mechanical (i.e. impact) insult. Traditionally hydrocodes have been successfully used in aiding in the development and evaluation of munitions. However when representing the energetic component of the munitions the codes use material models that resolve the response of the material to shock loading but not response to thermal insult. What is needed is a material response model that is applicable to both mechanical and thermal insult.

The difficulty in simulating the various scenarios that need to be represented to assess explosive hazards is that the response of the energetic material to mechanical insult such as impact or detonation is fundamentally different from its response to thermal insult. The modeling tools used to date for energetic materials focus on the performance of the system when functioned as designed. This means they are formulated to represent the detonation process. The most common are the prescriptive methods that assume a detonation wave already exists and originates at the point of ignition or impact. One example of such an approach is the Detonation Shock Dynamics (DSD) model [12]. These do not have the basic fundamentals incorporated to be able to be applied to the thermal initiation scenario.

Another class of models use pressure based kinetics modeling to represent how impacts to explosives can initiate combustion which will drive the growth of a detonation wave. An example of this approach is the Ignition and Growth model [13,14]. This type model has the potential of representing the impact initiated response of energetic materials important in some scenarios but cannot address the thermal response spectrum of the problems. This requires a more fundamental representation of the chemical processes.

Temperature based kinetic schemes have been shown to accurately represent the combustion process in a cook-off type scenario [15]. This type model employs a multi-step, temperature based model such as

$$\begin{aligned} A \rightarrow B, \quad & r_1 = Z_1 \exp(-E_1 / RT)\rho_A \\ B \rightarrow C, \quad & r_2 = Z_2 \exp(-E_2 / RT)\rho_B \\ C \rightarrow D, \quad & r_3 = Z_3 \exp(-E_3 / RT)\rho_C^2 \end{aligned} \tag{4}$$

where A and B are solids and C and D are product gases. R is the Universal Gas Constant, Z_i the pre-exponential factor, and E_i is the activation energy. This model has been successfully used to simulate cook-off type test [15]. Figure 9 shows how the model correctly predicts the time to explosion. Shown are results from the original study as well as results using our current simulation code.

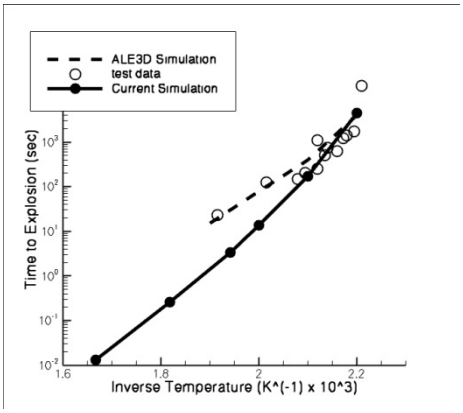

Figure 9. Prediction of time to explosion using the kinetics based model. ALE3D results are from Ref. [15] and Simulation results are from the current model.

In the past when the kinetics based model has been used it was only active up to the point where a detonation formed. From then on a prescription type model is used. This is because the kinetics based model in its basic form cannot reproduce a detonation. Figure 10 how the 3-step kinetics mechanism performs for the case where a detonation wave forms in RDX after being impacted with a flyer plate. When compared to models developed specifically for the detonation scenario it clearly does not perform satisfactory. However it should be recalled that these models cannot represent the thermal insult scenarios. The ideal situation for explosive hazard assessments would be a unified model that covers the complete spectrum of possible responses of the energetic material to all types of insult. And furthermore the best case would be if switching between the response models required no prior knowledge of how the material will response and therefore would be a true prediction tool.

To formulate a model for the response of the energetic material to both thermal and mechanical insult it is clear that the model must include the temperature based kinetics approach needed to simulate scenarios such as cook-off. The issue is how to extend this type model to include accurate performance for the detonation regime. In earlier work we have demonstrated how a temperature based kinetics scheme can also accurately reproduce the detonation scenario [16]. The solution can be achieved by looking at the reaction model at a more fundamental level.

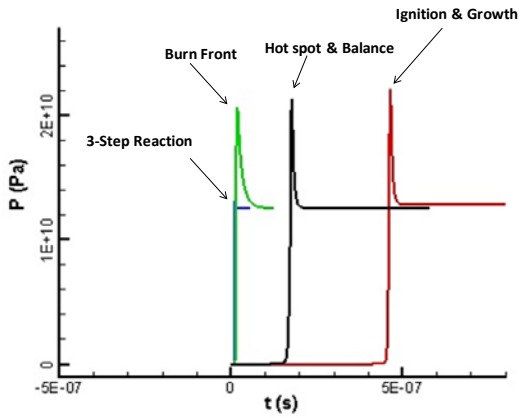

Figure 10. Performance of the 3-step kinetics model for a detonation case. Shown for comparison is results from three models developed specifically for the detonation scenario.

Reaction models such as the 3-step shown here respond to different initial conditions such as pressure and temperature. In the case of thermal insult these conditions would be set by heat transferring through the explosive bed. In the case of a mechanical insult it would be elevated pressure behind a shock. It is beneficial to evaluate the "path" the reaction takes following different initial conditions. Based on our earlier work it was found that to be able to reproduce the full spectrum of responses alternate "paths" for the reaction to take have to be present. To test this approach for the current RDX model an additional step was added of the form

$$A \rightarrow D, \quad r_4 = Z_4 \exp(-E_4 / RT)\rho_A \tag{5}$$

This uses temperature as a trigger and as shown in Figure 11 when temperature rises at a point the alternative path for the original material (A) to transform into the final product (D).

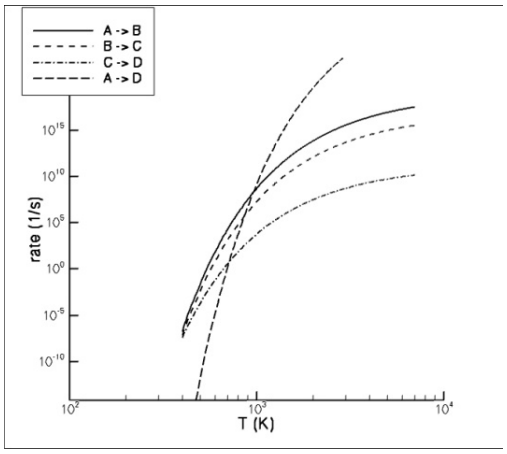

Figure 11. Response of each reaction step to temperature.

Figure 12 shows how both the original model and the one with the additional step perform for a thermal insult scenario when the initial temperature is 600K. For this case both give a time to explosion value of about .12 seconds and the variation of species with time are identical as is the pressure and temperature. Figure 12 shows data in a reaction-space manner as well. Plotting the intermediate species (B and C) versus the final specie (D) shows that both models dictate the same path (for example the C – 3 Step and C – 4 Step coincide).

When the initial temperature is raised to 1000K the models produce the response shown in Figure 13. The original model produces a time to explosion of 8×10^{-8} seconds but the 4-step model causes the reaction to take a different path and the time to explosion is orders of magnitude smaller and only takes two integration steps. Therefore the time history data is not shown in Figure 13 for this scheme. This difference in response to the insult is the type needed to accurately represent the complete spectrum that will be seen in safety and hazard assessment scenarios. This model is currently being tested.

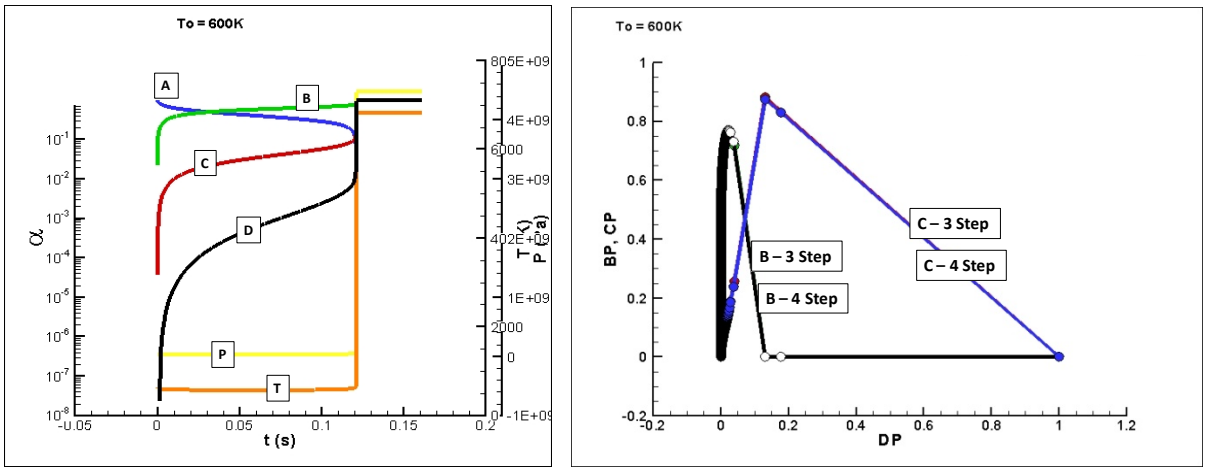

Figure 12. Performance of original 3-step and modified 4-step model to initial temperature loading of 600K.

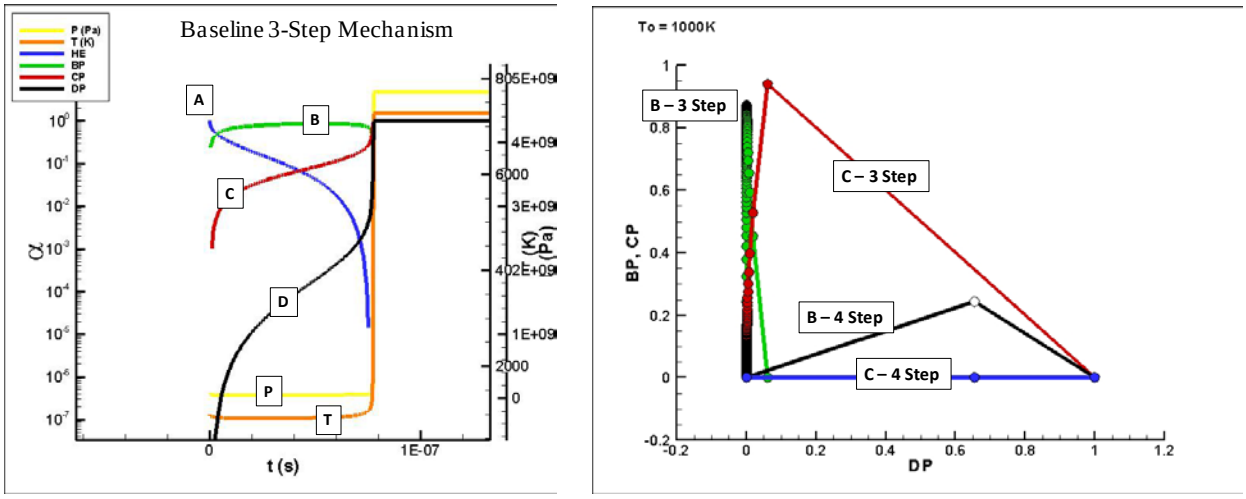

Figure 13. Performance of the two models to initial temperature loading of 1000K.

3.3 Numerical Solution Techniques

It has been shown that to perform hazard assessments for the full spectrum of possible explosive safety incidences there is a need to use multi-step, rate-based reaction models. This can increase the effort in terms of developing the models and increase the computational analysis times. Current methods used to solve reaction equation systems can have issues with the extensive computational time needed to compute the chemical kinetics. When working with explosive systems, computational complexity arises because the various processes involved are characterized on a wide range of temporal and spatial scales. This range difference, in most cases, is many orders of magnitude. This range of scales is manifested as "stiffness" in the sets of differential equations. Because of this, many industry models have been reduced to using complex combustion systems only in simple flow problems.

Table 1. Seven-step Hydrogen-Oxygen Reaction Mechanism.

#	Reaction	Z_n	B_n	E_n / R
1	$H_2 + O_2 = 2OH$	1.70E+13	0	24233
2	$H + O_2 = OH + O$	1.42E+14	0	8254
3	$H_2 + OH = H + H_2O$	3.16E+07	1.8	1525
4	$O + H_2 = OH + H$	2.07E+14	0	6920
5	$2OH = H_2O + O$	5.50E+13	0	3523
6	$H + OH + M = H_2O + M$	2.21E+22	-2	0
7	$2H + M = H_2 + M$	6.53E+17	-1	0

To counter these inherent extensive calculations, there is a lot of effort in the area of mechanism reduction methods, which reduce the number of reactions in the mechanism to a few primary steps. This is typically done in gas phase combustion mechanisms where a full scheme may involve hundreds of steps. For the explosive scenario here the mechanism is only on the order of a few steps and it has been shown that these steps are needed to represent all possible scenarios. Therefore we are pursuing other methods to increase efficiency when applying computational modeling to explosive hazard analysis. Here the utility of using numerical derivatives in solving the governing equations for the explosion scenario is evaluated. The proposed approach is compared to solution methods commonly used in the simulation tools for reactive flows. To do this the reaction scheme in Table 1 is used. This was selected because it has a similar number of steps as what is expected to be required for the explosive analysis problems.
The reaction process within the explosion event can be represented in the vector form

$$\frac{dQ}{dt} = \Omega \tag{6}$$

where only the dependent variables and their source terms remain in the equation. The terms Q and Ω represent vectors containing density and terms specific to each species represented in a mechanism. The vector of dependent variables and source terms are

$$Q = [\rho, \rho\alpha_1, ..., \rho\alpha_{NS-1}], \qquad \Omega = [0, \omega_1, ..., \omega_{NS-1}] \tag{7}$$

where α_i is the mass fraction of the ith species, and NS is the total number of species in the mechanism and ρ is the total density. The source terms are only represented to the NS-1 species since overall mass conservation is maintained. The

source terms represent the production rates of each individual species in the mechanism and incorporate both the forward and backward production rates of each species throughout all of the reactions of the mechanism in the form

$$\sum_{j=1}^{NS} \nu_{ij}^{'} X_j \underset{k_{bi}}{\overset{k_{fi}}{\Leftrightarrow}} \sum_{j=1}^{NS} \nu_{ij}^{''} X_j \quad , \quad i = 1, \ldots, NR \tag{8}$$

where $\nu_{ij}^{'}$ and $\nu_{ij}^{''}$ are the stoichiometric coefficients for species j in reaction I and X_j is the molecular concentration of species j. NR is the number of reactions and NS the number of species. The forward and backward reaction rates are represented in the terms $k_{f\,i}$ and $k_{b\,i}$ for each reaction. The forward reaction rates are calculated using the Arrhenius rate expression

$$k_{fi} = Z_i T^{b_i} \exp\left[\frac{-E_i}{RT}\right] \tag{9}$$

where the constants are as presented earlier with the addition of b_i which is the temperature exponent. The values used here can be found in Table 1. The backward rates are calculated using the relationship between the equilibrium constant and the forward and backward rates. The production rates of each individual species are calculated through a summation of contributions from each reaction. More details are provided in reference [17] but are not presented here in the interest of space.

The governing equations result in a system of equations that must be integrated in time. When coupled with the fluid dynamic terms the time step is typically set by the CFL condition associated with the flow. The restrictions from the fluid aspects of the problem are typically less restrictive than the reaction aspects. A common approach is the use of a split operator method that allows different solutions methods to be used for the fluid dynamic operation versus the reaction operations [18]. This allows the overall time step to be set based on parameters of the study. If it is set such that the restrictions associated with the fluid dynamics will define the time step between state n and n+1, then the reaction operator is used to advance the chemical reactions from one state to the next. This approach is used in CEBAM and has proven to allow for explosion scenarios seen in safety and hazard assessments to be performed efficiently. The focus here is on numerical methods used for the reaction operator.

The reaction system of equations can be solved in a variety of means and one of the simplest options is the use of explicit methods. Explicit schemes allow for the dependent variables to be directly computed in terms of known quantities. In the reaction model, the partial derivative of the dependent variables, Q, with respect to time can be found using a first order upwind difference operator [19]. Although explicit schemes are simple, they have very low stability and can easily produce non-physical solutions. Because of this, careful attention must be paid when choosing a step size.

Another option is the use of a Runge-Kutta scheme, which will increase the size of time step that can be taken compared to the generic explicit method. This approach is an attempt to benefit from the simplicity of explicit methods while decreasing computational time. For the reaction process, if a Fourth-Order Runge-Kutta is used then the source terms in the governing equations must be solved four times for each time step. Thus, the value at the next time step is found using the present value plus a weighted average of the slopes at several points.

Another option is to solve the set of equations using an implicit approach. Such an approach helps address issues such as stability in the solution process and numerical restrictions associated with the time steps. The difference equations that are actually solved come from the linearization of the chemical reaction source term, which produces a Jacobian matrix ($d\Omega/dQ$) and requires the simultaneous solution of the set of equations

$$\left[I - A^n \frac{\Delta t}{2}\right]\left(Q^{n+1} - Q^n\right) = \Omega^n \Delta t \quad . \tag{10}$$

There are options on how to solve the equation set and here the LU Decomposition method will be used. Another option is the formulation of the Jacobian matrix. This choice itself can affect the computational time required to perform a simulation. One approach that can greatly reduce run times is to determine the Jacobian analytically [18]. However, this approach requires preprocessing using other software tools rather than just the tool to be used for the reaction simulation. This is not inline with the objective to increase the ease of use of computational modeling for hazard assessments. This can be overcome by using a method that establishes the Jacobian is a generic approach where the computational routine uses a set of *do* loops with the parameters being set by an input file. Such an approach does not require extensive preprocessing before executing a simulating but only reading a simple file that defines the mechanism such as Table 1. This approach will be one of the methods evaluated here.

Algorithms utilizing *do* loops can be computationally expensive because of the many looping functions. To overcome this limitation, an alternative approach using numerical derivatives will be evaluated. By calculating the Jacobian numerically, only a single looping function is required as opposed to several embedded functions used in the general implicit approach. Studies which compared the use of numerical derivatives and analytic representation of the Jacobian terms in the full Navier-Stokes equations, found that numerical derivatives can be used to accurately represent the terms of the Jacobian[20]. Assuming that one has the function f(x), the definition of its derivative is

$$\frac{df}{dx} = f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \quad . \tag{11}$$

Using the form above, the derivatives of each source term that composes the Jacobian can be represented as

$$\frac{d\Omega_i}{dQ_j} = \frac{\Omega\left[Q(j+h)\right]_i - \Omega\left[Q(j)\right]_i}{h} \tag{12}$$

where i and j represent each species in the mechanism and *h* a deviation in the concentration of species *j*. Because the source term at state n has to be calculated for any method used to solve the reaction equations, the numerical derivative approach requires only an additional calculation, which can be done within a single looping function.

The numerical derivatives contain two sources of error, truncation error found in any difference equation and roundoff error. The roundoff error is inherent in any computer model due to machine accuracy. If the size of *h* is outside the precision of the computer or programming environment, the machine's floating-point representation may not accurately represent the effect of *h*. The worst case of this is where the value of *h* is lost in rounding and the resulting derivative becomes zero or undefined.

The potential advantage of using numerical derivatives is assessed by solving a reaction represented by the scheme in Table 1. Explosions are initiated by sudden exposure of an energetic material in elevated temperature and pressure. Here an explosive mixture of hydrogen and air are exposed to an initial temperature of 1600K and a pressure of 1125kPa. Each scheme was run using a constant time step of 1E-9 seconds. This small value was chosen to ensure stability of all methods. Figure 14 shows plots of temperature versus time for each method. The temperature histories produced by the general explicit, Runge-Kutta, and numerical derivative schemes are almost identical but the history of the general implicit scheme differs. Although the general implicit method deviates, it reaches the same steady state conditions as the explicit methods and the numerical derivative method. The deviation of the generic implicit method can be attributed to round off error accumulated in the calculations which has been found in other studies[20].

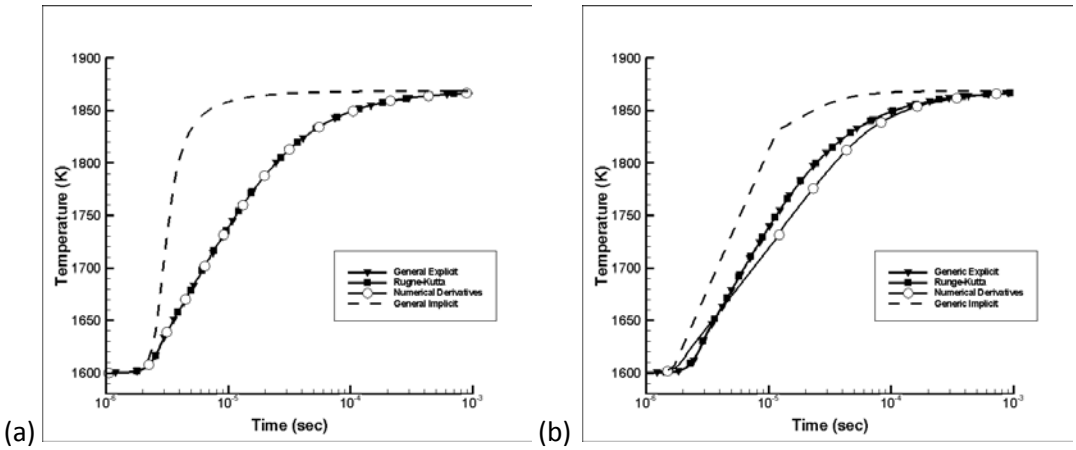

Figure 14. Comparison of temperature history results for each numerical scheme (a) using a constant time step of dt=1e-9 seconds, and (b) using a variable time step.

In this first set of results, the governing equations were integrated in time using a constant time step. This allows for the quantification of the computational requirements of each method. Table 2 shows the required run time for each method. All simulations were conducted on the same single processor, 3.0GHz desktop computer. The explicit method requiring the least run-time for each mechanism and the Runge-Kutta method also proved to be extremely efficient in the constant time step. This is due to the fact that no matrix operations are required. However, when using the explicit methods small time steps must be taken to maintain stability. Another possible occurrence is that the solution can produce non-physical solutions such as having mass fractions less than zero.

Table 2. Computational times (in seconds) to integration reaction to completion for the different solution schemes.

Scheme	Constant Δt	Variable Δt
Explicit	335.00	77.00
Runge-Kutta	625.00	1464.00
Numerical Derivatives	687.00	3.000
Generic Implicit	3845.00	4.000

Another notable result in Table 2 is the times for the generic implicit and numerical derivative methods. The generic implicit method required the greatest amount of run time to model the reactions due to the number of looping functions required to calculate the Jacobian terms. An earlier study shows that the time saved by evaluating the Jacobian terms numerically grows as the mechanism becomes more complex [x17]. The numerical derivatives are, on average, approximately nine times more efficient than the generic implicit solution when run with constant time steps.

The true improvement and efficiency of the different schemes comes from the use of variable time steps. Unless the time step in the simulation is limited to resolve a process of interest, the desire is to integrate the governing equations through the time period as rapidly as possible. Therefore in the following simulations each scheme was allowed to set its Δt based on stability requirements and to keep the solution out of non-physical regions. A maximum allowable value for Δt was set to be 1E-5 seconds. Each reaction mechanism was again modeled with initial conditions of T=1600K and P=1125kPa and out to a time of one millisecond.

Figure 14 shows temperature histories for each scheme and mechanism using a variable time step. In order to retain accuracy the explicit methods had to be forced to take smaller time steps at the beginning of the calculations. The explicit results using a variable time step matches the results when the constant time step is employed. The numerical derivatives and generic implicit method have shapes similar to that of their constant time step solutions with the numerical derivative approach again more closely tracking the explicit results. This is due in part to the fact that the implicit method using the numerical derivatives was found to take smaller time steps initially. The generic implicit time history contains fewer points, thus it gives a rougher solution than the numerical derivatives method. This could easily be corrected by further restricting the time step of the generic implicit scheme to better capture the profile

The run time results for each method when using a variable time step to solve the various mechanisms are shown in Table 2. The overall run times are reduced due to the fact that the schemes are allowed to take the maximum time step they can and maintain a stable and physically possible solution. Employing the numerical derivatives is found to reduce the computational time and in other studies the advantage improves as the mechanism complexity increases [x17]. The advantage of using an implicit method is clear and can be attributed directly to the increase stability allowing for larger integration time steps. It should be noted that some of the advantage would be diminished if the maximum allowable time step were reduced out of necessity to resolve important intermediate stages in the reaction process.

4 Conclusions

Though computational modeling is used in many areas when addressing explosive systems, they have not been used extensively in performing hazard assessments for these devices. This is due in part to the fact that where the engineering R&D community handles the design and evaluation of explosive systems, hazard assessments for safety and security are routinely managed by a separate group of individuals. Though they may have engineering experience their experience with computational modeling is typically limited and routinely simplified modeling techniques are used.

It has been shown here that these models, which typically only consider the quantity of explosives and distance from items of interest, can under predict the consequence from potential events. This is highly undesirable when considering the fact that hazard assessments are being done to protect property and people from potential events. The major obstacle to the broader use of computational modeling for these applications is that the practitioners do not have the resources or time to use classic hydrocode or CFD codes. There are models available that have been specifically developed for hazard assessment applications such as CEBAM presented here. The introduction of such models is helping to increase the use of the better technology for hazard assessments.

Even with the introduction of tools such as CEBAM there remains the need to continue to tailor models for use in explosive hazard assessments. Needed are models that provide intelligence to the user in terms of determining the explosive event such as whether a deflagration or detonation will occur. One example of a unified reaction model that will resolve either cook-off or detonation events was presented. Such model will be valuable in helping guide the hazard analysis.

Equally important are techniques that increase efficiency when setting up scenarios and performing analysis. Here the implementation of an adaptive meshing method that makes it easier to define problems involving complex geometries has been presented. It streamlines the process of importing CAD models and allows for hazard assessments to be

conducted in much less time. This same adaptation technique has been shown to reduce the required computational time by refining the mesh only where needed. This refinement requires no inputs from the user, which meets an objective of providing intelligent algorithms to those performing safety and security analysis. Likewise a numerical technique has been presented that more efficiently solves the more complex kinetics schemes which are needed to represent the wide range of scenarios. All of these techniques will increase efficiency.

Performing explosive hazard assessments to address safety and security issues directly impact that resilience of properties and reduces the risk to people. These assessments are only good if they are accurate. To achieve the necessary accuracy computational models need to be used. Obstacles to the use of traditional models by the practitioners in this area include the level of experience needed with the codes and time required to perform the analysis. Here solutions to overcoming these obstacles have been presented.

5 References

[1] Clutter, J.K., Mathis, J.T., and Stahl, M.W., "Modeling Environmental Effects in the Simulation of Explosion Events," *Int. Journal of Impact Engineering*, Vol. 34, No. 5, May 2007, pp. 973-989.

[2] Clutter, J.K., and Stahl, M.W., "Hydrocode Simulations of Air and Water Shocks for Facility Vulnerability Assessments," *Journal of Hazardous Material*, Vol. 106, No. 1,2, Jan 2004, pp. 9-24.

[3] Clutter, J.K., and Mathis, J., "Computational modeling of vapor cloud explosions in off-shore rigs using a flame-speed based combustion model," *Journal of Loss Prevention in the Process Industries*, Vol. 15, Issue 5, September 2002, pp. 391-401

[4] Luccioni, Ambrosini, & Danesi, "Blast load assessment using hydrocodes", *Engineering Structures*, Vol 28 (2006) 1736-1744.

[5] Clutter, Mathis, & Stahl, "Modeling Environmental Effects in the Simulation of Explosion Events," *Int. Journal of Impact Engineering*, Vol. 34, No. 5, May 2007, pp. 973-989.

[6] K. Nazridoust and G. Ahmadi, Airflow and pollutant transport in street canyons, J. Wing Eng. Ind. Aerodyn. 94, 491-522 (2006).

[7] M. J. Berger and J. Olier, Adaptive mesh refinement for hyperbolic partial differential equations, J. Comput. Phys. 53, 484-512 (1984).

[8] M. J. Berger and P. Colella, Local adaptive mesh refinement for shock hydrodynamics, J. Comput. Phys. 82, 64-84 (1989).

[9] M. J. Berger and R.J. LeVeque, An Adaptive Cartesian Mesh Algorithm for the Euler Equations in Arbitrary Geometries, AIAA Paper 89-1930-CP (1989).

[10] T. Akenine-Moller, Fast 3D Trianlge-box Overlap Testing, http://www.acm.org/jgt/AkenineMoller01 (2001).

[11] A.M. Khokhlov, Fully Threaded Tree Algorithms for Adaptive Refinement Fluid Dynamics Simulations, J. Comput. Phys. 143, 519-543 (1998).

[12] Bdzil, J., Stewart, D., Modeling two-dimensional detonations with detonation shock dynamics. Phys. Fluid A 1 (7), 1261- 1267 (1989).

[13] Lee, E.L., Tarver, C.M., Kasimov, A.., Phenomenological model of shock initiation in heterogeneous explosives. Phys. Fluids 23 (12), 2362-2372 (1980).

[14] Clutter, J.K. and Belk, D.M., "Simulation of detonation wave interaction using an ignition and growth model," *Shock Waves*, Vol 3, No. 3, Nov 2002, pp. 251-263

[15] Yoh, et. al., Simulating thermal explosion of RDX-based explosives: Model comparison with experiment, LLNL Report 312703, Oct 2004.

[16] Clutter, J.K., Mikolaitis, D.W., and Shyy, W., "Reaction Mechanism Requirements in on Shock-Induced Combustion Simulations," *Proceedings of the Combustion Institute*, Vol. 28, pp. 663-669, July, 2000

[17] Staszak, Andrew J., 2006, "A Comparison of Numerical Techniques Applied to Multi-Step, Finite-Rate Chemistry Models, a Thesis for The University of Texas at San Antonio.

[18] Clutter, J.K., 1997, "Computation of High-Speed Reacting Flows,", A Dissertation for the University of Florida, pp. 21-50.

[19] Kuo, Kenneth K., 1986, "Principles of Combustion," John Wiley & Sons, New York, NY.

[20] Orkwis, P.D., and Vanden, K.J., 1994, "On the Accuracy of Numerical Versus Analytical Jacobians," AIAA Paper 84-0176