

A NOVEL SCALABLE MODEL FOR SIMULATING EXPLOSIVE BLAST PROPAGATION

James Nutaro

Sudip K. Seal

Computational Science and Engineering Division
Oak Ridge National Laboratory
One Bethel Valley Road
Oak Ridge, TN 37831, USA

Computational Science and Engineering Division
Oak Ridge National Laboratory
One Bethel Valley Road
Oak Ridge, TN 37831, USA

David Sulfredge

Computational Science and Engineering Division
Oak Ridge National Laboratory
One Bethel Valley Road
Oak Ridge, TN 37831, USA

ABSTRACT

Based on the linear wave equation, we propose a new model for propagating explosive blast waves. This scalable model could offer an attractive alternative to ray tracing methods for performing site specific blast calculations while still being compatible with models for shock front interactions that are intended primarily for use with ray based calculations. A scalable parallel implementation of the proposed model is presented and we show that it can solve blast problems on a scale of practical interest. Two specific features of the model are demonstrated in a preliminary validation study: (i) that wave fronts produced by the new model encompass all rays reported by two ray tracing calculations and (ii) the wave model captures all paths from the explosion to the target, whereas the ray tracing model may omit some paths.

1 INTRODUCTION

The state of the art in blast modeling takes three forms. The first and simplest are empirical models that use line of sight distance to calculate the blast intensity (Z. Ul-Hassan Usmani 2012, Ngo, Mendis, Gupta, and Ramsay 2007). These have the benefit of being computationally simple, but they are not suitable for simulating explosive effects inside of specific, confined spaces. At the other extreme are detailed hydrocode models of blast effects that are derived from the conservation laws for mass, momentum, and energy; these bear a strong resemblance to computational fluid dynamics models both in terms of tremendous physical detail and formidable computational requirements (Lea and Ledin 2002). In the middle are ray tracing models that balance computational tractability with realistic physical effects (Roybal, Jeffers, McGillivray, Paul, and Jacobson 2009).

The Visual Interactive Site Analysis Code (VISAC, see Sulfredge and Tolliver 2011) is an explosive blast model developed for assessing the risk posed by an explosive package to machinery in the interior of a building, and it is a state of the art example of ray tracing methods. VISAC uses the method of ray tracing to propagate a blast wave from its point of origin to locations within the building. An analysis begins with the selection of a blast location and the locations of equipment of interest. Then the model creates several rays that are traced from the blast point in the direction of the equipment items. When a ray intersects a piece of equipment, the time of the intersection is recorded. When a ray encounters a concrete structure, such as a wall, the ray is either reflected or it is determined to breach the wall and leave the facility. The total calculation can involve several hundred rays that are traced throughout the facility.

VISAC and ray tracing methods in general are limited in their ability to resolve paths by which the blast wave may reach a target. This is a necessary consequence of casting a finite number of rays, each of which follows a single path through the propagation space. A related problem is approximating diffraction around obstacles, which is computationally impractical with the ray tracing technique as discussed in (Iskander and Yun 2002).

We propose that these problems can be resolved with a propagation model based on the linear wave equation. This approach yields two advantages. First, the wave model naturally explores every possible path to each location, thereby offering a more comprehensive analysis than is possible with a ray tracing model. Second, the wave model naturally diffracts the blast front around structures. Towards this end, we construct a wave based model for blast propagation and provide a comparison of the wave solution with solutions from the VISAC ray tracing model. Our preliminary investigation focuses on time of arrival and path length while accounting for reflections, and we use VISAC as a point of comparison with state of the art methods in ray tracing applied to explosive blast problems. We leave the problem of diffraction to future work.

The primary computational challenge offered by this new model is the amount of computer memory needed to store the 3D array arising from a finite difference approximation to the wave equation. We address this problem by creating a distributed memory implementation of the model and showing that it can solve problems on a scale of practical interest. The novel contributions of this work are the following:

1. It is the first model for large scale blast propagation simulation that is both computationally tractable as well as capable of providing comprehensive information about the shock fronts that reach a target;
2. It represents a novel, practical demonstration of the close relationship between ray tracing and the linear wave equation; and
3. It is a significant example of how parallel computing at a modest scale can enable fundamentally new modeling techniques.

The rest of the paper is organized as follows. The numerical model based on the three dimensional linear wave equation is described in Section 2 which includes an illustrative example in Section 2.2. The parallel implementation of the model is described in Section 3. Simulation results from the new model are compared with ray tracing results in Section 4. Conclusions and future work are discussed in Section 5.

2 NUMERICAL MODEL

The salient elements of the propagation problem are (1) the time of intersection and (2) the distance r traveled by the ray. The shock overpressure due to a single ray is proportional to $1/r^3$, and overlapping rays are combined in a nonlinear manner to emulate the effect of interacting shock fronts (Sulfredge and Tolliver 2011, Z. Ul-Hassan Usmani 2012). However, these nonlinear interactions do not (at least within the model) influence the actual propagation of the blast wave, and so we do not pursue this topic further here.

We use the linear wave equation with source term $\psi(t)$ and wave velocity v to describe how the shock front moves through space. With this model the pressure u at each location is described by

$$\frac{1}{v^2} \frac{\partial^2 u}{\partial t^2} = \nabla^2 u + \psi . \tag{1}$$

It is convenient for numerical reasons to make ψ a half sine wave located at the origin so that

$$\psi(t) = \begin{cases} 4\pi\rho \sin(\pi ft)\delta(\mathbf{x}) & 0 \leq t \leq 1/f \\ 0 & \text{otherwise} \end{cases} , \tag{2}$$

where $\delta(\mathbf{x})$ is the Dirac delta function. The term \mathbf{x} is location in the 3D space and the frequency f is chosen to approximate the duration of the initiating explosion. The constant ρ determines the amplitude of the blast wave.

If we use centered differences with step h in time and $k = hv\sqrt{3}$ in space to approximate Eqn. 1 (Sheaffer and Walstijn 2012), then at each point in the discrete space we get a difference equation

$$u_{n+1} = \frac{1}{3} \sum_{i=1}^6 u_{n,i} - u_{n-1} + \frac{h^2 v^2}{k^3} \psi(nh) . \quad (3)$$

The source ψ is given by Eqn. 2 at the point of the blast and it is zero everywhere else. Subscripted $u_{n,1}$, ..., $u_{n,6}$ in the sum are the pressures at the neighboring grid points in the finite difference stencil. In this derivation we will be interested in simulating blasts in closed structures. For this purpose, we use reflective boundary conditions that are obtained by setting $u_{n,i} = 0$ at points not in the computational grid.

This numerical solution for u will diminish with distance r from the blast in proportion to $1/r$, and the initial upward slope of the sine function will appear at time r/v . We can transform this solution to the wave equation into an approximate solution for the blast overpressure p by scaling and shifting u . First, p should diminish with $1/r^3$; to do this we multiply u by $1/r^2$. The distance $r = vt$ is given by the finite difference simulation. Second, the maximum value of u is shifted in time by half of $1/f$ so that it coincides with the time of arrival of the shock front.

Together these transformations give the blast overpressure as

$$p = \frac{\left| u \left(t + \frac{1}{2f} \right) \right|}{r^2} . \quad (4)$$

The absolute value is used because the wave equation can produce negative as well as positive amplitudes, whereas the overpressure data is expected to be positive.

The model's parameters are initialized to match the characteristics of the explosion. The force of the explosion can be described by the pressure p_0 produced at a reference distance r_0 . Given p_0 and r_0 we choose

$$\rho = p_0 / r_0^3 . \quad (5)$$

If the duration of the initial blast is d seconds then

$$f = \frac{1}{d} . \quad (6)$$

2.1 Extracting Rays from the Wave Solution

Blast damage to components and structures is a function of both the peak overpressure achieved and the shock impulse delivered per unit surface area, which is the time integral of the shock wave pressure trace. Both of these quantities are readily evaluated when the time history of the pressure at the target location is known. Developing an accurate pressure history is complicated because superposition of the shock overpressures from the primary and reflected waves is a highly nonlinear physical process.

The ray tracing algorithms in VISAC (Sulfredge and Tolliver 2011) accomplish this nonlinear superposition accurately for primary rays coming directly from the blast location as well as the rays from reflected shocks. On the other hand, the proposed wave model inherently combines intersecting shock waves with a linear superposition, which may underpredict the peak overpressure and impulse and understate the degree of damage. To get a more accurate damage assessment, we can extract rays from the wave data and then apply established methods to account for nonlinear interactions.

The overpressure data collected from the wave model can be transformed into a set of rays with a procedure that examines the distinct peaks in the wave solution. This procedure begins by picking out the

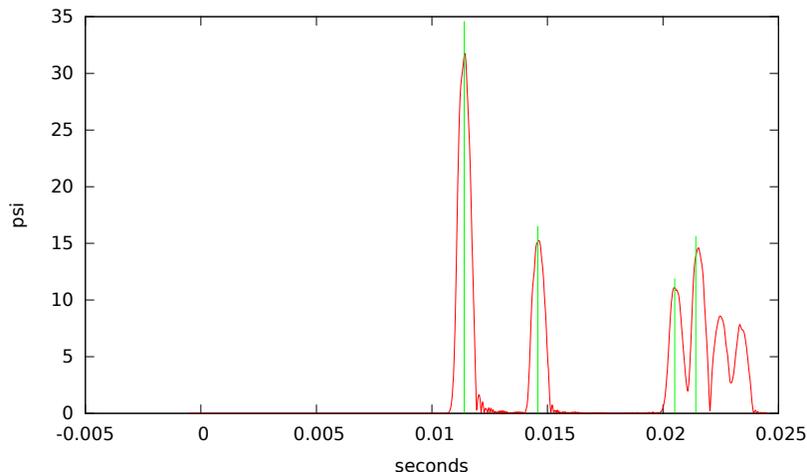


Figure 1: Overpressure data produced by a ray model and the proposed wave model. The overpressure spikes (green) are from the ray model and the curve (red) from the wave model.

largest m overpressures p_1, \dots, p_m from the wave data; these will correspond to m or more rays. Suppose that these occur at times t_1, \dots, t_m . For an initial blast overpressure of p_0 at r_0^3 the expected \tilde{p}_j at time t_j for a single ray is

$$\tilde{p}_j = \frac{p_0 r_0^3}{v^3 t_j^3} . \tag{7}$$

The number of rays R_j corresponding to the overpressure at time t_j can be calculated as

$$R_j = \frac{p_j}{\tilde{p}_j} , \tag{8}$$

with fractional rays being rounded up to the nearest whole number. Since the wave model solution is linear, this calculation yields the number of superimposed shocks of magnitude \tilde{p}_j which are contributing to the total overpressure at time t_j .

2.2 An Illustrative Example

Figure 1 shows two solutions for the overpressure in a blast problem. The first solution is calculated by hand using the ray tracing method (green) for rays following four groups of paths. The second solution is calculated using the wave model (red). The blast occurs in a closed room with perfectly reflective walls. The blast speed is $v = 2194$ ft/s with $p_0 = 20$ psi at a distance $r_0 = 30$ ft. These values were chosen arbitrarily but are within the range of values that could be expected in an actual application. The duration of the blast is 1 ms. These data require $\rho = 0.000741$ and $f = 1000$. The dimensions of the room in which the blast occurs are $40 \times 45 \times 30$ ft. The source of the explosion is at $x = 20, y = 10, z = 10$ and we record data at the point $x = 20, y = 35, z = 10$. The step size of the simulation is $h = 0.015$ ms.

We compare four groups of paths for the corresponding ray tracing model; these groups are illustrated in Figure 2. The single member of the first group is the direct path from the origin of the blast to the test point. This ray has a length of 25 ft and it arrives first. The second group has for its single member the ground reflection, which has two legs of equal length: the first from the blast point to the floor at $x = 20, y = 22.5, z = 0$ and the second from the floor to the test point. The length of this ray is 32 ft and it arrives second.

The third group has two rays. One of these is formed by tracing a ray from the blast point to the wall directly behind the test point and then reflecting that ray back to the test point. This path has a length of

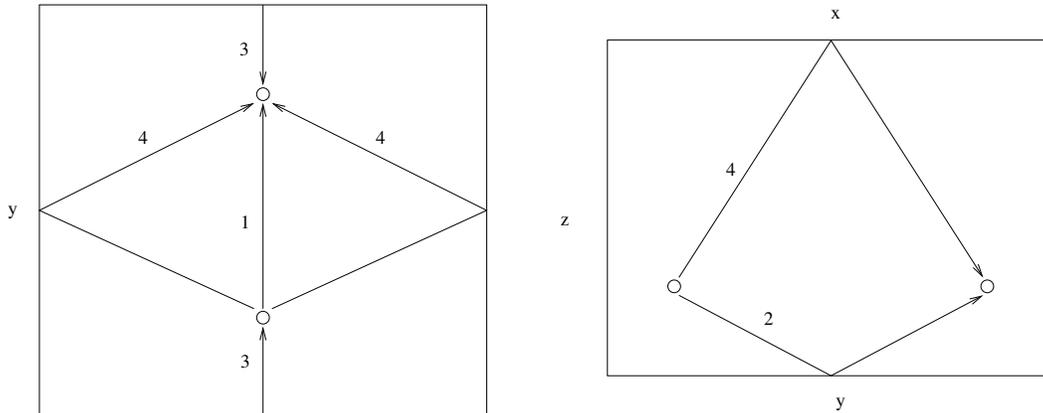


Figure 2: Selected path groups for the ray tracing calculations. Drawings are not to scale.

Table 1: Rays extracted from the wave solution.

group	t_i (s)	p_i	R	\tilde{p}_i (psi)	p error
1	0.011425	31.7536	1	34.2859	0.2741
2	0.01462	15.241	1	16.3622	0.1173
3	0.020605	10.8647	2	5.1091	0.8169
4	0.02155	14.595	3	5.8448	0.6996

45 ft. The other is formed by tracing a ray from the blast point to the wall at $y = 0$ and then from that wall to the test point. This ray also has a length of 45 ft. Both rays arrive simultaneously at the test point.

The fourth group that we consider has three members of length 47 ft. The first member is the reflection from the wall at $x = 0$, the second is the reflection from the ceiling at $z = 30$, and the third is the reflection from the wall at $x = 40$. These also arrive simultaneously at the test point.

These four groups of rays that can be derived from a ray tracing procedure are apparent in the wave solution shown in Figure 1 when we use the procedure described in Section 2.1 with $m = 4$. The overpressure indicated by the ray tracing solution can be calculated using Eqn. 7 with vt_j replaced by the ray length. Table 1 shows the amplitude of the rays extracted from the wave solution, relevant data for extracting them, and the error magnitude in the amplitude of the extracted rays relative to the ray tracing solution. It is also clear from the wave solution that there are additional waves not covered by our wave solution; large numbers of these waves will become apparent in our comparison with VISAC.

3 PARALLEL IMPLEMENTATION

The above wave propagation model was implemented in parallel using MPI (Message Passing Interface) for inter-processor data exchange. Performance of the parallel implementation was compared with a standalone sequential implementation of the model. Both implementations were written in the C++ language. A 48-node Linux cluster operating at 2.8GHz was used for our performance measurements. Each cluster node is a 6-core AMD Opteron processor. Two problem sizes, which we refer to as *small* and *large*, were used for parallel performance measurements. The small problem size represents a computational grid with ~ 7 million ($N = 7.4 \times 10^6$) points while the large problem size consists of ~ 60 million ($N = 5.9 \times 10^7$) points.

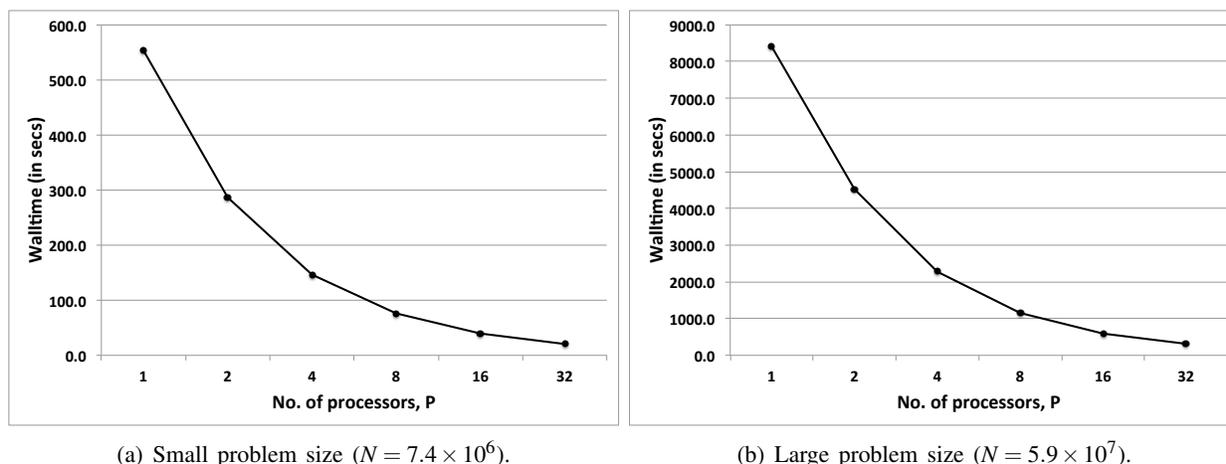


Figure 3: Parallel execution time using $P = 1, 2, 4, 8, 16$ and 32 .

3.1 Parallel Domain Decomposition

The computational grid $N_x \times N_y \times N_z = N$ is generated using a simulation time step h as described in Sect. 2. The resulting $N_x \times N_y \times N_z$ grid is block partitioned across a three-dimensional array of processors $P_x \times P_y \times P_z$ where $P = P_x \cdot P_y \cdot P_z$ is the total number of processors. The choice of a three dimensional array of processors instead of an easier to implement linear array allows for a larger number of processors to be deployed for the same computational grid. A linear partitioning across the longest grid dimension $N_\ell = \max\{N_x, N_y, N_z\}$ would limit P to be smaller than or equal to N_ℓ so that $N_\ell/P \geq 1$. Instead, a three dimensional processor topology requires that $P \leq N_x \cdot N_y \cdot N_z$ allowing for a much larger number of processors for the same computational grid.

3.2 Parallel Complexity

The computation performed at each grid point is a sum of field values at six neighboring grid points, which is a constant time operation. Therefore, the volume of computation performed by each processor per time step is

$$\text{Computation overhead} \propto \frac{N_x}{P_x} \cdot \frac{N_y}{P_y} \cdot \frac{N_z}{P_z} = O\left(\frac{N}{P}\right).$$

Note that computations at points located on the outer faces of each processor's local (grid) domain require non-local values that are available in the six neighboring processors (the processors at the edges of the three dimensional processors topology have lesser neighbors). For computational correctness, these remote values need to be available locally on a processor before updating the values at the local grid points for the next time step. As such, each processor sends and receives the appropriate edge values from its neighboring processors before the computations at a time step are performed.

The inter-processor communication overhead needed to accomplish this data exchange can be modeled using a permutation network. In a permutation network, the overhead from each round of communication is modeled as $\tau + \mu\ell$ and each processor is allowed to send and receive at most one message during a communication step. Here, τ is the start-up overhead for a communication step, μ is the transfer bandwidth of the communication network and ℓ is the size of the largest message. Permutation networks closely model the behavior of most multistage interconnection networks. Therefore, the overhead of communication per time step is

$$\text{Communication overhead} \propto \tau + \mu \left(\frac{N}{P}\right)^{\frac{2}{3}} = O\left(\frac{N^{\frac{2}{3}}}{P^{\frac{2}{3}}}\right).$$

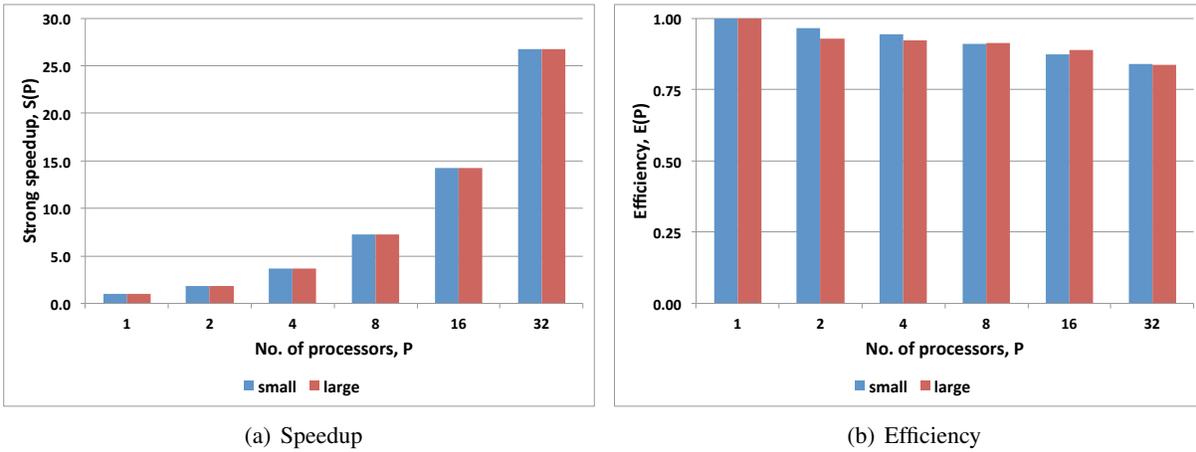


Figure 4: Speedup and efficiency using $P = 1, 2, 4, 8, 16$ and 32 .

The complexity, $T(N, P)$, of the parallel implementation is therefore

$$T(N, P) = O\left(\frac{N}{P}\right) + O\left(\frac{N^{\frac{2}{3}}}{P^{\frac{2}{3}}}\right) = O\left(\frac{N}{P}\right)$$

asymptotically as N becomes very large. Figure 3 confirms this parallel execution time on up to 32 processors for both the small and large problem sizes.

3.3 Speedup and Efficiency

Let $T(N)$ denote the sequential complexity, which scales as $O(N)$. Then, the asymptotic strong scaling speedup is

$$\text{Speedup, } S(P) = \frac{T(N)}{T(N, P)} = O(P) .$$

The corresponding efficiency, defined as speedup per processor, is

$$\text{Efficiency, } E(P) = \frac{O(P)}{P} = O(1) .$$

Figure 4(a) and Figure 4(b) confirm the expected speedup and efficiency of the parallel implementation. Note that the efficiency deteriorates as the number of processors increases. This performance slowdown at larger processor counts is to be expected, as discussed next.

3.4 Granularity

As the number of processors increase for a fixed problem size, the computational granularity decreases. As a result, most of the overhead is from data transfer across the neighboring processors with very little time spent for local computations. This behavior is captured by measuring the ratio of computation-to-communication overheads, as follows:

$$\frac{\text{Computation time}}{\text{Communication overhead}} = O\left(\frac{N^{\frac{1}{3}}}{P^{\frac{1}{3}}}\right) .$$

As the processor count increases, this ratio becomes smaller. Figure 5 confirms this. Note that even when the computation-to-communication ratio drops to 20%, the algorithm still delivers over 75% efficiency. Therefore, though the parallel implementation of this model favors coarse grained executions, it is still a

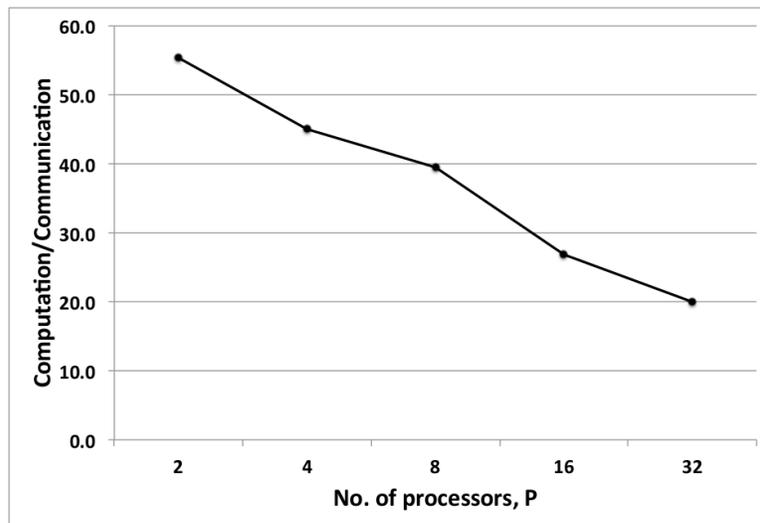


Figure 5: Computation-to-communication ratio for the large problem size.

very efficient parallel algorithm even at lower granularities. The scalability of the parallel implementation of the model clearly demonstrates that very realistic simulations with hundreds of millions of points can be handled by even modest sized parallel computers.

4 VISAC COMPARISON

To validate the correspondence between the arrival time of fronts in the wave model and rays in the ray tracing models, we conducted a blast simulation using the VISAC model and compared that with data generated by the parallel algorithm. This experiment comprises a closed room with its x -axis spanning [31.5, 150.5] ft, y -axis spanning [-275.5, -295] ft, and z -axis spanning [586, 608] ft. The origin of the blast is at $x = 141$, $y = -267$, $z = 587$ and the target of interest is at $x = 118.47$, $y = -290.75$, $z = 590.3$. The speed v of the blast wave, frequency f , and the other wave model parameters are as described in Section 2.2. The shock overpressure in these two models is not directly comparable because of the nonlinear way in which rays adjacent in time are combined by VISAC to produce an overpressure, and the capability to use the extracted rays in this procedure has not yet been created. Instead, we focus on comparing the time of arrival and from that extrapolate the expected range of the peak pressure errors.

The VISAC model reports four rays incident on the target. These arrive at 14.89 ms, 15.08 ms, 29.45 ms, and 35.95 ms; the solid impulses in Figure 6 show these rays. Table 2 quantifies the difference in the time of arrival of the VISAC rays and their nearest peak in the wave data and the corresponding difference in the path length. The latter is most significant because it contributes directly to the overpressure calculation. The maximum error in time is just slightly above the 0.5 ms, resulting in a path length difference of about 1 ft.

We can use Eqn. 7 to estimate the expected difference in peak overpressure. For this example, we can determine from the model geometry that the first, direct path in the VISAC model has a length of 32.9 ft (VISAC does not report path lengths, and so we cannot repeat this calculation for the other rays). The largest pressure difference results if the corresponding ray extracted from the wave model is shorter by a foot (the largest error in Table 2), having a length of 31.9 ft. For these two rays, the difference in the calculated overpressures will be

$$p_0 r_0^3 \left(\frac{1}{32.9^3} - \frac{1}{31.9^3} \right) \approx -2.72 \times 10^{-6} p_0 r_0^3 .$$

Table 2: Comparison of ray arrival times computed with VISAC and the wave model.

VISAC (ms)	wave (ms)	time difference (ms)	path length difference (ft)
14.89	14.70	0.19	0.42
15.08	15.45	0.37	0.81
29.45	29.85	0.40	0.88
35.95	35.44	0.51	1.1

Most applications of interest for VISAC involve 10^1 to 10^2 of pounds of TNT or its equivalent. For the larger cases, p_0 and r_0 are on the order of 10^1 and the overpressure error will be on the order of $10^1 \times 10^3 \times 10^{-6} = 10^{-2}$ (Alonso, Ferradás, Pérez, nana Aznar, Gimeno, and Alonso 2006, Kingery and Bulmash 1984). For comparison, windows are shattered at overpressures of approximately 0.15 – 0.22 psi (Chiple, Kaminskas, Lyon, Beshlin, and Hester 2003), and so these anticipated errors are well below the threshold for inflicting any significant structural damage.

An interesting feature in the wave solution is the collection of peaks for which there are no analogous rays reported by VISAC, and we can trace at least one of these missing rays by hand. This ray reflects from the x -most wall and comprises two parts: the vector $\mathbf{v}_1 = [x_1, y_1, z_1]$ from the blast point at $\mathbf{p}_1 = [141, -267, 587]$ to the wall at $x = 150.5$ ft and $\mathbf{v}_2 = [x_2, y_2, z_2]$ from the wall to the target at $\mathbf{p}_2 = [118.47, -290.75, 590.3]$. It is described by the set of equations

$$\begin{aligned} \mathbf{p}_1 + \mathbf{v}_1 + \mathbf{v}_2 &= \mathbf{p}_2 , \\ x_1 &= -kx_2 , \\ y_1 &= ky_2 , \\ z_1 &= kz_2 , \\ x_1 &= 150.5 - 141 = 9.5 , \\ x_2 &= 118.47 - 150.5 = -32.03 , \text{ and} \\ k &= -x_1/x_2 \approx -3.3716 . \end{aligned}$$

The first equation states that the path must travel from the blast point to the target. The next three equations state that the only change in direction is along the x axis; otherwise, the vectors can differ only in their length. The last set of equations are necessary for a reflection to occur at the wall.

Solving for the two vectors gives

$$\mathbf{v}_1 = [9.50000 , -5.43300 , 0.36600] \text{ and} \tag{9}$$

$$\mathbf{v}_2 = [-32.0300 , -18.3170 , 2.5451] , \tag{10}$$

which yields a total path length of $\|\mathbf{v}_1\| + \|\mathbf{v}_2\| = 47.955$ ft and the time of arrival is 0.021857 s. This ray is shown by the dotted impulse in Figure 6, and it closely coincides with a peak produced by the wave model.

5 CONCLUSION

We have demonstrated that the wave model provides time of arrival and maximum pressure data that closely resemble data computed with a ray tracing model and have proposed a method for extracting rays from wave data. The proposed model is sufficient to account for line of sight and reflections, and it captures all blast wave paths that result from these circumstances to within the spatial resolution of the computational grid. Significantly, many of these paths do not appear in the ray tracing solution because of the computational infeasibility of casting very large numbers of rays. An open question is how the natural diffraction in the wave calculation can be mapped into an appropriate overpressure. A solution to this problem would offer

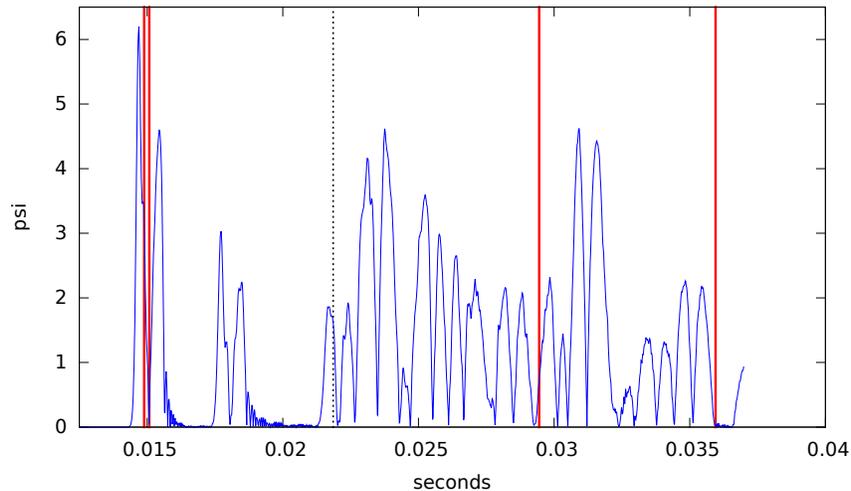


Figure 6: Comparison of shock front arrival times in the VISAC ray model and proposed wave model.

a fundamental advancement over the state of the art as ray tracing does not account for movement of the blast wave around a structure.

A simple approach to this problem would be to treat it in precisely the same manner as direct and reflected waves; that is, multiply the numerical solution by $1/(vt)^2$. Because the diffracted solution is naturally attenuated by the wave model, this would lead to a diffracted blast front somewhat less intense than a direct or reflected front of the same path length. However, it remains an open question whether this, or some other, approach can properly mimic the bending of blast waves around obstacles.

A second open problem is characterization of the simulation errors introduced by the proposed method, both with respect to actual measurements from experiments and a theoretical examination of numerical artifacts. Errors arising from the central difference approximation to the wave equation are well understood. However, it is unknown how these interact with the proposed procedure for extracting rays from the computational wave data. The preliminary results that we have obtained are encouraging in the sense that the ray tracing and wave based methods are in general agreement, but a formal, comprehensive treatment of this problem is an important topic for future work.

ACKNOWLEDGEMENTS

This manuscript has been authored by UT-Battelle, LLC, under Contract No. DE-AC0500OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for the United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (<http://energy.gov/downloads/doe-public-access-plan>). Research sponsored by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the U. S. Department of Energy.

REFERENCES

- Alonso, F. D., E. G. Ferradás, J. F. S. Pérez, A. M. nana Aznar, J. R. Gimeno, and J. M. Alonso. 2006. "Characteristic Overpressure-Impulse-Distance Curves for the Detonation of Explosives, Pyrotechnics or Unstable Substances". *Journal of Loss Prevention in the Process Industries* 19:724–728.

- Chiple, M., M. Kaminskas, W. Lyon, D. Beshlin, and M. Hester. 2003. "Reference Manual to Mitigate Potential Terrorist Attacks Against Buildings". Technical Report FEMA 426, ch. 4, pg. 4-19, Table 4-3, FEMA.
- Iskander, M. F., and Z. Yun. 2002. "Propagation Prediction Models for Wireless Communication Systems". *IEEE Transactions on Microwave Theory and Techniques* 50 (3): 662–673.
- Kingery, C. N., and G. Bulmash. 1984. "Airblast Parameters from TNT Spherical Air Burst and Hemispherical Surface Burst". Technical Report ARBRL-TR-02555, US Army Ballistic Research Laboratory, Aberdeen, Maryland.
- Lea, C., and H. Ledin. 2002. "Review of the State-of-the-art in Gas Explosion Modelling". Technical Report HSL/2002/02, Health & Safety Laboratory, Harpur Hill, Buxton, SK17 9JN.
- Ngo, T., P. Mendis, A. Gupta, and J. Ramsay. 2007. "Blast Loading and Blast Effects on Structures - An Overview". *Electronic Journal of Structural Engineering* Special issue: Loading and Structures:76–91.
- Roybal, L. G., R. F. Jeffers, K. E. McGillivray, T. D. Paul, and R. Jacobson. 2009. "Modeling and Simulating Blast Effects on Electric Substations". In *IEEE Conference on Technologies for Homeland Security, 2009 (HST '09)*, 351–357. IEEE.
- Sheaffer, J., and M. V. Walstijn. 2012. "A Physically Constrained Source Model for FDTD Acoustic Simulation". In *Proceedings of the 15th International Conference on Digital Audio Effects (DAFx-12)*, 1–7. The Audio Lab.
- Sulfredge, C. D., and J. S. Tolliver. 2011. "Development of Improved Shock Propagation Algorithms for VISAC Facility Vulnerability Code". Technical Report ORNL/TM-2011/335, Oak Ridge National Laboratory, Oak Ridge, TN.
- Z. Ul-Hassan Usmani 2012. "Explosion Modeling - A Tutorial". In *Autumn Simulation Multiconference 2012 (AutumnSim 2012)*. The Society for Modeling & Simulation International.

AUTHOR BIOGRAPHIES

JAMES NUTARO is a Distinguished Research & Development Scientist at Oak Ridge National Laboratory in the Computational Science and Engineering Division. He holds a PhD in Computer Engineering from the University of Arizona in Tucson, Arizona. His email address is nutarojj@ornl.gov.

SUDIP SEAL is a research staff member in the Computational Sciences and Engineering Division at Oak Ridge National Laboratory. His primary research is in the design and development of parallel algorithms for science and engineering applications. At present, his research focusses on scalability of parallel discrete event simulations on very large parallel platforms. His other major research interests include computational biology and computational physics. Dr. Seal has a PhD in Computer Engineering (Iowa State University, 2007) and a PhD in Theoretical Physics (New Mexico State University, 2002).

DAVID SULFREDGE is a Senior R&D staff member with the Modeling and Simulation Group at ORNL, where he works on facility vulnerability and various types of thermo-fluid modeling. He has developed numerous nuclear facility models together with event/fault tree algorithms for estimating downtime and calculating the probability of facility kill or radiological releases. Many of these models have been incorporated into the Visual Interactive Site Analysis Code (VISAC) for assessment of nuclear facility vulnerability. His work also has included the design of scaled experiments to measure the blast fragility of nuclear power plant equipment and calculations with weapons effects codes for blast wave propagation. In addition to his explosives-related research, he has performed extensive fluid flow analysis of the hydraulic behavior and separative performance in gas centrifuges, as well as heat transfer calculations to determine the temperature profile inside an operating centrifuge. David Sulfredge has written or contributed to over 65 published papers, research reports, conference proceedings and book chapters. He received his B.S., M.S., and Ph.D. degrees in mechanical engineering from the University of Kentucky.