

Comparative analysis of blast prediction software for far-field shock wave effects behind a blast wall

Alastair Chester^{1,2} , Rachael Hazael² and Richard Critchley² 

International Journal of Protective Structures

2024, Vol. 0(0) 1–28

© The Author(s) 2024



Article reuse guidelines:

sagepub.com/journals-permissions

DOI: 10.1177/20414196241284299

journals.sagepub.com/home/prs



Abstract

This paper investigates a selection of current and emerging software used in the prediction of overpressure generated through the detonation of a high explosive in the far field behind a blast wall. In particular, this paper compares the software Autodyn, blastFoam, ProSAir, Viper::Blast and WALAIR++. These packages are compared by simulating a 100 kg TNT explosive charge at a stand-off distance of 25 m from a complex structure, then reviewing the performance in terms of the overpressure results, speed of each modelling package, the degree of effect from mesh, and domain sizes and ease of use. A live experimental trial representing the simulation was also performed, although it used a similar but different explosive, for a high-level comparison. The live-trial instrumentation design details are reviewed and compared with best practice. The choice of software is found to lead to variations in peak pressure predictions of 28%, specific impulse by 10% and the simulation speed can vary by a factor of up to 1600 for this type of study. This shows that the choice of blast software package can have a significant impact on the accuracy and attainability of blast predictions.

Keywords

simulation, blast testing, computational modelling, high explosives

Introduction

In accidental initiation or deliberate detonation of a high explosive device (Cooper, 2018) in an urban area, it is unlikely that the affected structures correspond to the ideal reflectors that can be used to predict the pressure and specific impulse loading of the shock wave (Davison and Silberschmidt, 2018). Through the early work of Friedlander (Friedlander, 1946) and the following work of

¹Spadeadam Research and Testing, DNV Energy Systems, Brampton, UK

²Cranfield University, Cranfield Forensics Institute, Defence Academy of the UK, Shrivenham, UK

Corresponding author:

Alastair Chester, Spadeadam Research and Testing, DNV Energy Systems, MoD R5, Brampton CA8 7AU, UK.

Email: alastair.chester@cranfield.ac.uk

Kingery and Bulmash (1984), the pressure time history of a charge detonated near an infinitely dimensioned reflective surface parallel to the shock front at ground level has been well defined. Where structures begin to deviate from these ideal conditions, this leads to errors in prediction (Ballantyne et al., 2010), and computational fluid dynamics (CFD) tools can then be used to enhance accuracy. There is a wide range of different CFD tools available to the user for the prediction of blast effects. Some of these are legacy packages such as Air3D that have been replaced by their modern equivalents, in this case ProSAir. There is also more recent software to the market such as blastFoam, Viper:Blast and WALAIR++, along with packages with a long history such as Autodyn and many others that will generate overpressure predictions from high explosive events. Chester et al. (2024) performed a systematic study of five predictive software solutions for scenarios of 15 kg and 100 kg TNT_{eq}. The set-up studied was a reflective wall populated with simulated pressure gauges on the front facing the charge. The article found that in this reflective regime, through the choice of software even after optimisation, the peak pressure and specific impulse values still ranged over 50% and 15%, respectively. This paper continues that work, but for a more complex scenario. The scenario studied here is the area behind a finite reflective structure, which is of direct interest to the design of the blast wall (Bangash, 2001). Additionally, in the area behind the wall, extra reflecting surfaces have also been placed to create multiple reflective waves superimposing at different times throughout the pressure time history as measured in the middle at the base of the rear wall. These were added with the aim of providing additional complexity to the CFD models to incite variations.

As noted by Chester et al. (2024) the size of the variations in the output of the simulation programmes is of great interest, both academically and practically, to make sure that the simulated blast loads are accurate representations of a real event and that the user's choice of software does not have a negative effect on the validity of the predictions. In addition to the differences in the results, the time taken to obtain them is also important, as typically these models can take a number of hours to run for fine mesh sizes. This paper examines the variations in these run-times for the wide range of software investigated.

To understand the range of performance of the differing software packages in a complex far-field event and compare it to a less complex scenario, the output of several commonly used software packages has been analysed. All of these packages simulated the same physical setup and charge size. In addition, a live experimental trial was performed that replicated the simulation construct, although for a similar performing but different type of explosive, to allow a high-level comparison. Cell and domain size fidelity studies have been conducted for each programme to ensure that relative performance is represented fairly, along with run times and cell size versus output comparisons.

As stated by Chester et al. (2024), several studies have already been conducted investigating the comparative performance of software. Tang et al. (2018) conducted a validation study that compared live test data with Air3D and CONWEP predictions for a wide range of charge sizes from 0.25 to 500 kg. Previous work by Shin et al. (2014) performed the largest and most comprehensive package comparison found prior to his comparison, comparing Autodyn, Kingery Bulmash, and Air3D, where it was found that in 2D simulations the overpressure and specific impulse values were typically within 10% of Kingery Bulmash. Jha and Kumar (2014) compared 1D Autodyn with CONWEP for a single 10 kg charge in a three-metre standoff, finding that the overpressures were within 4%. Many of these comparisons were made for legacy software such as Air3D and CONWEP. Large reviews of commercial blast software have not been found in the published literature, and very little information is available comparing any of the recently developed blast

software, and this document aims to contribute to this knowledge gap, critically analysing both the results and the usability.

Software selection criteria

To continue the work of [Chester et al. \(2024\)](#) the same software solutions were used to allow a deeper comparison of a more complex scenario. Furthermore, these programmes are all commercially available, have been used throughout the open literature, and are available to the author. Emblast ([Bassam, 2020](#)) is not studied here as it is neither designed nor has the functionality to investigate the scenario in this paper ([Angelides et al., 2022](#)).

As the graphical processor unit (GPU) based Viper::Blast was shown to be significantly faster than the other central processing (CPU) based software, WALAIR++ has been added to the study as another GPU driven package for comparison. blastFoam has developed a GPU version, but it is not publicly available as of December 2023.

Methodology

This section details all the software packages explored in this study and provides an overview of each one. All simulations were run on the same computer as detailed in [Table 1](#).

Each of the CFD programmes are hydrocodes. Hydrocodes are CFD tools specifically designed for modeling the behavior of fluids in high-speed, high-pressure conditions typically found in explosions and impacts. They help predict the performance and effects of explosives by accounting for the complex interactions between shock waves, gas dynamics, and solid material responses ([Handley et al., 2018](#)).

Autodyn

Autodyn ([ANSYS, Inc, 2023](#)) is a software programme developed by Ansys Inc. to simulate the response of materials to impact, high pressure, and explosive events. It utilises nonlinear CFD techniques to model the interactions between solids, fluids, and gases. Autodyn is a proprietary software package, and pricing information is not publicly disclosed.

Widely used in publications with over 4000 results in Google Scholar for the search term “autodyn blast explosive”, Autodyn has the most search results compared to the other packages studied here. The ubiquity of its use is probably due to its wide range of capabilities, which go far beyond the predictions of overpressure investigated here, such as the study of blast on materials such as reinforced concrete ([Attia et al., 2021](#)).

Table 1. Simulation computer specification.

Item	Specification
Operating System	Windows 11
Processor	Intel(R) Core (TM) i7-12700K
RAM	16 GB of 4400 MHz DDR5
Graphics Card	NVIDIA GeForce RTX 3060

Autodyn operates through a graphical user interface (GUI) on Windows systems. It offers extensive simulation capabilities, including the import of computer-aided design (CAD) drawings and the placement of gauges at various locations within the mesh to monitor cell conditions such as overpressure.

This study used Autodyn version 2022 R2, which was the latest available version at the time of writing.

blastFoam

blastFoam (Heylmun et al., 2023a), an open source computational fluid dynamics (CFD) software developed by Synthetik Applied Technologies, has been engineered with a special focus on the precise simulation of blast effects, which encompasses detonation phenomena and air blast dynamics. The software framework is built on the open source OpenFOAM library (OpenFOAM Foundation, 2024), which forms its core foundation. blastFoam has extended OpenFOAM to produce a dedicated solver tailored for high explosive and detonation simulations. blastFoam has the capability to account for afterburn, which can occur after a detonation when explosive products continue to burn due to insufficient internal oxygen. This additional combustion releases additional energy beyond the initial detonation, significantly contributing to the overall energy output (Tyas et al., 2016).

blastFoam has the widest flexibility in configuration for the packages studied here, with users having the ability to manipulate a range of variables, including equation of state, flux schemes, and afterburn models. However, using these additional variables necessitates a detailed comprehension of the software's intricacies.

As the only open-source package studied here, blastFoam is available directly for download from the GitHub repository (Heylmun et al., 2023c). Among these distributions are comprehensive installation guidelines that cater to various operating systems. Operating through a command line interface, the software outputs results that can be visualised using ParaView (Kitware, 2024), an open source scientific visualisation tool. Blast properties such as overpressure are recorded by locating gauges within the computational domain. A feature of the software corrects potential user error by relocating the gauges from inactive cells to the nearest active cell (Heylmun et al., 2022). Structures are incorporated into simulations through the software's native code, utilising primitive shapes or by direct import from CAD models.

blastFoam supports mesh refinement and adaptive meshing. The former enables increased cell density in specific domain regions, emulating the benefits of a reduced cell-size domain, while minimising computational overhead. Adaptive meshing forms a dynamic cell density adjustment in regions where cell data are changing with the intention of optimising run times and enhancing accuracy. As an example of its published use, blastFoam has been used by Murong et al. to study blast dissipation in urban areas (Murong et al., 2023). Throughout the course of this study, blastFoam version 6.2 was used, the latest iteration available on 27 March 2023 (Heylmun et al., 2023b).

ProSAir

ProSAir (Forth, 2022b), (PROpagation Of Shocks in AIR) is a finite-volume solver for compressible fluid dynamics. It uses the second-order MUSCL-Hancock time integration and the AUSMDV hybrid flux vector/difference splitting method developed by Wada and Liou (Wada and Liou, 1997). The software specialises in modelling air blasts in and around structures and estimating the resulting structural loading. It is an improved version of Air3d version 9, maintaining the same numerical

algorithm. ProSAir states that it offers an improved user interface and additional features, such as easy CFD modelling of blast waves from single high explosive devices, fast and robust performance in capturing shocks accurately, second-order accuracy with the MUSCL-Hancock integration scheme, and support for multicore/processor computers (Forth, 2022a). ProSAir is Windows-based and compatible with 64- and 32-bit versions. The software includes a graphical user interface for the creation of models, a graphical display of solutions during execution and batch job capabilities via the command line, and the ability to provide a history of pressure, velocity, density, and temperature at multiple target points. ProSAir has certain limitations, since it only simulates spherical explosive charges with no afterburn capability. Structural models are created using primitive shapes and ProSAir does not support CAD import. ProSAir is the second most widely used programme in this study and has been used to investigate a variety of scenarios such as tunnels, blast walls, and cityscapes (Burrows et al., 2021; Mullett et al., 2013; Ouyang et al., 2020).

Viper::Blast

Viper::Blast (Stirling, 2022), Viper::Blast is a CFD tool developed by Stirling Simulation Services Ltd. for simulating the detonation of high explosives in the air. It is compatible with various operating systems and utilises the GPU for simulations, unlike the CPU used by other programmes discussed in this paper. Viper::Blast specifically requires a NVIDIA CUDA capable GPU, which is commonly found in Personal Computers (PCs). The software features a GUI with a concise layout, providing users with all the necessary simulation-setup functionalities on a single page. A detailed manual is included that offers guidance on software setup and optimisation through several worked examples. Viper::Blast offers a wide range of capabilities, including afterburn modelling, multiple charges, Monte Carlo simulations, Jones-Wilkins-Lee (JWL) charge detonation, P-I curves, and real-time output of simulation results and gauge traces. The analysis in this article was performed using the latest version available at the time of writing, 1.20.3. Examples of published works for Viper::Blast include tunnel shape openings (Pang and Shin, 2021), Monte Carlo simulations for airblast variation in urban environments (Marks et al., 2021), and studies of shock focussing from multiple charges (Zaghloul et al., 2021).

WALAIR++

WALAIR++ (Thornton Thomasetti Defence Ltd., 2023a), is a hydrocode developed by Thornton Thomasetti Defence Ltd specifically for blast analysis. It operates solely on the GPU, offering faster performance compared to standard CPU hydrocodes. It claims to be at least ten times faster than CPU hydrocodes while being more cost-effective as it does not require large-scale computer clusters. WALAIR++ supports larger models by using data migration procedures. It is compatible with both Windows and Unix operating systems.

As a CFD code focused on blast analysis, WALAIR++ solves compressible Euler equations on a structured Cartesian grid using the finite-volume method. It provides the option to use either an ideal gas system or a multimaterial system with an ideal gas equation of state for air and the JWL equation for explosive products. It can handle basic symmetry-based analyses to reduce computational time and allows high-resolution simulations in the early stages of a blast event without excessively large 3D models (Thornton Thomasetti Defence Ltd., 2023b).

WALAIR++ finds applications in cityscape blast modelling and internal detonation events, typically involving complex geometries. It utilises a mesh generation algorithm to handle such

geometries, claiming to execute quickly even on meshes with hundreds of millions of cells. The user needs to provide stereolithographic (STL) files that represent the geometry for analysis.

WALAIR++ offers various features to model blast scenarios, including arbitrary charge shapes, sympathetic detonation events, breakable obstacles using pressure-impulse curves, human injury measurements, and damage quantification of structures using pressure-impulse curves.

The hydrocode provides output in the form of VTK files which are compatible with the ParaView visualisation software. Users can request specific spatial variables for output, such as pressure, density, and velocity. It also provides pressure time histories from user-placed pressure gauges.

WALAIR++ claims to have undergone extensive validation against experimental results and other comparable hydrocodes. There are currently no published works for WALAIR++.

The analysis process using WALAIR++ is designed to be user-friendly and flexible. All interactions occur within a text-based input file, following a standardised format.

Experimental setup

CFD models evolve from a theoretical understanding of the dynamics along with the physical equations that govern an event. For complex scenarios such as the prediction of a pressure time history from a high explosive event, these CFD models require calibration and fine-tuning based on real world trials data (Jing et al., 2013). The CFD models being studied here are compared against each other in terms of run time, domain size artefacts, etc. but to demonstrate that the actual predictions they have made are valid, a full-scale live experimental trial was performed. The trial was designed to match the simulation setups as closely as possible. TNT was not available for this test because of cost, so a high explosive of similar performance was used. The charge used for the experiment was 91 kg of Nitromethane contained in a spherical ABS plastic container, the TNT equivalence of 1.1 was used according to the studies by Witty and data from Viper::Blast (Stirling, 2022; Witty, 2023). As in the simulations, the charge centre was elevated 1.2 m above the ground. The explosive was located on a 100 m × 100 m concrete test pad and supported on a polystyrene stand, which was placed on a 2 m × 2 m × 75 mm thick steel plate to prevent secondary fragment generation, the polystyrene was not modeled. The reflective target was placed perpendicularly to the radius of the arena at a distance of 25 m from the charge centre. This experimental setup would be very similar to an arena firing as described in ISO 16933:2007 EXV25 (ISO/TC 160, 2007). The reflective target was constructed from five concrete blocks each of dimension 3.5 × 0.75 × 0.81 m (width × depth × height) stacked on top of each other. Two concrete box culverts of internal dimension 3 × 3.5 m (depth × height) and external 1 × 3.5 × 4 m (width × depth × height) were placed on either side of the concrete blocks to form a nominally flat reflective concrete wall, 7.5 m wide and 4 m high. This setup is shown in Figure 1.

Live experimental trial

Instrument details. Seven PCB pressure sensors, model 113B26, were mounted into the concrete structure, four in the centre of the front face and three centrally on the ground at the rear. Each pressure sensor was connected to a PCB 003-EB-003-AB cable, connected to a 100 m length of RG58 C/U coaxial cable followed by a PCB 482C15 signal conditioner. Data were recorded on an HBM GN815 card within a GEN7tA chassis located next to the signal conditioner. Pressure gauges,



(a) Front view



(b) Rear view, also showing location of gauge position at base

Figure 1. Experimental target representing the gauge block, 7.5×4 m (width $\times$ height). (a) Front view (b) rear view, also showing location of gauge position at base.

signal conditioners, and data acquisition systems had a calibration traceable to national standards. Due to the long cable length, the signal conditioner was adjusted to produce a 20 mA output current. Using the methodology described in the PCB white paper, “Signal Transmission on Long Cable Lengths with ICP® Sensors” (Christman, 2021), this set-up would allow the highest possible full-scale output frequency to be around 60.5 kHz. The data recorder details are as follows:

Data acquisition

The Nyquist Sampling Theorem is a fundamental concept in signal processing and digital data acquisition. It states that to accurately reconstruct a continuous signal from its discrete samples, the sampling frequency must be at least twice the highest frequency component present in the signal. The Nyquist theorem ensures that no information is lost during the sampling process and allows for a near-perfect reconstruction of the original signal. If the sampling frequency is below the Nyquist rate, aliasing occurs, leading to distortions and loss of information in the reconstructed signal (National-Instruments, 2024). Although the frequency of the component wave may be recorded correctly, the amplitude may not have been, based on the bandwidth of the system. The bandwidth of a system is defined as the frequency below which the signal has not been attenuated by more than -3 dB by, in this case, the analogue-to-digital converter in the logger. The potential loss in amplitude is given by equation (1) (National-Instruments, 2024) where the amplitude error is in % and R is dimensionless.

$$AmplitudeError = 100 \left(1 - \frac{R}{\sqrt{1 + R^2}} \right) \quad (1)$$

where $R = \frac{Bandwidth}{samplefrequency}$

To account for these potential issues, National Instruments recommends a sample frequency of three to five times the highest frequency of interest in the measured signal (National-Instruments, 2024). To find the highest frequency of interest, we need to know the range of frequencies that could be used to create the Friedlander waveform (Figure 2). This can be performed using a Fast Fourier Transform (FFT). FFTs are algorithms that efficiently compute the discrete Fourier transform (DFT), allowing the conversion of a signal in the time domain to its representation in the frequency domain (Oppenheim et al., 2010). Figure 2 shows a Friedlander curve that has a peak pressure, shape, and impulse similar to the time history curve generated by a 100 kg TNT_{eq} explosive charge 25 m from a reflective structure. Figure 3 shows the FFT of this curve (Friedlander curve generated by EmBlast (Bassam, 2020)). The equation used is the modified Friedlander equation (Friedlander, 1946). This is an altered version of the original equation (2)

$$p = p_0 e^{-\frac{t}{t_d}} \left(1 - \frac{t}{t_d} \right) \quad (2)$$

Here p is the pressure at time t, p_0 is the peak pressure and t_d is the positive phase duration. In modified form, equation (3) increases the validity of the approximations for maximum pressures greater than 1 bar(g) (Demby, 2018). This is achieved by including the decay factor *beta*.

$$p = p_0 e^{-\beta \frac{t}{t_d}} \left(1 - \frac{t}{t_d} \right) \quad (3)$$

Figure 3 shows that the primary frequencies of the FFT for both the real data and the Friedlander curves are nominally close. The primary frequencies within the curves are those < 1000 Hz, but small contributions are still made by the higher frequencies, as expected from a FFT of a non-sinusoidal wave, as long as the frequency pots are available.

A power spectrum provides information on the relative power contribution of each frequency component in the signal. Integrating the power spectrum obtains the cumulative power distribution

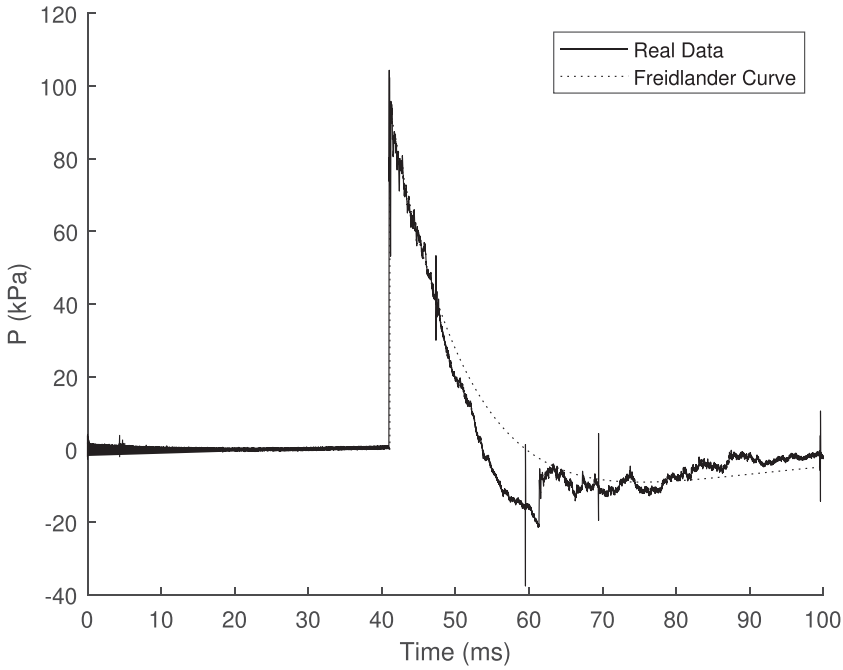


Figure 2. Experimental data from a reflective gauge mounted in a 3×3 m wall from a 91 kg Nitromethane charge at 25 m, along with a modified Freidlander equation plot with $p_0 = 94$, $t_d = 18.5$, $\beta = 1.18$, (the time axis has been offset to have the time of arrivals coincide).

across frequencies. This integration reveals the total power contained within specific frequency ranges and helps to assess the overall energy distribution of the signal. From this it can be determined that 99.5% of the energy in the signal is obtained from frequencies below 7000 Hz.

Using the value of three to five times the maximum frequency of interest to determine the bandwidth according to [National-Instruments \(2024\)](#), this gives a bandwidth range of 21 to 35 kHz. National Instruments also recommends that, to account for the Nyquist theorem, the sampling rate be 5 times the maximum frequency of interest ([National-Instruments, 2024](#)). To allow for some conservatism, it would be recommended that a minimum sample speed of 100 kHz with a minimum 35 kHz bandwidth be used to capture data at this scaled distance. Data acquisition systems with these specifications are readily available ([Emilio, 2013](#)).

Previous work performed in this area on data acquisition requirements by [Du et al. \(2015\)](#) showed that for a scaled distance of $5 \text{ kg} \cdot \text{m}^{-1/3}$ the required bandwidth of the sensor to obtain an error of 1% is 55 kHz, and the approach of PAS 300: 2018 ([British Standards Institution and Centre for the Protection of National Infrastructure \(Great Britain\), 2018](#)) both correlate well with these results.

When specifying the DAQ one final component needs to be considered, the number of measurement levels or bits (n) the system has. Where the number of available measurement levels is given by 2^n . Note that this is not the same as sensitivity, which is the smallest change an oscilloscope can measure. For the PCB 113B26 pressure sensor, the manufacturer's stated measurement range is 3450 kPa, with a resolution of 0.014 kPa. Measurement of the full positive range of the sensor at the smallest resolution would require $\sim 250,000$ measurement levels, corresponding to a bit resolution

of 18. More practically, recording the sensor from -1000 to 3450 kPa in 0.1 kPa steps would require a minimum of a 16 bit DAQ, which again are commercially available, such as the HBM GN815 in the specification [Table 2](#).

Simulations

This paper continues the work performed by [Chester et al. \(2024\)](#). In his paper Autodyn, blastFoam, EmBlast, ProSAir and Viper::Blast were used to simulate both a 15 kg TNT_{eq} charge at a stand-off of 6 m and a 100 kg TNT_{eq} charge at 25 m. Both of these simulations used the same reflective structure as defined in this paper, although the comparison was performed for gauges directly facing the charge. The manner in which each software package was set up is reproduced here and summarised below. Chester used a maximum run-time criterion of 12 h in the simulations to replicate an overnight run scenario. This approach demonstrates to readers the quality of data that could be obtained by a user on a realistic timescale. In this paper, the resolution of each software was

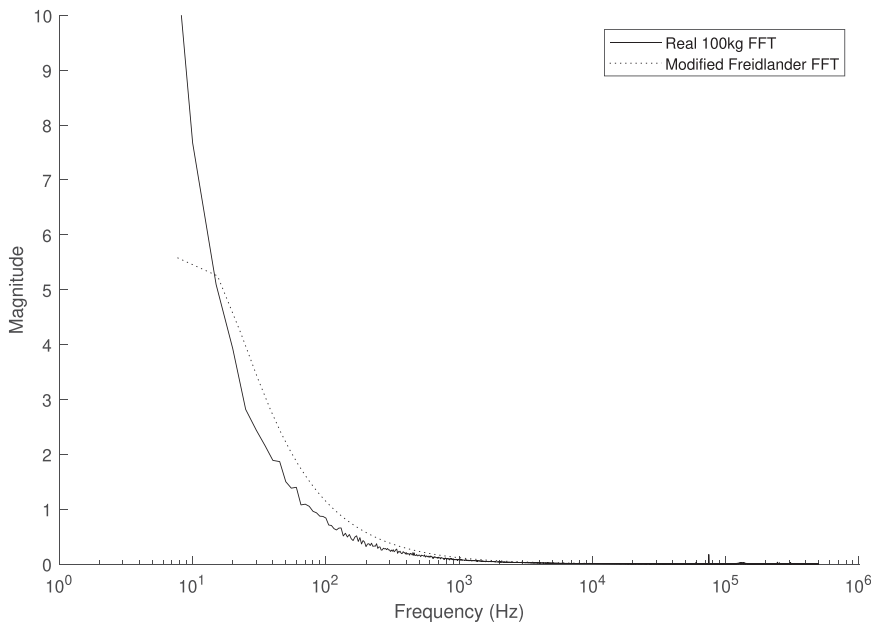


Figure 3. Fast Fourier transforms of the data from a 100 kg TNT eq charge at 25 m and it's associated Freidlander curve.

Table 2. Data acquisition system properties.

System	Bit rate	Sample speed (MHz)	Bandwidth (MHz)
Maximum	18	2	0.95
Trial	18	1	0.95

increased until a failure in the computer resources was encountered; this provides data on the maximum limit for the system.

Simulation methodology. Simulations were setup to replicate those from the live experimental trial. A 100 kg spherical charge of TNT was placed at a height of 1.2 m from a reflective ground surface at a distance of 25 m from the front reflective structure, as shown in [Figure 1](#). All other boundaries were set to transmissive. To allow a direct comparison between each package, as many variables as possible were standardised and are listed as follows:

- Reflected wall material
- Gauge locations
- Domain size
- Cell/mesh size in 1D, 2D and 3D domains
- CFD remapping timings and methodology
- Explosive type
- Charge height
- Boundary conditions

To ensure the models were setup using best-practice techniques, advice from the authors along with training courses were followed.

From the chosen software selection, only ProSAir does not support CAD file import, so the reflective structures were created using primitive shapes to create unused cell regions, a feature that is universally supported. These unused cell regions act as perfect reflectors. The gauges were placed to match the position of the gauges in the live experimental trial at the central base location at the rear of the wall, see [Figure 1](#).

Domain studies were carried out in each package to find the point at which decreasing the domain size further would start to affect the accuracy of the positive phase data, as previously used by [Li et al. \(2019\)](#). Initially a large domain that provided a 10 m clearance from the edge of the gauge block, to the rear, sides, and top of the domain was run to act as the baseline case. This formed a domain size of $38.45 \times 27.5 \times 13.95$ m, as shown in [Figure 4](#). The domain was then decreased in 1 metre steps along a single axis to reduce the clearance from the edge of the gauge block to the domain walls. This process was repeated until deviations in the positive phase were observed compared to the 10m baseline case. Then, this process was repeated for the remaining two axes.

Incorporating this domain size with the other standardised simulation variables, further runs were then performed. For each run, the cell/mesh size was decreased until the programme or computer crashed. A review of the achievable mesh sizes, run times, and comparison to the live test data is then made.

The following sections outline the approach taken for each simulation package and are the same as used by [Chester et al. \(2024\)](#), which also includes further detail.

Only ProSAir and Viper::Blast suggest particular resolution settings in the 1D and 2D domains in their software manuals ([Forth, 2022a](#); [Stirling, 2021](#)). Both recommend that the charge should have 50 cells across its spherical radius in the 1D domain. For the 2D domain ProSAir suggests a cell size of 100^{th} of the charge height for charges close to the ground to ensure a good resolution of the shock-ground interaction, as used in this study. Taking the density of TNT to be $1630 \text{ kg} \cdot \text{m}^{-3}$, this implies a 1D cell size of 4.89 mm for the 100 kg charge. The charge is elevated by 1.2 m, so the 2D mesh size was initially set to 12 mm.

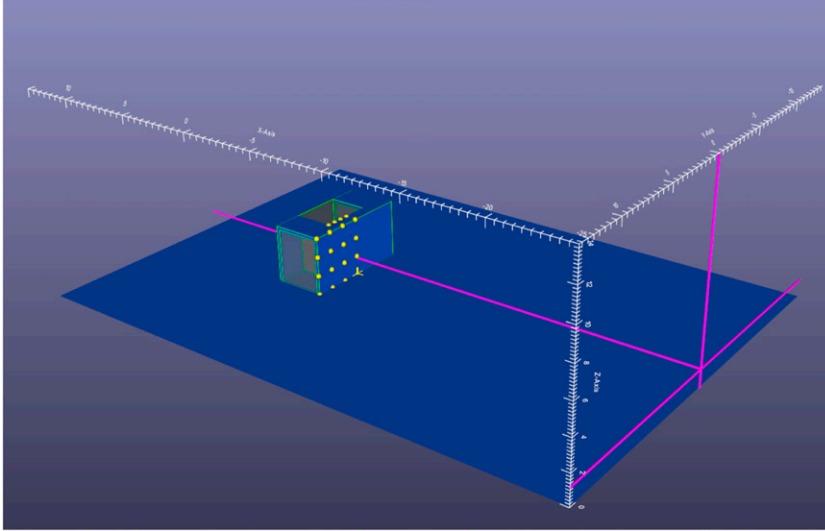


Figure 4. Representation of initial dimensions for domain study with a 10 m clearance from the edges of the gauge block to the domain wall, as shown in Viper::Blast.

In the 2D domain, the mesh size criteria equal to 100^{th} of the charge height for low charge heights and a large domain have the effect of very high cell counts that can exceed the capacity of the simulation computer or software. This was realised in Autodyn, blastFoam, and ProSAir. The minimum mesh size that could be run in these packages in the 2D domain was 40 mm. To provide consistency, 40 mm was used across all software.

Autodyn. The methodology used in this study is taken from the Ansys Autodyn ‘Workshop 12 Urban Blast’ and ‘Workshop 13 Urban Blast 2’ tutorials (Ansys, 2023). These training courses demonstrate step by step how to simulate the detonation of an explosive in air and then remap these data into a larger domain with idealised reflective structures. Both these training courses have the charge at ground level and perform the entire simulation in a single remap. This methodological approach needed to be adapted for this paper as the charge in this paper is raised. As stated previously to provide a good shock ground interaction, a fine mesh size is needed that would be too fine to be used to create a full 3D domain; hence a 1D-2D-3D remap is required. Initially, a simulation was conducted using a 1.2 m long one-dimensional (1D) multi-material Euler wedge filled with air. The domain length is set to 1.2 m to represent the height of the charge from the ground. Using the 4.89 mm cell size, this sets the domain length at 245 cells. A TNT sphere with a radius of 244.7 mm was placed at the origin. This represents a charge of 100 kg, assuming a density of $1630 \text{ kg} \cdot \text{m}^{-3}$. The detonation point of the charge is set at the origin of the model, which is also the centre of the charge. The simulation was run and terminated when the shock front approached the domain edge at 0.25 ms. The internal energy of the air was set to $2.068 \times 10^5 \text{ J/kg}$, representing an atmospheric pressure of 1 bar.

Then a new model with a two-dimensional (2D) ‘Euler Ideal Gas’ square was created. As the charge is elevated 1.2 and 25 m from the target, the x dimension was set to 24.9 m so that on remapping pressure data would not be inside the reflective structure. To allow the blast to meet both domain walls at approximately the same time, the height of the domain was set to 1.2 m more than

the standoff at 26.2 m. These same domain walls were set to be transmissible. With a cell size of 40 mm the Autodyn i and j axes had 656 and 624 cells, representing the vertical and ground axes, respectively. The 2D wedge file was imported and placed 1.2 m up the i-axis from the origin. The simulation was run for 40 ms, where the shock wave was then close to the domain edge.

A 3D Euler Ideal Gas Cube was then created, set to the domain and mesh size that was dependent on the run, and filled with Air. The unused index space option was then used to build the reflective structure from multiple unused space cuboids. The domain walls were set to be transmissible by setting the boundary condition to the Flow_Out option and applying it to the exterior walls of the domain, with the exception of the x-y plane that represented the ground. Gauges were placed in the first empty cell at the base and centre of the back face of the wall. The 2D asymmetric model was imported and placed halfway along the origin domain wall. The simulation was then run to 100 ms, which was sufficient for the full negative phase of the blast wave to be recorded.

blastFoam. The blastFoam installation package includes tutorials and validation cases in its installation (Heylmun et al., 2022). A range of tutorials is provided to help train the user; these are simple, fast-running models. The validation cases are highly optimised to generate the most accurate simulation with respect to the data to which they are being validated.

The methodology of the code setup is initially based on two tutorials called ‘mappedBuilding3D - sector’ and ‘mappedBuilding3D - wedge’. These are 1D-3D or 2D-3D remapping simulations that detonate a 25 kg cylindrical C4 charge against an L-shaped wall at a standoff of 2 m. The two tutorials were combined by remapping the 1D sector to the 2D wedge and then into 3D, giving the required 1D-2D-3D remapping solution.

As these tutorials are coarse, fast-running models, the validation case called “KingeryBulmash - wedge” was used to refine the inputs. In the validation case, a 10 kg hemispherical TNT charge is detonated within a 16 m wedge domain. The CFD values for TNT, the equation of state (EoS) used, etc. were transferred to the 1D-2D-3D model. The same values were then used as per the Autodyn setup, running the 1D model for 0.25 ms and the 2D model for 40 ms. In the 3D model the gauge block was created from primitive shapes as per the other software. Adaptive meshing and mesh refinement were used, but only to allow a halving of the initial cell dimension. This halved cell size would then be the same as the uniform cell size used in the other software packages.

ProSAir. The ProSAir package includes a detailed user guide for its correct implementation, along with recommendations on mesh sizes (Forth, 2022a). ProSAir is setup via a graphical user interface, which has four tab accessed data entry screens. The first tab page covers general settings and allows the number of cores to be specified, remapping regimes, etc. The recommended ‘Switch time’ which specifies the simulated time where the transition from first-to-second-order accuracy occurs in ProSAir was set to 0.0056 s for the 100 kg charge following the guidance in the user guide. Other options in the general settings atmospheric pressure and temperature were left at their default values of 101325 Pa and 288 K, respectively. The second setup tab details the 1D setup file; here the charge size of 100 kg was entered along with the explosive type of TNT, the cell size of 4.89 mm, and the domain size at 1.2 m. The problem time was set to 1 s, as ProSAir can stop the simulation early when the blast reaches the domain wall by setting the boundary to Stop, so does not have to be done manually as per Autodyn and blastFoam. The third tab is for the entry of the 2D simulation data; here the height of the charge, cell, and domain sizes are entered to the values stated previously, and the boundary conditions can again be set to stop. The final page sets up the 3D model domain size, problem time, cell size, etc. The boundary conditions were set to transmissive, apart from the ground

that was reflective. The gauge block was created using primitive cuboids and the gauges were placed at the base centre of the rear wall.

Viper::Blast. The Viper::Blast simulations followed the specified 1D-2D-3D remapping, and each of these dimensions is set via a separate tab in the interface, similar to ProSAir described above. The default values for TNT were used, as well as for the pressure and temperature of the atmosphere. The dimension tabs have the option to employ a JWL EoS, but to maintain consistency with the other programmes, this was kept to the default. The simulation employed the ‘AUSMDV, single GPU, out-of-core local’ solving method. This technique is specifically designed for large models that exceed the combined memory capacity of the GPU and the host PC. The JWL mode was excluded to maintain consistency with other programmes, as it is not universally available. No additional non-standard parameters were required to be modified for the simulations.

WALAIR++. WALAIR++ is used through a single text file that is referenced from a command line interface along with an executable. The inputs and defaults required for the text file are clearly detailed in the user manual. Values are written next to each command, for example: CHARGE-MASS 100 TIME 0.1 Which states a charge weight of 100 kg and a simulation time of 0.1 s. All values were set according to the recommendations, such as the CFL numbers, at 0.5. The gauge block was recreated using primitive shapes using the same methodology as for the other software. Initially, a solid cuboid of unused space is created, followed by filling sections of the cuboid with Air to form the final shape. This is coded in WALAIR++ as follows to form the gauge block with the base of the front face at the origin:

```
OBSRECT3D 0 -3.75 0 3.45 3.75 3.95
OBSNEGRECT3D 0.225 -3.75 0.225 3.225 -1.75 3.725
OBSNEGRECT3D 0.225 1.75 0.225 3.225 3.75 3.725
OBSNEGRECT3D 0.75 -1.75 0 3.45 1.75 3.95
```

Each of these commands is in the format xmin ymin zmin xmax ymax zmax and defines the coordinates of the cube to be created or subtracted.

Results

Live experimental trial

A live experimental trial was conducted on 8th February 2023 at DNV Energy Systems Spadeadam. The weather at the time of firing is provided in Table 3. The pressure data recorded from the front face were compared with test data from around 200 previous trials and shown to be consistent showing that this charge performed as expected. Data from this trial are shown in Figure 5. The pressure time history has various features that can be determined from the simulation images and timings, as shown in Figure 6. Point ‘1’ in Figure 5 shows where the pressure wave has struck the

Table 3. Weather conditions at the time of firing.

Test date	Temperature (°C)	Pressure (mbar)	Relative humidity (%)	Wind speed (m·s ⁻¹)	Wind direction
08/02/2023 13:00	6	990	74	9	South West

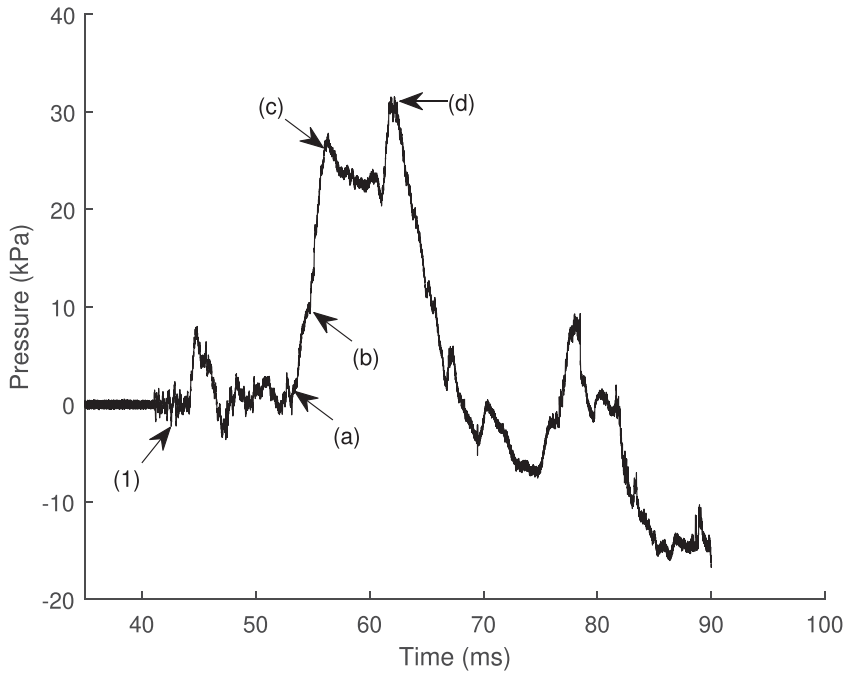


Figure 5. The averaged data from three pressure transducers located at the rear central base of the gauge block from a 100 kg TNT_{eq} explosive 25 m from the face of the gauge block. (1) Shock transmitted through the concrete gauge block from the blast wave striking the front face of the reflective structure. (a) Arrival of side pressure waves. (b) Arrival of top wave. (c) First peak from combined side and top waves. (d) Second peak from superposition of all wave including reflections from the rear of the culverts.

face of the gauge block, sending a pressure pulse through the concrete block that has registered on the pressure transducers located on the rear face. In the simulations, the reflective structure is created from unused space and has no material properties; hence, this feature is not simulated. Autodyn would be the only package studied here that may be able to predict this pulse as it can include material properties. Three types of wave are created by the interaction of the gauge block with the incident pressure wave. The first waves that arrive at the sensor are the waves that flow around the sides, enter through the culvert sections, and coincide at the centre of the rear face of the gauge block, as seen in 'a' in Figures 5 and 6. A second wave that has passed over the top of the gauge block and travelling down its rear face starts to combine with the first at 'b'. The point where they fully superimpose is shown in Figure 6 and forms the first of the two main peaks at 'c'. The final peak, 'd', is from the combination of the side waves, top wave, and reflections from the inside face of the culverts, all superimposing along the vertical access of the rear face of the gauge block. The overpressure then decreases as the pressure escapes from the structure. As the secondary pressure wave interacts with the gauge block, a subsequent peak is formed around 78 ms in Figure 5. As the domain sizing for this study is focused on the positive phase of the pressure pulse, details of this secondary peak will not be considered further. The peak pressure of the live test and the specific impulse of the positive phase, ignoring concrete shock data, are given in Table 4.

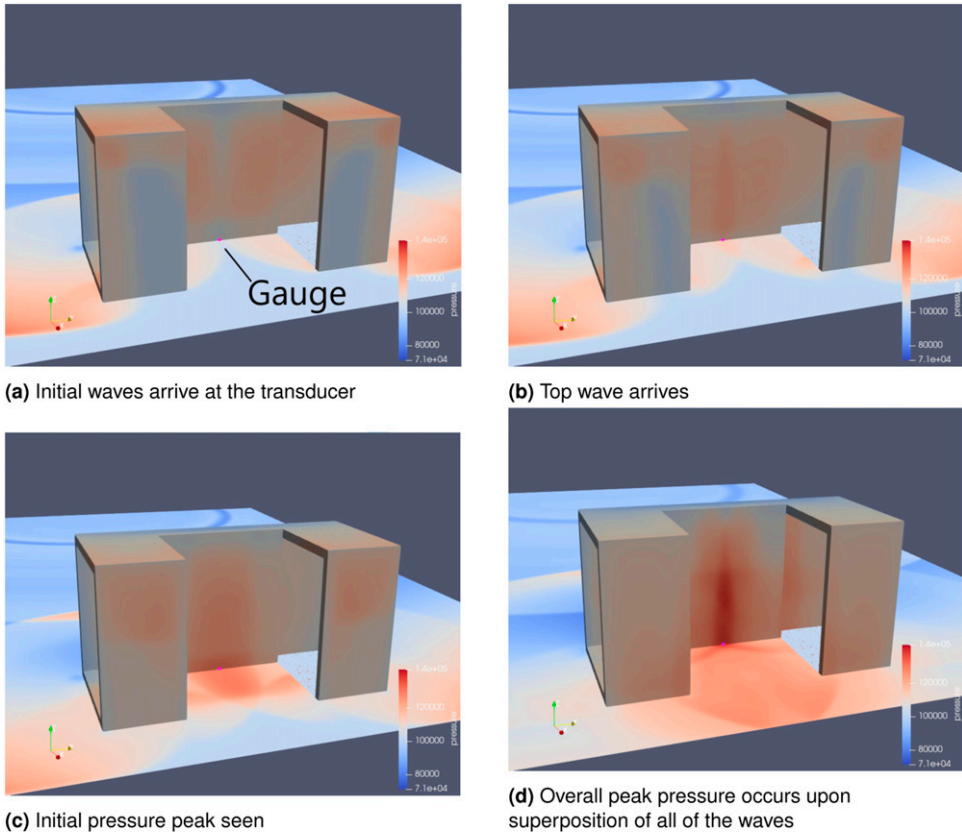


Figure 6. Pressure time history events from a live experimental trial, as simulated by Viper::Blast and displayed in ParaView. Pressure gauge placed centrally at rear base of the wall. (a) Initial waves arrive at the transducer (b) top wave arrives (c) initial pressure peak seen (d) overall peak pressure occurs upon superposition of all of the waves.

Table 4. Live test results for average of rear three pressure transducers.

Test date	Peak overpressure (kPa)	Specific impulse (kPa·ms)	Time of arrival (ms)	Positive phase duration (ms)
08/02/2023 13:00	31.6 ± 1	263 ± 4	52 ± 0.1	16 ± 0.25

Simulations

Run time. The run time of each simulation is detailed in Figure 7 and Table 5. Figure 8 shows for clarity just the GPU based Viper::Blast and WALAIR++, as they do not show up clearly in overall Figure 7. A wide range of run times across the software is seen, ranging for the 110 mm mesh from 17 s to over 9 h. The simulated experimental time was 100 ms for all runs documented in this paper;

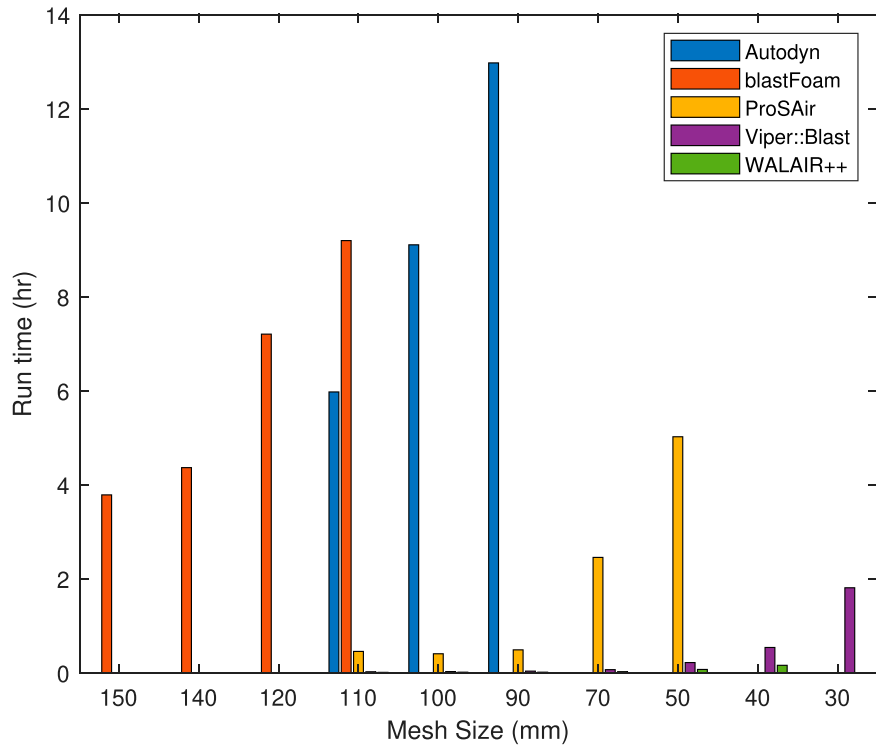


Figure 7. Showing the relative run times for each software across a range of mesh sizes.

Table 5. Run time summary.

Program	Time to run domain mesh size (hr:mm:ss)									
	150	140	120	110	100	90	70	50	40	30
Autodyn	–	–	–	5:58:00	9:06:00	12:58:00	–	–	–	–
blastFoam	3:46:51	4:20:08	7:14:00	9:11:33	–	–	–	–	–	–
ProSAir	–	–	–	0:27:00	0:24:00	0:29:00	2:27:00	5:00:60		
Viper::Blast	–	–	–	0:01:01	0:01:15	0:01:50	0:03:43	0:12:46	0:32:02	1:48:16
WALAIR++	–	–	–	0:00:17	0:00:23	0:00:29	0:01:08	0:03:59	0:09:16	–

the computer and programme setups were as similar as possible, so the range of times seen here should accurately present their difference in performance for this specific case, under the proviso that alternative setup options were available in some cases. General inverse cubic trends are seen in the data when comparing run time versus cell size, as expected due to the cuboid nature of the domain. Runs were not performed in each of the different mesh sizes. Initially, an upper mesh size limit of 110 mm was used and provides results for each of the packages. Within the limits of the simulation PC, blastFoam could not run at a mesh size of less than 110 mm, so additional runs were performed at courser meshes to provide additional results. By continually reducing the mesh size in each run until ultimately crashing the PC or programme, this finds the limit for the

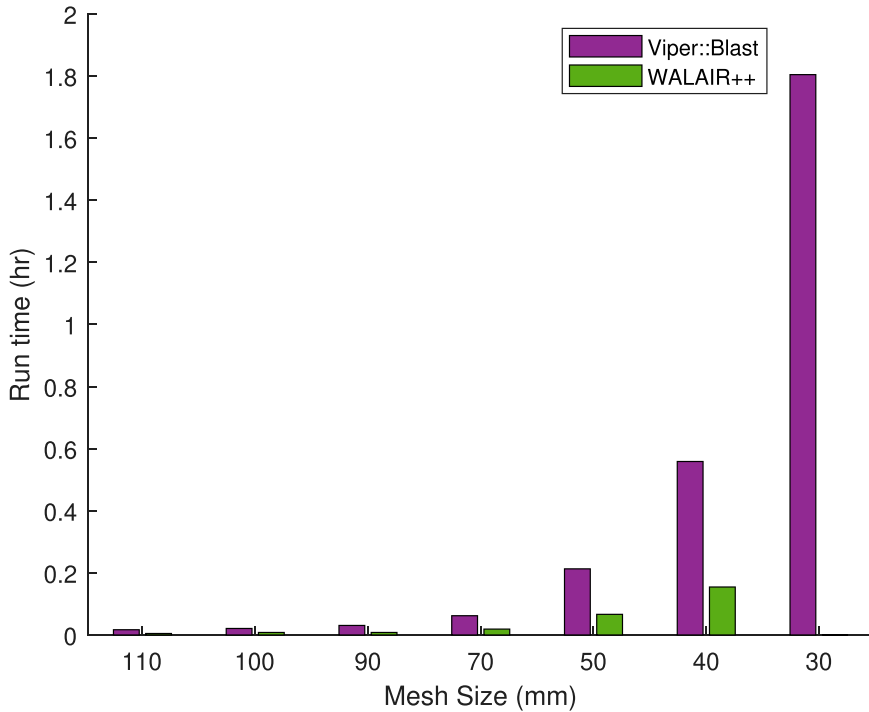


Figure 8. Showing the relative run times for Viper::Blast and WALAIR++ across a range of mesh sizes.

simulations in this set-up. These points can be seen in [Table 5](#) where results stop for smaller mesh sizes.

Domain study. The size of the simulated domain was reduced along a single axis in steps of one metre until deviations in the positive phase were observed compared to the largest domain run. This large domain had a clearance of 10 m from the edges of the simulated structure to the walls of the domain. An example of this domain effect is shown in [Figure 9](#). Once this effect was seen on the first axis, this dimension was then fixed to the last value where interference was not witnessed. The same process was then performed for the remaining two axes, each time ensuring that the positive phase was not affected compared to the initial ten-metre clearance model.

The results of the reduction in the size of the domain in metre steps from the initial $38.45 \times 27.5 \times 13.95$ m domain show that, for each software, once the clearance from the edges of the structure was less than 2 m in width, 7 m in length, and 6 m in height, effects on the positive phase were seen. The start of the domain boundary effect occurred at the same domain dimensions in each of the five packages studied. These 2, 7 and 6 m clearances imply distances from the gauge location to the domain walls of nominally 4.5, 10.8 and 10 m in the length, width, and height dimensions, respectively. The length-domain interaction distance is approximately half that of the others; this is because the blast wave has to interact with the rear-domain wall, and the effective wave then travels back a further 4.5 m before arriving at the transducer location. [Figure 10](#) shows the 2D data re-mapped in the 3D domain, occurring at $T = 40$ ms. This is the first instance where the interaction between the blast wave and the domain wall can occur, at a point nominally 10 m away from the pressure sensor. The interaction of the effect of the domain with the data from the rear sensor occurs

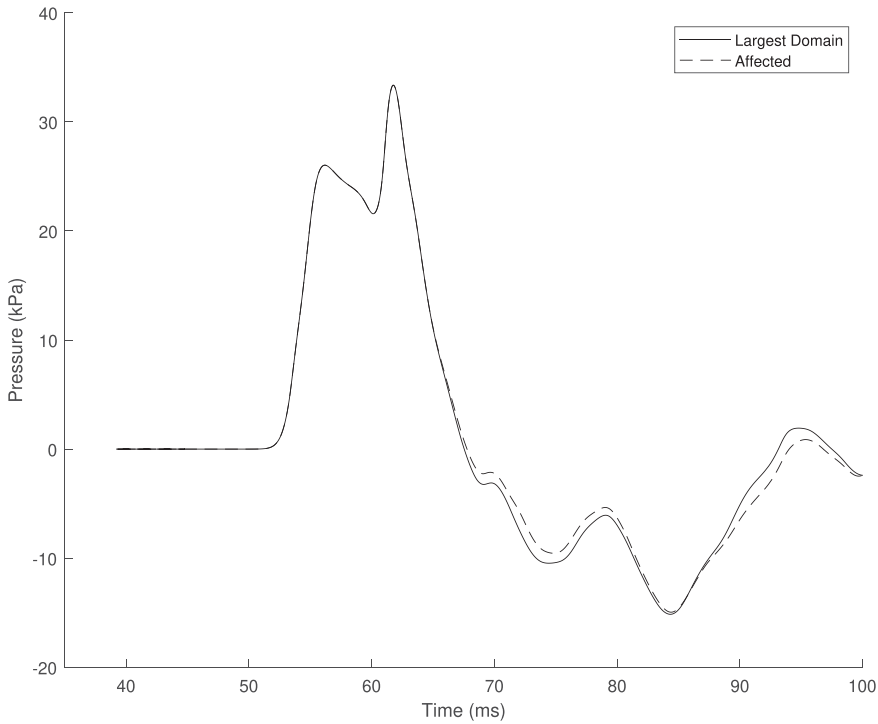


Figure 9. Showing the effect of reducing clearance in the y-axis from 10 m to 5 m where effects on the positive phase can start to be seen.

at the end of the positive phase at $T = 68$ ms. For a wave to travel 10 m in 28 ms implies a velocity of $\sim 360 \text{ m} \cdot \text{s}^{-1}$, closely matching the velocity of a shock wave having travelled 35 m from a 100 kg TNT_{eq} explosion, of $371 \text{ m} \cdot \text{s}^{-1}$, as given by Emblast (Bassam, 2020).

Mesh analysis. Each of the packages was run with a decreasing mesh size until failure, this was either that the software on repeated attempts displayed an error code or unexpectedly closed. The following section shows how, for each software, the level of refinement in the peak pressure and specific impulse varies with the reduction in cell size. This variation is shown in Figures 11 and 12.

To quantify the reduction in error as the mesh is refined in these numerical simulations, a Grid Convergence Index (GCI) has been employed. GCI is a commonly used metric in computational fluid dynamics that expresses the convergence error behaviour as a percentage. The Richardson extrapolation formula is used to calculate the GCI (NASA Glenn Research Center, 2021). Tables 6 and 7 show the GCI values, the convergence of the overpressure and the specific impulse values. They show that the simulations tend to converge consistently earlier for overpressure than for specific impulse. The expected decrease in GCI is seen with the reduction in mesh size.

Discussion

During this study, a wide range of run times was seen with comparative run durations varying by a factor of nearly 1600. The data was divided into three major sets categorised by their solve time. The

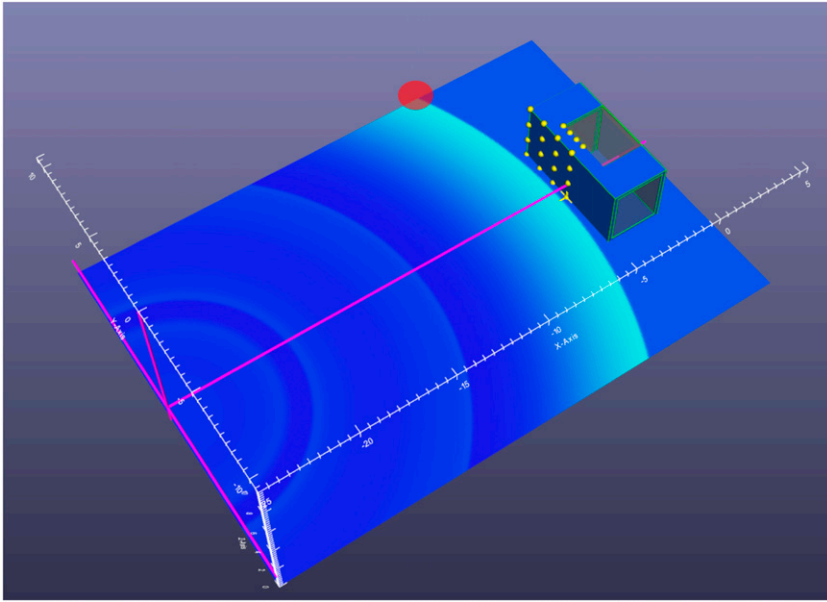


Figure 10. Image from Viper::Blast showing the imported 2D data remapped to the 3D domain, representing a time of nominally 40 ms post detonation. Red circle indicates the nearest and earliest point of interaction of the blast wave with the domain wall to the pressure sensors.

first, which includes Autodyn and blastFoam, performs the simulated scenario over a matter of hours, ranging from approximately 4 to 13 depending on the mesh size. The second set is ProSAir, where run times ranged from nominally 30 min to 5 hours, and finally the third set Viper::Blast and WALAIR++, where runs typically lasted from 17 s to 32 min. The first group of long-running programmes is not exclusively designed for the purpose of blast overpressure prediction, with Autodyn having many other capabilities and blastFoam being adapted from the widely capable OpenFOAM CFD tool. ProSAir was specifically designed for this type of simulation, and optimisation can clearly be witnessed by the comparative decrease in run time by a factor of between 13 and 26 against set one. The third set shows a decrease in run times compared to the slowest of set one (blastFoam) by a factor between 400 and 1600, both Viper::Blast and WALAIR++ are GPU based operating on CUDA capable graphics cards that have been shown to have computational advantages when compared to using a CPU alone (Kalaiselvi et al., 2017). Both have also been specifically designed to handle the type of simulations performed for this paper.

Comparing the effects of the reduction in mesh size in Figures 11 and 12, shows the variation in peak pressure and the specific impulse. For ProSAir, Viper::Blast, and WALAIR++ the variation in peak pressure gradually reduces until the 50 mm mesh size, where the variation is then very small with a GCI of 2% or less. In general, the refinement of the overpressure and the specific impulse through the cell refinement range studied here is of the order of 10%, demonstrating the benefit of such studies. Investigating the effect on the specific impulse in Figure 12 and Table 7 shows that the specific impulse has been fully resolved to 2% like the overpressure, with the specific impulse values continuing to increase in the last successful run before computer failure. The failures are likely due to insufficient memory.

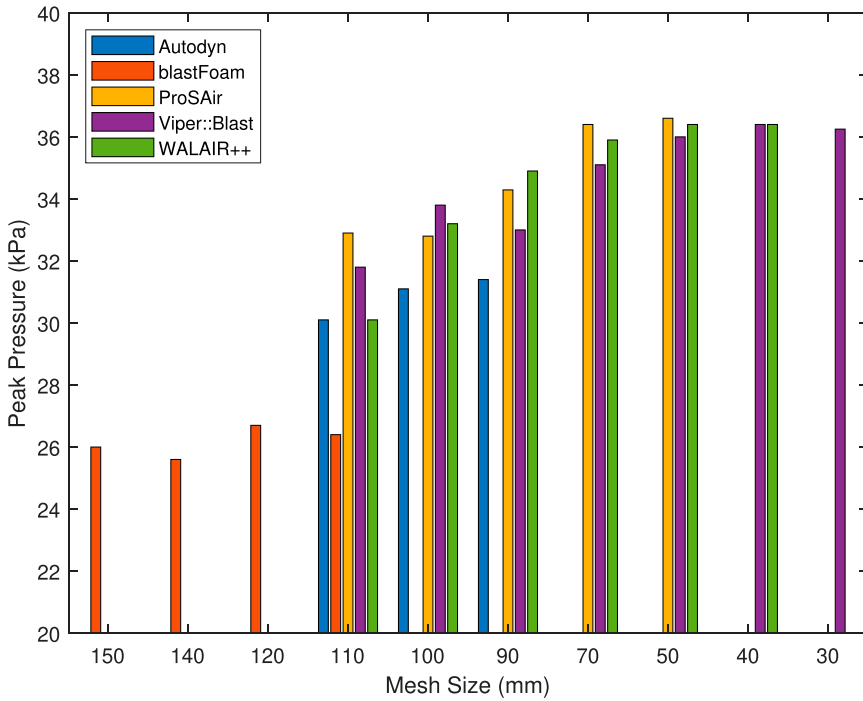


Figure 11. Showing the peak pressure for each mesh size and software. The live experimental value was 31.6 ± 1 kPa.

The root mean square error (RMSE) is a widely used technique to compare data curves within scientific research (Weaver et al., 2017). RMSE offers a quantitative gauge of the average deviation between the observed data points and the values predicted by a model or another data set. It is particularly valuable when evaluating the precision of the prediction or fitting models to empirical observations (Tyagi et al., 2022). Calculating an RMSE provides a value for the degree to which the data sets match the variations and trends in the actual data (Argyrous, 2011). A lower RMSE signifies a stronger alignment between the compared curves, implying a greater degree of agreement. In conjunction with the conventional analysis of Friedlander curve fits, involving the comparison of peak pressure, positive phase specific impulse, time of arrival, and positive phase duration, an RMSE fit was performed with the live experimental trial data and simulations. To facilitate the most precise curve-to-data comparison, arrival times were harmonised, as the explosive type of the live experimental trial was not TNT and there are differences in arrival times between explosives (Farrimond et al., 2022). This adjustment ensures the most accurate shape comparison. The RMSE values derived from this analysis are shown in Table 8, and range from 1.7 to 2.85 kPa. The 1.7 kPa value for Autodyn substantiates the alignment with the data, a trend visually apparent in Figure 13, and similarly the tightly arranged ProSAir, Viper::Blast and WALAIR++ at 2.2 kPa, and the worst fit of 2.85 for blastFoam. The RMSE assessment was conducted within the 53 to 69 ms timeframe, encompassing the complete positive phase. It should be noted that the pressure and specific impulse values of Autodyn were still changing with the mesh size and that the convergence value may well be different from the 90 mm mesh size values shown, since the GCI was 9% for overpressure and 5.9% for specific impulse.

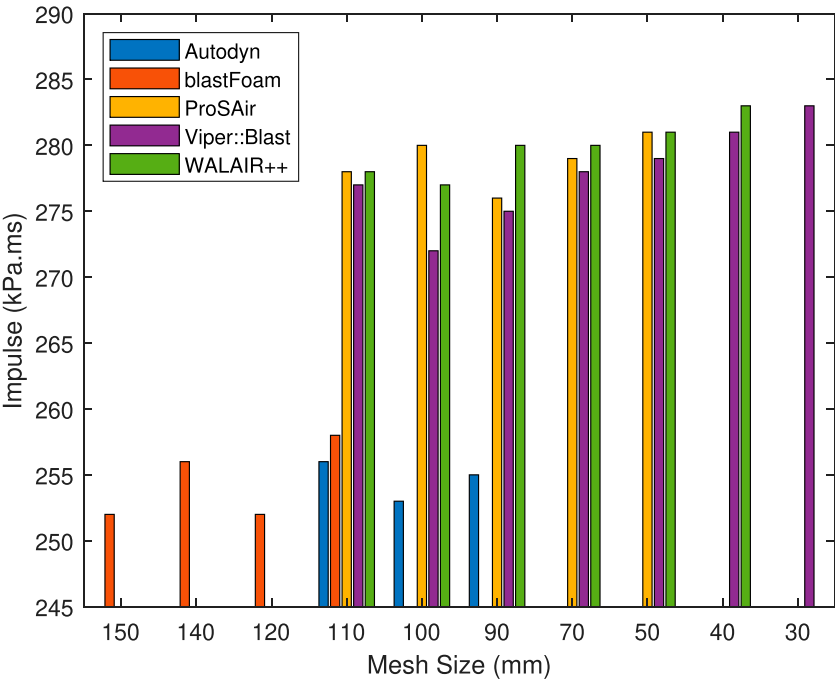


Figure 12. Showing the specific impulse for each mesh size and software. The live experiment value was 263 ± 4 kPams.

Table 6. GCI values for overpressure convergence.

Mesh range	GCI (%)				
	Autodyn	blastFoam	ProSAir	Viper::Blast	WALAIR++
150-140	—	24.5	—	—	—
140-130	—	19.2	—	—	—
130-120	—	23.4	—	—	—
120-110	—	21.6	—	—	—
110-100	10.0	—	105.0	68.7	86.8
100-90	9.0	—	40.5	21.3	42.6
90-70	—	—	19.9	21.0	10.2
70-50	—	—	1.6	6.7	3.1
50-40	—	—	—	4.4	0.2
40-30	—	—	—	1.5	—

*Note. For the GCI tables ‘—’ denotes values that cannot be calculated because this run was not performed. GCI: grid convergence index.

The simulations performed in this study show the range of results for the scenario. These results are detailed in [Table 9](#), and show that depending on the software used, after optimisation, variations in peak pressure of 28% and 10 % in positive phase specific impulse are still possible. The data in [Table 9](#) for the peak pressure and the specific impulse fall into two categories, Autodyn and

Table 7. GCI values for specific impulse convergence.

Mesh range	GCI (%)				
	Autodyn	blastFoam	ProSAir	Viper::Blast	WALAIR++
150-140	—	34.7	—	—	—
140-130	—	23.5	—	—	—
130-120	—	10.6	—	—	—
120-110	—	11.1	—	—	—
110-100	6.6	—	7.1	37.9	21.0
100-90	5.9	—	10.8	11.8	9.8
90-70	—	—	3.8	3.1	0.7
70-50	—	—	2.1	1.2	1.3
50-40	—	—	—	2.6	2.7
40-30	—	—	—	1.5	—

*Note. For the GCI tables ‘—’ denotes values that cannot be calculated because this run was not performed. GCI:

Table 8. Root mean square error comparison to live trial data.

Software	RMSE (kPa)
Autodyn	1.7
blastFoam	2.85
ProSAir	2.2
Viper::Blast	2.2
WALAIR++	2.2

blastFoam, both recording lower values than ProSAir, Viper::Blast, and WALAIR++, which form a very tight group, with minimal variation between them. There are two likely contributions to this pattern. The first is that ProSAir, Viper and WALAIR++ are all adaptations of the same underlying physical equations used in the legacy Air3D package, which have been optimised and hence reporting very similar data. The second factor is that both Autodyn and blastFoam’s final runs were based on significantly courser meshes; it has been previously shown by [Chester et al. \(2024\)](#) and others that decreasing cell size leads to an increase in peak pressure, and this is likely a factor here. We can further compare the variations in overpressure and specific impulse presented here with those found by Chester in the investigation of gauges placed on the front face of the gauge block facing the charge ([Chester et al., 2024](#)). The results for the same charge size and programmes were, respectively, 14% and 13%. This shows that for this more complex case, the peak overpressure prediction variation is twice that of the simple case. A similar variation between complex and simple is seen when comparing the specific impulse results. A possible explanation for the variance may be to do with the required increase in domain size for complex scenarios, where the positive phase duration in this case is 50% longer (~15 vs ~10 ms) than for the simple case due to multiple reflections. This increase in the positive phase duration requires that the domain is larger to prevent the effective wave, created by the interaction of the pressure wave with the domain wall, from reaching the simulated sensor within this positive phase, or pressure time history area of interest. By increasing the domain size, but keeping computational resources the same, only courser runs will be

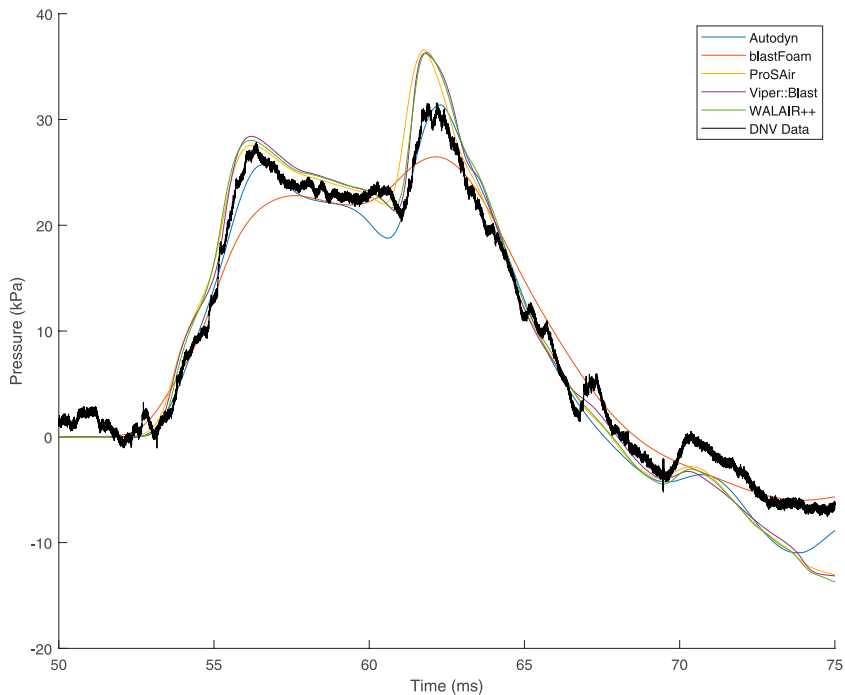


Figure 13. Showing the fit of the time of arrival adjusted simulation output against the live data, here the Autodyn curve has a RMSE of 1.7 kPa and the others 2.2 kPa.

Table 9. Simulation results for the finest mesh size achieved.

Software	Overpressure (kPa)	Specific impulse (kPa·ms)	Time of arrival (ms) ^b	Positive phase duration (ms) ^b
Autodyn	31.4	255	54.5	14.4
blastFoam	26.5	259	52.9	15.9
ProSAir	36.6	282	53.1	14.7
Viper	36.3	283	52.9	14.8
WALAIR++	36.4	283	52.9	14.6
Deviation ^a (%)	28	10	3	9

^aCalculated from the range of values divided by the maximum value and converted to a percentage.

^bAn error of ± 0.5 ms is possible here based on the interpretation of the time of arrival, as the simulated initial pressure rise is gradual due to limitations in cell size.

able to be simulated, potentially leading to a wider range in the final simulated mesh sizes in the range of programmes used, further highlighting any variation.

Conclusion

This paper investigated the performance of a range of blast software packages in simulating the propagation of shock waves from a high explosive charge in the far-field when interacting with a

complex structure. The software packages considered were Autodyn, blastFoam, ProSAir, Viper::Blast, and WALAIR++. These packages were compared with each other in terms of accuracy, run time, and ease of use. The simulated data were compared to a live experimental trial of the same physical setup, although with a different explosive. The paper explored the errors associated with the recording of pressure data from a live experimental trial. It shows that the errors in signal acquisition are $\sim 1\%$. This is before any further errors caused by the mounting of the gauge, or the challenges in physically recording the event due to the heat effect from fireballs, etc.

In terms of ease of use of software, the GUI of Viper::Blast and ProSAir were found to be the most user-friendly because both have a well-designed and intuitive GUI that is easy to navigate. The command-line interface (CLI) and multi-text file basis of blastFoam were found to be the least user-friendly and highly complex to use, requiring a greater level of technical expertise to use.

In general, the study found that there is no single blast software package that is clearly superior to the others in all respects. Where speed is critical, GPU-based packages such as Viper::Blast or WALAIR++ could be used, or for marginally slower performance ProSAir. Should further analysis of the structure be required such as deflection or damage, this may be possible in Autodyn. If cost is the significant factor, then blastFoam being free and open-source could be the preferred choice, or where ease of use is of primary concern, then ProSAir or Viper::Blast should be considered. Commercial prices are not freely available with the exception of blastFoam that is free and ProSAir that is vary from free for blastFoam,

As shown in previous studies, the accuracy of the different packages was affected by the size of the computational domain. In general, the larger the domain, the more accurate the results. This is because a larger domain allows for fewer interactions from the refraction waves created by the overpressure wave interacting with the domain boundaries. However, larger domains also require more computational resources, so there is a trade-off between accuracy and computational cost. Furthermore, this paper confirms that the accuracy of the different packages was affected by the mesh resolution. In general, the finer the mesh, the more accurate the results. This is because a finer mesh allows for a more accurate representation of the shock wavefront. However, depending on the size of the domain, finer meshes also require more computational resources.

When comparing run times, significant variation has been shown, with the GPU-based packages WALAIR++ and Viper::Blast being significantly faster than the CPU-based packages, in occasion by a factor of 1600. This is due to the fact that GPU-based packages can take advantage of the parallel processing capabilities of graphics processing units (GPUs).

The results of the study showed that there was a wide variation in the results of the different packages. The predictions of the peak pressure varied 28% and the specific impulse by 10%. This shows that the choice of blast software package can have a significant impact on the accuracy of blast predictions and that these variations can increase for complex scenarios where the time of interest in the pressure history is longer.

Acknowledgements

The author would like to express his thanks to the range team at DNV for collecting the experimental data, along with the authors of the software for their time and training.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iDs

Alastair Chester  <https://orcid.org/0000-0003-2617-890X>

Richard Critchley  <https://orcid.org/0000-0001-5423-9859>

References

- Angelides SC, Burgan BA, Kyprianou C, et al. (2022) Emblast: a software for calculating blast loads on structures from the detonation of energetic materials. *FABIG*. Available at: <https://eprints.whiterose.ac.uk/>.
- Ansys (2023) Introduction to ansys autodyn, ansys training. Available at: <https://www.ansys.com/training-center/course-catalog/structures/introduction-to-ansys-autodyn#tab1-1>.
- ANSYS, Inc (2023) Ansys autodyn, nonlinear dynamics analysis software. Available at: <https://www.ansys.com/products/structures/ansys-autodyn>.
- Argyrous G (2011) *Statistic for Research*. SAGE Publications Ltd, 214–233.
- Attia W, Elwan S and Kotb I (2021) Investigating the effect of the geometry of rc barrier walls on the blast wave propagation. *International Journal of Safety and Security Engineering* 11: 255–268.
- British Standards Institution and Centre for the Protection of National Infrastructure (Great Britain) (2018) *Civilian Armoured Vehicle : Test Methods for Ballistic And Blast protection*. Technical report. Available at: <https://www.en-standard.eu/pas-300-2018-civilian-armoured-vehicle-test-methods-for-ballistic-and-blast-protection/>.
- Ballantyne G, Whittaker A, Dargush G, et al. (2010) Air-blast effects on structural shapes of finite width. *Journal of Structural Engineering* 136(2): 152–159. DOI: [10.1061/\(ASCE\)ST.1943-541X.0000049](https://doi.org/10.1061/(ASCE)ST.1943-541X.0000049).
- Bangash M (2001) *Protective Construction: Design and Performance of Explosive Resistant Structures*. London: Thomas Telford Publishing.
- Bassam B (2020) Sci releases first software module of emblast for energetic materials blast information group (embig). Available at: <https://news-sci.com/sci-releases-first-software-module-of-emblast>.
- Burrows SJ, Forth SA and Sheldon RP (2021) Partial validation of cfd blast simulation in a cityscape environment featuring structural failure. *International Journal of Computational Methods and Experimental Measurements* 9(4): 365–380.
- Chester A, Critchley R and Hazael R (2024) A comparison of far-field explosive loads by a selection of current and emerging blast software. *International Journal of Protective Structures*. DOI: [10.1177/20414196241248192](https://doi.org/10.1177/20414196241248192).
- Christman M (2021) *Signal Transmission on Long Cable Lengths With icp Â® Sensors*. Technical report. PCB Piezotronics. Available at: https://www.pcb.com/contentstore/mktgcontent/whitepapers/WPL_76_SignalTransmissionLongCableLengthsICP%C2%AESensors.pdf.
- Cooper P (2018) *Explosives Engineering*. SE Alburquerque, NM: Wiley-VCH.
- Davison L and Silberschmidt V (2018) Fundamentals of shock wave propagation in solids. *Journal of Applied Mechanics* 85(12): 37–63.
- Demby DM (2018) *Blast Wave Generation and Interactions Declaration*. Technical report. Johannesburg, South Africa: University of the Witwatersrand.
- Du HM, He ZW and Ma TH (2015) Frequency domain characteristic of secondary instrument in the shock overpressure measurement system. *Baozha Yu Chongji/Explosion and Shock Waves* 35: 261–266.
- Emilio MDP (2013) *Data Acquisition Systems*. New York: Springer.

- Farrimond DG, Rigby SE, Clarke SD, et al. (2022) Time of arrival as a diagnostic for far-field high explosive blast waves. *International Journal of Protective Structures* 13: 379–402. DOI: [10.1177/20414196211062923](https://doi.org/10.1177/20414196211062923).
- Forth S (2022a) ProSAir 2022.11 user guide.
- Forth S (2022b) ProSAir computational blast loading tool. Available at: <https://www.cranfield.ac.uk/facilities/prosair-computational-blast-loading-tool>.
- Friedlander FG (1946) The diffraction of sound pulses; diffraction by a semi-infinite plane. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 186: 322–344. DOI: [10.1098/RSPA.1946.0046](https://doi.org/10.1098/RSPA.1946.0046).
- Handley CA, Lambourn BD, Whitworth NJ, et al. (2018) Understanding the shock and detonation response of high explosives at the continuum and meso scales. *Applied Physics Reviews* 5(1): 011303.
- Heylmun J, Vonk P, Shields B, et al. (2022) BlastFoam theory and user guide. Available at: https://github.com/synthetic-technologies/blastfoam/blob/master/BlastFoam_User_Guide.pdf.
- Heylmun J, Vonk P and Brewer T (2023a) blastFoam: a solver for compressible multi-fluid flow with application to high-explosive detonation. Available at: <https://github.com/synthetic-technologies/blastfoam>.
- Heylmun J, Vonk P and Brewer T (2023b) blastfoam release log. Available at: <https://github.com/synthetic-technologies/blastfoam/releases>.
- Heylmun J, Vonk P, Shields B, et al. (2023c) blastFoam github location. Available at: <https://github.com/synthetic-technologies/blastfoam>.
- ISO/TC 160 (2007) Iso 16933:2007 - glass in building “explosion-resistant security glazing” test and classification for arena air-blast loading. Available at: <https://www.iso.org/standard/38166.html>.
- Jha N and Kumar BSK (2014) Air blast validation using ansys/autodyn. *International Journal of Engineering Research & Technology* 3.
- Jing H, Wang X and Zhang Y (2013) Calibration and fine-tuning of cfd model for predicting the pressure history of detonating explosives. *Journal of Loss Prevention in the Process Industries* 26(1): 181–189.
- Kalaiselvi T, Sriramakrishnan P and Somasundaram K (2017) Survey of using gpu cuda programming model in medical image analysis. *Informatics in Medicine Unlocked* 9: 133–144. DOI: [10.1016/J.IMU.2017.08.001](https://doi.org/10.1016/J.IMU.2017.08.001).
- Kingery CN and Bulmash G (1984) *Airblast Parameters from Tnt Spherical Air Burst and Hemispherical Surface Burst*. Volume 2555 of Technical report ARBRL.
- Kitware I (2024) Paraview. Available at: <https://www.paraview.org>.
- Li Y, Wang X and Zhang Y (2019) Study of detonation wave propagation and pressure distribution in the vicinity of detonation wellhead in a cylindrical confined space using cfd method. *Powder Technology* 343: 9–16.
- Marks NA, Stewart MG, Netherton MD, et al. (2021) Airblast variability and fatality risks from a vbied in a complex urban environment. *Reliability Engineering & System Safety* 209: 107459.
- Mullett PJ, Moragasipitiya HNP and Ward J (2013) Analysis of a protective reinforced concrete slab subject to a vehicle-borne improvised explosive device. *Proceedings of the Institution of Civil Engineers-Engineering and Computational Mechanics* 166(3): 160–175.
- Murong F, Zheng K, Xiong C, et al. (2023) A machine learning-based method for fast prediction of explosion wave overpressure distribution in urban areas. In: 2023 2nd international conference on artificial intelligence and computer information technology (AICIT), Yichang, China, 15–17 September 2023. IEEE, 1–3.
- NASA Glenn Research Center (2021) Examining spatial (grid) convergence. <https://www.grc.nasa.gov/www/wind/valid/tutorial/spatconv.html> (Accessed 11 02 2024).
- National-Instruments (2024) Acquiring an analog signal: bandwidth, nyquist sampling theorem, and aliasing. Available at: <https://www.ni.com/instrument-fundamentals>.

- OpenFOAM Foundation (2024) Openfoam. Available at: <https://www.openfoam.com>.
- Oppenheim AV, Schafer RW and Buck JR (2010) *Discrete-time Signal Processing*. 4th edition. Upper Saddle River, NJ: Prentice Hall.
- Ouyang Z, Carlton A, Guo Q, et al. (2020) Blast vulnerability of drop ceilings in roadway tunnels. *Journal of Performance of Constructed Facilities* 34(6): 04020121.
- Pang S and Shin J (2021) Blast overpressure evaluation for blast valves in protective tunnels with rectangular-shaped tunnel entrances. *Journal of the Korean Society for Geothermal and Hydrothermal Energy* 17(4): 79–90.
- Shin J, Whittaker AS, Aref AJ, et al. (2014) *Air-Blast Effects on Civil Structures*. Technical report. New York: MCEER. Available at: <https://mceer.buffalo.edu>.
- Stirling C (2021) Viper::Blast user manual and examples for Version 1.20.0.
- Stirling C (2022) Viper::Blast. *Viper Applied Science*. Available at: <https://www.viper.as/>.
- Tang L, Rigby S and Tyas A (2018) Validation of air3d for scaled experimental pressure and impulse data. In: 25th international symposium on military aspects of blast and shock (MABS)â€™™, The Hague, Netherlands, pp. 24–27.
- Thornton Thomasetti Defence Ltd (2023a) Capabilities - tvra, security audits & benchmarking. Available at: <https://www.thorntontomasetti.com/capability/threat-vulnerability-risk-assessments-tvra-security-audits-benchmarking>.
- Thornton Thomasetti Defence Ltd (2023b) Walair++ user manual.
- Tyagi K, Rane C, Harshvardhan, et al. (2022) Regression analysis. *Artificial Intelligence and Machine Learning for EDGE Computing*: 53–63. DOI: [10.1016/B978-0-12-824054-0.00007-1](https://doi.org/10.1016/B978-0-12-824054-0.00007-1).
- Tyas A, Reay JJ, Fay SD, et al. (2016) Experimental studies of the effect of rapid afterburn on shock development of near-field explosions. *International Journal of Protective Structures* 7(3): 452–465. DOI: [10.1177/2041419616665931](https://doi.org/10.1177/2041419616665931).
- Wada Y and Liou MS (1997) An accurate and robust flux splitting scheme for shock and contact discontinuities. *SIAM Journal on Scientific Computing* 18(3): 633–657.
- Weaver KF, Morales V, Dunn SL, et al. (2017) *An Introduction to Statistical Analysis in Research*. Wiley.
- Witty L (2023) *The Sensitivities of Blast Parameters in Predicting TNT Equivalence from Experimental Data*. England: University of Sheffield.
- Zaghloul A, Remennikov A and Uy B (2021) Enhancement of blast wave parameters due to shock focusing from multiple simultaneously detonated charges. *International Journal of Protective Structures* 12(4): 541–576.



DSPACE

<https://dspace.org/>

Comparative analysis of blast prediction software for far-field shock wave effects behind a blast wall

Chester, Alastair; Hazael, Rachael; Critchley, Richard

2025-12

Sage

<https://doi.org/10.1177/20414196241284299>

Chester A, Hazael R, Critchley R. (2025) Comparative analysis of blast prediction software for far-field shock wave effects behind a blast wall. *International Journal of Protective Structures*, Volume 16, Issue 4, December 2025, pp. 825-852