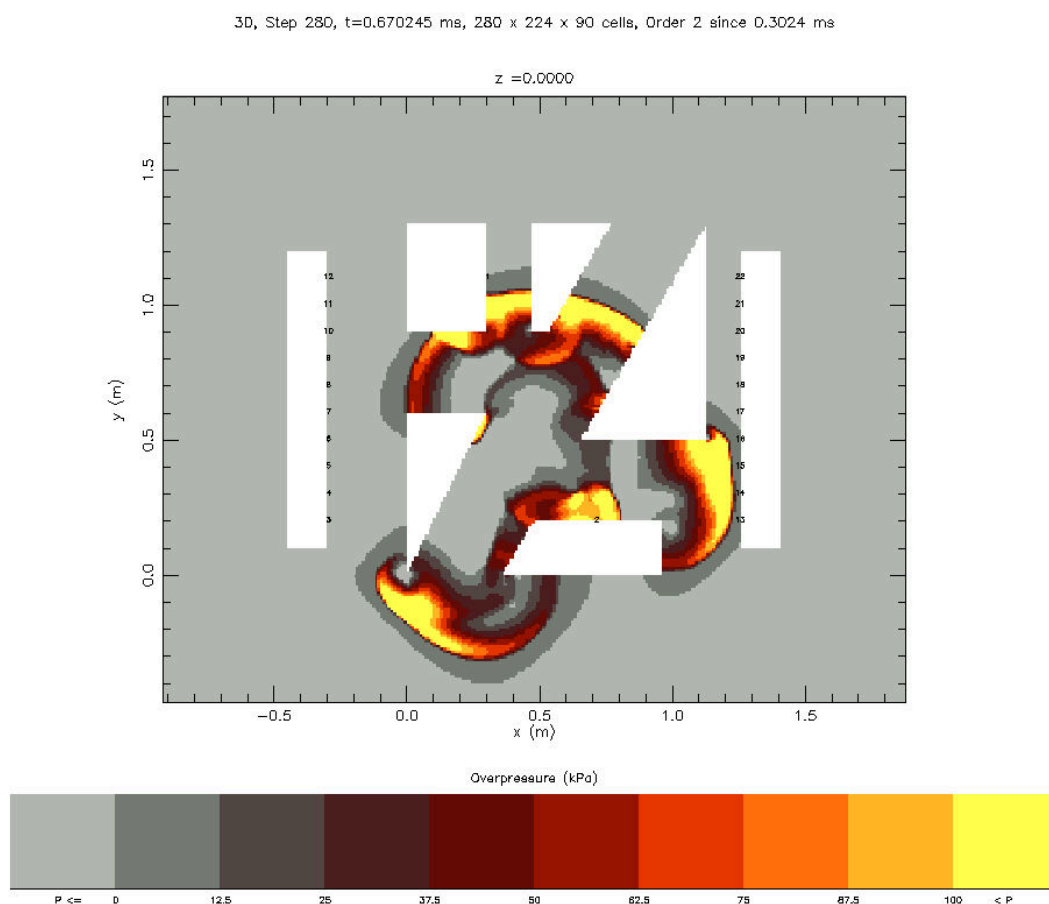


# ProSAir 2022.11 User Guide

Dr Shaun Forth  
Applied Mathematics and Scientific Computing  
CDS Shrivenham  
Swindon SN6 8LA  
UK  
[S.A.Forth@cranfield.ac.uk](mailto:S.A.Forth@cranfield.ac.uk)

November 2022



# Abstract

ProSAir (PROpagation Of Shocks in AIR) is a finite volume, compressible fluid dynamics solver based on second order MUSCL-Hancock time integration and the AUSMDV hybrid flux vector/difference splitting developed by Wada and Liou. ProSAir's primary use is for modelling air-blasts in and around structures and estimating the resultant structural loading. ProSAir is a successor to Air3d version 9 and version 2022.11 still uses the same numerical algorithm. ProSAir improves on Air3d by an improved user interface.

Key features are:

- Easy to use CFD modelling of blast waves caused by detonation of a single high explosive device.
- Fast and robust due to use of the well established AUSMDV finite volume scheme for accurately capturing shocks.
- Second order accuracy achieved by its MUSCL-Hancock integration scheme.
- Use of spherical and axi-symmetric calculation phases to reduce run times.
- Support for running on multi-core/processor computers.
- Windows XP/Vista/7/8/10 64-bit version supplied as standard; 32-bit version for use on legacy systems is still available.
- Graphical user interface allowing rapid building of geometric models made up of simple geometrical components.
- Graphical display of solution during execution.
- Batch jobs via command line invocation.
- Window failure model based on the SAFE/SSG Glazing hazard guide
- Can provide history of pressure, velocity, density and temperature at tens of thousands of target points around the domain.
- Graphical display may be captured to create video presentations.
- User guide with many examples including sample input and output files.

Restrictions:

- Explosive charges are assumed ideal, spherical and with no afterburning.
- No method to input geometry from CAD packages.
- Flowfield itself cannot be post-processed.

# Contents

|          |                                                                                    |           |
|----------|------------------------------------------------------------------------------------|-----------|
| <b>1</b> | <b>Introduction</b>                                                                | <b>7</b>  |
| 1.1      | Conventions in this User Guide . . . . .                                           | 7         |
| 1.2      | The Different Variants of ProSAir . . . . .                                        | 7         |
| 1.2.1    | ProSAir 64 . . . . .                                                               | 8         |
| 1.2.2    | ProSAir Double Precision . . . . .                                                 | 8         |
| 1.2.3    | ProSAir 32 . . . . .                                                               | 8         |
| 1.3      | Structure of this User Guide . . . . .                                             | 9         |
| <b>2</b> | <b>Using ProSAir</b>                                                               | <b>10</b> |
| 2.1      | Starting ProSAir . . . . .                                                         | 10        |
| 2.2      | The ProSAir main window . . . . .                                                  | 11        |
| 2.3      | <b>File</b> drop-down menu . . . . .                                               | 12        |
| 2.3.1    | <b>New simulation</b> . . . . .                                                    | 13        |
| 2.3.2    | <b>Load from file(s)</b> - Shortcut <b>Cntrl-L</b> . . . . .                       | 13        |
| 2.3.3    | <b>Save to current file</b> - Shortcut <b>Cntrl-S</b> . . . . .                    | 13        |
| 2.3.4    | <b>Save to new file</b> . . . . .                                                  | 13        |
| 2.3.5    | <b>Export plot</b> - Shortcut <b>Cntrl-E</b> . . . . .                             | 14        |
| 2.3.6    | <b>Exit</b> . . . . .                                                              | 14        |
| 2.4      | <b>Simulation</b> drop-down menu . . . . .                                         | 14        |
| 2.4.1    | <b>Start</b> - shortcut <b>Cntrl-G</b> . . . . .                                   | 15        |
| 2.4.2    | <b>Restart</b> . . . . .                                                           | 15        |
| 2.4.3    | <b>Pause</b> - shortcut <b>Cntrl-P</b> . . . . .                                   | 16        |
| 2.4.4    | <b>Split Targets</b> . . . . .                                                     | 16        |
| 2.4.5    | <b>Settings</b> . . . . .                                                          | 16        |
| 2.5      | <b>Help</b> drop-down menu . . . . .                                               | 16        |
| 2.5.1    | <b>Help</b> - Shortcut function key <b>F1</b> . . . . .                            | 17        |
| 2.5.2    | <b>Release Notes</b> . . . . .                                                     | 17        |
| 2.5.3    | <b>Change Log</b> . . . . .                                                        | 17        |
| 2.5.4    | <b>About</b> . . . . .                                                             | 17        |
| <b>3</b> | <b>Examples</b>                                                                    | <b>18</b> |
| 3.1      | Kingery and Bulmash - Spherical Airburst . . . . .                                 | 19        |
| 3.1.1    | Problem Definition . . . . .                                                       | 19        |
| 3.1.2    | ProSAir Simulation Settings . . . . .                                              | 20        |
| 3.1.3    | General Settings . . . . .                                                         | 21        |
| 3.1.4    | Spherical Geometry Settings . . . . .                                              | 23        |
| 3.1.5    | Running the Initial Simulation . . . . .                                           | 25        |
| 3.1.6    | Post-processing the <b>pt_spherical</b> ProSAir output files . . . . .             | 28        |
| 3.1.7    | Post-processing the <b>target_summary_spherical</b> ProSAir output files . . . . . | 31        |
| 3.1.8    | Reducing the Cell Size . . . . .                                                   | 34        |
| 3.2      | Height of Burst . . . . .                                                          | 36        |

|        |                                                    |            |
|--------|----------------------------------------------------|------------|
| 3.2.1  | Problem Definition . . . . .                       | 36         |
| 3.2.2  | ProSAir General Settings . . . . .                 | 37         |
| 3.2.3  | ProSAir Spherical Geometry . . . . .               | 38         |
| 3.2.4  | ProSAir Cylindrical Geometry . . . . .             | 39         |
| 3.2.5  | Running the Simulation . . . . .                   | 41         |
| 3.2.6  | Post-processing the ProSAir output files . . . . . | 49         |
| 3.2.7  | Convergence Study . . . . .                        | 51         |
| 3.3    | Cylindrical Containment . . . . .                  | 59         |
| 3.3.1  | Problem Definition . . . . .                       | 59         |
| 3.3.2  | ProSAir General Settings . . . . .                 | 60         |
| 3.3.3  | ProSAir Spherical Geometry . . . . .               | 61         |
| 3.3.4  | ProSAir Cylindrical Geometry . . . . .             | 62         |
| 3.3.5  | Running the Simulation . . . . .                   | 64         |
| 3.3.6  | Post-processing the ProSAir output files . . . . . | 67         |
| 3.4    | Blast in a Generic Cityscape . . . . .             | 68         |
| 3.4.1  | Problem Definition . . . . .                       | 68         |
| 3.4.2  | Using Charge Location <i>C1</i> . . . . .          | 71         |
| 3.4.3  | ProSAir General Settings . . . . .                 | 72         |
| 3.4.4  | Spherical Geometry Settings . . . . .              | 73         |
| 3.4.5  | Cylindrical Geometry Settings . . . . .            | 75         |
| 3.4.6  | 3d Geometry Settings . . . . .                     | 77         |
| 3.4.7  | Adding Obstacles . . . . .                         | 79         |
| 3.4.8  | Adding Target Points . . . . .                     | 82         |
| 3.4.9  | Running the Simulation . . . . .                   | 84         |
| 3.4.10 | Post-processing the ProSAir output files . . . . . | 92         |
| 3.4.11 | Choosing the domain size . . . . .                 | 96         |
| 3.4.12 | Effect of using double precision . . . . .         | 100        |
| 3.4.13 | Effect of mesh refinement . . . . .                | 102        |
| 4      | <b>Simulation Settings</b> . . . . .               | <b>108</b> |
| 4.1    | <b>General Settings Tab</b> . . . . .              | <b>109</b> |
| 4.1.1  | <b>Run Geometries*</b> . . . . .                   | <b>109</b> |
| 4.1.2  | <b>Solver Settings</b> . . . . .                   | <b>109</b> |
| 4.1.3  | <b>Atmospheric Conditions*</b> . . . . .           | <b>110</b> |
| 4.1.4  | <b>Display</b> . . . . .                           | <b>110</b> |
| 4.1.5  | <b>Checkpoints</b> . . . . .                       | <b>110</b> |
| 4.1.6  | <b>Target Point Settings*</b> . . . . .            | <b>111</b> |
| 4.1.7  | <b>Materials*</b> . . . . .                        | <b>111</b> |
| 4.2    | <b>Spherical Geometry Tab</b> . . . . .            | <b>112</b> |
| 4.2.1  | <b>Plot</b> . . . . .                              | <b>112</b> |
| 4.2.2  | <b>Detonation Parameters*</b> . . . . .            | <b>113</b> |
| 4.2.3  | <b>Parameters</b> . . . . .                        | <b>114</b> |
| 4.2.4  | <b>Target Point Grids</b> . . . . .                | <b>115</b> |

|          |                                                                   |            |
|----------|-------------------------------------------------------------------|------------|
| 4.3      | Cylindrical Geometry Tab . . . . .                                | 117        |
| 4.3.1    | Plot . . . . .                                                    | 117        |
| 4.3.2    | Parameters . . . . .                                              | 119        |
| 4.3.3    | Obstacles* . . . . .                                              | 120        |
| 4.3.4    | Target Point Grids* . . . . .                                     | 121        |
| 4.4      | 3d Geometry Tab . . . . .                                         | 123        |
| 4.4.1    | Plot . . . . .                                                    | 123        |
| 4.4.2    | Parameters . . . . .                                              | 125        |
| 4.4.3    | Obstacles* . . . . .                                              | 126        |
| 4.4.4    | Target Point Grids* . . . . .                                     | 127        |
| <b>5</b> | <b>User-Defined Materials Library</b>                             | <b>129</b> |
| 5.1      | Additional Materials Directory . . . . .                          | 129        |
| 5.2      | Gas Materials . . . . .                                           | 129        |
| 5.3      | Explosive Materials . . . . .                                     | 129        |
| <b>6</b> | <b>Glazing</b>                                                    | <b>132</b> |
| 6.1      | Window Obstacle Type . . . . .                                    | 132        |
| 6.2      | Glazing Materials . . . . .                                       | 134        |
| 6.2.1    | PI-Curve - Glazing defined by Multiple Points on PI Curve . . . . | 134        |
| 6.2.2    | PI-Asymptotes - Glazing defined by asymptotes to PI Curve . . . . | 137        |
| 6.2.3    | PI-Hyperbola - Glazing PI Curve defined by a hyperbola . . . . .  | 139        |
| 6.2.4    | NoFailure - Glazing with no failure . . . . .                     | 140        |
| 6.3      | Glazing Failure Model . . . . .                                   | 142        |
| 6.3.1    | Differences compared to Air3d . . . . .                           | 143        |
| 6.3.2    | Criticisms of the Glazing Model . . . . .                         | 143        |
| 6.4      | Example - Glazed Cubicle . . . . .                                | 144        |
| 6.4.1    | 3kg at 8m Stand-off . . . . .                                     | 145        |
| 6.4.2    | 4.536kg at 8m Stand-off . . . . .                                 | 152        |
| <b>7</b> | <b>Practical Issues</b>                                           | <b>154</b> |
| 7.1      | Command Line Arguments . . . . .                                  | 154        |
| 7.2      | Batch Processing . . . . .                                        | 155        |
| 7.2.1    | Simple Batch Example . . . . .                                    | 155        |
| <b>8</b> | <b>Algorithms</b>                                                 | <b>159</b> |
| 8.1      | Automatic adjustment of cell size . . . . .                       | 159        |
| 8.2      | Detection of shock-front domain . . . . .                         | 159        |
| <b>9</b> | <b>Air3d Compatibility</b>                                        | <b>161</b> |
| 9.1      | Example Two: Height of Burst . . . . .                            | 161        |
| 9.2      | Example Three: Blast Wave Clearing . . . . .                      | 164        |
| 9.3      | Example Four: Testing Glazing in a Cubicle Structure . . . . .    | 166        |

|           |                                                                        |            |
|-----------|------------------------------------------------------------------------|------------|
| 9.4       | Example Five: Blast in a City Street . . . . .                         | 168        |
| <b>10</b> | <b>Trouble Shooting</b>                                                | <b>170</b> |
| 10.1      | FAQs . . . . .                                                         | 170        |
| 10.1.1    | My target point files contain no data . . . . .                        | 170        |
| 10.1.2    | The 64 bit version of ProSAir crashes . . . . .                        | 170        |
| 10.1.3    | My results are slightly different to those in the User Guide . . . . . | 170        |
|           | <b>Appendices</b>                                                      | <b>171</b> |
| <b>A</b>  | <b>Change Log</b>                                                      | <b>172</b> |

# 1 Introduction

ProSAir (Propagation Of Shocks in AIR) is a compressible fluid dynamics solver based on second order MUSCL-Hancock time integration [?] and the AUSMDV hybrid flux vector/difference splitting developed by Wada and Liou [?].

ProSAir is a successor to Air3d version 9 [?] and version 2022.11 still uses the same numerical algorithm. However, ProSAir improves on Air3d by an improved graphical user interface and in due course will feature improved algorithms and the ability to model a wider range of blast-related phenomena. For now, we refer to the Air3d version 9 user guide [?] for technical details of the algorithm. ProSAir's primary use remains the same as for Air3d – modelling air-blasts in and around structures and estimating the resultant structural loading.

## 1.1 Conventions in this User Guide

To assist the reader we have adopted the following conventions.

- The names of all component parts of the ProSAir graphical user interface, options, and menu choices are given this **colour**.
- All hyperlinked file names are displayed as, for example, `KingeryBulmashInitial.pr1`. **Note** - the hyperlinking allows the user to download the file to a folder of their choice by using the right mouse button to *click on* the hyperlink and, from the resulting browser-dependent menu, choosing the appropriate option (e.g., Gogle Chrome and Firefox use **Save link as**, Internet Explorer use **Save target as**).
- All non-hyperlinked file names are displayed as **filename**.
- Important points may be noted **thus**.
- Commands accessed by combination of the Control and another key are denoted, for example, **Cntrl-F1**.

## 1.2 The Different Variants of ProSAir

The standard distribution of ProSAir is for 64 bit windows operating systems and uses either single precisions (see Section 1.2.1) or double precision (Section 1.2.2) arithmetic. For legacy systems, 32 bit systems are available on request (see Section 1.2.3).

Note that command line variants supported in previous versions of ProSAir are no longer supported as their functionality regarding batch processing is now replicated in the regular versions (see Section 7.2).

### 1.2.1 ProSAir 64

This is now the standard version of ProSAir and will run on any 64-bit Intel or AMD based computer running a 64-bit version of Microsoft Windows. The memory limitations present in the 32-bit version (see Section 1.2.3) are not applicable and ProSAir 64 has been tested with 64GB of memory and over a billion computational cells.

Default installation of ProSAir stores the associated 64 bit executable at

```
"C:\Program Files (x86)\ProSAir64\ProSAir.exe"
```

### 1.2.2 ProSAir Double Precision

ProSAir 64 is supplied with a variant compiled to use double precision arithmetic. This will use approximately twice as much memory as the regular versions of ProSAir but this will not result in a 50% loss of performance. If you suspect rounding errors may be causing unacceptable loss of accuracy for your simulation, you may wish to test the same problem with this double precision variant.

Default installation of ProSAir places the executable at

```
"C:\Program Files (x86)\ProSAir64\ProSAir_DP.exe"
```

### 1.2.3 ProSAir 32

This was the default version of ProSAir Windows computers which will run on 32 bit Windows desktop PCs. However, due to an inherent limitation in Windows, ProSAir 32 can only allocate approximately 1.3GB of memory for each simulation (allowing around 29 million computational cells, or slightly over 300 cells in each dimension for 3d calculations). While 32-bit Windows can theoretically address up to 3GB of memory per process, only 1.3GB can be contiguously allocated. As most all modern PCs use 64 bit operating systems, ProSAir 32 is only supplied on request.

Installation of ProSAir 32 stores the associated 32 bit executable at

```
"C:\Program Files\ProSAir32\ProSAir.exe"
```

Installation of the 32 bit double precision version under 32 bit windows stores the associated executable at

```
"C:\Program Files\ProSAir32\ProSAir_DP.exe"
```

Due to the increased memory requirements you may find using ProSAir 32 Double Precision non-feasible, particularly on 32 bit platforms.

### 1.3 Structure of this User Guide

Section 2 introduces the ProSAir tool and its graphical user interface. Section 3 takes the user, step-by-step, through several examples for which input files are provided. Section 4 describes all the settings and options that may be used when configuring a run of the ProSAir tool. Section 5 describes how users may store their own libraries of material properties for use in ProSAir. Section 6 details ProSAir's crude glazing failure model. Section 7 covers practical issues such as running from the command line and batch processing.

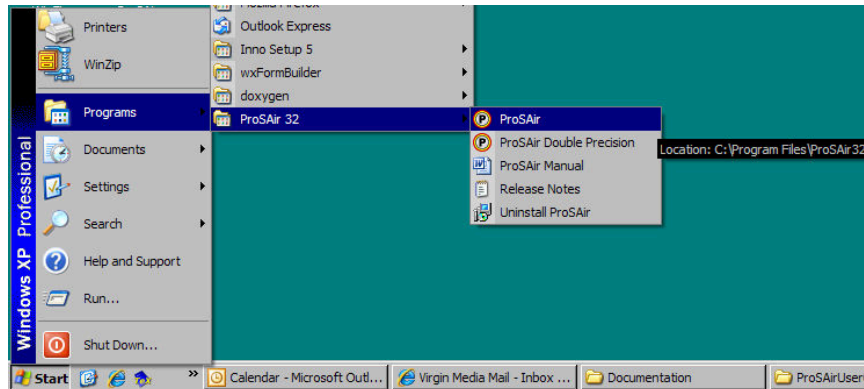
A full change log may be found in Appendix A.

## 2 Using ProSAir

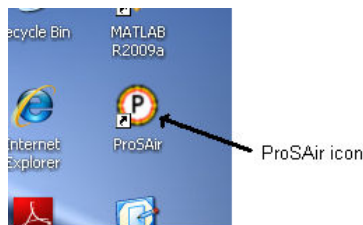
### 2.1 Starting ProSAir

ProSAir can be started in the following ways:

1. From your Windows **Start**-menu.



2. By double clicking the ProSAir desktop icon.



3. By double clicking on any ProSAir input file (`.pr1`) or ProSAir checkpoint file(`.prc`). The latter approach will load the chosen simulation at start-up.
4. By invoking ProSAir or its command line variants from a DOS prompt or within a windows batch file.

## 2.2 The ProSAir main window

Once ProSAir has started the main window of Fig. 1 appears.

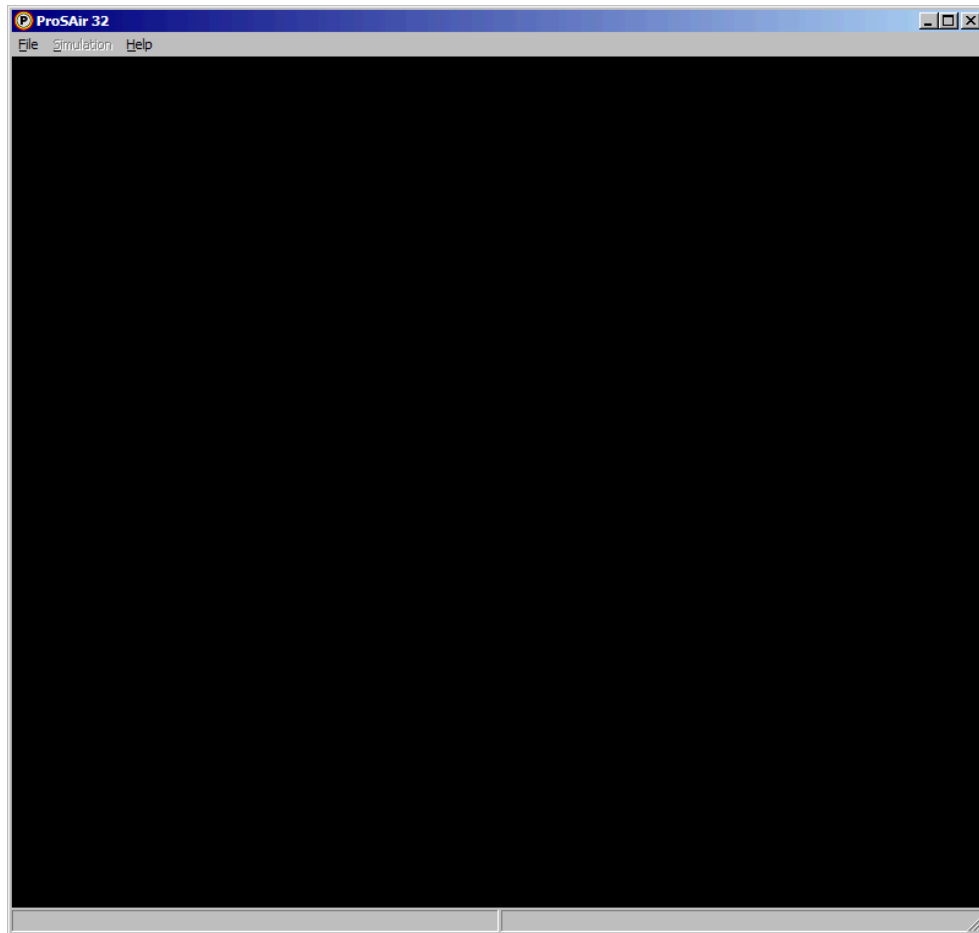


Figure 1: The ProSAir main window

The window has three drop-down menus:

1. **File** drop-down menu (see §2.3) allows you to either create a new simulation with default parameters (which you may later change) or load an existing simulation from file. Also, whenever you have loaded a simulation, you may save the settings to the existing or a new file. Use this to save the legacy simulations as new style ProSAir simulation. If you have opened ProSAir from the Windows **Start** menu or by clicking on the ProSAir desktop icon then it is essential you save a simulation before you run it as described in §2.3.4.
2. **Simulation** drop-down menu (see §2.4) allows you to start, pause or restart as well as change settings and parameters for the loaded simulation.

3. **Help** menu (see §2.5) provides access to this document as well as some release information.

## 2.3 File drop-down menu

If a simulation has not been loaded then the **File** drop-down menu appears as in Fig. 2.

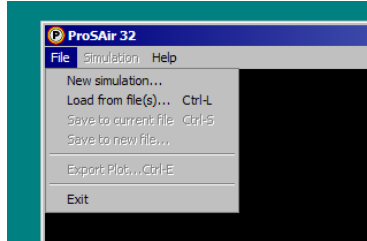


Figure 2: The initial **File** drop-down menu

At this stage it is possible to

1. Start a **New simulation** - see §2.3.1. In this case it is essential you save your simulation to a new file before you run it as described in §2.3.4.
2. **Load from file(s)** an existing ProSAir input (**.pr1**) or checkpoint file (**.prc**) - see §2.3.2.
3. **Exit** the ProSAir program - see §2.3.6.

If a simulation is already loaded then the File drop-down menu appears as in Fig. 3.

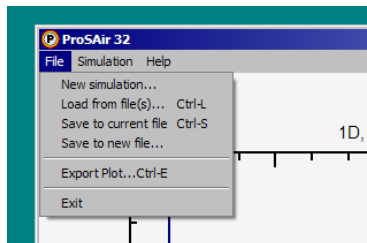


Figure 3: The full File drop-down menu

At this it is now possible to

1. Start a **New simulation** - see §2.3.1.
2. **Load from file(s)** an existing ProSAir input file (**.pr1**), or checkpoint file (**.prc**), or legacy Air3d input - see §2.3.2.

3. **Save to current file** the present settings for ProSAir - see §2.3.3.
4. **Save to new file** the present settings for ProSAir - see §2.3.4.
5. **Export plot** the current plot displayed by ProSAir to a graphics file for later use - see §2.3.5.
6. **Exit** the ProSAir program - see §2.3.6.

### 2.3.1 New simulation

In this case ProSAir's settings are initialised with the computation corresponding to 1kg of an explosive with the mass density, energy density and detonation velocity of TNT detonating 1m above the ground. This is modelled using spherical symmetry until the shock reaches 1m (i.e. the ground), then cylindrical symmetry is used until the shock reaches  $r = 5\text{m}$ , at which point a 3-D calculation is performed over the quarter domain  $0 \leq x, y \leq 10$ .

### 2.3.2 Load from file(s) - Shortcut Cntrl-L

A file open dialogue box is opened using which the user can navigate to an arbitrary folder and, by changing the **Files of type:** drop-down option may input one of:

1. A ProSAir input file with file extension `.pr1`.
2. A ProSAir checkpoint file from a previous execution of ProSAir and that has extension `.prc`.
3. An Air3d input file `[?]` which may have arbitrary file extension.

### 2.3.3 Save to current file - Shortcut Cntrl-S

The current ProSAir settings are saved to the current ProSAir input file. If no file has been defined then a dialogue box to allow the user to input the folder and name of a new ProSAir input file name is opened. On saving the file ProSAir's working folder is switched to that specified and all file output will be to files in that folder.

### 2.3.4 Save to new file

A dialogue box is opened to allow the user to specify a folder and new file for saving the current ProSAir settings. On saving the file ProSAir's working folder is switched to that specified and all file output will be to files in that folder.

### 2.3.5 Export plot - Shortcut Cntrl-E

The current plot displayed by ProSAir is saved to a graphics file for later use in reports etc. A dialogue box is opened to allow the user to save the plot in eps (encapsulated postscript), jpg (Joint Photographic Experts Group), or png (portable network graphics) format.

### 2.3.6 Exit

Closes all files and exits ProSAir.

## 2.4 Simulation drop-down menu

If ProSAir is initiated without loading a file (see §2.1) and a **New Simulation** (see §2.3.1) is selected from the **File** drop-down menu (§2.3) then the Simulation drop-down menu appears as in Fig. 4.

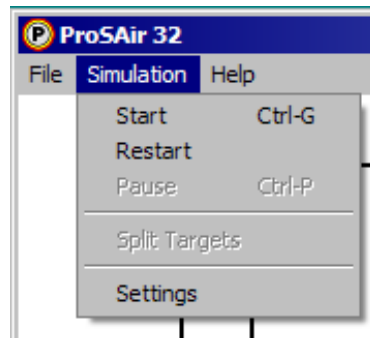


Figure 4: The initial Simulation drop-down menu

At this stage it is possible to

1. **Start** the simulation running - see §2.4.1.
2. **Restart** a paused simulation - see §2.4.2.
3. Change the **Settings** for the geometry, charge, grid, solver, etc - see §2.4.5 or §4.

If a simulation has been loaded from a file (see §2.3) or a running simulation is paused (see §2.4.3) then the Simulation drop-down menu appears as in Fig. 5.

At this stage it is possible to

1. **Start** the simulation running - see §2.4.1.
2. **Restart** a paused simulation - see §2.4.2.
3. **Split Targets** - split the target files containing multiple target point data sets into a file for each target point - see §2.4.4.

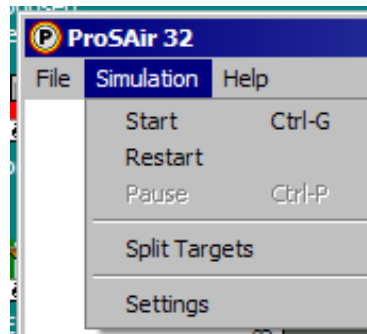


Figure 5: The paused Simulation drop-down menu

4. Change the **Settings** for the geometry, charge, grid, solver, etc - see §2.4.5 or §4.

If a simulation is already executing (i.e., running) then the Simulation drop-down menu appears as in Fig. 6.

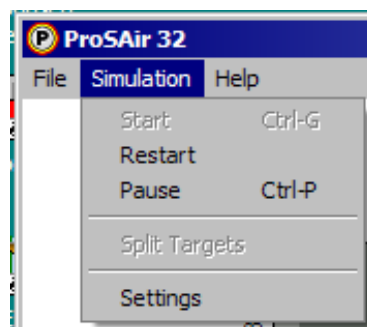


Figure 6: The Simulation drop-down menu during execution

At this it is now possible to

1. **Restart** a the simulation - see §2.4.2.
2. **Pause** an executing simulation - see §2.4.3.
3. Change the **Settings** for the geometry, charge, grid, solver, etc - see §2.4.5 or §4.

#### 2.4.1 **Start** - shortcut **Cntrl-G**

Execution of the simulation starts from its current state.

#### 2.4.2 **Restart**

The simulation is reset to its initial conditions at time  $t = 0s$ . **Start** must be used to commence execution.

### 2.4.3 Pause - shortcut Cntrl-P

Simulation execution is paused at its current time step. This may be used so that **Settings** might be changed.

### 2.4.4 Split Targets

ProSAir allows the user to specify so-called **target points** - locations in the computational domain at which flow variables (pressure, temperature etc.) will be recorded (see §4.1.6, §4.2.4, §4.3.4 and §4.4.4). By default the data collected is stored in files containing data for banks of 20 targets named, for example, `pt_3d_1_to_20.csv`. As described in §4.1.6 the user may specify to **Split target points** in which case at the end of computation the target point files are re-read by ProSAir and an individual target point file is written for each target point. However, if a computation should fail part way through, or if a user has not selected to **Split target points**, or if a user wishes to split targets part way through a computation, then this option may be used to split the targets.

- **Splitting existing target files** - if your computation has already been performed (and perhaps even crashed) but the target files not split then open the corresponding ProSAir input file (see §2.3.2) then select **Simulation - Split Targets** and the target point data will be split.
- **Splitting part way through a computation** - if you wish to view individual target point data part way through a computation then pause the computation (see §2.4.3) then select **Simulation - Split Targets** and the target point data will be split.

### 2.4.5 Settings

This menu item allows the user to set or change all of ProSAir's settings and details are given in §4.

## 2.5 Help drop-down menu

The **Help** drop-down menu appears as in Fig. 7.

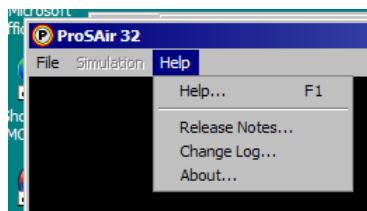


Figure 7: The Help drop-down menu

At this stage it is possible to

1. Get **Help** - by opening the User Guide §2.5.1.
2. Open the **Release Notes** §2.5.2.
3. Open the **Change Log** §2.5.3.
4. Open the **About** the ProSAir program information §2.5.4.

#### **2.5.1 Help - Shortcut function key F1**

A html version of the User Guide (this document) is opened in the user's default web-browser.

#### **2.5.2 Release Notes**

The current release notes are opened in a text viewer.

#### **2.5.3 Change Log**

The **change log**, specifying when changes were made to the ProSAir package, is opened in a text viewer.

#### **2.5.4 About**

A window detailing the ProSAir version number and the variant (see §1.2) of ProSAir is displayed.

### 3 Examples

In this Section we take the user, step-by-step, through several examples of creating, running, post-processing and assessing the results from ProSAir.

**Kingery Bulmash Spherical Airburst** example of Section 3.1 models the detonation of a spherical charge in free air and corresponds to the experimental studies and review of Kingery and Bulmash [?].

**Height of Burst** example of Section 3.2 models the interaction of a spherical blast wave with the ground.

**Cylindrical Containment** example of § 3.3 considers an explosion on the axis of symmetry in the interior of a cylindrical containment vessel with venting. This example is taken from Rose et al [?]

**Cityscape** example of Section 3.4 considers the complex interaction of a spherical blast wave with the streets of a generic cityscape as studied previously by Brittle [?] and Smith et al [?].

### 3.1 Kingery and Bulmash - Spherical Airburst

In this example we show how to apply ProSAir to obtain spherical airburst parameters and compare with the curve fits derived from experimental data of Kingery and Bulmash [?].

#### 3.1.1 Problem Definition

This problem concerns the estimation of side-on blast parameters for a 1kg spherical charge of TNT centrally detonated in free air and measured at subset of the target locations defined in [?, Table 1]. The target points chosen are as in Table 1.

| Target No. | Distance (m) | Target No. | Distance (m) |
|------------|--------------|------------|--------------|
| 1          | 0.10         | 17         | 1.00         |
| 2          | 0.12         | 18         | 1.20         |
| 3          | 0.14         | 19         | 1.40         |
| 4          | 0.16         | 20         | 1.60         |
| 5          | 0.18         | 21         | 1.80         |
| 6          | 0.20         | 22         | 2.0          |
| 7          | 0.25         | 23         | 2.5          |
| 8          | 0.30         | 24         | 3.0          |
| 9          | 0.35         | 25         | 3.5          |
| 10         | 0.40         | 26         | 4.0          |
| 11         | 0.45         | 27         | 4.5          |
| 12         | 0.50         | 28         | 5.0          |
| 13         | 0.60         |            |              |
| 14         | 0.70         |            |              |
| 15         | 0.80         |            |              |
| 16         | 0.90         |            |              |

Table 1: Target point locations for the Kingery and Bulmash problem. The horizontal lines in the table and the table bottom split the points into groups with constant spacing which will be utilised in **Target Point Grids** of §3.1.4

Since the density of TNT may be taken as 1600kg/m<sup>3</sup> this gives a volume  $V_{TNT}$  of explosive,

$$V_{TNT} = \frac{1\text{kg}}{1600\text{kg/m}^3} = 6.25 \times 10^{-4}\text{m}^3,$$

and thus the radius of the spherical charge is

$$r_{TNT} = \sqrt[3]{\frac{3}{4\pi}V_{TNT}} \approx 0.0530\text{m}.$$

We see that the last target point is some 750 charge radii away from the detonation point. For accurate simulation of the charge we recommend using 50 cells across the radius of

a spherical charge leading, in this case, to a cell size of about 1mm. However, a 1mm cell size requires some 5,000 cells to extend to the last target point at 40m. In order to quickly ascertain other settings such as the problem time and outer boundary location we will initially choose use a cell size of 10mm as then the simulation should run some 100 times faster (10 times fewer cells and the time-step can be 10 times bigger).

### 3.1.2 ProSAir Simulation Settings

An initial ProSAir input file `KingeryBulmashInitial.pr1` may be found in the `Documentation/ProSAirUserGuide/KingeryBulmash/Initial` folder of the ProSAir distribution. When loaded into ProSAir this produces the ProSAir main window of Fig. 8. The main window shows the overpressure associated with the solution state at time  $t =$

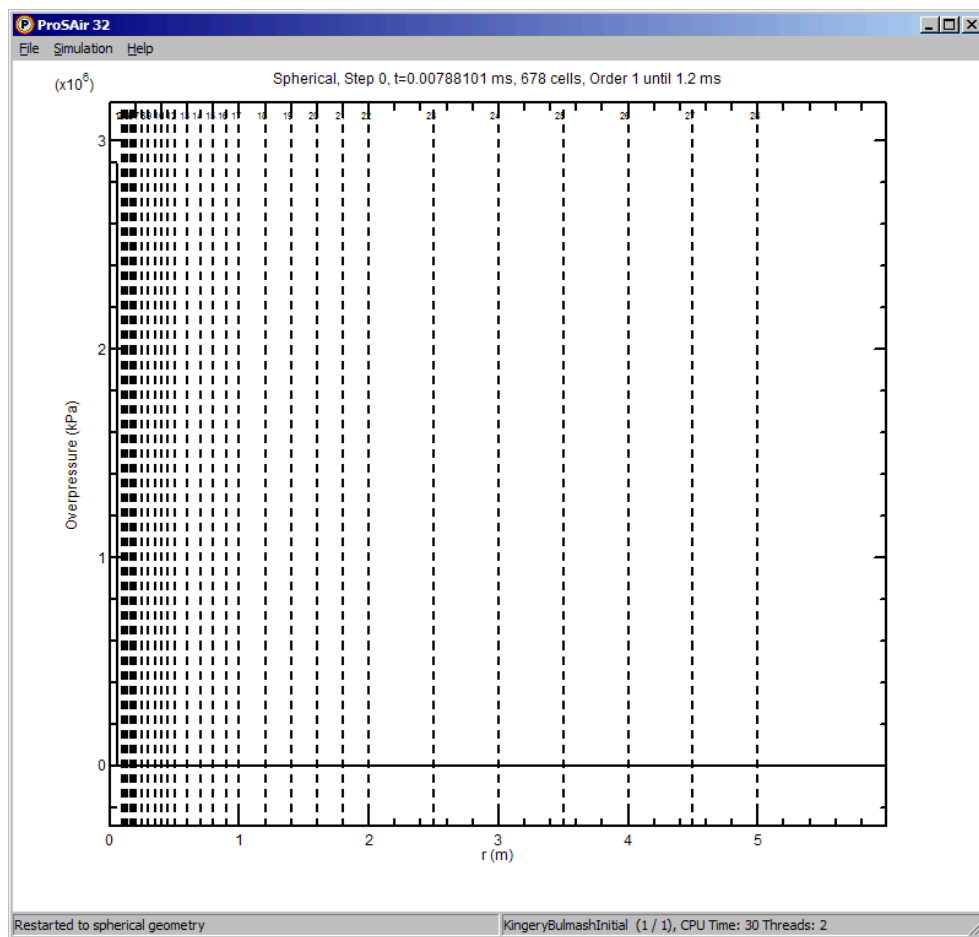


Figure 8: Main window of ProSAir on loading the Kingery Bulmash initial input file `KingeryBulmashInitial.pr1`

0.00788ms which is when ProSAir has determined the charge's detonation wave will have

reached the surface of the charge. The charge radius is about 53mm and we discern the associated peak in overpressure close to the origin. The vertical dashed lines show the locations of the 28 target points and are labelled, towards the top of the plot, with the target point number.

### 3.1.3 General Settings

On choosing to view the Simulation Settings as in 2.2 the **General Settings** tab of Fig. 9 is displayed.

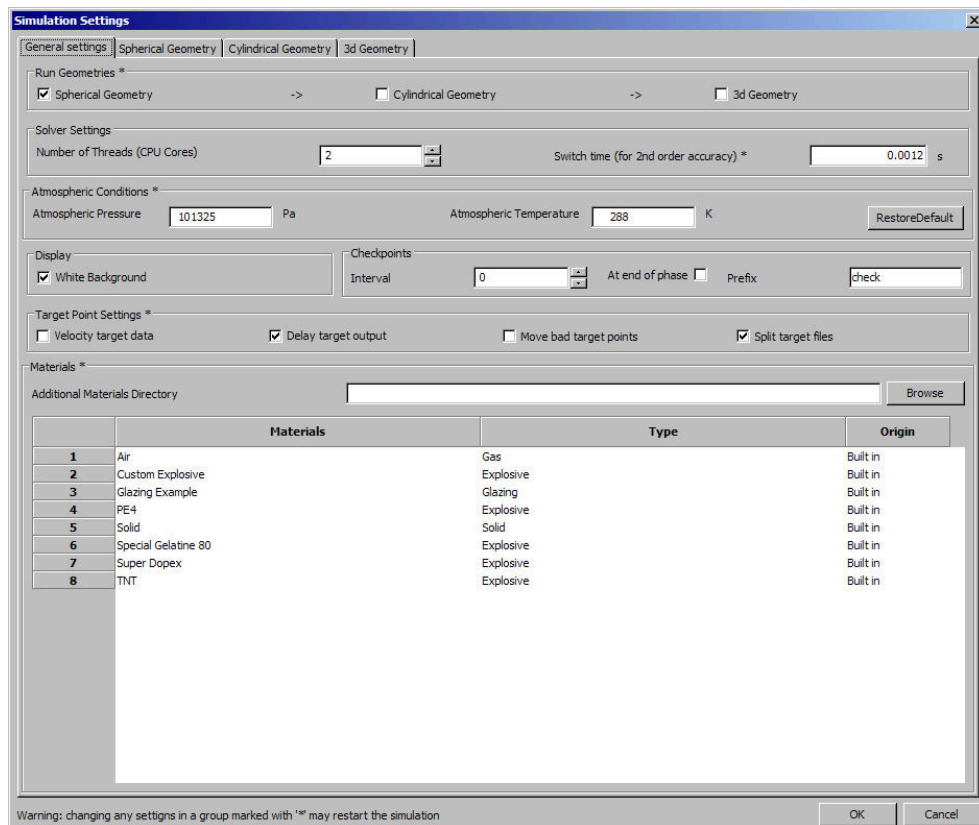


Figure 9: **General settings** tab for the Kingery Bulmash input file KingeryBulmash.pr1

In each panel settings have been made as follows.

**Run Geometries** Note how only the **Spherical Geometry** box is checked because the solution will maintain spherical symmetry as there are no obstacles, no ground and a centrally detonated spherical charge. There is thus no requirement to run using cylindrical or 3d geometries. See § 4.1.1 for more details on **Run Geometries**.

**Solver Settings** Here

- We have indicated to ProSAir that a maximum of 2 CPU cores should be used by ProSAir, in fact currently the spherical phase solver in ProSAir is not parallelised for multiple cores and will use just one.
- The **Switch time (for 2nd order accuracy)** has been set to 0.0012s in line with the recommended value for 1kg TNT given in § 4.1.2.

See § 4.1.2 for more details on **Solver Settings**.

**Atmospheric Conditions** have been left at the default ISO-standard atmosphere as described in § 4.1.3.

**Display** is left at the default white background. See § 4.1.4 for more information on **Display**.

**Checkpoints** we have opted to turn off all check-pointing. See § 4.1.5 for more information on **Checkpoints**.

**Target Point Settings** we have opted to:

- Delay the recording of each target point's data until the primary shock reaches the target point.
- Split the target point files so that an individual target point file is produced for each target point.

See § 4.1.6 for more information on **Target Point Settings**.

**Materials** we will just use the default material properties supplied with ProSAir. See § 4.1.7 for more information on **Materials**.

### 3.1.4 Spherical Geometry Settings

On clicking the **Spherical Geometry** tab in the **Simulation Settings** window of Fig. 9 the **Spherical Geometry** tab of Fig. 10 is displayed.

**Simulation Settings**

General settings | **Spherical Geometry** | Cylindrical Geometry | 3d Geometry

**Plot**  
Variable: **Overpressure** Interval: **100** ☐ Save Plots ☒ Show Targets

**Detonation Parameters \***  
Explosive Type: **TNT**  
Charge mass: **1** kg

**Details \***  
Explosive density: 1600 kg/m<sup>3</sup>  
Heat of explosion: 4.52e+006 J/kg  
Detonation velocity: 6730 m/s  
Cv: 715  
Initial temperature: 6321.68 K  
Ratio of specific heats: 1.4

**Parameters**  
Problem time: **0.025** s  
CFL safety number: **0.5**  
Cell size: ☒ fit to charge radius **0.01** m  
Domain size: **6** m  
Boundary: **Transmissive**

**Target Point Grids \***

|   | Radius From | Radius To | # in Range |
|---|-------------|-----------|------------|
| 1 | 0.1         | 0.18      | 5          |
| 2 | 0.2         | 0.4       | 6          |
| 3 | 0.5         | 0.9       | 5          |
| 4 | 1           | 1.8       | 5          |
| 5 | 2           | 5         | 7          |

Add Target Grid Delete Selected Target Grids

Warning: changing any settings in a group marked with "\*" may restart the simulation

OK Cancel

Figure 10: **Spherical Geometry** tab for the Kingery Bulmash input file KingeryBulmash.pr1

In each panel settings have been made as follows.

**Plot** we have opted to:

- Use **Variable** to plot the **Overpressure**.
- Update the plots at an **Interval** of every **100** time steps.
- Use **Show Targets** to display the target point locations. If this option were not selected then the target point locations would not be displayed as the vertical broken lines in the main ProSAir window as in Fig. 8.

See § 4.2.1 for more information on **Plot** settings.

**Detonation Parameters** we have selected 1kg of TNT as specified in § 3.1.1. Note that the properties of TNT and its resultant post-detonation ideal gas approximation are shown in the sub-panel **Details**. See § 4.2.2 for more information on **Detonation Parameters** settings.

**Parameters** we have opted for:

- A **Problem time** of 0.025s which we trust will be sufficient to gather information on the positive phases of all target points out to 5m. We estimated this value by taking the 6m domain size, dividing by the atmospheric sound speed  $330\text{ms}^{-1}$  to give 0.018s and increasing by a little for good measure.
- The default **CFL safety number** of 0.5.
- A nominal **Cell size** of 0.01m is used initially as explained in § 3.1.1. We have chosen to **fit to charge radius** (see § 4.2.3 and §8.1) so this **Cell size** will be internally adjusted by ProSAir to ensure the charge surface exactly aligns with a cell boundary.
- A **Domain size** of 6m is chosen to lie comfortably beyond the most distant target point.
- The **Boundary** at  $r = 6\text{m}$  is set to be **Transmissive** to allow the shock to pass out of the domain without reflecting off the outer boundary.

See § 4.2.3 for more information on spherical **Parameters**.

**Target Point Grids** we have defined 8 target point grids for which the associated target points will correspond to those of Table 1. For example, the first grid of 5 points starts at 0.1m and ends at 0.18m and so will have a spacing of  $0.02\text{m} = (0.18\text{m} - 0.1\text{m}) / (5\text{points} - 1)$  so matching points 1 to 5 of Table 1. See § 4.2.4 for more information on spherical **Target Point Grids**.

We now hit the **OK** button at the bottom of the **Simulation Settings** window of Fig. 10 to accept the settings.

### 3.1.5 Running the Initial Simulation

We now choose the menu option **Simulation -> Start** or press Control-G as described in §2.4 to start the simulation. Fig. 11 shows the display of the ProSAir main window after 1.7ms simulated time. The blast wave and associated region of underpressure are clearly

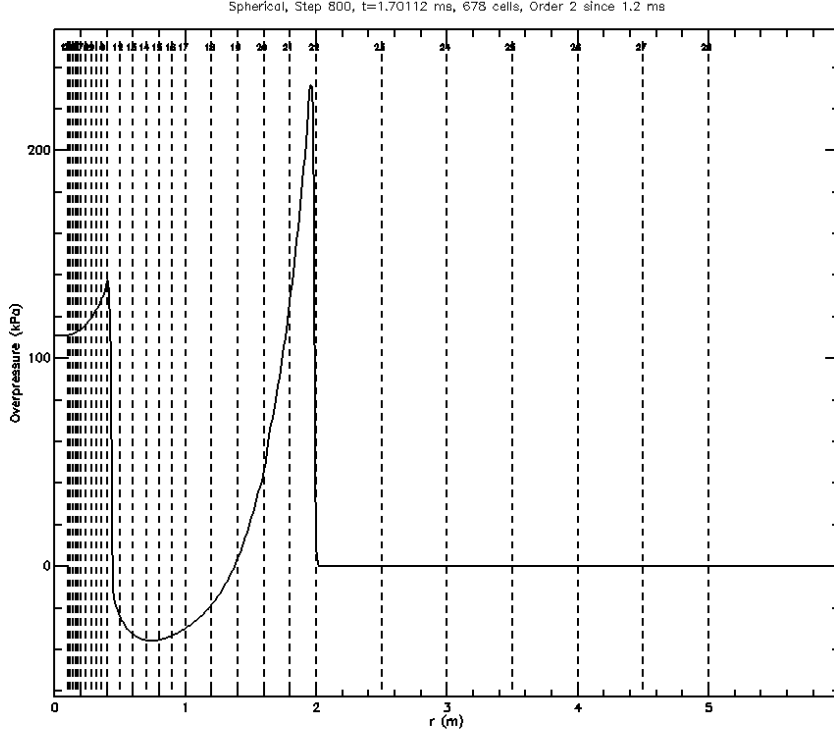


Figure 11: Overpressure versus radius after about 1.7ms simulated time when running the Kingery Bulmash input file `KingeryBulmashInitial.pr1`

visible.

After about 10ms simulated time the shock approaches the outer boundary at  $r = 6\text{m}$  as seen in Fig. 12.

Thereafter the shock exits the domain and, while the flow remains supersonic close to the boundary, the solution remains valid. However, from about 20ms flow features start to enter the domain from the outflow boundary at the right hand side of the domain as seen in Fig. 13. The solution now is no longer valid as atmospheric conditions are being *sucked* back into the domain at  $r = 6\text{m}$ . Physically there should be post-shock conditions here as the blast wave would continue to propagate outwards.

For the rather coarse cell size we have chosen this initial simulation takes about 2mins. The alert user may note that the wall clock time required per time step seems to increase with time until it approximately plateaus. This is because ProSAir detects the approximate

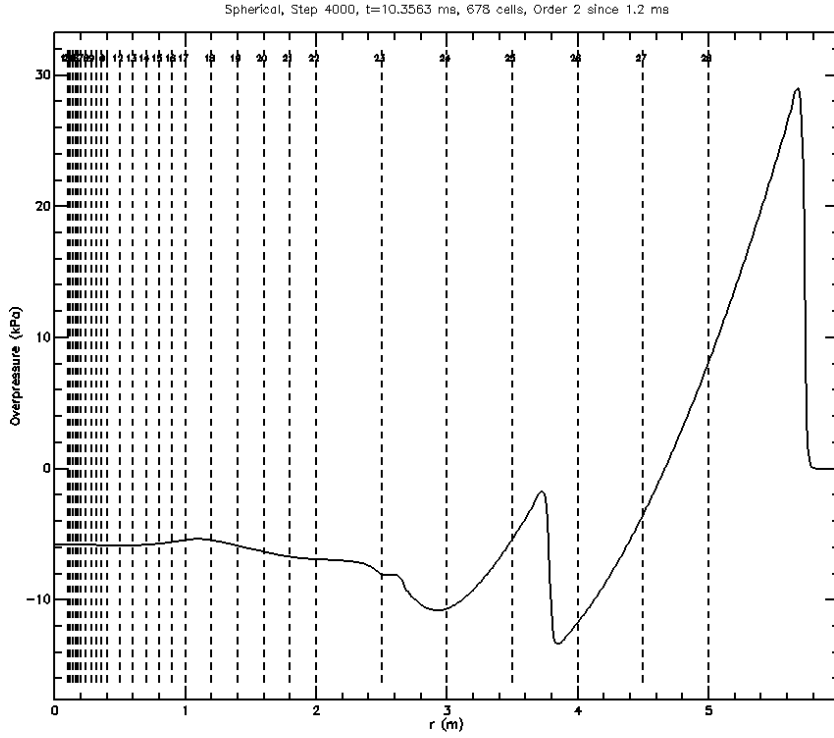


Figure 12: Overpressure versus radius after about 10ms simulated time when running the Kingery Bulmash input file `KingeryBulmashInitial.pr1`

location of the blast wave and only computes the solution on the cells within that region - see §8.2.

If we now examine the folder where the simulation has run we will discover numerous ProSAir output files.

- `pt_spherical_1.csv` – `pt_spherical_28.csv`: comma separated value (.csv) files containing the solution history at the target points.
- `pt_spherical_1_20.csv` and `pt_spherical_21_40.csv` - amalgamated data from up to 20 target points. i.e., `pt_spherical_1_20.csv` contains data for points 1 to 20, `pt_spherical_21_40.csv` for points 21 to 28.
- `target_summary_spherical.csv` - integrated blast parameters at the target points.

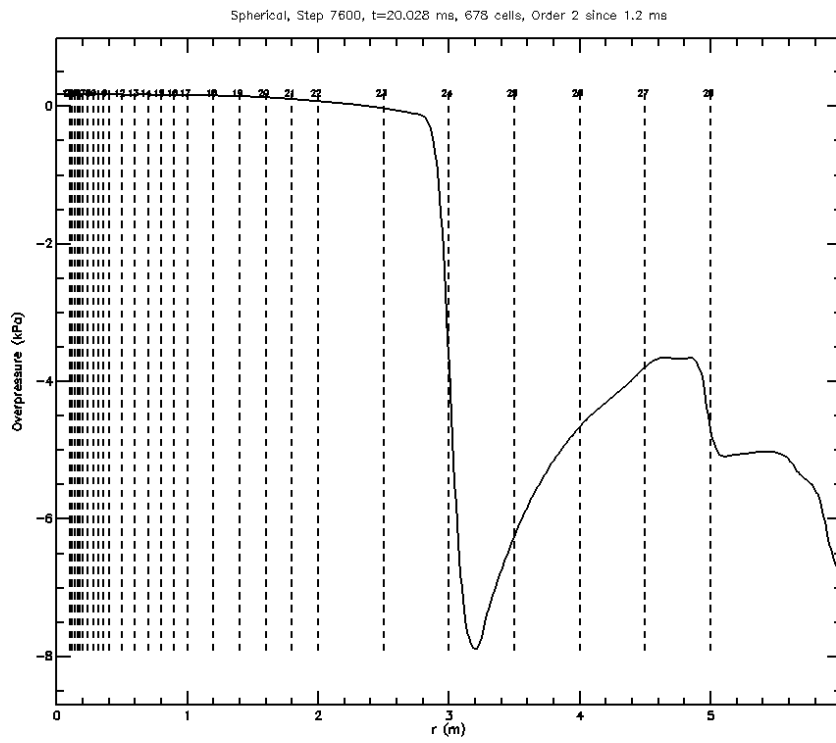


Figure 13: Overpressure versus radius after about 20ms simulated time when running the Kingery Bulmash input file `KingeryBulmashInitial.pr1`

### 3.1.6 Post-processing the pt\_spherical ProSAir output files

Firstly consider the file `pt_spherical_1.csv`. This contains the target point data for target number 1. The first few lines of the file are shown in Fig. 14.

```
Target 1
Position 0.101659 0 0
Cell 11 0 0
Distance 0.101659
Status OK
Time (ms) Density (kg/m^3) Energy (kJ/kg) Overpressure (kPa)
Temperature (C)
1.07E-02
1.22E-02
1.32E-02
1.41E-02 1.23E+00 2.06E+05 -7.81E-06 1.49E+01
1.50E-02 1.23E+00 2.06E+05 -7.81E-06 1.49E+01
1.59E-02 2.66E+00 4.15E+06 4.31E+03 5.53E+03
1.67E-02 5.55E+00 4.75E+06 1.05E+04 6.37E+03
1.76E-02 8.89E+00 4.19E+06 1.48E+04 5.59E+03
1.85E-02 1.22E+01 3.67E+06 1.77E+04 4.85E+03
1.94E-02 1.54E+01 3.29E+06 2.01E+04 4.32E+03
2.02E-02 1.85E+01 3.02E+06 2.22E+04 3.94E+03
```

Figure 14: First few lines of `pt_spherical_1.csv` produced by ProSAir from input file `KingeryBulmashInitial.pr1` (commas omitted)

Working from line 1 we see the information:

**line 1** `Target 1` indicating that the data is from target point 1.

**line 2** `Position 0.101659 0 0` indicating that the point is at 0.101659m in the first ( $r$ ) coordinate direction; the other two coordinates are not used in this one dimensional simulation. Of course in the input file we specified that the first target should be at 0.1m but ProSAir has *adjusted* it to be the location of the nearest cell centre.

**line 3** `Cell 11 0 0` indicating that the data is from the cell with index 11 in the first ( $r$ ) grid direction.

**line 4** `Status OK` indicates that the target point was within the flow field and hence data will be corrected.

**line 5** gives the headings of the data collected and their units.

**lines 6-8** give just the time data since at this time the shock has not impacted on the cell and no data is collected (see § 8.2).

lines 9 onwards the full data is written.

The overpressure data from this file may easily be plot, using for example Microsoft Excel or Matlab, to give the blast parameter history plot of Fig. 15.

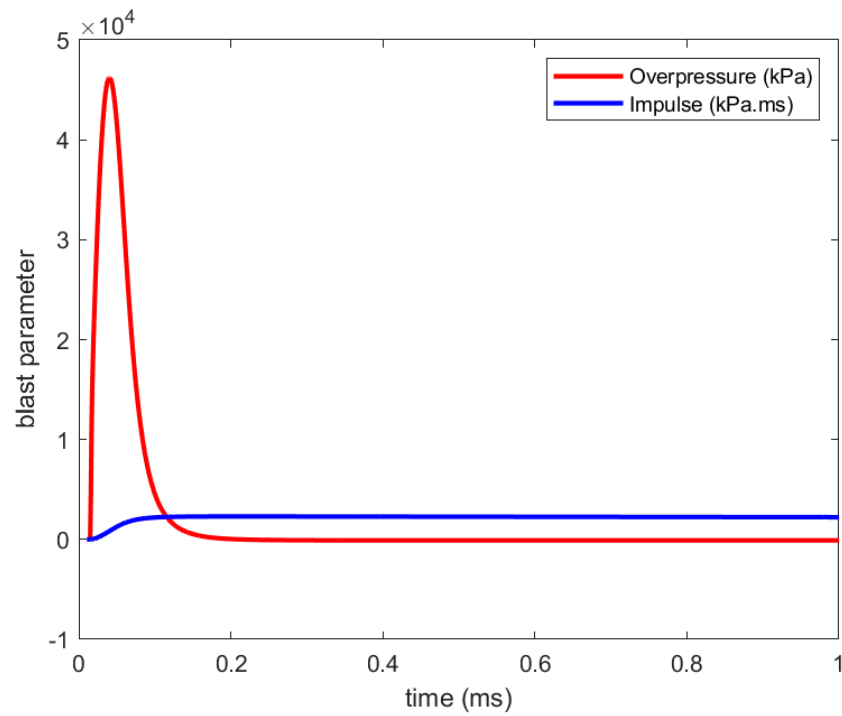


Figure 15: Overpressure versus time for target point 1 from `pt_spherical_1.csv`.

Similarly we may plot similar data for target point 28 as in Fig. 16.

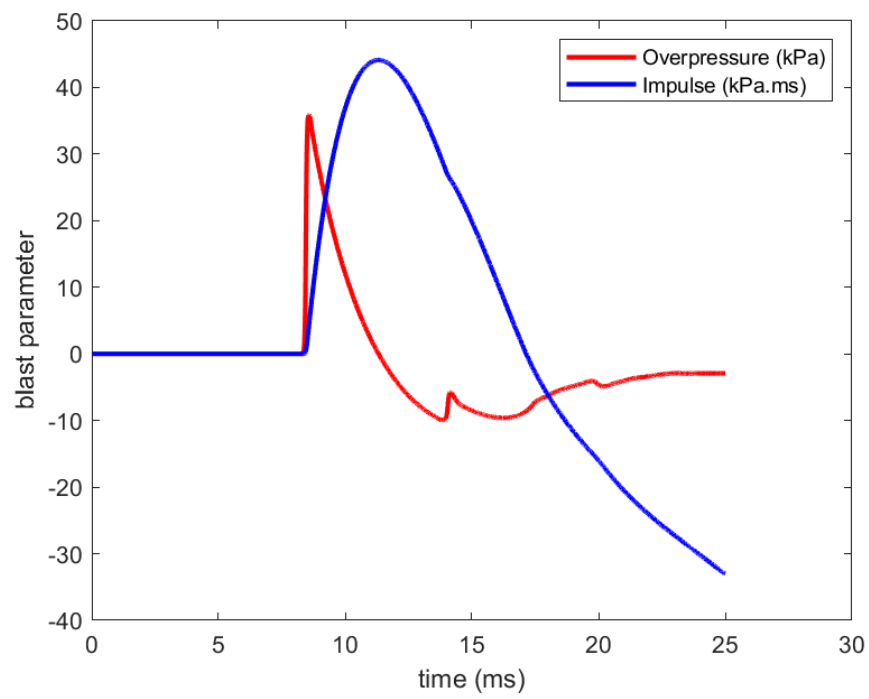


Figure 16: Overpressure versus time for target point 28 from `pt_spherical_28.csv`.

### 3.1.7 Post-processing the target\_summary\_spherical ProSAir output files

The rows of data in the file `target_summary_spherical.csv` contains summary data for each of the target points. The first 8 columns mimic lines 1 to 4 of the target point files as seen, for example, in Fig. 14. The next 7 columns give respectively:

- **Max Pressure (kPa)** - the maximum overpressure recorded at the target point.
- **Time Max Pressure (msec)** - the time at which the maximum overpressure is recorded.
- **Min Pressure (kPa)** - the minimum overpressure (i.e., the most negative underpressure) recorded at the target point.
- **Time Min Pressure (msec)** - the time at which the minimum overpressure is recorded.
- **Max Impulse (kPa-msec)** - the maximum impulse recorded at the target point.
- **Time Max Impulse (msec)** - the time at which the maximum impulse is recorded.
- **Total Impulse (kPa-msec)** - the total (cumulative) impulse recorded at the point.

Thereafter the columns of data are arranged in banks of 4 corresponding to successive positive and negative phases of the blast wave at each target point. For example columns 9-12 give the peak pressure, arrival time, impulse and phase duration of the first phase (usually the positive phase) of the blast profile at each target. Columns 13-16 give the data for the second, usually negative, phase.

In Fig. 17 we present a log-log plot of the three key blast parameters (peak pressure, positive impulse and arrival time) obtained for the first (positive) phase `target_summary_spherical.csv` and compared with the curve fits of Kingery and Bulmash [?].

Figure 18 shows the relative errors for the blast parameters:

$$\text{Relative error in peak pressure} = \frac{|P^{KB} - P^{ProSAir}|}{P^{KB}} \quad (1)$$

$$\text{Relative error in positive impulse} = \frac{|I^{KB} - I^{ProSAir}|}{I^{KB}} \quad (2)$$

$$\text{Relative error in arrival time} = \frac{|t_a^{KB} - t_a^{ProSAir}|}{t_a^{KB}} \quad (3)$$

where  $P^{KB}$ ,  $I^{KB}$ ,  $t_a^{KB}$  are the positive overpressure, positive impulse and arrival time from the Kingery Bulmash data-fits and  $P^{ProSAir}$ ,  $I^{ProSAir}$ ,  $t_a^{ProSAir}$  are those from ProSAir.

The computed data and the experimentally derived curve fits of Kingery and Bulmash are seen to be in good agreement, with an relative error of, or less than, %20 for stand-offs beyond about 0.5m.

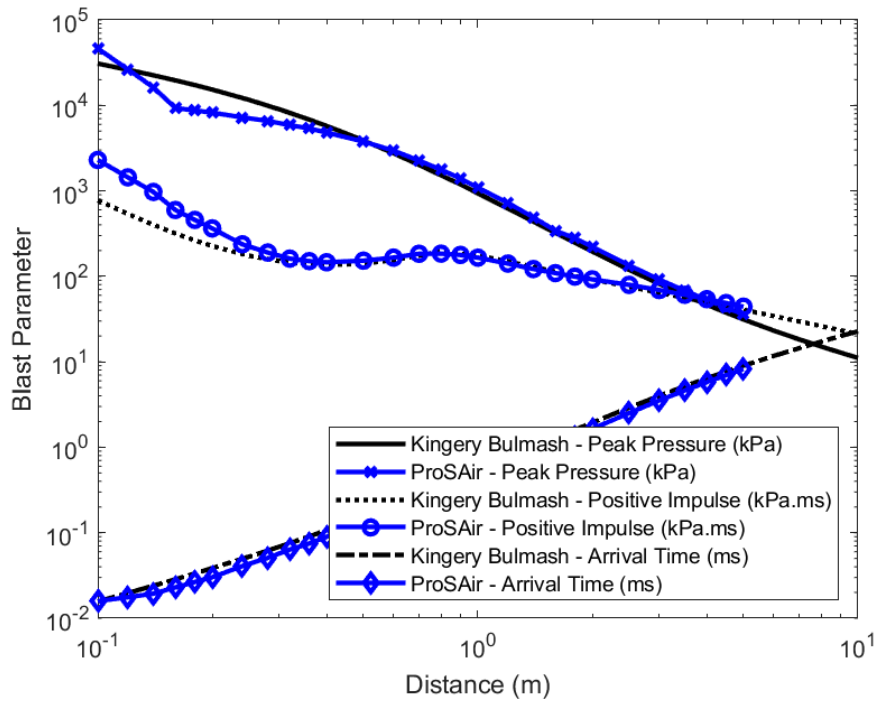


Figure 17: Comparison of Kingery Bulmash data fit blast parameters with those calculated by ProSAir on a mesh with nominal cell size 0.01m

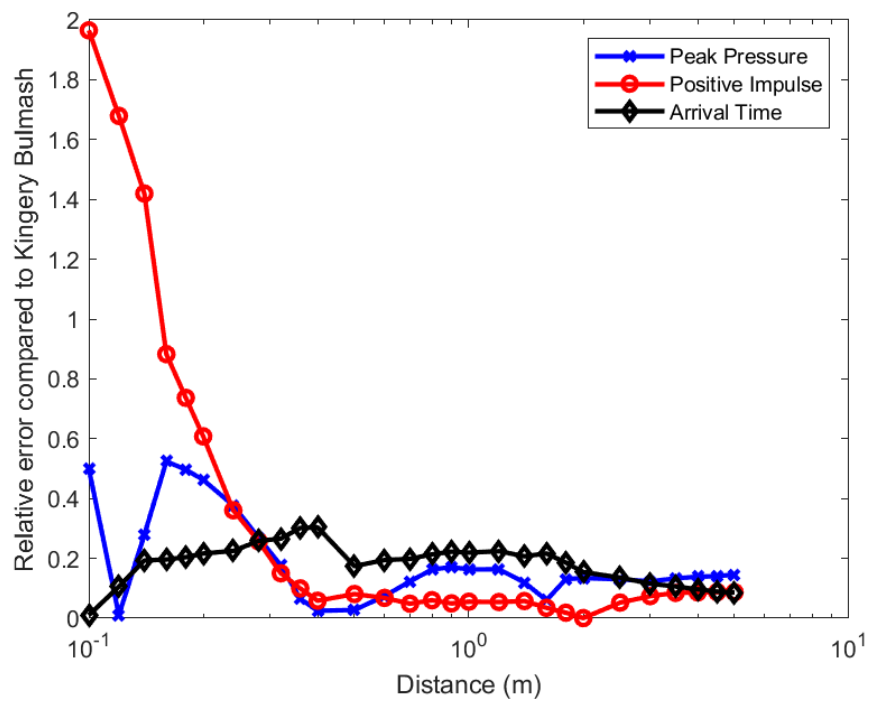


Figure 18: Relative errors of blast parameters calculated by ProSAir on a mesh with nominal cell size 0.01m compared to Kingery Bulmash data-fits

### 3.1.8 Reducing the Cell Size

It is important to perform a mesh sensitivity study for any calculation to ensure that discretisation errors (that decrease with a smaller cell size) are acceptable for the user's requirements. ProSAir input file `KingeryBulmashRefined.pr1` specifies the same problem as in the previous sections but with a **Cell Size** specified as 0.001m giving approximately 50 cells from the centre of the charge to its radius. This simulation takes significantly longer to run.

In Fig. 19 we present a log-log plot of the three key blast parameters compared with the curve fits of Kingery and Bulmash [?].

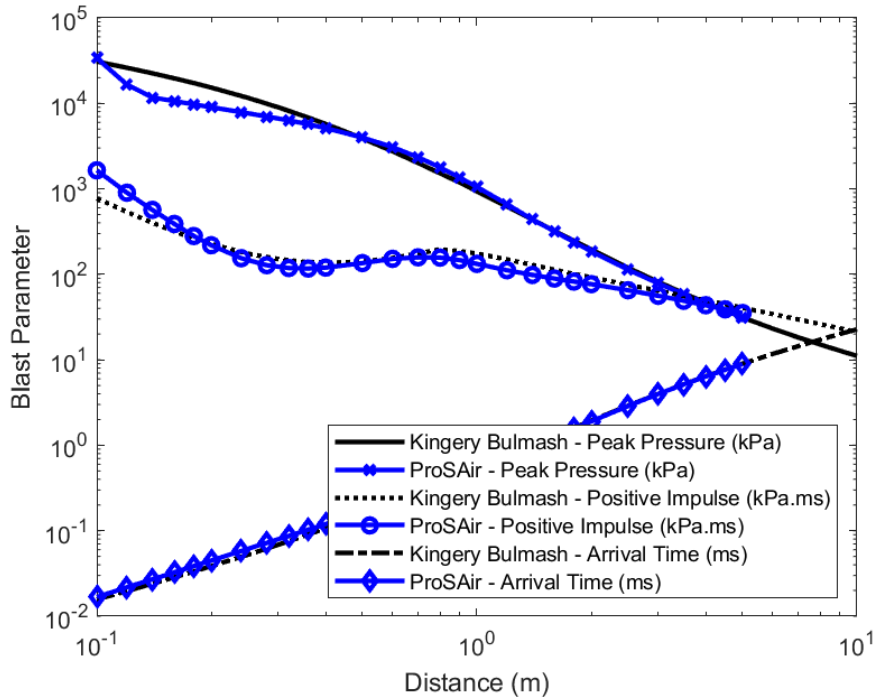


Figure 19: Comparison of Kingery Bulmash data fit blast parameters with those calculated by ProSAir on a mesh with nominal cell size 0.001m

Figure 20 shows the relative errors for the blast parameters.

Comparing Figures 19 and 20 with those on the coarse mesh (Fig. 17 and 18) we see improvements in the prediction of arrival time and peak pressure as we would expect. Paradoxically we see some worsening in the agreement of the computed positive impulse compared to the Kingery Bulmash curve fits as we decrease the mesh size. We might speculate that this is due to inadequate modelling of some physical effect - perhaps afterburning of the TNT product gases.

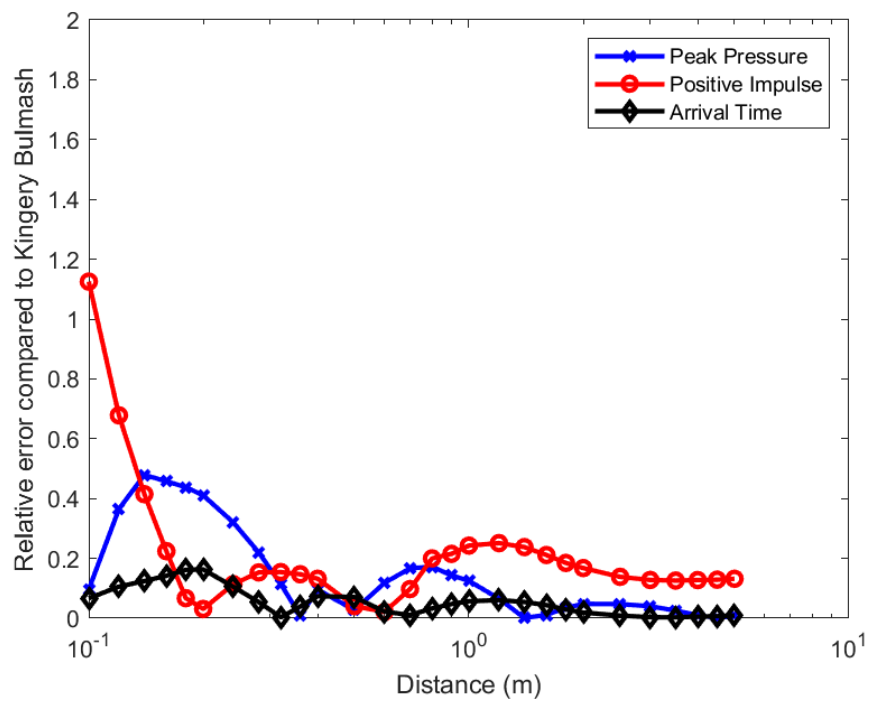


Figure 20: Relative errors of blast parameters calculated by ProSAir on a mesh with nominal cell size 0.001m compared to Kingery Bulmash data-fits

## 3.2 Height of Burst

The following example is a two-dimensional, radially symmetrical analysis of a *height of burst* scenario and is taken from the **Air3d** Users' Guide [?]. Such computations have their roots in nuclear weapons effects, but are still of interest to high explosive blast investigators because the majority of explosive accidents/incidents/experiments involve charges which are detonated a short distance above the ground. The aim of this example is to introduce the user to the cylindrically symmetric solution phase, it is not intended to be representative of any particular situation.

### 3.2.1 Problem Definition

The present example has a spherical, centrally detonated, 1kg TNT charge with its centre 2m above the ground. A schematic diagram of the scenario is shown in Figure 21. It has

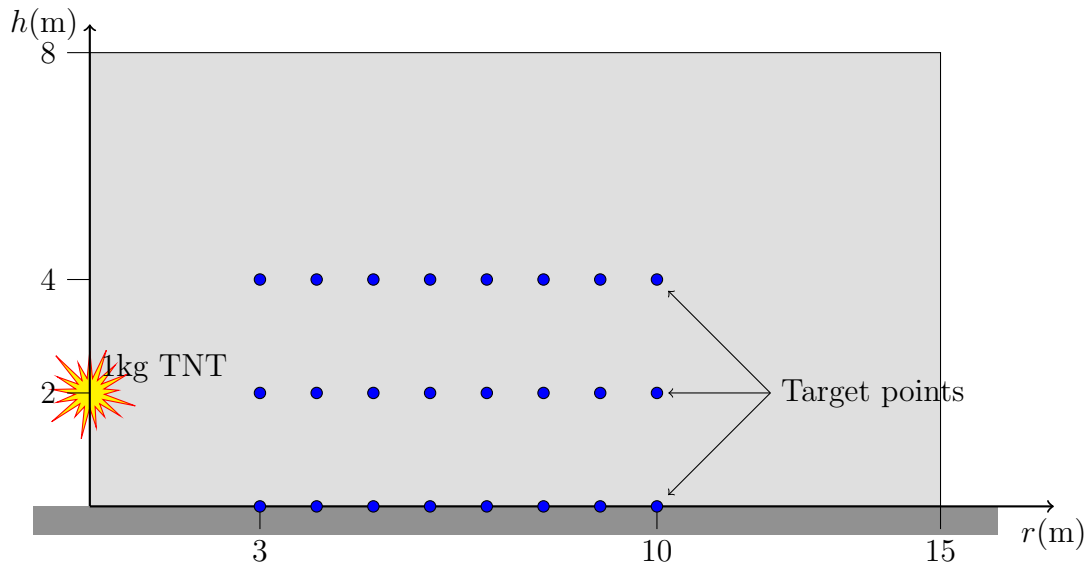


Figure 21: Schematic diagram of the Height of Burst example of Section 3.2

three linear arrays of eight target points from 3m to 10m, horizontally, in increments of 1m. The first array is at the ground surface, the second is 2m above the ground and the third is 4m above the ground. There are twenty-four monitoring points in total.

### 3.2.2 ProSAir General Settings

A suitable ProSAir input file for this example is `HoB.pr1`. On loading this file into ProSAir and selecting **Simulation->Settings** the Simulation Settings window appears displaying the **General Settings** tab as in Figure 22. The settings here are much the same as for

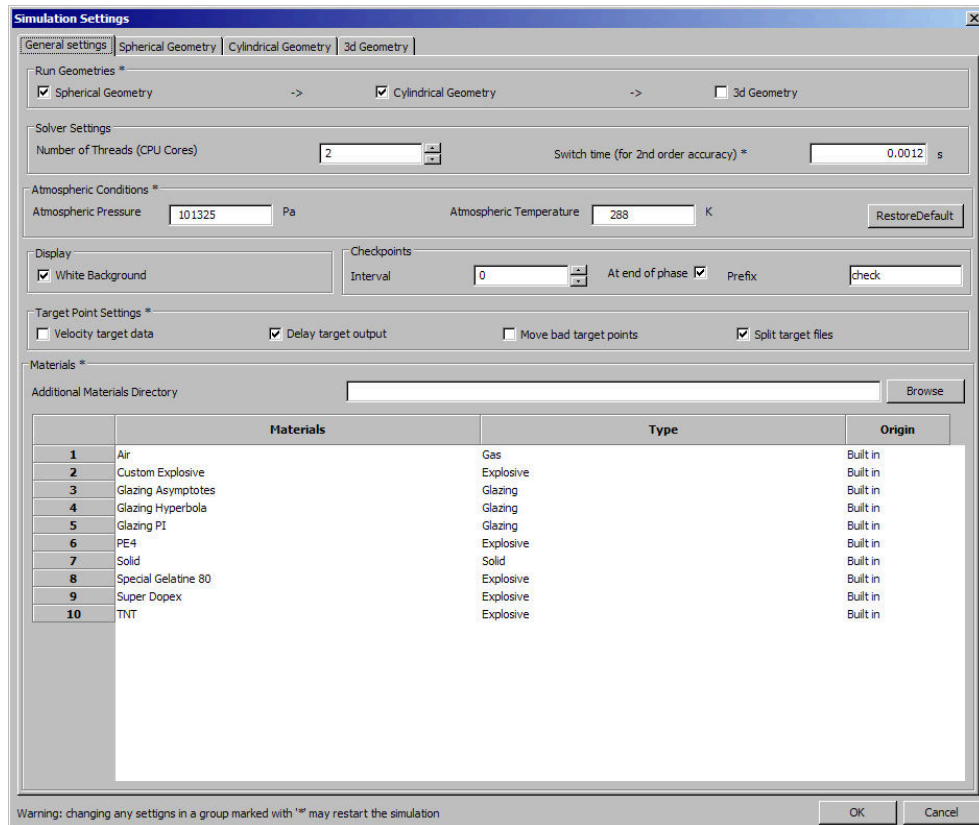


Figure 22: **General settings** tab for the Height of Burst input file `HoB.pr1`

the Kingery Bulmash test case of Section 3.1.3. The differences are:

- The **Cylindrical Geometry** tick box has been ticked, as well as the **Spherical Geometry** tick box, so that the simulation will be performed in two stages:
  1. A spherically symmetric phase until the blast wave nears the ground.
  2. A cylindrically symmetric phase to model the blast-ground interaction.
- The **Checkpoints-At end of phase** box is ticked so that a solution check point is stored at the end of the spherically symmetric phase. This permits changes to be made to the definition of the cylindrical phase and rerun without having to rerun the spherical phase.

For more information on **General Settings** see Section 4.1.

### 3.2.3 ProSAir Spherical Geometry

On selecting the **Spherical Geometry** tab the tab is displayed as in Figure 23. The

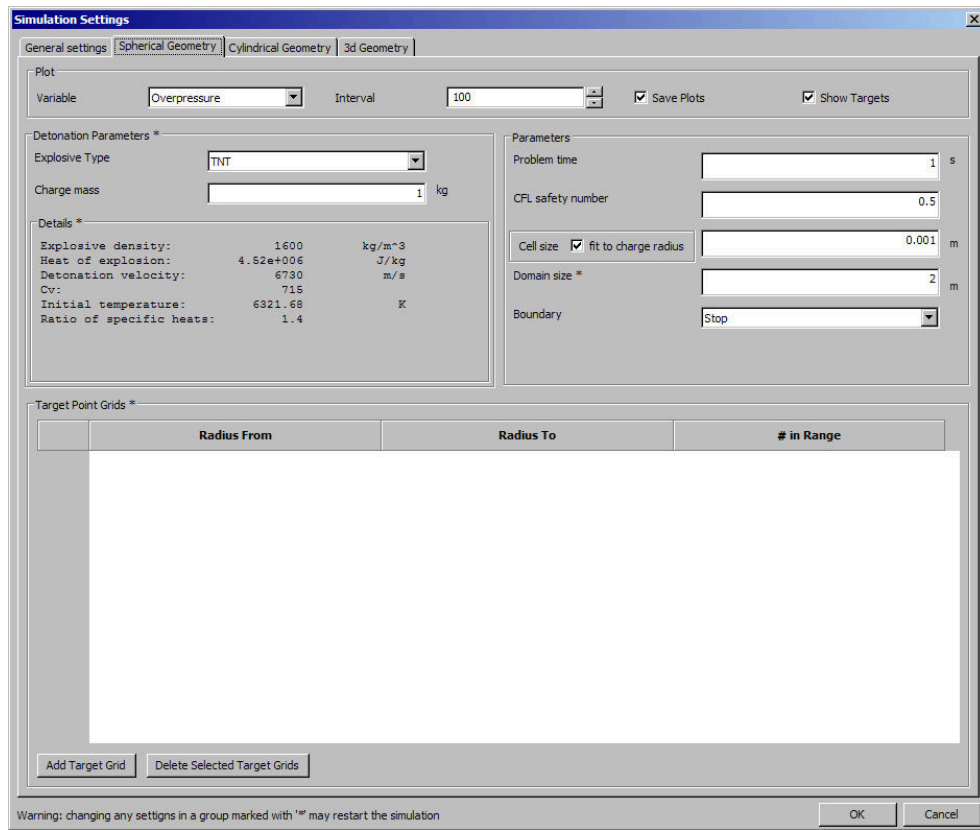


Figure 23: **Spherical Geometry** tab for the Height of Burst input file HoB.pr1

settings here are much the same as for the Kingery Bulmash test case of Section 3.1.4. The differences are:

- The **Domain size** is set to 2m, the distance from the charge centre to the ground.
- The **Boundary** has been set to **Stop** so the spherical computation will stop as the blast wave reaches 2m (the **Domain size**) from the centre of detonation.
- No target points are set as, from Figure 21, we see that these lie beyond 2m from the detonation point.

The combination of the first two settings ensures the spherical computation stops as the blast wave approaches the ground.

For more information on **Spherical Geometry** settings see Section 4.2.

### 3.2.4 ProSAir Cylindrical Geometry

On selecting the **Cylindrical Geometry** tab the tab is displayed as in Figure 24.

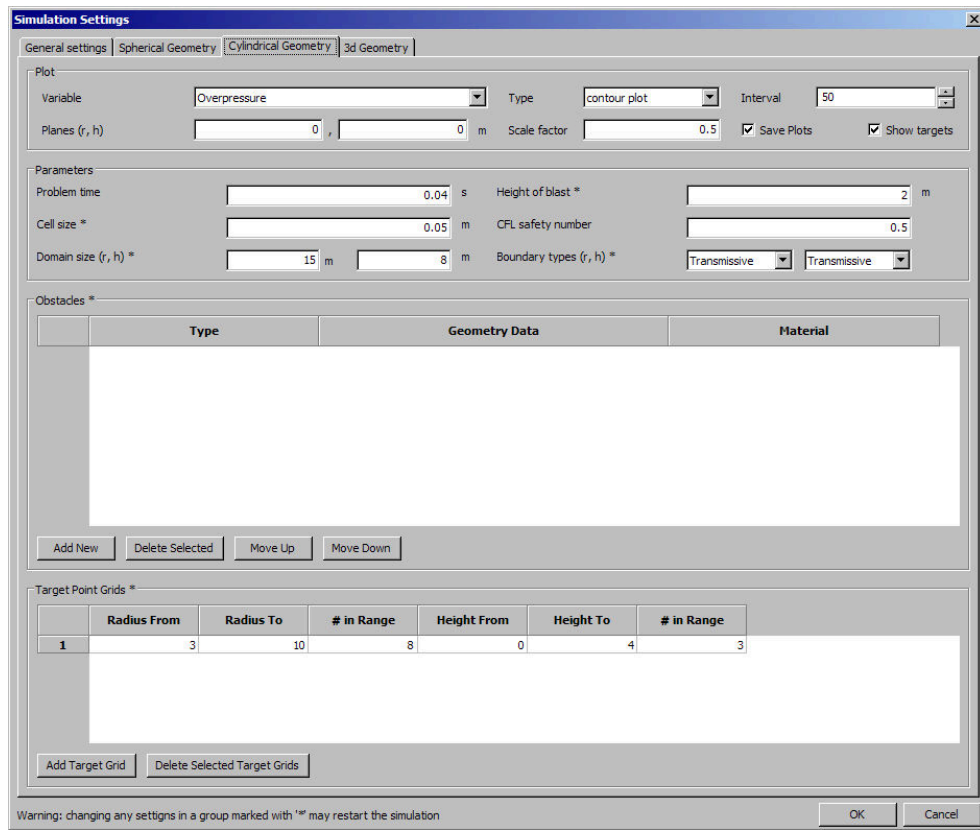


Figure 24: **Cylindrical Geometry** tab for the Height of Burst input file HoB.pr1

In each panel settings have been made as follows.

**Plot** we have opted to:

- Use **Variable** to plot the **Overpressure**.
- Use **Type** to choose a **contour plot**. Other options are: line plots for constant height **h** or constant radius **r**; or to plot **all** (contour and two line plots).
- Update the plots at an **Interval** of every 50 time steps.
- Use **Save Plots** to save png format files of the plots.
- Use **Show Targets** to display the target point locations using the target number.

See § 4.3.1 for more information on **Plot** settings.

**Parameters** we have opted for:

- A **Problem time** of 0.04s which we found from experience is sufficient for the blast wave to exit the domain.
- A **Height of blast** of 2m corresponding to the problem definition of Section 3.2.1.
- A **Cell size** of 0.05m.
- The default **CFL safety number** of 0.5.
- A **Domain size** extending to radius  $r = 15\text{m}$  and height  $h = 8\text{m}$  to match the problem specification of Section 3.2.1.
- The **Boundary types** are set to be **Transmissive** at the far field radial (**r**) and upper (**h**) boundaries to allow the shock to pass out of the domain without reflecting. Note that the lower boundary is always set to a **Reflective** boundary condition to model (perfect) shock reflection from the ground. The boundary condition at  $r = 0$  corresponds to a symmetry condition for a cylindrically symmetric computation.

See § 4.3.2 for more information on cylindrical **Parameters**.

**Obstacles** no obstacles are defined.

**Target Point Grids** we have defined a single target point grids for which the associated target points will correspond to those of Figure 21 namely a grid of 8 points in the radial direction, between  $r = 3\text{m}$  and  $r = 10\text{m}$ , and 3 points in the vertical, from  $h = 0$  to  $h = 4\text{m}$ .

See § 4.3.4 for more information on cylindrical **Target Point Grids**.

For more information on **Cylindrical Geometry** settings see Section 4.3.

### 3.2.5 Running the Simulation

On running the simulation the spherical solution phase starts and ProSAir displays a plot of overpressure with radius, evolving in time, until the shock approaches the outer boundary at radius  $R = 2\text{m}$  as seen in Figure 25.

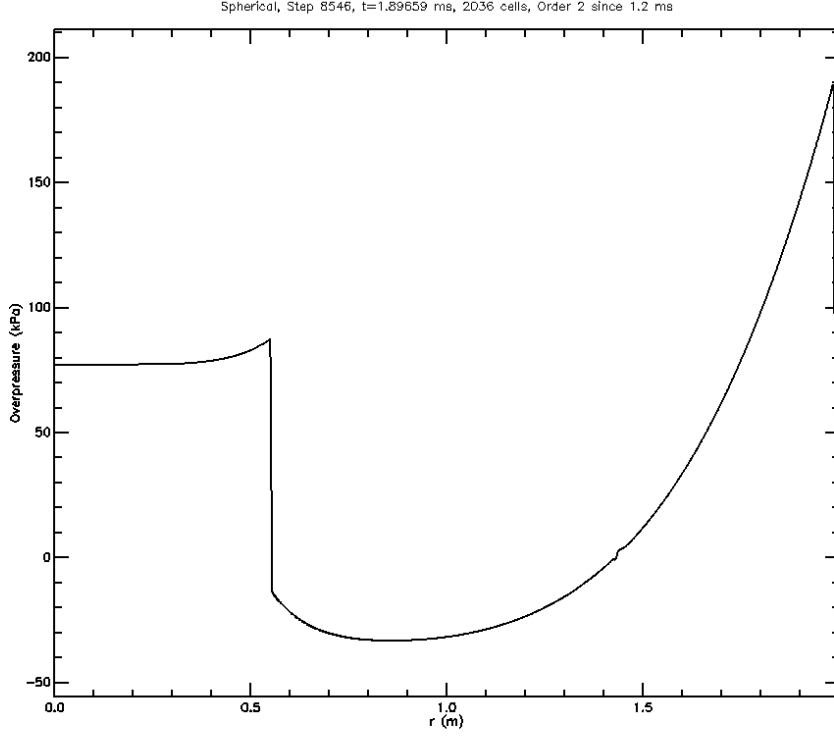


Figure 25: Overpressure versus radius after at the end of the spherically symmetric solution phase (about 1.9ms) for the Height of Burst case of Section 3.2

Solution then switches to the cylindrical phase. There is an apparent pause in computation as ProSAir transfers the spherical symmetric solution onto the cylindrically symmetric grid given the initial plot of Figure 26.

At about 3.46ms, Figure 27 shows the blast wave arriving at target point 2, the first on the  $h = 2\text{m}$  level. The reflected shock from the ground is clearly seen but has not had chance to interact with the main blast wave at target point 2. Therefore we would expect time of arrival and peak pressure at this target point to be unaffected by the presence of the ground and so in line with results from Kingery and Bulmash [?] and §3.1. It is also likely that first positive phase impulse will also be in line with Kingery and Bulmash as there will be insufficient time for the reflected shock to reach this target point before the first positive phase ends.

At about 4.99ms, Figure 28 shows the blast wave arriving at target point 1, the first

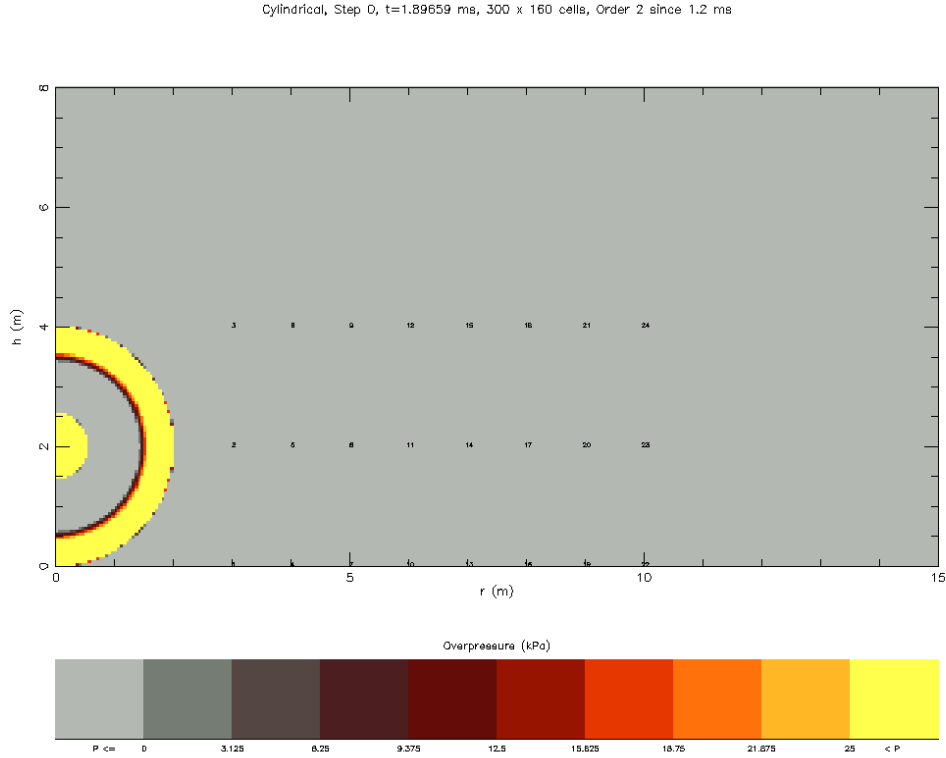


Figure 26: Overpressure contour plot at the start of the cylindrically symmetric solution phase (about 1.9ms) for the Height of Burst case of Section 3.2

on the ground, and perhaps slightly delayed at target point 3, the first on the  $h = 4\text{m}$  level. The reflected shock still has not had sufficient time to influence the flow for the target points on levels  $h = 2\text{m}$  and  $h = 4\text{m}$  but clearly a small Mach stem has formed so increasing pressures on the ground at target point 1.

By about 6.96ms, Figure 29 shows the blast wave has already gone passed target point 5 on the  $h = 2\text{m}$ , and is about to impact target 6 on the  $h = 4\text{m}$  level. Again these target points have yet to feel the effect of the reflected shock and so peak pressure, time of arrival and, most likely, impulse will not be affected by the reflection. However the increased pressure associated with the Mach stem will clearly augment the pressure and impulse on the ground at target point 4.

The flowfield continues to evolve until about 14.33ms at which point Figure 30 shows that the reflected shock will most definitely prolong the positive phase duration and impulse of target point 11 on the row  $h = 2\text{m}$  but the peak pressure and arrival time for target 14 on the same row will remain in line with that predicted by Kingery and Bulmash.

At 20.84ms Figure 31 shows that the Mach stem now extends as high as 2m so we expect the pressure and impulse on target points 20 and 23 will be increased compared to the Kingery Bulmash predictions (applicable without the ground reflection).

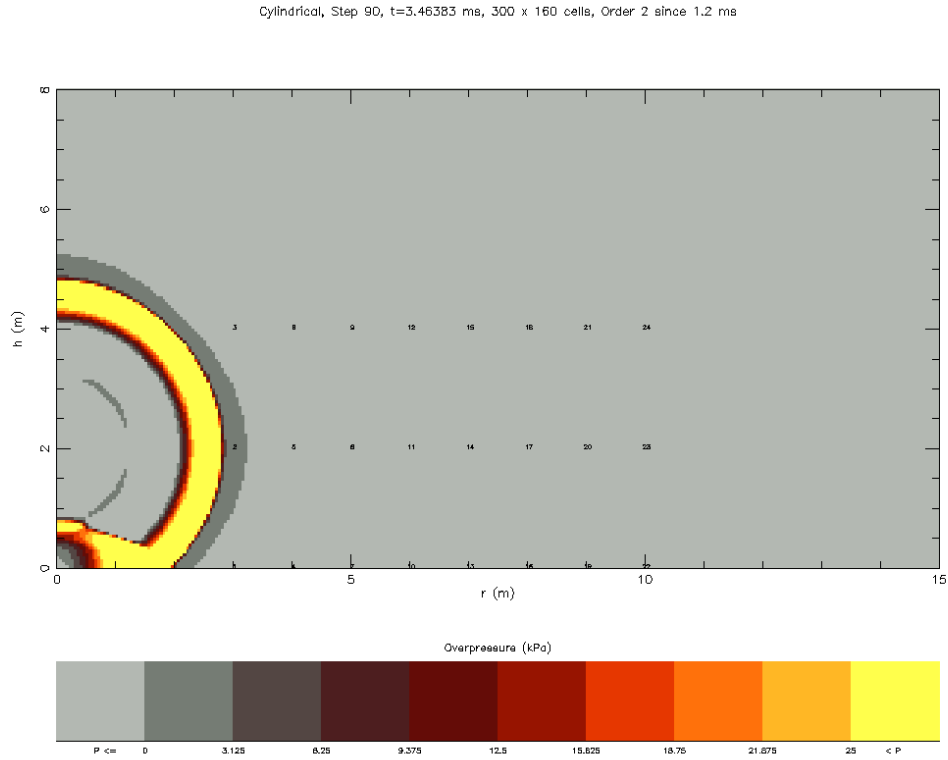


Figure 27: Overpressure contour plot as the blast wave reaches target point 2 at 3.46ms for the Height of Burst case of Section 3.2

The shock is seen arriving at the last target point on the row at 4m height in Figure 32. The target point would be expected to be unaffected by the ground reflection in terms of peak pressure and time of arrival; its impulse is likely to be augmented.

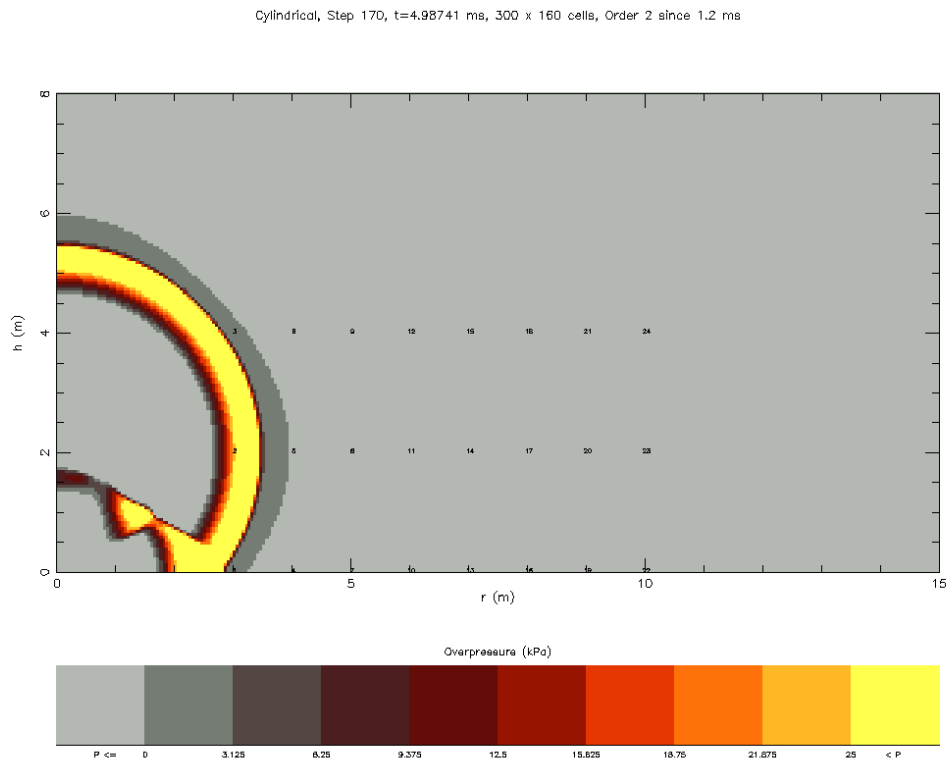


Figure 28: Overpressure contour plot as the blast wave reaches target points 1 and 3 at 4.99ms for the Height of Burst case of Section 3.2

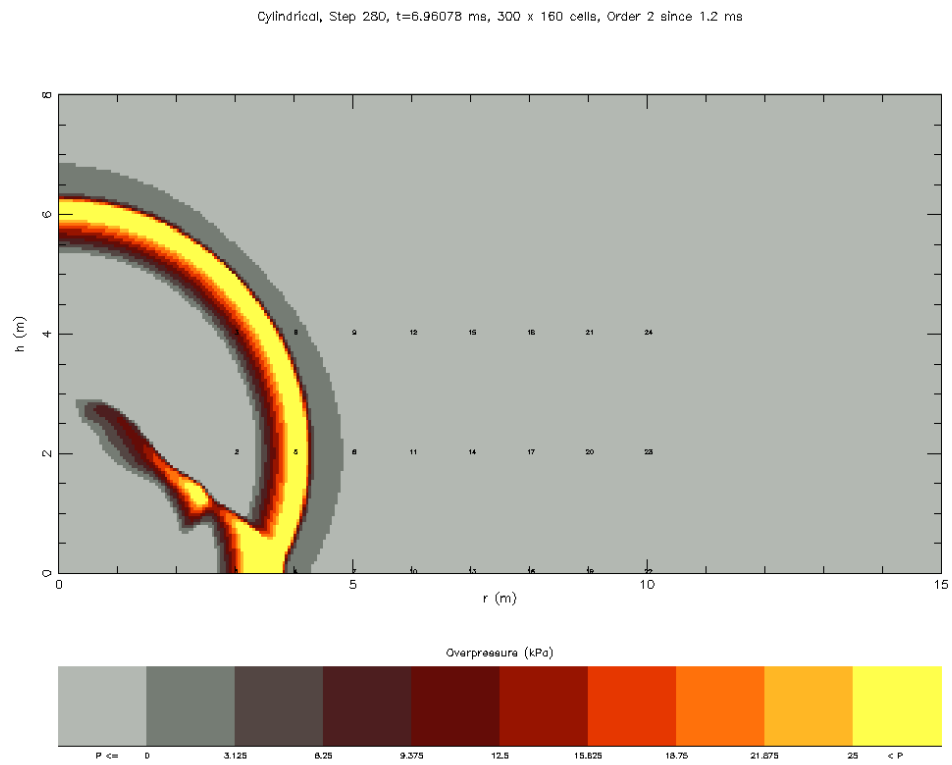


Figure 29: Overpressure contour plot at 6.96ms for the Height of Burst case of Section 3.2

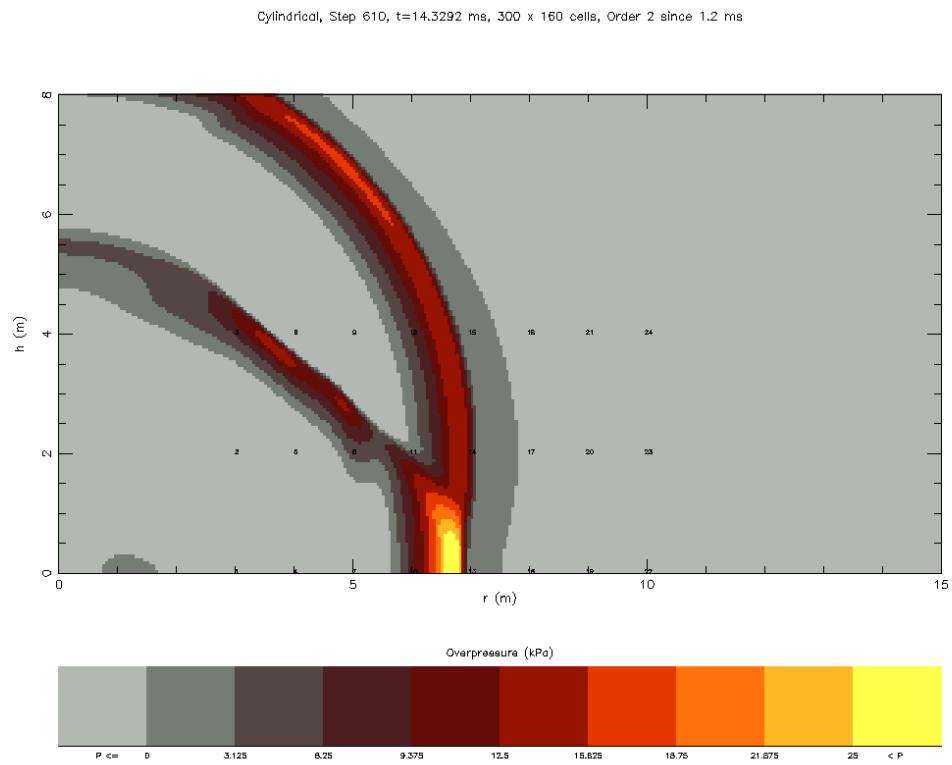


Figure 30: Overpressure contour plot at 14.33ms for the Height of Burst case of Section 3.2

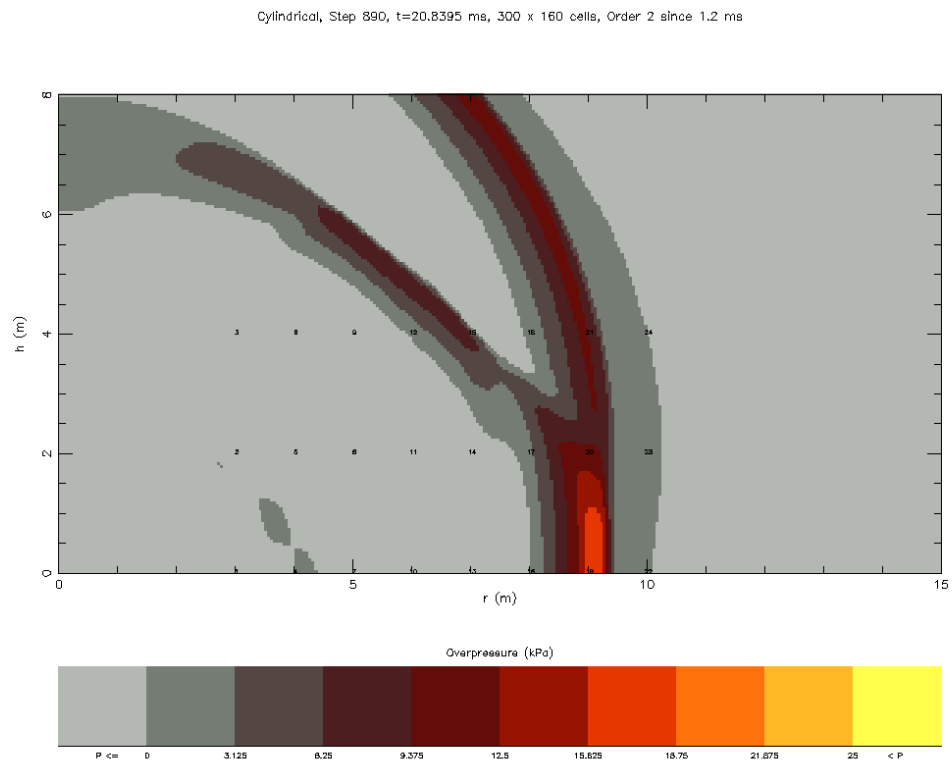


Figure 31: Overpressure contour plot at 20.84ms for the Height of Burst case of Section 3.2

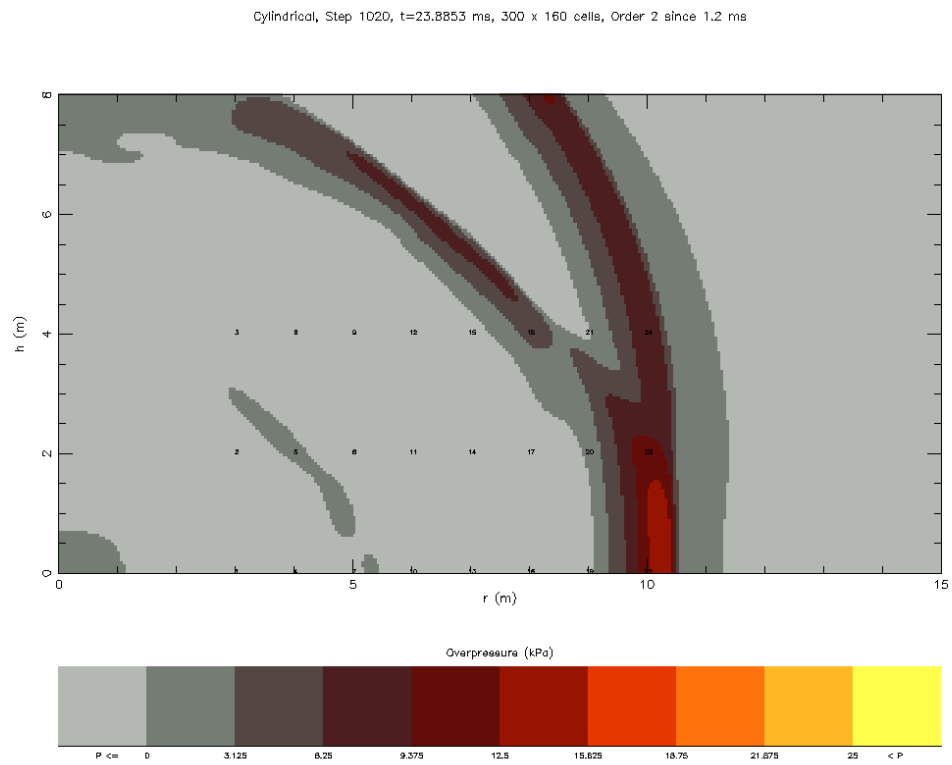


Figure 32: Overpressure contour plot at 23.89ms for the Height of Burst case of Section 3.2

### 3.2.6 Post-processing the ProSAir output files

During the course of simulation a target summary file `target_summary_cylindrical.csv` and 24 target point files `pt_cylindrical_1.csv` - `pt_cylindrical_24.csv` are produced.

On examining `target_summary_cylindrical.csv` we see that target point 12 corresponds to the target point at  $(r, h) = (6, 4)\text{m}$ . The overpressure and impulse are plotted for this point, using the data from `pt_cylindrical_12.csv`, in Figure 33. There are

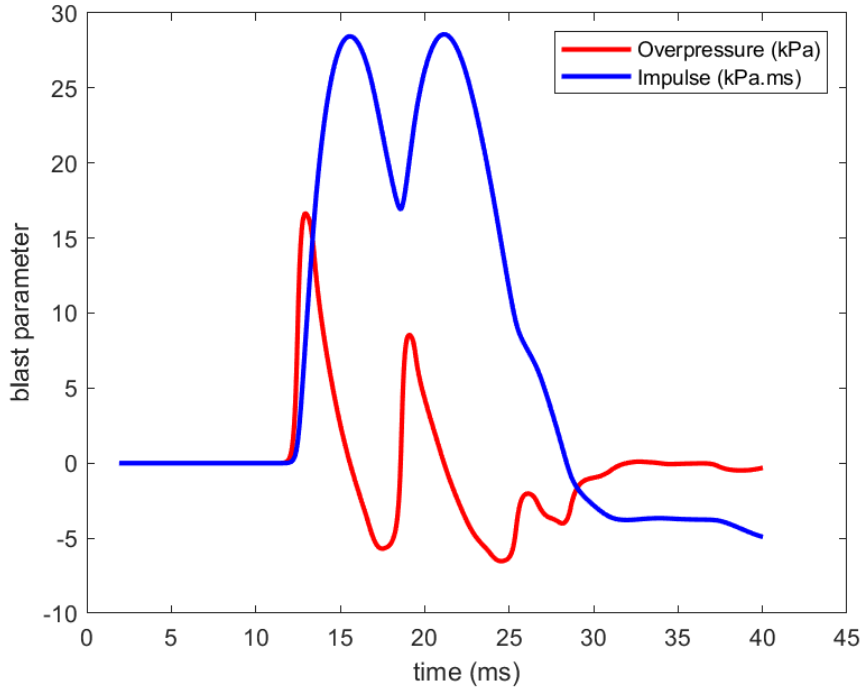


Figure 33: Overpressure and impulse for target point 12 at  $(r, h) = (6, 4)\text{m}$  for the Height of Burst case of Section 3.2

clearly two pressure peaks. The first at 11.33ms is due to the blast wave from the explosion arriving directly at the target point. The second at 18.58ms is due to the ground reflection of the blast wave. The *roundness of the peaks* indicates that this simulation may be under-resolved and that the cylindrical phase **Cell size** should be further reduced.

Figure 34 shows the maximum overpressure in the first positive phase for each of the targets for this computation. Different colours are used to distinguish targets at different heights  $h$ . In the solid black curve we also show the maximum overpressure curve from the Kingery and Bulmash curve fits  $P_s(r)$  - this may be considered as appropriate to the target point data at  $h = 2\text{m}$  (since the height of burst is 2m) provided the reflected shock off the ground has no effect. The dashed black curve shows the same Kingery Bulmash curve but adjusted for the extra distance the shock must travel from the detonation point at  $(r, h) = (0, 2)\text{m}$  to the target points at  $h = 0\text{m}$  or  $h = 4\text{m}$ , in both case  $P_s(\sqrt{r^2 + 4})$ .

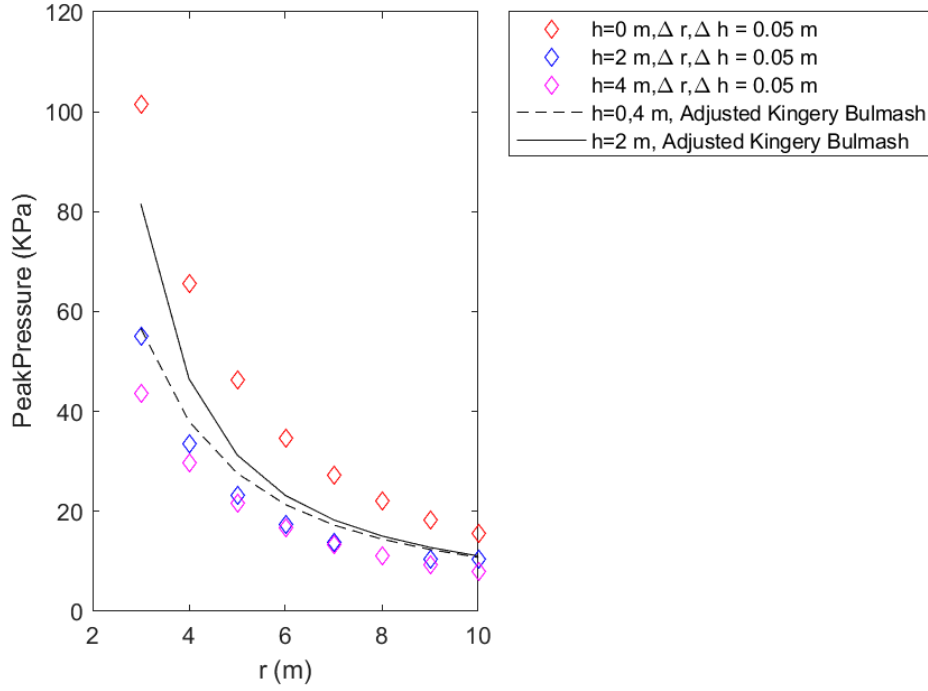


Figure 34: Maximum Overpressure versus radial coordinate  $r$  for all the targets of Fig. 21 on a mesh with cylindrical spacing 0.05m for the Height of Burst case of Sec. 3.2

We note the ProSAir data for  $h = 2\text{m}$  underpredicts the Kingery Bulmash data of the solid black curve. The ProSAir data for  $h = 0\text{m}$  is significantly above the corresponding dashed Kingery Bulmash curve - this is to be expected as the loading on the targets at  $h = 0\text{m}$  has been augmented by the ground reflection and Mach stem formation. The ProSAir data for  $h = 4\text{m}$  is below the corresponding dashed Kingery Bulmash curve. The underprediction of the data at  $h = 2\text{m}$  and  $h = 4\text{m}$  would indicate the computation is not mesh converged.

Figure 35 shows the impulse in the first positive phase for each of the targets for this computation. In a similar manner to that for the overpressure in Fig. 34 we also show the Kingery Bulmash data for spherical airblast without ground reflection. At  $h = 2\text{m}$  the ProSAir data is in reasonable agreement with the Kingery Bulmash data out to  $r = 5\text{m}$ . Beyond that the ProSAir data is significantly higher. This is due to the extra impulse being carried by the ground reflection which reaches the target points at  $h = 2\text{m}$  sufficiently soon after the primary shock for such targets beyond  $5\text{m}$  as seen in Fig. 30. The targets at  $h = 0\text{m}$  all have enhanced impulse compared to Kingery and Bulmash due to the ground reflection. The impulse for points at  $h = 4\text{m}$  is underpredicted.

The difference in arrival times at the different target point heights seen in Figure 36 is explained at small radius due to the variation in stand-off distance - stand-off distance for  $h = 0, 4\text{m}$  is larger than that for  $h = 2\text{m}$  at the same radius  $r$ . The magnitude of the

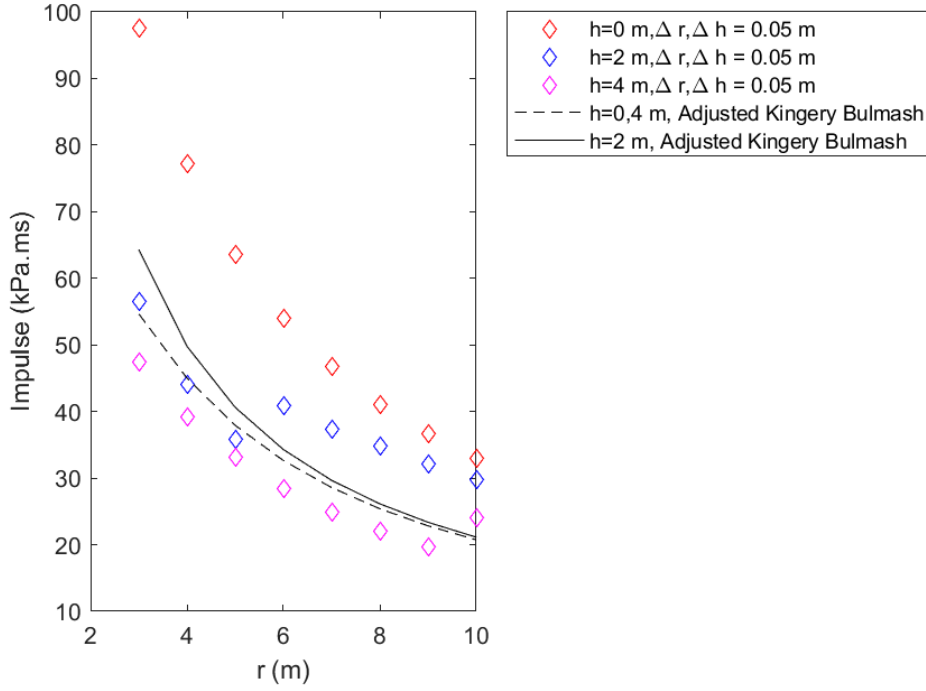


Figure 35: Impulse versus radial coordinate  $r$  for all the targets of Fig. 21 on a mesh with cylindrical spacing 0.05m for the Height of Burst case of Sec. 3.2

difference is in line with that from the Kingery Bulmash curves. At larger scaled distance the arrival time for  $h = 0\text{m}$  is seen to be in advance of that for  $h = 4\text{m}$  - this is due to the higher pressures and hence shock speed of the Mach stem formed by the ground reflection.

### 3.2.7 Convergence Study

We repeat the analysis of §3.2.2-3.2.6 on a sequence of meshes, halving the cell spacing for each.

Figure 37 shows the convergence of the peak overpressure. At all target points we see an increase in the overpressure as the mesh is refined. This is to be expected as the shock peak tends to be reduced by the actions of numerical diffusion. As the mesh is refined the numerical diffusion is reduced and the peak increases. We note that the size of the difference in peak pressure for each refinement of the mesh at each target point is reducing as the mesh is refined indicating that the solution is converging as the mesh size is reduced. We also note that the absolute size of the difference strongly reduces with radial distance and weakly reduces with height.

Figure 37 shows the convergence of the impulse associated with the first positive phase. This demonstrates convergence to engineering accuracy on all meshes. The only noticeable change is at  $(r, h) = (6, 2)\text{m}$  for which there is a small decrease in impulse as the mesh size

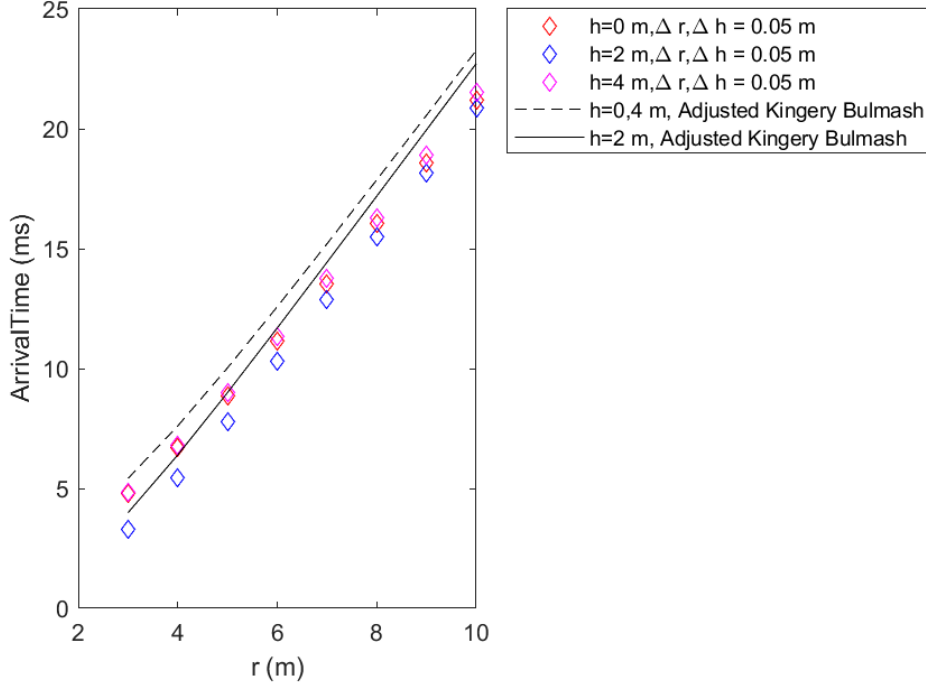


Figure 36: Arrival time versus radial coordinate  $r$  for all the targets of Fig. 21 on a mesh with cylindrical spacing 0.05m for the Height of Burst case of Sec. 3.2

is reduced from 0.025m to 0.0125m - this effect is likely due to the improved resolution of the main and reflected shocks as they interact at target point 11 as indicated by Figure 30.

Figure 39 shows the convergence of the shock arrival time. There is clearly some variation as the mesh is refined though again the values do appear to be converging with decreasing mesh size.

Figure 40 shows the peak overpressure for the targets on the most refined mesh used together with the adjusted Kingery Bulmash curves and the similarly adjusted ProSAir data from the Kingery Bulmash simulation of Sec. 3.1.8. We would expect the Kingery Bulmash data, and the Kingery Bulmash simulation to be in good agreement with the present simulations for all the targets at  $h = 4\text{m}$  and up to about  $r = 9\text{m}$  for  $h = 2\text{m}$  (c.f., Fig. 31) - this is indeed the case. Any differences may be attributed to the present simulation using a mesh spacing about 6 times larger than that of the Kingery Bulmash simulation. Clearly the  $h = 0\text{m}$  data cannot be approximated by the Kingery Bulmash data due to the strong ground reflection.

Figure 41 shows the first positive phase impulse for the targets on the most refined mesh used together with the adjusted Kingery Bulmash curves and the similarly adjusted ProSAir data from the Kingery Bulmash simulation of Sec. 3.1.8. We would expect the Kingery Bulmash data, and the Kingery Bulmash simulation to be in good agreement with

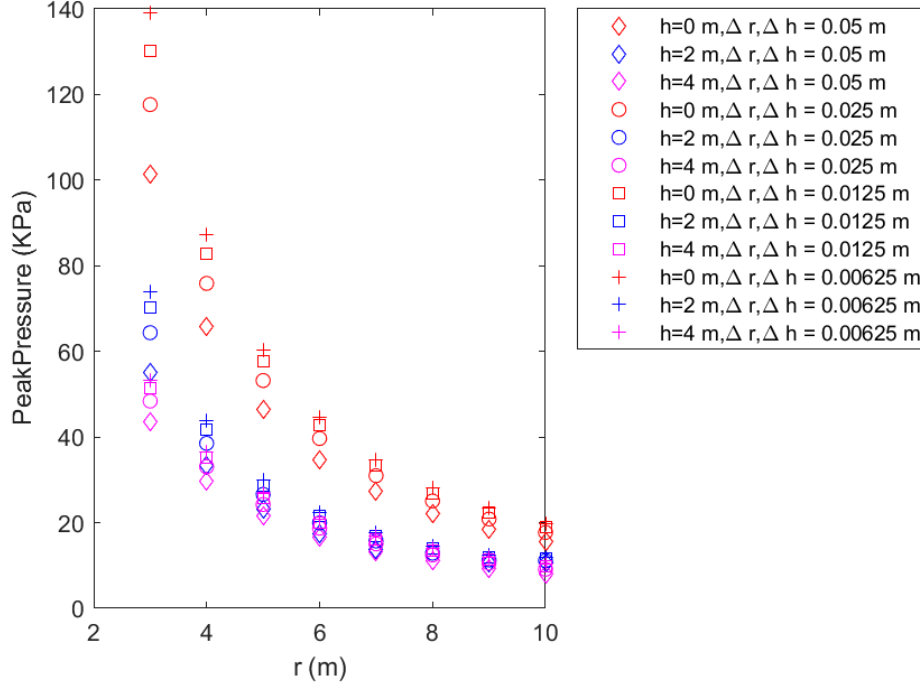


Figure 37: Convergence of peak overpressure versus radial coordinate with mesh spacing for the Height of Burst case of Sec. 3.2

the present simulations for all the targets at  $h = 4\text{m}$  and up to about  $r = 9\text{m}$  for  $h = 2\text{m}$  (c.f., Fig. 31) - qualitatively this is so but both the Kingery Bulmash simulation and the current simulation underpredict the impulse in line with Fig. 19. There is a large increase in impulse for the  $h = 2\text{m}$  data at  $r = 6\text{m}$  due to the reflected ground shock augmenting the impulse of the first positive phase. Clearly the  $h = 0\text{m}$  data cannot be approximated by the Kingery Bulmash data due to the strong ground reflection which was also seen in the overpressure data of Fig. 40.

Figure 42 shows the shock arrival time for the targets on the most refined mesh used together with the adjusted Kingery Bulmash curves and the similarly adjusted ProSAir data from the Kingery Bulmash simulation of Sec. 3.1.8. We would expect the Kingery Bulmash data, and the Kingery Bulmash simulation to be in good agreement with the present simulations for all the targets at  $h = 4\text{m}$  and up to about  $r = 9\text{m}$  for  $h = 2\text{m}$  (c.f., Fig. 31) - this is indeed so. The arrival time at the  $h = 0\text{m}$  target points underpredicts the adjusted Kingery Bulmash data: this is as expected as the shock, augmented by the ground reflection, has higher pressure and hence greater shock speed so reducing arrival time.

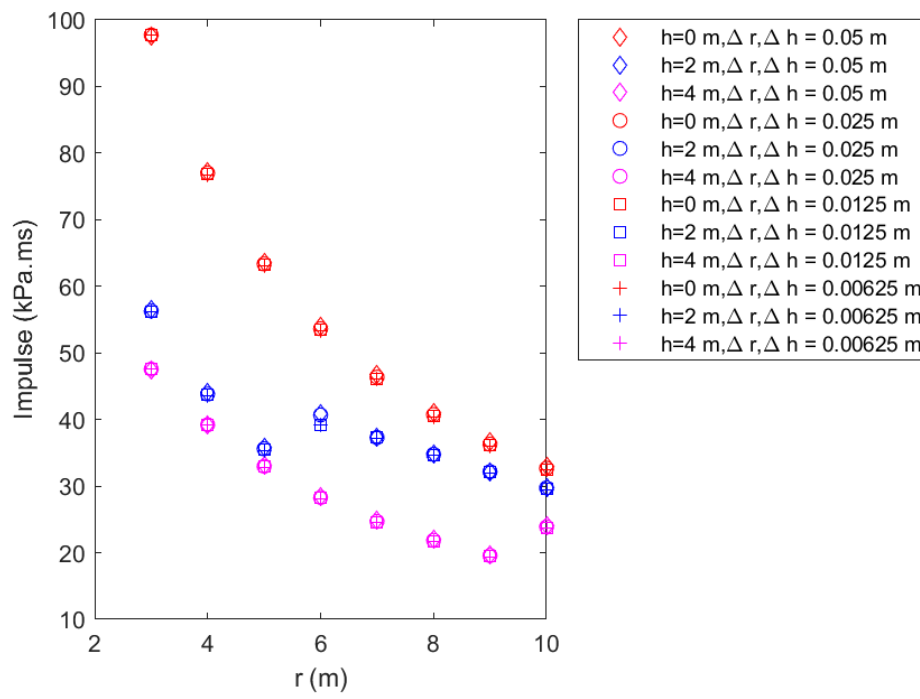


Figure 38: Convergence of impulse versus radial coordinate with mesh spacing for the Height of Burst case of Sec. 3.2

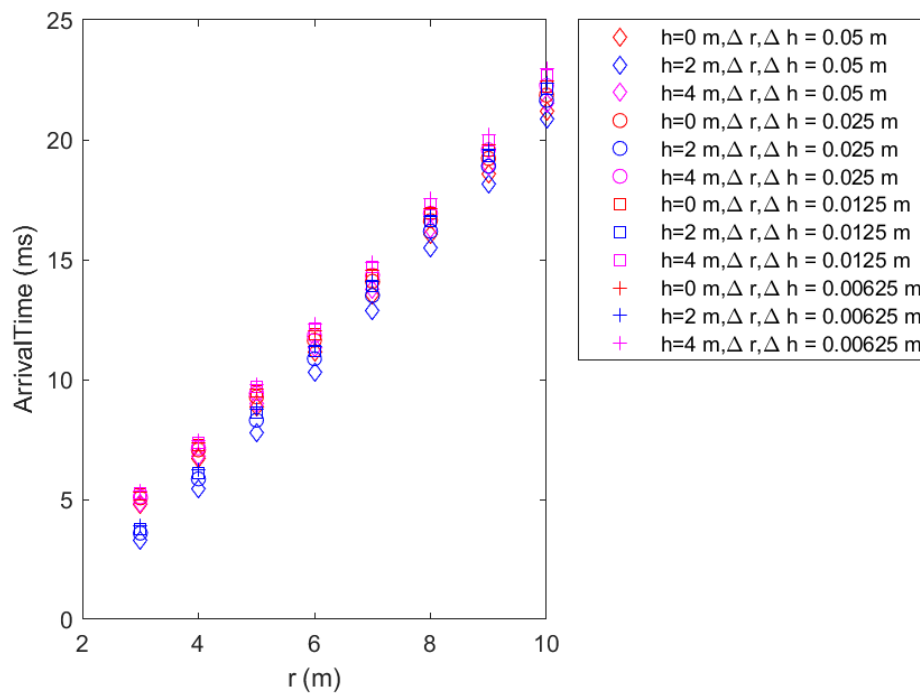


Figure 39: Convergence of arrival time versus radial coordinate with mesh spacing for the Height of Burst case of Sec. 3.2

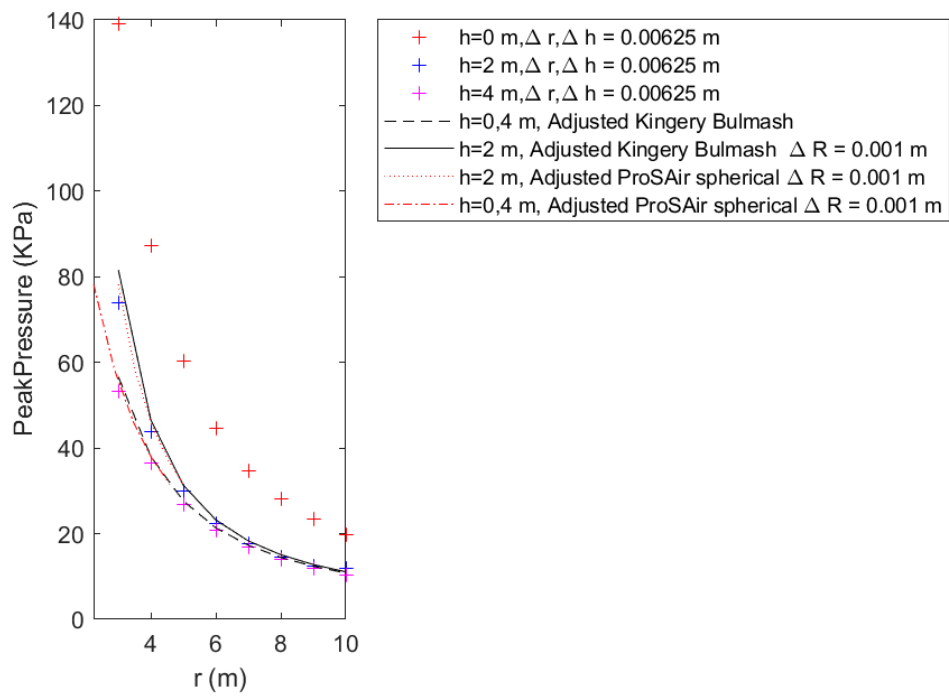


Figure 40: Peak overpressure versus radial coordinate for a mesh spacing of 0.00625m for the Height of Burst case of Sec. 3.2

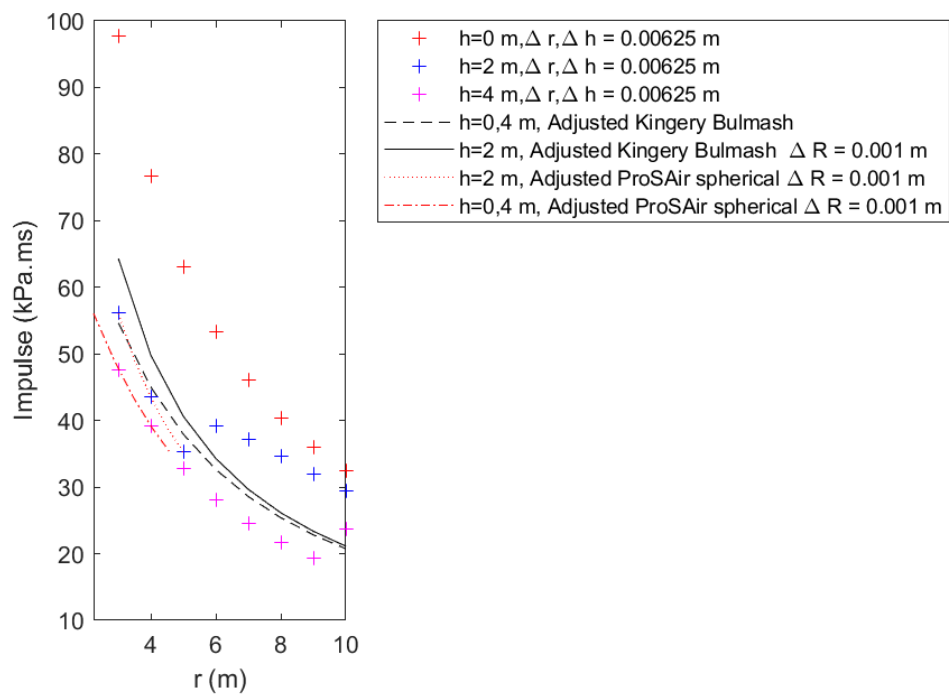


Figure 41: Impulse versus radial coordinate for a mesh spacing of 0.00625m for the Height of Burst case of Sec. 3.2

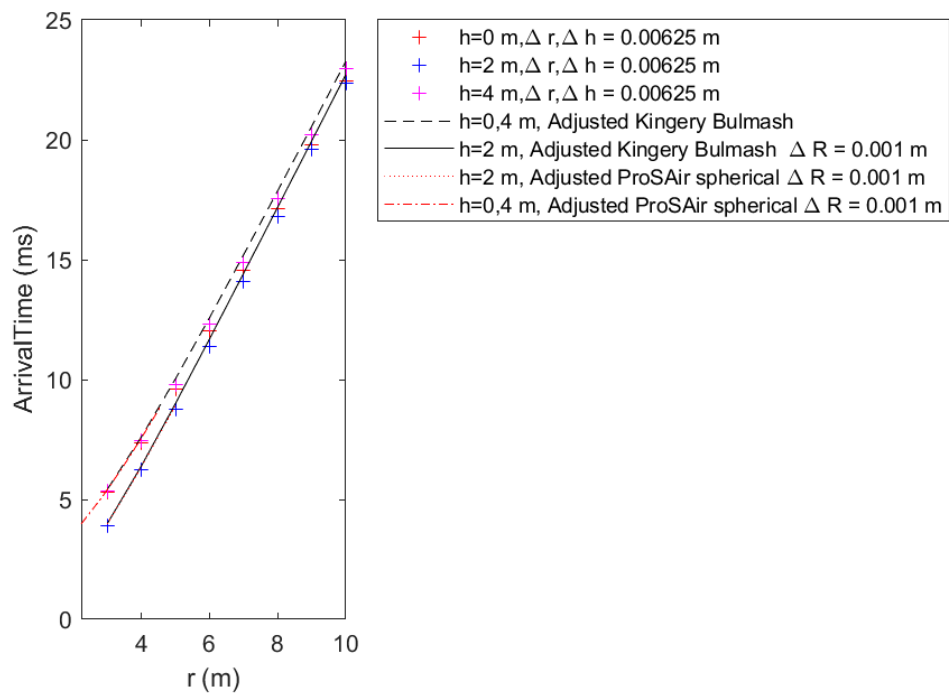


Figure 42: Arrival Time versus radial coordinate for a mesh spacing of 0.00625m for the Height of Burst case of Sec. 3.2

### 3.3 Cylindrical Containment

Another example where cylindrical symmetry may be exploited is when we consider an explosion centred on the axis of symmetry of a structure with cylindrical symmetry.

#### 3.3.1 Problem Definition

We consider a spherical charge of 1kg of PE4 detonated 1m above the centre of the internal base of a cylindrical containment vessel of internal radius 3m, internal height 4m, wall thickness 0.2m and that has a circular 250mm vent centred on the centre of the cylinder's top as seen in Fig. 43(a) [?]. Both cylinder and top are 0.096m thick.

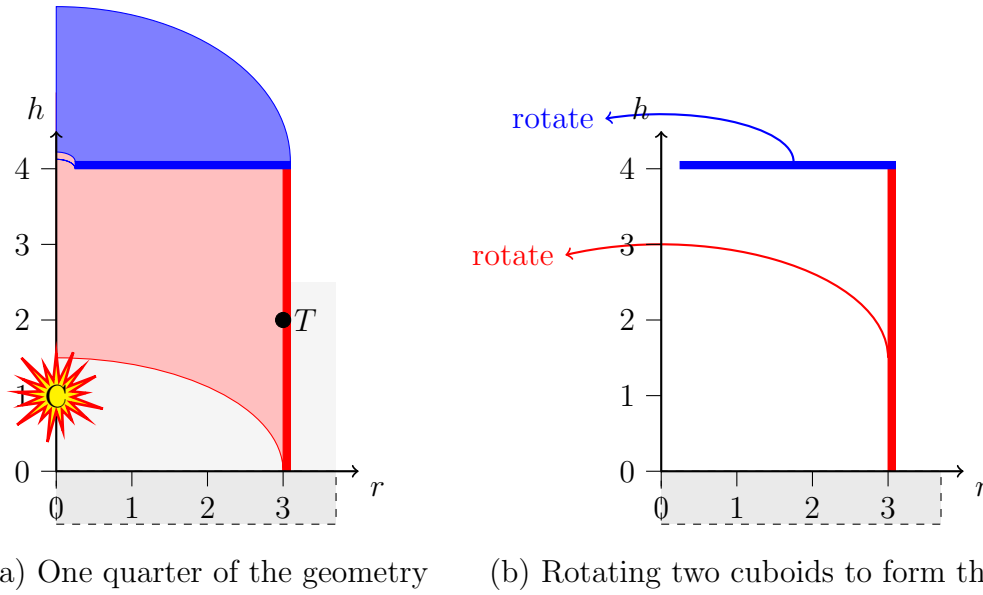


Figure 43: Geometry of the cylindrical containment cylinder. C and T denote the location of the charge and required target point

We wish to determine the pressure and temperature history at a point located on the cylindrical inner wall at a height of 2m above the cylinder's floor.

### 3.3.2 ProSAir General Settings

A suitable ProSAir input file for this example is `CylindricalContainment.pr1`. On loading this file into ProSAir and selecting **Simulation->Settings** the Simulation Settings window appears displaying the **General Settings** tab as in Figure 44. The settings here

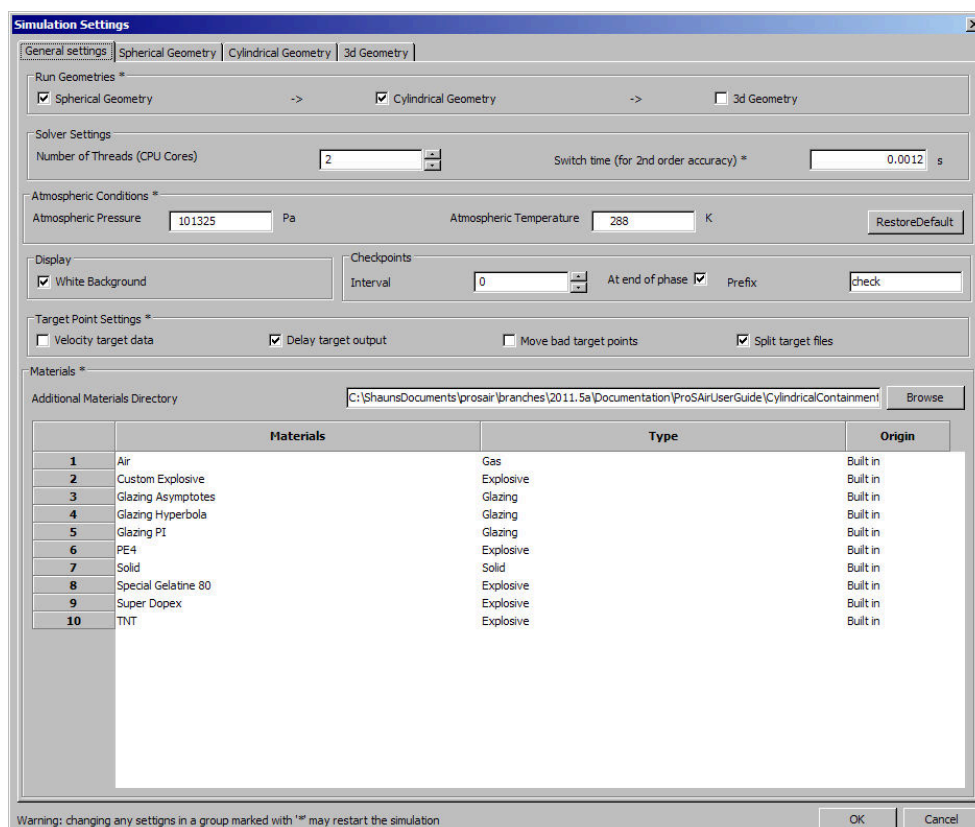


Figure 44: **General settings** tab for the Cylindrical Containment input file `CylindricalContainment.pr1`

are the same as for the Height of Burst test case of Section 3.2.

For more information on **General Settings** see Section 4.1.

### 3.3.3 ProSAir Spherical Geometry

On selecting the **Spherical Geometry** tab the tab is displayed as in Figure 45. The

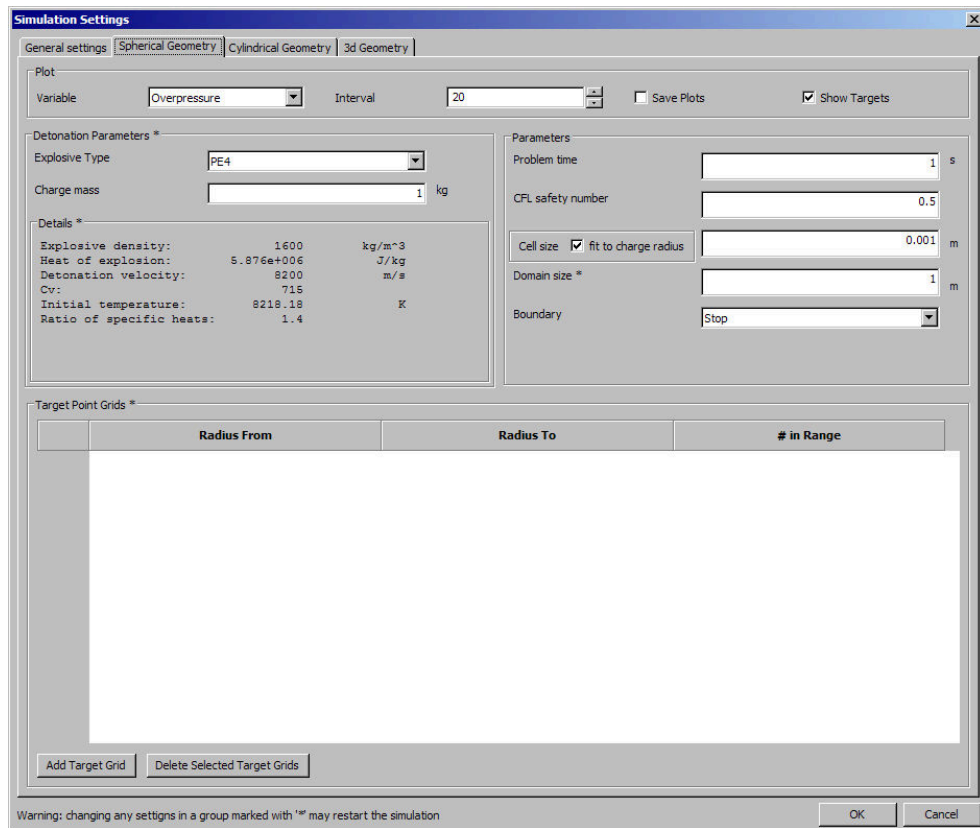


Figure 45: **Spherical Geometry** tab for the Height of Burst input file `CylindricalContainment.pr1`

settings here are much the same as for the Height of Burst test case of Section 3.2.3. The differences are:

- The **Domain size** is set to 1m, the distance from the charge centre to the ground.
- The **Explosive Type** is set to **PE4**.

For more information on **Spherical Geometry** settings see Section 4.2.

### 3.3.4 ProSAir Cylindrical Geometry

On selecting the **Cylindrical Geometry** tab the tab is displayed as in Figure 46.

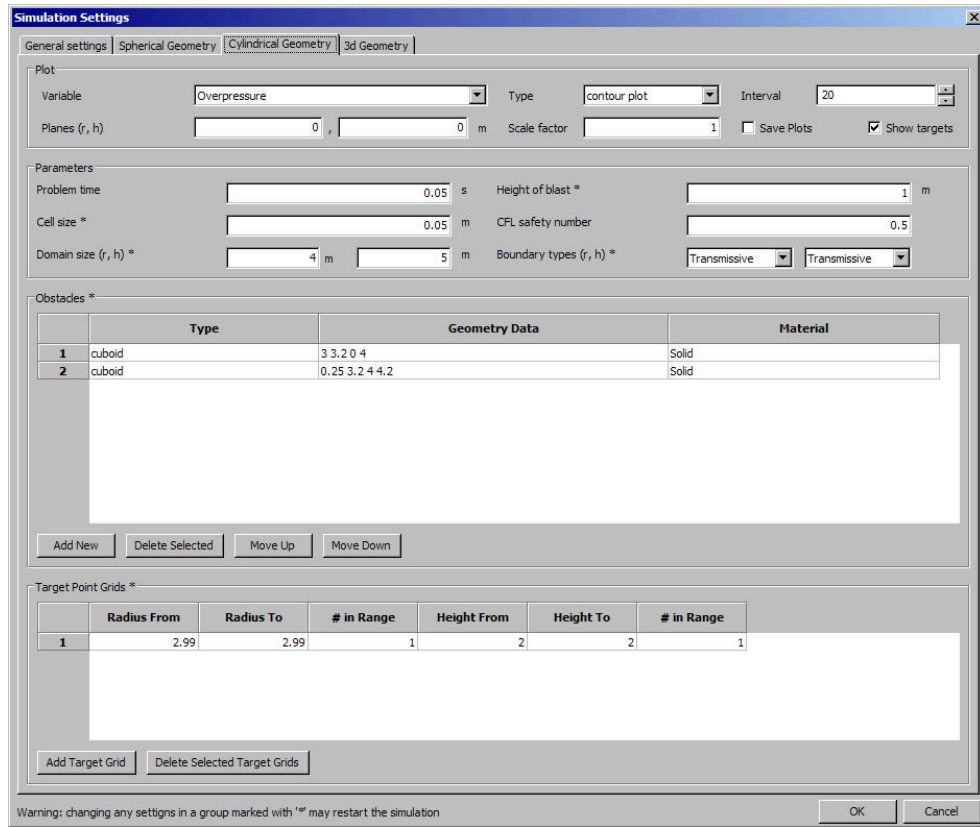


Figure 46: **Cylindrical Geometry** tab for the Cylindrical Containment input file CylindricalContainment.pr1

In the **Plot**, **Parameters** and **Target Point Grids** panel settings are much as for the Height of Burst test case of Section 3.2.4 except for:

**Plot** we have opted to:

- Update the plots at an **Interval** of every **20** time steps.
- Opted not to **Save Plots**.

See § 4.3.1 for more information on **Plot** settings.

**Parameters** we have opted for:

- A **Problem time** of 0.05s.
- A **Height of blast** of 1m corresponding to the problem definition of Section 3.3.1.

- A **Cell size** of 0.05m.
- A **Domain size** extending to radius  $r = 4\text{m}$  and height  $h = 5\text{m}$  to exceed the cylinder dimensions specified in Section 3.3.1.

See § 4.3.2 for more information on cylindrical **Parameters**.

**Target Point Grids** we have defined a single target point grids containing a single target point at  $(r, h) = (2.99, 2)\text{m}$ . Note that the radial distance  $r = 1.99\text{m}$  is slightly less than the desired radial location  $r = 2\text{m}$  to ensure the target point lies in a flow field grid cell and not the cylinder wall.

See § 4.3.4 for more information on Cylindrical **Target Point Grids**.

However in the **Obstacles** panel we have defined two **solid** obstacles.

1. A cuboid (rotated rectangle) with  $r$  coordinate ranging from 3m to 3.2m and  $h$  coordinate ranging from 0m to 4m. Under cylindrical symmetry this rectangle will be rotated about the  $h$ -axis, as seen in red in Fig. 43(b) to form the cylindrical wall of the vessel.
2. A cuboid (rotated rectangle) with  $r$  coordinate ranging from 0.25m to 3.2m and  $h$  coordinate ranging from 4m to 4.2m. Under cylindrical symmetry this rectangle will be rotated about the  $h$ -axis, as seen in blue in Fig. 43(b) to form a disc making up the roof of the vessel.

For more information on **Cylindrical Geometry** settings see Section 4.3.

### 3.3.5 Running the Simulation

On running the simulation the spherical solution phase starts and ProSAir displays a plot of overpressure with radius, evolving in time, until the shock approaches the outer boundary at radius  $R = 1\text{m}$  as seen in Figure 47.

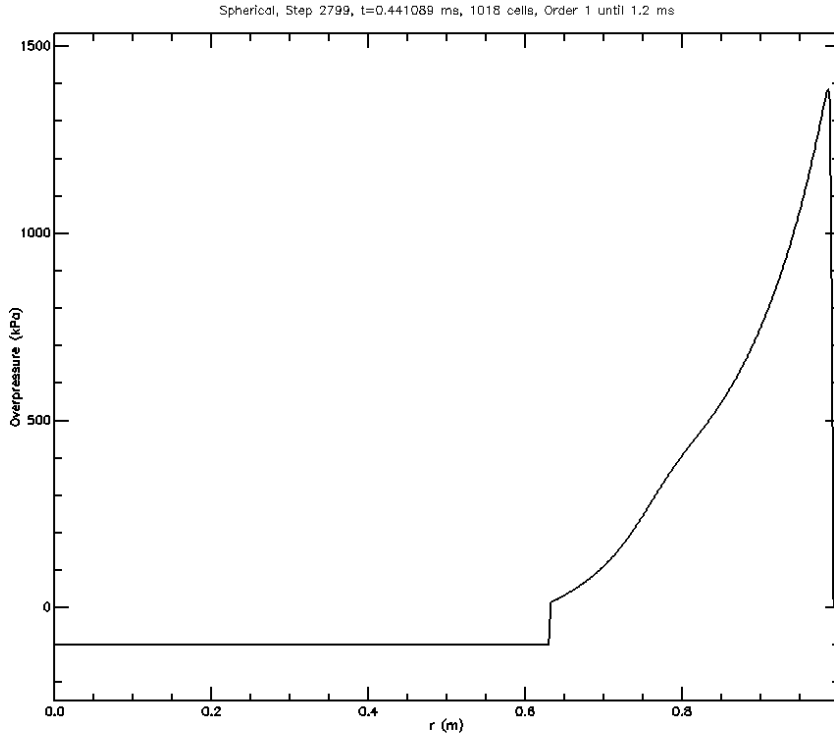


Figure 47: Overpressure versus radius after at the end of the spherically symmetric solution phase (about 0.44ms) for the Cylindrical Containment case of Section 3.3

Solution then switches to the cylindrical phase. There is an apparent pause in computation as ProSAir transfers the spherical symmetric solution onto the cylindrically symmetric grid given the initial plot of Figure 48.

Almost simultaneously the blast wave hits the cylinder wall and the vent at about 3.70ms as seen in Fig. 49. Thereafter a cycle of shock wave reflections and shock vents are seen.

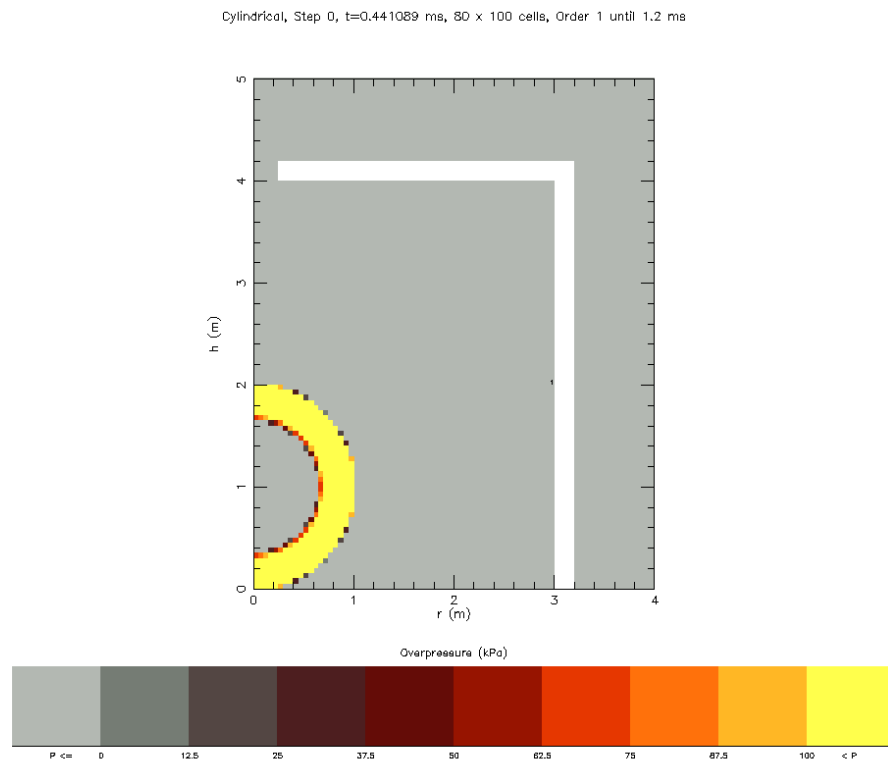


Figure 48: Overpressure contour plot at the start of the cylindrically symmetric solution phase (about 1.9ms) for the Height of Burst case of Section 3.3

Cylindrical, Step 180, t=3.70449 ms, 80 x 100 cells, Order 2 since 1.2 ms

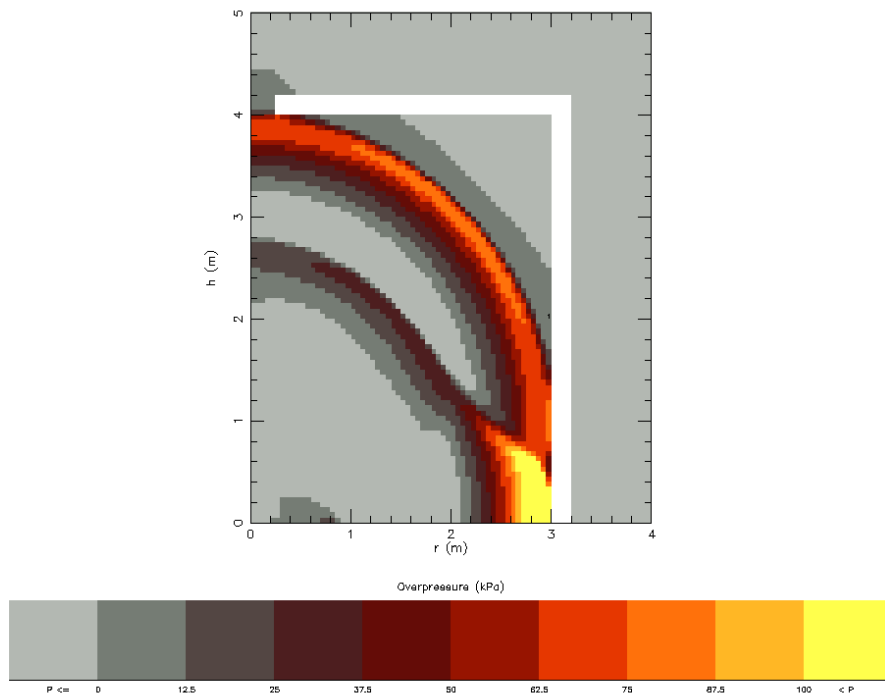


Figure 49: Overpressure contour plot at about 3.70ms for the Height of Burst case of Section 3.3

### 3.3.6 Post-processing the ProSAir output files

During the course of simulation a single target point file `pt_cylindrical_1.csv` is produced corresponding to the single target point. The overpressure and temperature are plotted for this point, using the data from `pt_cylindrical_1.csv`, in Figure 50.

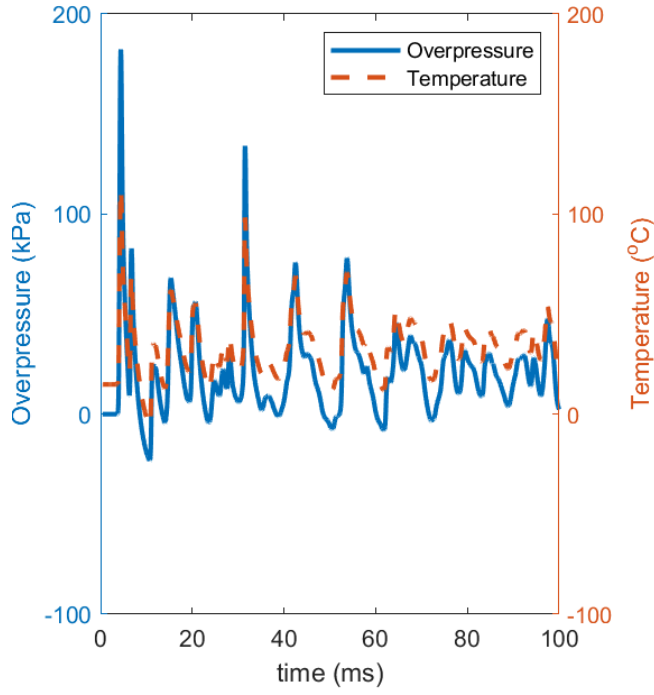


Figure 50: Overpressure and temperature for the single target point at  $(r, h) = (3, 2)$ m for the Cylindrical Containment case of Section 3.3

### 3.4 Blast in a Generic Cityscape

This example corresponds to that studied, both experimentally and numerically, by Brittle [?] and Smith et al [?]. The experiment, with *buildings* cast from unreinforced concrete is shown in Fig. 51.



Figure 51: Experimental configuration for the Cityscape example

#### 3.4.1 Problem Definition

Annex A of Brittle [?] defines the geometric configuration for this problem and is represented in plan view in Fig. 52.

The height and  $(x, y)$  coordinates of the buildings of Fig. 52 are given in Table 2.

| Building   | Base shape    | height (m) | $(x, y)$ coordinates (m)                               |
|------------|---------------|------------|--------------------------------------------------------|
| 1          | triangle      | 0.4        | $(0, 0), (0.3, 0.6), (0, 0.6)$                         |
| 2          | rectangle     | 0.4        | $(0, 0.9), (0.3, 0.9), (0.3, 1.3), (0, 1.3)$           |
| 3          | quadrilateral | 0.3        | $(0.47, 0.9), (0.533, 0.9), (0.77, 1.3), (0.47, 1.3)$  |
| 4          | triangle      | 0.4        | $(0.656, 0.5), (1.13, 0.5), (1.13, 1.3)$               |
| 5          | quadrilateral | 0.3        | $(0.36, 0), (0.96, 0), (0.96, 0.2), (0.478, 0.2)$      |
| left wall  | rectangle     | 0.45       | $(-0.45, 0.1), (-0.3, 0.1), (-0.3, 1.2), (-0.45, 1.2)$ |
| right wall | rectangle     | 0.45       | $(1.26, 0.1), (1.41, 0.1), (1.41, 1.2), (1.26, 1.2)$   |

Table 2: Building coordinates for the Cityscape example

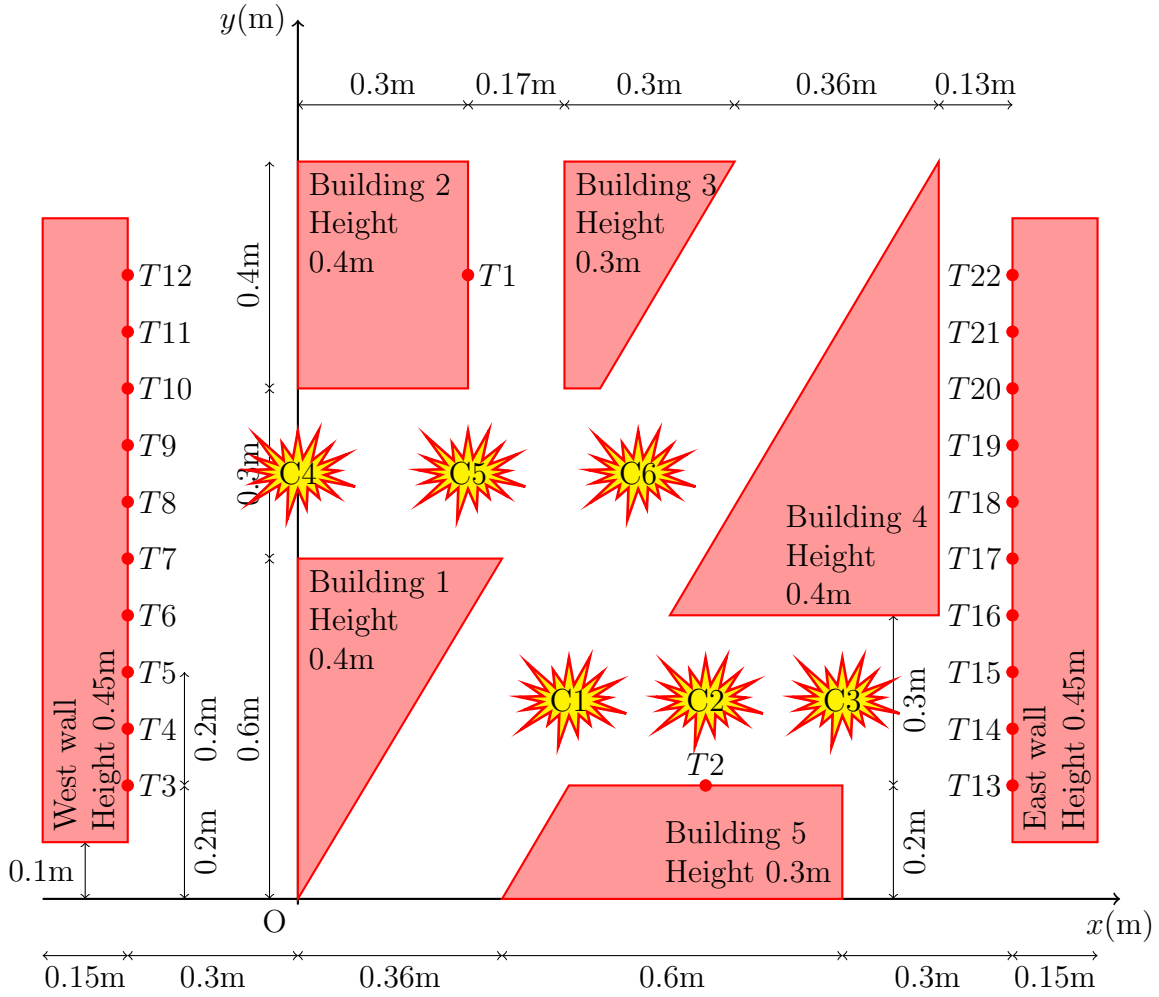


Figure 52: Plan view schematic of the Cityscape example

In Brittle's experiments pressure transducers could be placed at the locations  $T1$  to  $T22$  of Fig. 52. In practice only the odd numbered locations were used on the west and east walls and only 4 transducers were used in each experiment. Used transducer coordinates are given in Table 3.

Six possible charge locations, labelled  $C1$  to  $C6$  are depicted in Fig. 52 with coordinates given in Table 4. The  $x$ -coordinates for charge locations  $C1$  to  $C3$  were chosen so that the charge was exactly opposite the two ends and centre of the north wall of Building 5. In all cases the spherical charge was equivalent to 16g TNT (1.2g TNT equivalent detonator and 14.8g bare TNT).

| Target | coordinates (m)     |
|--------|---------------------|
| $T1$   | (0.3, 1.1, 0.105)   |
| $T2$   | (0.719, 0.2, 0.105) |
| $T3$   | (−0.3, 0.2, 0.075)  |
| $T5$   | (−0.3, 0.4, 0.075)  |
| $T9$   | (−0.3, 0.8, 0.075)  |
| $T11$  | (−0.3, 1.0, 0.075)  |
| $T13$  | (1.26, 0.2, 0.075)  |
| $T15$  | (1.26, 0.4, 0.075)  |
| $T17$  | (1.26, 0.6, 0.075)  |
| $T19$  | (1.26, 0.8, 0.075)  |
| $T21$  | (1.26, 1.0, 0.075)  |

Table 3: Target point coordinates for the Cityscape example

| Charge | coordinates (m)     |
|--------|---------------------|
| 1      | (0.478, 0.35, 0.04) |
| 2      | (0.719, 0.35, 0.04) |
| 3      | (0.96, 0.35, 0.04)  |
| 4      | (0.6, 0.75, 0.04)   |
| 5      | (0.3, 0.75, 0.04)   |
| 6      | (0.0, 0.75, 0.04)   |

Table 4: Charge coordinates for the Cityscape example

### 3.4.2 Using Charge Location $C1$

For the purposes of this user guide we shall just consider charge location  $C1$ . From Fig. 52 and Table 4 we see that the charge is detonated 0.04m above the ground and 0.15m due north of the nearest obstacle (Building 5) as depicted in the cross-section of Fig. 53. The

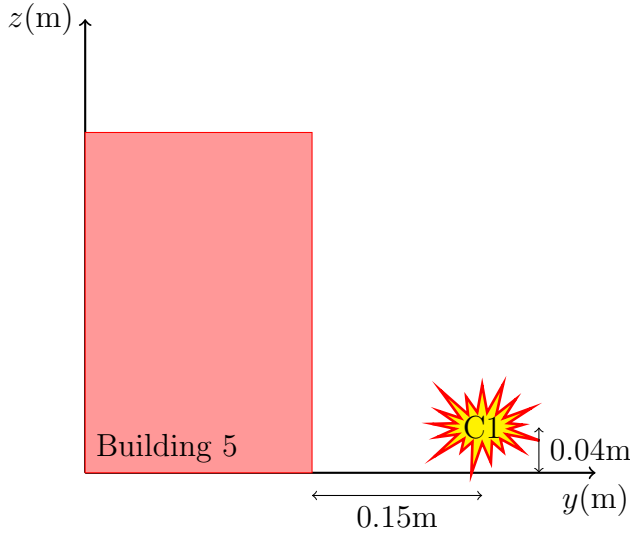


Figure 53: Vertical cross-section at  $x = 0.478\text{m}$  of configuration for charge  $C1$

most effective use of ProSAir will therefore involve a three stage calculation:

1. Spherically symmetric solution from detonation until the explosive shock approaches the ground after travelling 0.04m.
2. Cylindrically symmetric solution from the shock reaching the ground until it travels a further 0.15m horizontally and reaches Building 5.
3. 3D Simulation from then onwards.

### 3.4.3 ProSAir General Settings

On opening ProSAir and defining a **File** -> **New simulation** and opening the **Simulation** -> **Settings** we may set the **General settings** tab to be as that in Fig. 54. Here, in con-

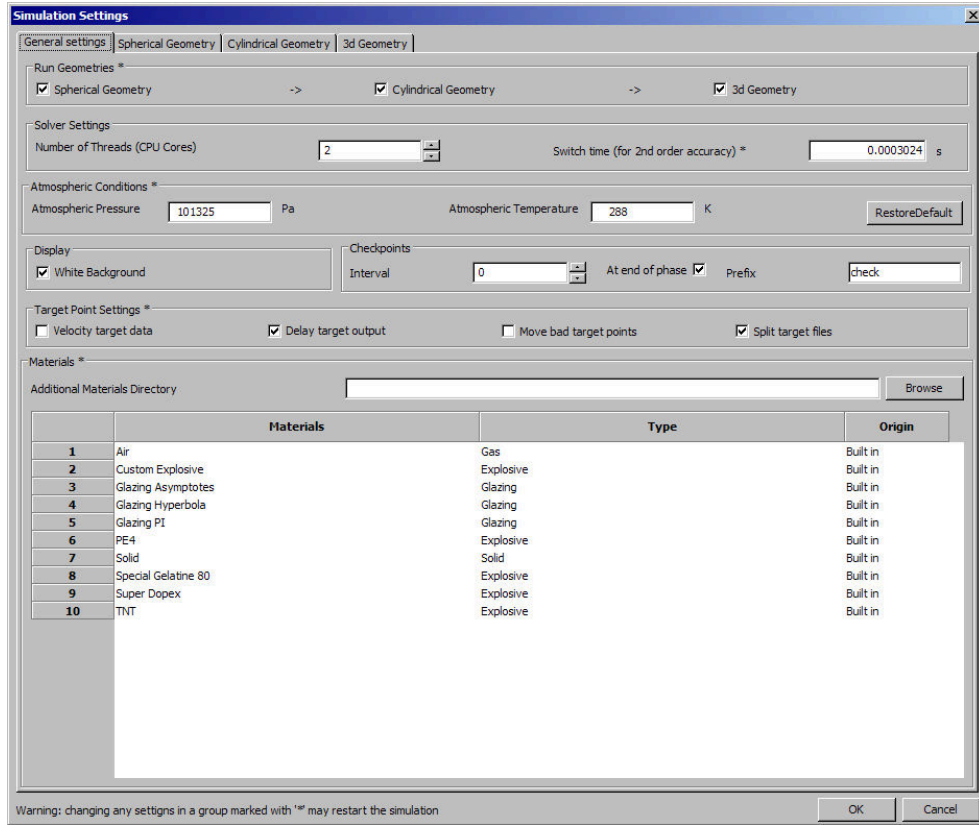


Figure 54: **General settings** tab for the Cityscape example

trast to the examples of Sec. 3.1 and 3.2, we ensure that within the **Run Geometries** panel all three of the **Spherical Geometry**, **Cylindrical Geometry** and **3d Geometry** boxes have been selected. Additionally we have set the **Switch time** of the **Solver Settings** panel to  $\sqrt[3]{0.016} \times 0.0012 = 3.024 \times 10^{-4}$  s to account for the 0.016kgTNT charge mass. All other settings are the defaults.

### 3.4.4 Spherical Geometry Settings

On clicking the **Spherical Geometry** tab in the **Simulation Settings** window of Fig. 54 the **Spherical Geometry** tab appears and may be modified to that of Fig. 55. Here:

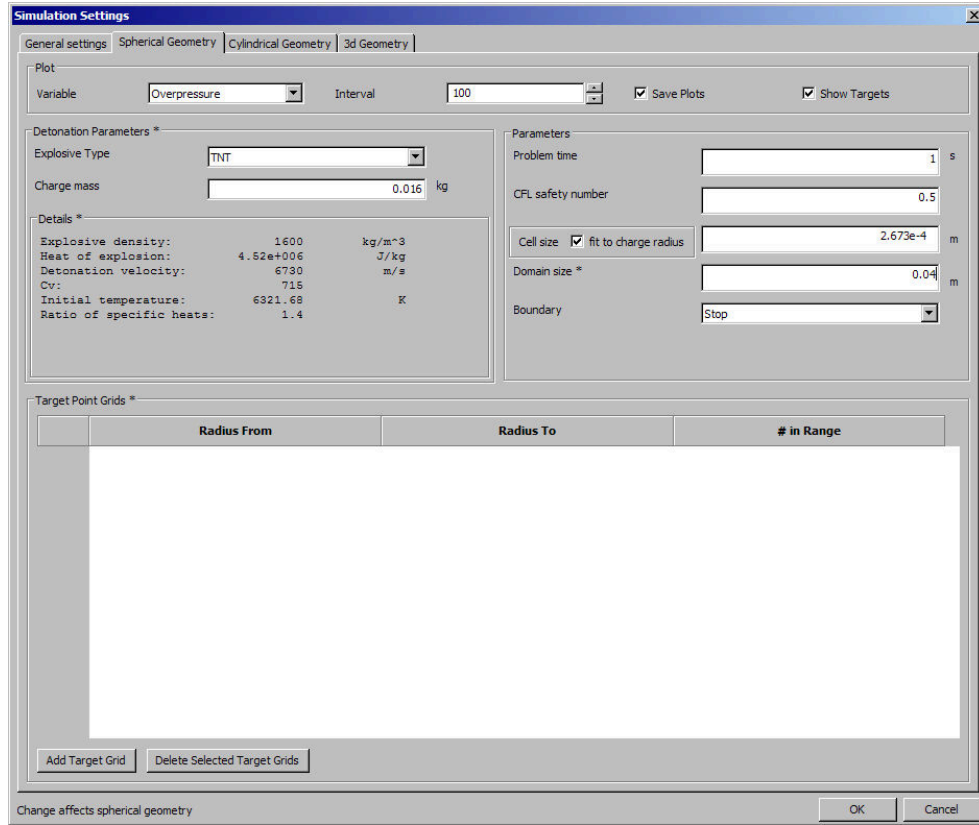


Figure 55: **Spherical Geometry** tab for the Cityscape example

1. We have chosen the **Explosive type** to be **TNT**.
2. A quick calculation shows the charge radius to be,

$$r_c = \sqrt[3]{\frac{3 \times 0.016}{4\pi 1600}} = 0.013365\text{m},$$

so choosing a cell size of  $r_c/50 = 2.673 \times 10^{-4}$  gives 50 cells across the charge which should yield sufficient accuracy for our purposes. We also choose to **fit** the cell size to the charge radius so 50 cells exactly cover the charge radius irrespective of small computer round-off errors.

3. The **Domain size** is chosen to be 0.04m, the height of burst of the charge (c.f., Fig. 53).

4. We choose a **Stop Boundary** so execution will pause as the blast wave approaches the outer boundary at 0.04m and the solution will be transferred to the next, cylindrical, stage.

### 3.4.5 Cylindrical Geometry Settings

On clicking the **Cylindrical Geometry** tab in the **Simulation Settings** window of Fig. 54 the **Cylindrical Geometry** tab appears and may be modified to that of Fig. 56. Here:

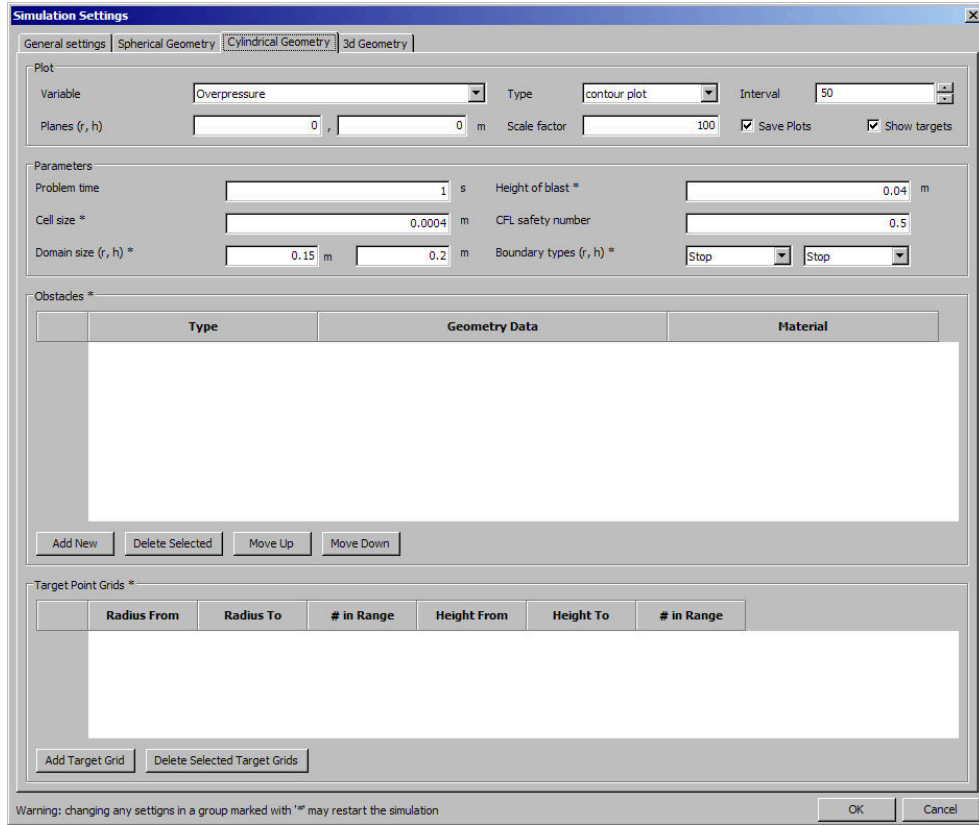


Figure 56: **Cylindrical Geometry** tab for the Cityscape example

1. We have set the Plot **scale factor** to be 100 so for Overpressure contours from 0 to 10,000 will be drawn (See 4.3.1 for details.).
2. We have set the **Height of blast** to be 0.04m in accordance with Fig. 53.
3. We have set the **Cell size** to be 0.0004m i.e., one hundredth of the height of blast to ensure good resolution of the shock-ground interaction.
4. The **Domain size** has been set to 0.15m in the radial direction to allow the shock to propagate as far as Building 5 (c.f., Fig. 53). In the vertical direction we have chose the domain size to be 0.2m so that the shock can propagate 0.16m (the domain size minus the height of blast) which, as it is less than the radial size should ensure the shock reaches the radial boundary before the vertical one.

5. We set both **Boundary types** to be **Stop** to ensure the simulation pauses, and transfers to the 3D simulation, just before the blast wave reaches either of the cylindrical outer boundaries.

### 3.4.6 3d Geometry Settings

On clicking the **3d Geometry** tab in the **Simulation Settings** window of Fig. 54 the **3d Geometry** tab appears and may be modified to that of Fig. 57. Here,

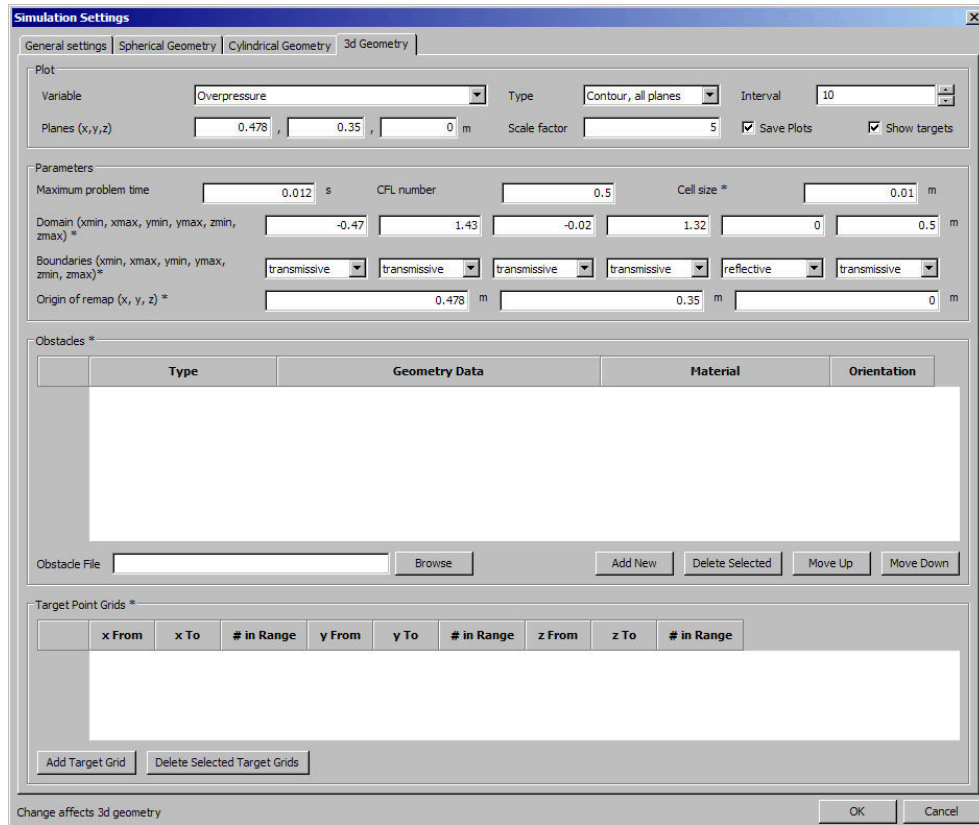


Figure 57: Initial **3d Geometry** tab for the Cityscape example

1. Within the **plot** panel we have chosen to **Contour, all planes** with coordinates **Planes (x,y,z)** set to  $x = 0.478$ ,  $y = 0.35$ , and  $z = 0$ . Consequently, ProSAir will produce vertical slices through the domain in the  $y$  and  $x$  directions aligned with the detonation point and also a plot of conditions on the ground ( $z = 0$ ). We have also specified a **scale factor** of 5 so that overpressures from 0KPa to 500KPa will be displayed (See 4.4.1 for details).
2. Within the **Parameters** pane we make several settings:
  - (a) We set the **Maximum problem time** to 0.012s so the problem will be simulated for 12ms of simulation time.
  - (b) The **Cell size** is 0.01m - this might be considered small as there are just 15 cells between the detonation site and Building 5.

- (c) The **Domain** has been set to surround the buildings with a margin of 0.02m (i.e., two cells) in the horizontal and set to 0.5m in the vertical.
- (d) The **Boundaries** are all **Transmissive** to allow the blast wave to leave the domain with the exception of that on the ground (**zmin**,  $z = 0$ ) which is a reflecting surface.
- (e) The **Origin of remap** specifies where the origin of the previous solution phase, in this case the cylindrical phase, sits in the 3D domain. Clearly here it is on the ground, directly under the detonation site.

The ProSAir input file `Cityscape0.pr1` contains the problem specification up to this stage of definition.

### 3.4.7 Adding Obstacles

We must now add the buildings, specified in Sec. 3.4.1, into our ProSAir simulation. Currently ProSAir uses a crude constructive solid geometry technique to build up a 3D geometry by adding primitive obstacles which may be **Solid** or **Air** as detailed in Sec. 4.4.3. Looking at our required buildings in Figs. 51 and 52 it is clear that we may build up our configuration using **Solid cuboids** and **wedges** (a wedge is a 3D shape with a triangular base extruded in one of the coordinate directions) as indicated in Fig. 58.

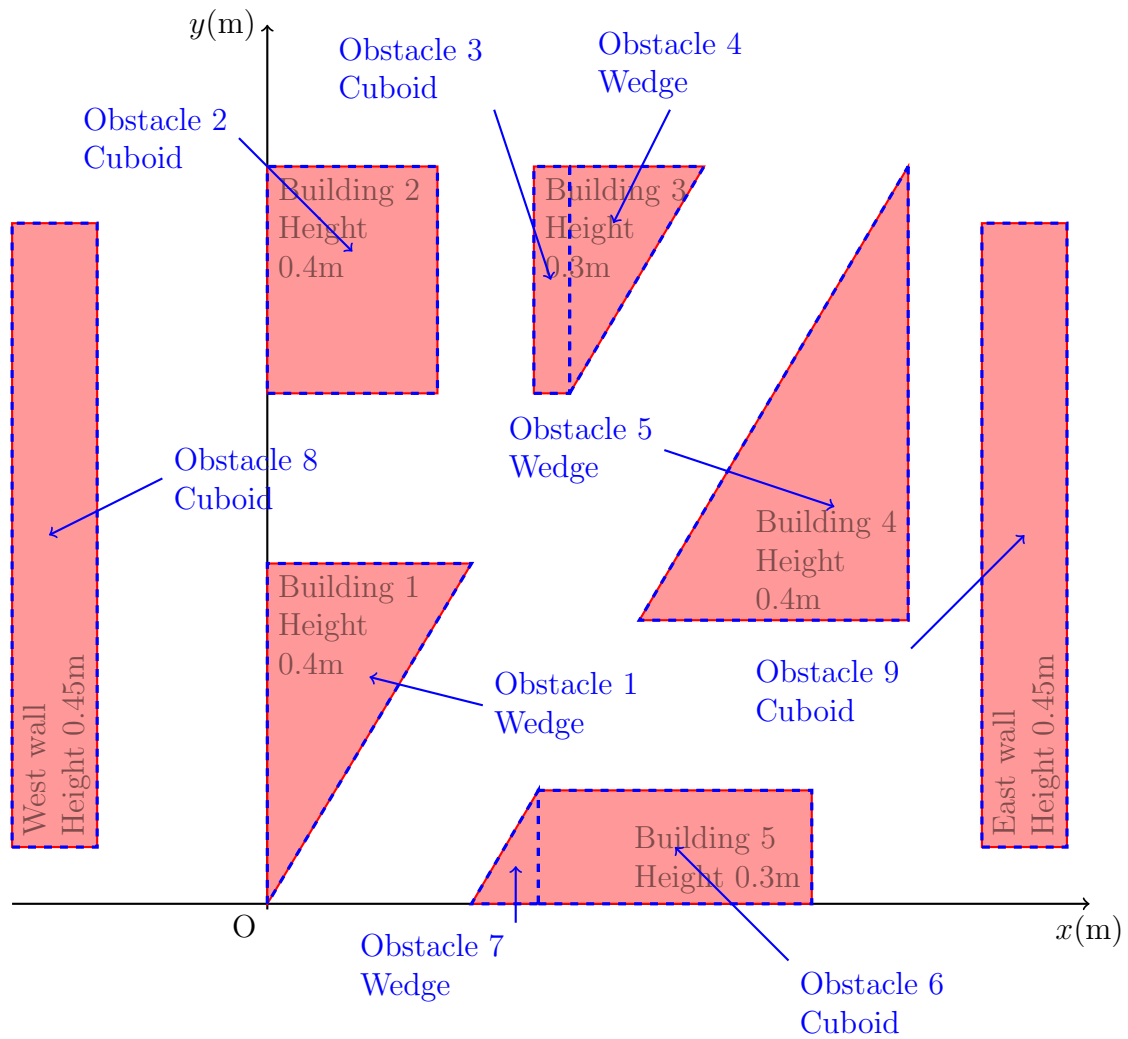


Figure 58: Plan view schematic of the obstacles used to make up the Cityscape

## Adding the first obstacle

Let's add the first obstacle, the wedge that will make up Building 1.

1. Click on the **Add New** button of the **Obstacles** panel of the **3d Geometry** tab of the **Simulation Settings** as seen in Fig. 57. A new default cuboid obstacle 1 gets added to the list of obstacles in the **Obstacles** panel as seen in Fig. 59.

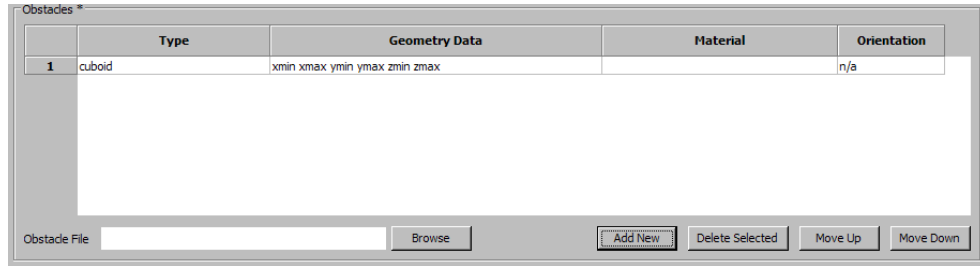


Figure 59: **Obstacles** panel of the **3d Geometry** tab after adding a new obstacle

2. Using the drop down menu associated with the **Type** of the newly created Obstacle 1 we select **wedge** as seen in Fig. 60. This modifies the default **Geometry Data** and

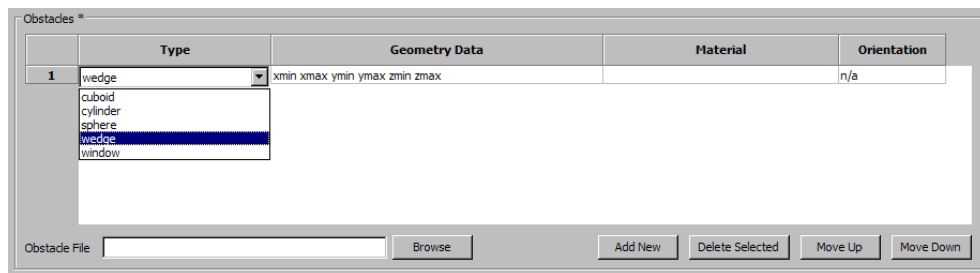


Figure 60: Choosing Obstacle 1's **Type** to be **wedge** in the **Obstacles** panel of the **3d Geometry** tab

**Orientation** to be as in Fig. 61

3. We now replace the place-holders **x0 y0 z0 x1 y1 z1 x2 y2 z2 height** with the coordinates of the triangular base of Building 1 and the height as found in Table 2 for Building 1 (c.f., Fig. 97). We also use the drop down menu for Obstacle 1's **Material** to select the material to be **Solid**. We leave the **Orientation** as **z** to indicate we are extruding the triangular base a distance **height** in the **z**-direction. The **Obstacles** panel now looks as in Fig. 62.

The ProSAir input file **Cityscape1.pr1** contains the problem specification up to this stage of definition.

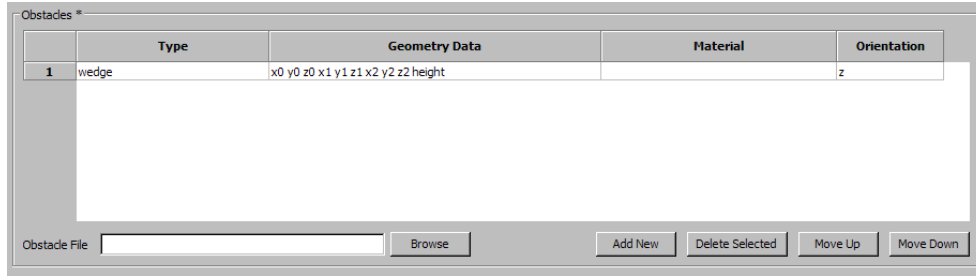


Figure 61: Default settings for a **wedge** for Obstacle 1

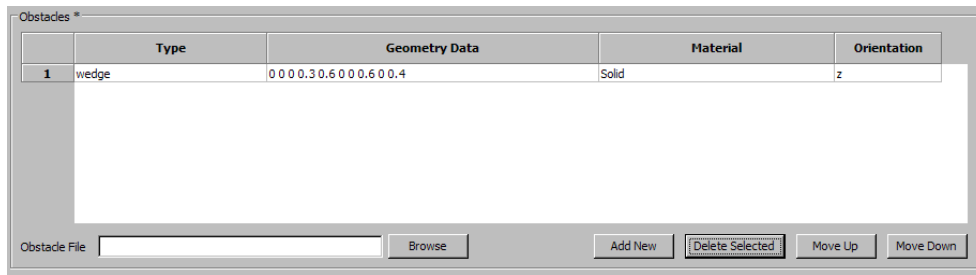


Figure 62: Completed settings for the **wedge** to form Obstacle 1 corresponding to Building 1

### Adding the remaining obstacles

In a similar manner to that used for Obstacle 1 we may add the remaining obstacles of Fig. 58 making use of the data of Table 2. The **Obstacles** panel now looks as in Fig. 63. The ProSAir input file **Cityscape2.pr1** contains the problem specification up to this stage

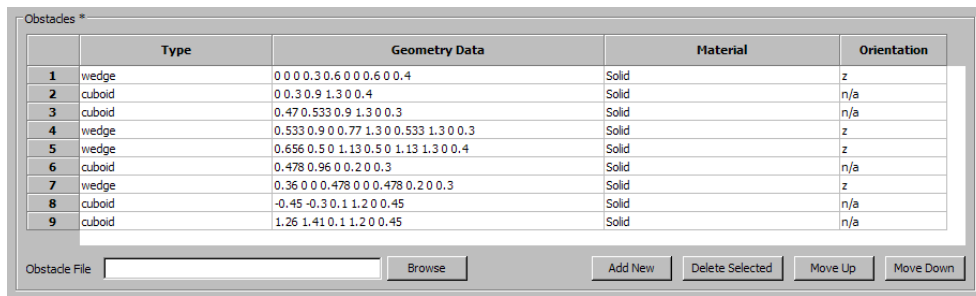


Figure 63: Completed **Obstacles** panel for the Cityscape problem of definition.

### 3.4.8 Adding Target Points

We must now add the target points, specified in Table 3, into our ProSAir simulation. We do this in the **Target point grids** panel of the **3d Geometry** tab. We click on the **Add Target Grid** button to create a new target point grid with default settings as in Fig. 64.

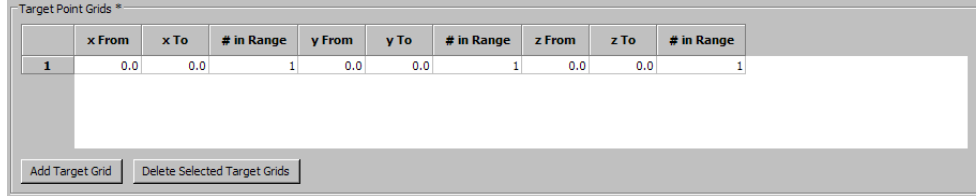


Figure 64: New **Target Point Grid** for the Cityscape problem

A target point grid lets us specify a grid of target points, evenly spaced in the  $x$ ,  $y$  and  $z$  directions. For target point  $T1$  we require just a single point at  $(x, y, z) = (0.3, 1.1, 0.105)$ . As seen in Fig. 52, this point is on the east wall of Building 2. If we specify the  $x$  coordinate as exactly 0.3m then the target point will fall on a mesh line and ProSAir may choose to place it in the solid mesh cell lying inside Building 2, or the air-filled mesh cell just outside Building 2. To avoid ambiguity we will specify the  $x$  coordinate as 0.301m, this will comfortably place the target point in an air-filled cell but the *artificial displacement* of 0.001m is less than half the specified **Cell size** and so the target point is placed in the air-filled cell abutting its true location. We thus complete the target point settings as in Fig. 65. In a similar manner we may add target point  $T2$  but artificially displaced to

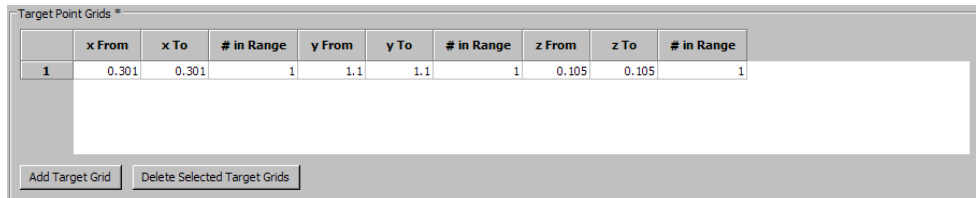


Figure 65: **Target Point Grid** specifying target point  $T1$  for the Cityscape problem

(0.719, 0.201, 0.105) in the  $y$ -direction as it lies on the northern face of Building 5.

Target points  $T3$  to  $T12$  are evenly spaced in the  $y$  direction so may be specified by a single target grid. We artificially displace in the  $x$  direction to ensure target cells do not lie in the West Wall. Consequently we choose:

- A single number of targets in the  $x$  direction located at  $x = -0.299$ m.
- 10 targets in the  $y$  direction ranging from  $y = 0.2$ m to  $1.1$ m.
- A single number of targets in the  $z$  direction located at  $z = 0.075$ m.

The completed target point grid, and that required to define targets  $T_{13}$ - $T_{22}$ , are seen in Fig. 66. The ProSAir input file `Cityscape3.pr1` contains the completed ProSAir input

|   | x From | x To   | # in Range | y From | y To  | # in Range | z From | z To  | # in Range |
|---|--------|--------|------------|--------|-------|------------|--------|-------|------------|
| 1 | 0.301  | 0.301  | 1          | 1.1    | 1.1   | 1          | 0.105  | 0.105 | 1          |
| 2 | 0.719  | 0.719  | 1          | 0.201  | 0.201 | 1          | 0.105  | 0.105 | 1          |
| 3 | -0.299 | -0.299 | 1          | 0.2    | 1.1   | 10         | 0.075  | 0.075 | 1          |
| 4 | 1.259  | 1.259  | 1          | 0.2    | 1.1   | 10         | 0.075  | 0.075 | 1          |

Figure 66: **Target Point Grid** specifying all target points for the Cityscape problem file.

### 3.4.9 Running the Simulation

On running the simulation the spherical solution phase starts and ProSAir displays a plot of overpressure with radius, evolving in time, until the shock approaches the outer boundary at radius  $R = 0.04\text{m}$  as seen in Figure 67. Note that the overpressure at the

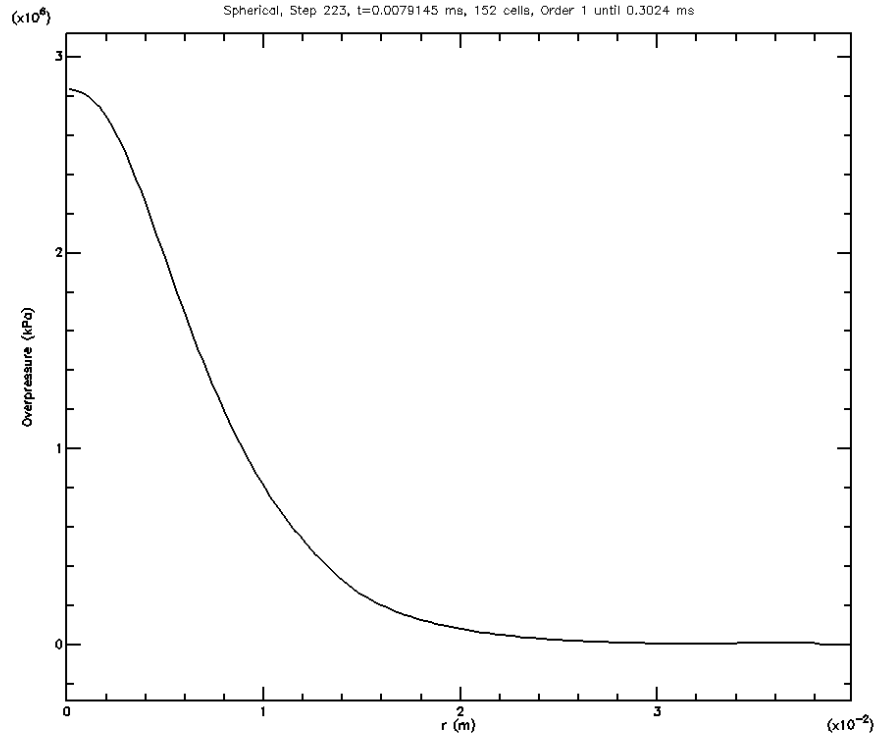


Figure 67: Overpressure versus radius after at the end of the spherically symmetric solution phase (about 0.008ms) for the Cityscape case of Section 3.4

shock ( $r \approx 0.0038\text{m}$ ) is small compared to that persisting at the centre of the detonation ( $r = 0\text{m}$ ).

Solution then switches to the cylindrical phase. There is an apparent pause in computation as ProSAir transfers the spherical symmetric solution onto the cylindrically symmetric grid given the initial plot of Figure 68. Solution then continues in the cylindrical phase until 0.043ms shown in Figure 69.

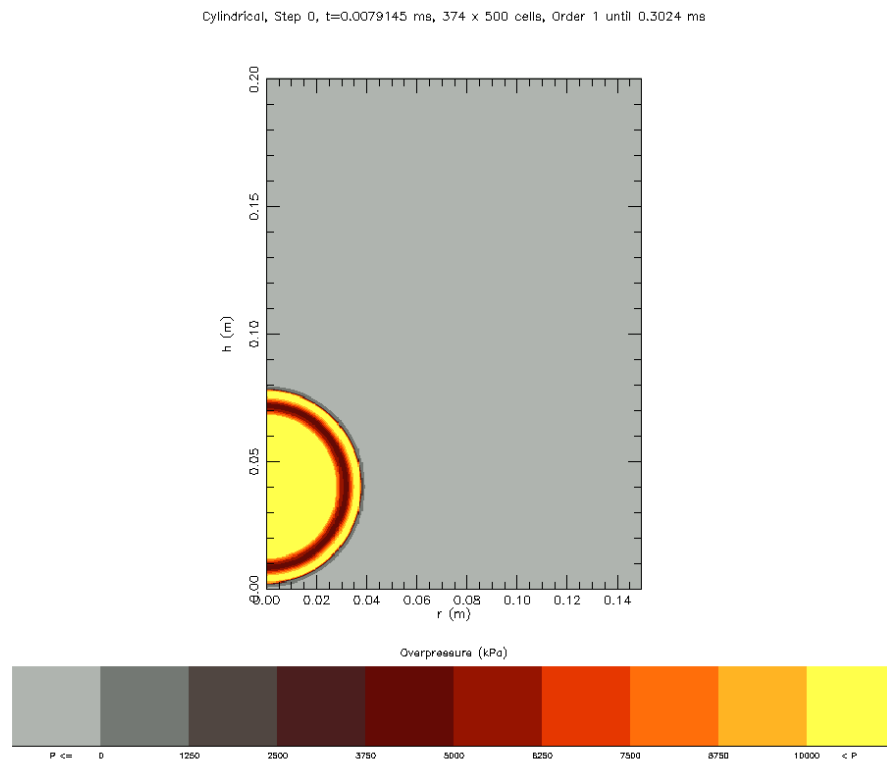


Figure 68: Overpressure contour plot at the start of the cylindrically symmetric solution phase (about 0.0079ms) for the Cityscape case of Section 3.4

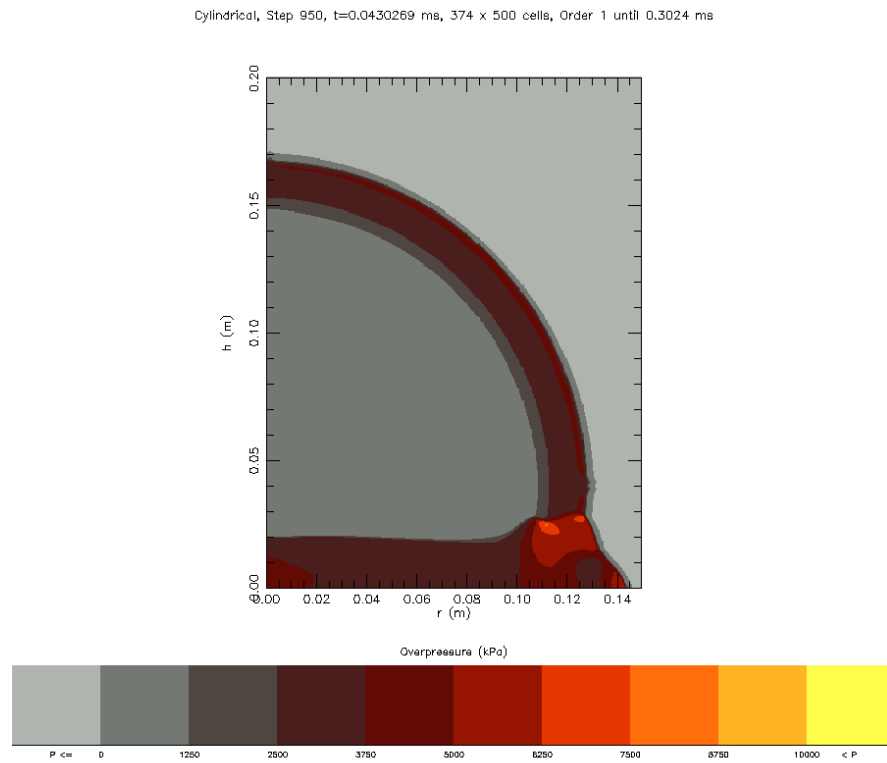


Figure 69: Overpressure contour plot in the cylindrically symmetric solution phase at about 0.043ms for the Cityscape case of Section 3.4

Solution then switches to the 3d phase. There is an apparent pause in computation as ProSAir transfers the cylindrically symmetric solution onto the 3d grid. The evolution of the explosion is seen in Figures 70-74.

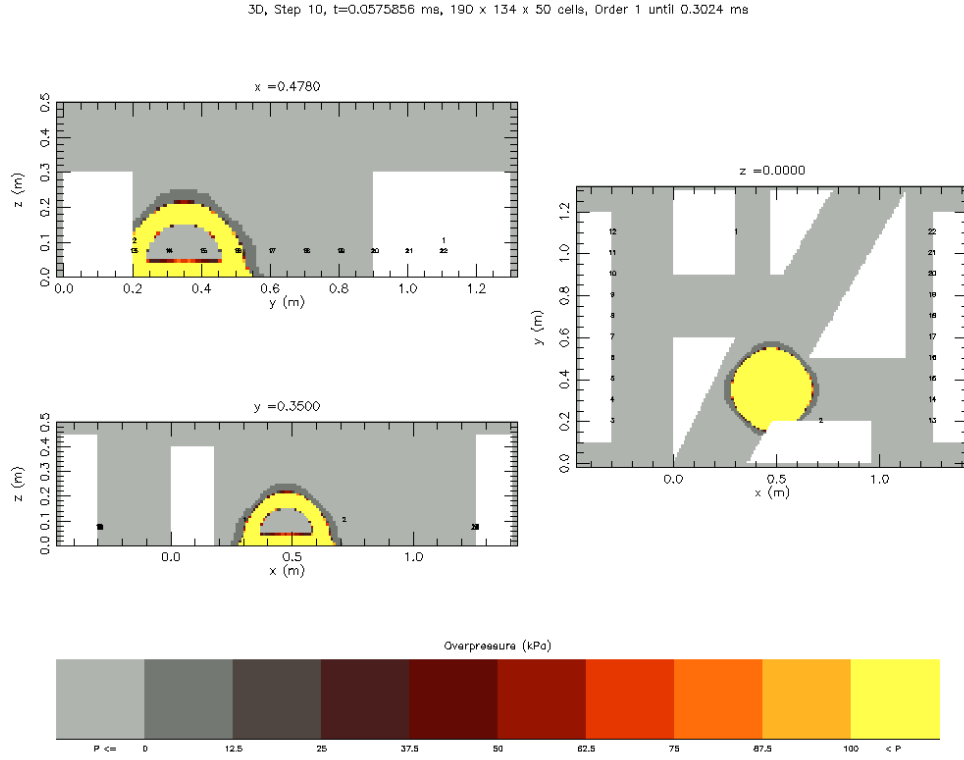


Figure 70: Overpressure contour plot at about 0.13ms for the Cityscape case of Section 3.4

The entire computation requires about 2 hours processing time using 2 cores on a Windows XP PC with Intel dual core 6300 (1.86 GHz) processor and 4GB of RAM.

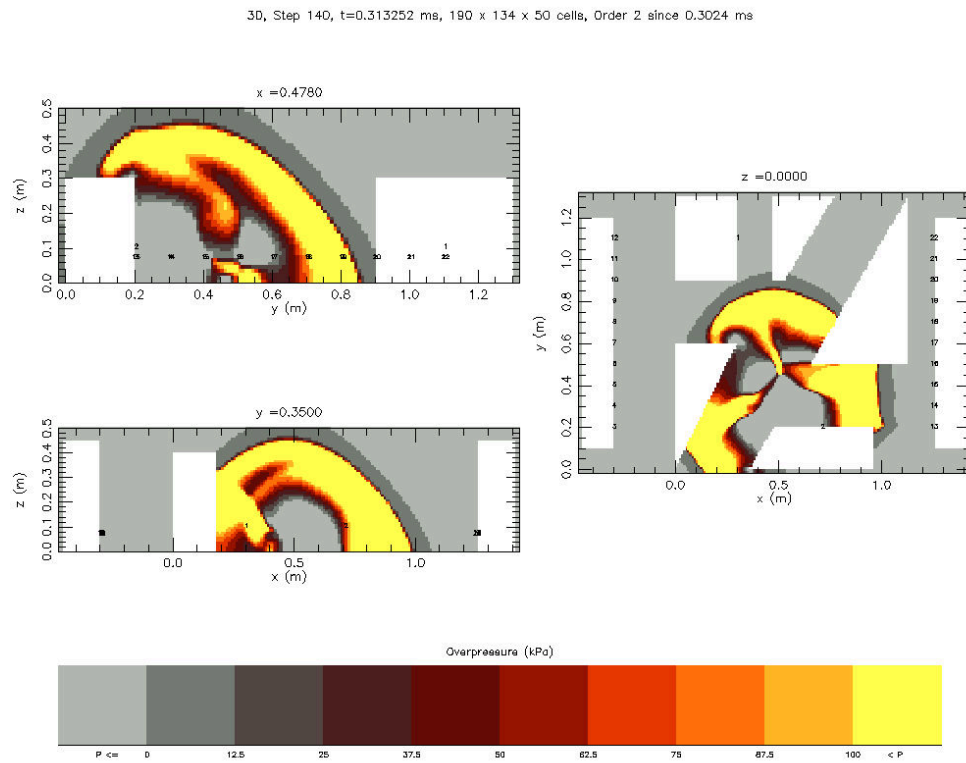


Figure 71: Overpressure contour plot in the 3d symmetric solution phase at about 0.269ms for the Cityscape case of Section 3.4

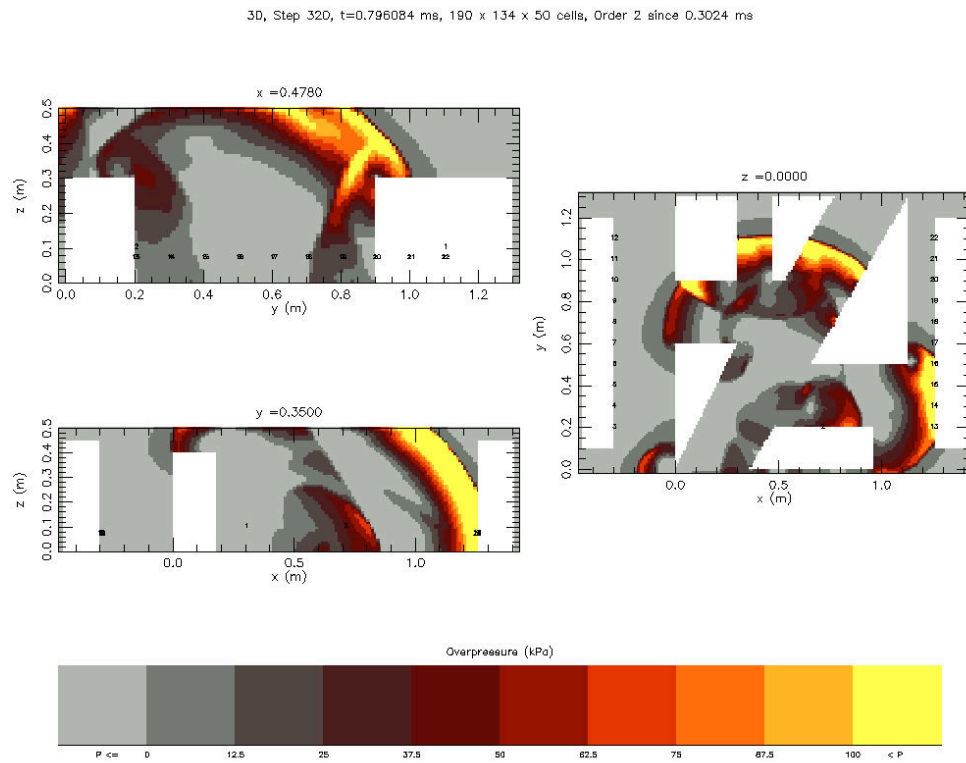


Figure 72: Overpressure contour plot in the 3d symmetric solution phase at about 0.762ms for the Cityscape case of Section 3.4

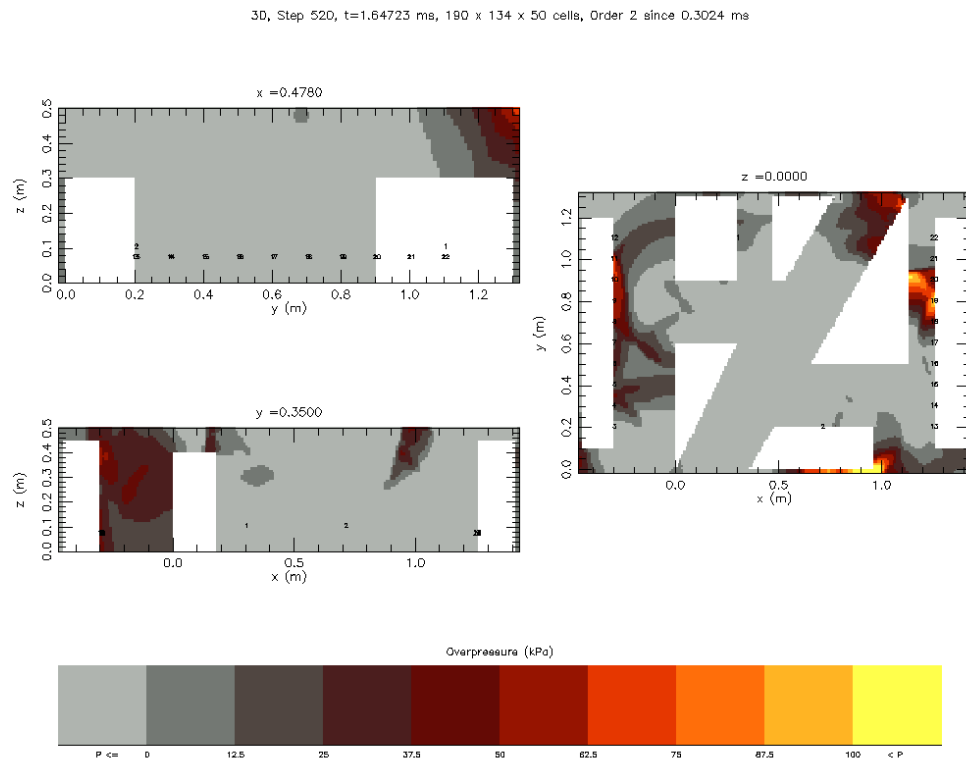


Figure 73: Overpressure contour plot in the 3d symmetric solution phase at about 1.63ms for the Cityscape case of Section 3.4

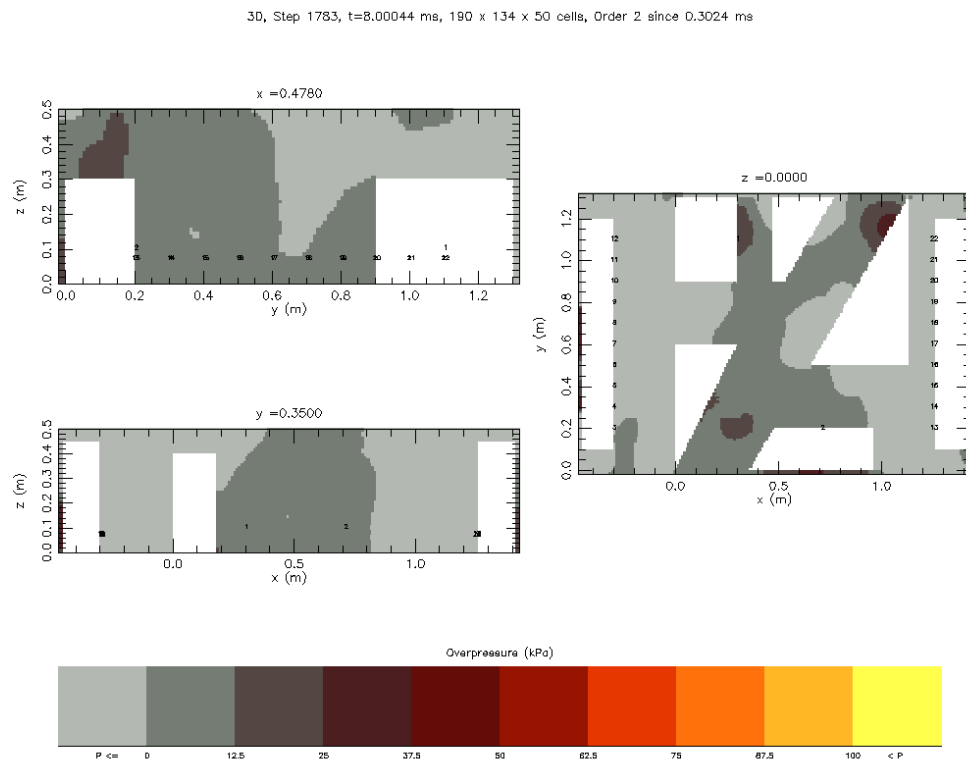


Figure 74: Overpressure contour plot in the 3d symmetric solution phase at the calculation end ( about 8.0ms) for the Cityscape case of Section 3.4

### 3.4.10 Post-processing the ProSAir output files

During the course of simulation a target summary file `target_summary_3d.csv` and 22 target point files `pt_3d_1.csv` - `pt_3d_22.csv` are produced. These contain information collected at the target points during the calculation. Brittle [?] produced experimental data for target points  $T1$ ,  $T3$ ,  $T11$  and  $T21$  available in files `L1T1.csv`, `L1T3.csv`, `L1T11.csv` and `L1T21.csv` respectively. Corresponding to these are the 4 ProSAir target point files `pt_3d_1.csv`, `pt_3d_3.csv`, `pt_3d_11.csv` and `pt_3d_21.csv`. In Figures 75 to 78 we compare the ProSAir calculated blast parameters with those obtained experimentally. For all

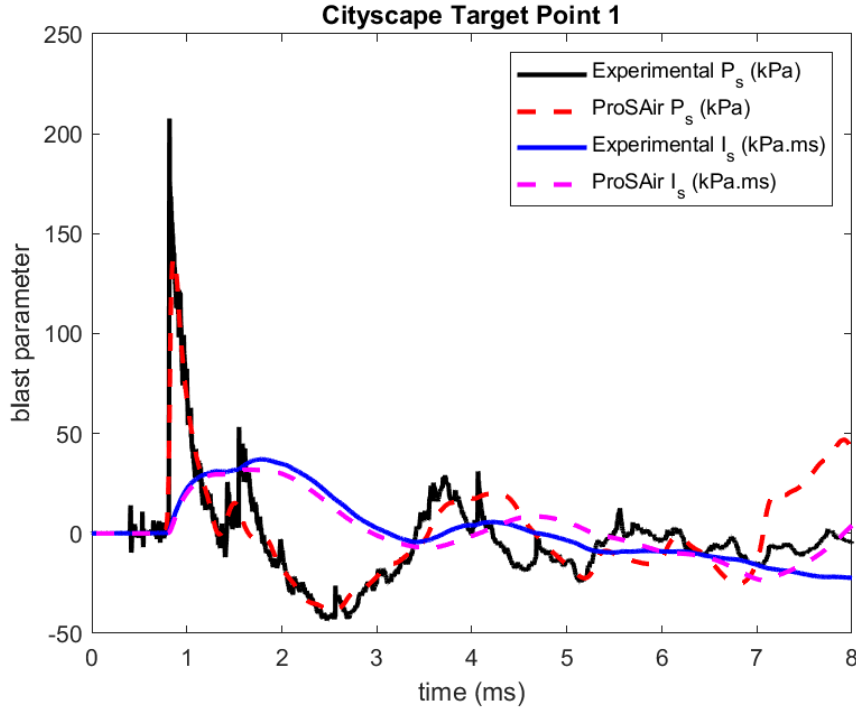


Figure 75: Overpressure and impulse versus time for target point  $T1$  for the Cityscape case of Section 3.4.10

except  $T21$  we see some underprediction of the peak pressure of the primary shock, however in all cases the first positive phase's duration and impulse is well captured albeit with some discrepancy in time of arrival. The degree of agreement for target  $T21$  is remarkable given the complex nature of the shock interactions associated with the shock passing along the corridor between Building 4 and the East Wall.

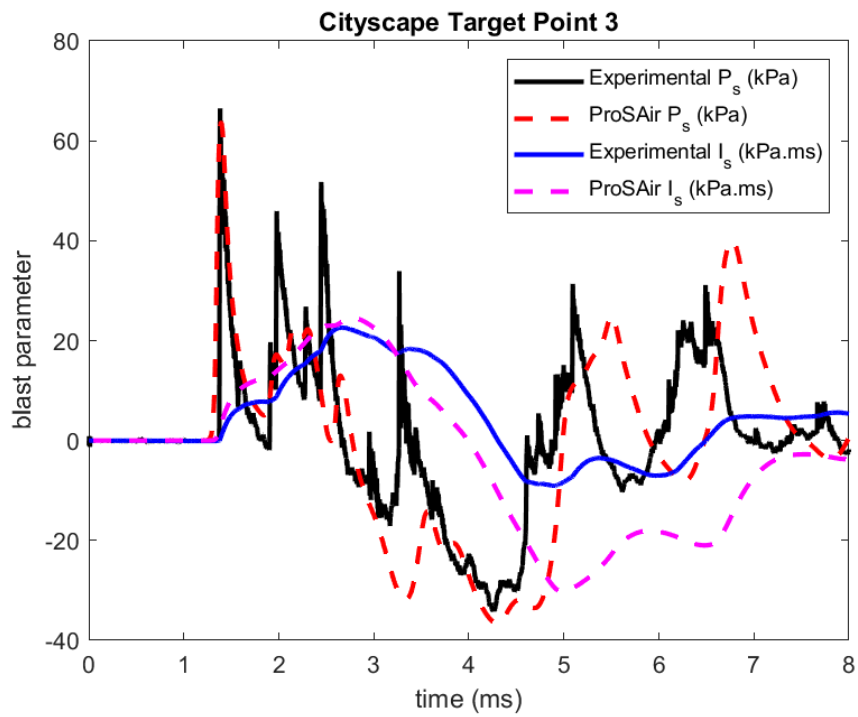


Figure 76: Overpressure and impulse versus time for target point  $T3$  for the Cityscape case of Section 3.4.10

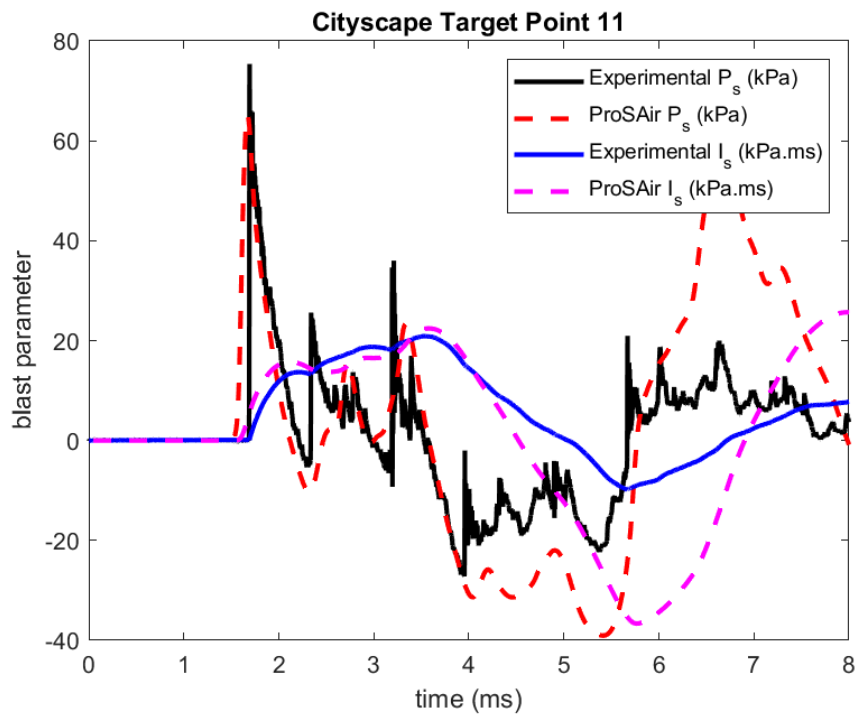


Figure 77: Overpressure and impulse versus time for target point  $T11$  for the Cityscape case of Section 3.4.10

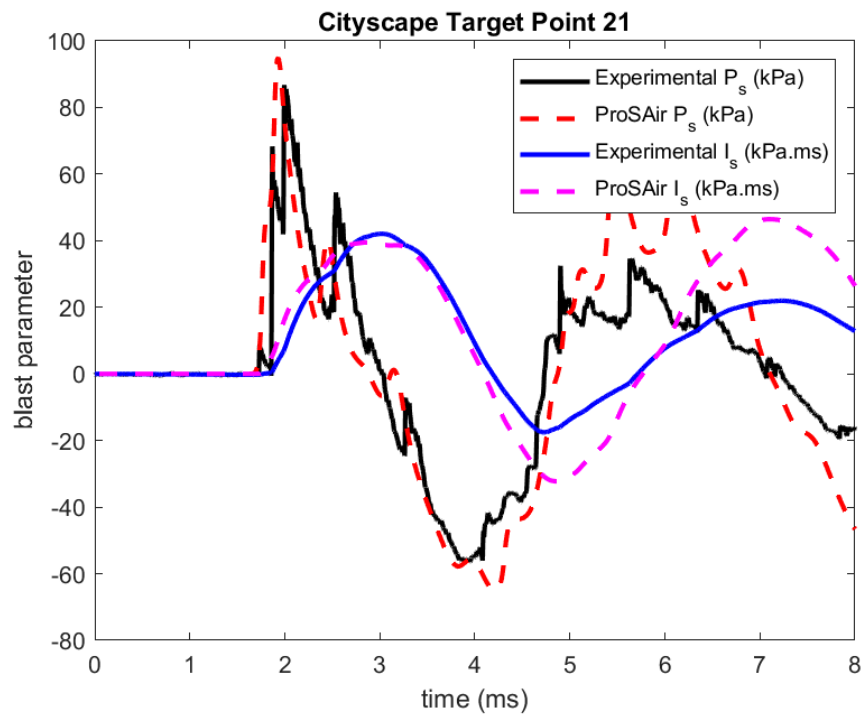


Figure 78: Overpressure and impulse versus time for target point  $T21$  for the Cityscape case of Section 3.4.10

### 3.4.11 Choosing the domain size

It is important to assess the sensitivity of the numerical solutions we have obtained in Section 3.4.9 and 3.4.10 to the numerical parameters we have set for our simulation. First we consider the effects of changing domain size.

Table 5 shows the domain sizes used for 3 calculations. Domain 1 corresponds to that

| Domain | Domain Extents |      |       |      |      |      |
|--------|----------------|------|-------|------|------|------|
|        | xmin           | xmax | ymin  | ymax | zmin | zmax |
| 1      | -0.47          | 1.43 | -0.02 | 1.32 | 0    | 0.5  |
| 2      | -0.82          | 1.78 | -0.37 | 1.67 | 0    | 0.8  |
| 3      | -0.92          | 1.88 | -0.47 | 1.77 | 0    | 0.9  |

Table 5: Domain sizes used for Cityscape domain sensitivity analysis

used in Section 3.4.9. We ran simulations for Domains 2 and 3 keeping all other settings identical to those used in Section 3.4.9. Overpressure history plots are shown in Figures 79 to 82. Domain 2 required 54 minutes run time on a dual quad core PC and Domain 3 required 69 minutes.

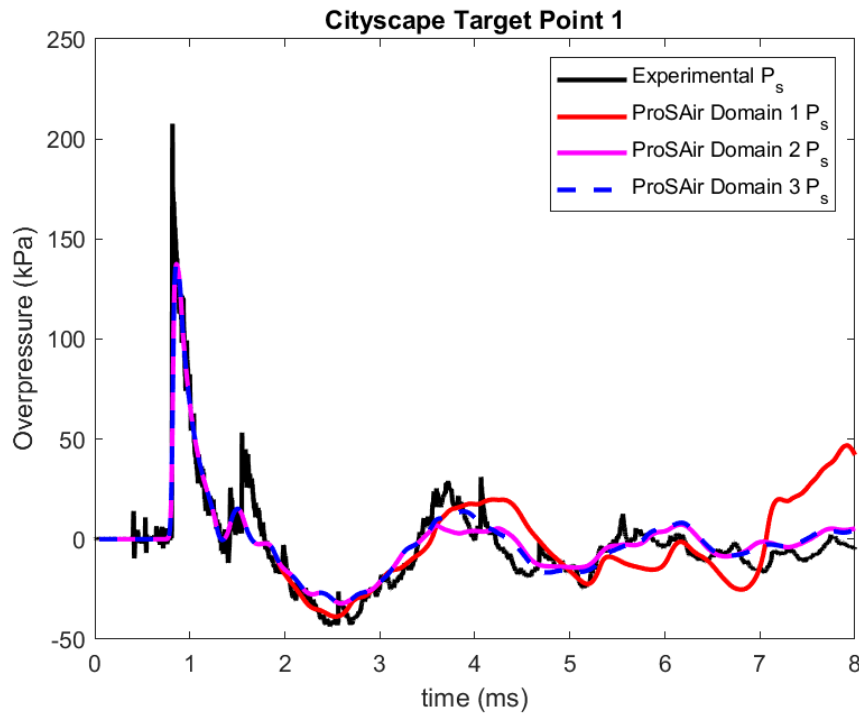


Figure 79: Overpressure history for target point  $T1$  on the three domains of Table 5

In Fig. 79 we see that on all three domains the overpressure histories are indistinguishable until about 2ms beyond which Domain 1's trace deviates from that of the others and so must plainly be in error. Overpressures for Domains 2 and 3 are seen to agree until about 3.6ms, thereafter the two data sets are separated by a small amount with the discrepancy at a maximum at about 3.9ms. This computation indicates that Domains 2 and 3 are plainly sufficient for prediction up to about 3.6ms - it is likely that Domain 3 is sufficient for the duration of the computation and errors using Domain 2 are tolerable.

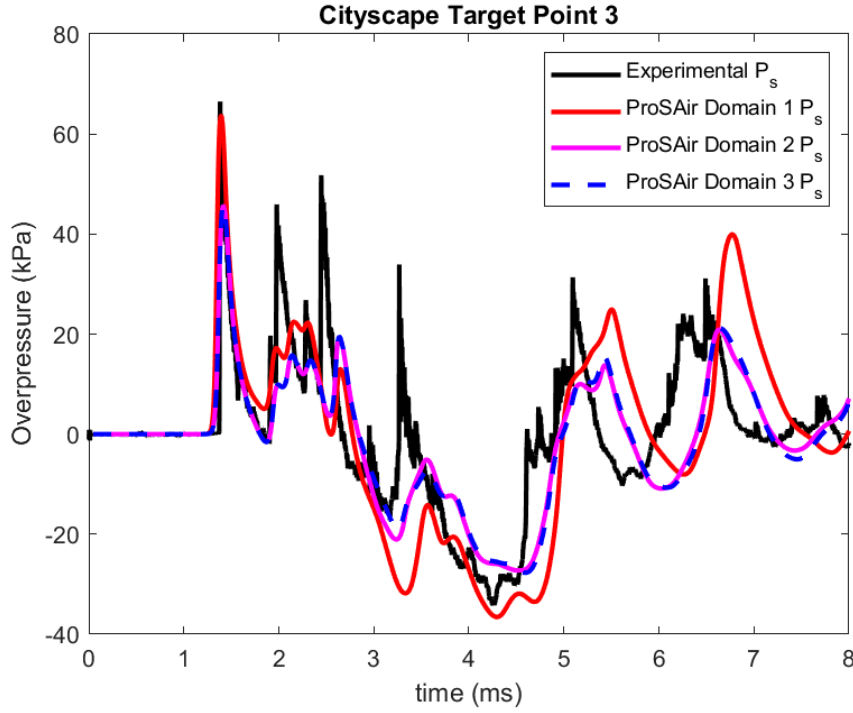


Figure 80: Overpressure history for target point  $T3$  on the three domains of Table 5

In Fig. 80 we see that for target  $T3$  on Domain 1 the shock peak pressure is higher than that obtained on Domains 2 and 3 and, although it is closely to the experimental trace, is poorly converged. We also see differences in the trace for Domain 1 compared to those for Domains 2 and 3 indicating that Domain 1 is poorly converged for target point  $T3$ . Domains 2 and 3 are in agreement up to about 2.8ms thereafter they differ slightly. This computation indicates that Domains 2 and 3 are sufficient for predictions throughout the period of integration.

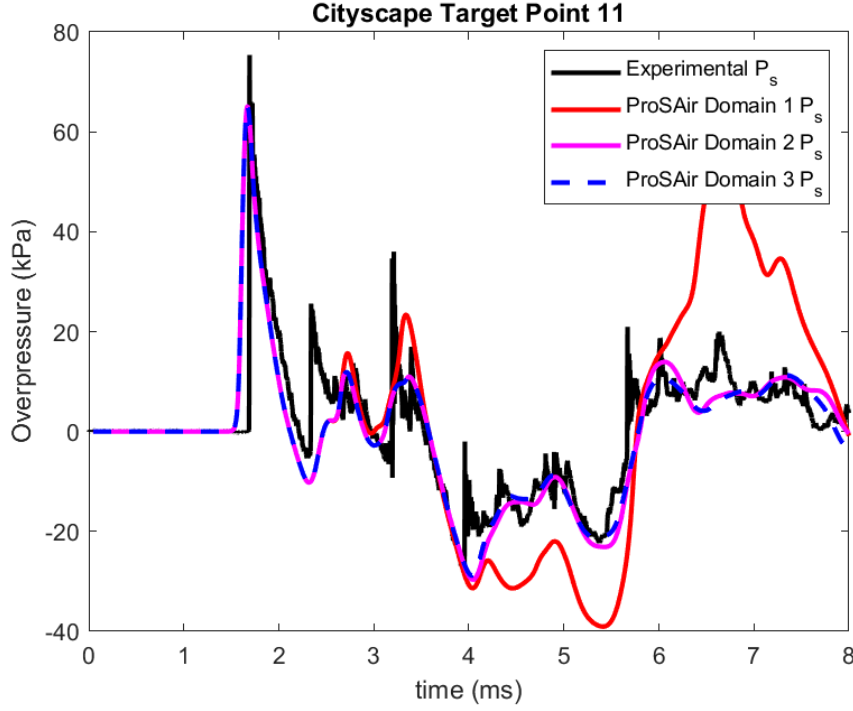


Figure 81: Overpressure history for target point  $T11$  on the three domains of Table 5

In Fig. 81 we see that for Target  $T11$  on all three domains the shock arrives earlier than the experiment. However Domains 1 to 3 are in agreement until about 2.7ms with qualitative agreement until 4.2ms. Domains 2 and 3 are in good agreement, and therefore suitable for predictive use, throughout the period of integration.

In Fig. 82 the experimental data indicates the arrival of shocks at target  $T21$  at 1.72ms, 1.86ms and 2.0ms - this fine structure is not quite matched in the simulations which appear to predict arrival of two shocks in this period. We note that Domain 1 maintains qualitative agreement with Domains 2 and 3 until about 4.7ms though with exaggeration peaks and troughs. Domains 2 and 3 are in good agreement up to about 6.3ms thereafter they deviate slightly. This computation indicates that Domains 2 and 3 are sufficient for prediction up to about 6.3ms - it is likely that Domain 3 is sufficient throughout the period of integration.

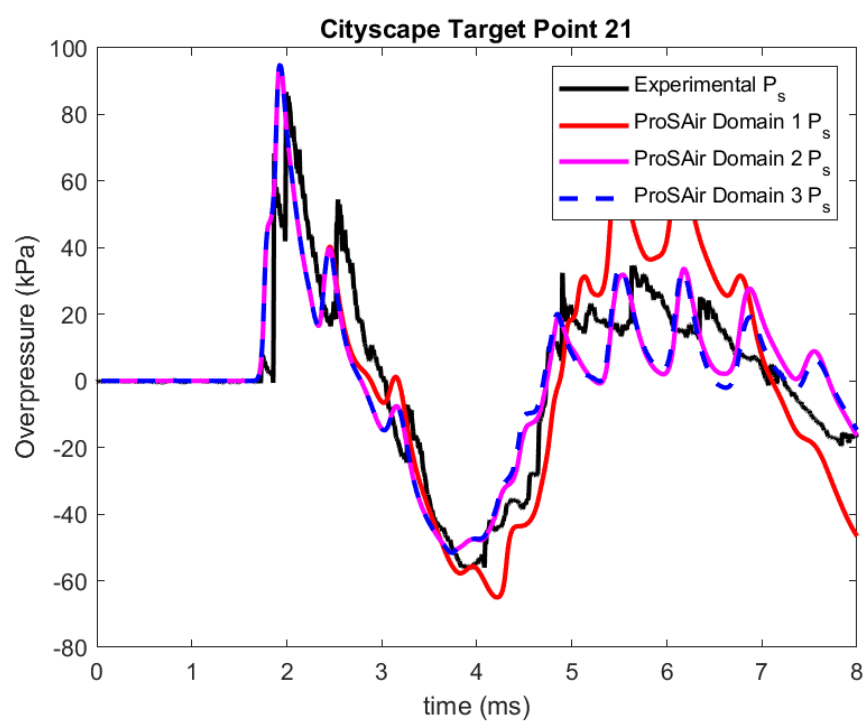


Figure 82: Overpressure history for target point  $T21$  on the three domains of Table 5

On examining the `target_summary_3d.csv` files for these 3 computations we see that the blast parameter information recorded for the target points are as in Tables 6 and 7.

| Target | Domain | $P_s$ (KPa) | $T_a$ (ms) | $I_s$ (KPa · ms) | $T_s$ (ms) |
|--------|--------|-------------|------------|------------------|------------|
| 1      | 1      | 137.486000  | 0.722558   | 29.562100        | 0.605108   |
|        | 2      | 137.486000  | 0.722558   | 29.562100        | 0.605113   |
|        | 3      | 137.486000  | 0.722558   | 29.562100        | 0.605113   |
| 3      | 1      | 63.617500   | 1.191170   | 22.992100        | 1.350880   |
|        | 2      | 45.602500   | 1.195460   | 8.481360         | 0.606481   |
|        | 3      | 45.602600   | 1.195460   | 8.481360         | 0.606481   |
| 11     | 1      | 65.004000   | 1.438190   | 15.723500        | 0.679714   |
|        | 2      | 65.087400   | 1.437920   | 15.725400        | 0.679979   |
|        | 3      | 65.087400   | 1.437920   | 15.725400        | 0.679979   |
| 21     | 1      | 94.765500   | 1.589420   | 39.490700        | 1.294680   |
|        | 2      | 94.835500   | 1.590530   | 38.482400        | 1.190500   |
|        | 3      | 94.835400   | 1.590530   | 38.482400        | 1.190500   |

Table 6: Blast parameters for first positive phase for the Cityscape example on the three domains of Table 5

From Table 6 for the first positive phase we see that results from targets 1, 11 and 21 are in excellent agreement on all 3 domains. For target point 3 the results on Domain 1 are not well predicted. In particular the phase impulse and phase duration are in significant error. Examination of Fig. 80 shows that on Domain 1 the small period of underpressure at about 1.8ms is not captured leading to an extended first positive phase. We presume the domain does not extend sufficient far in the  $-y$  direction to capture blast propagation round the south of building 1.

From Table 7 for the first negative phase we again see that results from Domains 2 and 3 are in excellent agreement. There are some substantial discrepancies for Targets  $T3$  and  $T21$  in the results for Domain 1.

### 3.4.12 Effect of using double precision

We repeated the simulation on Domain 3 using the double precision variant of ProSAir. the simulation required 78 minutes on a dual quad core PC. The results were graphically indistinguishable from those of for Domain 3 in Figures 79 to 82.

| Target | Domain | $P_s(\text{KPa})$ | $T_a(\text{ms})$ | $I_s(\text{KPa} \cdot \text{ms})$ | $T_s(\text{ms})$ |
|--------|--------|-------------------|------------------|-----------------------------------|------------------|
| 1      | 1      | -0.432188         | 1.327670         | -0.009106                         | 0.032374         |
|        | 2      | -0.431984         | 1.327670         | -0.009091                         | 0.032356         |
|        | 3      | -0.431977         | 1.327670         | -0.009091                         | 0.032356         |
| 3      | 1      | -0.090109         | 2.542050         | -0.000853                         | 0.015013         |
|        | 2      | -1.566770         | 1.801940         | -0.108827                         | 0.106941         |
|        | 3      | -1.566770         | 1.801940         | -0.108827                         | 0.106941         |
| 11     | 1      | -10.208900        | 2.117910         | -2.132310                         | 0.349584         |
|        | 2      | -10.210200        | 2.117900         | -2.132830                         | 0.349754         |
|        | 3      | -10.210300        | 2.117900         | -2.132830                         | 0.349754         |
| 21     | 1      | -6.544850         | 2.884110         | -0.911804                         | 0.229831         |
|        | 2      | -51.596800        | 2.781030         | -57.194000                        | 1.954610         |
|        | 3      | -51.589500        | 2.781030         | -55.330800                        | 1.920780         |

Table 7: Blast parameters for first negative phase for the Cityscape example on the three domains of Table 5

### 3.4.13 Effect of mesh refinement

In addition to the domain size and numerical precision, we must also assess the effect of mesh refinement. The computational method used by ProSAir will converge to the solution of the Euler equations of fluid motion with the given boundary and initial conditions as the mesh is refined. It therefore important to conduct a mesh refinement study to assess the effect of the finite size mesh cell size we have used.

Table 8 shows the cell sizes used for 3 calculations. The medium mesh corresponds to

| Mesh   | Cell Size (m) |
|--------|---------------|
| Coarse | 0.02          |
| Medium | 0.01          |
| Fine   | 0.005         |

Table 8: Mesh cell sizes used for Cityscape mesh sensitivity analysis

the computation on Domain 3 of Section 3.4.11. Keeping all other settings, apart from cell size, fixed we ran simulations for the Coarse and Fine Meshes. Overpressure history plots are shown in Figures 83 to 86. The Coarse mesh required 7 minutes run time on a dual quad core PC, the Medium 69 minutes and the Fine mesh required 360 minutes.

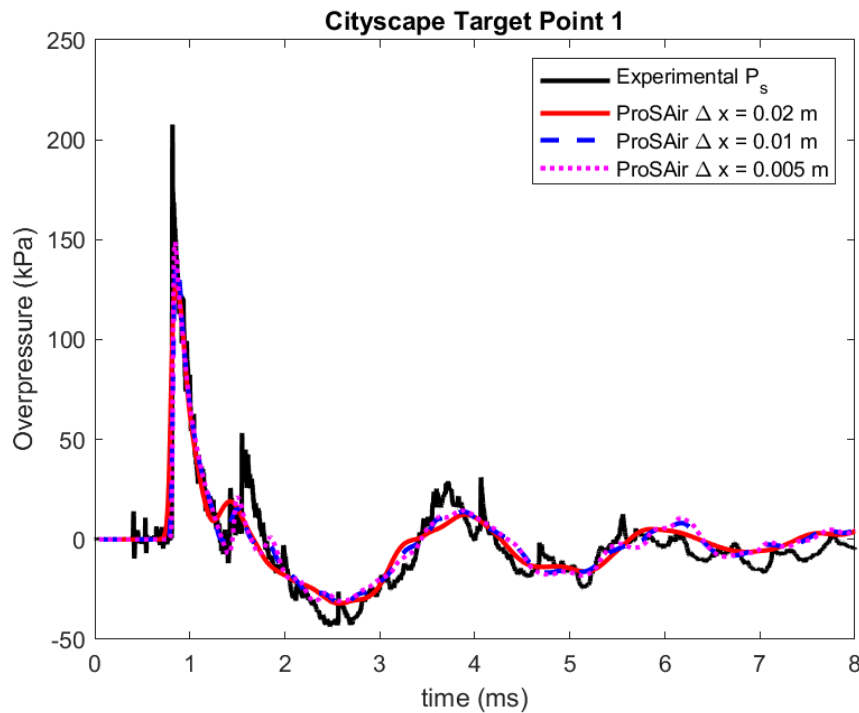


Figure 83: Overpressure history for target point  $T1$  on the three meshes of Table 8

In Fig. 83 we see that for target  $T1$  the shock arrival times are consistent for all three computations though the associated peak pressure increases on the finer meshes. The first small underpressure at 1.4ms is better resolved in the Medium and Fine computations. Otherwise the three computations are in good agreement up to, and beyond, the 3.6ms for which the domain size was assessed as being adequate in Section 3.4.11. From the experimental data we see a significant underprediction of the secondary shock at about 1.5ms.

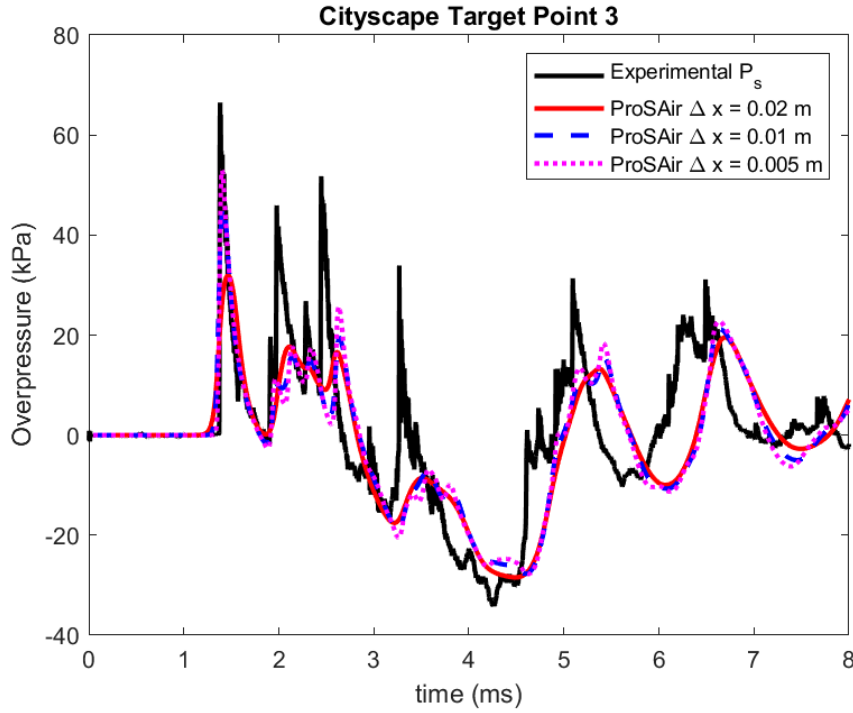


Figure 84: Overpressure history for target point  $T3$  on the three meshes of Table 8

In Fig. 84 we see that for target  $T3$  the three computations are qualitatively and quantitatively similar. The leading shock's peak pressure is poorly resolved on the coarse mesh and is significantly better captured on the fine mesh compared to the coarse. The large secondary shocks of the experimental data are underpredicted in magnitude.

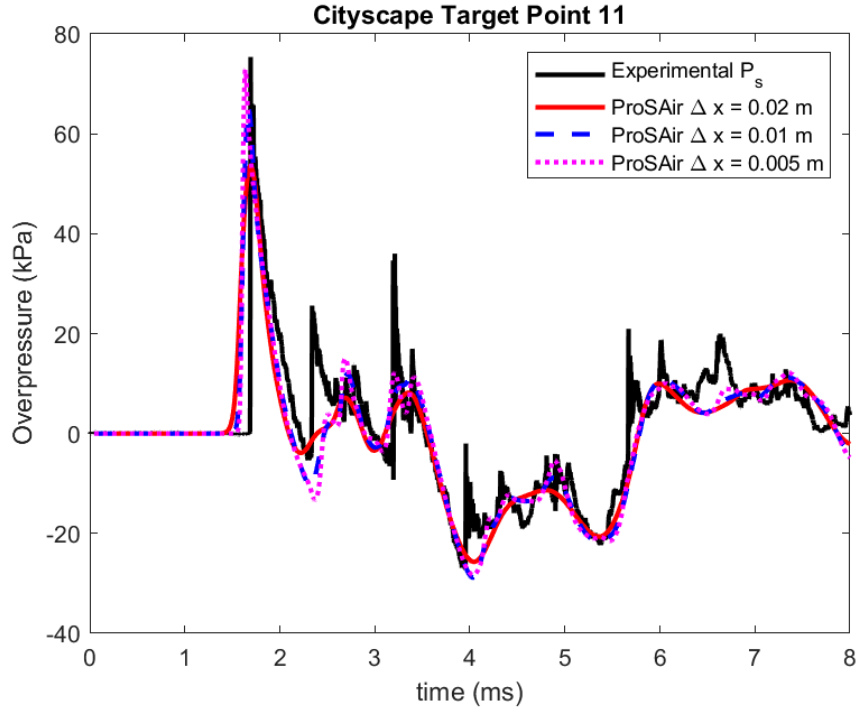


Figure 85: Overpressure history for target point  $T11$  on the three meshes of Table 8

In Fig. 85 we see that for Target  $T11$  the peak overpressure is computed in excellent agreement with the experimental data on the Fine mesh. Again there is good qualitative agreement with the experimental data except for the size of the secondary shocks.

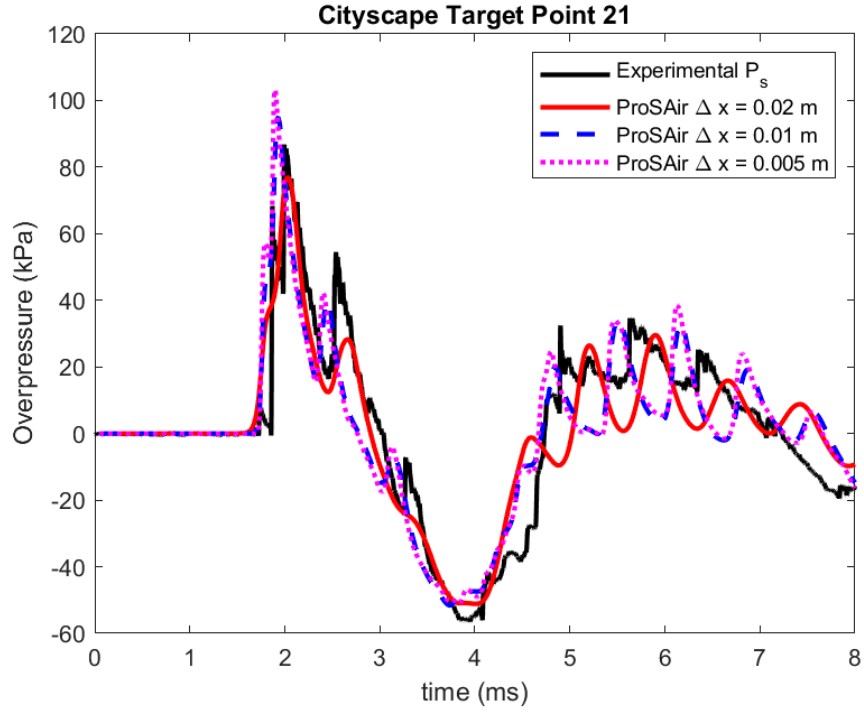


Figure 86: Overpressure history for target point  $T21$  on the three meshes of Table 8

In Fig. 86 the experimental data indicates the arrival of shocks at target  $T21$  at 1.72ms, 1.86ms and 2.0ms. The computations on the fine mesh pick up most of this fine structure, but with the first two shocks coalesced and with the shock arrival times slightly earlier than the experiment. Thereafter the general agreement with experiment is good.

On examining the `target_summary_3d.csv` files for these 3 computations we see that the blast parameter information recorded for the target points are as in Table 9 and 10. For the positive phase we also include data extracted from the experimental data.

| Target | Mesh       | $P_s$ (KPa) | $T_a$ (ms) | $I_s$ (KPa · ms) | $T_s$ (ms) |
|--------|------------|-------------|------------|------------------|------------|
| 1      | Coarse     | 127.129     | 0.631076   | 34.5196          | 1.03871    |
|        | Medium     | 137.486     | 0.722558   | 29.5621          | 0.605113   |
|        | Fine       | 148.755     | 0.767392   | 29.6784          | 0.575004   |
|        | Experiment | 207.7       | 0.818      | 31.09            | 0.5        |
| 3      | Coarse     | 31.9117     | 1.08415    | 18.8103          | 1.73928    |
|        | Medium     | 45.6026     | 1.19546    | 8.48136          | 0.606481   |
|        | Fine       | 52.9419     | 1.27914    | 8.45529          | 0.521335   |
|        | Experiment | 66.48       | 1.382      | 7.76             | 0.421      |
| 11     | Coarse     | 53.5616     | 1.30622    | 15.3234          | 0.79012    |
|        | Medium     | 65.0874     | 1.43792    | 15.7254          | 0.679979   |
|        | Fine       | 72.8603     | 1.50504    | 15.9425          | 0.600866   |
|        | Experiment | 75.36       | 1.692      | 13.70            | 0.518      |
| 21     | Coarse     | 76.9364     | 1.44461    | 38.481           | 1.45879    |
|        | Medium     | 94.8354     | 1.59053    | 38.4824          | 1.1905     |
|        | Fine       | 103.232     | 1.66432    | 39.0172          | 1.10188    |
|        | Experiment | 86.8        | 1.725      | 42.07            | 1.295      |

Table 9: Blast parameters for first positive phase for the Cityscape example on the three meshes of Table 8

From Table 9 for the first positive phase we see that results from Domains 2 and 3 are in excellent agreement. With the exception of time of arrival  $T_a$  and phase duration  $T_s$  even the results on Domain 1 are well converged. The agreement with experiment is also good and sufficient to give confidence in use of ProSAir as a predictive tool.

| Target | Mesh   | $P_s$ (KPa) | $T_a$ (ms) | $I_s$ (KPa · ms) | $T_s$ (ms) |
|--------|--------|-------------|------------|------------------|------------|
| 1      | Coarse | -32.0358    | 1.67738    | -33.77           | 1.70736    |
|        | Medium | -0.431992   | 1.32978    | -0.00885626      | 0.0324678  |
|        | Fine   | -7.25509    | 1.3424     | -0.482408        | 0.107338   |
| 3      | Coarse | -14.5705    | 2.82462    | -4.79267         | 0.516853   |
|        | Medium | -1.56677    | 1.80505    | -0.108648        | 0.104051   |
|        | Fine   | -2.62952    | 1.80048    | -0.182729        | 0.116849   |
| 11     | Coarse | -3.87261    | 2.10255    | -0.696633        | 0.294479   |
|        | Medium | -10.2102    | 2.12241    | -2.13207         | 0.346355   |
|        | Fine   | -13.2537    | 2.10591    | -2.67389         | 0.353556   |
| 21     | Coarse | -51.1512    | 2.90414    | -54.8689         | 2.12461    |
|        | Medium | -51.5809    | 2.78531    | -55.3248         | 1.91983    |
|        | Fine   | -51.1743    | 2.7662     | -55.8494         | 1.90532    |

Table 10: Blast parameters for first negative phase for the Cityscape example on the three meshes of Table 8

## 4 Simulation Settings

The default **Simulation Settings** window is shown in Fig. 87.

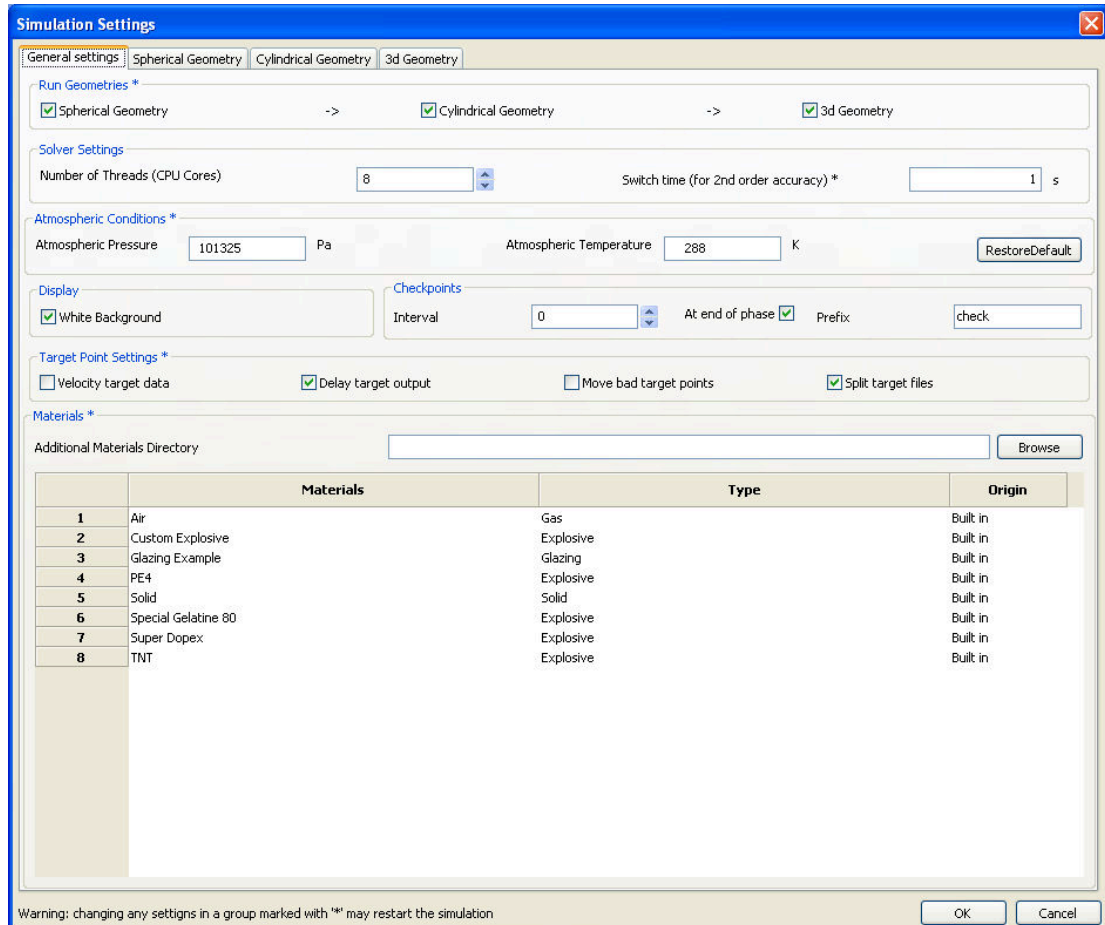


Figure 87: The default **Simulation Settings** window showing the **General Settings** tab

The window has 4 tabs:

1. **General Settings** - controls: which geometries are run (spherical, cylindrical, 3d); some generic solver settings used across all 3 geometric types; display and checkpoint control; generic target point information; materials. See § 4.1.
2. **Spherical Geometry** - controls any spherical computation phase and detonation of spherical charges. See § 4.2.
3. **Cylindrical Geometry** - controls any cylindrical computation phase. See § 4.3.
4. **3d Geometry** - controls any three dimensional computation phase. See § 4.4.

**Meaning of asterisk\* in Simulation Settings window** In any of these tabs an asterisk by a setting indicates that a computation must be, at least partially, restarted if that setting is changed - ProSAir will perform this restart automatically. For example, changing properties of the explosive charge necessitates re-performing the whole simulation no matter what the current state of the computation is. However, changing the cell size in the cylindrical phase does not affect the spherical solution phase and so computation recommences from the end of the spherical phase.

## 4.1 General Settings Tab

The **General Settings** tab is displayed whenever the **Simulation Settings** window is opened from the **Simulation** drop-down menu described in §2.4. The default settings for this window may be seen in Fig. 87. The **General Settings** tab is divided into 7 panes described below.

### 4.1.1 Run Geometries\*

The user can decide which geometric symmetries to exploit in the computation by ticking the appropriate boxes for **Spherical Geometry**, **Cylindrical Geometry** or **3d Geometry**. Note that, in the present version of ProSAir, explosive charges are modelled as spherical and so all associated computations will have an initial spherical phase. A cylindrical phase may then be used if the charge is at the centre of a geometry with cylindrical symmetry or the explosion interacts with a single planar surface (the ground?) for a significant time before interacting with any three-dimensional geometry.

### 4.1.2 Solver Settings

These are generic settings used across spherical, cylindrical and 3d computations.

- **Number of Threads (CPU Cores)** determines how many parallel execution threads should be used in the computation. To minimise run times this should be set to the number of cores on the computer. Under the *ProSAir Workstation and Multiuser licences* a maximum of 4 cores can be used. The *ProSAir8 Core Add-on* allows for up to 8 cores. Users with specialist requirements should contact the author. Note that, at present, spherical computations are performed using a single core.
- **Switch time (for 2nd order accuracy)\*** determines the simulated time at which ProSAir switches from first to second order accuracy. Using second order accuracy too early in the simulation may result in failure of the solver. Using first order for too long gives an unnecessarily inaccurate solution. Note that in ProSAir version 2009.10 and earlier (and in Air3d) this time was given in terms of a *switch factor*, i.e., a multiple of the  $1.2 \times 10^{-3} s / (\text{charge mass})^{1/3}$  taken for the second shock to form at the centre of a spherical airburst.

The recommended value is

$$\text{switch time} = 1.2 \times 10^{-3} \times \sqrt[3]{\text{TNT equivalent charge mass}}. \quad (4)$$

For compatibility with ProSAir prior to version 2010.5 and Air3d multiply the *switching factor* used by  $1.2 \times 10^{-3} \text{s}/(\text{charge mass})^{1/3}$  and use this figure as the **Switch time (for 2nd order accuracy)**.

#### 4.1.3 Atmospheric Conditions\*

As of Release 2011.3 atmospheric conditions may be varied in the solver - perhaps to simulate detonations at altitude.

- **Atmospheric pressure** - the atmospheric pressure in Pa may be specified.
- **Atmospheric temperature** - the atmospheric temperature in K may be specified.
- **Restore Default** - restores atmospheric pressure and temperature to their default values of one atmosphere 101,325Pa and 288K as per Air3d [?] and (almost<sup>1</sup>) corresponding to the International Standard Atmosphere at zero altitude.

#### 4.1.4 Display

Tick the box for a white background in ProSAir's graphical display otherwise a black background is used (as was the default for Air3d).

#### 4.1.5 Checkpoints

Checkpoints are files produced by ProSAir containing the solution settings and the entire flow solution at particular time steps. They can be used to restart a partially complete solution or provide flow field plots during post-processing. Checkpoint files can become large, particularly for three-dimensional simulations, and so will consume storage resources and CPU time.

- **Interval** - the number of time steps between checkpoints; a value of 0 (zero) indicates that no checkpoints are to be produced expect for those at the end of the spherical, cylindrical and 3d solution phases.
- **At end of phase** - if ticked (the default) then checkpoints are saved at the end of each solution phase, if not ticked checkpoints at not saved at the end of each solution phase. Before Release 2011.3 checkpoints were always saved at the end of phases. Note that checkpoints must be saved at the end of phases to enable automatically restarting from the end of a previous phase if parameters are changed requiring (partial) restarts. If checkpoints have not been saved then the user must use the menu option **Simulation-Restart** (§2.4) to restart the computation in the event of such changes.

---

<sup>1</sup>ISA has temperature of 15°C = 288.15K at zero altitude

- **Prefix** - by default checkpoint files are of the form `check_spherical_22.prc` (time step 22 of the spherical phase), `check_spherical_final.prc` (the final time step of the spherical phase), `check_cylindrical_22.prc` (time step 22 of the cylindrical phase), `check_3d_22.prc` (time step 22 of the 3d phase) etc. Changing the **prefix** changes the naming of these files. e.g., changing **prefix** to **hello** would result in checkpoint files named `hello_spherical_22.prc` etc.

#### 4.1.6 Target Point Settings\*

ProSAir allows the user to nominate *Target Points* in the **Spherical Geometry**, **Cylindrical Geometry** and **3d Geometry** tabs described in §4.2-4.4. For each target point ProSAir records the simulation time and gas density, energy, overpressure and temperature. The following settings may also be used:

- **Velocity Target Points** - if ticked then the gas velocity components are recorded in addition to gas density, energy, overpressure and temperature. By default this option is un-ticked as it doubles the size of the target files.
- **Delay Target Output** - if ticked then ProSAir only starts recording target data once the blast reaches the target point, otherwise target data is recorded from the start of the simulation phase.
- **Move bad target points** - If set, ProSAir will attempt to move target points found to be within obstacles. The target points will be moved by a maximum of one cell and ProSAir will attempt to move them in the following order: negative  $x$ -direction, positive  $x$ -direction, negative  $y$ -direction, etc. Unlike Air3d, ProSAir will not attempt to move the target points diagonally.
- **Split target files** - if set then at the end of the computation phase ProSAir will write a target point file for each individual target point. Otherwise, target point data is left in files containing up to 20 target points' data each.

#### 4.1.7 Materials\*

ProSAir uses a simple library of parameter definitions for *Explosives*, *Fluids* and *Glazing*. Properties for several such materials are supplied in the **Materials** folder of the ProSAir distribution - such materials are said to be *Built in* to the ProSAir distribution. There is the option for users to provide their own material parameter definitions files.

- **Additional Materials Directory** - the user may specify the path to a folder of their own Material parameter definition files either directly, or via the **Browse** button. See § 5.
- Table of **Materials**, **Type** and **Origin**. This presents the currently available materials. Such materials cannot be modified from this table and **Built in** items should

not be modified at all. See §5 for definition of you own materials or §4.2.2 for use of the **Custom Explosive** option.

## 4.2 Spherical Geometry Tab

The default **Spherical Geometry** tab is shown in Fig. 88. The settings of this tab may

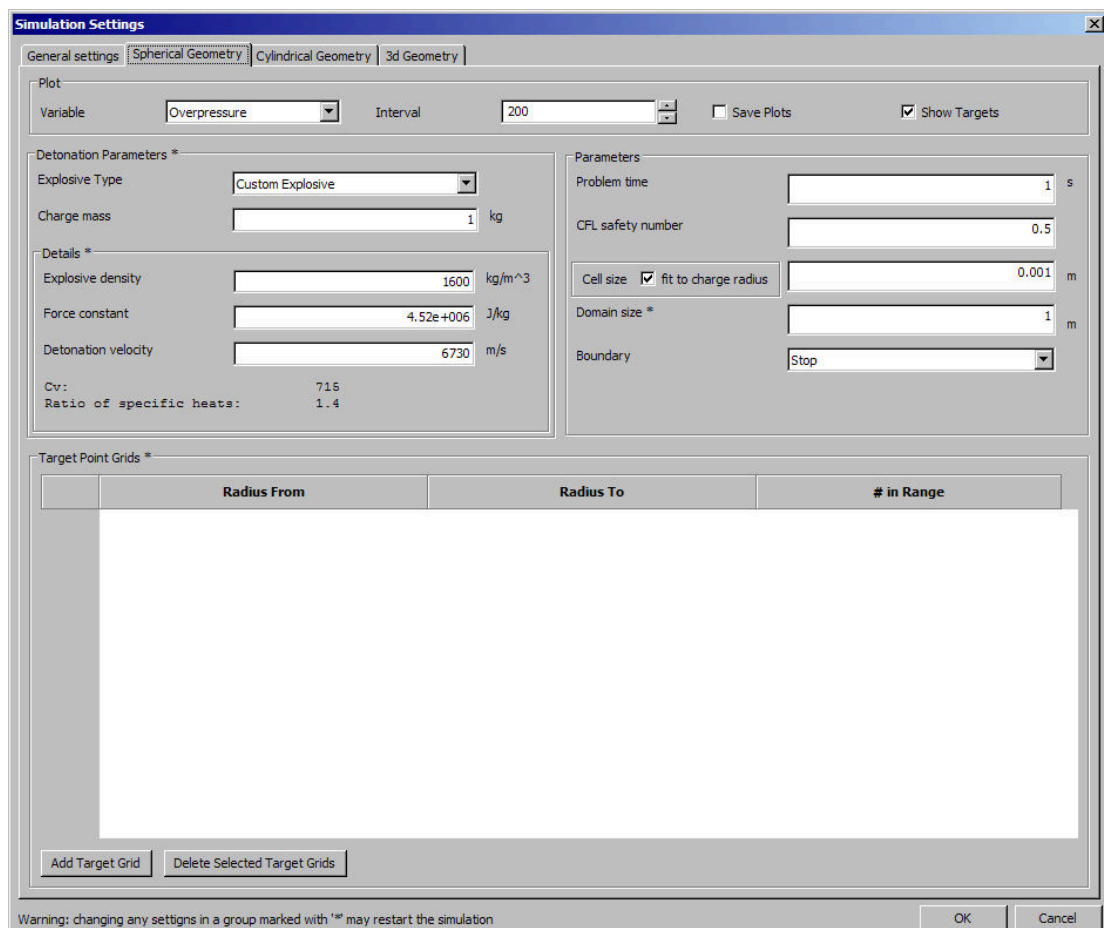


Figure 88: The default **Spherical Geometry** tab

be used to simulate an initial spherical, or hemispherical, detonation. The **Spherical Geometry** tick box of the **Run Geometries** pane of the **General Settings** tab must be ticked to allow this tab's settings to be changed as described in §4.1.1.

This tab contains four panes which we now describe.

### 4.2.1 Plot

This pane allows the user to specify what variables are plotted in ProSAir's main window and how they are plotted.

- **Variable** - allows the user to specify which flow variable is plotted via a drop-down menu. Choices are:
  - **Density** - the gas density ( $\text{kg/m}^3$ ).
  - **Vx** - gas velocity in the radial direction ( $\text{m/s}$ ).
  - **Vy** - although permitted by the ProSAir GUI this selection results in no useful information being displayed for spherical geometry calculations.
  - **Vz** - although permitted by the ProSAir GUI this selection results in no useful information being displayed for spherical geometry calculations.
  - **Energy** - the gas's internal energy ( $\text{KJ/kg}$ ).
  - **Overpressure** - the gas's pressure above atmospheric ( $\text{KPa}$ ).
  - **Temperature** - the gas's temperature ( $^{\circ}\text{C}$ ).
  - **Propellant Fraction** - not yet supported.

Note that **Propellant Fraction** is not yet supported.

- **Interval** - the number of time steps performed between the plot in the ProSAir main window being refreshed. i.e., **Interval** = 1 results in the plot being updated every time step. **Interval** is an integer number greater than zero.
- **Save Plots** - if this tick box is ticked then each plot displayed in the ProSAir main window is also saved as a png file to a file in the current folder with the file named `frame_spherical_[tstep].png` where `[tstep]` denotes the time step number.
- **Show Targets** - if this tick box is ticked then **target points**, as defined in §4.2.4, will be displayed by a vertical line labelled with target point number in the display of the ProSAir main window.

#### 4.2.2 Detonation Parameters\*

This pane allows the user to specify an initial, centrally detonated, spherical charge for the computation. By default, this pane opens with the **Explosive Type** set as **Custom Explosive**, as in Fig. 88, and allows the following to be specified.

- **Explosive Type** - the user chooses from a drop down menu of Built in and user defined explosives from the **Materials** pane (§ 4.1.7) from the **General Settings** tab (§4.1).
- **Charge Mass** - the explosive charge's mass in kg must be supplied. If a hemispherical ground burst computation is to be performed then the mass should be twice the actual charge mass to allow for half the computation to be removed as it is transferred to the cylindrical or 3d computation - a value of 1.8 might be used to account for losses due to cratering effects.

Provided the **Explosive Type** is kept as **Custom Explosive** then the **Details** sub-pane allows the user to specify:

- **Explosive density** - mass density of the explosive in  $\text{kg/m}^3$ .
- **Force Constant** - specific chemical energy of the explosive, i.e., how much energy is released on detonation per unit mass of explosive, ( $\text{J/kg}$ ).
- **Detonation velocity** - the speed with which the detonation front traverses the explosive starting from the charge center.

The explosive's detonation products are assumed to be an ideal gas with the same properties as air, i.e. a ratio of specific heats of 1.4 and a specific heat at constant volume of  $715\text{J/kg/K}$ .

If the **Explosive Type** is changed to be one of the Built in or user supplied explosives then the **Charge Mass** must be selected but the **Details** sub-pane is filled with data from the explosive definitions file and may not be modified directly - see § 5. The **Detonation Parameters** pane then looks as in, for example, Fig. 89.

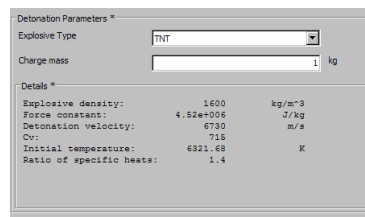


Figure 89: Detonation parameters pane for Built in explosive TNT.

#### 4.2.3 Parameters

These parameters determine the numerical integration for the spherical computation.

- **Problem time** - the maximum simulated time in s for the spherical computation. Note that if **Boundary** is set to **STOP** the spherical computation may finish before this time.
- **CFL safety number** - this number governs how large the time-step taken is compared to the time taken for a shock wave to cross a computational cell. A sensible default is 0.5 and the theoretical maximum for a smoothly evolving computation is 0.75. Selecting too small a CFL safety number results in protracted run times. Selecting too high a number results in instability and non-physical results, perhaps even negative pressures or densities leading to the computation to be stopped.

- **Cell size** - the length (m) of a computational cell. Rose [?] recommends having 50 cells across the radius of a spherical charge. If the tick box **fit to charge radius** is checked then the **Cell size** is adjusted to ensure that the charge radius exactly coincides with a cell boundary (for details see §8.1). This adjustment to the **Cell size** does not show up in the **Cell size** input box.
- **Domain size** - the maximum radius (m) for the computation. Typically this might be the distance from the charge's centre to the ground or the nearest obstacle as once the shock wave reaches such an obstacle then spherical symmetry is no longer valid.
- **Boundary** - drop down menu to choose the boundary condition at the edge of the spherical domain. Choices are:
  - **Reflective** - a rigid outer spherical boundary is used - shock waves are reflected back into the interior of the domain.
  - **Stop** - the pressure in the cell adjacent to the outer boundary is monitored and calculation stops once the pressure changes. This might be used to then trigger a switch to a cylindrical or 3d computation. Note that on coarse meshes the effects of numerical dissipation may result in this mechanism stopping computation before the shock appears to *reach* the boundary.
  - **Transmissive** - the shock is allowed to pass out of the domain with computation proceeding. Note that this boundary condition is only valid for supersonic outflow of gas and will result in inaccuracies if applied inappropriately.

#### 4.2.4 Target Point Grids

A *Target Point* may be thought of as a measuring point in the flow field at which the gas's state will be monitored and recorded. If one wishes to compare simulated results to data from experimental pressure taps or thermal images then target points should be placed at the measurement locations. Target points are specified as a grid, i.e., a number of points equally spaced between an inner and an outer radius.

- **Add Target Grid** - clicking on this button adds a new grid of points as seen in Fig. 90. By default a grid of a single point running from a radius of 0m to 0m is set. The user modifies the **Radius From**, **Radius To** and **# in Range** as required. For example for 10 points equally spaced from 1m to 10m the pane should be adjusted to as in Fig 91.
- **Delete Selected Target Grids** - by clicking on the left hand side number of a Target Grid to select it, and then clicking this button, the Target Grid is removed from the computation.

|   | Radius From | Radius To | # in Range |
|---|-------------|-----------|------------|
| 1 | 0.0         | 0.0       | 1          |

Add Target Grid    Delete Selected Target Grids

Figure 90: Adding a **Target Point Grid** to the **Spherical Geometry** tab.

|   | Radius From | Radius To | # in Range |
|---|-------------|-----------|------------|
| 1 | 1           | 10        | 10         |

Add Target Grid    Delete Selected Target Grids

Figure 91: Specifying a **Target Point Grid** of 10 points equally spaced between  $r = 1\text{m}$  to  $r = 10\text{m}$ .

## 4.3 Cylindrical Geometry Tab

The default **Cylindrical Geometry** tab is shown in Fig. 92. The settings of this tab may be

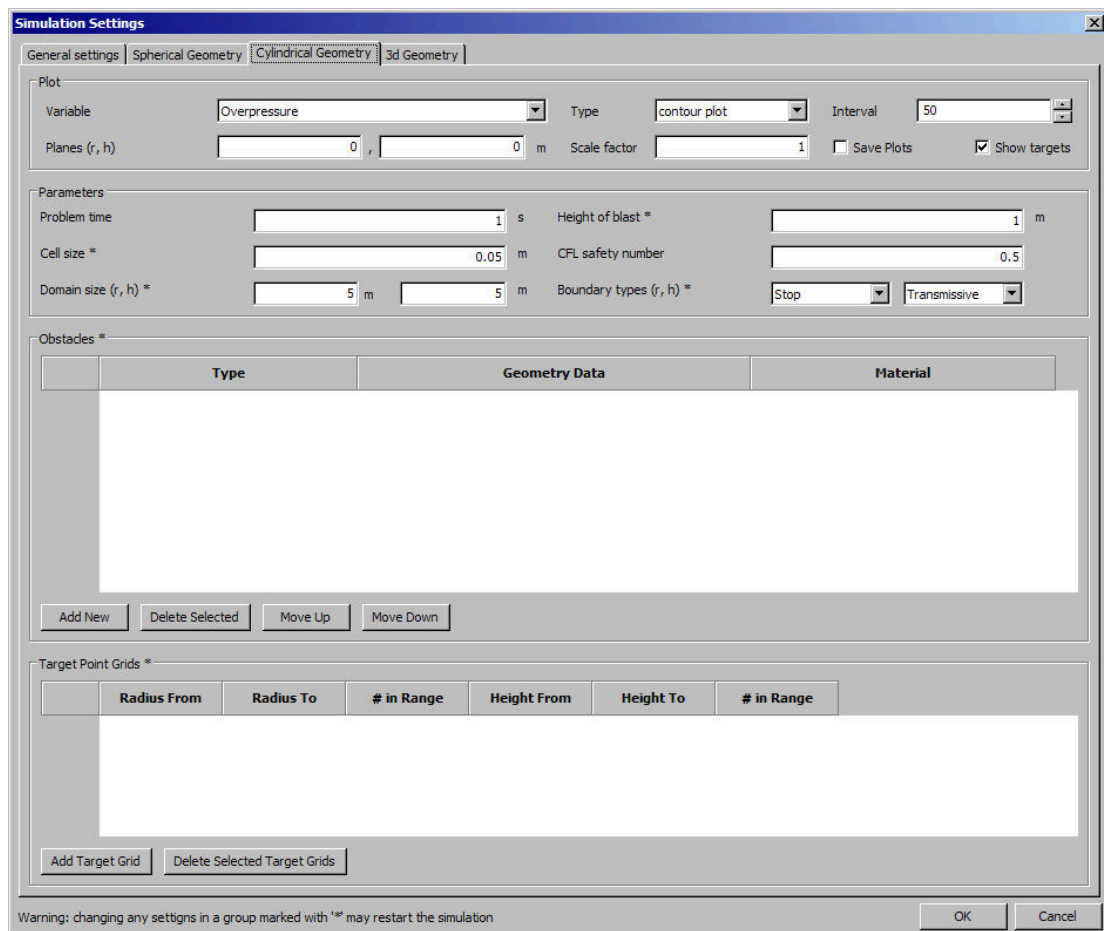


Figure 92: The default **Cylindrical Geometry** tab

used to progress an initially spherical detonation through a cylindrically symmetric phase during which the explosion interacts with a single planar reflective (solid) lower boundary. In addition reflective outer and upper boundaries may be set and there is scope for specifying cylindrically symmetric geometry components (cylinders, hollow cylinders, etc.). The **Cylindrical Geometry** tick box of the **Run Geometries** pane of the **General Settings** tab must be ticked to allow this tab's settings to be changed as described in §4.1.1.

This tab contains four panes which we now describe.

### 4.3.1 Plot

This pane allows the user to specify what variables are plotted in ProSAir's main window and how they are plotted.

- **Variable** - allows the user to specify which flow variable is plotted via a drop-down menu. Choices are:
  - **Density** - the gas density ( $\text{kg/m}^3$ ).
  - **Vx** - gas velocity in the radial direction ( $\text{m/s}$ ).
  - **Vy** - gas velocity in the vertical direction ( $\text{m/s}$ ).
  - **Vz** - although permitted by the ProSAir GUI this selection results in no useful information being displayed for spherical geometry calculations.
  - **Energy** - the gas's internal energy ( $\text{KJ/kg}$ ).
  - **Overpressure** - the gas's pressure above atmospheric ( $\text{KPa}$ ).
  - **Temperature** - the gas's temperature ( $^{\circ}\text{C}$ ).
  - **Propellant Fraction** - not yet supported.

Note that contour plots are only supported for **Density**, **Overpressure** and **Temperature**.

- **Type** - the type of plot to be displayed in the ProSAir main window while the cylindrical geometry computation phase is running. The drop down menu offers:
  - **line (r = constant)** - a plot of the **Variable** along a line of constant radius given by the first, **radius r** box of **Planes (r,h)** is displayed.
  - **line (h = constant)** - a plot of the **Variable** along a line of constant height given by the second, **height h** box of **Planes (r,h)** is displayed.
  - **contour plot** a contour plot of the **Variable** through the domain is produced.
  - **all plots** - all 3 of the above plots are produced.
- **Interval** - the number of time steps performed between the plot in the ProSAir main window being refreshed. i.e., **Interval**= 1 results in the plot being updated every time step. **Interval** is an integer number greater than zero.
- **Planes (r,h)** - if **Type** is set to
  - **line (r = constant)** then the first number supplied is used as the required radius **r** for a plot of the **Variable** for constant, given radius and height **h** varying through the domain.
  - **line (h = constant)**, then the second number supplied is used as the required height **h** for a plot of the **Variable** for constant, given height and radius **r** varying through the domain.
  - **all plots** - then supplied radius **r** and height **h** are both used.
- **Scale factor** - by default the positive range of scalar quantities for display purposes is:

- Density - 0 – 2kg/m<sup>3</sup>
- Overpressure - 0 – 100Pa
- Temperature - 0 – 500°C

Use **Scale factor** to change the upper end of this range. e.g., a scalar factor of 2 changes the ranges to: density 0 – 4kg/m<sup>3</sup>, overpressure 0 – 200Pa, temperature 0 – 1000°C.

- **Save Plots** - if this tick box is ticked then each plot displayed in the ProSAir main window is also saved as a png file to a file in the current folder named `frame_cylindrical_[tstep].png` where `[tstep]` denotes the time step number.
- **Show Targets** - if this tick box is ticked then **target points** will be displayed by a cross labelled with target point number in the display of the ProSAir main window.

#### 4.3.2 Parameters

This pane allows the user to specify the parameters for the cylindrical solution phase.

- **Problem Time** - the maximum time for the cylindrical simulation in seconds (s). Note that if one of the **Boundary Types** is set to **Stop** then the simulation may end when the shock front reaches the corresponding boundary and the **Problem Time** may not be reached.
- **Cell size\*** - the size of the computational cells in m used. Also known as the grid or mesh spacing. The same size is used in the  $r$  and  $h$  directions. Users should try to use a cell size that exactly divides into the **Height of blast** in order to give an integer number of cells between the centre of detonation and the ground.
- **Domain size (r, h)\*** - the size of the domain in the  $r$  and  $h$  directions in m. When possible, these sizes should be integer multiples of the **Cell size** in order to exactly match the domain's extents.
- **Height of blast\*** - the height in m above the ground  $h = 0$  of the central detonation point of the spherical charge.
- **CFL Safety Number** - this number governs how large the time-step taken is compared to the time taken for a shock wave to cross a computational cell. A sensible default is 0.5 and the theoretical maximum for a smoothly evolving computation is 0.75. Selecting too small a CFL safety number results in protracted run times. Selecting too high a number results in instability and non-physical results, perhaps even negative pressures or densities leading to the computation to be stopped.
- **Boundary types (r,h)\*** - drop down menu to choose the boundary condition at the outer radial (r) edge and upper (h) edge of the cylindrical domain. Choices are:

- **Reflective** - a rigid outer boundary is used - shock waves are reflected back into the interior of the domain.
- **Stop** - the pressure in the cell adjacent to the outer boundary is monitored and calculation stops once the pressure changes. This might be used to then trigger a switch to a 3d computation. Note that on coarse meshes the effects of numerical dissipation may result in this mechanism stopping computation before the shock appears to *reach* the boundary.
- **Transmissive** - the shock is allowed to pass out of the domain with computation continuing. Note that this boundary condition is only valid for supersonic outflow of gas and will result in inaccuracies if applied inappropriately.

### 4.3.3 Obstacles\*

This pane allows users to define obstacles which have cylindrical symmetry, i.e., may be created by defining a two-dimensional shape in the  $(r, h)$  plane which corresponds, in three-dimensions, to that shape being rotated about the  $r = 0$  axis. For example, consider the cylindrical containment vessel of §3.3.1 consisting of a hollow cylinder with inner radius 3m, outer radius 3.096m, inner height 4m, outer height 4.096m and with a central 0.25m radius circular vent in the centre of the roof. One quarter of the geometry is shown in Fig. 43(a).

The cylinder is also bounded by the ground at  $h = 0$ . In the cylindrical domain this geometry may be realised by rotating two rectangles about the  $h$ -axis as shown in Fig. 43 (b).

- **Add New** - clicking on this button adds a new obstacle to the list of obstacles, by default it is a cuboid obstacle as shown in Fig. 93. For such an obstacle:

|   | Type   | Geometry Data       | Material |
|---|--------|---------------------|----------|
| 1 | cuboid | rmin rmax hmin hmax | Solid    |

Figure 93: Adding a new obstacle to the cylindrical domain.

- **Type** may be set to **cuboid** (rotated rectangle), **wedge** (rotated triangle) or **sphere** (rotated circle - a torus).
- **Geometry Data** is used to specify the obstacles dimensions as specified by the text entries as the obstacle is created or has its **Type** changed.

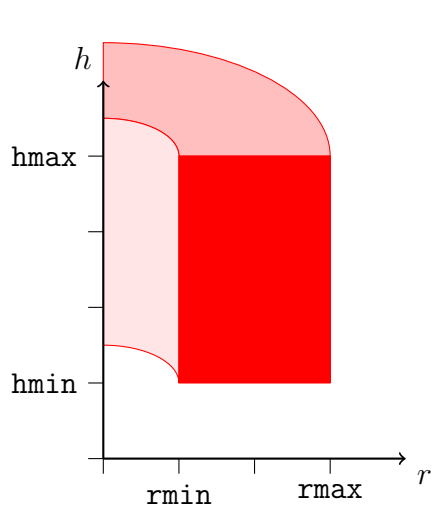
- **Material** may be used to set the obstacle to be **Solid** or **Air**. By combining Solid and Air objects in the specified obstacle number order hollowed geometries may be created.

The dimensions used to define these obstacles are depicted in Fig. 94.

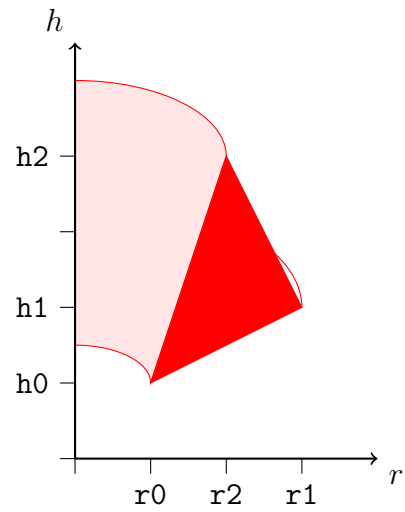
- **Delete Selected** - deletes the selected obstacle from the obstacle list.
- **Move Up** - moves the selected obstacle one place up the obstacle list.
- **Move Down** - moves the selected obstacle one place down the obstacle list.

#### 4.3.4 Target Point Grids\*

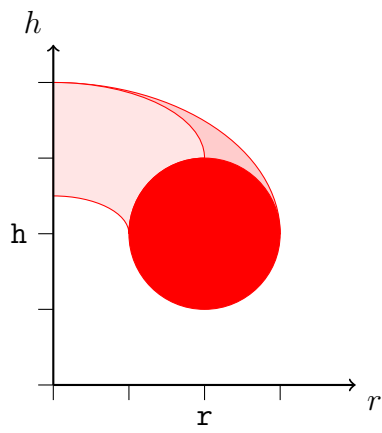
This panel allows for the setting of target points in the cylindrical domain in much the same way as for the spherical domain described in §4.2.4. The only difference is that now a grid of target points is specified with start and end coordinates and number of points in both the  $r$  and the  $h$  coordinate directions.



(a) Cylindrical cuboid object



(b) Cylindrical wedge object



(c) Cylindrical sphere object

Figure 94: Cylindrical Obstacles

## 4.4 3d Geometry Tab

The default 3d Geometry tab is shown in Fig. 95. The settings of this tab may be used to

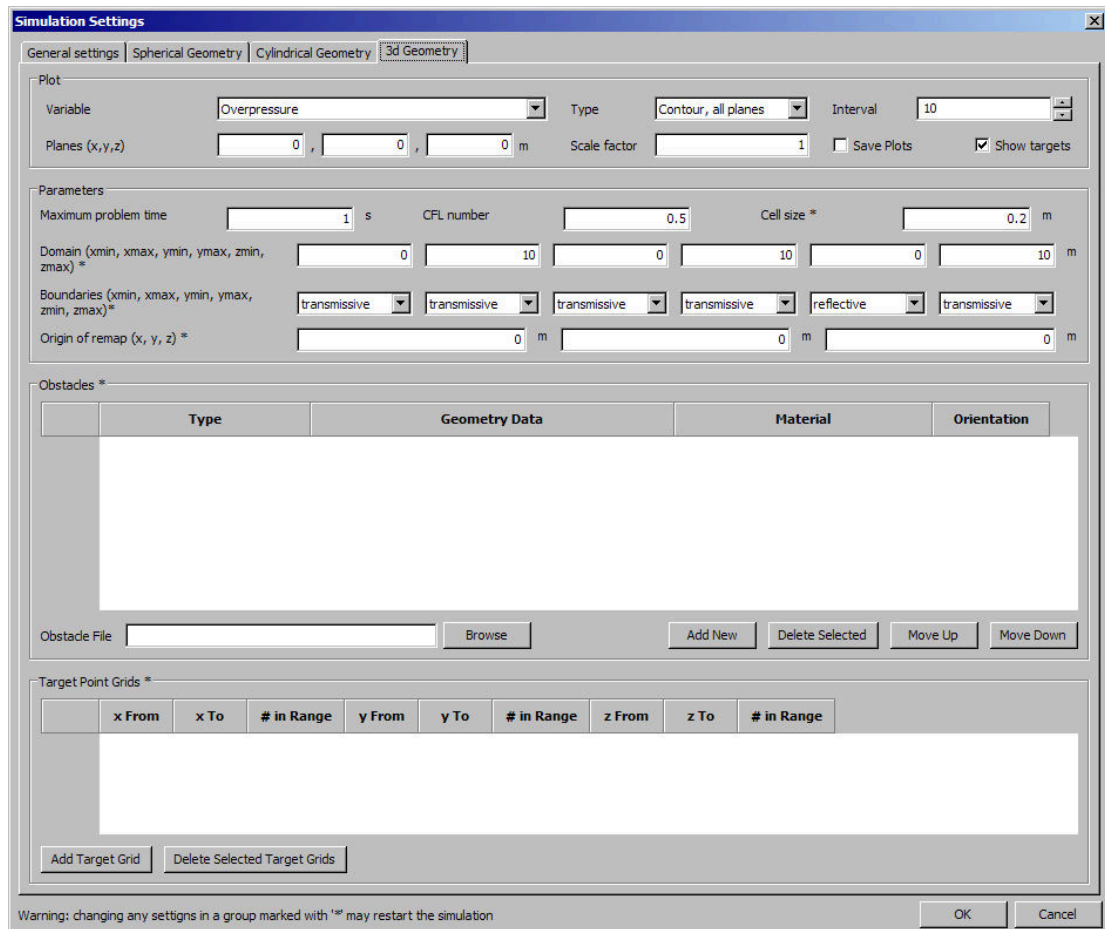


Figure 95: The default 3d Geometry tab

progress an initially spherically symmetric or cylindrically symmetric solution through a three-dimensional solution phase during which the explosion interacts with outer boundaries which may transmit or reflect the shockwave. There is scope for specifying geometry components (cuboids, cylinders, spheres, wedges) and frangible windows. The 3d Geometry tick box of the Run Geometries pane of the General Settings tab must be ticked to allow this tab's settings to be changed as described in §4.1.1.

This tab contains four panes which we now describe.

### 4.4.1 Plot

This pane allows the user to specify what variables are plotted in ProSAir's main window and how they are plotted.

- **Variable** - allows the user to specify which flow variable is plotted via a drop-down menu. Choices are:
  - **Density** - the gas density ( $\text{kg/m}^3$ ).
  - **Vx** - gas velocity in the horizontal  $x$ -coordinate direction ( $\text{m/s}$ ).
  - **Vy** - gas velocity in the horizontal  $y$ -coordinate direction ( $\text{m/s}$ ).
  - **Vz** - gas velocity in the vertical  $z$ -coordinate direction ( $\text{m/s}$ ).
  - **Energy** - the gas's internal energy ( $\text{KJ/kg}$ ).
  - **Overpressure** - the gas's pressure above atmospheric ( $\text{KPa}$ ).
  - **Temperature** - the gas's temperature ( $^{\circ}\text{C}$ ).
  - **Propellant Fraction** - not yet supported.
- **Type** - the type of plot to be displayed in the ProSAir main window while the cylindrical geometry computation phase is running. The drop down menu offers:
  - **Contour, (x = constant)** - a contour plot of the **Variable** on a plane of constant horizontal coordinate  $x$  given by the first,  $x$  box of **Planes (z,y,z)** is displayed.
  - **Contour, (y = constant)** - a contour plot of the **Variable** on a plane of constant horizontal coordinate  $y$  given by the second,  $y$  box of **Planes (z,y,z)** is displayed.
  - **Contour, (z = constant)** - a contour plot of the **Variable** on a plane of constant vertical coordinate  $z$  given by the second,  $z$  box of **Planes (z,y,z)** is displayed.
  - **Contour, all planes** - all 3 of the above plots are produced.
- **Interval** - the number of time steps performed between the plot in the ProSAir main window being refreshed. i.e., **Interval**= 1 results in the plot being updated every time step. **Interval** is an integer number greater than zero.
- **Planes (x,y,z)** - if **Type** is set to
  - **Contour, (x = constant)** then the first number supplied is used as the required horizontal coordinate  $x$  for a plot of the **Variable** for constant, given coordinate.
  - **Contour, (y = constant)** then the second number supplied is used as the required horizontal coordinate  $y$  for a plot of the **Variable** for constant, given coordinate.
  - **Contour, (z = constant)** then the third number supplied is used as the required vertical coordinate  $z$  for a plot of the **Variable** for constant, given coordinate.

- **Contour, all planes** - then all three planes of constant  $x$ ,  $y$  and  $z$  are plotted corresponding to the three, supplied coordinates.
- **Scale factor** - by default the positive range of scalar quantities for display purposes is:
  - Density - 0 – 2kg/m<sup>3</sup>
  - Overpressure - 0 – 100Pa
  - Temperature - 0 – 500°C

Use **Scale factor** to change the upper end of this range. e.g., a scalar factor of 2 changes the ranges to: density 0 – 4kg/m<sup>3</sup>, overpressure 0 – 200Pa, temperature 0 – 1000°C.

- **Save Plots** - if this tick box is ticked then each plot displayed in the ProSAir main window is also saved as a png file to a file in the current folder named `frame_3d_[tstep].png` where `[tstep]` denotes the time step number.
- **Show Targets** - if this tick box is ticked then **target points** will be displayed by a cross labelled with target point number in the display of the ProSAir main window.

#### 4.4.2 Parameters

This pane allows the user to specify the parameters for the 3d solution phase.

- **Problem Time** - the maximum time for the cylindrical simulation in seconds (s). Note that if one of the **Boundary Types** is set to **Stop** then the simulation may end when the shock front reaches the corresponding boundary and the **Problem Time** may not be reached.
- **CFL Number** - this number governs how large the time-step taken is compared to the time taken for a shock wave to cross a computational cell. A sensible default is 0.5 and the theoretical maximum for a smoothly evolving computation is 0.75. Selecting too small a CFL safety number results in protracted run times. Selecting too high a number results in instability and non-physical results, perhaps even negative pressures or densities leading to the computation to be stopped.
- **Cell size\*** - the size of the computational cells in m used. Also known as the grid or mesh spacing. The same size is used in the  $r$  and  $h$  directions.
- **Domain (xmin, xmax, ymin, ymax, zmin, zmax)\*** - the extremes of the domain in the  $x$ ,  $y$  and  $z$  coordinate directions in m.
- **Boundaries (xmin, xmax, ymin, ymax, zmin, zmax)\*** - drop down menu to choose the boundary condition at the edges of the 3d domain. Choices are:

- **Reflective** - a rigid outer boundary is used - shock waves are reflected back into the interior of the domain.
- **Stop** - the pressure in the cell adjacent to the outer boundary is monitored and calculation stops once the pressure changes. Note that on coarse meshes the effects of numerical dissipation may result in this mechanism stopping computation before the shock appears to *reach* the boundary.
- **Transmissive** - the shock is allowed to pass out of the domain with computation continuing. Note that this boundary condition is only valid for supersonic outflow of gas and will result in inaccuracies if applied inappropriately.
- **Origin of remap (x,y,z)\*** - location in m of the origin of the cylindrical phase calculation<sup>2</sup> or, if no cylindrical phase were computed, the origin of the spherical phase computation<sup>3</sup>.

#### 4.4.3 Obstacles\*

This pane allows users to define obstacles in the 3d domain.

- **Obstacle File**- allows the user to nominate a file from which obstacle definitions may be read. The **Browse** button opens a file browser window to aid in this. This feature is currently not implemented.
- **Add New** - clicking on this button adds a new obstacle to the list of obstacles, by default it is a cuboid obstacle as shown in Fig. 96. For such an obstacle:

The screenshot shows a window titled "Obstacles \*". It contains a table with the following data:

|   | Type   | Geometry Data                 | Material | Orientation |
|---|--------|-------------------------------|----------|-------------|
| 1 | cuboid | xmin xmax ymin ymax zmin zmax | Solid    | n/a         |

Below the table, there is an "Obstacle File" text input field, a "Browse" button, and four buttons: "Add New", "Delete Selected", "Move Up", and "Move Down".

Figure 96: Adding a new obstacle to the 3d domain.

- **Type** may be set to one of the primitive obstacles **cuboid**, **cylinder**, **sphere**, **wedge** or **window**.
- **Geometry Data** is used to specify the primitive obstacle's dimensions as specified by the text entries as the obstacle is created or has its **Type** changed.

<sup>2</sup>Usually the ground coordinate directly below the charge centre.

<sup>3</sup>Usually the 3d coordinates of the centre of the spherical charge.

- **Material** may be used to set the obstacle to be **Solid** or **Air**. By combining Solid and Air objects in the specified obstacle number order hollowed geometries may be created.
- **Orientation** - some obstacle types have an associated orientation. For example, a **cylinder** may be oriented with its axis aligned with the  $x$ ,  $y$  or  $z$  coordinate axis. Windows may be aligned in the  $x$ ,  $-x$ ,  $y$ ,  $-y$ ,  $z$  or  $-z$  directions.

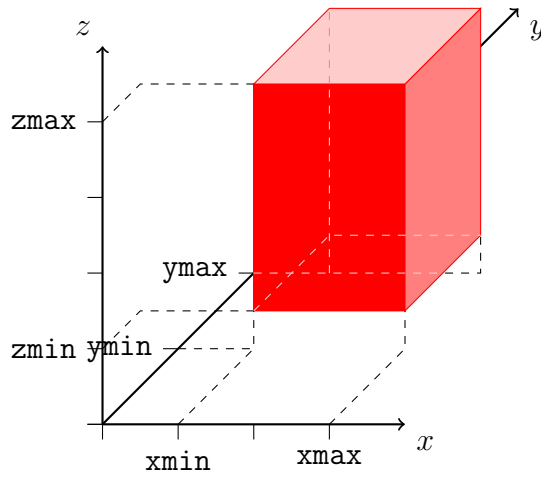
The dimensions used to define these obstacles are depicted in Fig. 97.

- **Delete Selected** - deletes the selected obstacle from the obstacle list.
- **Move Up** - moves the selected obstacle one place up the obstacle list.
- **Move Down** - moves the selected obstacle one place down the obstacle list.

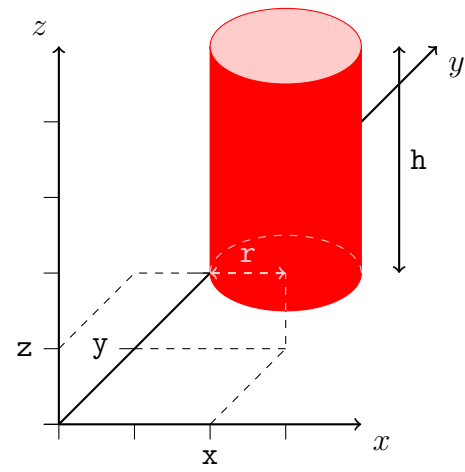
ProSAir uses a form of Constructive Solid Geometry (CSG) in which obstacles are constructed by combining primitive obstacles. The order of obstacles is important if they intersect. If the coordinates  $(x, y, z)$  of a given point  $\mathbf{x}$  lie within multiple obstacles then  $\mathbf{x}$  will have the **Material** (**Solid** or **Air**) of the latest of those obstacles in the list. The effect of this is that ProSAir assumes the domain is initially *filled* with Air. The first obstacle in the list is then added to the domain, say it is a **Cuboid** of **Material Solid**, then all points within the region occupied by the cuboid now are of **Material Solid**, and points outside remain of **Material Air**. Now suppose the second obstacle in the list is a **Sphere** of **Material Air** centred on the centre of the cuboid and with radius less than half the minimum edge length of the cuboid. The effect of this is to hollow out a spherical region of **Air** within the cuboid. Now any point that lies within the sphere (the last object) is of **Material Air**, any point lying within the cuboid but not the sphere is of **Material Solid**, and all other points in the domain are of **Material Air**.

#### 4.4.4 Target Point Grids\*

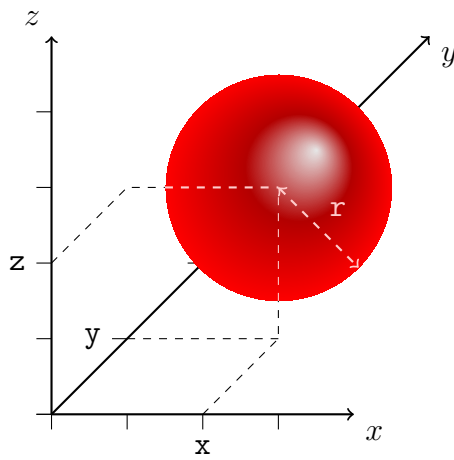
This panel allows for the setting of target points in the three dimensional domain in much the same way as for the cylindrical and spherical domains described in §4.3.4 and §4.2.4. The only difference is that now a grid of target points is specified with start and end coordinates and number of points in the three coordinate directions  $x$ ,  $y$  and  $z$ .



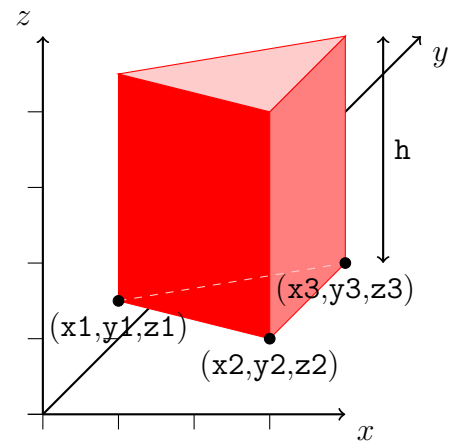
(a) 3d cuboid



(b) 3d cylinder with orientation  $z$



(c) 3d sphere



(d) 3d wedge with orientation  $z$

Figure 97: 3d Primitive Obstacles

## 5 User-Defined Materials Library

As described in §4.1.7 the user may choose to define a material's properties using one of ProSAir's built-in materials descriptions or may define their own. In this section we detail how to define materials of type **Explosive** and **Gas**. We defer description of **Glazing** materials until §6.

Materials are defined in files appropriate to gases (as described in §5.2) and explosives (as described in §5.3). These must reside in subfolders of a user's single user-defined materials folder as defined in §5.1.

### 5.1 Additional Materials Directory

To add additional materials you must first create an appropriate file structure.

1. Create a new folder in your folders named, say, **Additional Materials**.
2. Within that folder create further folders:
  - **Explosives** - to hold explosive definition files as explained in §5.3.
  - **Fluids** - to hold additional gas definition files as explained in §5.2.
  - **Glazing** - to hold additional glazing definition files (see §6.2 for further details on glazing).

Using the **Additional Materials Directory** box of the **Materials Panel** of the **General Settings** tab described in §4.1.7 the user may inform ProSAir of the location of their **Additional Materials Directory** and the materials definition files in the subfolders are automatically read by ProSAir and the materials definitions become available in the **General Settings** tab of Fig. 87.

### 5.2 Gas Materials

**Gas** materials define the equation of state and nominal temperature, pressure and density for an ideal gas. At present only the single **built in** ideal gas named **Air** may be used.

### 5.3 Explosive Materials

**Explosive** materials define the energy and mass released when a solid explosive is detonated together with the detonation velocity required to determine how long it takes for all its mass to be converted to gas once detonated. A typical explosive detonation file is that for TNT in the supplied **TNT.prmat** file of the **Materials/Explosives** folder of the ProSAir distribution and shown in Fig. 98.

We note:

- All explosive materials definitions are stored in a file with extension **.prmat** and the explosive is given the name of the file - in this case **TNT**.

```
#####
# Detonation parameter for TNT #
# Baker WE, Cox PA, Westine PS, Kulesz JJ & Strehlow RA #
# "Explosion Hazards and Evaluation." p. 110 #
# Elsevier Scientific Publishing Company, #
# ISBN 0-444-42094-0 (Vol 5), 1983. #
#####

[Explosive]
# Detonation velocity (m/s).
detonation_velocity = 6730

# Heat of explosion (previously labelled as "energy") (J/kg)
heat_of_explosion = 4.52e+6

# Explosive's density (kg/m^3)
density = 1600
```

Figure 98: The TNT.prmat supplied with ProSAir

- Within such a file all lines starting with a hash # are treat as comments and ignored by ProSAir.
- Explosive materials definition files contain a single section starting with the section label **[Explosive]**.
- Within the **[Explosive]** section three parameters must be supplied each identified by an appropriate keyword and a value must be assigned using=

1. The explosive's **density** in kg/m<sup>3</sup> by keyword **density**, e.g.,

```
density = 1600
```

assigns the explosive's density to be 1600kg/m<sup>3</sup>.

2. The explosive's **heat of explosion** in J/kg by keyword **heat\_of\_explosion**, e.g.,

```
heat_of_explosion = 4.52e+6
```

assigns the explosive's heat of explosion to be  $4.52 \times 10^6 \text{ J/kg} = 4520 \text{ KJ/kg}$ . Note that the heat of explosion is the heat released on detonation of the explosion and is less than the **heat of combustion** which accounts for burning of detonation

products. At present there is no recognised mechanism in ProSAir to account for afterburning of explosive products.

3. The explosive's **detonation velocity** in m/s by keyword **detonation\_velocity**, e.g.,

```
detonation_velocity = 6730
```

assigns the explosive's detonation velocity to be 6730m/s.

## 6 Glazing

ProSAir possesses a simple element failure model designed to handle failure of glazing elements. The model implemented is a slight variation on that of **Air3d** [?]. The model is based on Reflective Pressure vs. Reflective Impulse **Iso-Damage Curves** [?, ?]. Windows are defined via a special obstacle type **Window**.

### 6.1 Window Obstacle Type

A window obstacle is defined as for a cuboid of **Air** (See 4.4.3) except an **Orientation** is also give. The user supplied **xmin**, **xmax**, **ymin**, **ymax**, **zmin** and **zmax** define a region which is assumed to be occupied by **Air** (as shown in Fig. 99) except for a sheet, one cell thick, of **Solid** cells on the face specified by the **Orientation** (as shown in Fig. 100). In addition

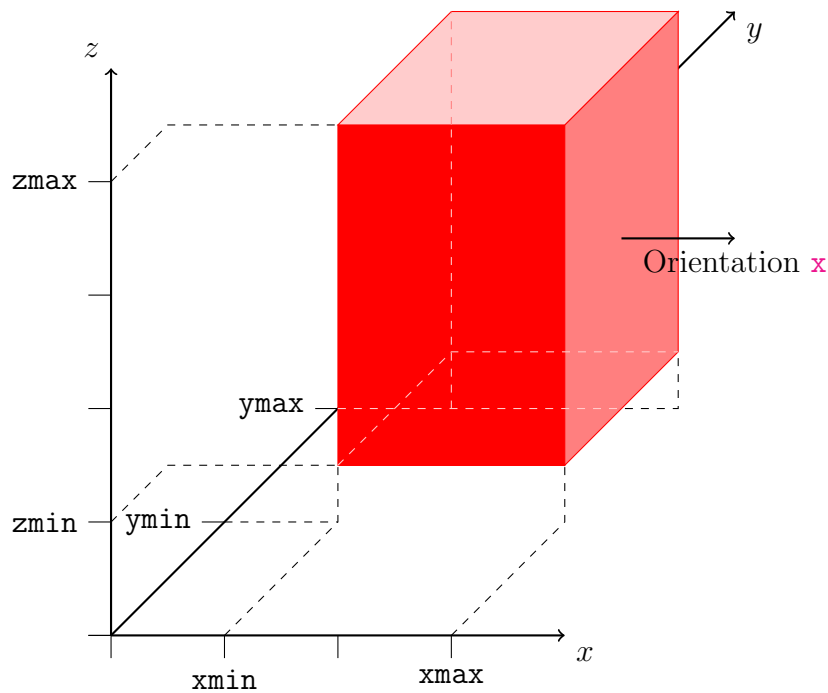


Figure 99: Extent of a window obstacle

to these geometric settings a window obstacle must also have a **Glazing Material**.

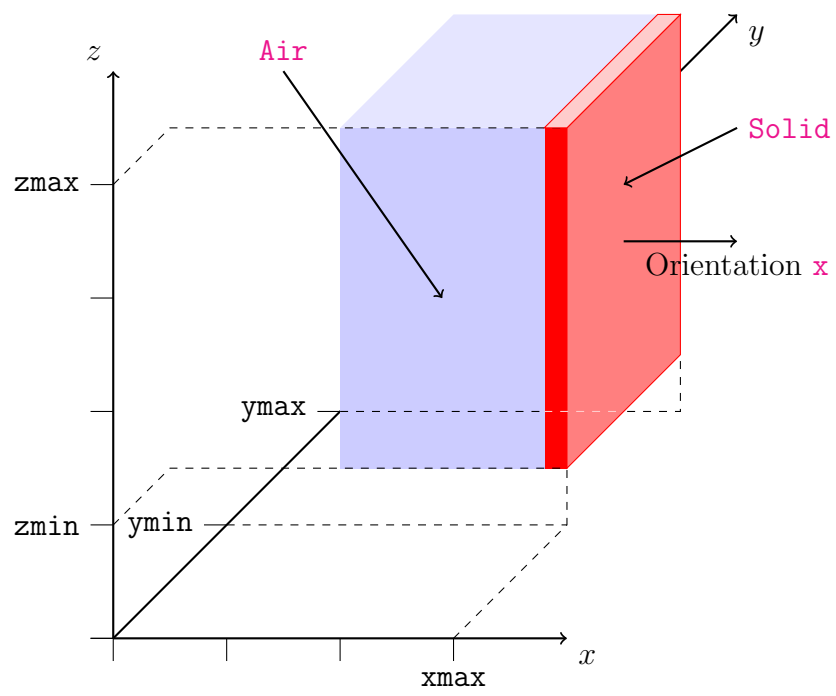


Figure 100: Location of **Air** and **Solid** cells within a Window obstacle with orientation  $-x$

## 6.2 Glazing Materials

Glazing materials are defined in a ProSAir Glazing materials file. Such files reside either in the `Materials\Glazing` folder supplied with ProSAir at `C:\Program Files\ProSAir32\Materials\Glazing` or in a user defined `Materials\Glazing` folder (see Sec. 5). A glazing materials file defines the pressure-impulse *PI* iso-damage failure curve for a window. There are currently three ways in which this failure curve may be defined.

### 6.2.1 **PI-Curve** - Glazing defined by Multiple Points on PI Curve

An example definition file is `Glazing PI.prmat` of the `Materials\Glazing` folder supplied with ProSAir and shown in Fig. 101,

```
#####  
# Glazing P-I curve definition  
# Breaks safely curve  
# 1.25m * 1.55m, 7.5mm laminated, 2*3mm + 1*1.52mm  
#####  
[Glazing]  
  
[PI-curve]  
# pressure  
pressure = 2000 100 70 46 30 23 20 18 17.5 16.7 16  
  
# impulse  
impulse = 175 180 185 200 250 300 400 500 700 1000 10000
```

Figure 101: The ProSAir supplied `GlazingPI.prmat` glazing definition file

Within this file

- Any line starting with the `#` character is treated as a comment and ignored by ProSAir.
- Blank lines are ignored.
- The `[Glazing]` mark-up confirms to ProSAir that this is a glazing definition file and must be present before any other non-comments.
- The `[PI-curve]` mark-up indicates that any following data will define the points making up the PI iso-damage failure curve.
- The line starting `pressure =` defines 11 values of the pressure on the PI iso-damage failure curve and the values must be **strictly decreasing**. Pressures must be given in KPa.

- The line starting `impulse =` defines 11 values of the impulse on the PI iso-damage failure curve corresponding to the supplied pressure values and the impulse values must be **strictly increasing**.
- Both the pressure and impulse values **must** be supplied and the key words **pressure** and **impulse** must be followed by an `=` sign.
- There **must** be an **equal number** of pressure and impulse values supplied.
- There should be no trailing space after the final pressure or impulse value.

Using such a file we may define an arbitrary number,  $n_{PI}$  say, of points on the iso-damage failure curve denoted  $(I_j, P_j)_{j=1, \dots, n_{PI}}$ . For any given peak pressure-impulse pair  $(P, I)$  we determine if the window has failed by the following process:

1. If  $I < I_1$  or  $P < P_{n_{PI}}$  then window does not fail as it is less than the impulse or peak pressure asymptotes.
2. If  $I \geq I_{n_{PI}}$  then  $I$  is beyond the data range of the impulse values and we extrapolate the curve beyond  $I = I_{n_{PI}}$  giving window failure if  $P \geq P_{n_{PI}}$ .
3. If  $I \geq I_1$  then:
  - (a) Find  $j$  such that  $I_j \leq I < I_{j+1}$ .
  - (b) Linearly interpolate the data in  $(\log P, \log I)$  space to give failure if,

$$\log(P) \geq \log P_j + \frac{(\log P_{j+1} - \log P_j)(\log I - \log I_j)}{(\log I_{j+1} - \log I_j)}.$$

In this way `Glazing PI.prm` defines the PI iso-damage failure curve of Fig. 102.

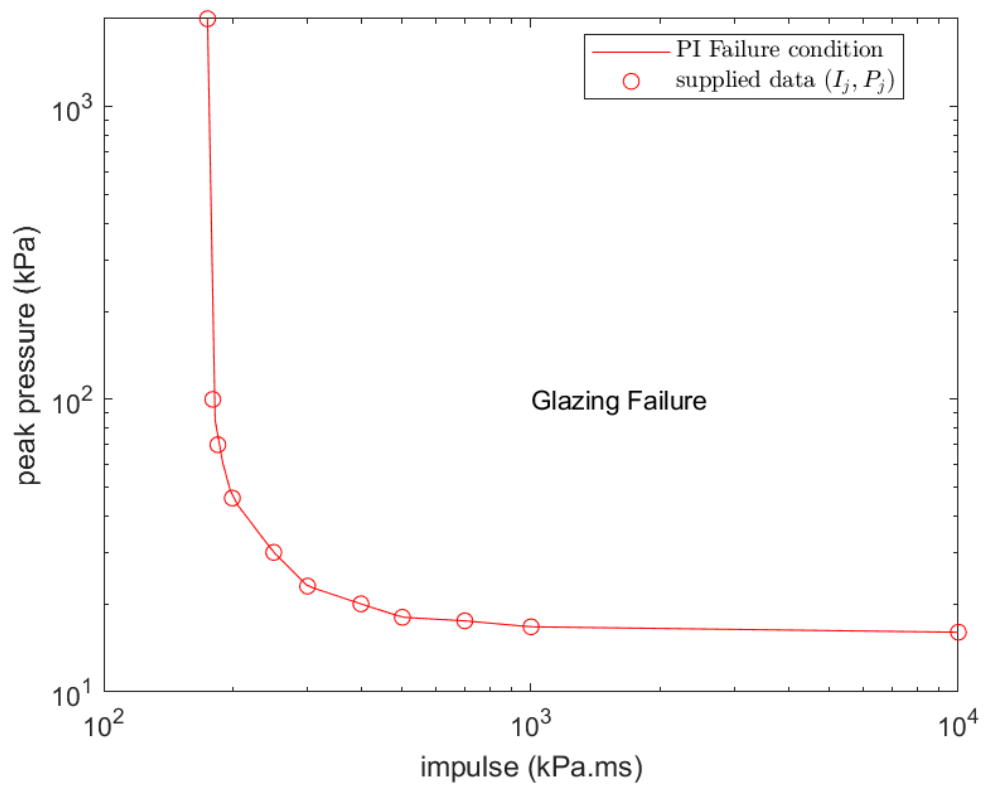


Figure 102: The iso-damage failure curve defined by the data of Glazing PI.prm of Fig. 101

### 6.2.2 PI-Asymptotes - Glazing defined by asymptotes to PI Curve

An example definition file is `Glazing Asymptotes.prm` of the `Materials\Glazing` folder supplied with ProSAir and shown in Fig. 103,

```
#####  
# Glazing asymptotes definition  
# Breaks safely curve  
# 1.25m * 1.55m, 7.5mm laminated, 2*3mm + 1*1.52mm  
#####  
[Glazing]  
  
[PI-asymptotes]  
# impulse (kPa.ms)  
minimum_impulse = 175  
  
# minimum pressure asymptote (kPa)  
minimum_pressure = 16
```

Figure 103: The ProSAir supplied `Glazing Asymptotes.prm` glazing definition file

Within this file

- Any line starting with the `#` character is treated as a comment and ignored by ProSAir.
- Blank lines are ignored.
- The `[Glazing]` mark-up confirms to ProSAir that this is a glazing definition file and must be present before any other non-comments.
- The `[PI-asymptotes]` mark-up indicates that any following data will define the points making up the glazing asymptotes iso-damage failure curve.
- The line `minimum_pressure = 16` defines the minimum the peak pressure for failure to be 16KPa.
- The line `minimum_impulse = 175` defines the minimum impulse for failure to be  $175\text{KPa} \cdot \text{ms} = 175\text{Pa} \cdot \text{s}$ .
- Both the minimum pressure and minimum impulse values **must** be supplied and the key words `minimum_pressure` and `minimum_impulse` must be followed by an `=` sign.

Using such a file we may define the minimum pressure and impulse pair ( $P_{\min}, I_{\min}$ ). For any given peak pressure-impulse pair ( $P, I$ ) we determine if the window has failed by the following process:

1. If  $I \geq I_{\min}$  and  $P > P_{\min}$  then window fails as both the minimum peak pressure and impulse have been exceeded.
2. Otherwise the window remains intact.

In this way `Glazing Asymptotes.prmat` defines the PI iso-damage failure curve of Fig. 104.

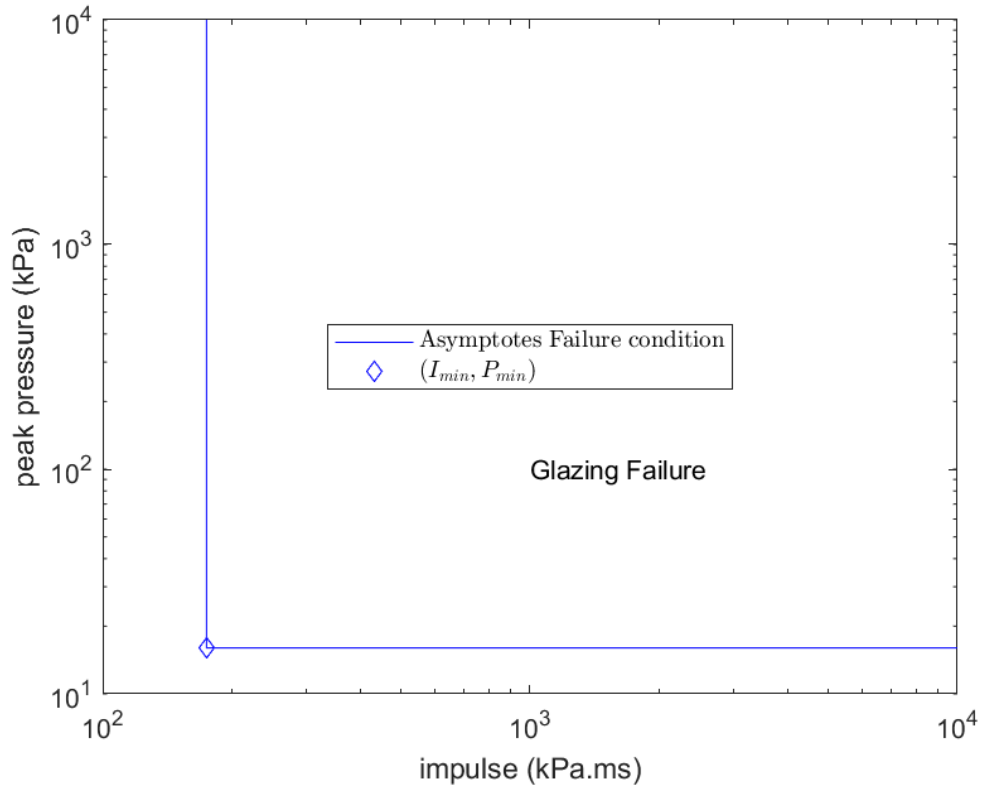


Figure 104: The iso-damage failure curve defined by the data of `Glazing Asymptotes.prmat` of Fig. 103

### 6.2.3 PI-Hyperbola - Glazing PI Curve defined by a hyperbola

An example definition file is Glazing Hyperbola.prmat of the Materials\Glazing folder supplied with ProSAir and shown in Fig. 105,

```
#####  
# Glazing asymptotes + point definition  
# Breaks safely curve  
# 1.25m * 1.55m, 7.5mm laminated, 2*3mm + 1*1.52mm  
#####  
[Glazing]  
  
[PI-hyperbola]  
  
# impulse (kPa.ms)  
minimum_impulse = 175  
  
# minimum pressure asymptote (kPa)  
minimum_pressure = 16  
  
# define one point  
pressure_point = 46  
impulse_point = 200
```

Figure 105: The ProSAir supplied Glazing Hyperbola.prmat glazing definition file

Within this file

- Any line starting with the # character is treated as a comment and ignored by ProSAir.
- Blank lines are ignored.
- The [Glazing] mark-up confirms to ProSAir that this is a glazing definition file and must be present before any other non-comments.
- The [PI-hyperbola] mark-up indicates that any following data will define a hyperbola defining the glazing iso-damage failure curve.
- The line `minimum_pressure = 16` defines the minimum peak pressure for failure to be 16KPa.
- The line `minimum_impulse = 175` defines the minimum impulse for failure to be  $175\text{KPa} \cdot \text{ms} = 175\text{Pa} \cdot \text{s}$ .

- The lines `pressure_point = 46` and `impulse_point = 200` define that the hyperbola must pass through the point  $(P, I) = (46\text{KPa}, 200\text{KPa} \cdot \text{ms})$ .
- All four of the `minimum_pressure`, `minimum_impulse`, `pressure_point`, and `impulse_point` values **must** be supplied and the key words `minimum_pressure`, `minimum_impulse`, `pressure_point`, and `impulse_point` must be followed by an = sign.

Using such a file we may define the minimum pressure and impulse pair  $(P_{\min}, I_{\min})$  and the supplied point  $(I^*, P^*)$ . For any given peak pressure-impulse pair  $(P, I)$  we determine if the window has failed by the following process:

1. If  $I \geq I_{\min}$  and  $P > P_{\min}$  then if,

$$P \geq P_{\min} + \frac{(P^* - P_{\min}) * (I^* - I_{\min})}{(I - I_{\min})}$$

the window fails as  $P$  is above the hyperbola defined by the asymptotes and which passes through the supplied point.

2. Otherwise the window remains intact.

In this way `Glazing Hyperbola.prmat` defines the PI iso-damage failure curve of Fig. 106.

#### 6.2.4 NoFailure - Glazing with no failure

An example definition file is `Glazing NoFailure.prmat` of the `Materials\Glazing` folder supplied with ProSAir and shown in Fig. 107,

Within this file

- The `[Glazing]` mark-up confirms to ProSAir that this is a glazing definition file and must be present before any other non-comments.
- The `[NoFailure]` mark-up indicates this is a no failure model so no further data is needed.

Using such a file we can get obtain the P-I loading on a window so a suitable glass type may be chosen to resist the loading, or the loading may be used in an external tool.

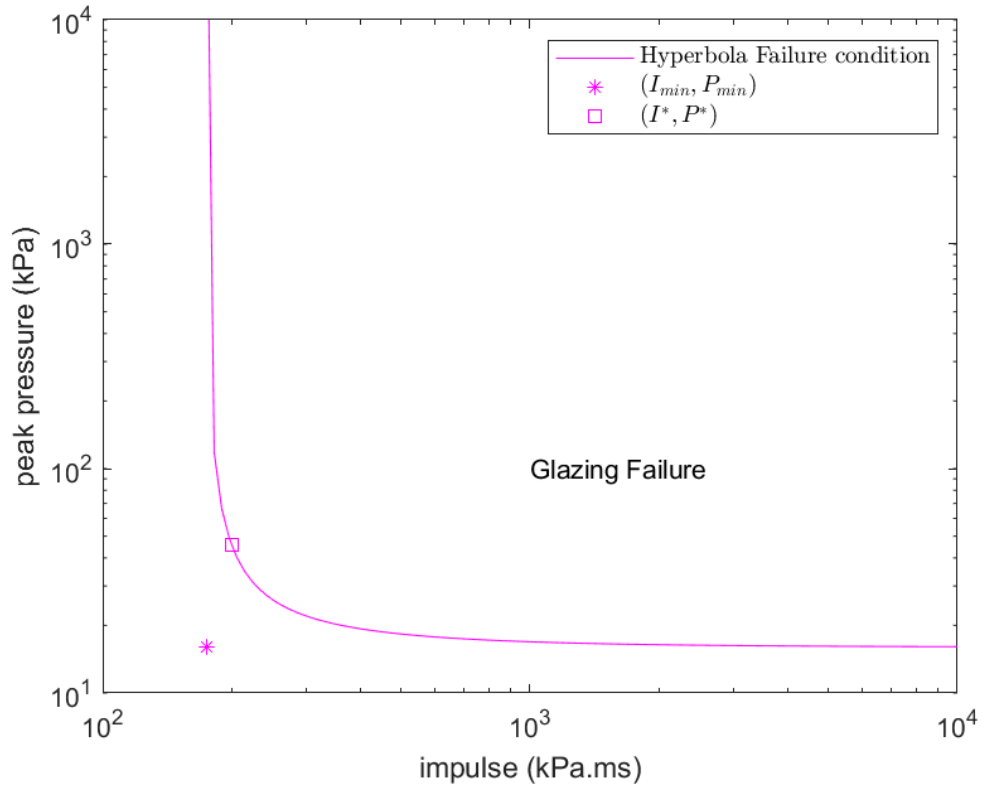


Figure 106: The iso-damage failure curve defined by the data of Glazing Hyperbola.prmat of Fig. 105

```
#####
# Glazing P-I curve definition
# Glazing model with no failure
#####
[Glazing]

[NoFailure]
```

Figure 107: The ProSAir supplied Glazing NoFailure.prmat glazing definition file

### 6.3 Glazing Failure Model

The glazing failure model involves determining the spatially averaged pressure loading and impulse on the window and deeming the window to fail when the peak spatially averaged pressure loading and its impulse exceed the specified iso-damage failure curve. Each time the loading changes phase, e.g., from negative to positive loading, this process restarts.

- Initialise window phase peak pressure  $P_{phase} = 0$  and phase impulse  $I_{phase} = 0$ .
- For each time step  $n$  at time  $t^n$ 
  1. For each **Solid** cell  $i$  of the  $N_w$  within the window obstacle (see Fig. 100) is visited. The pressure in the adjacent cell in the direction given by the **Orientation** is determined and from it is subtracted the pressure in the adjacent cell in the opposite direction - this give the **pressure difference**  $\Delta P_i^n$  across cell  $i$  in the window at time step  $n$ .
  2. The pressure differences across every cell in the window are averaged to give an **average pressure difference**  $\Delta P^n$ ,

$$\Delta P^n = \frac{1}{N_w} \sum_{i=1}^{N_w} \Delta P_i^n,$$

across the window for that time step.

3. If  $\Delta P_i^n$  changes sign then the phase has changes and we set  $P_{phase} = I_{phase} = 0$ .
4. The pressure difference at the previous time-step and the time step size are used to update the impulse  $I_{phase}$  using a trapezium rule integration,

$$I_{phase} = I_{phase} + \frac{1}{2}(t^n - t^{n-1})(\Delta P^n + \Delta P^{n-1}).$$

This rule is adapted if the loading has just changed phase to,

$$I_{phase} = \frac{1}{2}(t^n - t^*)\Delta P^n.$$

where  $t^*$  is the zero of the linear interpolation between  $(t^{n-1}, P^{n-1})$  and  $(t^n, P^n)$ .

5. If  $|\Delta P^n| > |P_{phase}|$  the peak pressure is updated  $P_{phase} = \Delta P^n$ .
6. Check if  $(P_{phase}, I_{phase})$  causes window failure (note if  $P_{phase} < 0$  then  $-P_{phase}$  and  $-I_{phase}$  are used in the check) using the appropriate failure model of Sec 6.2.
7. If failure has occurred then the window is said to **shatter** and:
  - For each **Solid** cell in the window change its type to **Air** and initialise with the averaged conservative variables from its neighbours in the **Orientation**, and its opposite, directions.
  - We stop integration of window pressures.
8. Return to step 1.

### 6.3.1 Differences compared to Air3d

This glazing failure model is similar to that used in `Air3d` [?] except:

- We use the peak of the spatially averaged pressure loading whereas `Air3d` uses the peak value in both space and time.
- Our algorithm allows for negative loading, so if  $(-I, -P)$  exceed the iso-damage failure curve the window also fails.

### 6.3.2 Criticisms of the Glazing Model

The glazing model may be criticised for several reasons:

1. In the present model the window shatters immediately once the iso-damage failure curve is passed. In most cases this will not correspond to physical reality since the blast loading, being impulsive, gives sufficient momentum to the window to ensure that it will fail some short time in the future. During this delay pressure and impulse will continue to build up giving momentum to the window fragments so as, when the window does shatter, the fragments are thrown. As a result of this windows will fail earlier than might be expected in reality. This might cause unexpected relief of loading on adjacent windows and differing loads behind the shattered window. Consequently, results from calculation with arrays of windows should be treat with caution.
2. We would like to be able to predict the glazing as breaking safely, or projecting low hazard, or high hazard, fragments [?]. This is not possible with the present algorithm which merely shatters the window once the breaking safely iso-damage curve is passed as described in 1.
3. The author has heard glazing experts claim that a 10% variation in iso-damage criteria for ostensibly identical glazing units is to be expected. This clearly impacts on how well a simulation might model a particular experiment.
4. The present model behaves symmetrically under negative loading - this is desirable for modelling complex configurations in which one cannot determine a priori whether loading would be from the front or back of the glazing unit. In practice one might expect glazing units to behave asymmetrically, particularly if the glazing unit fails due to the glazing being ripped from its frame.

## 6.4 Example - Glazed Cubicle

The *Glazing Hazard Guide* [?, ?, ?, ?] contains tables of charge weights and stand-off distances corresponding to threshold levels of damage to glazing elements tested in a cubicle structure with frontal dimensions  $2.15\text{m} \times 2.15\text{m}$ .

As an appropriate example, a single laminated pane 7.5mm thick and  $1.25\text{m} \times 1.55\text{m}$  was selected. Johnson [?] indicates that 4.536kg of TNT at 8m stand-off, detonated 1.0m above the ground, will cause a *low hazard* (as defined by the Glazing Hazards Guide [?]-[?]). Johnson [?] further indicates that, although 4.536kg is sufficient to cause breakage, 3.0kg, at the same stand-off, is not. The geometric layout is depicted in Fig. 108 for the 3kg charge.

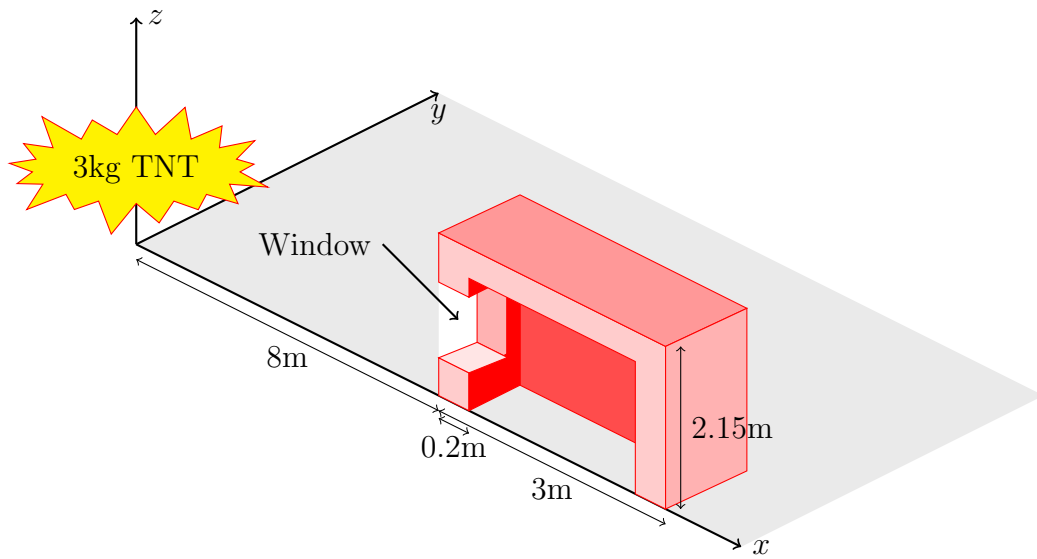


Figure 108: The Glazed Cubicle geometry for the examples of Sec. 6.4 and for  $y \geq 0$ .

### 6.4.1 3kg at 8m Stand-off

An appropriate input file for this computation is `GlazedCubicle3kg.pr1`.

The **General Settings** tab for this computation is shown in Fig. 109.

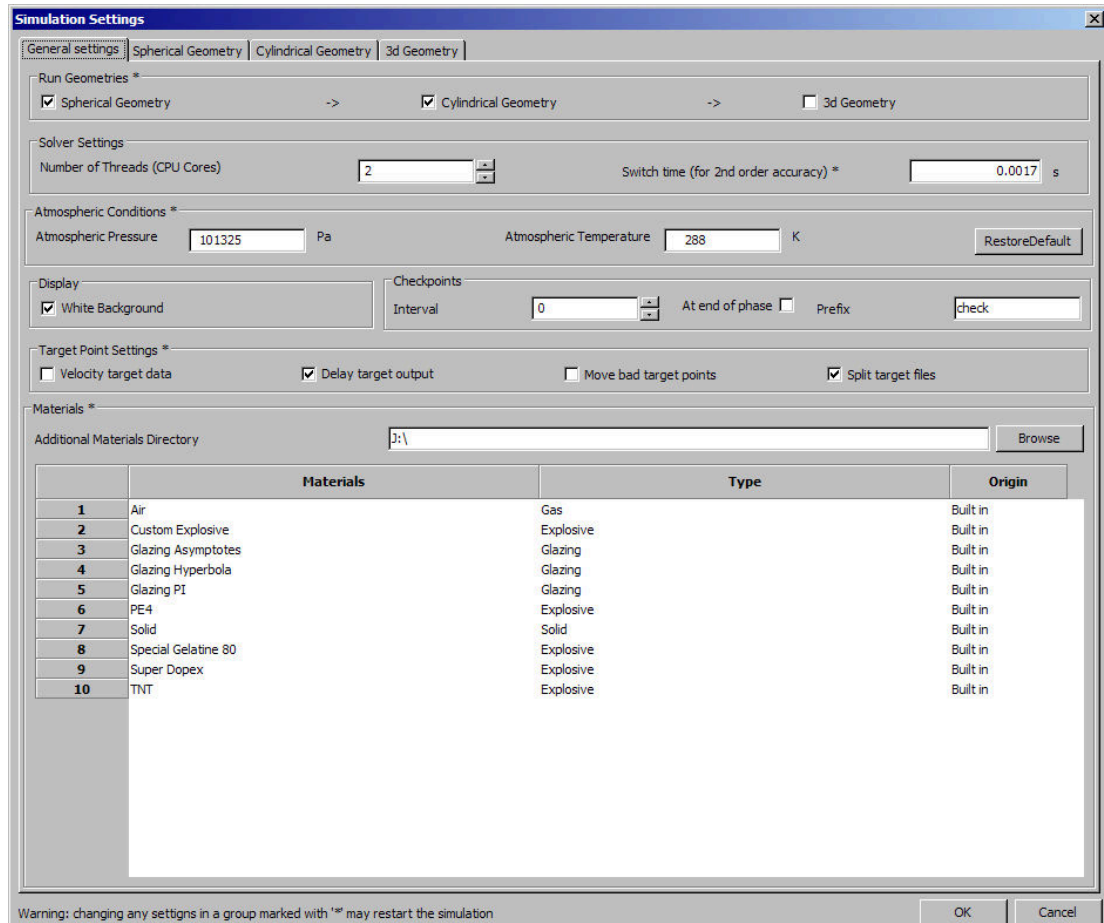


Figure 109: **General Settings** tab for the Glazed Cubicle at 3kg stand-off example of Sec. 6.4.1

Note:

- We have adjusted the **Switch time** to be appropriate for 3kg of TNT by multiplying the default 1kg value of 0.0012s by  $\sqrt[3]{3}$  to give 0.0017s.
- Within the **Materials** pane we can see the materials type **Glazing PI**. This glazing material definition is supplied with ProSAir as described in Sec. 6.2.1 and was written specifically to correspond to the single laminated pane 7.5mm thick and 1.25m × 1.55m of this problem.

- All 3 **Run Geometries** have been selected to allow for the spherical charge's shock wave to be modelled with spherical symmetry until it reaches the ground, from that point cylindrical symmetry will be used until the shock reaches the cubicle at 8m stand-off whereupon a 3D simulation will be started.

The **Spherical Geometry** tab for this computation is shown in Fig. 110.

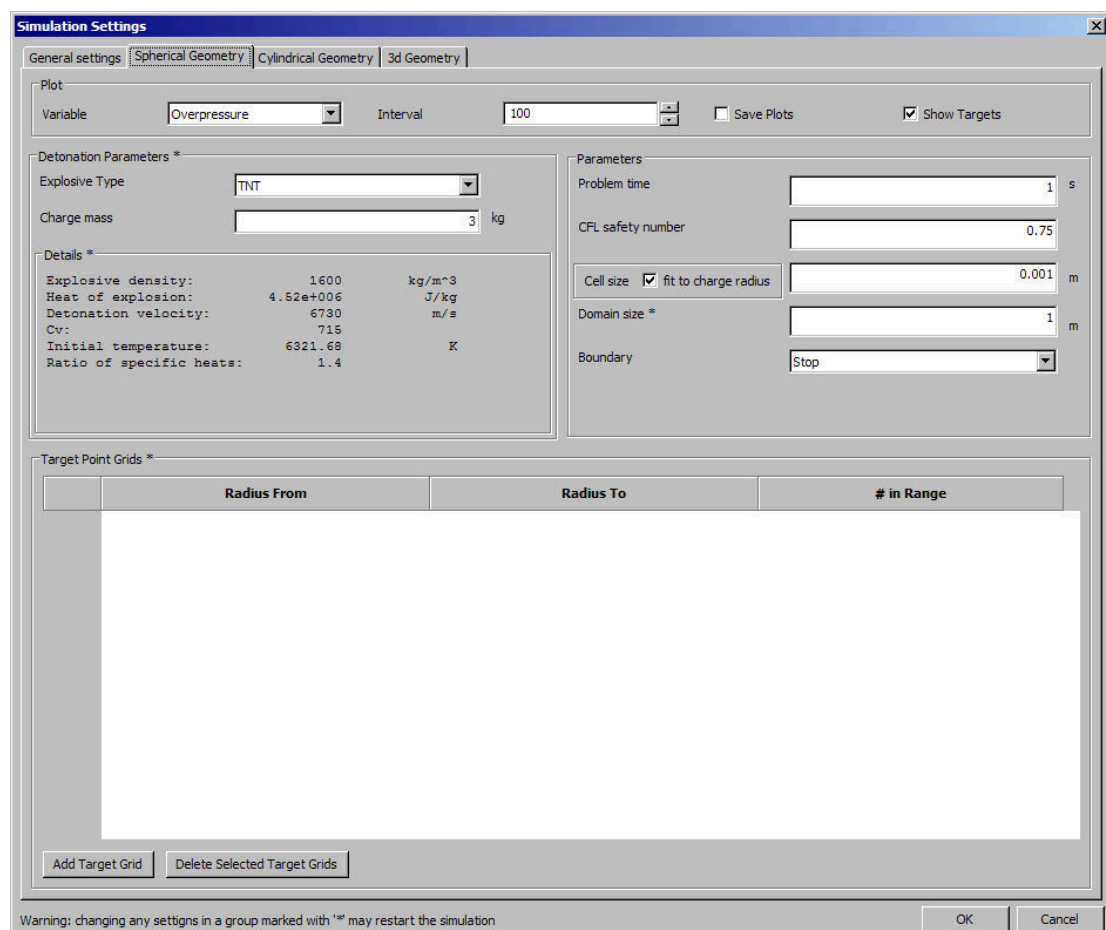


Figure 110: **Spherical Geometry** tab for the Glazed Cubicle at 3kg stand-off example of Sec. 6.4.1

Note:

- We set the **Domain size** to be 1m - the height of detonation of the charge.
- A **Stop Boundary** condition is used so the 1D spherically symmetric simulation will cease as the shock reaches the ground.

The **Cylindrical Geometry** tab for this computation is shown in Fig. 111.

**Simulation Settings**

General settings | Spherical Geometry | **Cylindrical Geometry** | 3d Geometry

**Plot**

Variable: Overpressure Type: contour plot Interval: 100

Planes (r, h): 0, 0 m Scale factor: 1 ☐ Save Plots ☒ Show targets

**Parameters**

Problem time: 1 s Height of blast \*: 1 m

Cell size \*: 0.025 m CFL safety number: 0.5

Domain size (r, h) \*: 8 m 5 m Boundary types (r, h) \*: Stop Transmissive

**Obstacles \***

| Type | Geometry Data | Material |
|------|---------------|----------|
|      |               |          |

Add New Delete Selected Move Up Move Down

**Target Point Grids \***

| Radius From | Radius To | # in Range | Height From | Height To | # in Range |
|-------------|-----------|------------|-------------|-----------|------------|
|             |           |            |             |           |            |

Add Target Grid Delete Selected Target Grids

Warning: changing any settings in a group marked with \* may restart the simulation

OK Cancel

Figure 111: **Cylindrical Geometry** tab for the Glazed Cubicle at 3kg stand-off example of Sec. 6.4.1

Note:

- The **Height of blast** is set to 1m - the height of detonation of the charge.
- The **Domain Size** is set to 8m in the radial direction so the cylindrically symmetric shock can be modelled until just before it impacts the cubicle. The height of 5m is set significantly higher than the cubicle's height to allow for further 3D modelling and relief over the top of the cubicle.
- A **Stop Boundary type** condition is used on the outer **r** radial boundary so the 2D cylindrically symmetric simulation will stop as the shock reaches a radius of 8m and is about to impact on the cubicle.

The **3d Geometry** tab for this computation is shown in Fig. 112.

Note:

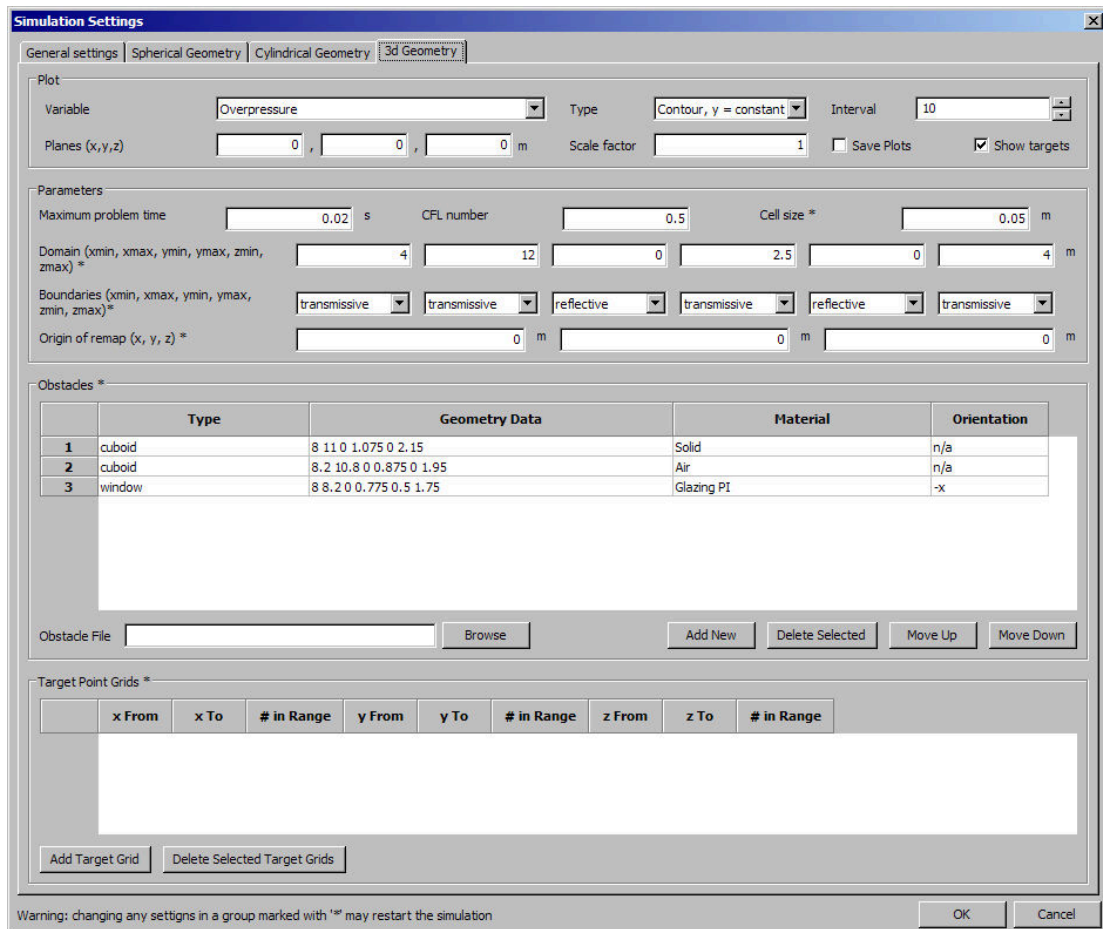


Figure 112: 3d Geometry tab for the Glazed Cubicle at 3kg stand-off example of Sec. 6.4.1

- We have opted to **Save Plots** each time the graphics display is updated at an **Interval** of 10 time steps.
- Simulation will stop at a **Maximum problem time** of 0.02s.
- The **Domain** has been set to run from 4m to 12m in the  $x$ -direction. This does not include the detonation site but should still accurately model the passage of the shock over the cubicle. We model from 0m to 2.5m in  $y$  going from the plane of symmetry to beyond the edge of the cubicle at 0.875m to allow for clearing effects. We model from the ground at 0m to a height of 4m in  $z$  - this is lower than in the cylindrically symmetric phase and should allow for clearing.
- **Boundaries** are set to give **reflective** conditions on the  $y = 0$  **ymin** symmetry plane and the  $z = 0$  **zmin** ground boundaries. Others are left as **Transmissive** to let the shocks pass out of the domain.

- The geometry has been modelled as a  $3 \times 1.1075 \times 2.15$  solid cuboid hollowed with a cuboid of air to give walls of 0.2m. The window element occupies the full 0.2m thickness of the front wall at  $x = 8\text{m}$  and has the appropriate width and height (half the window width as we are using symmetry).

On running the simulation we see that the shock clearly impacts on the window (Fig. 113) but the window remains intact (Fig. 114)

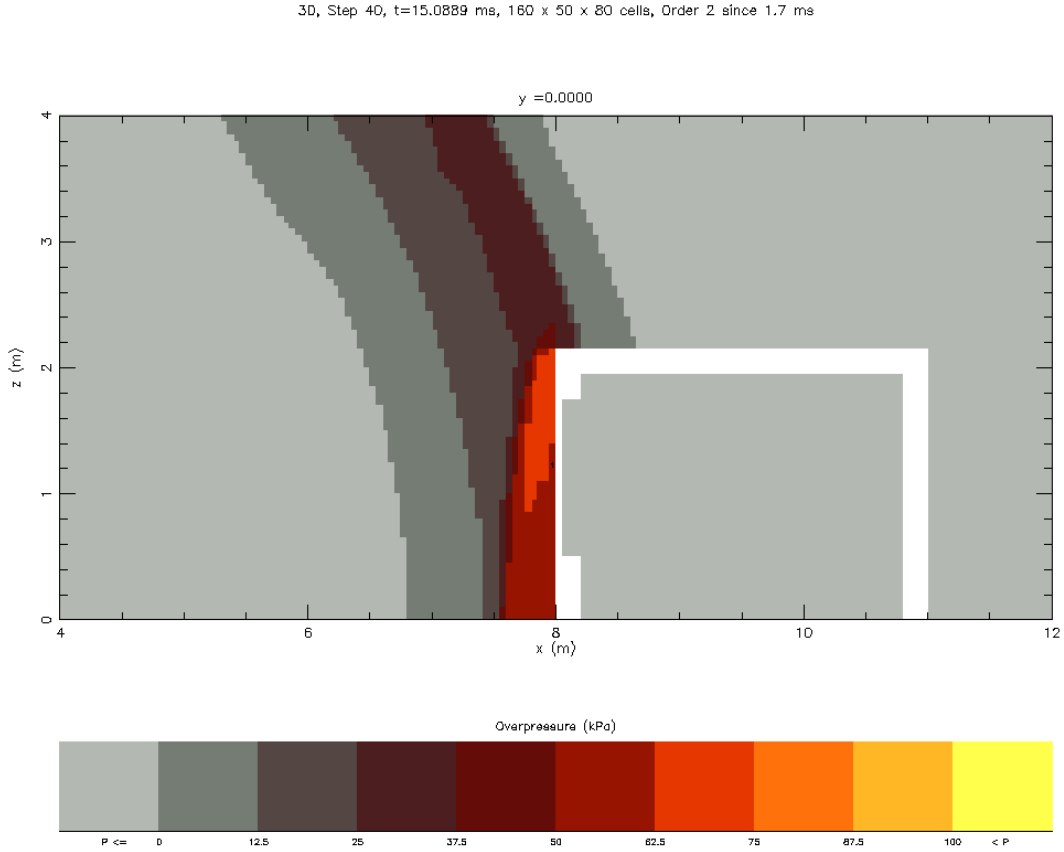


Figure 113: Overpressure field at 15ms for the Glazed Cubicle at 3kg stand-off example of Sec. 6.4.1

At the end of the simulation, in addition to the usual target data files, files containing window data are also produced. In this example there are two such files:

- **wn\_3d\_1.csv** contains the time history of the load information on the window. The first few and last few lines of this file are shown in Fig. 115. The columns give, as defined in Sec. 6.3,:
- **Time (ms)**: time  $t^n$ ,

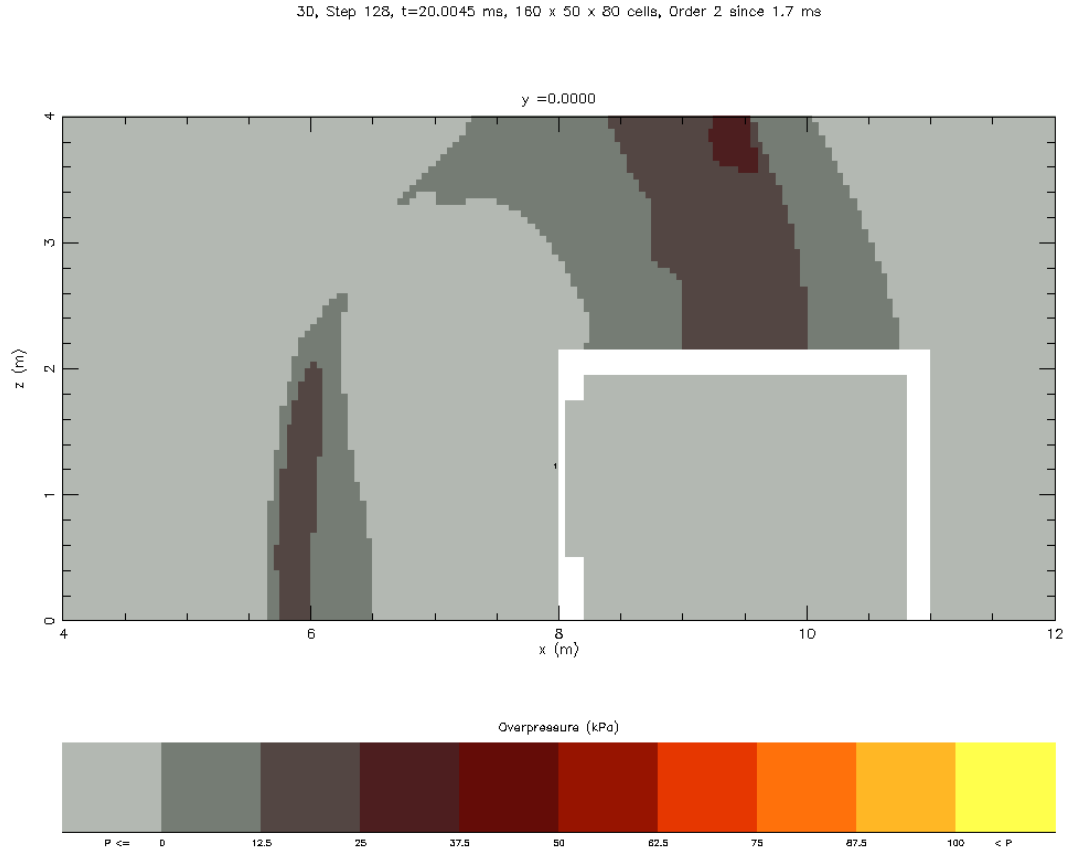


Figure 114: Overpressure field at 20ms for the Glazed Cubicle at 3kg stand-off example of Sec. 6.4.1

- **Pressure (kPa)**: average pressure difference  $\Delta P^n$ ,
  - **Impulse (kPa-msec)**: cumulative impulse  $I^n$  since the start of the computation,
 
$$I^n \approx \int_{t=0}^{t^n} \Delta P(t) dt,$$
  - **Phase Peak Pressure(kPa)** the peak average pressure difference in current phase  $P_{phase}$ ,
  - **Phase Impulse (kPa-msec)** the impulse obtained by integrating the pressure difference over the current phase  $I_{phase}$ .
- **window\_summary\_3d.csv**, as shown in Figure 6.4.1, contains summary information the all the windows in a calculation. The columns give:
    - **Nr**: the window number.

```

Window 1
Time (ms), Pressure (kPa), Impulse (kPa-msec), Phase Peak Pressure(kPa), Phase Impuls
12.922, -3.49E-16, -2.26E-15, 0, 0
12.9781, 9.00E-07, 2.53E-08, 0, 0
13.0343, 6.25E-06, 2.26E-07, 0, 0
13.0905, 2.81E-05, 1.19E-06, 0, 0
13.1466, 0.000106217, 4.96E-06, 0, 0
13.2027, 0.000380132, 1.86E-05, 0, 0
13.2589, 0.00131177, 6.61E-05, 0.00131177, 6.61E-05
13.315, 0.00433057, 0.000224634, 0.00433057, 0.000224634
.
.
.
19.9474, -19.5392, 133.458, -19.5392, -24.4732
20.0045, -19.752, 132.337, -19.752, -25.5941

```

Figure 115: Window time history file `wn_3d_1.csv` for the Glazed Cubicle at 3kg stand-off example of Sec. 6.4.1. Extra spaces have been added for clarity

```

Nr, peak impulse, peak pressure, broken (0/1=no/yes), break time (ms)
1, -25.5903, -19.7512, 0, 0

```

Figure 116: Window summary file `window_summary_3d.csv` for the Glazed Cubicle at 3kg stand-off example of Sec. 6.4.1. Extra spaces have been added, and units in header line removed, for clarity

- **peak impulse (kPa-msec)**: the peak impulse recorded for that window in the final loading phase (KPa · ms).
- **peak pressure (kPa)** the peak pressure recorded for that window in the final loading phase (KPa).
- **broken**: a **1** indicates the window broke and a **0** that it did not. Summing this column gives the total number of windows that failed.
- **break time**: if the window broke then the time at which the failure criteria was met is recorded (ms). If the window did not break then a **0** is recorded.

### 6.4.2 4.536kg at 8m Stand-off

We now repeat the calculation of Sec. 6.4.1 using the 4.536kg charge that Johnson [?] indicates causes failure. An appropriate input file for this computation is `GlazedCubicle4p536kg.pr1` - this differs from `GlazedCubicle3kg.pr1` only in the charge mass and switch time. As Fig. 117 in this case the window does break shortly before 15.6ms.

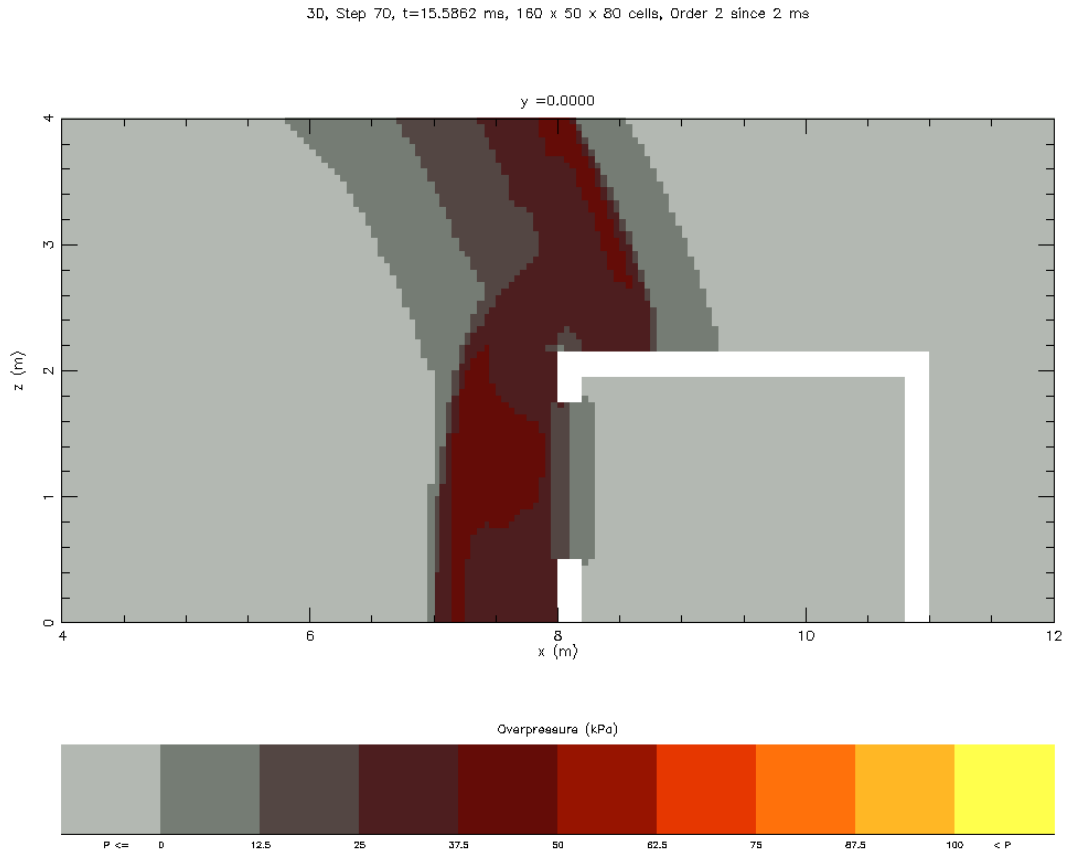


Figure 117: Overpressure field at 15.6ms for the Glazed Cubicle with 4.536kg charge example of Sec. 6.4.2

In Fig. 118 we show the Glazing failure P-I condition and loading for the 3kg and 4.536kg charges. We clearly see that the large charge imparts sufficient impulse to cause failure as suggested by experiment [?].

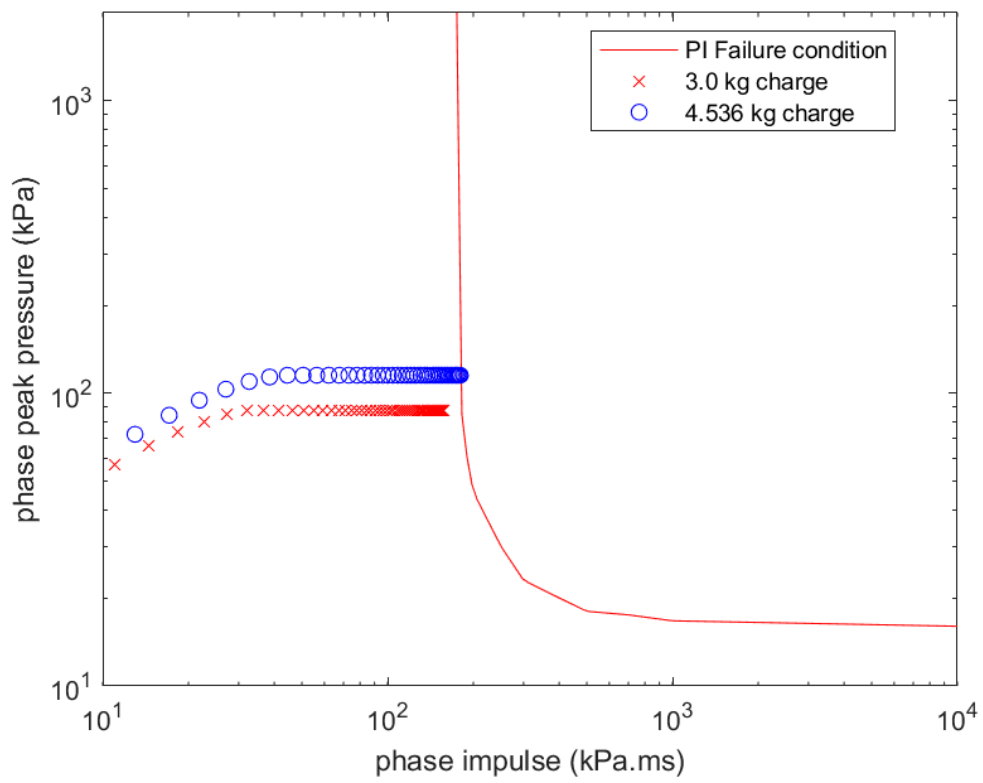


Figure 118: Glazing failure P-I condition and loading for 3kg and 4.536kg charges for the Glazed Cubicle example of Secs. 6.4.1 and 6.4.2

## 7 Practical Issues

In this section we cover various practical issues associated with utilising ProSAir.

### 7.1 Command Line Arguments

ProSAir may be run from the command line of a Windows command window or from within a batch file (see Section 7.2). There are four variants on how this might be done which we illustrate assuming a command window has already been opened.

1. We may start ProSAir from a command prompt using the absolute path of the executable. e.g. for the 32 bit single precision variant we would enter

```
"C:\Program Files\ProSAir32-2012-3\ProSAir.exe"
```

at the command line. Note the use of double quotes required due to the space in the path. This starts the ProSAir GUI and ProSAir may then be run as normal from then on. The default absolute paths for all variants of ProSAir may be found in Section 1.2.

2. We may start from the command prompt and load a ProSAir input file. e.g. assuming the ProSAir input file `Example.pr1` is stored in the windows folder `C:\temp` then typing

```
"C:\Program Files\ProSAir32-2012-3\ProSAir.exe" C:\temp\Example.pr1
```

starts ProSAir running, loads in the data from `Example.pr1`. If the simulation is now started (see Section 2.4) then output files are placed in the folder `C:\temp`.

3. We may start ProSAir from the command line, load in a file and run by adding the keyword `run` after the file name on the command line, e.g.,

```
"C:\Program Files\ProSAir32-2012-3\ProSAir.exe" C:\temp\Example.pr1 run
```

Once the simulation has finished ProSAir requires user intervention via the GUI to perform further tasks (e.g., read and process a further file, or exit).

4. Finally we may specify that ProSAir exits on completing the simulation by adding the keyword `exit` to the command line. e.g.

```
"C:\Program Files\ProSAir32-2012-3\ProSAir.exe" C:\temp\Example.pr1 run exit
```

Note that the order of the command line arguments is important, as is their position. The format of the command line instructions must be of the form

```
<ProSAir> <file> run exit
```

where

- **<ProSAir>** is the absolute path of a ProSAir executable.
- **<file>** is the absolute path to a ProSAir input file and must be the first argument supplied to ProSAir.
- **run** must be the second argument and makes no sense if argument **<file>** is not supplied.
- **exit** must be the third argument and makes no sense if arguments **<file>** and **run** are not supplied.

## 7.2 Batch Processing

It is frequently required to run a series of simulations one after another. This might be in order to perform a mesh refinement study, or perhaps to perform a parametric study of blast loading with stand-off distance. It might also be that these simulations need to be performed unsupervised, perhaps overnight, and so as one simulation finishes another must start automatically. Such a series of simulations may be performed using a [Windows batch file](#).

### 7.2.1 Simple Batch Example

Consider running the two simulations of Sec. 3.1 for the Kingery Bulmash problem with two cell sizes. We assume two subfolders have been created within a common parent folder:

- Subfolder **Initial** containing ProSAir input file **KingeryBulmashInitial.pr1**.
- Subfolder **Refined** containing **KingeryBulmashRefined.pr1**.

We require that ProSAir simulations be run using these two input files one after the other.

Within the parent folder we have a batch file **KingeryBulmash64.bat** shown in Figure 119. When this file is executed the Windows operating systems will start a command window in which will be executed each non-blank line with the exception of those starting with **rem** which are comments (or reminders) and are ignored. The executable lines perform the following

- The line

```
echo off
```

Turns off echoing of all the commands to the display.

- The line

```

rem Runs Kingery Bulmash examples

echo off

rem set variable for ProSAir executable
rem set PROSAIR="D:\ProSAir\trunk\x64\Release\ProSAir.exe"
set PROSAIR="C:\Program Files (x86)\ProSAir64-2022-11\ProSAir.exe"
echo ProSAir executable =%PROSAIR%

rem Initial configuration
set FILE="%cd%\Initial\KingeryBulmashInitial.pr1"
echo Running with input file FILE=%FILE%
%PROSAIR% %FILE% run exit
echo completed Initial

rem Refined configuration
set FILE="%cd%\Refined\KingeryBulmashRefined.pr1"
echo Running with input file FILE=%FILE%
%PROSAIR% %FILE% run exit
echo completed Refined

```

Figure 119: Batch file KingeryBulmash64.bat

```
set PROSAIR="C:\Program Files (x86)\ProSAir64-2022-11\ProSAir.exe"
```

sets a batch variable named **PROSAIR** to the string "C:\Program Files (x86)\ProSAir64-2022-11\ProSAir.exe" - note the use of double quotes around the folder path which are necessary due to the space in the string. Within our batch file this variable holds the absolute path to the ProSAir executable.

- The line

```
echo ProSAir executable =%PROSAIR%
```

writes the string **ProSAir executable =** followed by the value of the variable **PROSAIR** - note the use of % either side of the variable name to force its value to be used instead of the string "PROSAIR". Consequently this line will produce output

```
ProSAir executable = C:\Program Files (x86)\ProSAir64-2022-11\ProSAir.exe
```

to the Windows command window when the batch file is run.

- The line

```
set FILE=%cd%\Initial\KingeryBulmashInitial.pr1
```

uses the value of the Windows environment variable `%cd%` to determine the path to the current folder in which the batch file is running. This path is then appended with the first subfolder name and the ProSAir input file name and the combined string is assigned to the batch file variable `FILE`. The next line `echo` (i.e., writes) the variable `FILE` to the command window.

- The line

```
%PROSAIR% %FILE% run exit
```

expands the value of the `PROSAIR` and the `FILE` variable, then has two further strings `run` and `exit`. This line, like the other non-comment lines, will treat as a command and results in the 32 bit single precision variant of ProSAir being launched with three command line arguments

1. The expansion of `FILE` giving the required input file name.
2. `run` - this causes ProSAir to start running once the input file has been read.
3. `exit` - this causes ProSAir to exit once the simulation is completed.

For details see Section 7.1.

- Once execution is completed the line

```
completed Initial
```

will be echoed to the command line.

- The following lines repeat the process for the second input file.

To start execution the user should open the folder using Windows Explorer and double click on the batch file's icon. This results in a command shell window opening and executing the batch file's commands, as seen in Figure 120, launching ProSAir to perform the two required computations. Note the window's contents will vary depending on the location of the ProSAir input files.

The batch file may be edited using a text editor (e.g., Notepad, Wordpad, XEmacs) but remember to open the editor and then read in the batch file (if you double click on the batch file's icon to open it you will execute the batch file).

```
C:\WINDOWS\system32\cmd.exe
C:\ShaunsDocuments\prosair\branches\2011.5a\Documentation\ProSAirUserGuide\KingeryBulmash>rem Runs Kingery Bulmash examples
C:\ShaunsDocuments\prosair\branches\2011.5a\Documentation\ProSAirUserGuide\KingeryBulmash>ProSAir executable = "C:\Program Files\ProSAir32\ProSAir.exe"
Running with input file FILE=C:\ShaunsDocuments\prosair\branches\2011.5a\Documentation\ProSAirUserGuide\KingeryBulmash\Initial\KingeryBulmashInitial.prl
completed Initial
Running with input file FILE=C:\ShaunsDocuments\prosair\branches\2011.5a\Documentation\ProSAirUserGuide\KingeryBulmash\Refined\KingeryBulmashRefined.prl
```

Figure 120: The command shell window part way through the second computation of batch file KingeryBulmash.bat of Figure 119

## 8 Algorithms

This section summarises some of the algorithms used in ProSAir. Due to the mathematical notation used it should be viewed in the pdf version of the manual. Algorithms derived from those of Air3D may be found in the Air3d version 9 user guide[?].

### 8.1 Automatic adjustment of cell size

In the Kingery and Bulmash example 3.1.4 we stated that the use of the **fit to charge radius** option permits ProSAir to adjust the **Cell size** to ensure that a spherical charge's surface exactly aligns with a cell boundary. The algorithm for performing this is straightforward and seeks to reduce the cell size to exactly match the charge's radius. It does this by dividing the charge radius by the current, nominal, cell size and rounding this value up to the nearest integer to give the number of cells in the charge which will ensure the cell size is smaller than the nominal and a cell boundary lies on the charge radius. It must also allow for the given cell spacing satisfying this requirement subject to floating point round-off. Pseudocode for this is given in Algorithm 1.

```
Data: Explosive radius  $r_e$ , nominal Cell size  $\Delta r_{nom}$ , relative tolerance Tol
Result: Required cell size  $\Delta r$  such that  $r_e/\Delta r$  is an integer
double  $X_{cells} = r_e/\Delta r_{nom}$  ;
int  $N_{ceil} = \text{ceiling}(X_{cells})$  ;           // round up to guarantee  $\Delta r \geq \Delta r_{nom}$ 
int  $N_{floor} = \text{floor}(X_{cells})$  ;           // round down
int  $N_{cells} = N_{ceil}$  ;                     // use rounded up by default
/* See if  $X_{cells}$  is within round-off of  $N_{floor}$  */
if  $X_{cells} - N_{floor} < Tol * X_{cells}$  then
    |  $N_{cells} = N_{floor}$ ;
end
 $\Delta r = r_e/N_{cells}$ ;
```

**Algorithm 1:** Automatic adjustment of cell size

### 8.2 Detection of shock-front domain

ProSAir calculates updates to the flow field in strips. For each strip in a given grid direction ProSAir starts at the out edges of the domain and works inwards looking for:

- A solid cell
- Or a cell with all the following satisfied:
  - Momentum in the given direction less than a tolerance.
  - Difference in energy and density with adjacent cells less than the tolerance.

In such cases the cell is not regarded as part of the current flow regime and the next cell in the given direction is considered. The first cell met that does not satisfy the given conditions is regarded as giving the limit for flow-field computation in that direction.

## 9 Air3d Compatibility

ProSAir is designed to be compatible with input files created for **Air3d** version 9 [?]. Due to differences in compiler optimisations and some small algorithmic differences results will not be exactly identical. It is intended that users should read in an **Air3d** input file using ProSAir then save the computation details to a ProSAir input file. Alternatively the **Air3d** input file may be used directly to perform the computation in ProSAir. We recommend that users convert their input files to the ProSAir input format. All support for **Air3d** input files is on a best endeavours basis, may be withdrawn in the near future, and new ProSAir features may not be available when using this format.

In §9.1-9.4 we present *standard test cases* from the **Air3d** user guide [?].

### 9.1 Example Two: Height of Burst

This example forms the ProSAir *Height of Burst* example of §3.2 but here performed from the **Air3d** input file and compared to **Air3d** generated data. The **Air3d** input file is available as `0_example_02.txt` and may be downloaded and viewed from the html version of this document. For a description of the input file see §4.1 of the **Air3d** user guide [?].

Figures 121 to 123 compare the pressure histories computed from **Air3d** with those computed by ProSAir from the same input file `0_example_02.txt` at 3 of the 25 target points. It may be seen that the two solvers are giving very similar results for this

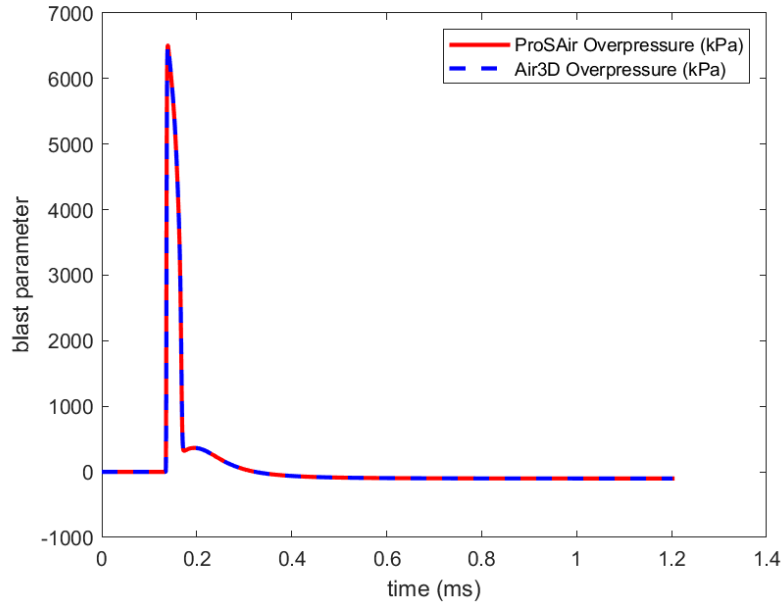


Figure 121: Overpressure history at the spherical phase target point at  $R = .5\text{m}$  for the **Air3d** compatibility Height of Burst case of Section 9.1

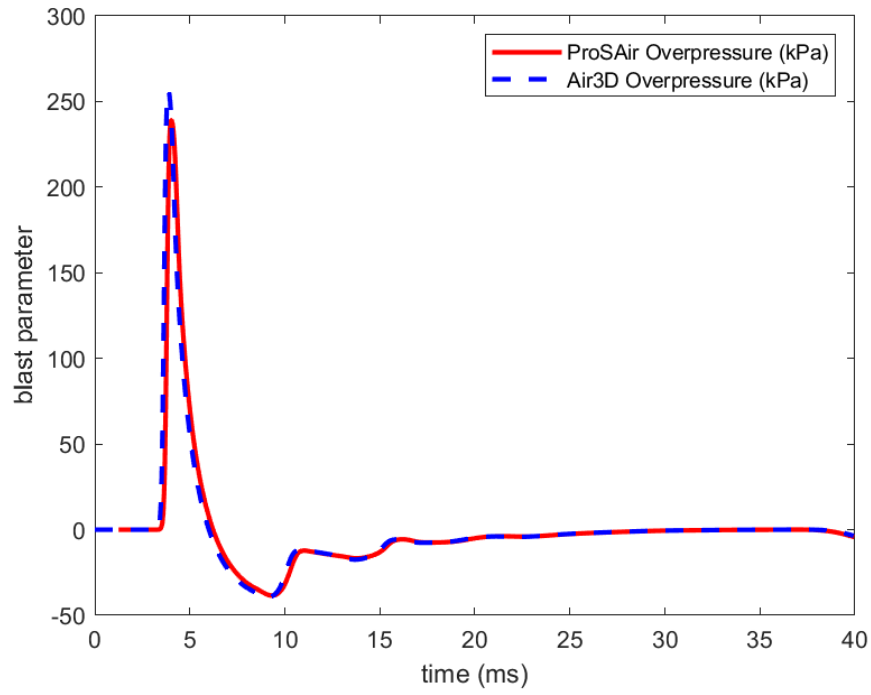


Figure 122: Overpressure history at the first cylindrical phase target point at  $(r, h) = (3, 0)$ m for the Air3d compatibility Height of Burst case of Section 9.1

computation - the small differences are likely due to variations in round-off errors.

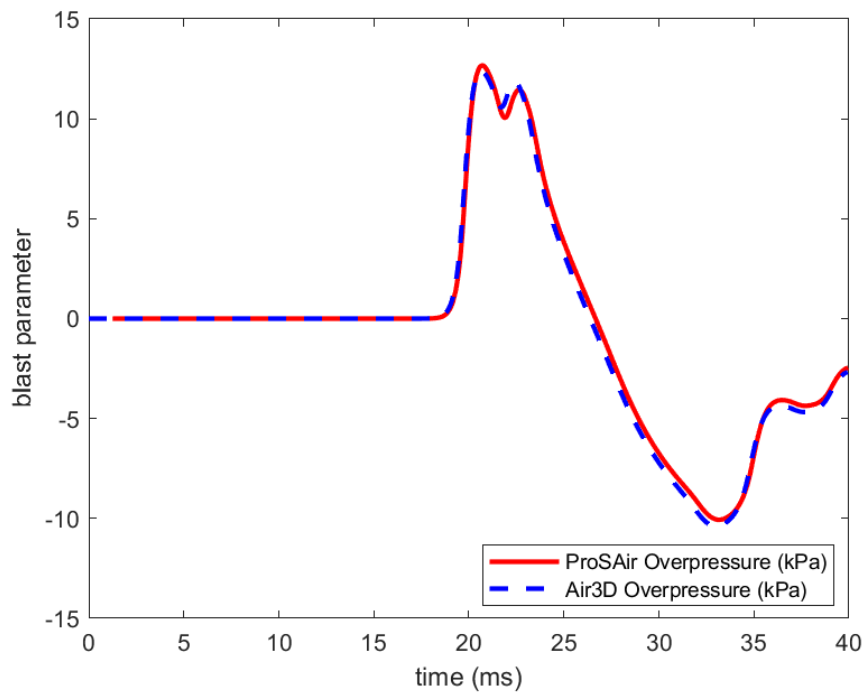


Figure 123: Overpressure history at the 24<sup>th</sup> cylindrical phase target point at  $(r, h) = (10, 4)$ m for the Air3d compatibility Height of Burst case of Section 9.1

## 9.2 Example Three: Blast Wave Clearing

This case concerns blast wave clearing about a cuboid obstacle. The `Air3d` input file is available as `0_example_03.txt` and may be downloaded and viewed from the html version of this document. For a description of the input file see §5.1 of the `Air3d` user guide [?]. There are two target points: one on the front and one the rear of the obstacle.

Figures 124 and 125 compare the pressure histories computed from `Air3d` with those computed by `ProSAir` from the same input file `0_example_03.txt` at the two target points.

They are clearly in excellent agreement up to about 4ms - the discrepancy beyond then

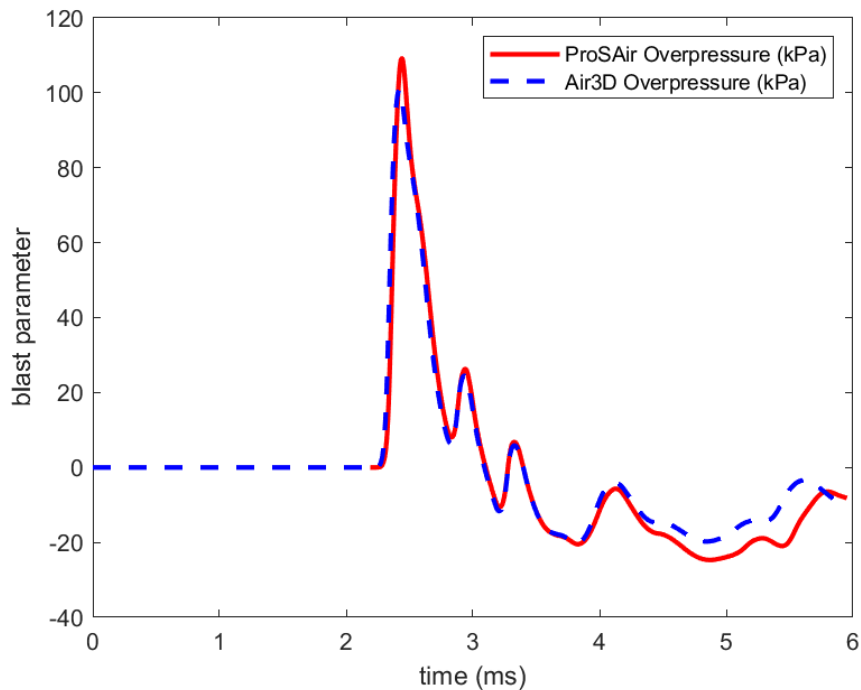


Figure 124: Overpressure history at the shock facing target point at  $(x, y, z) = (1.499, 0.001, 0.090)$ m for the `Air3d` compatibility Clearing case of Section 9.2

is likely due to variations in round-off errors.

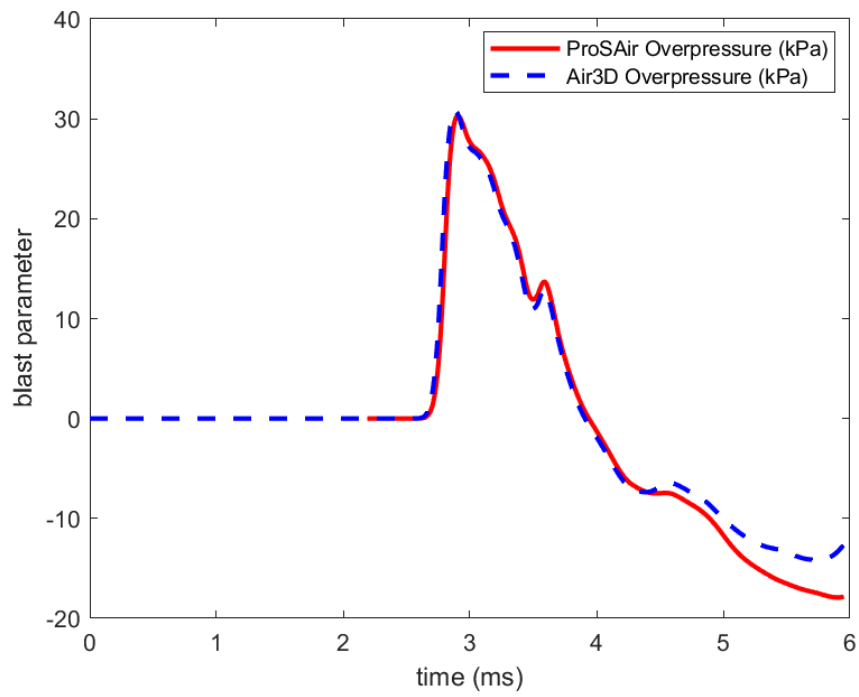


Figure 125: Overpressure history at the rear facing target point at  $(x, y, z) = (1.5611, 0.001, 0.090)$ m for the **Air3d** compatibility Clearing case of Section 9.2

### 9.3 Example Four: Testing Glazing in a Cubicle Structure

This example forms the ProSAir *Glazed Cubicle* example of §6.4 but here performed from the **Air3d** input file and compared to **Air3d** generated data. The **Air3d** input files are available as `0_example_04_3kg.txt` for the 3kg charge and `0_example_04_4kg.txt` for the 4.536kg charge and may be downloaded and viewed from the html version of this document. For a description of the input file see §5.2 of the **Air3d** user guide [?].

Figure 126 compares the pressure histories computed from **Air3d** with those computed by ProSAir from the same two input files (`0_example_04_3kg.txt` and `0_example_04_4kg.txt`) for both charges at the single target point. The ProSAir simulation captures a higher peak

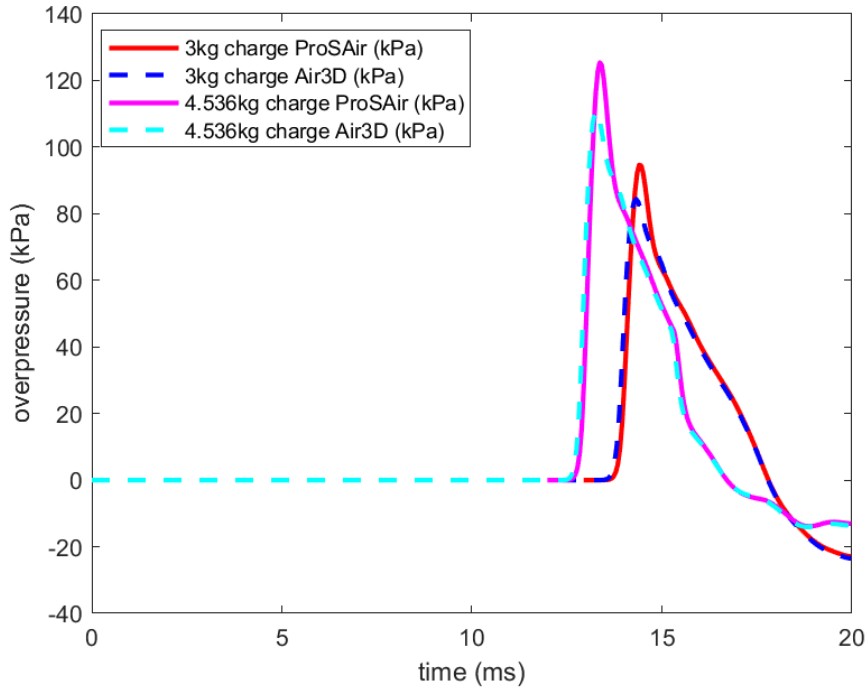


Figure 126: Overpressure history at the shock facing target point at  $(x, y, z) = (1.499, 0.001, 0.090)$ m for the **Air3d** Glazed Cubicle case of Section 9.3

pressure at the target point than does **Air3d**, otherwise the ProSAir and **Air3d** traces are in excellent agreement.

Figure 127 shows the history of the pressure-impulse loading for both charges together with the window's PI-failure curve. In both cases the ProSAir peak pressure is seen to be lower as loading starts compared to **Air3d**. This is because ProSAir uses the average pressure loading across the window element while **Air3d** uses the maximum pressure loading (see §6.3). Very rapidly the corresponding ProSAir/**Air3d** loading curves converge and both predict glazing failure for the 4.536kg charge and retained integrity for the 3kg charge.

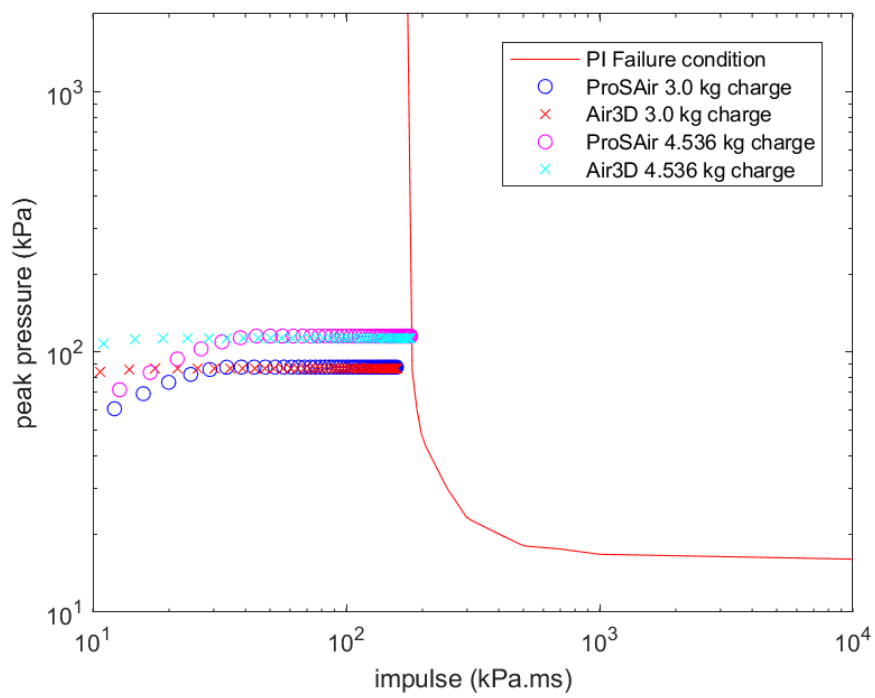


Figure 127: Window Pressure-Impulse curve for the 3kg and 4KG charges and failure curve for the Air3d compatibility Glazed Cubicle case of Section 9.3

## 9.4 Example Five: Blast in a City Street

This case concerns blast wave propagation along a street-like channel. The **Air3d** input file is available as `0_example_05.txt` and may be downloaded and viewed from the html version of this document. For a description of the input file see §5.3 of the **Air3d** user guide [?]. There are 23 target points along the length of the street. Somewhat confusingly experiments were performed with just 8 blast gauges [?, §5.3]. Target points 13 and 23 corresponding to the location of gauges 3 and 8.

Figures 128 and 129 compare the pressure histories computed from **Air3d** with those computed by ProSAir from the same input file `0_example_05.txt` at two target points.

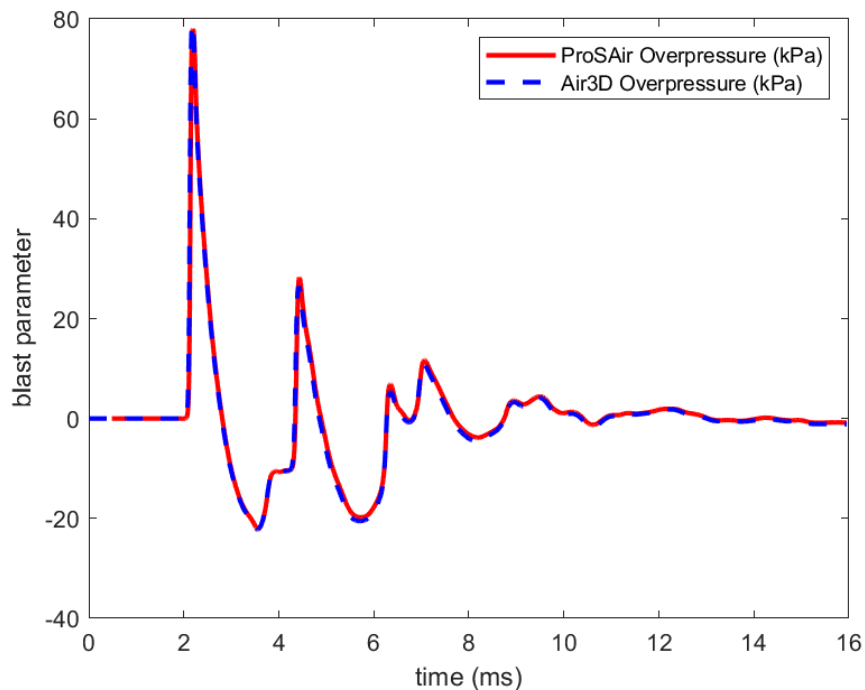


Figure 128: Overpressure history at the blast gauge 3, target point 13 for the **Air3d** compatibility City Streets case of Section 9.4

They are clearly in excellent agreement throughout the computation.

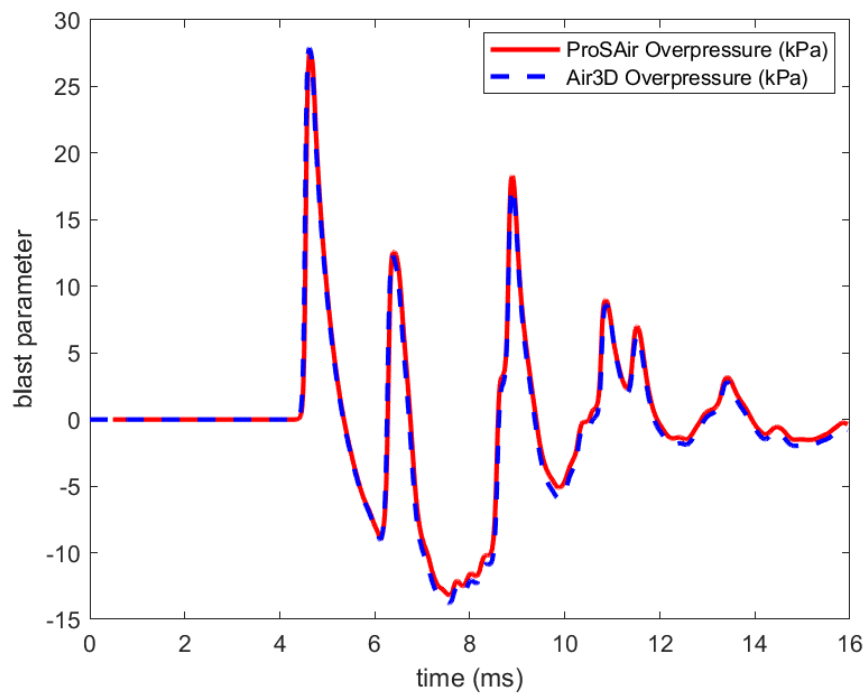


Figure 129: Overpressure history at the blast gauge 3, target point 23 for the Air3d compatibility City Streets case of Section 9.4

## 10 Trouble Shooting

### 10.1 FAQs

#### 10.1.1 My target point files contain no data

Target point files record data in the flow field so either:

- Your target point lies outside the domain - in which case please adjust the domain size as described in § 4.2.3 (spherical domain), § 4.3.2 (cylindrical domain) or § 4.4.2 (3d domain).
- It is probable that your target point lies within a cell that ProSAir thinks is a solid cell. It might be that you wish your target cell to lie on, or close to, the surface of a geometric feature (e.g., a wall). However, in ProSAir grid cells are either solid, or gas, and not a mixture. This situation may usually be accommodated by shifting the position of a target point half a cell width, or maybe less, away from the solid surface. See the examples in §3.4.8 and §3.3.4.

#### 10.1.2 The 64 bit version of ProSAir crashes

A now rarely encountered bug is that the 64 bit version of ProSAir may crash if the **Save Plots** option is on in either the spherical §4.2.1, cylindrical §4.3.1 or 3D §4.4.1 geometry settings. To prevent this turn the **Save Plots** option to off in these settings or use the 32 bit version.

#### 10.1.3 My results are slightly different to those in the User Guide

The results in the User Guide are all obtained using the single precision, 64 bit version of ProSAir. Results in double precision and on the 32 bit platform will vary slightly due to differing floating point round-off errors arising from use of double precision or different compiler optimisations. Typically results agree to about 4 significant figures.

## References

- [Bri04] Matthew A. Brittle. *Blast propagation in a geometrically complex urban environment*. MSc Dissertation, Weapons Effects on Structures, Cranfield University, Defence College of Management and Technology, Defence Academy of the UK, Shrivenham, Swindon, SN6 8LA, UK, July 2004.
- [Joh99] N.F. Johnson. European blast resistant glazing standards: the shock tube and range tests compared. In *Proceedings of the 9th International Symposium on the Interaction of the Effects of Munitions with Structure*, Strausberg, Germany, May 1999.
- [KB84] Charles N Kingery and Gerald Bulmash. Airblast parameters from TNT spherical air burst and hemispherical surface burst. Technical Report ARBRL-TR-02555, US Army Armament Research and Development Center, Ballistic Research Laboratory, Aberdeen Proving Ground, Maryland, USA, April 1984.
- [Ros01] Timothy Rose. *An Approach to the evaluation of blast loads on finite and semi-infinite structures*. PhD Thesis, Cranfield University, Engineering Systems Department, February 2001.
- [Ros06] Timothy A. Rose. A Computational Tool for Airblast Calculations: Air3d version 9 users' guide. User Guide, Cranfield University, Engineering Systems Department, Defence College of Management & Technology, Defence Academy of the United Kingdom, Shrivenham, Swindon SN6 8LA, UK, February 2006.
- [RSF06] Timothy A. Rose, Peter D. Smith, and Shaun A. Forth. A computation tool for the evaluation of blast-structure interactions - ftt-air3d users' guide. Technical report, Engineering Systems Department, Cranfield University, Defence Academy of the United Kingdom, Shrivenham, Swindon, SN6 8LA, UK, February 2006.
- [SC09] David Smith and David Cormie. Design of glazing. In Geoffrey Mays, Peter D. Smith, and David Cormie, editors, *Blast Effects on Buildings*, pages 177–215. Thomas Telford Ltd, 2nd edition, July 2009.
- [Spe97a] Special Services Group. Glazing Hazard Guide: Charts. Technical Report SSG/EP/3/97, Security Facilities Executive, Explosion Protection, St Christopher House, Southwark St., London SE1 0TE, 1997.
- [Spe97b] Special Services Group. Glazing Hazard Guide: Cubicle Stand-offs Tables and Charts. Technical Report SSG/EP/4/97, Security Facilities Executive, Explosion Protection, St Christopher House, Southwark St., London SE1 0TE, 1997.
- [Spe97c] Special Services Group. Glazing Hazard Guide: Descriptive Tables. Technical Report SSG/EP/1/97, Security Facilities Executive, Explosion Protection, St Christopher House, Southwark St., London SE1 0TE, 1997.

- [Spe97d] Special Services Group. Glazing Hazard Guide: Tables. Technical Report SSG/EP/2/97, Security Facilities Executive, Explosion Protection, St Christopher House, Southwark St., London SE1 0TE, 1997.
- [Spe98] Special Services Group. Design for glazing protection - current UK design philosophy. Technical Report SSG/EP/1/98, Security Facilities Executive, Explosion Protection, St Christopher House, Southwark Street, London SE1 0TE, UK, May 1998.
- [SRB05] Peter D. Smith, Timothy A. Rose, and Matthew A. Brittle. Analysis of a generic cityscape using an adaptive mesh CFD code. In *Proceedings of the 12th International Symposium on Interaction of the Effects of Munitions with Structures*, New Orleans, Louisiana, USA, September 2005.
- [Tor99] Eleuterio F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*. Springer, 2nd edition, June 1999.
- [WL97] Yasuhiro Wada and Meng-Sing Liou. An Accurate and Robust Flux Splitting Scheme for Shock and Contact Discontinuities. *SIAM Journal on Scientific Computing*, 18(3):633, 1997.

## Appendices

## A Change Log

- 2022-12-05 \* Try to force save to file on creation of a new simulation
- 2022-11-30 \* Added new material Glazing NoFailure - adjusted documentation
  - \* New obstacles default to "Glazing NoFailure" for windows and "Solid" otherwise
- 2021-12-01 \* If ProSAir opened without a file, then defaults to user's home folder.
- 2018-02-20 \* Recompiled for 2018.3a release, documentation adusted for 64 bit being usual version.
- 2012-05-04 \* Released 2012.5a
- 2012-05-04 \* - Change Log added to user guide.
  - Height of Burst test case expanded to include convergence study and comparisons with Kingery Bulmash curves.
- 2012-03-28 \* bug fixed in array index for MUSCL slope calculation at boundaries
  - \* reflection applied in MUSCL only at reflection boundary conditions
- 2012-03-26 \* Bug fixed for STOP boundary condition on lower x,y,z boundaries in 3D
- 2012-02-01 \* Tag 2012.1d created
  - bug fixed so cumulative impulse etc continues beyond MaxPhases=4
- 2012-02-01 \* Tag 2012.1c created
  - \* Target summary files updated:
    - Include maximum impulse and its occurrence time
- 2012-01-23 \* Tag 2012.1b created.
- 2012-01-20 \* Target summary files updated:
  - Include Maximum/Minimum pressures and their occurrence time
  - Total impulse
  - Extra header line included to clarify phase of phase data
- 2012-01-16 \* Tag 2012.01a created.
- 2012-01-13 \*
  - Changed target point output data to include cumulative impulse.
  - Improved algorithm for detecting changes of phase in target point and window sumary data to use linear interpolation to find phase end time.
  - Integration of impulse also modified to account for this.
- 2012-01-13 \* Fixed bug leading to "About" information not being displayed.
- 2011-12-05 \* Added "Split Targets" facility to Simulation menu to split existing target point files
- 2011-12-01 \* Released 2011.11b
- 2011-11-29 \* removed thread unsafe parallel loop in Mesh::AddObstacles to correct crash in large window simulations.
- 2011-11-25 \* Corrected input of orientation for Wedge and cylinder objects from legacy Air3D input files.
- 2011-11-16 \* Corrected glazing input and behaviour from legacy Air3D input files.
- 2011-11-01 \* Ensured number of threads is not greater than environment variable NUMBER\_OF\_PROCESSORS
  - Modified CPU time display in GUI to Run Time display and ensured

works on 64 bit.

2011-10-31 \* Example documentation finished.

2011-10-17 \* Enabled saving of plot scale factor in cylindrical phase

2011-07-20 \* Added command line and batch processing documentation.

2011-07-18 \* GUI version now takes additional command line arguments run and exit to permit running from batch scripts.

2011-07-13 \* window summary files now have 1/0 for broken

2011-07-04 \* window history files now log pressure as well as maximum pressure, window summary files now have yes/no for broken and break time logged.

2011-07-01 \* removed Materials/Glazing/Glazing Example.csv as prmat forms now working.

2011-06-16 \* Modified glazing to fail on average, not maximum, pressure.

2011-06-03 \* Corrected bug in which command line version did not terminate.

2011-06-03 \* Command line version displays simulated time.

2011-06-03 \* Corrected bug preventing completion of command line version.

2011-05-27 \* Corrected bug for cylindrical charges when adjacent to ground.

2011-05-08 \*

- Postinstallation option to open ChangeLog added to installer.

- ChangeLog accessible from ProSAir help menu.

2011-05-18 \* Help now opens the LaTeX defined and Hevea generated ProSAirUserGuide via the user's default browser. wxWidget's browser is not used as its linking across sub-pages is inconsistent under the Hevea generated html and it has little support for special characters (e.g. <=). User guide also added to Start->ProSAir menu.

2011-05-05 \* Corrected bugs in reading user defined explosives:

1) User defined explosive files may be read in and correctly set GUI and solver properties.

2) Simulations with user defined explosives may be saved to, and restored from a .pr1 ProSAir simulation file.

2011-05-27 \* Corrected bug for cylindrical charges when adjacent to ground.

2011-03-17 \* Removed ability to read obstacles from file in cylindrical settings.

2011-03-14 \* New setting "at end of phase" in Checkpoints panel allows user to turn off saving of checkpoints at the end of each solution phase - note this option prevents restarting from the end of a previous solution phase and user must manually Restart the solution.

2011-03-11 \* Corrected bug when folder not set correctly on saving to a new file.

2011-03-04 \* Allows for cylindrical charges from input file but not GUI

2011-02-07 \* Variation in atmospheric conditions added

2009-11-06: \* Ported changes from 2009.10b branch over to trunk.

2009-11-02: \* First bugfix release (2009.10a)

\* Fixed bug causing crashes when deleting obstacles in the GUI.

\* Scale factor is read in properly for 3d geometries.

2009-10-30: \* Full release

\* Parameters are stored in double precision and only converted to single precision (if using single precision executable) when transferred to the

mesh. This means reduced loss of precision when calculating derived parameters for the mesh.

- \* ProSAir now outputs windows and target summaries. Find peak pressure/impulse for each target point.
- \* Fixed bug causing occasional crashes when restarting simulations.
- \* Updated documentation.
- \* Fixed bug causing cylinders to be read wrongly into the Settings dialog.
- \* Boolean variables ("true"/"false") are no longer required to be lower-case only in the legacy input files.
- \* Command line client is included.
- \* More direct mapping between cells and the contour plots.
- \* Shifting of target points is now optional.
- \* Target point files may be split into individual files at the end of the simulation.
- \* ProSAir now does not attempt to plot contour plots of velocity data.
- \* ProSAir should now handle bad memory allocations better and should not crash when the user attempts to create a too large mesh.
- \* CPU time does not get messed up when restarting simulations.

2009-10-22:

- \* Changed versioning system.
- \* Fixed problems with target point output.
- \* Added draft for ProSAir manual.
- \* Obstacles can now be moved up and down in the evaluation order inside the GUI.
- \* New plot options for cylindrical geometry.
- \* Changed the tolerance for switching geometry to more closely match Air3D. ProSAir should switch to higher geometries at a time closer matching Air3d.
- \* Changed geometry tolerances somewhat to be closer to what Air3D does.

2009-10-16:

- \* 1.0 Release Candidate, with a whole host of updates:
- \* New file input system
- \* Ability to add target point grids
- \* Settings can be changed and saved in a graphical user interface.
- \* objects are supported in 2D as well as 3D.
- \* check point system.
- \* You can change many settings without restarting the simulation.
- \* New colour scheme
- \* Directives (i.e. !TARGET) for legacy input removed. Default values assumed.
- \* Installation program
- \* Much faster target data output
- \* Target points can now be shown in the plots.
- \* And much more

2009-08-21:

- \* BETA 2 released with the following updates:
- \* Added support for 2D/axisymmetric objects.
- \* More information is output into the plots themselves.

- \* Support for exporting plots into PNG and JPEG added.
  - \* Error messages for target points are now more descriptive.
  - \* Bug fixed where ProSAir would run unnecessary remaps increasing error in certain circumstances.
  - \* Target point output data can be reduced with the ! TARGETS directive. See the release notes.
  - \* Time not corrupted when negative pressure is encountered.
  - \* Expiry date postponed until 2009-11-01.
  - \* Other smaller fixes.
- 2009-05-22: \* BETA 1b released, with the following updates:
- \* Fixed window code to work correctly for certain window orientations
  - \* Added preliminary support for exporting plots. So far only Encapsulated PostScript can be exported.  
Check out Ghostscript and GSview for ways of converting to other formats (<http://pages.cs.wisc.edu/~ghost/index.htm>)
  - \* Removed the "Start ProSAir" shortcut as it refers to a directory which will be non-existent on a client computer. Users may create their own shortcut and drag it to the Start Menu.



**DSPACE**

<https://dspace.org/>

## **ProSAir 2022.11 User Guide**

**Forth, Shaun A.**

**2022-11-01**

Cranfield University

<https://dspace.lib.cranfield.ac.uk/handle/1826/24030>

Forth S. (2022) ProSAir 2022.11 User Guide. November 2022, CDS Shrivenham, Cranfield University