Image Cover Sheet

CLASSIFICATION UNCLASSIFIED	**SYSTEM NUMBER** 51601
TITLE COMPENDIUM OF BLAST WAVE PROPERTIES	

System Number:

Patron Number:

Requester:

Notes:

DSIS Use only:

Deliver to: JR

UNCLASSIFIED UNLIMITED

DRES

CONTRACT REPORT 17/87

COMPENDIUM OF BLAST WAVE PROPERTIES

J.M. Dewey and D.J. McMillin
University of Victoria
Victoria, B.C.

February 1987

DEFENCE RESEARCH ESTABLISHMENT SUFFIELD, RALSTON, ALBERTA

UNCLASSIFIED

COMPENDIUM OF BLAST WAVE PROPERTIES

J.M. Dewey and D.J. McMillin

Final Report

Contract # 8SG83-00211

20 February 1987

University of Victoria
Victoria, B.C.
Canada

SUMMARY

40 // This report describes the blast wave produced when one kilogram of TNT (trinitrotoluene) is detonated in air at sea-level. Rules are given for scaling to other charge weights, explosive materials and atmospheric conditions. Data are presented for both free field and surface burst charge configurations. //

Data, such as static pressures in the blast waves, are presented as functions of time at fixed distances from the charge and as functions of distance at fixed times. Also presented are the variations with distance of the peak data values.

The range of distances over which data are presented is from 1 meter to 15 meters from 1 kg TNT. Over this range in the free field case the peak static overpressure ratio falls from about 10 to about 0.065, or from about 150 psi to about 1 psi in a sea-level atmosphere.

The range in time over which data are presented, except at a few of the smaller distances, is the entire positive phase of the blast wave.

The data presented here comprise a summary of the large amount of data which have been measured experimentally and calculated over the past forty years. Those data were collated, smoothed and interpolated to bridge data gaps, and related data were computed. For this, a numerical reconstruction technique was used and it is described. The results are compared with some of the better-known source data.

A computer software package is described which displays the data described here and, in addition, scales the data to user-specified yields and ambient conditions with options for interpolation and integration. The data base and source code for the program are supplied on a magnetic tape with this report.

Some preliminary work on a future extension to this package is described. The extension will include data for a complete range of elevated or height-of-burst charge configurations.

CONTENTS

total number of pages: 151

Chapter 1

INTRODUCTION

Since World War II a large amount of quantitative information has been accumulated on the effects of nuclear and conventional explosions. The fundamental physical parameters which define the strength of a blast wave produced in the air surrounding an explosion have been measured in attempts to relate the amount and kind of damage done, or predicted, to the amount of energy released in the explosion. Theories have been formulated which attempt to predict those fundamental parameters.

More recently, purely numerical methods have been used on large computers to predict both the fundamental properties of blast waves and the responses of various structures to blast wave loading.

This report summarizes the blast wave data which have been measured experimentally for TNT in air. It was originally thought that those data would be simply gathered and compared, by plotting all the data of each kind on a common scale. The data would then be summarized by drawing best-fit curves and noting by how much the data were scattered around those fits. Data not sufficiently well-known experimentally would be interpolated from the data summarized in this manner.

It was soon established, however, that nearly all of the primary, measured data were pressures measured using gauges, and shock and particle trajectories measured photographically. The only additional data which could be easily and unambiguously computed from these were flow velocities. Other data, needed to completely define the flow, such as densities or temperatures, had not been sufficiently well measured or, when they could be computed from the primary data, showed too much scatter.

A technique more sophisticated than simple gathering and smoothing was therefore required. The technique which was adopted for this report is one known as numerical reconstruction, wherein the parameters defining a purely numerical model are adjusted until the predictions of the model are matched 'as well as is possible' with all of experimentally-known data. The results from the model then summarize the measured data, and can be used to predict with some confidence the other, unmeasured data.

The information in this report is presented in the order that the work was done. The measured properties of blast waves in air which are currently available in the literature, and some privately obtained data, are reviewed in chapter 2. In chapter 3, the concept of a numerical reconstruction is introduced and several examples are given.

The reconstructions actually used to obtain the data presented in this report are described in chapter 4. The basic results are presented, that is, blast wave data for one kilogram of TNT in air, for both free field and surface burst charge configurations.

Pressures, densities and flow velocities are presented over the entire middle distance range and throughout the entire positive phase of the blast waves, as time histories at fixed positions, wave profiles at fixed distances and peak profiles (peak data values versus distance).

Data are compared to the better known of the experimentally measured data, and to one set of independently computed data.

Rules are given by which other data can be computed. These include primary data for other charge sizes, explosive materials other than TNT and ambient conditions not those used for the basic reconstructions, which were for a dry sea-level atmosphere at 15° C.

Rules are also given by which secondary data can be computed. For example, densities and flow velocities can be used to calculate dynamic pressures, which can in turn be used with static pressures to calculate total pressures, energy densities, etc. Examples are given.

Chapter 5 describes a computer software package which was developed to perform those functions described in the previous chapter, namely, the retrieval of primary data, scaling and secondary calculations, with the added options of interpolation and integration of the data after they are displayed. The package is supplied on a magnetic tape with this report.

In the final chapter, some work is described which was done towards a future extention of the data base, to include height-of-burst blast properties.

Note:

To obtain blast wave data, it is necessary to read only chapter 4. To make use of the data display package which is described in chapter 5, only chapter 5 is required. Each of those chapters stands on its own.

Chapter 2

REVIEW OF SOURCE MATERIAL

This chapter describes briefly the better known and more generally available source material on explosions in air. It is not intended that the review be complete. Rather, the review is included here to show that considerable disagreement exists even for the most frequently measured of the important physical parameters, viz., peak static overpressure as a function of distance from charge centre. It will be shown that there are differences in the known data which must be removed by averaging. More importantly, when the collation of experimental data was complete, the conclusion was reached that there do not yet exist sufficient experimental data to completely describe the flows behind the shock fronts in blast waves.

In order to completely describe these flow fields, it is necessary to know three of the physical properties over the entire flow, such as static pressure, density and flow velocity, and γ, the ratio of specific heats. γ can be assumed constant beyond the point at which the peak static overpressure has fallen to below about 10 atmospheres, which is about one meter from a one kilogram charge of TNT.

Although sufficient pressure and flow velocity data exist in this flow field region, the same is not true for densities (or temperatures, etc.). Average densities in the spaces between particle tracers can and have been computed from particle trajectory photogrammetry in this region, but they are generally too scattered to be smoothed by simple averaging. This is partly because they are average densities and partly because small errors in reading the tracer positions affect the computed densities for adjacent cells in opposite directions, increasing the average density computed in one cell and decreasing it in the other.

It will be shown later that the incompleteness of the experimental flow field data does not limit the present work.

2.1 Free field data

Free field blast waves have not been studied experimentally as often as those from surface burst charges, because experiments are more easily done on the ground, especially when using large charges. Free field blast waves are almost always chosen for analytical and numerical study, however, as calculations are simpler in the absence of a reflecting surface.

Probably the best known source for blast wave data is Baker (1973). Incident and normally reflected shock front parameters are compiled, together with times of positive duration and pressure impulses. Specific free field source data reproduced by Baker include those of Kennedy (1946), Stoner & Bleakney (1948), Goodman (1960) and Lutsky (1965). Few flow field data behind the shock front are presented, however.

A more recent, but less extensive, source of free field blast wave data is Held (1983), who reproduces Goodman (1960) and adds TNT data from Strehlow & Baker (1975). Flow field data are not presented.

Some free field data exist for explosives other than TNT. For example, Schardin (1954) presented data for small lead azide explosions and Boyer, Brode, Glass & Hall (1958) and Boyer (1959), for pressurized glass spheres. Nuclear data were first made generally available by Glasstone (1957, 1962, 1977). Few flow field data are included.

Two types of measurement have provided most of the flow field data: pressure time histories, measured using stationary gauges, and particle trajectories, measured photographically, using smoke tracers. Examples of the first are in the reports of the Ballistic Research Laboratory (BRL) and those of the Defence Research Establishment Suffield (DRES).

Particle trajectory data are to be found mostly in papers and reports by Dewey and Dewey & McMillin (1963 to present). More such data exist in unpublished form at the University of Victoria.

In the free field case, pressure gauge data and particle trajectory data are usually obtained from height-of-burst experiments, from above the triple point trajectories, and so are often limited in time because the free field data end with the arrival of a reflected shock front.

Two series of experiments are especially valuable sources of TNT data, the FE589 series (1969, shots 1 through 9) and the DIPOLE WEST series (1975, shots 17 through 24). All were height-of-burst experiments using 1000 lb spherical charges, with pressure gauges and particle tracers positioned in both the free field and Mach stem regions.

Experiments using Pentolite are numerous, and both pressure time histories and particle trajectory data are available, for example, from the MIGHTY MACH IV series (1982, shots 10, 11 and 12). Pentolite is a 50:50 mix of TNT and PETN (pentaerythritol-tetranitrate) and is slightly more energetic than 100 percent TNT

Free field data have been measured for ANFO (ammonium nitrate-fuel oil), in the DIRECT COURSE experiment, in 1983.

A second broad class of source material for free field data is the analytical and numerical work which has been published. Analytical work has been pioneered mainly by von Neumann (1941), Taylor (1941), Sedov (1946), Sakurai (1965) and Bach & Lee (1970), and numerical work, by Brode (1955-59), Thornhill (1960), Lewis (1961), Gorsch & Dunn (1969), Needham, Havens & Knauth (1975) and Needham & Crepeau (1981), to name only a few. A recent numerical source is ANSI (1983). Flow field data are presented in most of the numerical references cited above. Free-field data are computed by all of the current 2-dimensional hydrodynamic computer codes (the HULL code, etc.).

2.2 Surface burst data

Numerical calculations for the surface burst charge configuration are few. Most, e.g. Groth (1985), simply use data from some free field calculation and assume reflection factors to account for the geometry.

Experimental measurements, on the other hand, are numerous. Baker (1973) reproduces the results of Kennedy (1946), Dewey (1964), Dewey & Anson (1965), Reisler, Giglio-Tos & Kellner (1966), Jack (1963), the well known summary by Kingery & Pannill (1964), and Kingery (1968). As was the case for the free field data, most of the measured data are at the shock front rather than over the complete flow field.

For flow field data, as it was for the free field case, pressure time histories and particle trajectories comprise the bulk of the measured data. The former kind of data may be found, for example, in BRL reports and most of the latter are 'on-line' at the University of Victoria. Experiments of particular relevance to the work here are SNOWBALL (1964) and SAILOR HAT (1965), both 500 tons TNT, and DIPOLE WEST shot 25 (1975), 500 lbs TNT.

Some ANFO and nuclear data are summarized by Sadek & Gottlieb (1983). Flow field data for the largest-ever controlled propane-oxygen explosion, DISTANT PLAIN shot 2A, are described by Dewey & McMillin (1981).

2.3 Height-of-burst data

Data from height-of-burst experiments are presented by Brode & Carpenter (1975), Reisler (1976), Anderson (1977), Anderson & Deckker (1977), Hu & Glass (1985) and Worsfold & Rees (1986). Baker (1973) describes some of the original data of Kennedy (1946) and Bryant, et al (1952), and also the data of Schleuter, et al (1965).

A large body of particle trajectory data for TNT height-of-burst experiments exists at the University of Victoria, much of it unpublished. The FE589, DIPOLE WEST and MIGHTY MACH IV experiments cited in the section on free field data are examples. Data from the largest non-nuclear height-of-burst experiment ever conducted, DIRECT COURSE, are available. Photographic and gauge data are also available for a series of height-of-burst experiments using micro-charges (as small as 16 grams) of Hexogen, conducted at the Ernst Mach Institute in 1983 and 1984.

2.4 Other sources

Books on the general topic of explosions and blast waves include those by Baker (cited above), Glasstone (cited above), Henreych (1979), Bowen, Manson, Oppenheim & Solokhin, eds. (1981), Field (1982), Baker, Cox, Westine, Kulesz & Strehlow (1983) and Kinney & Graham (1985). The book by Field deals with dust explosions.

The bibliography at the end of this report contains many publications not mentioned here, which might be useful in a search for specific data on blast waves, but for reasons mentioned above neither that list nor this brief review should be considered complete and exhaustive.

2.5 Reduction and comparison of source data

Data which were presented graphically in sources such as those described above were digitized, that is, the x-y coordinates of the origin and axes end points, and points on various curves in the source figures were measured and stored in a computer, in arbitrary (digitizer) units. These data were then translated, rotated and scaled so that they could be re-plotted, all on a common scale, and combined and compared.

The digitized data were scaled to account for both the conversion from digitizer units to actual data units, and the adjustment for various charge yields and ambient conditions. Distances and times which were presented in the source material were scaled for charge yield and ambient conditions as follows:

$$R(\text{scaled}) = R(\text{measured})/S$$
$$T(\text{scaled}) = T(\text{measured})/St$$

where S is the cube root of the TNT-equivalent charge yield in kilograms divided by the ratio of the measured atmospheric pressure to a standard pressure. St is this value of S times the ratio of the measured speed of sound to a standard sound speed. This type of scaling (Sach's scaling) is done frequently in blast wave research, to compare data from different experiments, and has been shown to be valid for a large range of energy yields.

Data other than distance and time (pressures, for example) which were digitized were normalized to ambient atmospheric values. Such ratios are invariant under the scaling described above. For example, P/Po at R(measured) remains P/Po at R(scaled), where Po is the ambient, atmospheric pressure (measured as part of the experiment in the first case and a standard ambient pressure in the second, scaled case).

2.6 Results

Figures 1a and 1b respectively show a comparison of peak hydrostatic overpressure ratios versus scaled distance from the charge centre, for free field TNT and Pentolite. Data from the literature were digitized as described above and scaled to a one kilogram charge yield, in a standard, sea level atmosphere. Figure 2 shows a similar comparison for surface burst TNT, scaled to one kilogram in a standard atmosphere.

The data presented in figures 1 and 2 are perhaps the best known experimental blast measurements. Measurements of other peak parameters are scarce and showed as much or more scatter when similarly scaled and compared, with the exception of time of shock front arrival, which is more easily and more accurately measured than peak hydrostatic overpressure.

Chapter 3

NUMERICAL RECONSTRUCTION TECHNIQUES

The scatter in the data shown in figures 1 and 2 requires that some kind of average or best-fit curves be used to describe the variation of peak static overpressure with distance. Data not as well known as peak static overpressure, when compared in the same way, were found to be even more scattered and therefore in more need of averaging, or smoothing.

Averaging sets of related physical data should be done in such a way that the different results remain self-consistent. If, for example, peak pressures are averaged and, separately and independently, densities and flow velocities, the averaging should be done in such a way that the results remain consistent with the laws of hydrodynamics, which require that these parameters are related in specific ways at all times. It is equally important that this be done for data in the flow field behind the shock front as it is for the peak data, at the shock front. Thus, a specific averaging or smoothing method must be used.

To ensure this self-consistency, a smoothing technique known as numerical reconstruction was adopted. In general terms, numerical reconstruction is an independent and wholly numerical calculation which duplicates other data, usually data obtained from a particular experiment. As in any numerical calculation, assumptions are made, for example, concerning equations of state, parameters such as viscosity coefficients are defined and boundary conditions are chosen. Numerical reconstruction is the process wherein those assumptions, parameters and boundary conditions are adjusted and the calculation repeated until it duplicates or nearly duplicates the measured data.

It might be argued that all numerical modelling, whether predictive or merely descriptive, is a numerical reconstruction, in that it is an iterative attempt to reproduce reality.

If the data which are to be reproduced are scattered, as is generally the case, even for data measured in a single experiment, then not an exact, but rather some 'best' agreement must be sought in the reconstruction. A weighted, least-squares fit to the source data can be sought, for example.

Once the sought-for agreement has been obtained, the results of the numerical calculation, which now represent the smoothed source data, can be interpolated to fill any gaps in the source data. Moreover, because the algorithm of numerical reconstruction is hydrodynamically consistent, it can be used with reasonable confidence to extrapolate source data, a process which is unreliable using other smoothing techniques.

For example, if pressures and flow velocities are reproduced in a reconstruction, then the densities which are calculated will be accurate even though no measured densities may be available for comparison.

In the applications of numerical reconstruction to blast wave data discussed here, no attempt is made to model the complete explosion process. Only a reconstruction of the flow field some distance from the charge is attempted. This simplifies the numerical calculations since at moderate pressures air can be treated as a thermally and calorically perfect gas and real gas effects can be ignored. Computing times can be reduced.

3.1 Early techniques

A crude form of numerical reconstruction was used by Glasstone (1957, 1962, 1977), who computed all the flow properties at a point using only the pressure-time history measured there. The underlying assumptions were crude and only approximate results were obtained. Details and an assessment of this method are given by Gottlieb & Ritzel (1979).

A technique was developed by Makino & Shear (1961) to compute flow field properties from a section of the observed shock front trajectory, using the method of characteristics. The shock front trajectory was determined by fitting time-of-arrival data measured near the test site. A serious limitation to this technique was that that the region over which flow field data could be computed was always relatively small whatever the length of known shock trajectory, although the size of this region was later extended by Gottlieb & Ritzel (1979), and again by Celmins (1981), who used measured pressure-time histories as well as the shock trajectory data.

A different approach has been used by Dewey (1963, 1964, 1971) and Dewey & McMillin (1977 through 1986). Particle trajectories were measured photographically using smoke tracers, and the results used to compute all parameters in the flow field in the region of the tracers. Flow velocities were easily computed as they are the first derivatives of curves fitted to the particle trajectories. Densities were computed from the relative closeness of the tracers. Pressures were then calculated from the densities and entropy changes which had been calculated from the shock velocities obtained by fitting a curve to observed times-of-arrival at the tracer starting positions.

The only limitation to the particle trajectory technique is one of accuracy. Unless the tracer positions are measured very accurately, the computed densities and pressures have large uncertainties. This scatter is greater in two-dimensional flows, e.g., height-of-burst experiments, than it is in one-dimensional flows. Physically significant features, e.g., shock fronts, can be lost when the data are smoothed.

3.2 The piston path method

The first true numerical reconstructions of blast waves are those of Lau & Gottlieb (1984), who reproduced the shock trajectories, particle trajectories and pressure time histories measured in several surface burst experiments. Lau & Gottlieb assumed that a piston, moving some distance from the centre of an explosion, could be used to drive a numerical calculation which would reproduce the trajectories and pressures measured in a blast wave, allowing for some random experimental error. The exact motion of the piston which should be used was not known in advance, although the piston would be expected to move exactly as a particle tracer placed at the piston's starting position. The piston path, in the radius-time plane, was first guessed and then adjusted repeatedly and intelligently until a best agreement with the experimental data was obtained. The details of the explosion process itself were ignored. The adjustments to the piston path path were explicit, and not hidden in an unnecessary, complex and expensive calculation modelling the explosion.

Figure 3 illustrates the underlying assumption of the piston path method of reconstruction. At the 'observation window', the flows produced in one case by an explosion and in the other case by a piston cannot be distinguished, as long as the piston motion matches the explosively-driven flow at the piston's starting position.

A one-dimensional, linear flow is represented in figure 3, but the principle also applies to radially symmetrical (pseudo-one-dimensional) flows. In the case of a cylindrical flow, a cylindrically expanding piston is used to drive the calculation, and for spherical and hemispherical flows, such as free field and surface burst blast waves, spherically and hemispherically expanding pistons are used.

Figure 4 illustrates the piston path method in the position-time plane. The piston, initially at rest, is started abruptly to generate a shock. The piston starting velocity determines such parameters as the pressure behind the shock front at the start time. The piston speed is then reduced, as in the blast wave case, and eventually made negative to produce a negative phase. At some time in the calculation, the piston's motion is slightly, but abruptly altered, to generate either a second, weak shock front (piston speed increased) or a rarefaction wave (piston speed decreased). In figure 4 the piston is simply brought abruptly to rest. The small disturbance in the piston's motion produces a wave front, which defines an upper boundary on the computed flow field.

3.3 Hydrodynamic codes

A variety of numerical techniques have been developed for computing hydrodynamic flows, differing mainly in the way in which they handle embedded discontinuities, such as shock fronts. Three general approaches to embedded discontinuities are identified by Woodward & Colella (1984), in a review of the then current, two-dimensional computer codes. The first approach, due originally to von Neumann & Richtmeyer (1950), involves the use of an artificial viscosity to smear the discontinuities over several computational cells.

The second approach to discontinuities uses a high-order scheme in regions of smooth flow and a low-order scheme at the discontinuities, so truncation errors in the latter scheme, acting as 'numerical viscosity', again smear the discontinuities. To conserve mass, etc., fluxes of the conserved quantities must be matched at the interface between the two computational zones. Boris and Book (1973) introduced the method of flux-corrected transport to maximize the sharpness of the shocks, keeping them monotonic and preventing unwanted oscillations.

In both approaches outlined above, something is added to a high order computational method to smear the discontinuities and dampen out numerical oscillations. This effect is not accidental nor inevitable, but deliberately added to make the flow variables smooth and continuous.

The third general approach to discontinuities, and the one used here, was first suggested by Godunov (1959). Smooth solutions are not sought. Rather, explicit nonlinearity is introduced everywhere in the difference calculation to produce a solution 'full' of discontinuities, but approximating the true solution very well both over its smooth parts and its actual discontinuities.

The elemental, discontinuous solutions sought at each grid point are those of the Riemann, or shock tube, problem, which describe the nonlinear flow developing from the discontinuous jump separating two different, but constant states. This flow consists of two nonlinear waves, either shocks or centred rarefaction waves, with a contact discontinuity between (e.g., Courant & Friedrichs, p. 181).

Figures 5 through 7 illustrate the various stages in the Godunov approach to solving blast wave flow problems, in one spatial dimension. The hydrodynamic flow is approximated by a large number of constant states. The exact interactions of these states are computed and the results averaged in a conservative fashion to obtain an accurate and well-behaved solution.

Higher order extensions to Godunov's method have been discussed by van Leer (1977, 1979) and Colella & Glaz (1985). The best results in two dimensions had, until 1984, been obtained using a 'piecewise parabolic method' (Colella & Woodward, 1984).

3.4 The random choice method

An important and different way of using Riemann solutions was developed by Glimm (1965), Chorin (1976) and Sod (1978) in one dimension, and Colella (1982) in two dimensions. The 'random choice method' assigns solutions of the Riemann problem which have been randomly sampled within each computational cell, to the cell centre point. The diffusive errors caused by spatial averaging are now avoided, but random oscillations in the time dimension appear in their place, and the total solution is correct only 'on average'. A schematic illustration of the manner in which the random choice method samples the explicit solution is given in figure 8.

The random choice method (RCM) works very well for applications having only one spatial dimension, but not two spatial dimensions, except in special cases such as planar and axisymmetric steady flows (Shi & Gottlieb, 1986).

A comparison of the RCM with other one-dimensional methods is given in the review paper by Sod (1978). Sod favours the RCM. It is compared to Godunov's original and 6 other methods, including artificial viscosity (Lapidus 1967) and flux-corrected transport (Boris & Book 1973). Presumably, the methods compared by Sod were the best methods then available (1977-78).

The RCM has the advantage of being able to keep discontinuities sharp, whether a single shock front or interacting shocks.

One of the biggest users of one-dimensional RCM is the University of Toronto Institute for Aerospace Studies (UTIAS). The review by Saito & Glass (1979) describes applications to a number of interacting plane wave experiments, exploding glass spheres and reacting gas flows. Since then, the RCM has been used to study shock flows in dusty gases, to reconstruct hemispherical blast waves and to model other, more complex (but always inviscid) flows. The RCM was used recently by Zhang & Gottlieb (1986) to study blast wave simulation in shock tubes.

3.5 The present code

The RCM modelling of piston-driven flows was developed at UTIAS originally by Miura & Glass (1984) and Lau & Gottlieb (1984), the latter for blast wave reconstruction. The code described in the present report contains several small improvements on the codes listed in these references; improvements made mainly to increase machine efficiency. An incorrect description of operator splitting by Sod and by all subsequent authors, and several inconsistencies in Lau & Gottlieb's (1984) work, were discovered.

The code used to reconstruct the data to be presented in this report, while modelled on the codes of Miura & Glass (1984) and Lau & Gottlieb (1984), was written from scratch using original sources, namely, Glimm (1965), Chorin (1976), Sod (1977) and Saito & Glass (1979), with additional reference to Igra, Gottlieb & Saito (1984).

In the present code a spline function is fitted to a set of discrete points in the radius-time plane, which control the movement of the piston just as in the Lau & Gottlieb program. This spline function is used to interpolate piston position and speed at arbitrary times in the subsequent flow calculations.

A computational grid is defined, initialized to the ambient atmospheric conditions of unit pressure and density, and zero flow velocity, and the piston is abruptly started at a position and time consistent with the shock trajectory expected in the explosion being modelled. The starting speed given to the piston is computed from the expected shock velocity at the starting position, using the Rankine-Hugoniot equation. Boundary conditions at the front surface of the piston are computed from the piston velocity and the conditions just in front of the piston, assuming an intervening shock when the piston is overtaking the gas ahead and a rarefaction wave when it is not. The piston is started at a distance from the charge which is large enough that the ratio of specific heats for air can be considered constant, i.e., where the peak static overpressure has fallen to below about ten atmospheres.

The initialized grid is then allowed to evolve through a half time step according to established RCM procedures set out by Godunov (1959), Glimm (1965) and Chorin (1976), and described and improved upon by others since. The solution is sampled, using a random number algorithm known as that of van der Corput (evaluated by Igra, Gottlieb & Saito, 1984), and assigned to shifted cell centre points after being adjusted for geometry. The linear data which are computed are adjusted to radial data using the technique of operator splitting described originally, though incorrectly, by Sod (1977). Particle tracer positions are updated at this stage.

The boundary conditions are updated and the complete procedure is repeated over the second half time step. The time step sizes are in general not constant, the size chosen at each step being small enough that the fastest moving signal remains in one computational cell (the so-called Courant-Friedrichs-Lewy condition).

At the end of each full time step, the blast wave front is located and its position updated. Particle tracer positions and pressures at a set of specified fixed positions are stored for later comparison with experimentally or otherwise known data. The position of the disturbance produced by bringing the piston to rest is also tracked, so that particle trajectories and pressure time histories can be properly terminated. The calculation is terminated when the disturbance has reached the outer grid boundary, or when it has hit all tracers and gauges, or when it has hit the shock front, or arbitrarily.

3.6 Test flow calculations

The code developed here was tested by computing known flows. The five test flows outlined in Lau & Gottlieb were successfully computed, along with several others. The five test flows of Lau & Gottlieb are described below.

1) An abruptly started constant velocity piston pushing a planar shock tube flow. Excellent agreement (several tenths of one percent) was obtained between the numerically computed flow and the theoretically predicted Rankine-Hugoniot flow.

2) An abruptly started constant velocity piston pulling a centred rarefaction wave. Excellent agreement (several tenths of a percent) was obtained between the numerically computed flow and the flow computed by Lau & Gottlieb using the method of characteristics. An error, presumably typographical, was noted in Lau & Gottlieb (piston Mach number 1.277 should be 1.2677).

3) A smoothly started piston, constantly accelerated to a constant velocity, creating a focused compression wave from which emerges a plane shock. Good agreement was found between the flow and the flow computed using the method of characteristics and plotted by Lau & Gottlieb (their figure 13). Inconsistencies were found in the text of Lau & Gottlieb between their description of the flow on page 15 and their figure 13.

4) A smoothly started piston, uniformly accelerated to a constant backwards velocity , creating a non-centred rarefaction wave. Agreement is difficult to assess in this case, as the inconsistencies between Lau & Gottlieb's description (page 15) and their plotted results (figure 16) is even greater than it was in test case 3 described above. Their description is not even self-consistent. A flow was produced which 'looks like' their plotted flow, however.

5) An abruptly started constant velocity, spherically expanding piston, creating a blast wave of the type predicted analytically by Taylor (1946). Excellent agreement was obtained, between the numerically computed flow and the analytically predicted flow, in the case chosen by Lau & Gottlieb (Taylor's case of α=0.5). Details are given below for comparisons made with this and the other Taylor blast waves.

3.7 Taylor blast waves

In the test case 5 above, a spherically expanding piston flow was computed, using a constant piston velocity of 178.0 m/s, or Mach Number 0.523 (Taylor's β) and, it turns out, local Mach number 0.500 (Taylor's α). The shock front created by this abruptly started piston was computed to have a constant velocity of Mach number 1.026, or 1.015 using Lau & Gottlieb's stated shock pressure ratio (1.035, their page 16). Taylor's shock Mach number is 1.021, computed from his stated shock pressure ratio (y=1.050, his table 1). The agreement with Taylor's shock Mach number is therefore +0.5%, compared to Lau & Gottlieb's -0.6%.

The Taylor wave tested above has a shock strength (static overpressure ratio) of 0.05 atmospheres. The program was also tested using the balance of Taylor's tabled data, with local piston Mach numbers, α, running from 0.4 to 1.8 (piston speeds from 139.5 m/s to 1224 m/s), the created shocks having strengths running from about 0.004 atmospheres to just over 10 atmospheres (static overpressure ratio). In other words, the tests covered the range of shock strengths expected to be necessary for the compendium of blast wave properties. Excellent agreement was obtained in all cases.

The only Taylor wave not duplicated was his strongest wave (α=2.1), a limiting case in which the shock pressure jump becomes infinitely large.

The wave behind the the Taylor shock front is self-similar; its shape does not change with time. Its shape was reproduced in all cases, as in Lau & Gottlieb's 0.05 atmosphere case (their figures 20 & 21). The Taylor wave is not, however, the classical HE or nuclear blast wave, which decays behind the shock front, slowing as it expands. The Taylor waves were used for test purposes because they are known exactly, and they are the best tests which can be made easily. It was concluded that the code had been satisfactorily tested, especially considering also the results obtained in the rarefaction wave, or backwards moving piston tests 2 and 4 described above.

3.8 Lau & Gottlieb surface bursts

It was hoped that it would be possible to use the piston trajectory data presented by Lau & Gottlieb (1984) for TNT and ANFO surface bursts, to reconstruct the flow field data required for the compendium, but when these were examined, at least in the TNT case, inconsistencies were again discovered. For example, the piston trajectory data for TNT listed in their table 2 starts at time t=0.4788 ms and ends at t=4.3463 ms, whereas the trajectory shown in their figures 22 and 24 clearly ends at something between t=5.5 and t=6 ms.

Using their tabled piston trajectory data, the computed shock front trajectory was significantly different from the one plotted in their figure, viz., the shock front radius was about 5% larger than theirs at time t=6 ms. The shock front trajectory that they plot presumably comes from their plotted piston trajectory, and not their tabled one, but it is not clear which of their results (if either) is 'correct'.

Because a large part of the experimental shock and particle trajectory data used by Lau & Gottlieb in their reconstruction came from the present authors, it was decided to use these and other data to start from scratch, as when writing the reconstruction program code.

Chapter 4

FREE FIELD AND SURFACE BURST RECONSTRUCTIONS

The piston path-random choice method, described in the previous chapter, was used to reconstruct two flow fields for this report. The first reconstruction describes the free field detonation of one kilogram of TNT in a standard, sea level ambient atmosphere. The reconstructed flow field is spherical, having been created by a spherically expanding piston. The reconstruction was judged successful and no further adjustments to the piston trajectory were made when the peak overpressure vs distance curve which was computed, successfully averaged the free field TNT data gathered from the literature and scaled to the same charge yield and ambient conditions. The agreement obtained is shown in figure 9.

The second flow field which was reconstructed also describes the detonation of one kilogram of TNT, in the same standard atmosphere, but for a hemispherical charge placed on a rigid flat surface. The piston path was adjusted until the peak overpressure profile successfully averaged the surface burst TNT data gathered from the literature, scaled to the same charge yield and ambient conditions. The agreement obtained in this case is shown in figure 10.

The shock front trajectory reconstructed in each case was then compared with shock front trajectories measured in specific experiments. Excellent agreement between trajectories was obtained in all cases, but this is not surprising as the reconstructed and measured trajectories were matched at one point exactly, viz., the position and time at which the piston started moving, and shock front trajectory matching is not as sensitive as peak pressure profile matching, for reasons outlined below.

Once the peak overpressure profiles are matched, the shock front trajectories should also match, because peak overpressure depends on shock front velocity, the slope of the trajectory, in a very sensitive manner, especially at larger radial distances. Peak overpressure varies directly with shock Mach number squared minus one (M^2-1), and small adjustments in a shock front trajectory, and hence in shock speed, can cause non-negligible changes in the peak pressure profile, especially if the shock speed is approaching unity. Small adjustments to a pressure profile, on the other hand, usually cause only negligible changes in the shock trajectory.

Small adjustments to the piston path were nonetheless required after agreement in peak overpressure profile and shock front trajectory were obtained, to obtain agreement between reconstructed and measured data in the flow field behind the shock front. These extra adjustments were always small and easily made.

The blast wave flow fields behind the shock fronts were compared to pressure time histories and particle trajectories measured in experiments in both the free field and surface burst cases. To do this, the reconstructed data were scaled to the yield and conditions of the experiment from which the measurements were taken. This was done using the data retrieval, scaling and interpolation software which will be described in the next chapter of this report.

The agreement obtained between the reconstructed data and pressure gauge records and photographically measured particle trajectories was always satisfactory, to such an extent that non-agreement could be attributed to errors in the experimental data. Typical comparisons between reconstructed and measured data are given in the next chapter.

The regions over which flow field data were computed are shown in figures 11 (free field) and 12 (surface burst), for one kilogram TNT. In both cases the range in radial distance is between one meter, where the piston was initially positioned, and 15 meters, where peak overpressure has dropped to about 1 psi. The time range is over the complete positive phase of the blast waves and slightly into the negative phase, at all but a few distances where the piston arrives before completion of the positive phase. Except at these small distances, the flow field data end approximately at the time of arrival of the secondary shock front, which was not modelled.

The reconstruction calculations were done using a grid of 1500 points to represent a range of distance from zero to 15 meters (points behind the piston are not used, however), and either 1824 (free field) or 1902 (surface burst) time steps between the piston start time and about 40 milliseconds. The reconstructed data were thinned for use with a retrieval and scaling program, by defining 100 representative 'gauge' positions, distributed logarithmically along the position axis. At each of these 100 gauge positions, records or time histories of the reconstructed pressures, densities and flow velocities were kept, rather than at the 1500 calculation grid points. The 100 time histories were then thinned by selecting data at certain times only. The selection time base (117 times for free field and 114 for surface burst data) was built up of intervals whose length increased with the square of time. The final time histories each contain a sufficient number of data points (10 or 12 typically) that linear interpolation can be used satisfactorily. Peak data profiles and shock front trajectories keep all 100 points.

Results presented in this report were obtained from this reduced data base, using the retrieval software and linear interpolation.

4.1 Basic results

The profiles of the peak values of the three basic parameters, pressure, density and flow velocity, are presented in figures 13, 14 and 15, for 1 kg TNT in a standard atmosphere. The free field and the surface burst profiles are plotted in each of these figures.

Representative wave profiles, that is, pressure, density and flow velocity versus radius at a number of fixed times, are presented in figures 16 through 21.

Representative time histories of the same three basic parameters are presented in figures 22 through 27.

Times of shock front arrival as a function of position can be obtained from the shock front trajectories in figures 11 (free field) and 12 (surface burst). Some particle trajectories are shown in figure 28 (free field).

4.2 How to interpret the basic results

Only the basic data which were reconstructed are presented here, viz., pressures, densities and flow velocities, and they are normalized (figures 13 through 27). Specifically, these data are:

static overpressure ratio $\underline{P} = (p-p_0)/p_0$;

density ratio $\underline{D} = \rho/\rho_0$, and

flow velocity Mach number $\underline{U} = u/a_0$,

where p_0, ρ_0 and a_0 are the ambient atmospheric pressure, density and sound speed, respectively. The ambient atmosphere is at rest (u_0=0). $\underline{P}$, $\underline{D}$ and $\underline{U}$ are dimensionless numbers.

Positions, r, are distances from the charge centre, in meters, and times, t, are times after charge detonation, in milliseconds. Distances and times plotted in figures 13-27 describe the detonation of a one kilogram charge of TNT in a standard, sea level atmosphere.

The standard atmosphere is dry air at 15° C. Blast overpressures are everywhere low enough that the ratio of specific heats can be considered constant. The following data are assumed:

ratio of specific heats $\gamma = 7/5$;

standard ambient pressure $p_0 = 101325.00$ Pascals;

standard ambient density $\rho_0 = 1.2250140$ kg/m^3, and

standard ambient sound speed $a_0 = 340.29205$ m/s, so that

$$a_0 = (\gamma \cdot p_0/\rho_0)^{1/2} .$$

To obtain absolute pressures, densities and flow velocities, the definitions of the basic data given above, are rearranged so that:

static pressure $p = (\underline{P}+1) \cdot p_0$;

density $\rho = \underline{D} \cdot \rho_0$, and

flow velocity $u = \underline{U} \cdot a_0$,

where $\underline{P}$, $\underline{D}$ and $\underline{U}$ are overpressure ratio, density ratio and flow velocity Mach number as plotted in figures 13 through 27, and p_0, ρ_0 and a_0 are the standard ambient atmospheric data, defined above in mks units.

To obtain p, ρ and u in British units, the standard ambient data are replaced by $p_0 = 14.696$ psi, $\rho_0 = 0.076475$ lb/ft^3, and $a_0 = 1116.4$ ft/s.

4.3 How to compute other blast data

All properties of the air in the blast waves can be computed from the basic results described above, using the recipes listed below ('•' denotes multiplication and '••', exponentiation; units are mks units):

static overpressure $hop = p-p_0$ (Pa);

dynamic pressure $pd = \rho \cdot u^2/2$ (Pa);

total head-on pressure $pt = p \cdot \left((\gamma-1) \cdot m^2/2 + 1\right) \cdot \cdot \left(\gamma/(\gamma-1)\right)$, when $m^2 = 2/\gamma \cdot pd/p$ is $\leqslant 1$. Otherwise, if $m^2 > 1$:

$$pt = \frac{p \cdot m^2 \cdot \left((\gamma-1)/2\right) \cdot \cdot \left(\gamma/(\gamma-1)\right)}{\left(2 \cdot \gamma/(\gamma+1)-(\gamma-1)/(\gamma+1)/m^2\right) \cdot \cdot \left(1/(\gamma-1)\right)} \quad \text{(Pa)};$$

total overpressure $top = pt-p_0$ (Pa);

sound speed	$a = (\gamma \bullet p/\rho)^{1/2}$	(m/s);
signal speeds	$\alpha^{+} = u + a$ and $\alpha^{-} = u - a$	(m/s);
local Mach number	$m = u/a$ (or use m^2 defined above);	
temperature	$T = 288.15 \bullet (a/a_0)^2$	(K);
excess energy density	$E = (p - p_0)/(\gamma - 1) + \rho \bullet u^2/2$ $= \mathrm{hop}/(\gamma - 1) + \mathrm{pd}$	(J/m^3);
available work density	$W = \mathrm{pt}/(\gamma - 1) \bullet \left(1 - (\mathrm{pt}/p_0) \bullet\bullet ((1 - \gamma)/\gamma)\right)$ $+ p_0 \bullet \left(1 - (\mathrm{pt}/p_0) \bullet\bullet (1/\gamma)\right)$	(J/m^3);
specific entropy change	$\Delta s = 287.05/(\gamma - 1) \bullet \ln\left(\underline{P}/\underline{D} \bullet\bullet \gamma\right)$	(J/kg/K).

The pressures defined above (hop, pd, pt and top) can be normalized by dividing by the ambient pressure, p_0; the sound speed (a) and signal speeds (α^{+} and α^{-}), by dividing by a_0, and the temperature (T), by dividing by 288.15 K. The local flow Mach number, m, is dimensionless.

Excess energy density is the energy per unit volume over and above ambient energy density. Available work is the work that can be done by capturing a unit volume of gas at its stagnation pressure (total head-on pressure) and letting it expand adiabatically and reversibly to ambient conditions (see for example, Dewey & McMillin, 1979, 1985). Energy and work densities, though not dimensionless, are invariant under scaling in the same way that pressure ratios, etc. are invariant.

The relationships above assume Eulerian coodinates so that, for example, the entropy change, Δs, at a position is the change from the pre-shock, ambient value at that position, and is dependent on time. Entropy behind the shock front is constant with time in Lagrangian coordinates, until other shocks arrive.

The above relationships hold everywhere in the flow field behind the shock front. The shock Mach number at any position can be computed using the Rankine-Hugoniot relations and the peak pressure, density or flow velocity at that position, i.e., the value of p, ρ or u just behind the shock front there. For example:

shock front Mach number $M = \left((pmax/p_0-1)\cdot(\gamma+1)/(2\cdot\gamma) + 1\right)^{1/2}$, and

shock front velocity $V = M\cdot a_0$.

Particle trajectory data are presented in figure 28. Particle trajectories represent lines of constant entropy. Their slope at any point equals the particle or flow velocity at that point (dr/dt=u).

Other data which can be obtained from this report are times of positive phase duration (times between shock front arrival and the return to the ambient value of a parameter such as overpressure), and integrated values, either time integrals of pressure (impulses) or volume integrals of energy or work density (total energy or available work). Integration must be done by the reader (integrated data are not compiled here), unless the retrieval software package is used. That software, described in a subsequent chapter, has options for both data interpolation and integration, as well as options for scaling to different charge weights, explosive materials and ambient atmospheric conditions.

4.4 How to scale to other charge weights and atmospheres

The basic data presented here are for one kilogram of TNT in a standard atmosphere, as described above. To apply the results presented in this report to other charge weights and different ambient conditions, the distances and times presented must be scaled. Other data (pressures, densities, etc.) do not need to be scaled, as they are normalized.

Two scale factors must be computed, one for distances and the other for times. The scale factor for distances is

$$S = \left((w_1/w_0)\cdot(p_0/p_1)\right)^{1/3} ,$$

where w_1 is the new charge weight and p_1 is the new atmospheric pressure. The standard charge weight, w_0, and atmospheric pressure, p_0, have been defined above (w_0=1 kg and p_0=101.325 kPa). S=10 for a 1000 kg TNT charge in the standard atmosphere, for example.

The scale factor for times is

$$St = S\cdot a_0/a_1 ,$$

where a_1 is the speed of sound in the new atmosphere. The standard sound speed, a_0, has been defined above (a_0=340.29205 m/s).

The sound speed a_1 depends only on the temperature of the ambient atmosphere and, over the usual range of atmospheric temperatures,

$$a_1 = a_0\cdot\left(T_1/288.15\right)^{1/2} ,$$

where T_1 is the new ambient temperature, in °K. The standard ambient temperature, T_0, is 288.15°K (15°C). An adjustment can be made to the ambient sound speed to account for moisture in the air, but it is not described here as it is small (always less than one half percent).

The relationship between the distances, r, and times, t, presented in figures 13 through 27 and those for the new event defined by w_1, p_1 and a_1 above, is simply

$$r' = r\cdot S \text{ and}$$

$$t' = t\cdot St ,$$

where r' and t' are the new distance and time, respectively. Therefore, the distances and times presented in figures 13 through 27 should be multiplied by the scale factors S and St, to obtain a general description of the new blast wave. However, to obtain particular data it is often more convenient to invert the scaling, that is, divide the distance or time of interest by S or St and use figures 13 through 27 'as is'.

For example, consider a 500 kg TNT charge in an ambient atmosphere with pressure 97 kPa and temperature 20°C (S=8.0532 and St=0.9914). To obtain the free-field pressure time history at a distance r'=100 meters, divide r' by the distance scale factor S and interpolate the data plotted in figure 16 at distance $r'/S = 12.42$ m. Then adjust the interpolated data by multiplying all times by the time scale factor St. The result can then be used as overpressure ratio as plotted in figure 16 vs time, or the overpressures can be made absolute by multiplying the plotted ratios by the new ambient pressure (p_1=97 kPa).

4.5 How to adjust for different explosive materials

A good approximation to the blast waves that result from the detonation of materials other than TNT can be obtained from the TNT data presented here, using simple energy scaling. If the relative energy yield of the new explosive is known, the TNT-equivalent charge weight is calculated using

$$w_1 = F \cdot w_2 ,$$

where F is the TNT equivalence factor and w_2 is the actual weight of the new charge. The computed TNT-equivalent weight, w_1, is then used to compute the scale factors S and St described in the previous section.

TNT equivalence factors have been calculated and/or measured for many explosive materials. Specific energy yields have been measured in bomb calorimeters, and in standard ballistic mortar and plate damage tests. Results of vary according to the method used. For example, Kinney & Graham (1985, table I) list the following percentage energy yields compared to TNT, for a common explosive 'RDX': sand crush test, 130; plate dent test, 135; ballistic mortar test, 150; Trauzl block test, 160, and, Berthelot calculation (fireball), 176.

For use with the data presented in this report, the best method by which explosive yields can be measured is by comparing peak overpressure profiles which have been measured using gauges or by photogrammetry, and matching the profiles using the scaling laws described above.

If two peak pressure profiles are measured over a similar range of pressures, and r_1 and r_2 are the distances in each case where a specific, normalized pressure is measured, then the ratio of charge yields is

$$w_1/w_2 = (r_1/r_2)^3 \cdot (p_{01}/p_{02}) ,$$

where p_{01} and p_{02} are the ambient pressures in the two experiments. The ratio w_1/w_2 is the TNT equivalence factor, F, if the subscript 2 is for TNT. An overpressure ratio of 1 ($p_1/p_{01}= p_2/p_{02}= 1$) is a convenient level at which to match profiles, being the logarithmic midpoint between 10 and 0.1, the approximate pressure limits for this compendium.

Matching at one point on a profile does not guarantee a match over the complete profile, nor does it mean that flow field data behind the shock front are also matched. Different explosives have characteristic peak profiles and time history signatures. Over the range of distances being considered here, however, most explosives do not deviate from TNT so much that simple energy scaling cannot be used to obtain a good approximation, provided that the peak pressure profile is matched near the 'central' 1 atmosphere peak overpressure level.

Several matches of peak pressure profiles have been made. ANFO has been matched to TNT more or less on an equal weight basis (F=1) for both free field (DIRECT COURSE, 1983, 609 tons) and surface burst (DICE THROW, 1976, 628 tons) experiments. Pentolite has been matched frequently to TNT in free field and height-of-burst experiments (F=1.22). Micro-scale height-of-burst experiments (Enrst Mach Institute, 1984) yielded the result F=1.02 for Hexogen.

A large surface burst experiment using a stoichiometric mix of propane and oxygen (DISTANT PLAIN shot 2a, 1966, 20 tons TNT equivalent) yielded the volume equivalence of 1 kg TNT per 1.25 cubic meters of gas.

Nuclear explosions are usually rated in terms of their TNT energy equivalence. However, about one-half of the total energy released in the nuclear case appears in the blast wave. If one matches pressure profiles, therefore, one obtains F=0.5, approximately, for nuclear devices, taking w_2 to be the nominal yield. So for a 'one kiloton' nuclear explosion, w_1 should be set to about 454,000 kg TNT (about 1,000,000 lbs TNT).

Most non-TNT data deviate from TNT data, after profile matching, at the high pressure end (at low radii), if at all. The peak pressure measured 1 meter from a 1 kg ANFO surface burst is about 5 atmospheres greater than that measured for 1 kg TNT at the same radius, and that measured for the equivalent propane-oxygen surface burst is about 5 atmospheres lower than the TNT value (these are worst-case examples). These differences are comparable to the differences in energy yield measured by sand crush tests, ballistic mortar tests, etc.

The most notable difference in time history data occurs in the nuclear explosion case, where there is a much larger negative phase than predicted by the model. This is because a nuclear explosion adds a negligible amount of matter to the atmosphere, compared to the detonation products from a chemical explosion.

Quantitative deviations from TNT are not documented in this report, however.

Chapter 5

THE UVIC/AIRBLAST DATA DISPLAY PACKAGE

The basic data presented in the previous chapter, describing the detonation of 1 kg TNT in a standard atmosphere, were used to create a data base for a 'user-friendly' computer program which interpolates the primary data (static pressures, densities and flow velocities), scales them to a user-specified charge yield and ambient atmospheric conditions, and calculates secondary data (dynamic pressures, etc.) as requested.

The data base and the retrieval and scaling program are supplied on a magnetic tape with this report. Together they are referred to as the UVic/AIRBLAST data display package.

5.1 Display options

The UVic/AIRBLAST program offers three display options. First, the user may request that 'all data at one point' be displayed, then specify a distance from charge centre. The following data will be returned: the time of shock front arrival, the shock Mach number, the peak pressures (static, dynamic and total) and the peak density, flow velocity, temperature, etc. The time of positive duration (static overpressure) and the static overpressure impulse are displayed, if they can be determined. The user is then given the option of asking for all data at specified times after the time of shock arrival, or asking for a complete time history of all pressure data. Other time histories can be obtained using the time history display option described below.

The second display option which is offered is for radial profiles, either of a particular peak parameter or complete waveforms of particular data at user-specified times. Options are given for interpolation in the first case and for interpolation and/or integration in the second case.

The third display option offered is the display of time histories, at user-specified locations, of particular data. Both interpolation and integration options are given. Data contours in the radius-time plane can be computed using this display option and inverse interpolation, as defined below.

5.2 Particular data codes

Particular data displays are requested using integer codes. The available data kinds are listed below with their codes:

0 all data at one point
1 shock front radii
2 times of shock arrival
3 shock front velocities
4 particle trajectories

5 hydrostatic overpressures
6 dynamic pressures
7 total head-on pressures
8 positive durations
9 pressure impulses

10 particle velocities
11 gas densities
12 temperatures
13 sound speeds
14 entropy changes

15 energy densities
16 available work densities
17 energy integrals
18 work integrals

Other data can be computed by the user, such as local Mach numbers (particle velocity divided by sound speed) and signal speeds (particle velocity plus or minus sound speed).

Times of positive duration can be determined by specifying a particular data kind (e.g., static overpressure), and using the time history display and inverse interpolation options. Inverse interpolation determines the time corresponding to a given parameter value, when the latter are listed as a monotonic function of the former. Normal interpolation determines the parameter value corresponding to a given time.

Pressure impulses are determined by specifying a particular data kind (e.g., static overpressure), and using the time history display and integration options.

The same holds true for volume integrals: the user can specify a particular data kind, for example, energy density, and use the wave profile display and integration options, to obtain, for example, total energy in the positive phase.

Neither times of positive duration nor pressure impulses, nor total energies, etc., have been compiled as such, as the system just described is both easy to use and more flexible.

Particle trajectories are not supplied with the present data base, but can be obtained by request. The size of the data base source files will be increased by approximately 30%, if this option is required.

5.3 Charge materials other than TNT

The current program uses the TNT data base to simulate data that might be expected from other HE charges (ANFO, for example), or from a nuclear device, by simple energy scaling. The TNT yield-equivalence factors which are used, were derived by matching peak pressure profiles as described in the previous chapter. The yield-equivalence factor is displayed to the user when a non-TNT charge material is selected.

5.4 How to use the program

The program is intended to be run interactively from a computer terminal. It was developed using IBM's Conversational Monitor System (CMS), and should run on any similar operating system. Presently, the program source code is being supplied on magnetic tape and it will need to be compiled. The source language is FORTRAN 77.

The compiled program must be loaded and started in accordance with the host computer's operating system. Two data files must be defined at load time, to be available for input at execution time. The data are supplied with the program. The two files are:

file 8 = free field data, and
file 9 = surface burst data.

The program also reads from file 5 and writes to file 6, both of which are normally defaulted as the user's terminal. Requested data can be displayed at the terminal or saved as file 1 (optional) for subsequent printing and/or plotting. File 1, if it is used, may be defined either at program load time or, interactively, at execution time.

The current version of the program uses 264.7k bytes of computer storage (IBM CMS). This includes 25.7k bytes of array space for the data from either file 8 or file 9 and work space. Only one set of data resides in storage at any one time. It is input when the charge-ground configuration is specified.

Execution is interactive, the program prompting the user for input, usually after presenting to the user a menu of options. The program is supposed to be self-explanatory. The following notes may be helpful:

-General program information is available at execution time (a description of what the program does and reminders about how to exit/restart the program). The display of this information can be suppressed by entering a null line (defined below) after the offer to provide information is written on the terminal screen.

-The input of option indices other than those listed in a menu may produce unpredictable results. In most cases, however, an attempt has been made to provide a default option or to branch so that a recovery can be made.

-Alphabetic characters should never be input when numeric input is requested. Unpredictable results may occur.

-Decimal points are not required with input of real numbers, though they may of course be included: '12', '12.' and '12.0' are all read the same. Integers are read with or without blanks: blanks are not zeros. Trailing blanks and commas are delimiters, and vectors can be input from one or more lines: '3.4 0' or '3.4, 0' from one line, or '3.4' from one line then '0' from the next. All data input is done using the FORTRAN 'READ(5,*,END=m) list' statement.

-To exit the program, a null line can be entered instead of data (hit the enter/return key without entering any data or blanks), anytime in a run. A program restart option is offered before the exit.

-A null line must sometimes be entered to get the program to continue executing, at times when no specific data are being sought. Pausing in this way gives the user a chance to clear his or her screen, so that displays of the output data needn't be split between screens.

-Interpolation and extrapolation of the retrieved data can be done in a normal manner (find 'y' at a given 'x') or in an inverse manner (find 'x' for a given 'y'), the data being monotonic.

-Excessive extrapolation produces a warning message.

5.5 Program output

The UVic/AIRBLAST program displays data at the user's terminal or outputs the data to disk (output file number 1), as requested by the user. Output to disk, the data can be reviewed, printed or input to a plot program at some later time.

Examples of data which were obtained using the UVic/AIRBLAST display program are shown in figures 29 and 30. It was by using comparisons such as these that the piston paths were 'fine tuned' when the blast waves were numerically reconstructed, after the peak pressure profiles were matched.

5.6 Description of the tape supplied (UVic AIRBLAST/20feb87)

The tape is non-labelled, ASCII and written at 1600 bpi. It has fixed-length, 80-character records, blocked at 8000 bytes. It contains the following three files:

1 'AIRBLAST FORTRAN' = 2193 records of program source code

2 'FFT DATA' = 464 " " free field TNT data

3 'SBT DATA' = 435 " " surface burst TNT data

5.7 Listings

A listing of the UVic/AIRBLAST program source code is supplied here as appendix A (a listing of the file AIRBLAST FORTRAN). Listings of the data files FFT DATA and SBT DATA are included as appendices B and C.

Appendix D is a listing of the 'console record' for a typical session using the UVic/AIRBLAST package. It is a listing of program prompts and user responses. Appendix E is a listing of the output disk file created in the session (file number 1).

Chapter 6

FUTURE EXTENSIONS

Work was done during the course of this project towards extending the UVic/AIRBLAST data base and data display program. Subroutines were written and successfully tested, which can be used with the program and the free field data described here to display the peak parameters at the ground surface behind the reflected shock front for a height-of-burst configuration defined by the user, in the regular reflection region before transition to Mach reflection.

It was assumed that over a small region near the reflection point at the ground surface, the primary shock and reflected shocks can be treated as if they were plane shocks. Plane two-shock theory was used to compute the reflected shock front angle and strength, and the peak pressure, etc. immediately beind the reflected shock front. These data were computed over the range of distances between ground zero (normal reflection at the point directly below the charge) and the 'detachment' point, slightly ahead of actual transition to Mach reflection.

It was shown that flow fields in the Mach stem region could be reconstructed using the same piston path-random choice method used to obtain the free field and surface burst data presented here. Pressure time histories and particle trajectories which were measured near the ground surface were successfully reproduced using a pseudo-one-dimensional numerical model driven by a spherical piston, started just past the transition point to Mach reflection. This has never been shown before.

Two Mach stem flows were successfully reconstructed, one for a comparatively low height of burst, 0.5 meters on the 1 kg TNT scale, and the other, for the highest height of burst for which data could be found, 3.4 meters on the same scale. It is likely that flows for intermediate heights of burst can be simulated equally well, and experimental data have been gathered for the intermediate heights. It is anticipated that ten selected heights can be used to represent the full range of heights of burst, and that the corresponding Mach stem flow fields added to the

UVic/AIRBLAST data base will permit interpolation for arbitrary heights of burst.

The two Mach stem flows successfully reconstructed were those for the DIRECT COURSE experiment (1983) and DIPOLE WEST shot 24 (1975). In the first experiment 609 tons (8.5×10^5 kg) of ANFO were detonated at a height of 166 feet (50.6 m, or 0.5 m on the 1 kg TNT scale). In the second, 1000 lbs (454 kg) of TNT were detonated at a height of 90 ft (27.4 m, or 3.4 m on the 1 kg TNT scale).

The DIRECT COURSE Mach stem flow field was reconstructed between ground radii of 175 m and 850 m, or 2 m and 10 m on the 1 kg TNT scale. Agreement of 1% or better was obtained for the radial positions of the shock front and 28 particle tracers. Similar agreement was obtained between computed pressures (static and dynamic) and densities, and the corresponding gauge data. It was shown that the reconstructed data could be extrapolated beyond the position of the last observed particle tracer to obtain good agreement with gauge data. Results of this reconstruction were presented at the 6th Mach Reflection Symposium, in Beer Sheva, Israel in 1986.

The DIPOLE WEST/24 Mach stem was reconstructed between ground radii of 85 m and 108 m, or 10.6 m and 13.5 m on the 1 kg TNT scale. Excellent agreement was obtained with the 10 particle trajectories measured closest to the ground, in the Mach stem region below the low triple point trajectory. Agreement was better than 1%.

Details concerning the Mach stem reconstructions were given in the various interim reports provided earlier. They are not repeated here.

With complementary data for the regular reflection region, computed as described above, it should be possible to describe the flow near the ground surface for any height-of-burst, from ground zero to the limit of known experimental measurements, or slightly beyond.

The UVic/AIRBLAST package was successfully implemented on a personal computer (an IBM PC 'clone'). The program changes which were required for this were relatively minor ones and it should be possible to provide source code for virtually all of the common micro-computers.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the Department of National Defence, Canada which provided major funding for this project. The help and understanding of Mr. D. Ritzel throughout the project is greatly appreciated. Additional funding for the project was provided by the National Sciences and Engineering Research Council, the University of Victoria and the Defense Nuclear Agency. Dr. W. Heilig's valuable contribution in the provision and interpretation of results from micro-scale experiments was made possible by a grant from NATO. The development of the UVic/AIRBLAST program and its implementation on a micro-computer were done by Mr. G. Cross with the support of a British Columbia Challenge '86 award. Many researchers and laboratories were generous in their provision of experimental results, and the help of Mr. R.E. Reisler, then of the Ballistic Research Laboratory, is particularly acknowledged. Special thanks are offered to Dr. J.J. Gottlieb of the University of Toronto Institute for Aerospace Studies. Dr. Gottlieb first introduced us to the Piston Path-Random Choice Method, and the program listings and helpful advice which he provided on several occasions were essential to the successful completion of the project.

BIBLIOGRAPHY

Anson, W.A., & Dewey, J.M. 1965. Density Measurements in the Blast Wave from a Surface Burst 500-ton TNT Hemispherical Charge. Technical Paper No. 305, Defence Research Establishment Suffield, Ralston, Alta.

Anderson, J.H.B. 1974. 1969 1000-lb Airburst Series: Pressure-Time and Time-of-Arrival Results. Memorandum No. 48/73, Defence Research Establishment Suffield, Ralston, Alta.

Anderson, J.H.B., & Deckker, B.E.L. 1977. Explosion near the ground: a solution to the multi-shock problem in cylindrical symmetry by the particle trajectory method. Proc. Roy. Soc. Lond. A 358, 31-46.

ANSI (American National Standard). 1983. Estimating Air Blast Characteristics for Single Point Explosions in Air, with a Guide to Evaluation of Atmospheric Propogation and Effects. ANSI S2.20-1983, Standards Secretariat, Acoustical Society of America, New York, NY.

Bach, G.G., & Lee, J.H.S. 1970. An Analytic Solution for Blast Waves. AIAA J. 8, 2, 271-275.

Baker, W.E. 1973. Expolsions in Air. University of Texas Press (1st printing); Wilfred Baker Engineering, San Antonio (2nd printing, 1983).

Baker, W.E., Cox, P.A., Westine, P.S., Kulesz, J.J., & Strehlow, R.A. 1983. Fundamental Studies in Engineering, Vol. 5: Explosion Hazards and Evaluation. Elsevier Scientific Publishing, New York, NY.

Bethe, H.A., Fuchs, K., von Neumann, J., Peierls, R., & Penney, W.G. 1944. Shock Hydrodynamics and Blast Waves. AECD-2860, Los Alamos Scientific Lab., U. Calif., Los Alamos, NM.

Boris, J.P., & Book, D.L. 1973. Flux-corrected Transport: I. SHASTA, a Fluid Transport Code That Works. J. Comput. Phys. 11, 38.

Bowden, J.R., Manson, N., Oppenheim, A.K., & Soloukhin, R.I., eds. 1983. Shock Waves, Explosions and Detonations. Progress in Astronautics and Aeronautics 87, American Institute of Aeronautics and Astronautics, New York, NY.

Boyer, D.W., Brode, H.L., Glass, I.I., & Hall, J.G. 1958. Blast From a Pressurized Sphere. Report No. 48, University of Toronto Institute for Aerospace Studies, Toronto, Ont.

Boyer, D.W. 1959. Spherical Implosions and Explosions. Report No. 58, University of Toronto Institute for Aerospace Studies, Toronto, Ont.

Brode, H.L. 1955. Numerical Solutions of Spherical Blast Waves. J. Appl. Phys. 26, 6, 766-775.

Brode, H.L. 1956. Point Source Explosion in Air. Research Memorandum RM-1824-AEC, The Rand Corp., Santa Monica, Ca.

Brode, H.L. 1957. A Calculation of the Blast Wave from a Spherical Charge of TNT. Research Memorandum RM-1965. The Rand Corp., Santa Monica, Ca.

Brode, H.L. 1959. Blast Wave from a Spherical Charge. Phys. Fluids 2, 2, 217-229.

Bryant, E.J., Eberhard, R.A., & Kingery, C.N. 1952. Mach Reflection Over Hard Packed Dirt and Dry Sand. Report No. 809, Ballistic Research Laboratory, Aberdeen Proving Ground, Md.

Carpenter, H.J., & Brode, H.L. 1975. Height of Burst Blast at High Overpressure. Internal Report, R & D Associates, Santa Monica, Ca.

Celmins, A. 1981. Reconstruction of a Blast Field from Selected Pressure Observations. Proc. 7th Intl. Symp. Military Applications of Blast Simulation, Medicine Hat, Alta, pp. 2.5-1 through 2.5-17.

Chorin, A.J. 1976. Random Choice Solutions of Hyperbolic Systems. J. Comput. Phys. 22, 517-533.

Chorin, A.J. 1977. Random Choice Methods with Applications to Reacting Gas Flow. J. Comput. Phys. 25, 253-272.

Colella, P. 1982. Glimm's Method for Gas Dynamics. SIAM J. Sci. Stat. Comp. 3, 76-110.

Colella, P., & Woodward, P.R. 1984. The Piecewise Parabolic Method (PPM) for Gas Dynamical Calculations. J. Comput. Phys. 54, 174-201.

Colella, P. 1985. A Direct-Eulerian MUSCL Scheme for Gas Dynamics. SIAM J. Sci. Stat. Comp. 6, 1, 104-117.

Colella, P., & Glaz, H.M. 1985. Efficient Solution Algorithms for the Riemann Problem for Real Gases. J. Comput. Phys. 59, 2, 264-289.

Courant, R., & Friedrichs, K.O. 1948. Supersonic Flows and Shock Waves. Interscience, New York, NY.

Dewey, J.M. 1963. The Air Velocity and Density in Blast Waves from TNT Explosions. Ph.D. Thesis, University of London.

Dewey, J.M. 1964. The Air Velocity in Blast Waves from from TNT Explosions. Proc. Roy. Soc. Lond. A 279, 366-385.

Dewey, J.M. 1971. The Properties of a Blast Wave Obtained from an Analysis of the Particle Trajectories. Proc. Roy. Soc. Lond. A 324, 275-299.

Dewey, J.M., & Anson, W.A. 1963. A Blast Wave Density Gauge Using Beta-Radiation. J. Sci. Instr. 40, 568-572.

Dewey, J.M., Classen, D.F., & McMillin, D.J. 1975. Photogrammetry of the Shock Front Trajectories on Dipole West Shots 8, 9, 10 and 11. DNA 3777F, Defense Nuclear Agency, Washington, DC.

Dewey, J.M., Heilig, W., & Reichenbach, H. 1985. Height-of-Burst Results from Small Scale Explosions. Proc. 9th Intl Symp. Military Applications of Blast Simulation, Oxford, U.K.

Dewey, J.M., McMillin, D.J., & Classen, D.F. 1977. Photogrammetry of Spherical Shocks from Real and from Ideal Surfaces. J. Fluid Mech. 81, 707-717.

Dewey, J.M., & McMillin, D.J. 1977. High Speed Photography of Real and Ideal Reflections of Blast Waves. Proc. 12th Intl Symp. Cong. High Speed Photography and Photonics, Toronto, Ont. (SPIE 97).

Dewey, J.M., & McMillin, D.J. 1977. The Effect of the Ground Surface and Charge Size on the Mach Reflection of Spherical Blast Waves. Proc. 5th Intl Symp. Military Applications of Blast Simulation, Stockholm, Sweden, pp. 6:4-1 through 6:4-13.

Dewey, J.M., & McMillin, D.J. 1977. Energy Density Measurements in Blast Waves. Proc. 6th Intl Symp. Military Applications of Blast Simulation, Cahors, France, pp. 1.3-1 through 1.3-12.

Dewey, J.M., & McMillin, D.J. 1977. Visualization of Shock and Blast Waves. Proc. 1st Intl Symp. Flow Visualization, Tokyo, Japan, 199-204.

Dewey, J.M., & McMillin, D.J. 1978. Energy Density Calculations: Dipole West Shot 11. ARBRL-CR-00395, Ballistic Research Laboratory, Aberdeen Proving Ground, Md.

Dewey, J.M., & McMillin, D.J. 1979. Measurements of Energy Density in Blast Waves. Proc. 6th Intl. Symp. Military Applications Blast Simulation, Cahors, France, pp. 1.3-1 through 1.3-12.

Dewey, J.M., & McMillin, D.J. 1979. Measurements of Energy Density in Shock and Blast Waves. Proc. 12th Intl Symp. Shock Tubes and Waves, Jerusalem, Israel, 695-703.

Dewey, J.M., & McMillin, D.J. 1981. An Analysis of the Particle Trajectories in Spherical Blast Waves Reflected from Real and Ideal Surfaces. Can. J. Phys. 59, 1380-1390.

Dewey, J.M., & McMillin, D.J. 1981. The Properties of a Blast Wave Produced by a Large-scale Detonable Gas Explosion. Proc. 7th Intl Symp. Military Applications of Blast Simulation, Medicine Hat, Alta, pp. 6.6-1 through 6.6-18.

Dewey, J.M., & McMillin, D.J. 1983. Photogrammetric Analysis of Multi-Burst Shots: Mighty Mach IV. ARBRL-CR-00515, Ballistic Research Laboratory, Aberdeen Proving Ground, Md.

Dewey J.M., & McMillin, D.J. 1985. Efficiency of Energy Conversion Using Shock Waves. Can. J. Phys. 63, 339-345.

Dewey, J.M., & McMillin, D.J. 1986. Analysis of Particle Trajectories in the Mach Stem (6320). Proc. Direct Course Symp./April 1984, 139-158. POR 7119-9, Defense Nuclear Agency, Washington, DC.

Dewey, J.M., McMillin, D.J., & Trill, D. 1977. Photogrammetry of the Particle Trajectories on Dipole West Shots 8, 9, 10 and 11. Vol. I - Shot 10. DNA 4326F-1, Defense Nuclear Agency, Washington, DC.
------------------ 1977. Vol. II - Shot 9. DNA 4326F-2.
------------------ 1978. Vol.III - Shot 8. DNA 4326F-3.
------------------ 1978. Vol. IV - Shot 11. DNA 4326F-4.

__________________ 1979. Vol. V - Comparison of Results. DNA 4326F-5.

__________________ 1979. Addendum to Vols I & II - Hydrostatic and Total Pressures Compared to Gauge Results.

Field, P. 1982. Handbook of Powder Technology, Vol. 4: Dust Explosions. Elsevier Scientific Publishing, New York, NY.

Gioulietti, A., Vaselli, M., & Giammanco, F. 1980. Shadowgraphic Detection of Spherical Implosive Shock Fronts Produced by Laser Pulses. Optics Comm. 33, 3, 257-261.

Glasstone, S., ed. 1957. The Effects of Nuclear Weapons. U.S. Atomic Energy Commission, Washington, DC. Revised 1962, and 1977, with P.J. Dolan ed., U.S. Dept. of Defense (and) Energy Research and Development Administration, Washington, DC.

Glimm, J. 1965. Solutions in the Large for Nonlinear Hyperbolic Systems of Equations. Comm. Pure Appl. Math. 18, 697-715.

Godunov, S.K. 1959. Finite Difference Methods for Numerical Computation of Discontinuous Solutions of the Equations of Fluid Dynamics. Mat. Sobornik 47, 271.

Goodman, H.J. 1960. Compiled Free-Air Blast Data on Bare Spherical Pentolite. Report No. 1092, Ballistic Research Laboratory, Aberdeen Proving Ground, Md.

Goresch, J.W. & Dunn, R.G. 1969. Tables of Blast Wave Parameters, Vol. I: Spherical Explosions. ARL 69-0011, Aerospace Research Laboratories, Office of Aerospace Research, U.S. Air Force.

Gottlieb, J.J., & Ritzel, D.V. 1979. Flow Properties of a Spherical Blast Wave. Proc. 6th Intl. Symp. Military Applications of Blast Simulation, Cahors, France. pp. 1.4-1 through 1.4-29.

Gottlieb, J.J., & Saito, T. 1983. An Analytical and Numerical Study of the Interaction of Rarefaction Waves with Area Changes in Ducts. Part 1: Area Reductions. Report No. 272, University of Toronto Institute for Aerospace Studies, Toronto, Ont. (for Part 2, see Igra, Gottlieb & Saito, 1984).

Groth, C.P.T. 1985. Quick Reproduction of Blast-Wave and Flow-Field Properties of Nuclear, TNT, and ANFO Explosions. Technical Note No. 259, University of Toronto Institute of Aerospace Studies, Toronto, Ont.

Hasegawa, K., & Sato, K. 1983. A Study of the Blast Wave from Deflagrative Explosions. Fire Safety Journal 5, 265-274.

Held, M. 1983. Blast Waves in Free Air. Propellants, Explosives and Pyrotechnics 8, 1-7.

Henrych, J. 1979. Developments in Civil Engineering, Vol. 1: The Dynamics of Explosion and Its Use. Elsevier Scientific Publishing, New York, NY.

Hu, T.C.J., and Glass, I.I. 1986. Blast Wave Reflection from a Height-of-Burst (HOB). Private communication. (Report in press, University of Toronto Institute for Aerospace Studies, Toronto, Ont.)

Igra, O., Gottlieb, J.J., & Saito, T. 1984. An Analytical and Numerical Study of the Interaction of Rarefaction Waves with Area Changes in Ducts. Part 2: Area Enlargements. Report No. 273, University of Toronto Institute for Aerospace Studies, Toronto, Ont.

Jack, W.H. Jr. 1963. Measurements of Normally Reflected Shock Waves From Explosive Charges. Memorandum Report No. 1499, Ballistic Research Laboratory, Aberdeen Proving Ground, Md.

Kennedy, W.D. 1946. Effects of Impact and Explosion, Vol 1: Explosions and Explosives in Air. Summary Tech. Rept, Div. 2, NDRC, Washington, DC.

Kingery, C.N. 1966. Air Blast Parameters vs Distance for Hemispherical TNT Surface Bursts. Report No. 1344, Ballistic Research Laboratory, Aberdeen Proving Ground, Md.

Kingery, C.N. 1968. Parametric Analysis of Sub-kiloton Nuclear and High Explosive Blast, Report No. 1393, Ballistic Research Laboratory, Aberdeen Proving Ground, Md.

Kingery, C.N., & Pannill, B.F. 1964. Peak Overpressure vs Scaled Distance for TNT Surface Bursts (Hemispherical Charges). Memorandum Report No. 1518, Ballistic Research Laboratory, Aberdeen Proving Ground, Md.

Kinney, G.F., & Graham, K.J. 1985. Explosive Shocks in Air. Springer-Verlag, New York, NY.

Klimkin, V.F., Soloukhin, R.I., & Wolansky, P. 1973. Initial Stages of a Spherical Detonation Directly Initiated by a Laser Spark. Combustion & Flame 21, 111-117.

Lapidus, A. 1967. A Detached Shock Calculation by Second-Order Finite Differences. J. Comput. Phys. 2, 154.

Lau, S.C.M., & Gottlieb, J.J. 1984. Numerical Reconstruction of Part of an Actual Blast-Wave Flow Field to Agree with Available Experimental Data. Technical Note No. 251, University of Toronto Institute for Aerospace Studies, Toronto, Ont.

Lewis, C.H. 1961. Plane, Cylindrical, and Spherical Blast Waves Based Upon Oshima's Quasi-Similarity Model. AEDC-TN-61-157, Arnold Engineering Development Center, Air Force Systems Command, U.S.A.F.

Lutsky, M. 1965. Theoretical vs Experimental Results for Air Blast from One-Pound Spherical TNT and Pentolite at Sea Level Conditions. NOLTR 65-67, U.S. Naval Ord. Lab, White Oak, Md.

MacCormack, R. 1971. Lecture Notes in Physics No. 8. Springer-Verlag, New York, NY.

Makino, R.C., & Shear, R.E. 1961. Unsteady Spherical Flow Behind a Known Shock Line. Report No. 1154, Ballistic Research Laboratory, Aberdeen Proving Ground, Md.

Miura, H., & Glass, I.I. 1982. On a Dusty Gas Shock Tube. Proc. Roy. Soc. Lond. A 382, 373-388.

Miura, H., & Glass, I.I. 1983. On the Passage of a Shock Wave Through a Dusty-Gas Layer. Proc. Roy. Soc. Lond. A 385, 85-105.

Miura, H., & Glass, I.I. 1984. Formation of a Shock Wave in a Dusty Gas. Report No. 275, University of Toronto Institute for Aerospace Studies, Toronto, Ont.

Miura, H., Saito, T., & Glass, I.I. 1984. Normal Reflection of a Shock Wave at a Rigid Wall in a Dusty Gas. Report No. 274, University of Toronto Institute for Aerospace Studies, Toronto, Ont.

Needham, C.E., & Crepeau, J.E. 1981. The DNA Nuclear Blast Standard (1 kt). DNA 5648T, Defense Nuclear Agency, Washington, DC.

Olivier, H., & Gronig, H. 1985. The Random Choice Application to Two-Dimensional Shock Focusing and Diffraction. J. Comput. Phys. 63, 85-106.

Reisler, R.E. 1976. H.E. Height-of-Burst Blast at Moderate to Low Overpressures. Private communication. Ballistic Research Laboratory, Aberdeen Proving Ground, Md.

Reisler, R.E., Giglio-Tos, L., & Kellner, R.C. 1966. Ferris Wheel Series, Flat Top Event, Project Officers Report - Project 1.1 Airblast Phenomena. POR-3001, Ballistic Research Laboratory, Aberdeen Proving Ground, Md.

Sadek, H.S.I., & Gottlieb, J.J. 1983. Initial Decay of Flow Properties of Planar, Cylindrical and Spherical Blast Waves. Technical Note No. 244, University of Toronto Institute of Aerospace Studies, Toronto, Ont.

Saito, T., & Glass, I.I. 1979. Applications of Random-Choice Method to Problems in Shock and Detonation-Wave Dynamics. Report No. 240, University of Toronto Institute for Aerospace Studies, Toronto, Ont.

Sakurai, A. 1965. Blast Wave Theory. Blast Development in Fluid Dynamics (M. Holt, ed.), Vol. I, 309-375. Academic Press, New York, NY.

Schleuter, S.D., Hippensteel, R.G., & Armendt, R.F. 1965. Measurements of Air Blast Parameters Above a Reflecting Surface. Memorandum Report No. 1645, Ballistic Research Laboratory, Aberdeen Proving Ground, Md.

Sedov, L.I. 1946. (Propagation of Intense Blast Waves.) Priklednaya Matematika i Mekhanika 10, Moscow, USSR. Revised in 1947, Similarity and Dimensional Methods in Mechanics, Gostekhizdat, Moscow, and translated in 1959, by M. Friedman (M. Holt, ed.), Academic Press, New York, NY.

Shardin, H. 1954. Measurement of Spherical Shock Waves. Comm. Pure & Appl. Math. 7, 223-243.

Shi, Z.C., & Gottlieb, J.J. 1986. Random-Choice-Method Solutions for Two-Dimensional Planar and Axisymmetrical Supersonic Flows. Report No. 297, University of Toronto Institute for Aerospace Studies, Toronto, Ont.

Sod, G.A. 1977. A Numerical Study of a Converging Cylindrical Shock. J. Fluid Mech. 83, 785-794.

Sod, G.A. 1978. A Survey of Several Finite Difference Methods for Systems of Nonlinear Hyperbolic Conservation Laws. J. Comput. Phys. 27, 1-31.

Strehlow, R.A., & Baker, W.E. 1975. The Characterization and Evaluation of Accidental Explosions. NASA CR 134779, NASA Lewis Research Center, Cleveland, Ohio.

Stoner, R.G., & Bleakney, W. 1948. The Attenuation of Spherical Shock Waves in Air. J. Appl. Phys. 19, 7, 690-678.

Taylor, G.I. 1941. The Formation of a Blast Wave by a Very Intense Explosion. British Report RC-210. Revised version, see Taylor 1950.

Taylor, G.I. 1946. The Air Wave Surrounding an Expanding Sphere. Proc. Roy. Soc. Lond. A 136, 273-292.

Taylor, G.I. 1950. The Formation of a Blast Wave by a Very Intense Explosion, Part I: Theoretical Discussion. Proc. Roy. Soc. Lond. A 201, 159-174. Part II: The Atomic Explosion of 1945. Ibid, pp. 175-186.

Teng, Lui & Chorin. 1982. Riemann Problems for Reacting Gas. SIAM J. Appl. Math. 42, 964.

Thornhill, C.K. 1960. Explosions in Air. ARDE Memorandum (B) 57/60, Armament Research and Development Establishment, Seven Oaks, Kent, UK.

van Leer, B. 1977. Towards the Ultimate Conservative Difference Scheme. IV. A New Approach to Numerical Convection. J. Comput. Phys. 23, 276.

van Leer, B. 1979. Towards the Ultimate Conservative Difference Scheme. V. A Second Order Sequel to Godunov's Method. J. Comput. Phys. 32, 101-136.

van Leer, B., & Woodward, P.R. 1979. (Title not available.) Proc. TICOM Conf., Austin, Texas.

von Neumann, J. 1941. The Point Source Solution. NDRC Division B Report No. AM-9. Since published in Bethe, et al, 1944 (referenced above), and in Collected Works, Vol. IV, 219-237, Pergammon, New York.

von Neumann, J., & Richtmeyer, R.D. 1950. A Method for the Numerical Calculation of Hydrodynamic Shocks. J. Appl. Phys. 21, 232-243.

Woodward, P.R., & Colella, P. 1981. Lecture Notes in Pysics No. 141. Springer-Verlag, New York, NY.

Woodward, P.R., & Colella, P. 1984. The Numerical Simulation of Two-Dimensional Fluid Flow with Strong Shocks. J. Comput. Phys. 54, 115-173.

Worsfold, W.E., & Rees, N.J.M. 1986. The Variation of Pressure on the Ground with Height of Burst of High Explosive Charges. AWRE Rept No. O 9/86. Atomic Weapons Research Establishment, Aldermaston, U.K.

Zhang, K.Y., & Gottlieb, J.J. 1986. Simulation of a Blast Wave Flow in a Shock Tube by Using Perforated Plates in the Driver. Report No. 304, University of Toronto Institute for Aerospace Studies, Toronto, Ont.

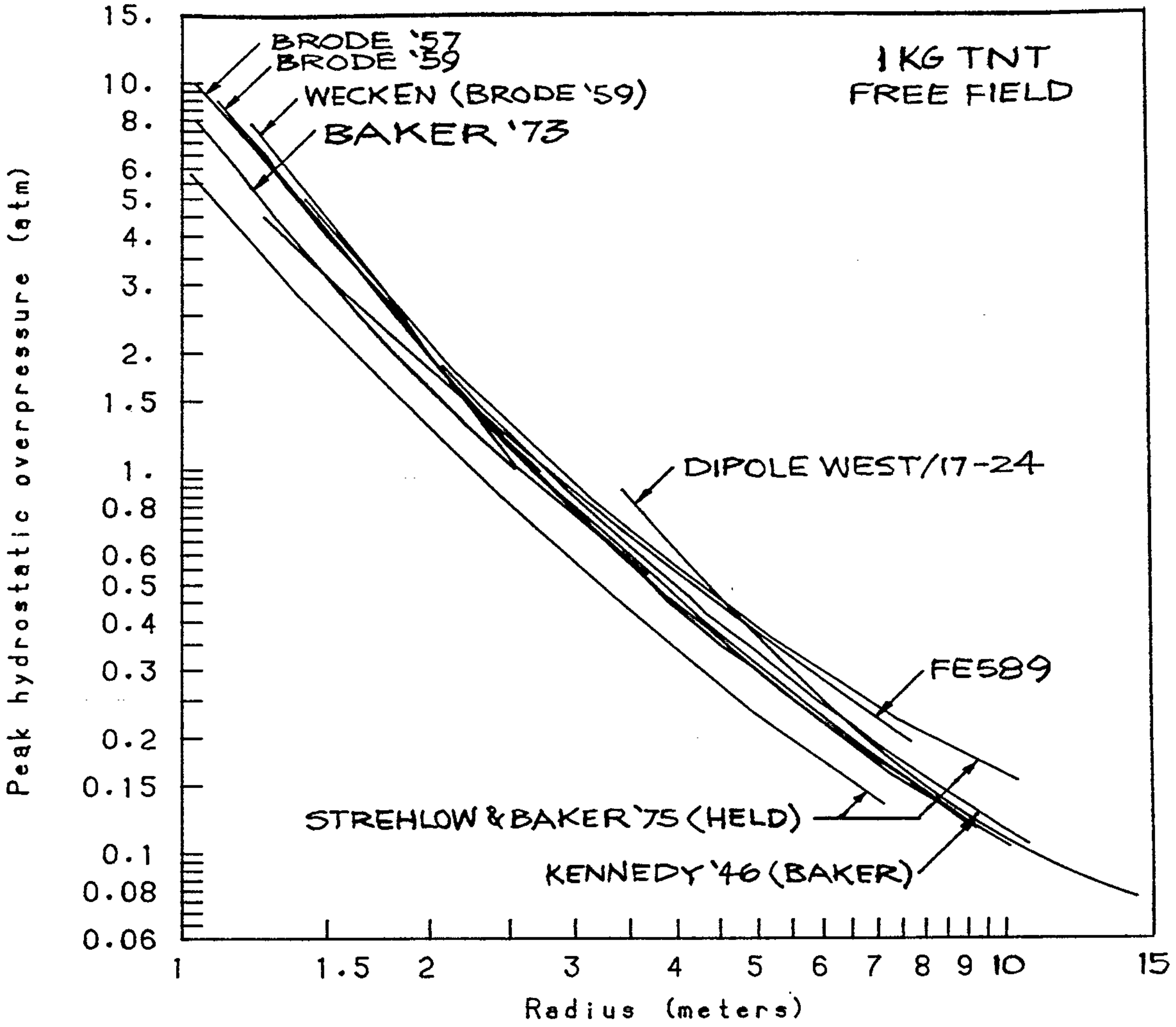

Figure 1a. Peak pressure profiles for free field TNT, scaled to a one kilogram charge. The sources from which the data were taken are given. Strehlow & Baker (1975), taken from Held (1983), defined upper and lower limits on their free field profiles, as indicated. The DIPOLE WEST and FE589 results were obtained by the authors at the University of Victoria.

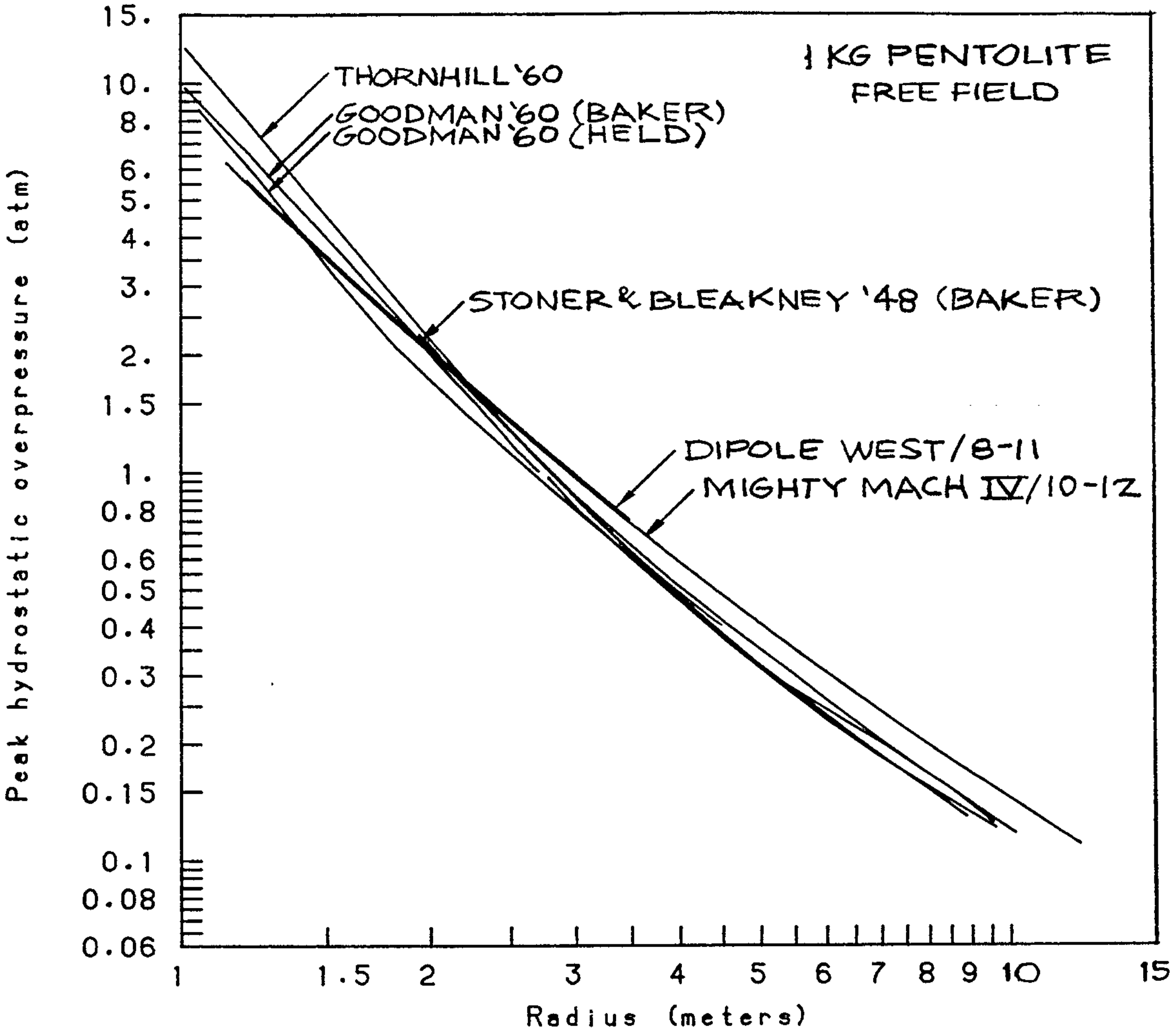

Figure 1b. Peak pressure profiles for free field Pentolite, scaled to a one kilogram charge. Pentolite is an explosive similar to TNT, but about 20% more energetic. The DIPOLE WEST and MIGHTY MACH IV results were obtained by the authors at the University of Victoria.

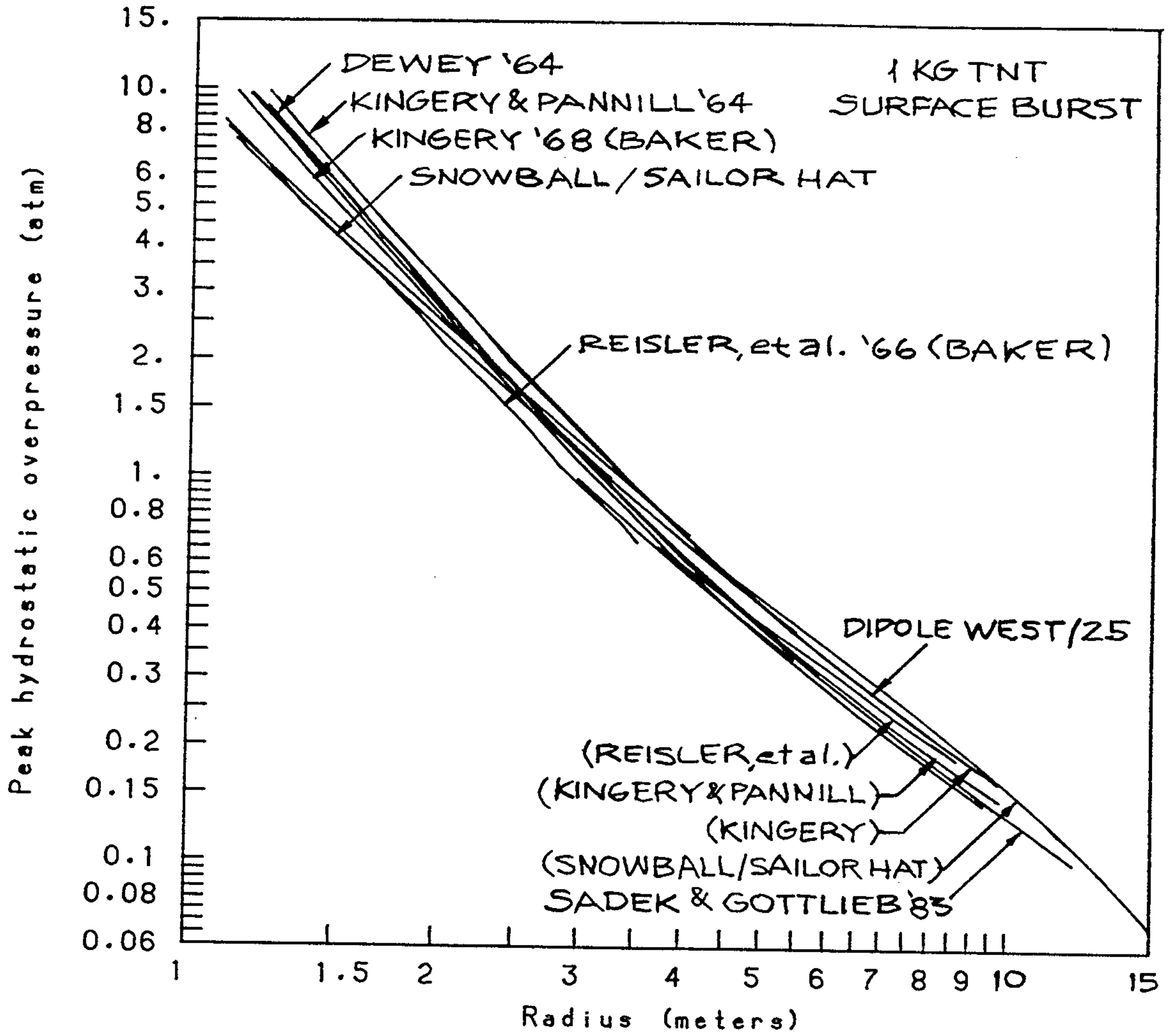

Figure 2. Peak pressure profiles for surface burst TNT, scaled to a one kilogram charge. The DIPOLE WEST, SNOWBALL and SAILOR HAT data were obtained by the authors at the University of Victoria.

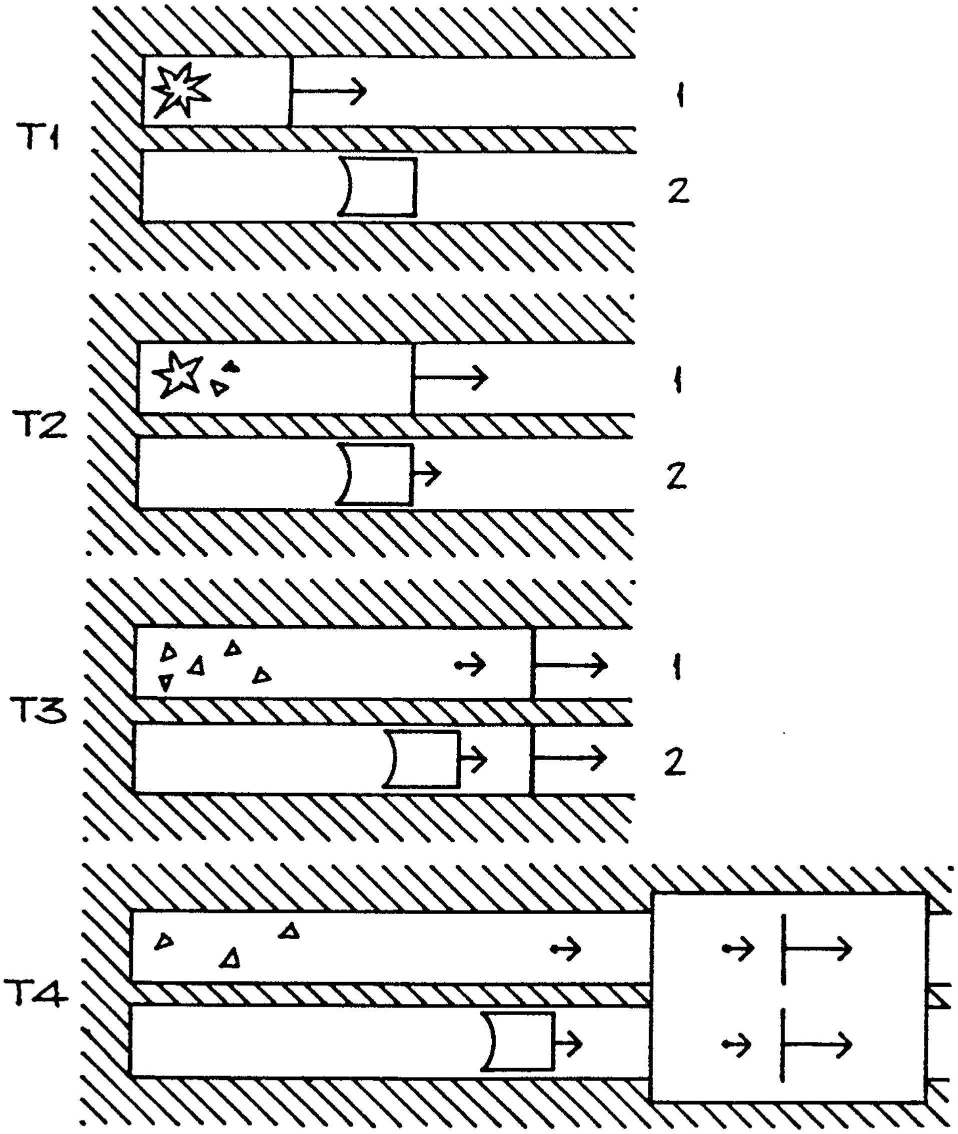

Figure 3. The piston path method. At time T1 a shock is explosively generated in tube 1. The shock front has not yet travelled as far as the stationary piston in tube 2. The shock reaches the piston at time T2 and the piston is impulsively set into motion. The piston's speed is at all times the same as that of the gas flow behind the shock in tube 1. To an observer at any later time, T4, the shocks and subsequent flows generated by the explosion and the piston are indistinguishable. The piston driven flow is easier to simulate numerically, however.

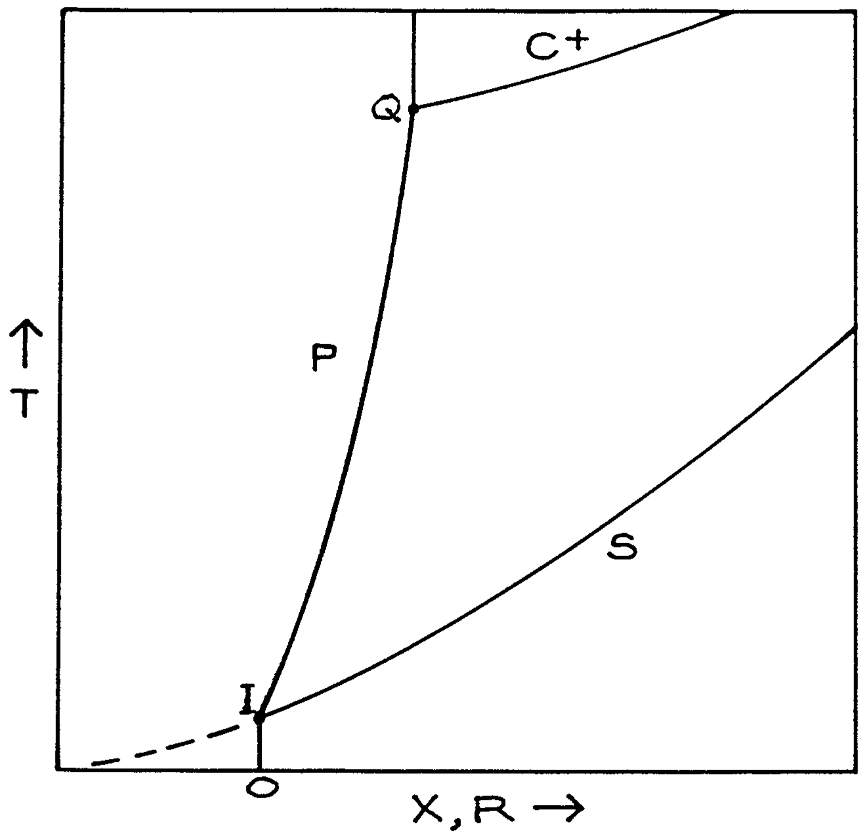

Figure 4. A one-dimensional, piston-driven flow in the XT plane (or in the case of a pseudo-one-dimensional flow, the RT plane). A piston is at rest between times 0 and I, then abruptly started to generate a shock flow. S is the shock front trajectory and P is the piston path. The piston is brought abruptly to rest again at Q and a disturbance is created. The disturbance is a rarefaction wave if the piston is moving forward when brought to rest as shown, the head of which follows a positive characteristic C+. The disturbance would be a second shock if the piston were moving backward. The region of useful, computed flow data is bounded by P, S and the disturbance trajectory.

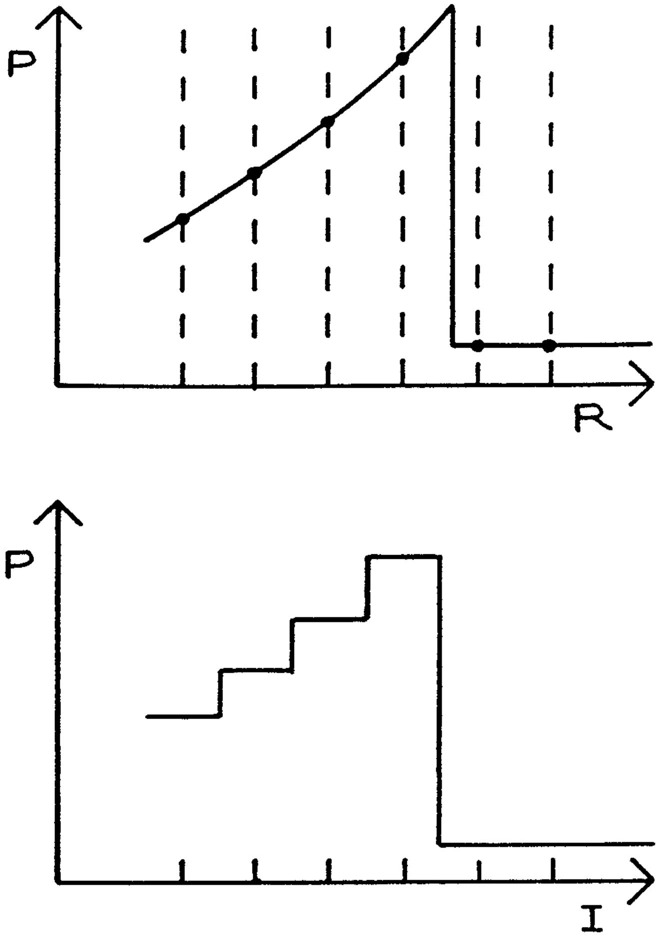

Figure 5. The representation of blast wave profiles. The actual profile, of pressure for example, is sampled at grid points shown as dashed lines (top). The profile obtained is then viewed as a series of constant states, one at each grid point, labelled I (bottom). The grid shown is excessively coarse. Hundreds or thousands of points may be used to produce a relatively smooth profile.

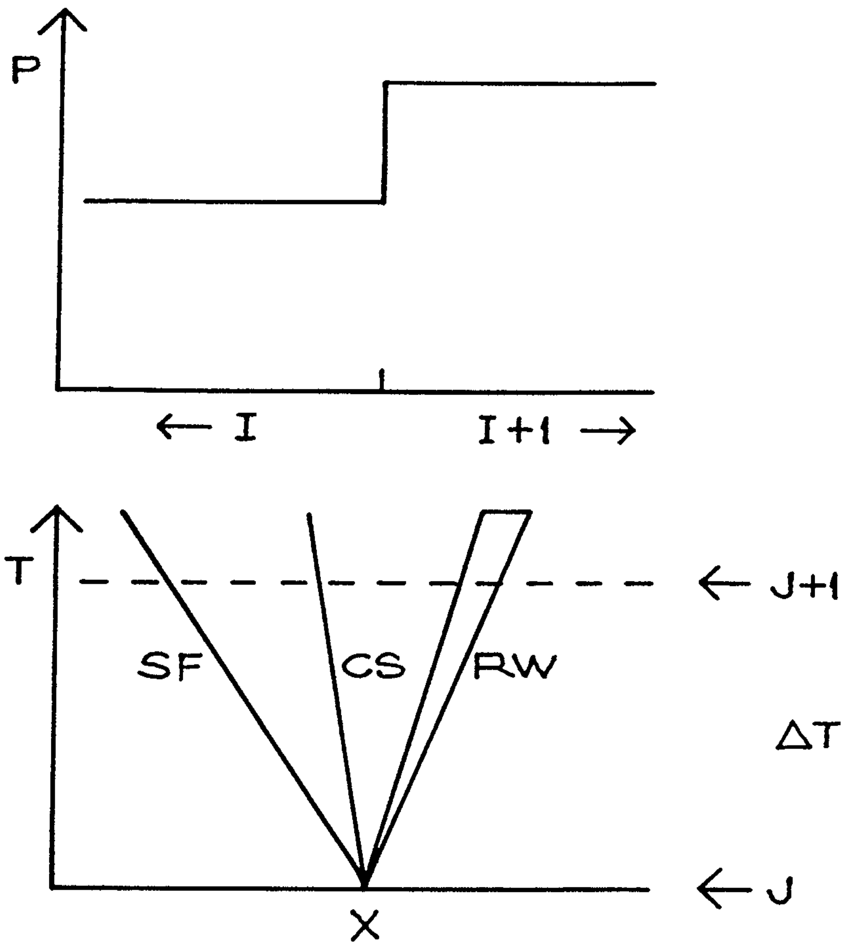

Figure 6. Two constant states in adjoining cells, I and I+1, are shown in the upper diagram at time T, corresponding to step J in a calculation. The density, pressure and flow velocity are constant in each state. The two states interact to generate a shock front SF and a rarefaction wave RW, moving in opposite directions, separated by a contact surface CS. Each moves with a constant velocity in the XT plane, shown in the lower diagram. If the initial flow velocities are zero, we have the standard shock tube problem. Generally they are not and we have a so-called Riemann problem. The time ΔT between steps J and J+1 is chosen small enough to prevent interactions with flows from neighbouring cells.

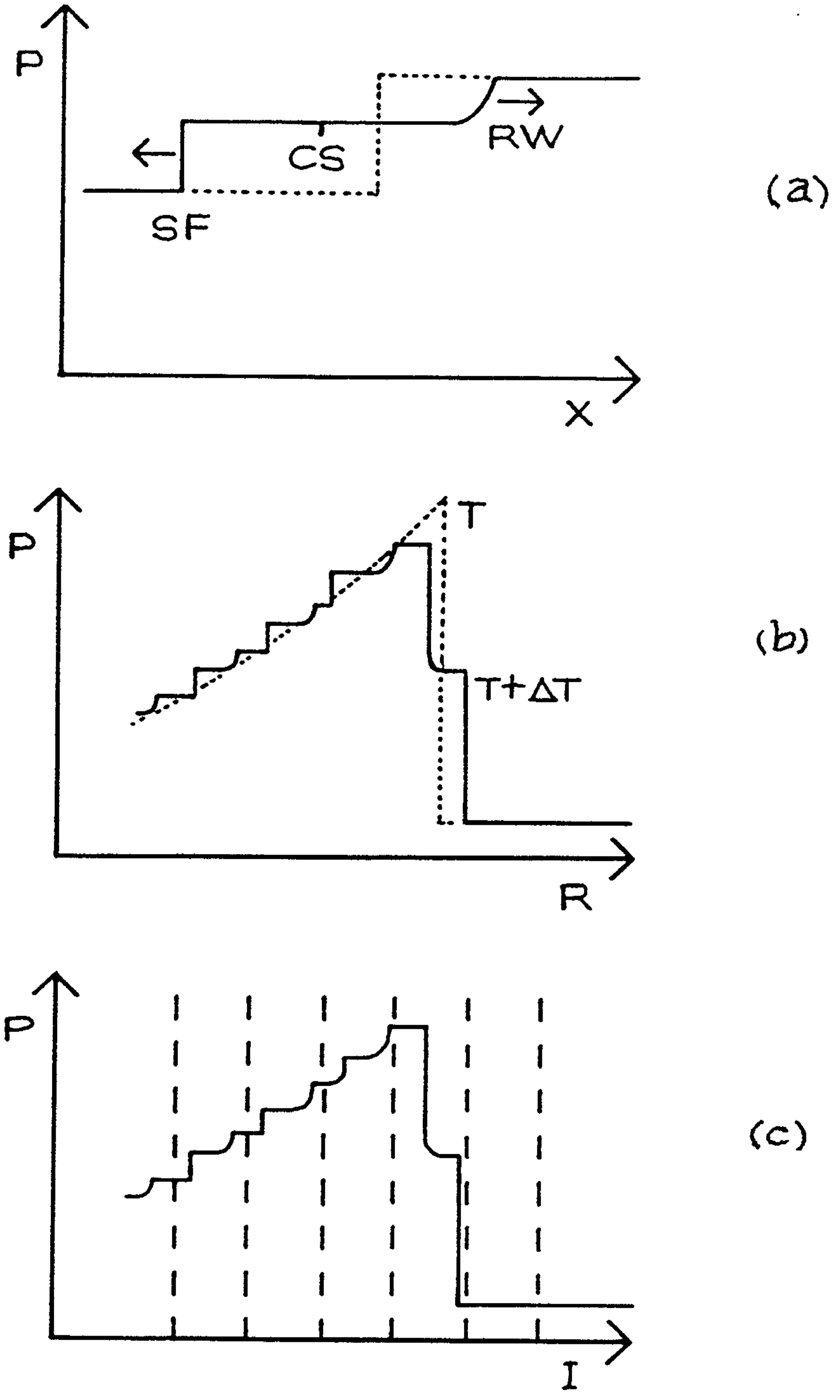

Figure 7. (a) Pressure profile obtained in the solution of a single Riemann problem. SF, RW and CS are, respectively, the shock front, the rarefaction wave and the contact surface. (b) The blast wave at time T (dotted line) is at time T+ΔT represented by a profile (solid line) built up from individual Riemann solutions, after a geometrical 'correction' to adjust from a linear position coordinate X to a radial coordinate R. (c) The new profile is again sampled at the grid points.

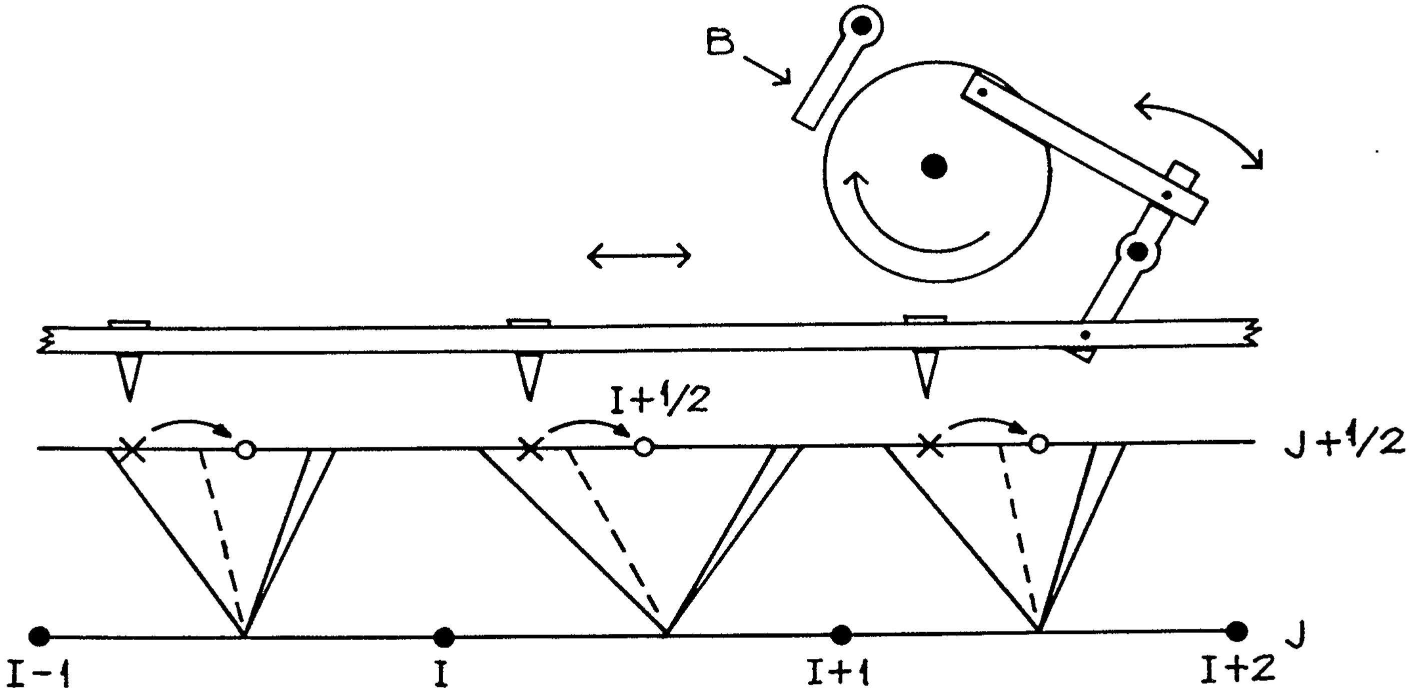

Figure 8. The random choice method. The contraption above samples the evolved flow field as follows: a rotating wheel drives a reciprocating rake whose teeth each sweep out one computational cell. A brake B is applied at a random time and the Riemann solutions are sampled (at points labelled X). Note that the sampling is done at the same position in each cell at any one time. The sampled flow data are temporarily assigned positions corresponding to cell centres (open circles), and a time equal to T+ΔT/2. A second 'half-step evolution' is then allowed and the result is again randomly sampled. Assignment of sampled data at the end of the full time step is made back to the original grid points (filled circles). Locating the individual discontinuities randomly within the cells before solving the Riemann problems and sampling only at the cell centre points is mathematically equivalent to what is described above. The device above is simply harmonic and could not be used to obtain a set of random positions, even if sampled at random times, but is described here for illustrative purposes.

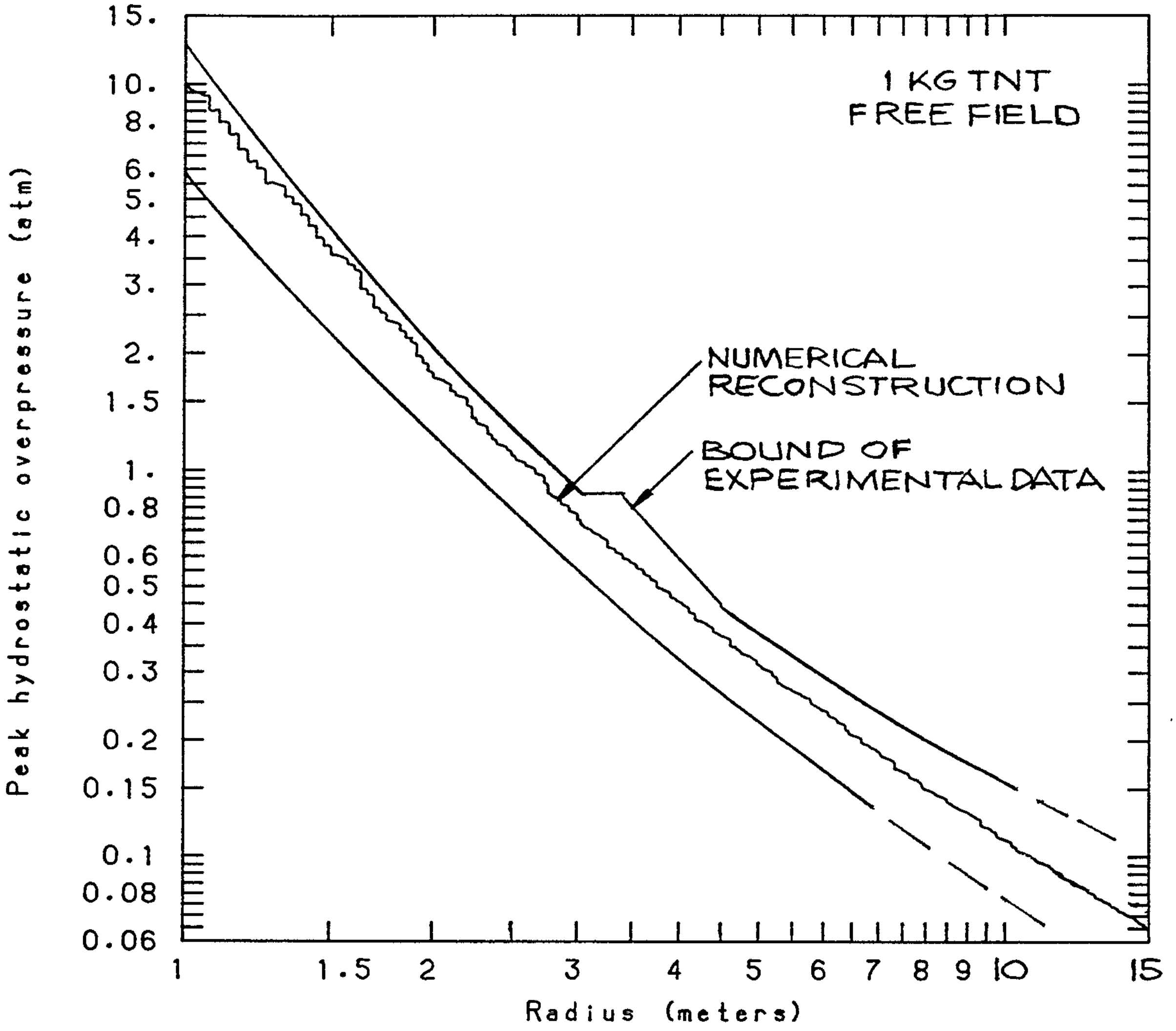

Figure 9. Peak pressure profile reconstructed for 1 kg TNT (free field) compared with the available experimental data shown in figure 1a. Noise on the reconstructed profile is a consequence of using the random choice method to compute the flow (the results are rigorously correct 'on the average'). The piston path used to compute the flow was adjusted until the computed profile best-matched the experimental data. The computed profile lies between those of Baker (1973) and Brode (1957, 1959) shown in figure 1a.

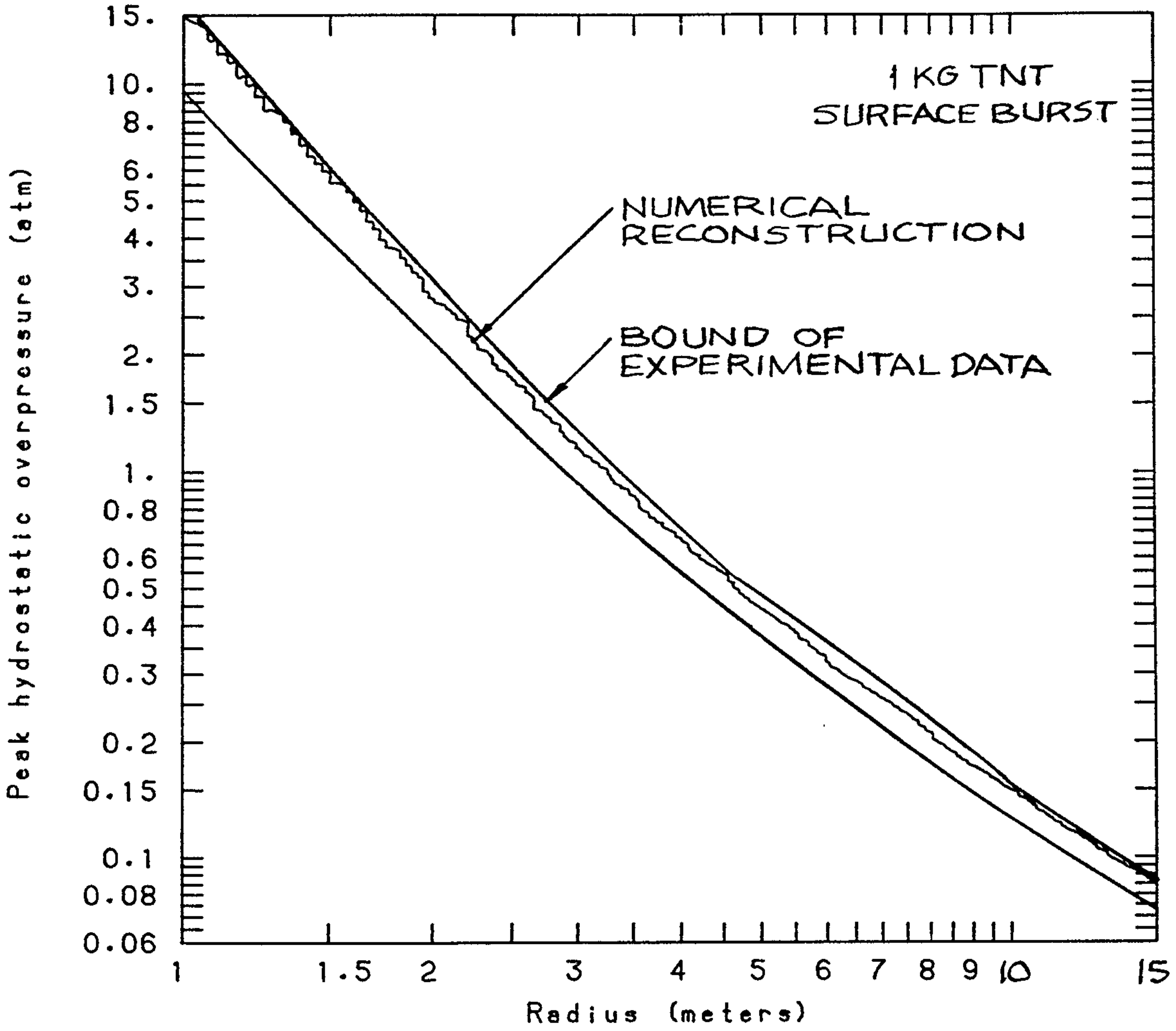

Figure 10. Peak pressure profile reconstructed for 1 kg TNT (surface burst) compared with the experimental data shown in figure 2. The piston path used to compute the flow was adjusted until the computed profile best-matched the experimental data. The computed profile lies between the results of Dewey (1964), Kingery & Pannill (1964) and Kingery (1968) shown in figure 2. The data of Reisler, Giglio-Tos & Kellner (1966) were given less weight in the comparison because they were for half-buried spherical, rather than hemispherical, charges.

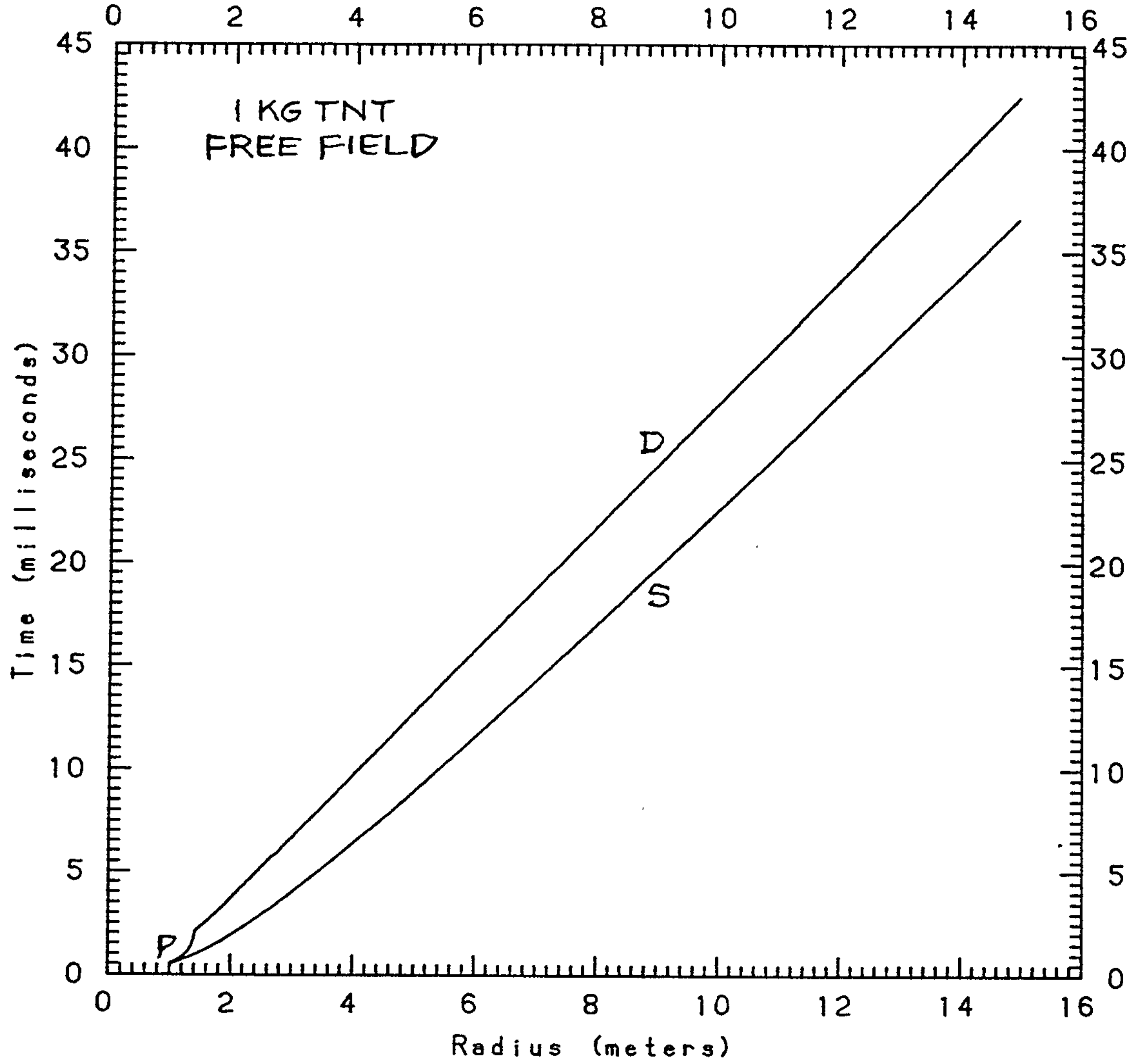

Figure 11. Region in the radius-time plane over which the free field flow was reconstructed. P, S and D are, respectively, the trajectories of the piston, the shock front and the disturbance caused by bringing the piston abruptly to rest. The disturbance in this case is a weak shock approximately like the secondary shock which occurs in actual explosions.

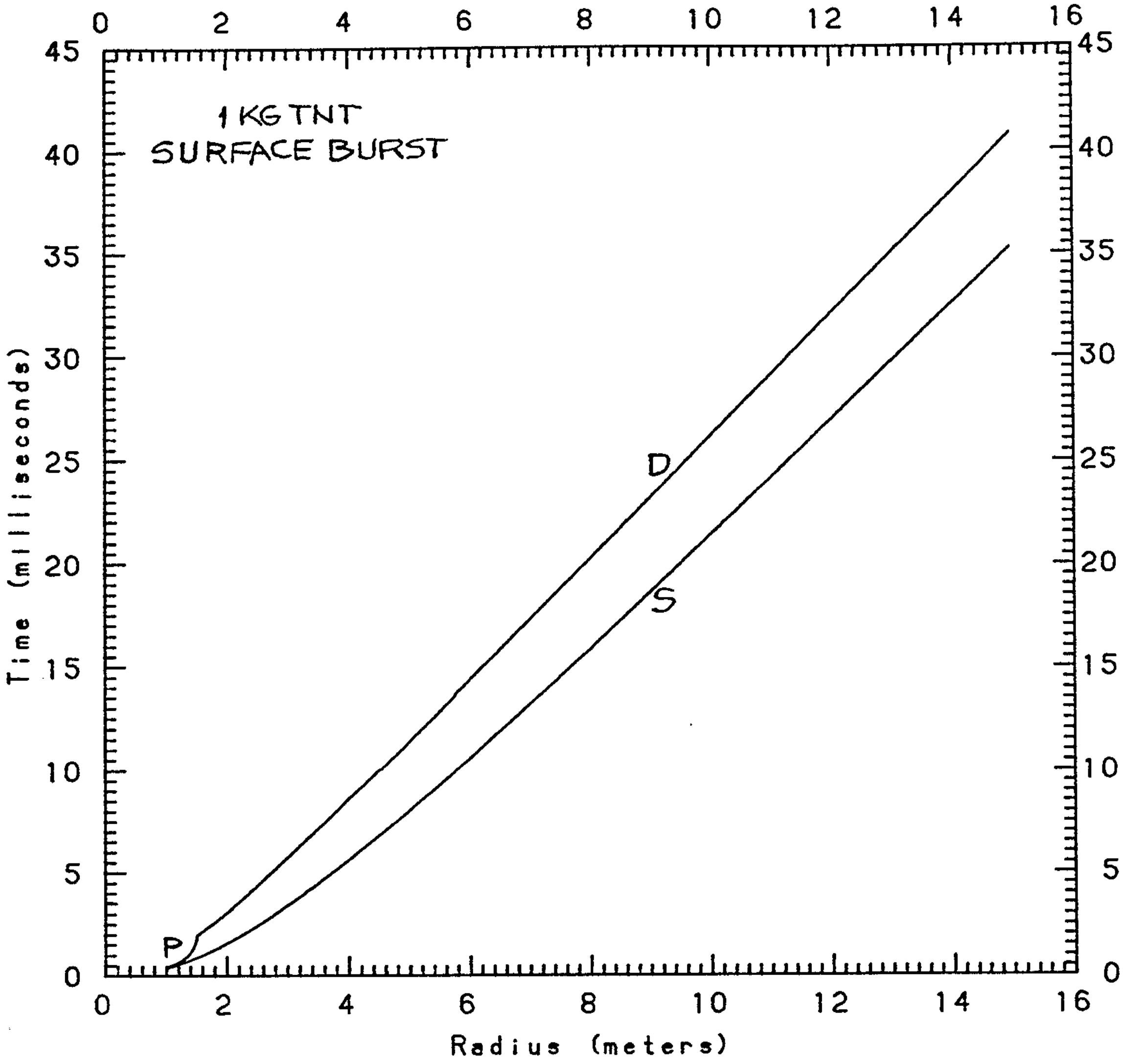

Figure 12. Region in the radius-time plane over which the surface burst flow was reconstructed. The trajectories P, S and D are described in the previous figure.

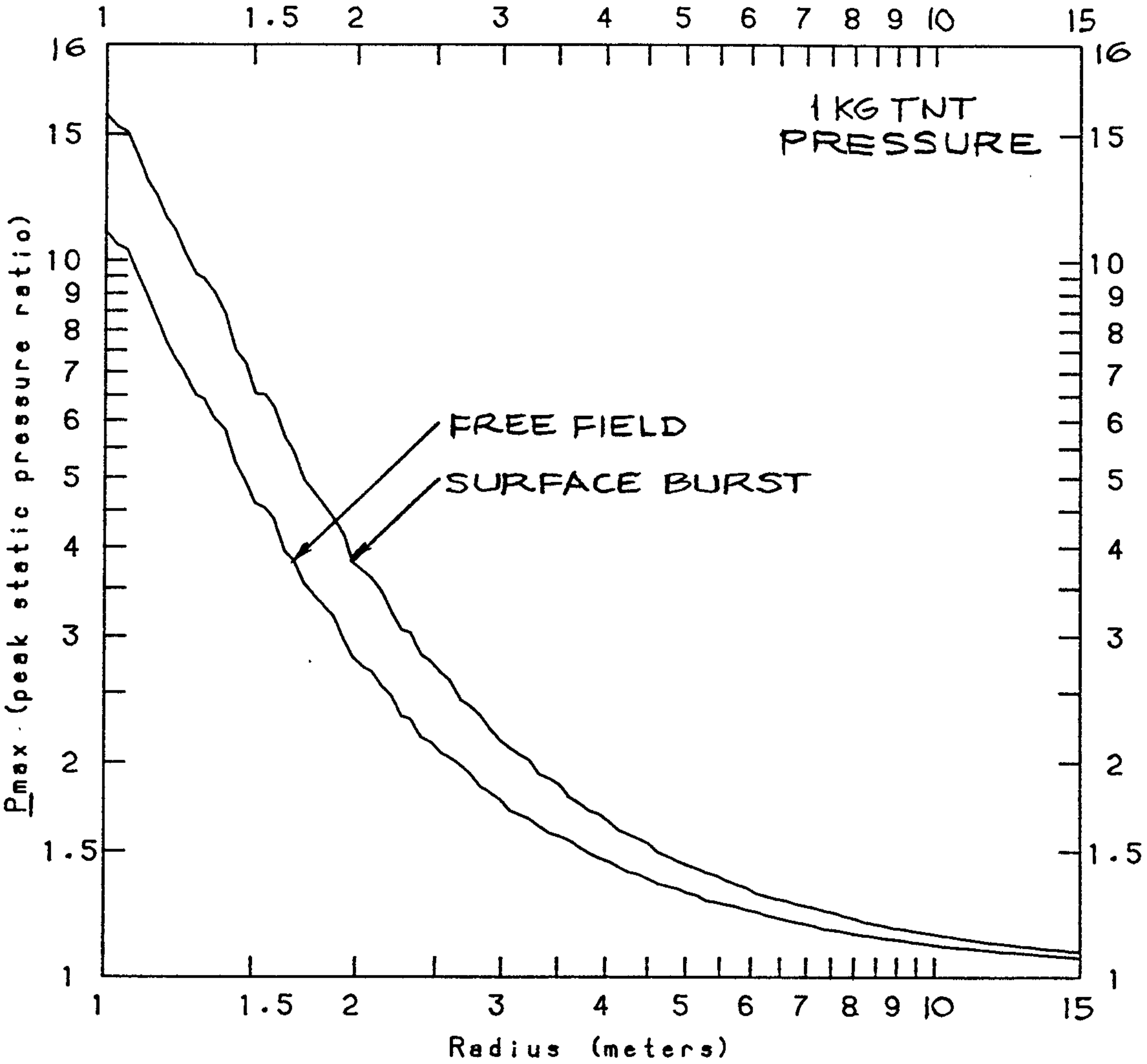

Figure 13. Profiles of peak static pressure in the free field and surface burst reconstructed flows. The noise which results from using the random choice method has been smoothed somewhat by selecting data at 100 'gauge' positions (cf. the 1500 grid points in figures 9 and 10). The data in this figure and all subsequent figures were obtained using the UVic/AIRBLAST data display package which is provided with this report.

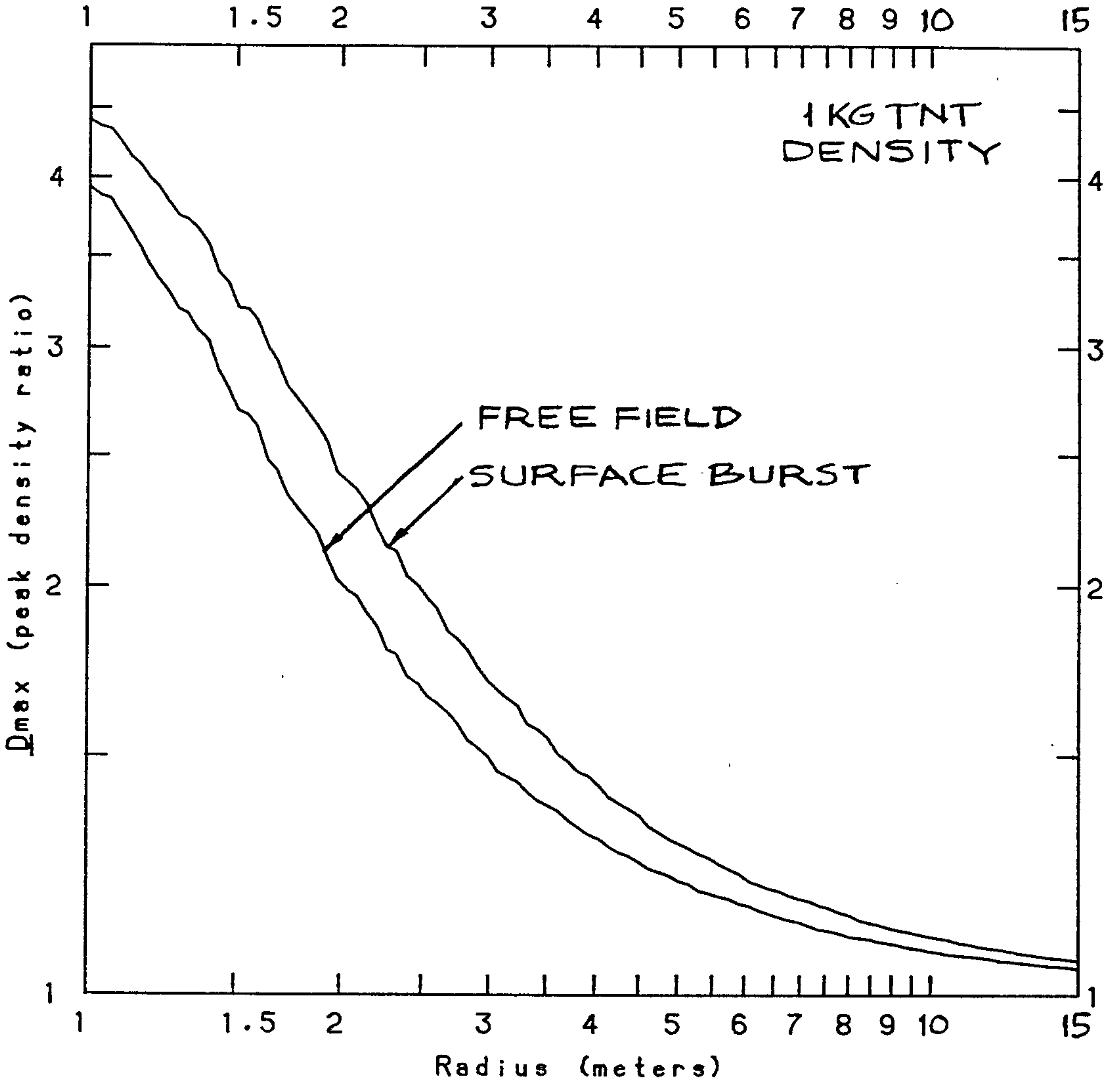

Figure 14. Peak density profiles, reconstructed.

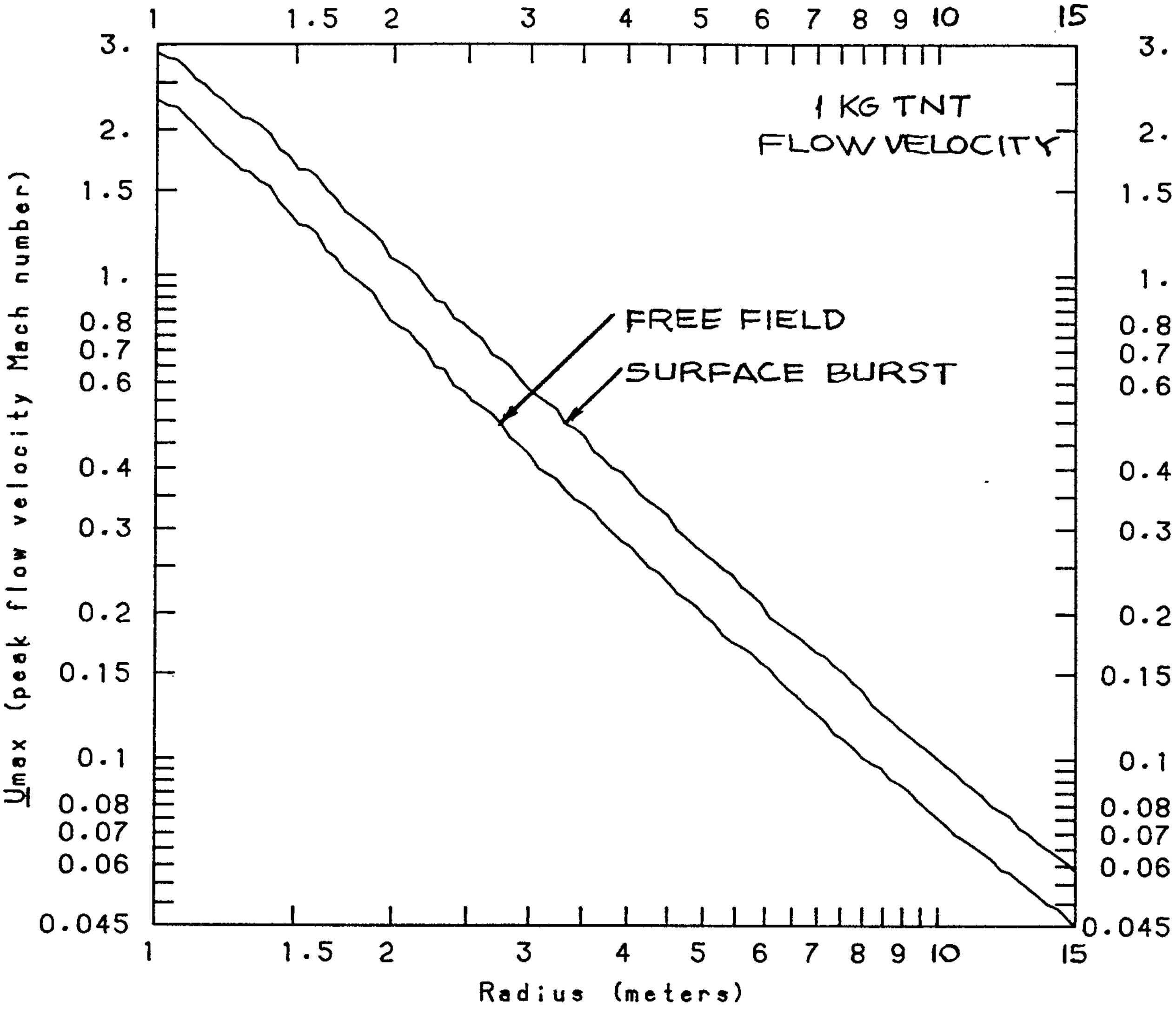

Figure 15. Peak flow velocity profiles, reconstructed.

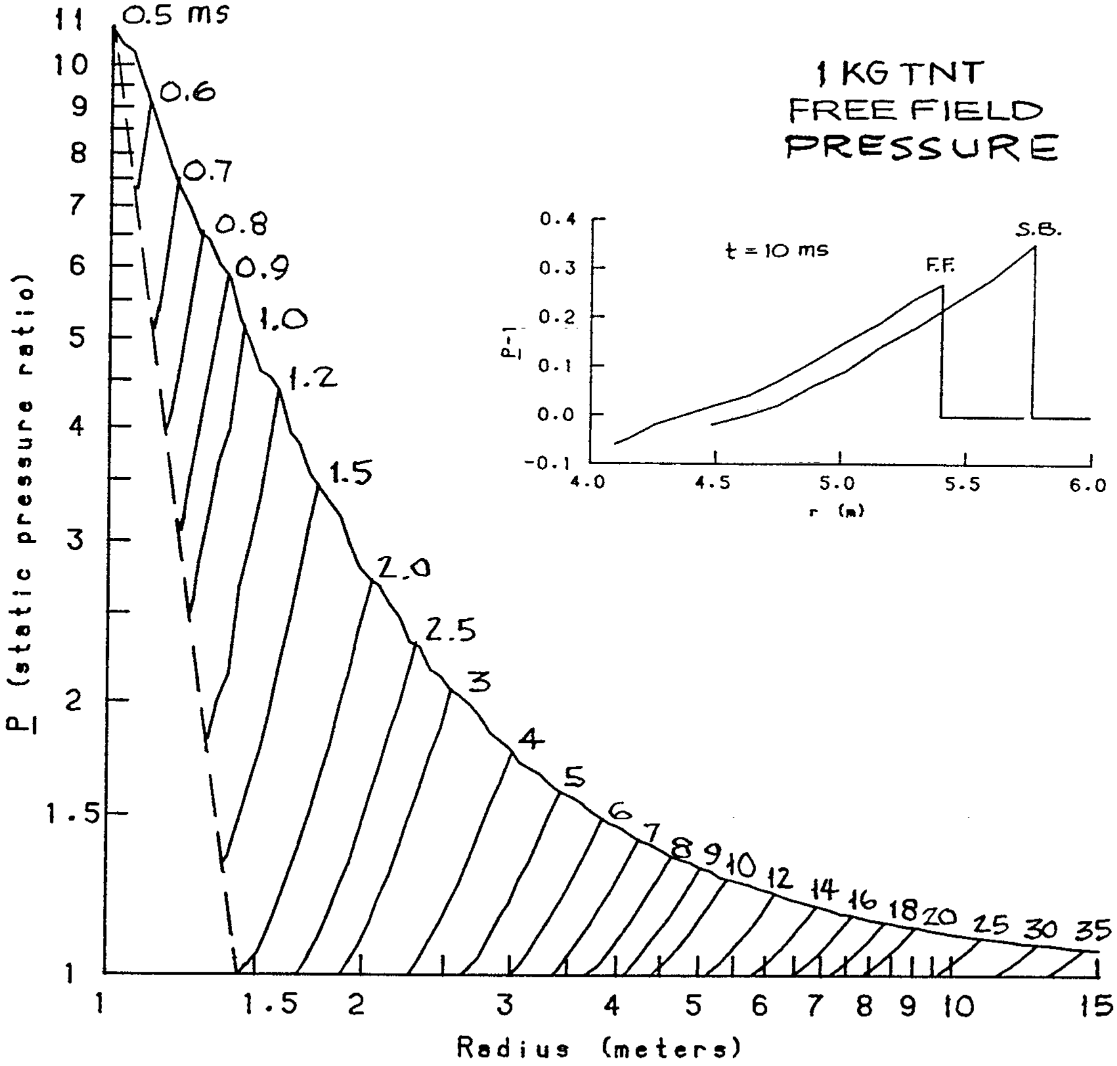

Figure 16. Blast wave pressure profiles for free field TNT. Static pressure is plotted versus position at 25 fixed times, together with the peak pressure data which were plotted in figure 13. The piston, which was started at position 1 m and time 0.5 ms, 'moves' through the region below the broken line. The inset shows the overpressure wave profile at 10 ms compared with the surface burst wave profile at the same time.

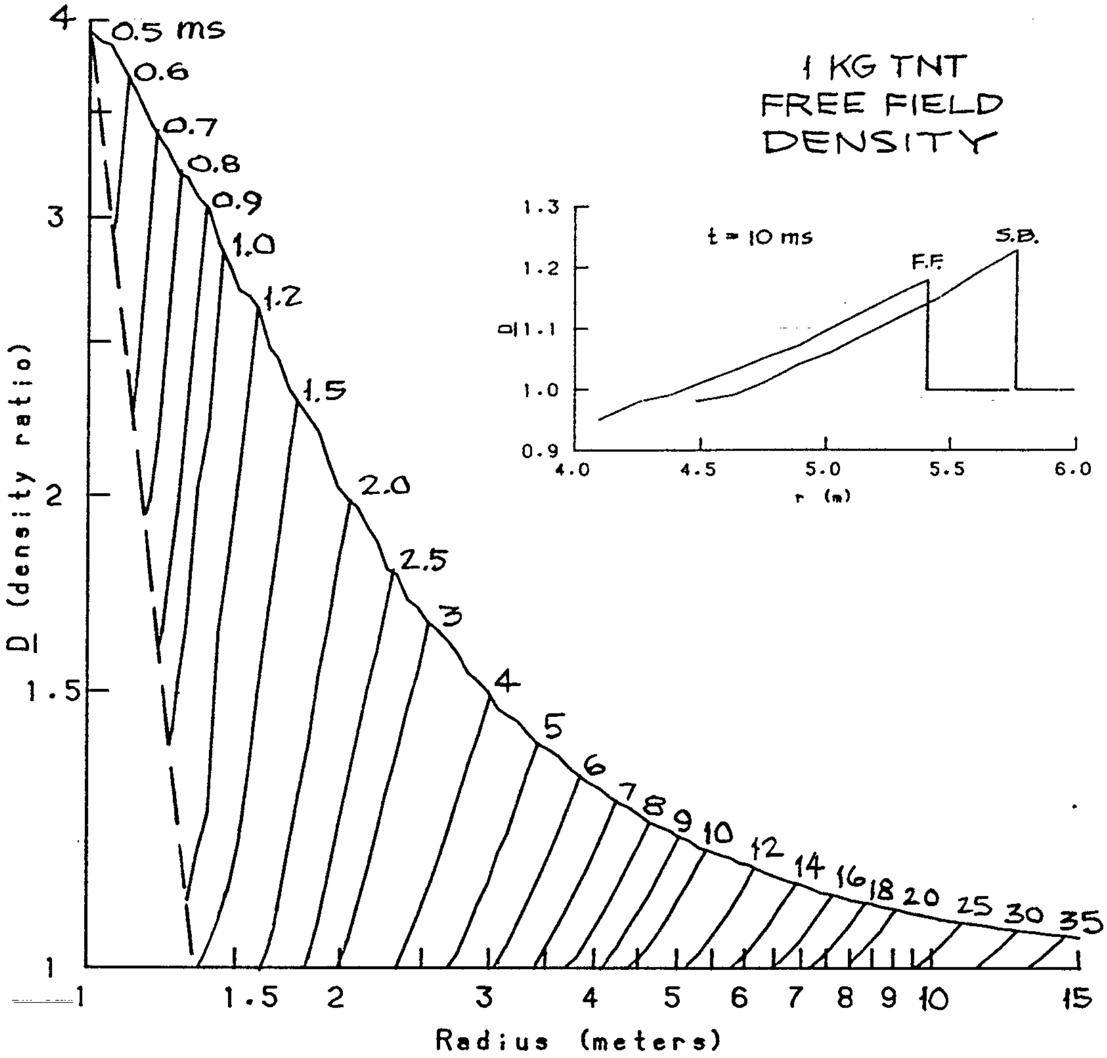

Figure 17. Density wave profiles for free field TNT.

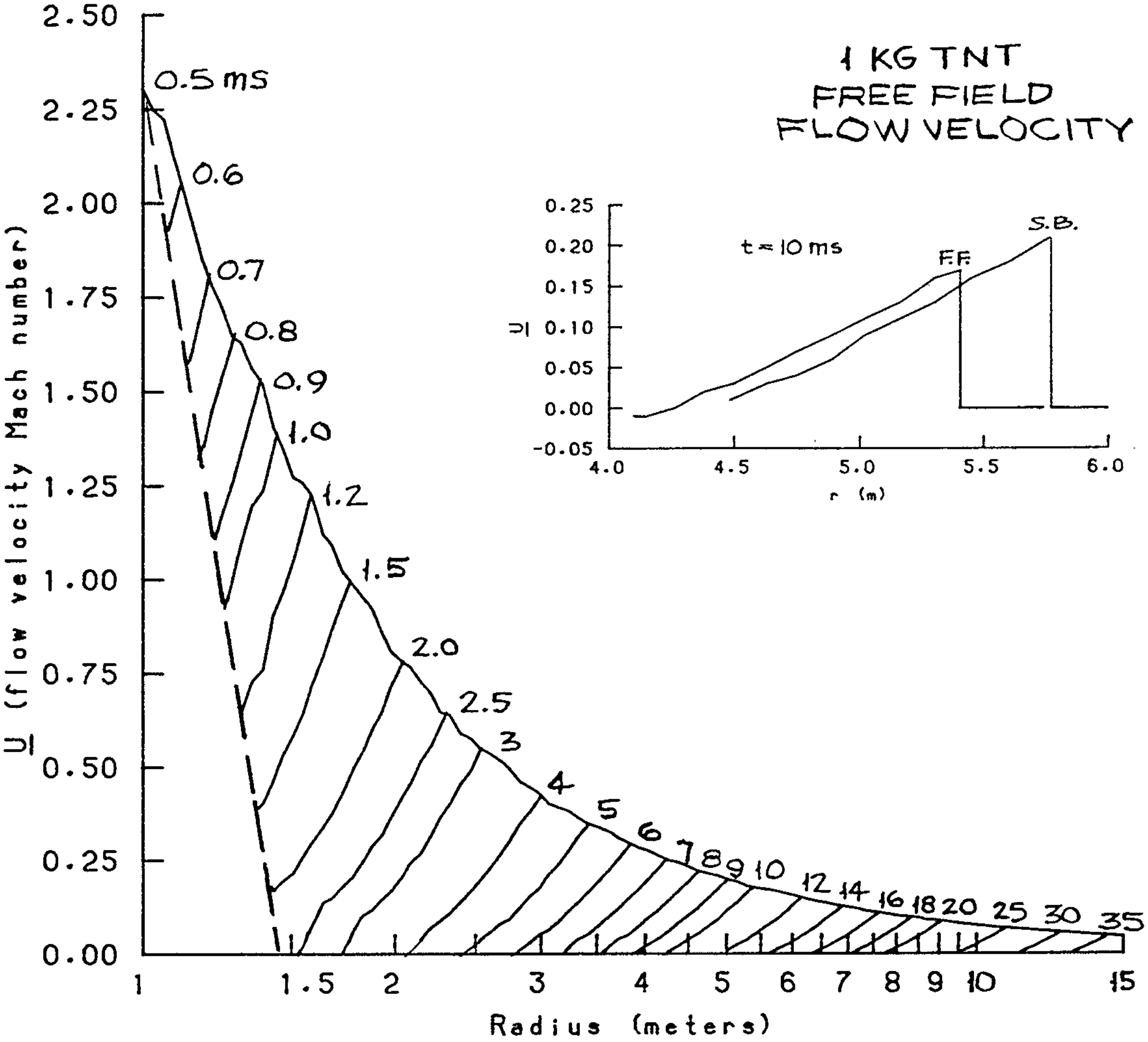

Figure 18. Flow velocity wave profiles for free field TNT. Using the data shown in this and the previous two figures, wave profiles for other physical parameters, temperature for example, can be computed.

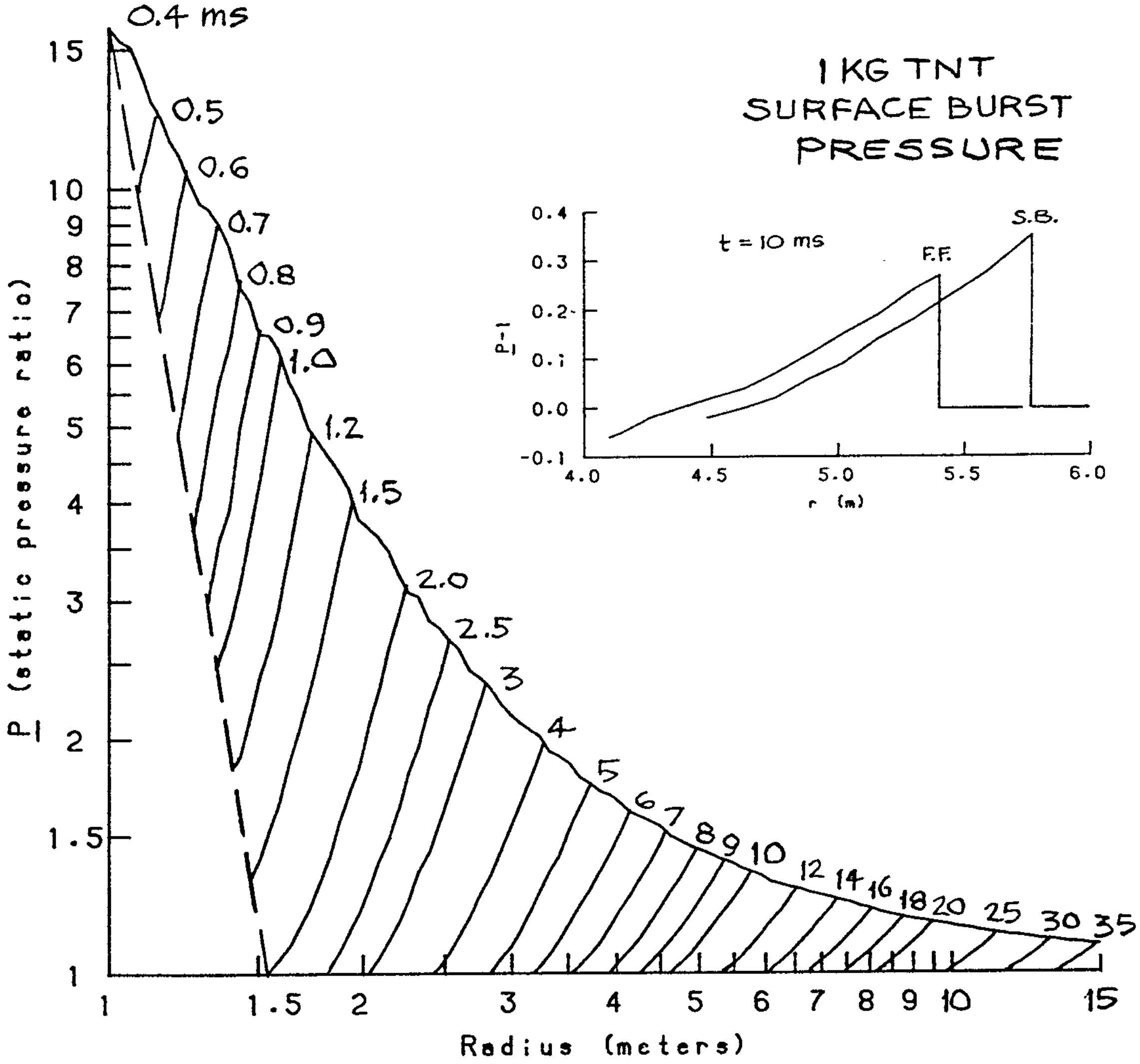

Figure 19. Pressure wave profiles for surface burst TNT.

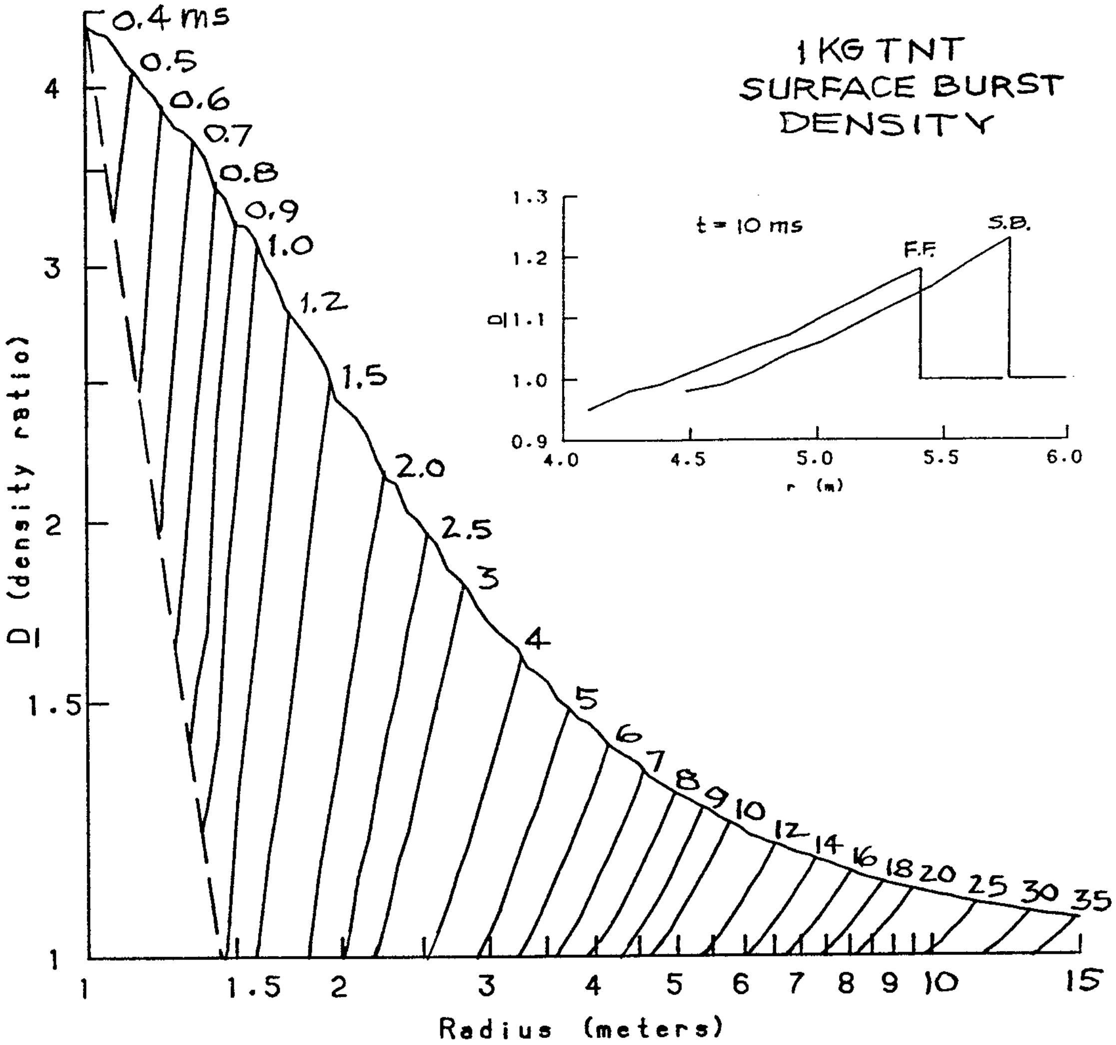

Figure 20. Density wave profiles for surface burst TNT.

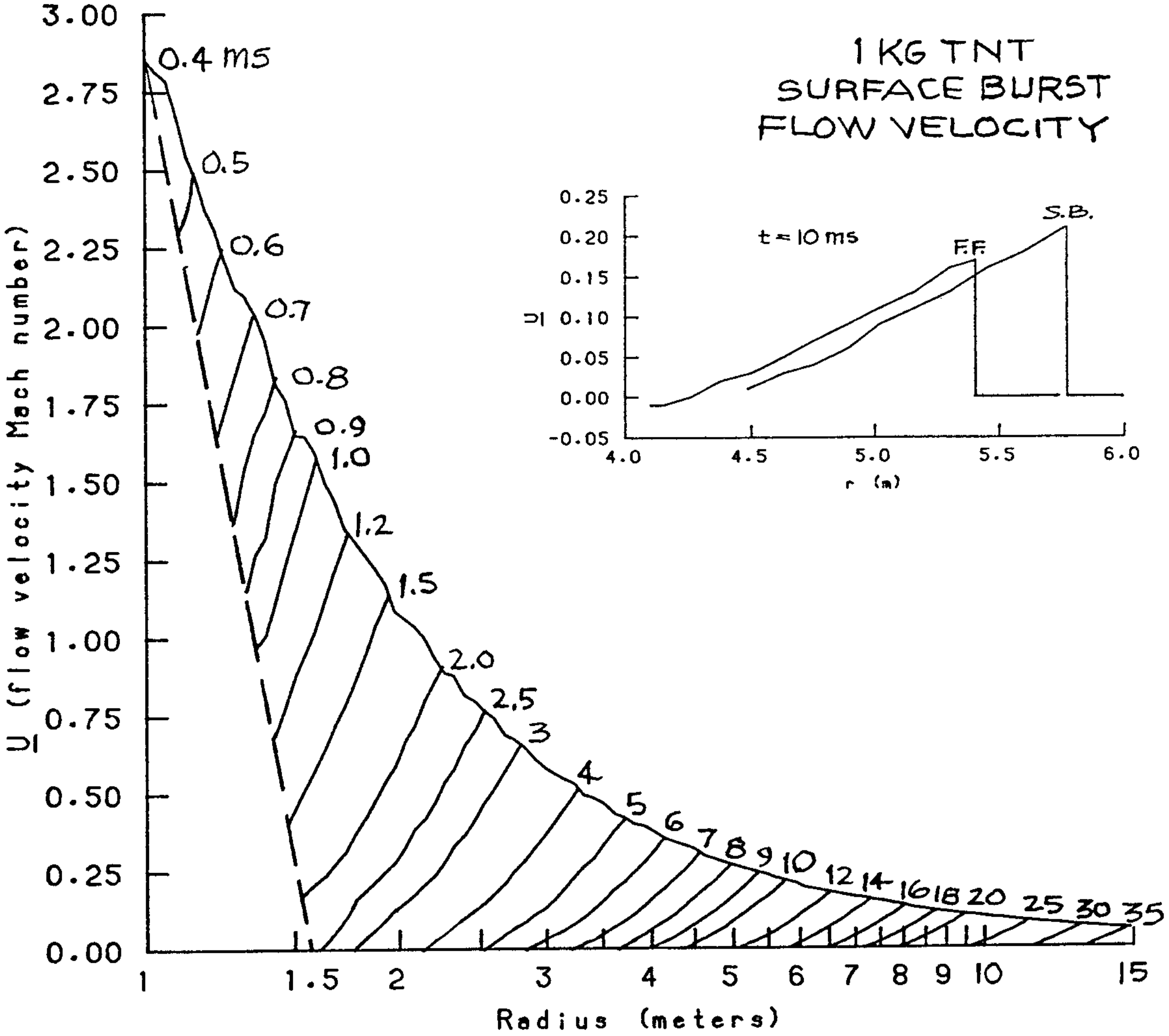

Figure 21. Flow velocity wave profiles for surface burst TNT.

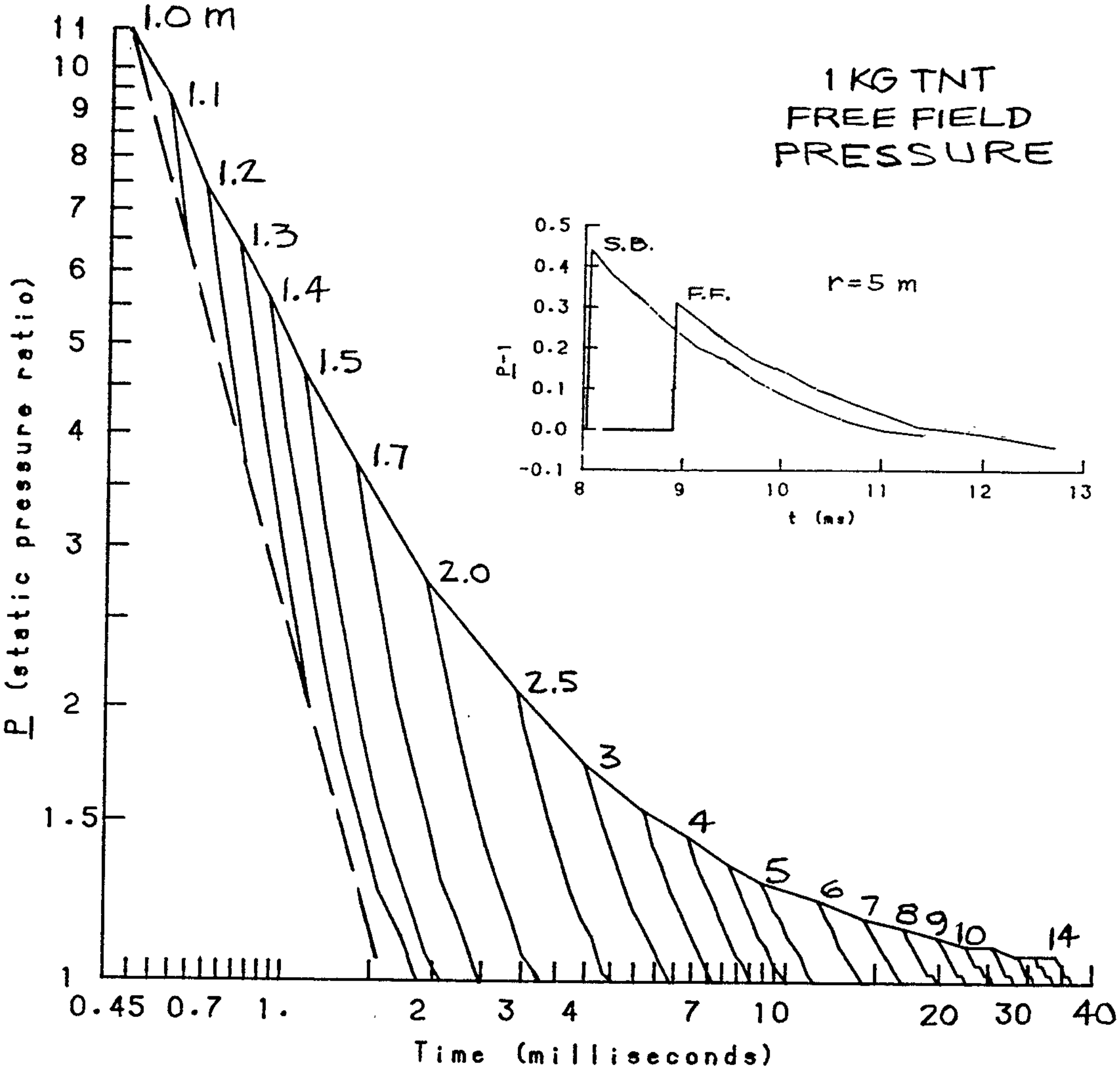

Figure 22. Pressure time histories for free field TNT. Static pressure is plotted versus time at 22 fixed positions, together with the peak data as a function of time. The piston, which was started at position 1 m and time 0.5 ms, 'moves' through the region below the broken line. The inset shows an overpressure time history at position 5 m, compared with the surface burst history at the same position.

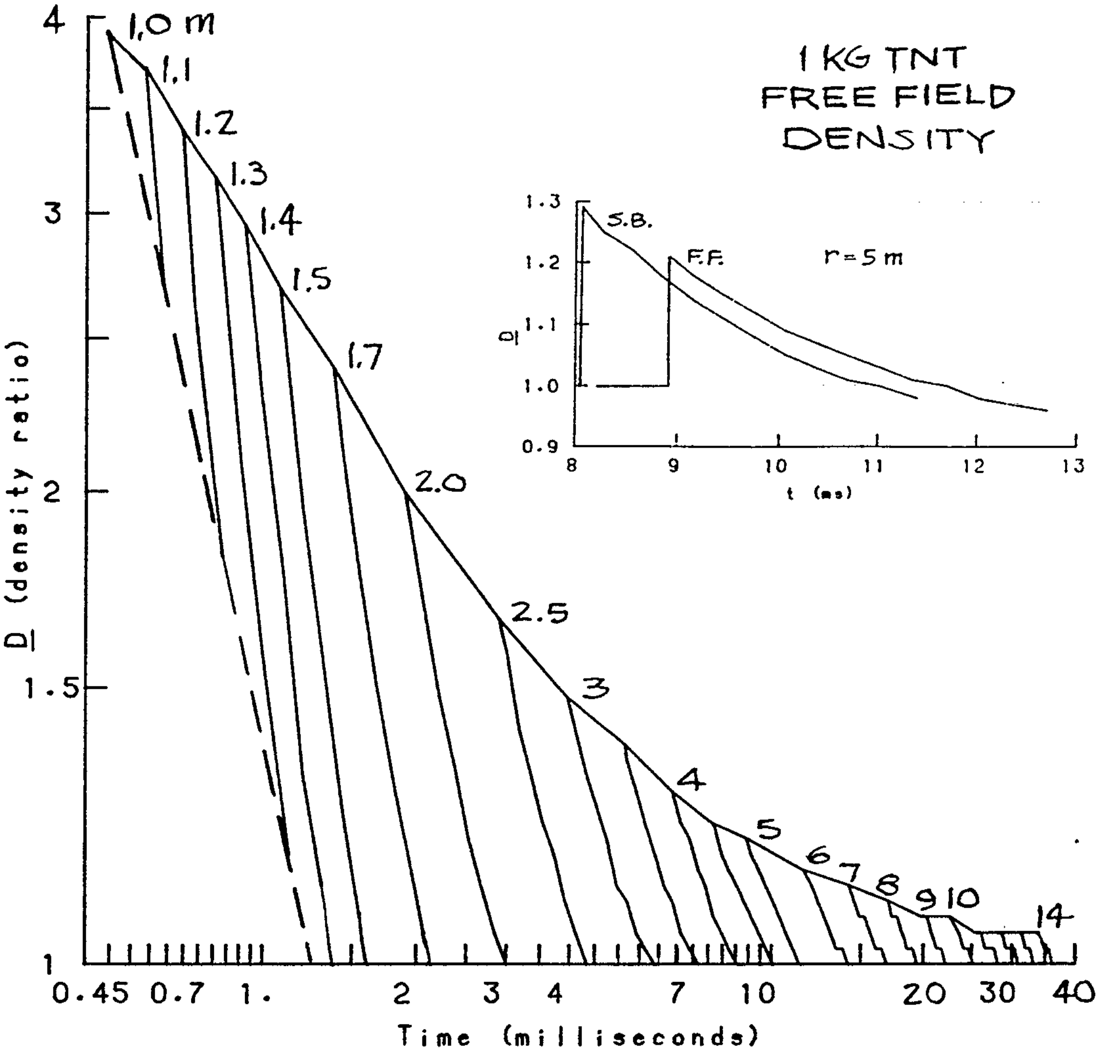

Figure 23. Density time histories for free field TNT.

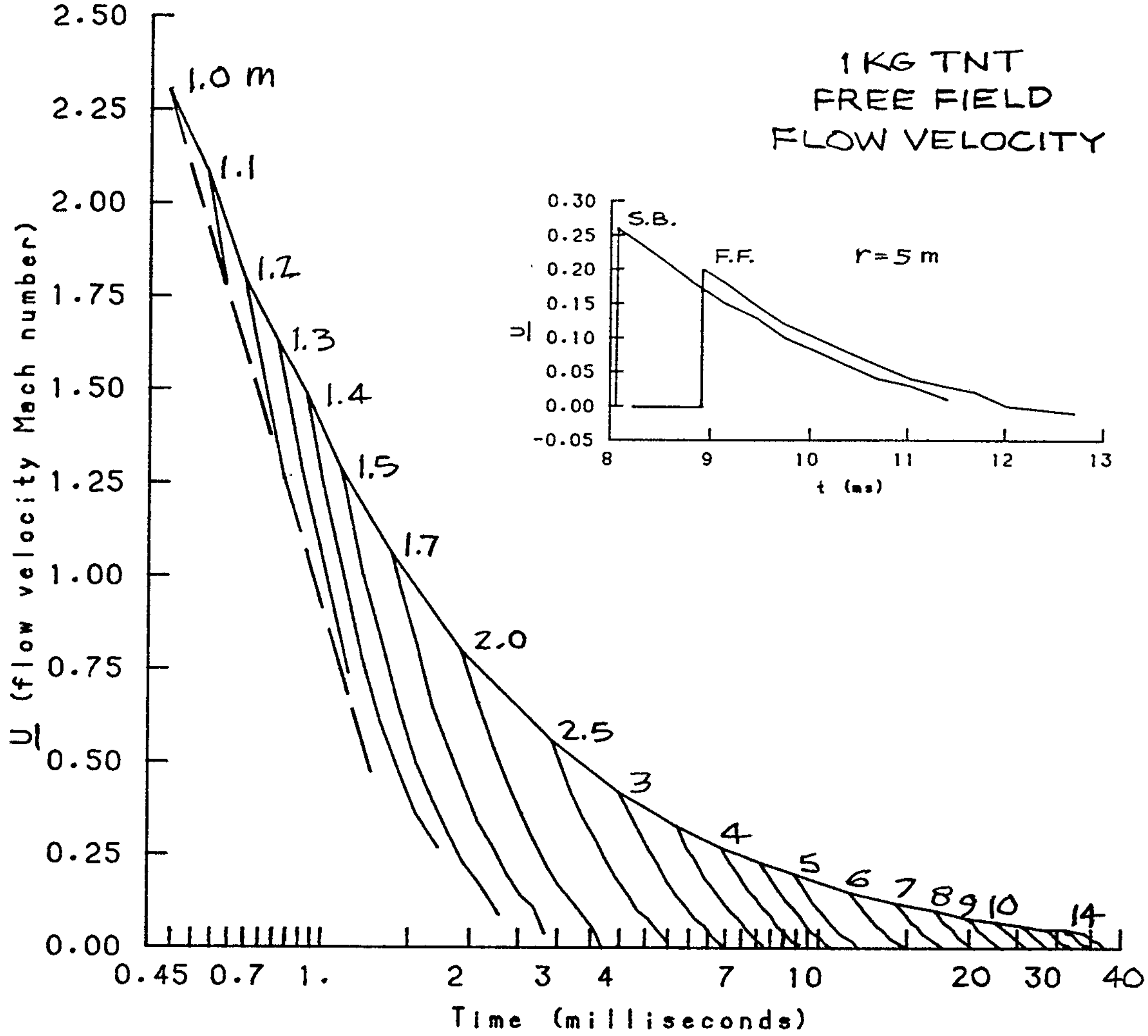

Figure 24. Flow velocity time histories for free field TNT. Using the data shown in this and the previous two figures, time histories for other physical parameters, temperature for example, can be computed.

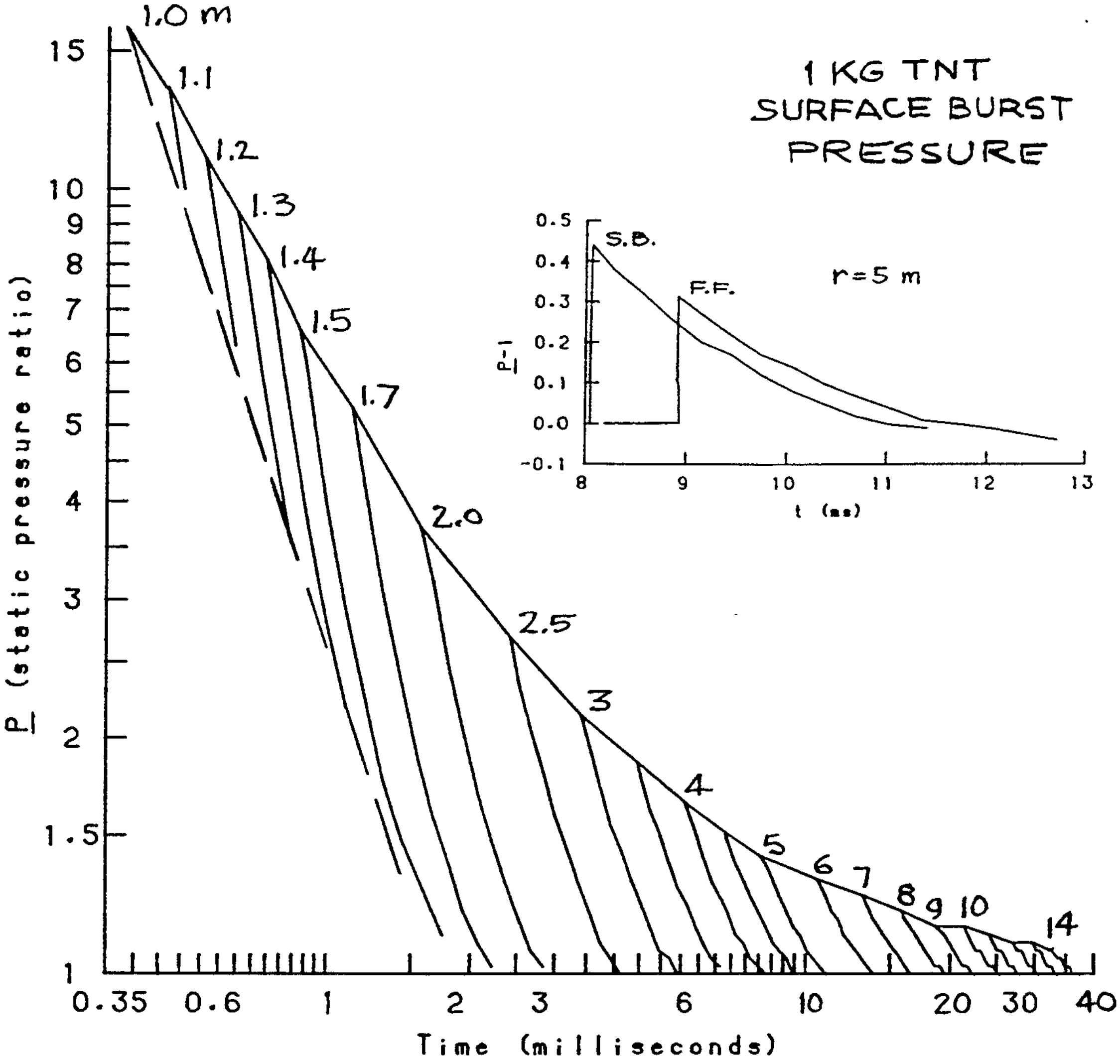

Figure 25. Pressure time histories for surface burst TNT.

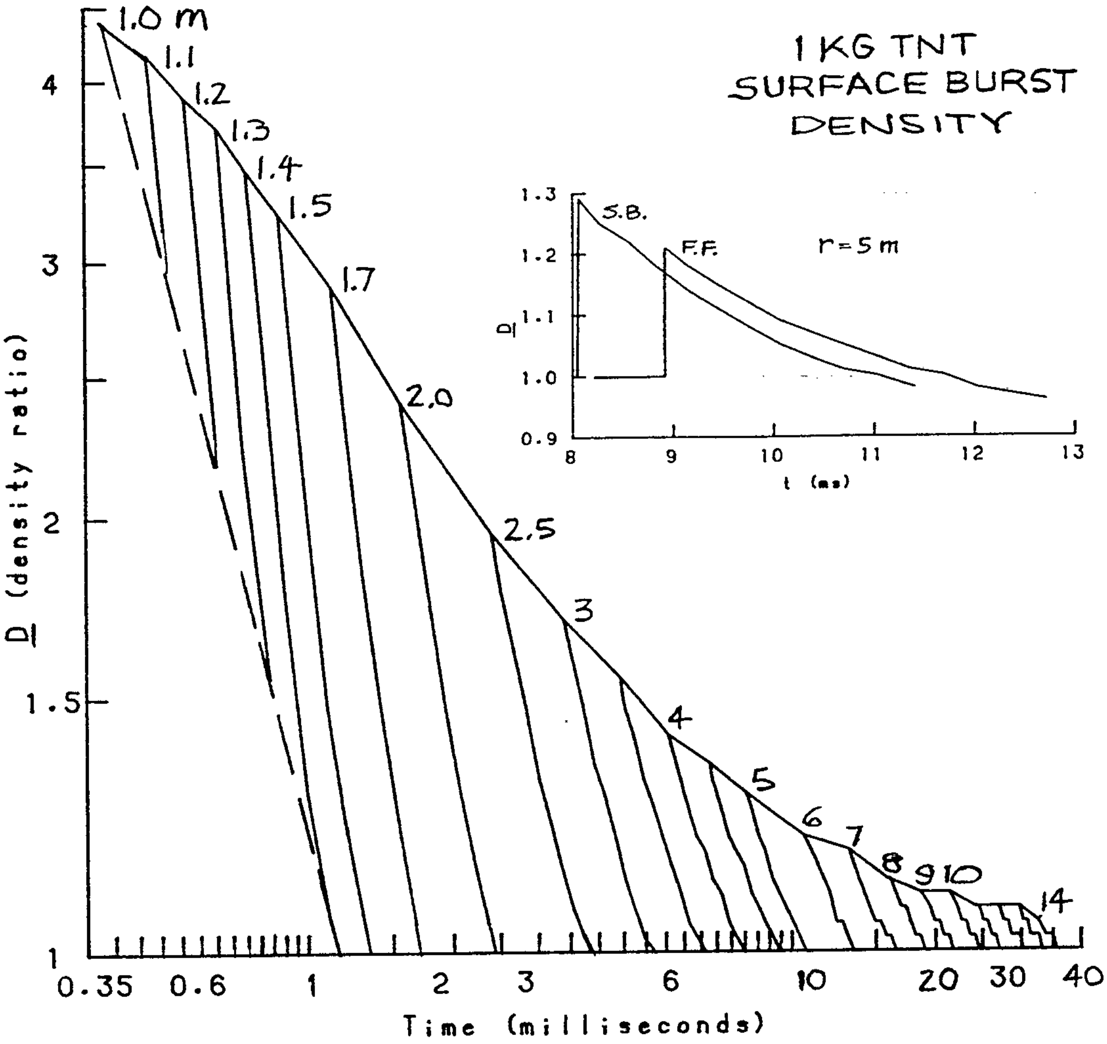

Figure 26. Density time histories for surface burst TNT.

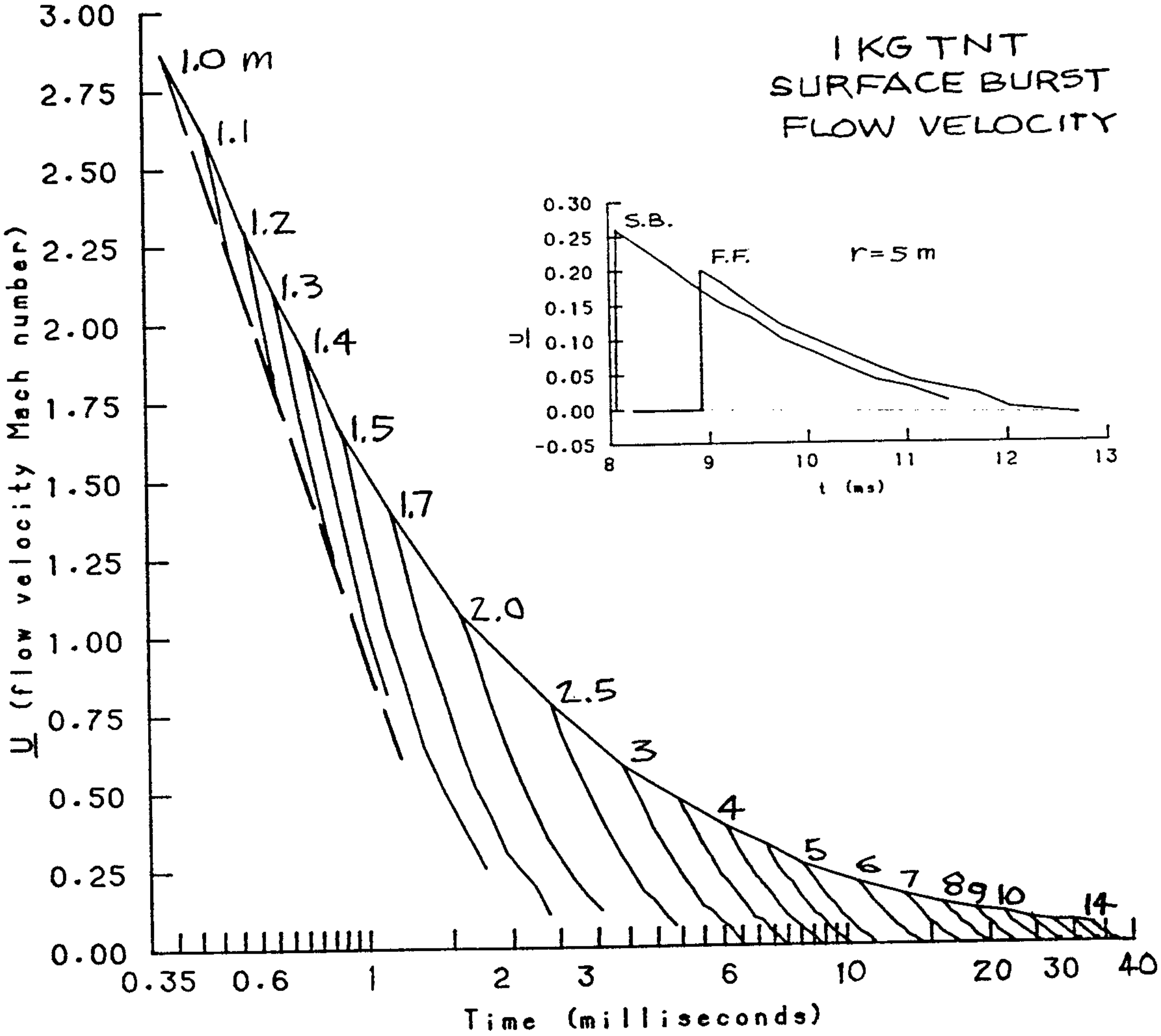

Figure 27. Flow velocity time histories for surface burst TNT.

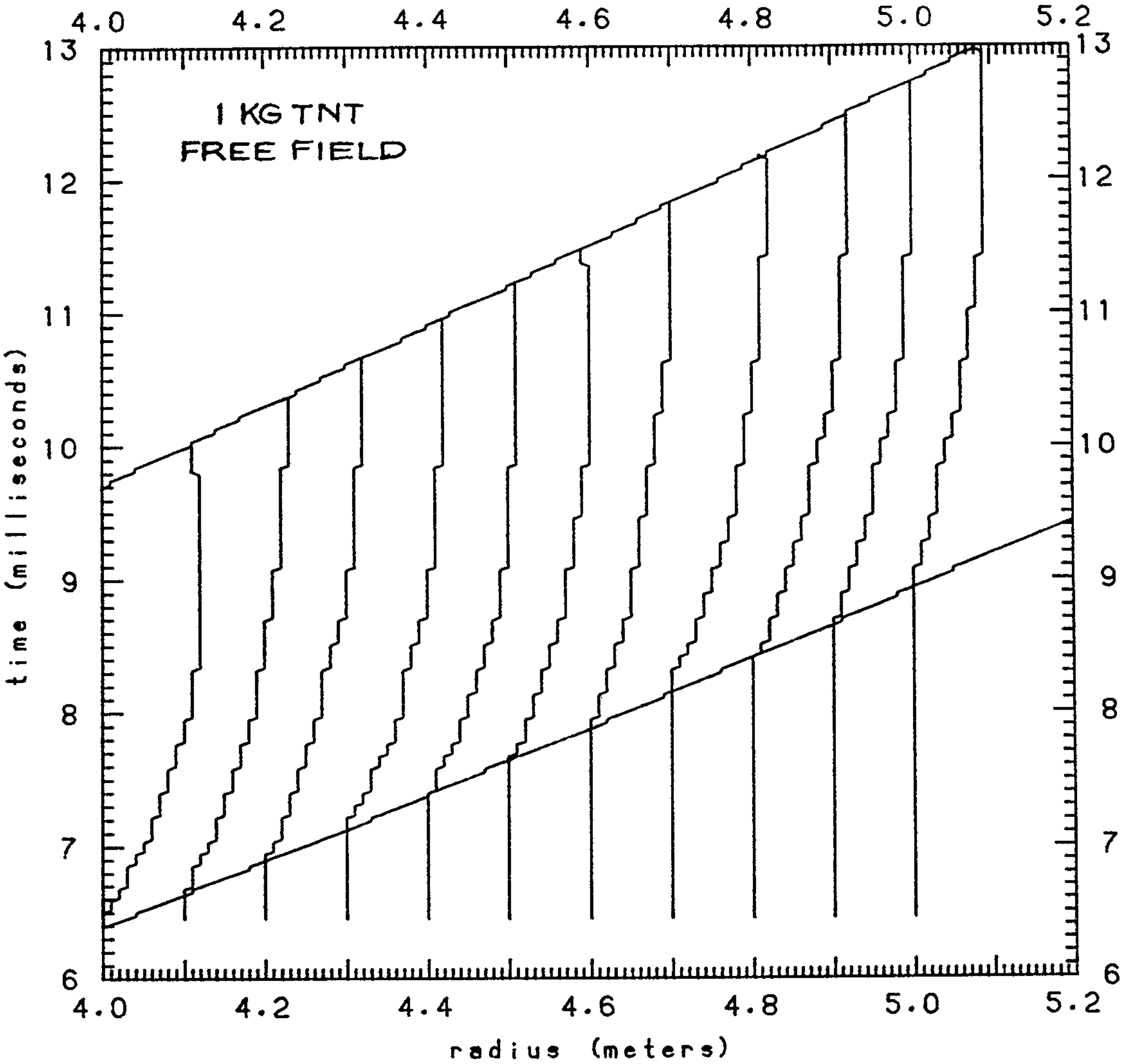

Figure 28. Some reconstructed particle trajectories in the free field TNT flow. Flow tracers are started at 0.1 m intervals between radial positions of 4 m and 5 m. S and D are segments of the shock front and piston disturbance trajectories shown in figure 11. The 'staircase' effect in the particle trajectories is a result of using a finite cell size (0.01 m) for the flow field calculation. In the shock and piston trajectories, the staircase effect corresponds to finite time steps.

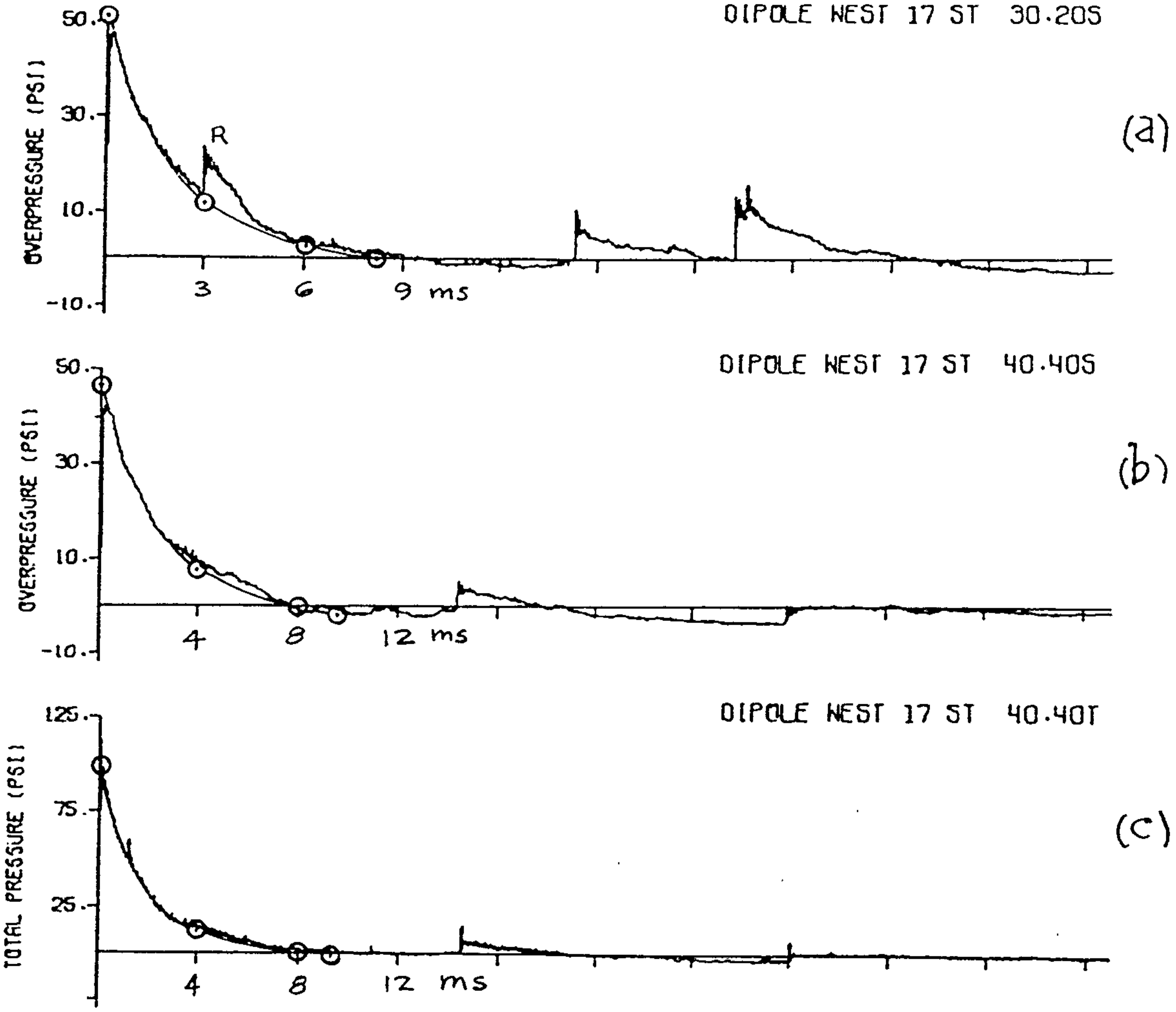

Figure 29. Comparisons of reconstructed pressure data with BRL gauge records for DIPOLE WEST shot 17. In this experiment a 1000 lb TNT charge was detonated 47 ft above the ground (1.72 m scaled to 1 kg TNT). The gauges used for the comparison were in the free field region, above the triple point trajectory. (a) Static overpressure at distance 11.9 m from the charge (1.48 m scaled). Data interpolated using UVic/AIRBLAST are shown as open circles. The reconstructed data agree with the gauge data until the reflected shock front R arrives. The reconstructed data are for free field only, and do not model reflected shocks above the ground surface. (b) Overpressure at distance 12.2 m (1.52 m scaled). Because this gauge was higher above the ground, the reflected shock is delayed. The first gauge was mounted at 30 ft ground radius and height 20 ft, while the second was mounted at 40 ft ground radius and height 40 ft. (c) Total head-on overpressure measured at the second gauge position.

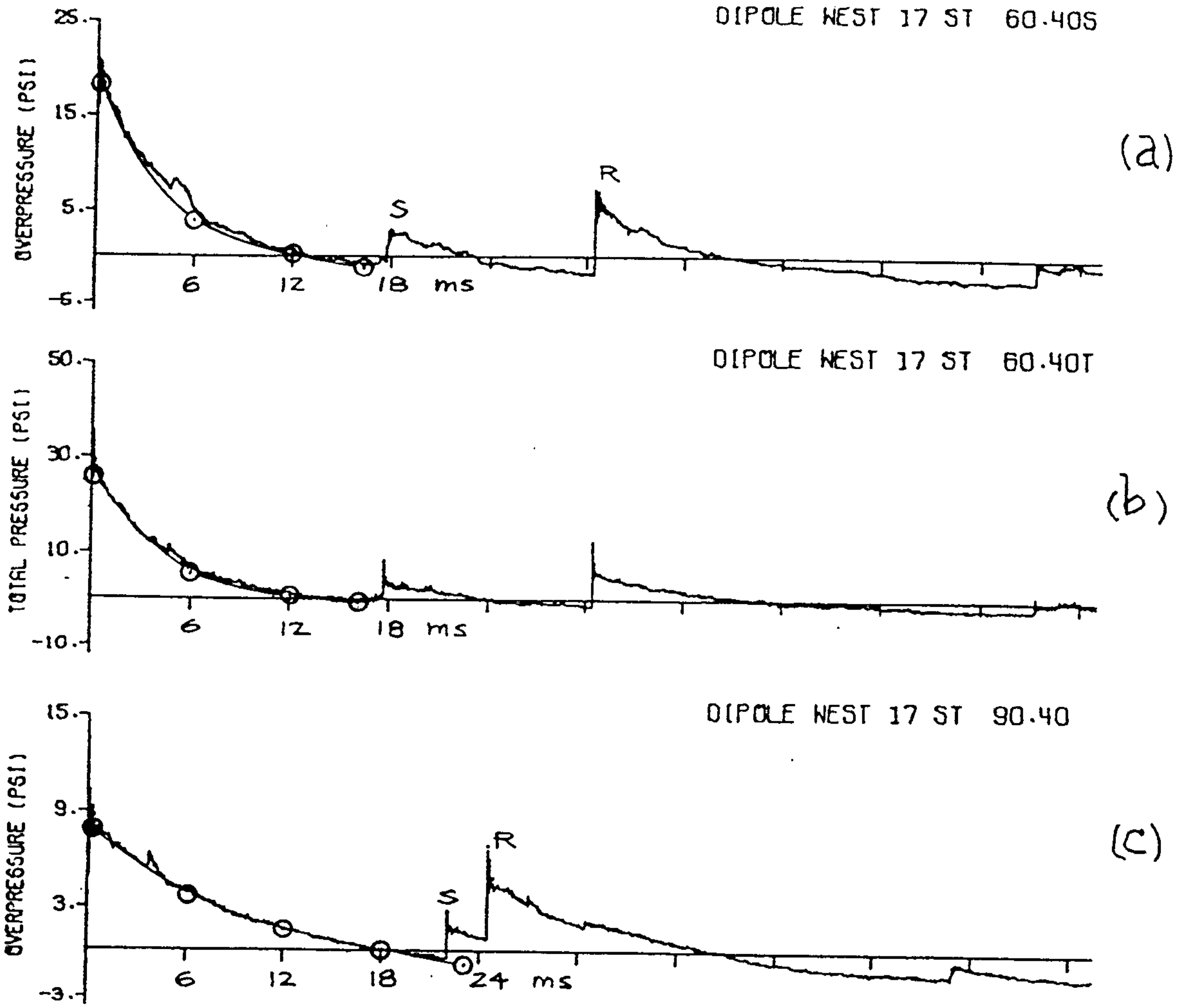

Figure 30. More comparisons of reconstructed pressure data with gauge records for DIPOLE WEST shot 17. (a) Overpressure at distance 18.3 m from the charge (2.27 m scaled to 1 kg TNT) and, (b) total overpressure at the same distance. The reconstructed data end at about the time of arrival of the secondary shock front S. (c) Overpressure at distance 27.3 m (3.39 scaled). The head-on pressure gauge at this position failed. Gauges were mounted in this experiment at distances as large as 75 m (9.4 m scaled), but either failed or were in the Mach stem region beneath the triple point trajectory.

The results presented in this and the previous figure are typical: many such comparisons between reconstructed data and gauge records were made, from other DIPOLE WEST experiments (shots 18-24), for example. Similar comparisons with particle trajectories measured using smoke tracers in these and other experiments were also made, with equal success.

Appendix A

DISPLAY PROGRAM SOURCE CODE

What follows is a listing of the FORTRAN 77 source code for the data retrieval and display program 'UVic/AIRBLAST'. This source code is being supplied as file 'AIRBLAST FORTRAN' on magnetic tape.

```
************************************************************************
*
*          UVic/AIRBLAST data retrieval and display program
*          ============================================
*
*            Copyright DND Canada, University of Victoria,
*           J.M. Dewey and D.J. McMillin, 20 February 1987.
*
************************************************************************

      CALL P1
      STOP
      END


*.P1
************************************************************************
* Subprogram to handle introductory dialogue.

      SUBROUTINE P1
      CHARACTER*21 PROGID/'UVic AIRBLAST/20feb87'/
      COMMON/BLOCK1/ IUNITS,IRATIO,IFILE,ITYPE,IATMOS,ISCALE,IBURST,IB2,
     ,IDATA,ITIME,IFORM,IPEAK
      COMMON/BLOCK2/ WS,TS,PS,CS,DS,CMASS,EMASS,TO,PO,RH,DO,CO,S,ST,HOB
      CHARACTER*1 UTEMP(2)/'C','F'/
      CHARACTER*2 UDIST(2)/' m','ft'/,UMASS(2)/'kg','lb'/
      CHARACTER*3 UPRESS(2)/'kPa','psi'/
      CHARACTER*7 FTYPE/'unknown'/

      GAMMA=1.4
      WS=1.
      TS=15.
      PS=101.32500
      CS=.34029205
      DS=1.2250140
      RS=PS/DS/288.15

C     Shake hands:
      WRITE(6,1) PROGID
1     FORMAT('1',58X,A21/' Do you want program information?  1=yes')
      READ(5,*,END=9000) INFORM
      IF(INFORM.NE.1) GO TO 090
      WRITE(6,2)
2     FORMAT(//' This program retrieves compiled air blast data.'//
     ,' The data were obtained by numerical reconstruction from a'/
     ,' body of experimental results, including shock and particle'/
     ,' trajectories, and are self-consistent.  They can be scaled'/
     ,' to simulate an explosion of any size under any conditions.'//
     ,' Various chemical HE and nuclear events can be simulated, in'/
     ,' free field, surface burst and height-of-burst configurations.'//
     ,' Output includes flow-field data in the entire middle distance'/
     ,' range (from about 10 atmospheres peak hydrostatic overpressure'/
     ,' down to just under 1/10th of an atmosphere), over the complete'/
     ,' positive (hydrostatic overpressure) phase of the blast wave.'///
     ,' You will be asked to input specific data (option codes, etc).'//
     ,' Instead of entering data or selecting options you may at any'/
     ,' time simply hit the "enter/return" key (a null entry) to back'/
     ,' up and restart or exit the program.  At certain times, when no'/
```

```
     ,' specific data are being sought, this action is required to get'/ABF00590
     ,' the program to continue executing.  Hit the enter/return key'/  ABF00600
     ,' now, for example, after clearing the screen if you wish.'//)    ABF00610
      READ(5,*,END=9000)                                                ABF00620
                                                                        ABF00630
C     Talk about units:                                                 ABF00640
  090 WRITE(6,3)                                                        ABF00650
3     FORMAT(' What UNITS would you like to use?'/                      ABF00660
     ,' 1=metric  [2=British]')                                         ABF00670
      READ(5,*,END=9002) IUNITS                                         ABF00680
      IF(IUNITS.NE.2) IUNITS=1                                          ABF00690
      IF(IUNITS.EQ.2) WRITE(6,4)                                        ABF00700
4     FORMAT(' Sorry,'/                                                 ABF00710
     ,' only metric units are available at present.'/                   ABF00720
     ,' British units option will be available soon.'/                  ABF00730
     ,' PLEASE CONTINUE.'/)                                             ABF00740
      IUNITS=1                                                          ABF00750
      WRITE(6,5) UPRESS(IUNITS)                                         ABF00760
5     FORMAT(                                                           ABF00770
     ,' Do you want pressures, etc. NORMALIZED?'/                       ABF00780
     ,' 1=yes, normalized to ambient atmospheric values'/               ABF00790
     ,' 2=no, in absolute units (pressures in ',A3,', for example)')    ABF00800
      READ(5,*,END=9002) IRATIO                                         ABF00810
      IF(IRATIO.NE.1) IRATIO=2                                          ABF00820
                                                                        ABF00830
C     Talk about output destination:                                    ABF00840
      WRITE(6,6)                                                        ABF00850
6     FORMAT(' Where do you want the output data SENT TO?'/             ABF00860
     ,' 1=this terminal  2=a disk file (for printing and/or plotting)') ABF00870
      READ(5,*,END=9002) ISEND                                          ABF00880
      IF(ISEND.EQ.1) IFILE=6                                            ABF00890
      IF(ISEND.NE.1) IFILE=1                                            ABF00900
      IF(IFILE.NE.6) WRITE(6,7)                                         ABF00910
7     FORMAT(' What would you like to NAME the file?'/                  ABF00920
     ,' (IBM CMS:  7 characters maximum, 1st alphabetic; "filedef" if)'/ABF00930
     ,' (a FILEDEF exists, otherwise FILEID will be "FILE yourname A")')ABF00940
      IF(IFILE.NE.6.AND.FTYPE.NE.'unknown') WRITE(6,8)                  ABF00950
8     FORMAT(' You may continue with the same file as before')          ABF00960
      IF(IFILE.NE.6) READ(5,9,END=9002) FTYPE                           ABF00970
9     FORMAT(A7)                                                        ABF00980
      IF(IFILE.NE.6.AND.FTYPE.NE.'FILEDEF') OPEN(UNIT=IFILE,FILE=FTYPE) ABF00990
      IF(IFILE.NE.6) CALL SCREEN('ZERO')                                ABF01000
      IF(IFILE.NE.6) CALL SCREEN(PROGID)                                ABF01010
                                                                        ABF01020
C     Talk about explosive type:                                        ABF01030
  100 WRITE(6,10)                                                       ABF01040
10    FORMAT(/' Explosive type?'/                                       ABF01050
     ,' 1=TNT  2=ANFO  3=Pentolite  4=hexogen  5=gaseous  6=nuclear')   ABF01060
      READ(5,*,END=9002) ITYPE                                          ABF01070
      IF(ITYPE.EQ.1) GO TO 101                                          ABF01080
      IF(ITYPE.EQ.2) GO TO 102                                          ABF01090
      IF(ITYPE.EQ.3) GO TO 103                                          ABF01100
      IF(ITYPE.EQ.4) GO TO 104                                          ABF01110
      IF(ITYPE.EQ.5) GO TO 105                                          ABF01120
      IF(ITYPE.EQ.6) GO TO 106                                          ABF01130
      GO TO 9002                                                        ABF01140
  101 FACTOR=1.00                                                       ABF01150
      GO TO 200                                                         ABF01160
  102 WRITE(6,12)                                                       ABF01170
12    FORMAT(                                                           ABF01180
```

```
     ,' ANFO will be treated as if it were TNT, equivalent for an'/
     ,' equal charge mass.  This is acceptable at middle distances.'/)
      FACTOR=1.00
      GO TO 200
  103 WRITE(6,13)
 13   FORMAT(
     ,' Pentolite will be treated as if it were TNT, but more'/
     ,' energetic by a factor of 22%, for an equal charge mass.'/
     ,' (For other factors specify TNT and adjust charge mass)'/)
      FACTOR=1.22
      GO TO 200
  104 WRITE(6,14)
 14   FORMAT(
     ,' Hexogen will be treated as if it were TNT, but more'/
     ,' energetic by a factor of  2%, for an equal charge mass.'/
     ,' (For other factors specify TNT and adjust charge mass)'/)
      FACTOR=1.02
      GO TO 200
  105 WRITE(6,15)
 15   FORMAT(
     ,' The gaseous mixture we have most experience with so far is a'/
     ,' stoichiometric mixture of propane and oxygen.  We found that'/
     ,' 1.25 cubic meters of this was energy equivalent to 1 kg TNT,'/
     ,' and, knowing this, we can simulate a variety of events.'//
     ,' Do you want to continue?'/
     ,' 1=yes, propane-oxygen')
      READ(5,*,END=9002) INDEX
      IF(INDEX.NE.1) GO TO 9002
      WRITE(6,16) UDIST(IUNITS)
 16   FORMAT(' What volume of gas? (cubic ',A2,')'/
     ,' (Original experiment= 14479 cubic m or 20 tons TNT equivalent)'/
     ,' (You may enter any volume you wish, however)')
      READ(5,*,END=9002) VOLUME
      IF(VOLUME.LE.0.) GO TO 9002
      FACTOR=1.2530899
      CMASS=VOLUME
      GO TO 204
  106 WRITE(6,19)
 19   FORMAT(
     ,' Nuclear data will be derived from TNT data '//
     ,' One kiloton of TNT (=907184.86 kg) releases the same amount of'/
     ,' energy as the standard 1 kiloton nuclear device, though a much'/
     ,' greater portion of this energy gets into the blast wave in the'/
     ,' TNT case.  We are using an adjustment factor of 52%'//
     ,' Over the range of middle distances, the adjusted TNT data give'/
     ,' a good approximation to the standard nuclear event.'//
     ,' What energy yield?  (kilotons, TNT equivalent energy yield)')
      READ(5,*,END=9002) YIELD
      IF(YIELD.LE.0.) GO TO 9002
      FACTOR=907184.86*0.52
      CMASS=YIELD
      GO TO 204

C     Talk about charge mass:
  200 WRITE(6,20) UMASS(IUNITS)
 20   FORMAT(' Charge mass?  (',A2,')')
      READ(5,*,END=9002) CMASS
      IF(CMASS.LE.0.) GO TO 200
  204 EMASS=CMASS*FACTOR
      IF(FACTOR.NE.1.) WRITE(6,21) EMASS,UMASS(IUNITS)
```

```
21    FORMAT(' TNT-equivalent mass=',E14.7,1X,A2/)                      ABF01790
                                                                        ABF01800
C     Talk about ambient atmosphere:                                    ABF01810
      WRITE(6,22)                                                       ABF01820
22    FORMAT(' Ambient atmosphere?'/                                    ABF01830
     ,' 1=standard sea level  [2=altitude dependent]  3=user defined')  ABF01840
      READ(5,*,END=9002) IATMOS                                         ABF01850
      IF(IATMOS.EQ.1) GO TO 201                                         ABF01860
      IF(IATMOS.EQ.2) GO TO 202                                         ABF01870
      IATMOS=3                                                          ABF01880
      WRITE(6,23) UTEMP(IUNITS)                                         ABF01890
23    FORMAT(' Air temperature? (',A1,')')                              ABF01900
      READ(5,*,END=9002) TO                                             ABF01910
      WRITE(6,24) UPRESS(IUNITS)                                        ABF01920
24    FORMAT(' Air pressure? (',A3,')')                                 ABF01930
      READ(5,*,END=9002) PO                                             ABF01940
      WRITE(6,25)                                                       ABF01950
25    FORMAT(' Relative humidity? (%)')                                 ABF01960
      READ(5,*,END=9002) RH                                             ABF01970
      CALL VAPOUR(P,TO,RH)                                              ABF01980
      DO=PO/(TO+273.15)/RS                                              ABF01990
      CO=SQRT(GAMMA*PO*1000./DO)/SQRT(1.-P/PO*(1.334/GAMMA-5./8.))/1000.ABF02000
      ISCALE=0                                                          ABF02010
      GO TO 203                                                         ABF02020
  201 TO=TS                                                             ABF02030
      PO=PS                                                             ABF02040
      CO=CS                                                             ABF02050
      DO=DS                                                             ABF02060
      ISCALE=0                                                          ABF02070
      IF(EMASS.EQ.WS) ISCALE=1                                          ABF02080
      GO TO 203                                                         ABF02090
  202 WRITE(6,26) UDIST(IUNITS)                                         ABF02100
26    FORMAT(' Altitude?  (',A2,')')                                    ABF02110
      READ(5,*,END=9002) ZO                                             ABF02120
      CALL ATMOS(ZO)                                                    ABF02130
      ISCALE=0                                                          ABF02140
  203 IF(IATMOS.NE.1) WRITE(6,27) CO,UDIST(IUNITS)                      ABF02150
27    FORMAT(' Computed sound speed=',F8.5,A2,'/ms'/)                   ABF02160
      S=(EMASS/WS*PS/PO)**.33333333                                     ABF02170
      ST=S*CS/CO                                                        ABF02180
      IF(ISCALE.NE.1) WRITE(6,29) S,ST                                  ABF02190
29    FORMAT(' Scales relative to 1 kg TNT in standard sea level air=', ABF02200
     ,F10.3,' for distance'/55X,F10.3,' for time'/)                     ABF02210
C     (later define ISCALE=0/1/2/3 as EMASS=1kg/1000kg/1kT/1kT ?)       ABF02220
                                                                        ABF02230
C     Talk about charge-ground configuration:                           ABF02240
  300 WRITE(6,30)                                                       ABF02250
30    FORMAT(' Charge-ground configuration?'/                           ABF02260
     ,' 1=free field  2=surface burst  [3=height of burst]'/            ABF02270
     ,' (spherical)   (hemispherical)  (reflected free field)')         ABF02280
      READ(5,*,END=9002) IBURST                                         ABF02290
      IF(IBURST.LE.0.OR.IBURST.GE.4) GO TO 300                          ABF02300
                                                                        ABF02310
C     Output summary of event:                                          ABF02320
      HOB=0.                                                            ABF02330
      IF(IFILE.NE.6) CALL EVENT                                         ABF02340
                                                                        ABF02350
C     Enter appropriate subprogram:                                     ABF02360
      IF(IBURST.EQ.1) CALL P234(INDEX)                                  ABF02370
      IF(IBURST.EQ.2) CALL P234(INDEX)                                  ABF02380
```

```
      IF(IBURST.EQ.3) GO TO 301
      IF(INDEX.NE.0) GO TO 9002
      GO TO 9001
  301 WRITE(6,31) UDIST(IUNITS)
 31   FORMAT(' Height of charge above ground surface? (',A2,')')
      READ(5,*,END=9002) HOB
      IF(HOB.LT.0.) GO TO 301
      IF(HOB.EQ.0.) GO TO 302
      IF(IFILE.NE.6) CALL EVENT
      CALL P234(INDEX)
      IF(INDEX.NE.0) GO TO 9002
      GO TO 9001
  302 WRITE(6,32)
 32   FORMAT(' If you mean surface burst, please specify code 2')
      GO TO 300

C     Conclude:
 9000 REWIND 5
      GO TO 090
 9001 WRITE(6,9901)
 9901 FORMAT(' Normal exit.  Bye.')
      IF(IFILE.EQ.2) CLOSE(UNIT=1)
      RETURN
 9002 WRITE(6,9902)
 9902 FORMAT(' Restart?  1=yes')
      REWIND 5
      READ(5,*,END=9003) INDEX
      IF(INDEX.EQ 1) GO TO 090
 9003 WRITE(6,9903)
 9903 FORMAT(' User terminated.  Bye.')
      IF(IFILE.EQ.2) CLOSE(UNIT=1)
      RETURN
      END


*.P234
************************************************************************
* Subprogram to handle free field, surface burst and HOB/ground cases

      SUBROUTINE P234(INDEX)
      COMMON/BLOCK1/ IUNITS,IRATIO,IFILE,ITYPE,IATMOS,ISCALE,IBURST,IB2,
     ,IDATA,ITIME,IFORM,IPEAK
      COMMON/BLOCK2/ WS,TS,PS,CS,DS,CMASS,EMASS,TO,PO,RH,DO,CO,S,ST,HOB
      REAL*4 TIMES(100),RADII(100),VALUES(100)
      INTEGER*4 NTMMAX/100/,NRMMAX/100/,NVMMAX/100/
      CHARACTER*2 UDIST(2)/' m','ft'/

      PARAMETER(NRM=100,NTM=125,NDM=15)
      REAL*4 R(NRM),T1(NRM),P1(NRM),D1(NRM),U1(NRM),T2(NRM),P2(NRM),
     ,D2(NRM),U2(NRM),T(NTM),P(NRM,NDM),D(NRM,NDM),U(NRM,NDM),
     ,W1(NRM),W2(NRM),W3(NRM),W4(NRM),W5(NRM),W6(NRM),W7(NRM),W8(NRM)
      INTEGER*4 NJ(NRM),J1(NRM)
      IF(NRM.LT.3*NDM) STOP 1

C     Input source data:
      CALL INPUT(R,T1,P1,D1,U1,T2,P2,D2,U2,T,J1,NJ,P,D,U,NR,NT,NRM,NTM,
     ,NDM)

C     Check regular reflection option:
      IF(IBURST.EQ.3) CALL FFR(R,T1,P1,NR,NRM,W1,W2,W3,W4,W5,W6,NR2,RD)
```

```
                                                                        ABF02990
C     Talk about data kind:                                             ABF03000
 091  WRITE(6,1)                                                        ABF03010
1     FORMAT(/' What kind of data would you like to see?'/              ABF03020
     ,        ' (data codes= 0,1,...,18, or 99 for menu)')              ABF03030
      READ(5,*,END=9001) IDATA                                          ABF03040
      GO TO 093                                                         ABF03050
 092  WRITE(6,2)                                                        ABF03060
2     FORMAT(/                                                          ABF03070
     ,'  0=all data at one point       10=particle velocities          '/ ABF03080
     ,'  1=shock front radii           11=gas densities                '/ ABF03090
     ,'  2=times of shock arrival      12=temperatures                 '/ ABF03100
     ,'  3=shock front velocities      13=sound speeds                 '/ ABF03110
     ,'  4=particle trajectories       14=entropy changes              '//ABF03120
     ,'  5=hydrostatic pressures       15=energy densities             '/ ABF03130
     ,'  6=dynamic pressures           16=available work densities     '/ ABF03140
     ,'  7=total head pressures        17=energy integrals             '/ ABF03150
     ,'  8=positive durations          18=work integrals               '/ ABF03160
     ,'  9=pressure impulses                                           '/ ABF03170
     ,'                                99=repeat menu                  ') ABF03180
      READ(5,*,END=9001) IDATA                                          ABF03190
 093  IF(IDATA.EQ.0) GO TO 098                                          ABF03200
      IF(IDATA.GE.1.AND.IDATA.LE.4) GO TO 100                           ABF03210
      IF(IDATA.GE.5 AND.IDATA.LE.18) GO TO 200                          ABF03220
      IF(IDATA.EQ.99) GO TO 092                                         ABF03230
      WRITE(6,3)                                                        ABF03240
3     FORMAT(' Invalid response, try again:')                           ABF03250
      GO TO 091                                                         ABF03260
                                                                        ABF03270
C     Output all known data at a fixed point:                           ABF03280
 098  IF(IBURST.EQ.1) WRITE(6,6) UDIST(IUNITS)                          ABF03290
      IF(IBURST.GE.2) WRITE(6,7) UDIST(IUNITS)                          ABF03300
6     FORMAT(' At what distance from charge centre? (',A2,')')          ABF03310
7     FORMAT(' At what distance from ground zero? (',A2,')')            ABF03320
      RL=R(1)*S                                                         ABF03330
      RU=R(NR)*S                                                        ABF03340
      IF(IBURST.EQ.3) RL=W1(1)*S                                        ABF03350
      IF(IBURST.EQ.3) RU=W1(NR2)*S                                      ABF03360
      WRITE(6,8) RL,RU                                                  ABF03370
8     FORMAT(' (between',F8.3,' and',F9.3,')')                          ABF03380
      READ(5,*,END=9001) RADIUS                                         ABF03390
      IF(IBURST.NE.3.AND.RADIUS.LE.0.) GO TO 098                        ABF03400
      IF(IBURST.EQ.3.AND.RADIUS.LT.0.) GO TO 098                        ABF03410
      RADII(1)=RADIUS                                                   ABF03420
      NRADII=1                                                          ABF03430
      INDEX=1                                                           ABF03440
      IF(IBURST.EQ.3) GO TO 099                                         ABF03450
      CALL FF4(INDEX,RADII,NRADII,R,T1,P1,D1,U1,T2,P2,D2,U2,T,J1,NJ,P,D,ABF03460
     ,U,NR,NT,NRM,NTM,NDM,W1,W2,W3,W4,W5,W6,W7)                         ABF03470
      CALL FF0(INDEX,RADIUS,R,T1,P1,D1,U1,W5,W6,NR,NDM,W1,W2,W3)        ABF03480
      GO TO 9000                                                        ABF03490
 099  CALL RR4(INDEX,RADII,NRADII,W1,W2,W3,W4,W5,NR2,W7,W8)             ABF03500
      CALL RR0(INDEX,RADIUS,W1,W2,W3,W4,W5,W6,W7,W8,NR2,1)              ABF03510
      GO TO 9000                                                        ABF03520
                                                                        ABF03530
C     Output basic shock and particle trajectory data:                  ABF03540
 100  CONTINUE                                                          ABF03550
      IF(IBURST.NE.3) CALL FF1(INDEX,R,T1,P1,W1,W2,NR)                  ABF03560
      IF(IBURST.EQ.3) CALL FF1(INDEX,W1,W2,W6,W7,W8,NR2)                ABF03570
      GO TO 9000                                                        ABF03580
```

```
                                                                        ABF03590
C     Talk about format:                                                ABF03600
  200 ITIME=0                                                           ABF03610
      IF(IDATA.GE.5.AND.IDATA.LE.7) ITIME=1                             ABF03620
      IF(IDATA.GE.10.AND.IDATA.LE.16) ITIME=1                           ABF03630
      IF(ITIME.EQ.0) GO TO 201                                          ABF03640
      WRITE(6,20)                                                       ABF03650
20    FORMAT(' In what format would you like the data output?'/         ABF03660
     ,' 1=spatial profiles  2=time histories'/                          ABF03670
     ,' (3=guide to contouring in R-T plane)')                          ABF03680
      READ(5,*,END=9001) IFORM                                          ABF03690
      IF(IFORM.EQ.1) GO TO 300                                          ABF03700
      IF(IFORM.EQ.2) GO TO 400                                          ABF03710
                                                                        ABF03720
C     Talk about how to get what is wanted, in some other way:          ABF03730
      IF(IFORM.EQ.3) WRITE(6,21)                                        ABF03740
21    FORMAT(/                                                          ABF03750
     ,' To obtain data for contours, specify a time history format and'/ABF03760
     ,' use the reverse interpolation option to locate points that can'/ABF03770
     ,' be connected, becoming specific contour lines in the RT plane'/)ABF03780
      GO TO 200                                                         ABF03790
  201 IF(IDATA.EQ.8) WRITE(6,22)                                        ABF03800
22    FORMAT(/                                                          ABF03810
     ,' To obtain positive durations specify the actual data code, say'/ABF03820
     ,' 5 for static overpressure, and a time history format.  Use the'/ABF03830
     ,' reverse interpolation option to get the positive duration.'//   ABF03840
     ,' The all-data-at-a-point code (99) can be specified for static'/ ABF03850
     ,' overpressure positive duration, as an alternative.'/)           ABF03860
      IF(IDATA.EQ.9) WRITE(6,23)                                        ABF03870
23    FORMAT(/                                                          ABF03880
     ,' To obtain pressure impulses specify the actual data code, say'/ ABF03890
     ,' 5 for static overpressure, and a time history format.  Use the'/ABF03900
     ,' reverse interpolation option to get the positive duration, and'/ABF03910
     ,' the subsequently-offered integration option to get impulse.'//  ABF03920
     ,' The all-data-at-a-point code (99) can be specified for static'/ ABF03930
     ,' overpressure impulse, as an alternative.'/)                     ABF03940
      IF(IDATA.EQ.17.OR.IDATA.EQ.18) WRITE(6,24)                        ABF03950
24    FORMAT(/                                                          ABF03960
     ,' To obtain energy and work density integrals, specify the data'/ ABF03970
     ,' code 17 or 18 accordingly, and a wave profile format.  Use the'/ABF03980
     ,' reverse interpolation option to get positive phase length, and'/ABF03990
     ,' the subsequently-offered integration option to get the volume'/ ABF04000
     ,' integral, that is, the total energy or available work.'/)       ABF04010
      GO TO 091                                                         ABF04020
                                                                        ABF04030
C     Output peak profiles:                                             ABF04040
  300 IF(ITIME.EQ.1) WRITE(6,30)                                        ABF04050
30    FORMAT(' Profile of PEAK data values only?'/                      ABF04060
     ,' 1=yes  2=no, full wave profiles')                               ABF04070
      IF(ITIME.EQ.1) READ(5,*,END=9001) IPEAK                           ABF04080
      IF(ITIME.EQ.1.AND.IPEAK.EQ.2) GO TO 301                           ABF04090
      IF(IBURST.NE.3) CALL FF2(INDEX,R,P1,D1,U1,NR,W1,W2)               ABF04100
      IF(IBURST.EQ.3) CALL FF2(INDEX,W1,W3,W4,W5,NR2,W7,W8)             ABF04110
      GO TO 9000                                                        ABF04120
                                                                        ABF04130
C     Output wave profiles:                                             ABF04140
  301 TL=T(1)*ST                                                        ABF04150
      TU=T(NT)*ST                                                       ABF04160
      IF(IBURST.EQ.3) TL=W2(1)*ST                                       ABF04170
      IF(IBURST.EQ.3) TU=W2(NR2)*ST                                     ABF04180
```

```
  302 WRITE(6,31) TL,TU
31    FORMAT(' Wave profiles at how many different times?'/
     ,         ' (time range= from',F9.3,' to',F9.3,' ms)')
      READ(5,*,END=9001) NTIMES
      IF(NTIMES.LE.0) GO TO 302
      IF(NTIMES.GT.NTMMAX) GO TO 302
      WRITE(6,32)
32    FORMAT(' What are the times?')
      READ(5,*,END=9001) (TIMES(I),I=1,NTIMES)
      IF(IBURST.NE.3) CALL FF3(INDEX,TIMES,NTIMES,R,T1,P1,D1,U1,T2,P2,
     ,D2,U2,T,J1,NJ,P,D,U,NR,NT,NRM,NTM,NDM,W1,W2,W3,W4,W5,W6,W7)
      IF(IBURST.EQ.3) CALL RR3(INDEX,TIMES,NTIMES,W1,W2,W3,W4,W5,NR2,W7,
     ,W8)
      GO TO 9000

C     Output time histories:
  400 RL=R(1)*S
      RU=R(NR)*S
      IF(IBURST.EQ.3) RL=W1(1)*S
      IF(IBURST.EQ.3) RU=W1(NR2)*S
  401 WRITE(6,40) RL,RU,UDIST(IUNITS)
40    FORMAT(' Time histories at how many different distances?'/
     ,         ' (distance range= from',F9.3,' to',F9.3,A3,')')
      IF(IBURST.GT.1) WRITE(6,42)
42    FORMAT(' (distances are measured from GZ)')
      IF(IBURST.EQ.3) WRITE(6,43)
43    FORMAT(' (distances above define RR only)')
      READ(5,*,END=9001) NRADII
      IF(NRADII.LE.0) GO TO 401
      IF(NRADII.GT.NRMMAX) GO TO 401
      WRITE(6,41)
41    FORMAT(' What are the distances?')
      READ(5,*,END=9001) (RADII(I),I=1,NRADII)
      INDEX=0
      IF(IBURST.NE.3) CALL FF4(INDEX,RADII,NRADII,R,T1,P1,D1,U1,T2,P2,
     ,D2,U2,T,J1,NJ,P,D,U,NR,NT,NRM,NTM,NDM,W1,W2,W3,W4,W5,W6,W7)
      IF(IBURST.EQ.3) CALL RR4(INDEX,RADII,NRADII,W1,W2,W3,W4,W5,NR2,W7,
     ,W8)
      GO TO 9000

C     Conclude:
 9000 IF(INDEX.EQ.1) GO TO 9001
      WRITE(6,9900)
 9900 FORMAT(/' Do you want other data?  99=yes (repeat menu)'/
     ,         ' or you may enter the data code (0 through 18)')
      READ(5,*,END=9001) IDATA
      IF(IDATA.EQ.99) GO TO 092
      GO TO 093

 9001 INDEX=1
      REWIND 5
      RETURN
      END


*.FF0
************************************************************************
* Subprogram to output all known data at one point (free field).

      SUBROUTINE FF0(INDEX,RADIUS,R,T1,P1,D1,U1,X,YI,NR,NDM,P,D,U)
```

```
      REAL*4 R(NR),T1(NR),P1(NR),D1(NR),U1(NR),X(NDM),YI(3,NDM),P(NDM)  ABF04790
      REAL*4 D(NDM),U(NDM),M,M2                                         ABF04800
      COMMON/BLOCK1/ IUNITS,IRATIO,IFILE,ITYPE,IATMOS,ISCALE,IBURST,IB2,ABF04810
     ,IDATA,ITIME,IFORM,IPEAK                                           ABF04820
      COMMON/BLOCK2/ WS,TS,PS,CS,DS,CMASS,EMASS,TO,PO,RH,DO,CO,S,ST,HOB ABF04830
      CHARACTER*1 UTEMP(2)/'C','F'/                                     ABF04840
      CHARACTER*2 UDIST(2)/' m','ft'/                                   ABF04850
      CHARACTER*3 UPRESS(2)/'kPa','psi'/                                ABF04860
      CHARACTER*5 UVEL(2)/'m/ms ','ft/ms'/                              ABF04870
      CALL SCREEN('                point data')                         ABF04880
                                                                        ABF04890
C     Interpolate for time of arrival:                                  ABF04900
      RI=RADIUS/S                                                       ABF04910
      CALL LINT(RI,TOA,R,T1,NR)                                         ABF04920
      TOUT=TOA*ST                                                       ABF04930
                                                                        ABF04940
C     Compute peak data:                                                ABF04950
      CALL EXPAND(YI(1,1),YI(2,1),YI(3,1),PVM,SSR,HOP,DEN,DYP,TOP,END,  ABF04960
     ,TEMP,PVML,ENT,WORK,M,1)                                           ABF04970
      M2=M*CO*1000.                                                     ABF04980
      HOP2=HOP*PO                                                       ABF04990
      DYP2=DYP*PO                                                       ABF05000
      TOP2=TOP*PO                                                       ABF05010
      TEMP2=TEMP*(273.15+TO)-273.15                                     ABF05020
      DEN2=DEN*DO                                                       ABF05030
      SSR2=SSR*CO*1000                                                  ABF05040
      PVM2=PVM*CO*1000.                                                 ABF05050
                                                                        ABF05060
C     Output data:                                                      ABF05070
      WRITE(IFILE,1) RADIUS,TOUT,M,M2,HOP,HOP2,DYP,DYP2,TOP,TOP2,       ABF05080
     ,DEN,DEN2,TEMP,TEMP2,SSR,SSR2,PVML,PVM2,END,WORK,ENT               ABF05090
1     FORMAT(                                                           ABF05100
     ,' Distance from charge centre=',F8.3,' m'/                        ABF05110
     ,' Time of shock front arrival=',F8.3,' ms'/                       ABF05120
     ,' Shock front velocity, Mach #',F8.3,'          ',F8.3,' m/s'//   ABF05130
     ,' Peak hydrostatic overpressure ratio=',F7.3,'   ',F8.3,' kPa'/   ABF05140
     ,'  "   dynamic pressure ratio        =',F7.3,'   ',F8.3,' kPa'/   ABF05150
     ,'  "   total head overpressure ratio =',F7.3,'   ',F8.3,' kPa'/   ABF05160
     ,'  "   density ratio                 =',F7.3,'   ',F8.3,' kg/m3'/ABF05170
     ,'  "   temperature ratio             =',F7.3,'   ',F8.3,' C'/     ABF05180
     ,'  "   sound speed ratio             =',F7.3,'   ',F8.3,' m/s'/   ABF05190
     ,'  "   particle velocity, local Mach #',F7.3,'   ',F8.3,' m/s'//  ABF05200
     ,' Peak excess energy  =',F9.6,' J/cc'/                            ABF05210
     ,'  "   available work =',F9.6,'  "  '/                            ABF05220
     ,' Entropy change      =',F9.6,' J/K/g')                           ABF05230
                                                                        ABF05240
C     Try for positive duration:                                        ABF05250
      LENGTH=INDEX                                                      ABF05260
      DO 101 N=1,LENGTH                                                 ABF05270
      P(N)=YI(1,N)-1.                                                   ABF05280
      IF(IRATIO.EQ.2) P(N)=P(N)*PO                                      ABF05290
  101 CONTINUE                                                          ABF05300
      TPOS=0.                                                           ABF05310
      IF(LENGTH.GE.2.AND.P(LENGTH).LE.0.) CALL LINT(0.,TPOS,P,X,LENGTH) ABF05320
      IF(TPOS.EQ.0.) GO TO 104                                          ABF05330
      IF(TPOS.GT.0.) WRITE(IFILE,2) TPOS                                ABF05340
2     FORMAT(/' Positive duration (static overpressure)=',F8.3,' ms')   ABF05350
                                                                        ABF05360
C     Try for impulse:                                                  ABF05370
      YTIME=0.                                                          ABF05380
```

```
      XL=0.
      YL=P(1)
      AREA=0.
      DO 102 I=2,LENGTH
      IF(X(I).GT.TPOS) GO TO 103
      AREA=AREA+(X(I)-XL)*(P(I)+YL)/2.
      XL=X(I)
      YL=P(I)
  102 CONTINUE
  103 AREA=AREA+(TPOS-XL)*(YTIME+YL)/2.
      IF(IRATIO.EQ.1) WRITE(IFILE,3) AREA
      IF(IRATIO.EQ.2) WRITE(IFILE,4) AREA,UPRESS(IUNITS)
3     FORMAT(' Static overpressure impulse=',F9.3,' atm-ms')
4     FORMAT(' Static overpressure impulse=',F9.3,1X,A3,'-ms')
      GO TO 200
  104 WRITE(6,5)
5     FORMAT(/' Time of positive duration cannot be determined,'/
     ,' nor can the static overpressure impulse')

C     Check time options:
  200 WRITE(6,7) X(LENGTH)
7     FORMAT(/
     ,' For more data, enter a time (ms after time of arrival); or 0,'/
     ,' for a time history of all pressures, maximum T-TO=',F8.3,' ms.')
  201 CONTINUE
      READ(5,*,END=9001) TIME
      IF(TIME.LT.0.) GO TO 200
      IF(TIME.EQ.0.) GO TO 300
      TI=TIME/ST
      IF(LENGTH.LE.2) GO TO 209
      DO 202 N=1,LENGTH
      P(N)=YI(1,N)
      D(N)=YI(2,N)
      U(N)=YI(3,N)
      IF(IRATIO.EQ.2) P(N)=P(N)*PO
      IF(IRATIO.EQ.2) D(N)=D(N)*DO
      IF(IRATIO.EQ.2) U(N)=U(N)*CO*1000.
  202 CONTINUE
      CALL LINT(TI,PI,X,P,LENGTH)
      CALL LINT(TI,DI,X,D,LENGTH)
      CALL LINT(TI,UI,X,U,LENGTH)
      CALL EXPAND(PI,DI,UI,PVM,SSR,HOP,DEN,DYP,TOP,END,TEMP,PVML,ENT,
     ,WORK,M,0)
      HOP2=HOP*PO
      DYP2=DYP*PO
      TOP2=TOP*PO
      TEMP2=TEMP*(273.15+TO)-273.15
      DEN2=DEN*DO
      SSR2=SSR*CO*1000.
      PVM2=PVM*CO*1000.
      WRITE(IFILE,21) TIME,HOP,HOP2,DYP,DYP2,TOP,TOP2,
     ,DEN,DEN2,TEMP,TEMP2,SSR,SSR2,PVML,PVM2,END,WORK,ENT
21    FORMAT(
     ,' Time after shock front arrival=',F8.3,' ms'/
     ,' Peak hydrostatic overpressure ratio=',F7.3,'    ',F8.3,' kPa'/
     ,'  "   dynamic pressure ratio          =',F7.3,'    ',F8.3,' kPa'/
     ,'  "   total head overpressure ratio  =',F7.3,'    ',F8.3,' kPa'/
     ,'  "   density ratio                   =',F7.3,'    ',F8.3,' kg/m3'/
     ,'  "   temperature ratio               =',F7.3,'    ',F8.3,' C'/
     ,'  "   sound speed ratio               =',F7.3,'    ',F8.3,' m/s'/
```

```
     ,'  "   particle velocity, local Mach #',F7.3,'     ',F8.3,' m/s'//
     ,' Peak excess energy  =',F9.6,' J/cc'/
     ,'  "   available work =',F9.6,'  "  '/
     ,' Entropy change      =',F9.6,' J/K/g')
      WRITE(6,22)
 22   FORMAT(/' What other time?')
      GO TO 201
  209 WRITE(6,23)
 23   FORMAT(' Insufficient data are available at this position')
      GO TO 9000

C     Output pressure histories:
  300 WRITE(IFILE,31)
 31   FORMAT(/' T-TO(ms)  Pstatic  Density  Velocity      Pd      Ptot',
     .'   Energy     Work')
      DO 301 N=1,LENGTH
      CALL EXPAND(YI(1,N),YI(2,N),YI(3,N),PVM,SSR,HOP,DEN,DYP,TOP,END,
     ,TEMP,PVML,ENT,WORK,M,1)
      IF(IRATIO.EQ.2) HOP=HOP*PO
      IF(IRATIO.EQ.2) DYP=DYP*PO
      IF(IRATIO.EQ.2) TOP=TOP*PO
      IF(IRATIO.EQ.2) DEN=DEN*DO
      IF(IRATIO.EQ.2) PVM=PVM*CO*1000.
      WRITE(IFILE,32) X(N),HOP,DEN,PVM,DYP,TOP,END,WORK
 32   FORMAT(8F9.3)
  301 CONTINUE
      IF(IRATIO.EQ.1) WRITE(IFILE,33)
      IF(IRATIO.EQ.2) WRITE(IFILE,34)
 33   FORMAT(/
     ,' Pressure, density and flow velocity are normalized to ambient'/
     ,' values.  Static and total pressures are overpressures.  Energy'/
     ,' is excess energy density.  It and available work are in J/cc.')
 34   FORMAT(/
     ,' Pressures are in kPa, density is in kg/m**3 and flow velocity,'/
     ,' in m/s.  Static and total pressures are overpressures   Energy'/
     ,' is excess energy density.  It and available work are in J/cc.')

 9000 INDEX=0
      RETURN
 9001 REWIND 5
      INDEX=0
      RETURN
      END



*.FF1
************************************************************************
* Subprogram to output basic trajectory data (free field).

      SUBROUTINE FF1(INDEX,R,T1,P1,X,Y,NR)
      REAL*4 R(NR),T1(NR),P1(NR),X(NR),Y(NR),GAMMA/1.4/
      COMMON/BLOCK1/ IUNITS,IRATIO,IFILE,ITYPE,IATMOS,ISCALE,IBURST,IB2,
     ,IDATA,ITIME,IFORM,IPEAK
      COMMON/BLOCK2/ WS,TS,PS,CS,DS,CMASS,EMASS,TO,PO,RH,DO,CO,S,ST,HOB
      CHARACTER*2 UDIST(2)/' m','ft'/
      CHARACTER*5 UVEL(2)/'m/ms ','ft/ms'/
      CHARACTER*6 UNITS(2)
      IF(IDATA.NE.4) CALL SCREEN('basic trajectory data')

```

```
C     Branch to output:                                                 ABF06590
      IF(IDATA.EQ.1) GO TO 100                                          ABF06600
      IF(IDATA.EQ.2) GO TO 200                                          ABF06610
      IF(IDATA.EQ.3) GO TO 300                                          ABF06620
      IF(IDATA.EQ.4) GO TO 400                                          ABF06630
      PRINT*,' ERROR(FF1): INVALID CALL'                                ABF06640
      GO TO 9000                                                        ABF06650
                                                                        ABF06660
C     Output shock front radii:                                         ABF06670
  100 DO 101 N=1,NR                                                     ABF06680
      X(N)=T1(N)*ST                                                     ABF06690
      Y(N)=R(N)*S                                                       ABF06700
  101 CONTINUE                                                          ABF06710
      WRITE(IFILE,10) UDIST(IUNITS),(X(I),Y(I),I=1,NR)                  ABF06720
      IF(IBURST.EQ.3) WRITE(6,11)                                       ABF06730
10    FORMAT(' Shock front radii:'/                                     ABF06740
     ,'     T(ms)      R(',A2,')'/(F9.3,F10.3))                         ABF06750
11    FORMAT(/                                                          ABF06760
     ,' (''Shock front'' is reflection point in RR part of profile')    ABF06770
      UNITS(1)='ms'                                                     ABF06780
      UNITS(2)=UDIST(IUNITS)                                            ABF06790
      GO TO 500                                                         ABF06800
                                                                        ABF06810
C     Output times of shock arrival:                                    ABF06820
  200 DO 201 N=1,NR                                                     ABF06830
      X(N)=R(N)*S                                                       ABF06840
      Y(N)=T1(N)*ST                                                     ABF06850
  201 CONTINUE                                                          ABF06860
      WRITE(IFILE,20) UDIST(IUNITS),(X(I),Y(I),I=1,NR)                  ABF06870
      IF(IBURST.EQ.3) WRITE(6,21)                                       ABF06880
20    FORMAT(' Times of shock arrival:'/                                ABF06890
     ,'     R(',A2,')      T(ms)'/(F9.3,F10.3))                         ABF06900
21    FORMAT(/                                                          ABF06910
     ,' (''Shock front'' is reflection point in RR part of profile')    ABF06920
      UNITS(1)=UDIST(IUNITS)                                            ABF06930
      UNITS(2)='ms'                                                     ABF06940
      GO TO 500                                                         ABF06950
                                                                        ABF06960
C     Output shock front velocities:                                    ABF06970
  300 DO 301 N=1,NR                                                     ABF06980
      X(N)=R(N)*S                                                       ABF06990
      Y(N)=SQRT((P1(N)-1.)*(GAMMA+1.)/2./GAMMA+1.)                      ABF07000
      IF(IBURST.EQ.3) Y(N)=P1(N)                                        ABF07010
      IF(IRATIO.EQ.2) Y(N)=Y(N)*CO                                      ABF07020
  301 CONTINUE                                                          ABF07030
      IF(IRATIO.EQ.1) WRITE(IFILE,31) UDIST(IUNITS)                     ABF07040
      IF(IRATIO.EQ.2) WRITE(IFILE,32) UDIST(IUNITS),UVEL(IUNITS)        ABF07050
      WRITE(IFILE,33) (X(I),Y(I),I=1,NR)                                ABF07060
      IF(IBURST.EQ.3) WRITE(6,34)                                       ABF07070
31    FORMAT(' Shock front velocities:'/                                ABF07080
     ,'     R(',A2,')  V(Mach#)')                                       ABF07090
32    FORMAT(' Shock front velocities:'/                                ABF07100
     ,'     R(',A2,')  V(',A5,')')                                      ABF07110
33    FORMAT(F9.3,F10.3)                                                ABF07120
34    FORMAT(/                                                          ABF07130
     ,' (Velocities are those of the reflection point along the ground'/ABF07140
     ,' surface in the regular reflection part of the velocity profile')ABF07150
      UNITS(1)=UDIST(IUNITS)                                            ABF07160
      UNITS(2)=' '                                                      ABF07170
      IF(IRATIO.EQ.2) UNITS(2)=UVEL(IUNITS)                             ABF07180
```

```
      GO TO 500

C     Output particle trajectories:
  400 IF(IBURST.NE.3) WRITE(6,40)
      IF(IBURST.EQ.3) WRITE(6,41)
40    FORMAT(/' Sorry.'/
     ,' Particle trajectories are not part of the current data base,'/
     ,' in spite of their being offered on the menu of options.  They'/
     ,' can be had by special arrangement, however.'/)
41    FORMAT(/' Sorry.'/
     ,' Particle trajectories are not part of the current data base,'/
     ,' in spite of their being offered on the menu of options.  They'/
     ,' can be had later by special arrangement, in the MR region at'/
     ,' least (flow field in the RR region is not 1-dimensional).'/)
      GO TO 9000

C     Conclude output:
  500 IF(IBURST.EQ.3) WRITE(6,50)
50    FORMAT(' and radius is ground radius, measured from ground zero)')
      IF(IFILE.NE.6) WRITE(6,51)
51    FORMAT(/' Data output to disk file')

C     Check for interpolation:
      CALL INTERP(X,Y,NR,UNITS,1)

 9000 INDEX=0
      RETURN
 9001 REWIND 5
      INDEX=0
      RETURN
 9002 REWIND 5
      INDEX=1
      RETURN
      END


*.FF2
************************************************************************
* Subprogram to output peak profile data (free field).

      SUBROUTINE FF2(INDEX,R,P,D,U,N,X,Y)
      REAL*4 R(N),P(N),D(N),U(N),X(N),Y(N)
      COMMON/BLOCK1/ IUNITS,IRATIO,IFILE,ITYPE,IATMOS,ISCALE,IBURST,IB2,
     ,IDATA,ITIME,IFORM,IPEAK
      COMMON/BLOCK2/ WS,TS,PS,CS,DS,CMASS,EMASS,TO,PO,RH,DO,CO,S,ST,HOB
      CHARACTER*1 UTEMP(2)/'C','F'/
      CHARACTER*2 UDIST(2)/' m','ft'/
      CHARACTER*3 UPRESS(2)/'kPa','psi'/
      CHARACTER*5 UVEL(2)/'m/ms ','ft/ms'/
      CHARACTER*6 UDEN(2)/' kg/m3','lb/ft3'/,UNITS(2)
      CALL SCREEN('         peak profile')

C     Fetch and adjust data:
      DO 090 I=1,N
      X(I)=R(I)*S
      CALL EXPAND(P(I),D(I),U(I),PVM,SSR,HOP,DEN,DYP,TOP,END,TEMP,PVML,
     ,ENT,WORK,DUM,1)
      IF(IDATA.EQ.5) Y(I)=HOP
      IF(IDATA.EQ.6) Y(I)=DYP
      IF(IDATA.EQ.7) Y(I)=TOP
```

```
      IF(IDATA.EQ.10) Y(I)=PVM                                          ABF07790
      IF(IDATA.EQ.11) Y(I)=DEN                                          ABF07800
      IF(IDATA.EQ.12) Y(I)=TEMP                                         ABF07810
      IF(IDATA.EQ.13) Y(I)=SSR                                          ABF07820
      IF(IDATA.EQ.14) Y(I)=ENT                                          ABF07830
      IF(IDATA.EQ.15) Y(I)=END                                          ABF07840
      IF(IDATA.EQ.16) Y(I)=WORK                                         ABF07850
      IF(IRATIO.EQ.2.AND.IDATA.EQ.5) Y(I)=Y(I)*PO                       ABF07860
      IF(IRATIO.EQ.2.AND.IDATA.EQ.6) Y(I)=Y(I)*PO                       ABF07870
      IF(IRATIO.EQ.2.AND.IDATA.EQ.7) Y(I)=Y(I)*PO                       ABF07880
      IF(IRATIO.EQ.2.AND.IDATA.EQ.10) Y(I)=Y(I)*CO                      ABF07890
      IF(IRATIO.EQ.2.AND.IDATA.EQ.11) Y(I)=Y(I)*DO                      ABF07900
      IF(IRATIO.EQ.2.AND.IDATA.EQ.12) Y(I)=Y(I)*(273.15+TO)-273.15      ABF07910
      IF(IRATIO.EQ.2.AND.IDATA.EQ.13) Y(I)=Y(I)*CO                      ABF07920
  090 CONTINUE                                                          ABF07930
                                                                        ABF07940
C     Branch to output:                                                 ABF07950
      IF(IDATA.EQ.5) GO TO 500                                          ABF07960
      IF(IDATA.EQ.6) GO TO 600                                          ABF07970
      IF(IDATA.EQ.7) GO TO 700                                          ABF07980
      IF(IDATA.EQ.10) GO TO 1000                                        ABF07990
      IF(IDATA.EQ.11) GO TO 1100                                        ABF08000
      IF(IDATA.EQ.12) GO TO 1200                                        ABF08010
      IF(IDATA.EQ.13) GO TO 1300                                        ABF08020
      IF(IDATA.EQ.14) GO TO 1400                                        ABF08030
      IF(IDATA.EQ.15) GO TO 1500                                        ABF08040
      IF(IDATA.EQ.16) GO TO 1600                                        ABF08050
      PRINT*,' ERROR(FF2): INVALID CALL'                                ABF08060
      GO TO 9000                                                        ABF08070
                                                                        ABF08080
C     Output peak hydrostatic overpressures:                            ABF08090
  500 IF(IRATIO EQ.1) WRITE(IFILE,1) UDIST(IUNITS)                      ABF08100
      IF(IRATIO.EQ.2) WRITE(IFILE,2) UDIST(IUNITS),UPRESS(IUNITS)       ABF08110
      WRITE(IFILE,3) (X(I),Y(I),I=1,N)                                  ABF08120
1     FORMAT(' Peak hydrostatic overpressure:'/                         ABF08130
     ,'    R(',A2,')  P(ratio)')                                        ABF08140
2     FORMAT(' Peak hydrostatic overpressure:'/                         ABF08150
     ,'    R(',A2,')    P(',A3,')')                                     ABF08160
3     FORMAT(F9.3,F10.3)                                                ABF08170
      UNITS(2)=' '                                                      ABF08180
      IF(IRATIO.EQ.2) UNITS(2)=UPRESS(IUNITS)                           ABF08190
      GO TO 5000                                                        ABF08200
                                                                        ABF08210
C     Output peak dynamic pressures:                                    ABF08220
  600 IF(IRATIO.EQ.1) WRITE(IFILE,4) UDIST(IUNITS)                      ABF08230
      IF(IRATIO.EQ.2) WRITE(IFILE,5) UDIST(IUNITS),UPRESS(IUNITS)       ABF08240
      WRITE(IFILE,6) (X(I),Y(I),I=1,N)                                  ABF08250
4     FORMAT(' Peak dynamic pressure:'/                                 ABF08260
     ,'    R(',A2,')  P(ratio)')                                        ABF08270
5     FORMAT(' Peak dynamic pressure:'/                                 ABF08280
     ,'    R(',A2,')    P(',A3,')')                                     ABF08290
6     FORMAT(F9.3,F10.3)                                                ABF08300
      UNITS(2)=' '                                                      ABF08310
      IF(IRATIO.EQ.2) UNITS(2)=UPRESS(IUNITS)                           ABF08320
      GO TO 5000                                                        ABF08330
                                                                        ABF08340
C     Output peak total head overpressures:                             ABF08350
  700 IF(IRATIO.EQ.1) WRITE(IFILE,7) UDIST(IUNITS)                      ABF08360
      IF(IRATIO.EQ.2) WRITE(IFILE,8) UDIST(IUNITS),UPRESS(IUNITS)       ABF08370
      WRITE(IFILE,9) (X(I),Y(I),I=1,N)                                  ABF08380
```

```
 7    FORMAT(' Peak total head overpressure:'/
     ,'     R(',A2,')  P(ratio)')
 8    FORMAT(' Peak total head overpressure:'/
     ,'     R(',A2,')     P(',A3,')')
 9    FORMAT(F9.3,F10.3)
      UNITS(2)=' '
      IF(IRATIO.EQ.2) UNITS(2)=UPRESS(IUNITS)
      GO TO 5000

C     Output peak particle velocites:
 1000 IF(IRATIO.EQ.1) WRITE(IFILE,10) UDIST(IUNITS)
      IF(IRATIO.EQ.2) WRITE(IFILE,11) UDIST(IUNITS),UVEL(IUNITS)
      WRITE(IFILE,12) (X(I),Y(I),I=1,N)
 10   FORMAT(' Peak particle velocity:'/
     ,'     R(',A2,')  V(Mach#)')
 11   FORMAT(' Peak particle velocity:'/
     ,'     R(',A2,')  V(',A5,')')
 12   FORMAT(F9.3,F10.3)
      UNITS(2)=' '
      IF(IRATIO.EQ.2) UNITS(2)=UVEL(IUNITS)
      GO TO 5000

C     Output peak gas densities:
 1100 IF(IRATIO.EQ.1) WRITE(IFILE,13) UDIST(IUNITS)
      IF(IRATIO.EQ.2) WRITE(IFILE,14) UDIST(IUNITS),UDEN(IUNITS)
      WRITE(IFILE,15) (X(I),Y(I),I=1,N)
 13   FORMAT(' Peak gas density:'/
     ,'     R(',A2,')  D(ratio)')
 14   FORMAT(' Peak gas density:'/
     ,'     R(',A2,') D(',A6,')')
 15   FORMAT(F9.3,F10.3)
      UNITS(2)=' '
      IF(IRATIO.EQ.2) UNITS(2)=UDEN(IUNITS)
      GO TO 5000

C     Output peak temperatures:
 1200 IF(IRATIO.EQ.1) WRITE(IFILE,16) UDIST(IUNITS)
      IF(IRATIO.EQ.2) WRITE(IFILE,17) UDIST(IUNITS),UTEMP(IUNITS)
      WRITE(IFILE,18) (X(I),Y(I),I=1,N)
 16   FORMAT(' Peak temperature:'/
     ,'     R(',A2,')  T(ratio)')
 17   FORMAT(' Peak temperature:'/
     ,'     R(',A2,')      T(',A1,')')
 18   FORMAT(F9.3,F10.3)
      UNITS(2)=' '
      IF(IRATIO.EQ.2) UNITS(2)=UTEMP(IUNITS)
      GO TO 5000

C     Output peak sound speeds:
 1300 IF(IRATIO.EQ.1) WRITE(IFILE,19) UDIST(IUNITS)
      IF(IRATIO.EQ.2) WRITE(IFILE,20) UDIST(IUNITS),UVEL(IUNITS)
      WRITE(IFILE,21) (X(I),Y(I),I=1,N)
 19   FORMAT(' Peak sound speed:'/
     ,'     R(',A2,')  A(ratio)')
 20   FORMAT(' Peak sound speed:'/
     ,'     R(',A2,')  A(',A5,')')
 21   FORMAT(F9.3,F10.3)
      UNITS(2)=' '
      IF(IRATIO.EQ.2) UNITS(2)=UVEL(IUNITS)
      GO TO 5000
```

```

C     Output entropy changes:
 1400 WRITE(IFILE,22) UDIST(IUNITS),(X(I),Y(I),I=1,N)
22    FORMAT(' Entropy change:'/
     ,'     R(',A2,') DS(J/K/g)'/
     ,(F9.3,F10.6))
      UNITS(2)='J/K/g'
      GO TO 5000

C     Output peak energy densities:
 1500 WRITE(IFILE,23) UDIST(IUNITS),(X(I),Y(I),I=1,N)
23    FORMAT(' Peak energy density:'/
     ,'     R(',A2,')    E(J/cc)'/
     ,(F9.3,F10.3))
      UNITS(2)='J/cc'
      GO TO 5000

C     Output peak available work:
 1600 WRITE(IFILE,24) UDIST(IUNITS),(X(I),Y(I),I=1,N)
24    FORMAT(' Peak available work:'/
     ,'     R(',A2,')    W(J/cc)'/
     ,(F9.3,F10.6))
      UNITS(2)='J/cc'
      GO TO 5000

C     Signal end of output and check for interpolation:
 5000 IF(IFILE.NE.6) WRITE(6,50)
50    FORMAT(/' Data output to disk file')
      UNITS(1)=UDIST(IUNITS)

C     Check for interpolation:
      CALL INTERP(X,Y,N,UNITS,1)

 9000 INDEX=0
      RETURN
 9001 REWIND 5
      INDEX=0
      RETURN
      END


*.FF3
************************************************************************
* Subprogram to interpolate profile data (free field).

      SUBROUTINE FF3(INDEX,TIMES,NTIMES,R,T1,P1,D1,U1,T2,P2,D2,U2,T,J1,
     ,NJ,P,D,U,NR,NT,NRM,NTM,NDM,X1,X2,Y1,Y2,X,YI,Y)
      REAL*4 TIMES(NTIMES),X1(NDM),X2(NDM),Y1(3,NDM),Y2(3,NDM),X(NDM),
     ,YI(3,NDM),Y(NDM)
      REAL*4 R(NRM),T1(NRM),P1(NRM),D1(NRM),U1(NRM),T2(NRM),P2(NRM),
     ,D2(NRM),U2(NRM),T(NTM),P(NRM,NDM),D(NRM,NDM),U(NRM,NDM)
      INTEGER*4 NJ(NRM),J1(NRM)
      COMMON/BLOCK1/ IUNITS,IRATIO,IFILE,ITYPE,IATMOS,ISCALE,IBURST,IB2,
     ,IDATA,ITIME,IFORM,IPEAK
      COMMON/BLOCK2/ WS,TS,PS,CS,DS,CMASS,EMASS,TO,PO,RH,DO,CO,S,ST,HOB
      CHARACTER*6 UNITS(2)/'(dist)','(data)'/

C     Enter main loop:
      DO 8000 N=1,NTIMES
      CALL SCREEN('           wave profile')
```

```
      TI=TIMES(N)/ST
      IF(TI.LT.T(1).OR.TI.GT.T(NT)) WRITE(6,1) TI
      IF(TI.LT.T(1).OR.TI.GT.T(NT)) GO TO 8000
    1 FORMAT(/' Time',F9.3,' is beyond data base range (no profile)')

C     Locate neighbouring row numbers:
      DO 101 J=1,NT
  101 IF(T(J).GE.TI) GO TO 102
      J=NT
  102 IF(J.EQ.1) J=2
      JU=J
      JL=J-1

C     Interpolate disturbance data on neighbouring rows:
      CALL LINT(T(JL),X2(1),T2,R,NR)
      CALL LINT(T(JL),Y2(1,1),T2,P2,NR)
      CALL LINT(T(JL),Y2(2,1),T2,D2,NR)
      CALL LINT(T(JL),Y2(3,1),T2,U2,NR)
      CALL LINT(T(JU),X1(1),T2,R,NR)
      CALL LINT(T(JU),Y1(1,1),T2,P2,NR)
      CALL LINT(T(JU),Y1(2,1),T2,D2,NR)
      CALL LINT(T(JU),Y1(3,1),T2,U2,NR)
      L2=1
      L1=1

C     Collect field data on neighbouring rows:
      DO 104 K=1,NR
      IF(NJ(K).EQ.0) GO TO 104
      JM=J1(K)+NJ(K)-1
      IF(JL.LT.J1(K).OR.JL.GT.JM) GO TO 103
      JR=JL-J1(K)+1
      L2=L2+1
      X2(L2)=R(K)
      Y2(1,L2)=P(K,JR)
      Y2(2,L2)=D(K,JR)
      Y2(3,L2)=U(K,JR)
  103 IF(JU.LT.J1(K).OR.JU.GT.JM) GO TO 104
      JR=JU-J1(K)+1
      L1=L1+1
      X1(L1)=R(K)
      Y1(1,L1)=P(K,JR)
      Y1(2,L1)=D(K,JR)
      Y1(3,L1)=U(K,JR)
  104 CONTINUE
      L2=L2+1
      L1=L1+1

C     Interpolate front data on neighbouring rows:
      CALL LINT(T(JL),X2(L2),T1,R,NR)
      CALL LINT(T(JL),Y2(1,L2),T1,P1,NR)
      CALL LINT(T(JL),Y2(2,L2),T1,D1,NR)
      CALL LINT(T(JL),Y2(3,L2),T1,U1,NR)
      CALL LINT(T(JU),X1(L1),T1,R,NR)
      CALL LINT(T(JU),Y1(1,L1),T1,P1,NR)
      CALL LINT(T(JU),Y1(2,L1),T1,D1,NR)
      CALL LINT(T(JU),Y1(3,L1),T1,U1,NR)

C     Interpolate front position at time input:
      CALL LINT(TI,RFRONT,T1,R,NR)
      X(1)=RFRONT
```

```
                                                                        ABF10190
C     Interpolate front data at time input:                             ABF10200
      FRAC=(RFRONT-X2(L2))/(X1(L1)-X2(L2))                              ABF10210
      YI(1,1)=Y2(1,L2)+FRAC*(Y1(1,L1)-Y2(1,L2))                         ABF10220
      YI(2,1)=Y2(2,L2)+FRAC*(Y1(2,L1)-Y2(2,L2))                         ABF10230
      YI(3,1)=Y2(3,L2)+FRAC*(Y1(3,L1)-Y2(3,L2))                         ABF10240
                                                                        ABF10250
C     Interpolate profile data:                                         ABF10260
      IF(L1.EQ.1) GO TO 109                                             ABF10270
      JJ=2                                                              ABF10280
      L=1                                                               ABF10290
      IF(L1.EQ.2) GO TO 108                                             ABF10300
      DO 107 II=2,L1-1                                                  ABF10310
      I=L1-II+1                                                         ABF10320
      J=L2-JJ+1                                                         ABF10330
      IF(X1(I).GT.X2(J)) GO TO 107                                      ABF10340
      IF(X1(I).GT.RFRONT) GO TO 107                                     ABF10350
      L=L+1                                                             ABF10360
      X(L)=X2(J)                                                        ABF10370
      YI(1,L)=Y2(1,J)+FRAC*(Y1(1,I)-Y2(1,J))                            ABF10380
      YI(2,L)=Y2(2,J)+FRAC*(Y1(2,I)-Y2(2,J))                            ABF10390
      YI(3,L)=Y2(3,J)+FRAC*(Y1(3,I)-Y2(3,J))                            ABF10400
      IF(JJ.LT.L2) JJ=JJ+1                                              ABF10410
  107 CONTINUE                                                          ABF10420
                                                                        ABF10430
C     Finalize the interpolation:                                       ABF10440
  108 LENGTH=L+1                                                        ABF10450
      CALL LINT(T1,X(LENGTH),T2,R,NR)                                   ABF10460
      FRAC=(X(LENGTH)-X2(1))/(X1(1)-X2(1))                              ABF10470
      YI(1,LENGTH)=Y2(1,1)+FRAC*(Y1(1,1)-Y2(1,1))                       ABF10480
      YI(2,LENGTH)=Y2(2,1)+FRAC*(Y1(2,1)-Y2(2,1))                       ABF10490
      YI(3,LENGTH)=Y2(3,1)+FRAC*(Y1(3,1)-Y2(3,1))                       ABF10500
      GO TO 200                                                         ABF10510
  109 LENGTH=1                                                          ABF10520
                                                                        ABF10530
C     Output interpolated data:                                         ABF10540
  200 CALL FF3OUT(TIMES(N),X,YI,Y,LENGTH,NDM,UNITS)                     ABF10550
      IF(IFILE.NE.6) WRITE(6,2)                                         ABF10560
2     FORMAT(/' Data output to disk file')                              ABF10570
                                                                        ABF10580
C     Check for interpolation:                                          ABF10590
      IF(LENGTH.GT.1) CALL INTERP(X,Y,LENGTH,UNITS,1)                   ABF10600
                                                                        ABF10610
C     Check for integration:                                            ABF10620
      IF(LENGTH.GT.1) CALL INTEGR(X,Y,LENGTH,UNITS)                     ABF10630
                                                                        ABF10640
 8000 CONTINUE                                                          ABF10650
      INDEX=0                                                           ABF10660
      RETURN                                                            ABF10670
      END                                                               ABF10680
                                                                        ABF10690
                                                                        ABF10700
*.FF3OUT                                                                ABF10710
************************************************************************ABF10720
* Subprogram to output wave profile data (free field).                  ABF10730
                                                                        ABF10740
      SUBROUTINE FF3OUT(T,X,YI,Y,LENGTH,NDM,UNITS)                      ABF10750
      REAL*4 X(NDM),YI(3,NDM),Y(NDM)                                    ABF10760
      COMMON/BLOCK1/ IUNITS,IRATIO,IFILE,ITYPE,IATMOS,ISCALE,IBURST,IB2,ABF10770
     ,IDATA,ITIME,IFORM,IPEAK                                           ABF10780
```

```
      COMMON/BLOCK2/ WS,TS,PS,CS,DS,CMASS,EMASS,TO,PO,RH,DO,CO,S,ST,HOB ABF10790
      CHARACTER*1 UTEMP(2)/'C','F'/                                     ABF10800
      CHARACTER*2 UDIST(2)/' m','ft'/                                   ABF10810
      CHARACTER*3 UPRESS(2)/'kPa','psi'/                                ABF10820
      CHARACTER*5 UVEL(2)/'m/ms ','ft/ms'/                              ABF10830
      CHARACTER*6 UDEN(2)/' kg/m3','lb/ft3'/,UNITS(2)                   ABF10840
      UNITS(1)=UDIST(IUNITS)                                            ABF10850
                                                                        ABF10860
C     Expand:                                                           ABF10870
      DO 101 I=1,LENGTH                                                 ABF10880
      CALL EXPAND(YI(1,I),YI(2,I),YI(3,I),PVM,SSR,HOP,DEN,DYP,TOP,END,  ABF10890
     ,TEMP,PVML,ENT,WORK,DUM,I)                                         ABF10900
                                                                        ABF10910
C     Finalize:                                                         ABF10920
      X(I)=X(I)*ST                                                      ABF10930
      IF(IDATA.EQ.5) Y(I)=HOP                                           ABF10940
      IF(IDATA.EQ.6) Y(I)=DYP                                           ABF10950
      IF(IDATA.EQ.7) Y(I)=TOP                                           ABF10960
      IF(IDATA.EQ.10) Y(I)=PVM                                          ABF10970
      IF(IDATA.EQ.11) Y(I)=DEN                                          ABF10980
      IF(IDATA.EQ.12) Y(I)=TEMP                                         ABF10990
      IF(IDATA.EQ.13) Y(I)=SSR                                          ABF11000
      IF(IDATA.EQ.14) Y(I)=ENT                                          ABF11010
      IF(IDATA.EQ.15) Y(I)=END                                          ABF11020
      IF(IDATA.EQ.16) Y(I)=WORK                                         ABF11030
      IF(IRATIO.EQ.2.AND.IDATA.EQ.5) Y(I)=Y(I)*PO                       ABF11040
      IF(IRATIO.EQ.2.AND.IDATA.EQ.6) Y(I)=Y(I)*PO                       ABF11050
      IF(IRATIO.EQ.2.AND.IDATA.EQ.7) Y(I)=Y(I)*PO                       ABF11060
      IF(IRATIO.EQ.2.AND.IDATA.EQ.10) Y(I)=Y(I)*CO                      ABF11070
      IF(IRATIO.EQ.2.AND.IDATA.EQ.11) Y(I)=Y(I)*DO                      ABF11080
      IF(IRATIO.EQ.2.AND.IDATA.EQ.12) Y(I)=Y(I)*(273.15+TO)-273.15      ABF11090
      IF(IRATIO.EQ.2.AND.IDATA.EQ.13) Y(I)=Y(I)*CO                      ABF11100
  101 CONTINUE                                                          ABF11110
                                                                        ABF11120
C     Branch to output:                                                 ABF11130
      IF(IDATA.EQ.5) GO TO 500                                          ABF11140
      IF(IDATA.EQ.6) GO TO 600                                          ABF11150
      IF(IDATA.EQ.7) GO TO 700                                          ABF11160
      IF(IDATA.EQ.10) GO TO 1000                                        ABF11170
      IF(IDATA.EQ.11) GO TO 1100                                        ABF11180
      IF(IDATA.EQ.12) GO TO 1200                                        ABF11190
      IF(IDATA.EQ.13) GO TO 1300                                        ABF11200
      IF(IDATA.EQ.14) GO TO 1400                                        ABF11210
      IF(IDATA.EQ.15) GO TO 1500                                        ABF11220
      IF(IDATA.EQ.16) GO TO 1600                                        ABF11230
      PRINT*,' ERROR(FF3): INVALID CALL'                                ABF11240
      RETURN                                                            ABF11250
                                                                        ABF11260
C     Output hydrostatic overpressure wave profile:                     ABF11270
  500 IF(IRATIO.EQ.1) WRITE(IFILE,1) T,UDIST(IUNITS)                    ABF11280
      IF(IRATIO.EQ.2) WRITE(IFILE,2) T,UDIST(IUNITS),UPRESS(IUNITS)     ABF11290
      WRITE(IFILE,3) (X(I),Y(I),I=1,LENGTH)                             ABF11300
1     FORMAT(' Hydrostatic overpressure at time=',F9.3,' ms'/           ABF11310
     ,' Shock is at first position listed'/                             ABF11320
     ,'    R(',A2,')  P(ratio)')                                        ABF11330
2     FORMAT(' Hydrostatic overpressure at time=',F9.3,' ms'/           ABF11340
     ,' Shock is at first position listed'/                             ABF11350
     ,'    R(',A2,')    P(',A3,')')                                     ABF11360
3     FORMAT(F9.3,F10.3)                                                ABF11370
      UNITS(2)=' '                                                      ABF11380
```

```
      IF(IRATIO.EQ.2) UNITS(2)=UPRESS(IUNITS)                           ABF11390
      RETURN                                                            ABF11400
                                                                        ABF11410
C     Output dynamic pressure wave profile:                             ABF11420
  600 IF(IRATIO.EQ.1) WRITE(IFILE,4) T,UDIST(IUNITS)                    ABF11430
      IF(IRATIO.EQ.2) WRITE(IFILE,5) T,UDIST(IUNITS),UPRESS(IUNITS)     ABF11440
      WRITE(IFILE,6) (X(I),Y(I),I=1,LENGTH)                             ABF11450
4     FORMAT(' Dynamic pressure at time=',F9.3,' ms'/                   ABF11460
     ,' Shock is at first position listed'/                             ABF11470
     ,'    R(',A2,')  P(ratio)')                                        ABF11480
5     FORMAT(' Dynamic pressure at time=',F9.3,' ms'/                   ABF11490
     ,' Shock is at first position listed'/                             ABF11500
     ,'    R(',A2,')    P(',A3,')')                                     ABF11510
6     FORMAT(F9.3,F10.3)                                                ABF11520
      UNITS(2)=' '                                                      ABF11530
      IF(IRATIO.EQ.2) UNITS(2)=UPRESS(IUNITS)                           ABF11540
      RETURN                                                            ABF11550
                                                                        ABF11560
C     Output total head overpressure wave profile:                      ABF11570
  700 IF(IRATIO.EQ.1) WRITE(IFILE,7) T,UDIST(IUNITS)                    ABF11580
      IF(IRATIO.EQ.2) WRITE(IFILE,8) T,UDIST(IUNITS),UPRESS(IUNITS)     ABF11590
      WRITE(IFILE,9) (X(I),Y(I),I=1,LENGTH)                             ABF11600
7     FORMAT(' Total head overpressure at time=',F9.3,' ms'/            ABF11610
     ,' Shock is at first position listed'/                             ABF11620
     ,'    R(',A2,')  P(ratio)')                                        ABF11630
8     FORMAT(' Total head overpressure at time=',F9.3,' ms'/            ABF11640
     ,' Shock is at first position listed'/                             ABF11650
     ,'    R(',A2,')    P(',A3,')')                                     ABF11660
9     FORMAT(F9.3,F10.3)                                                ABF11670
      UNITS(2)=' '                                                      ABF11680
      IF(IRATIO.EQ.2) UNITS(2)=UPRESS(IUNITS)                           ABF11690
      RETURN                                                            ABF11700
                                                                        ABF11710
C     Output particle velocity wave profile:                            ABF11720
 1000 IF(IRATIO.EQ.1) WRITE(IFILE,10) T,UDIST(IUNITS)                   ABF11730
      IF(IRATIO.EQ.2) WRITE(IFILE,11) T,UDIST(IUNITS),UVEL(IUNITS)      ABF11740
      WRITE(IFILE,12) (X(I),Y(I),I=1,LENGTH)                            ABF11750
10    FORMAT(' Particle velocity at time=',F9.3,' ms'/                  ABF11760
     ,' Shock is at first position listed'/                             ABF11770
     ,'    R(',A2,')  V(Mach#)')                                        ABF11780
11    FORMAT(' Particle velocity at time=',F9.3,' ms'/                  ABF11790
     ,' Shock is at first position listed'/                             ABF11800
     ,'    R(',A2,')  V(',A5,')')                                       ABF11810
12    FORMAT(F9.3,F10.3)                                                ABF11820
      UNITS(2)=' '                                                      ABF11830
      IF(IRATIO.EQ.2) UNITS(2)=UVEL(IUNITS)                             ABF11840
      RETURN                                                            ABF11850
                                                                        ABF11860
C     Output gas density wave profile:                                  ABF11870
 1100 IF(IRATIO.EQ.1) WRITE(IFILE,13) T,UDIST(IUNITS)                   ABF11880
      IF(IRATIO.EQ.2) WRITE(IFILE,14) T,UDIST(IUNITS),UDEN(IUNITS)      ABF11890
      WRITE(IFILE,15) (X(I),Y(I),I=1,LENGTH)                            ABF11900
13    FORMAT(' Density at time=',F9.3,' ms'/                            ABF11910
     ,' Shock is at first position listed'/                             ABF11920
     ,'    R(',A2,')  D(ratio)')                                        ABF11930
14    FORMAT(' Density at time=',F9.3,' ms'/                            ABF11940
     ,' Shock is at first position listed'/                             ABF11950
     ,'    R(',A2,') D(',A6,')')                                        ABF11960
15    FORMAT(F9.3,F10.3)                                                ABF11970
      UNITS(2)=' '                                                      ABF11980
```

```
      IF(IRATIO.EQ.2) UNITS(2)=UDEN(IUNITS)                             ABF11990
      RETURN                                                            ABF12000
                                                                        ABF12010
C     Output temperature wave profile:                                  ABF12020
 1200 IF(IRATIO.EQ.1) WRITE(IFILE,16) T,UDIST(IUNITS)                   ABF12030
      IF(IRATIO.EQ.2) WRITE(IFILE,17) T,UDIST(IUNITS),UTEMP(IUNITS)     ABF12040
      WRITE(IFILE,18) (X(I),Y(I),I=1,LENGTH)                            ABF12050
16    FORMAT(' Temperature at time=',F9.3,' ms'/                        ABF12060
     ,' Shock is at first position listed'/                             ABF12070
     ,'    R(',A2,')  T(ratio)')                                        ABF12080
17    FORMAT(' Temperature at time=',F9.3,' ms'/                        ABF12090
     ,' Shock is at first position listed'/                             ABF12100
     ,'    R(',A2,')      T(',A1,')')                                   ABF12110
18    FORMAT(F9.3,F10.3)                                                ABF12120
      UNITS(2)=' '                                                      ABF12130
      IF(IRATIO.EQ.2) UNITS(2)=UTEMP(IUNITS)                            ABF12140
      RETURN                                                            ABF12150
                                                                        ABF12160
C     Output sound speed wave profile:                                  ABF12170
 1300 IF(IRATIO.EQ.1) WRITE(IFILE,19) T,UDIST(IUNITS)                   ABF12180
      IF(IRATIO.EQ.2) WRITE(IFILE,20) T,UDIST(IUNITS),UVEL(IUNITS)      ABF12190
      WRITE(IFILE,21) (X(I),Y(I),I=1,LENGTH)                            ABF12200
19    FORMAT(' Sound speed at time=',F9.3,' ms'/                        ABF12210
     ,' Shock is at first position listed'/                             ABF12220
     ,'    R(',A2,')  A(ratio)')                                        ABF12230
20    FORMAT(' Sound speed at time=',F9.3,' ms'/                        ABF12240
     ,' Shock is at first position listed'/                             ABF12250
     ,'    R(',A2,')  A(',A5,')')                                       ABF12260
21    FORMAT(F9.3,F10.3)                                                ABF12270
      UNITS(2)=' '                                                      ABF12280
      IF(IRATIO.EQ.2) UNITS(2)=UVEL(IUNITS)                             ABF12290
      RETURN                                                            ABF12300
                                                                        ABF12310
C     Output entropy change wave profile:                               ABF12320
 1400 WRITE(IFILE,22) T,UDIST(IUNITS),(X(I),Y(I),I=1,LENGTH)            ABF12330
22    FORMAT(' Entropy change at time=',F9.3,' ms'/                     ABF12340
     ,' Shock is at first position listed'/                             ABF12350
     ,'    R(',A2,') DS(J/K/g)'/                                        ABF12360
     ,(F9.3,F10.6))                                                     ABF12370
      UNITS(2)='J/K/g'                                                  ABF12380
      RETURN                                                            ABF12390
                                                                        ABF12400
C     Output energy density wave profile:                               ABF12410
 1500 WRITE(IFILE,23) T,UDIST(IUNITS),(X(I),Y(I),I=1,LENGTH)            ABF12420
23    FORMAT(' Energy density at time=',F9.3,' ms'/                     ABF12430
     ,' Shock is at first position listed'/                             ABF12440
     ,'    R(',A2,')    E(J/cc)'/                                       ABF12450
     ,(F9.3,F10.3))                                                     ABF12460
      UNITS(2)='J/cc'                                                   ABF12470
      RETURN                                                            ABF12480
                                                                        ABF12490
C     Output available work wave profile:                               ABF12500
 1600 WRITE(IFILE,24) T,UDIST(IUNITS),(X(I),Y(I),I=1,LENGTH)            ABF12510
24    FORMAT(' Available work at time=',F9.3,' ms'/                     ABF12520
     ,' Shock is at first position listed'/                             ABF12530
     ,'    R(',A2,')    W(J/cc)'/                                       ABF12540
     ,(F9.3,F10.6))                                                     ABF12550
      UNITS(2)='J/cc'                                                   ABF12560
      RETURN                                                            ABF12570
      END                                                               ABF12580
```

```
                                                                        ABF12590
                                                                        ABF12600
*.FF4                                                                   ABF12610
************************************************************************ABF12620
* Subprogram to interpolate time history data (free field).             ABF12630
                                                                        ABF12640
      SUBROUTINE FF4(INDEX,RADII,NRADII,R,T1,P1,D1,U1,T2,P2,D2,U2,T,J1, ABF12650
     ,NJ,P,D,U,NR,NT,NRM,NTM,NDM,X1,X2,Y1,Y2,X,YI,Y)                    ABF12660
      REAL*4 RADII(NRADII),X1(NDM),X2(NDM),Y1(3,NDM),Y2(3,NDM),X(NDM),  ABF12670
     ,YI(3,NDM),Y(NDM)                                                  ABF12680
      REAL*4 R(NRM),T1(NRM),P1(NRM),D1(NRM),U1(NRM),T2(NRM),P2(NRM),    ABF12690
     ,D2(NRM),U2(NRM),T(NTM),P(NRM,NDM),D(NRM,NDM),U(NRM,NDM)           ABF12700
      INTEGER*4 NJ(NRM),J1(NRM)                                         ABF12710
      COMMON/BLOCK1/ IUNITS,IRATIO,IFILE,ITYPE,IATMOS,ISCALE,IBURST,IB2,ABF12720
     ,IDATA,ITIME,IFORM,IPEAK                                           ABF12730
      COMMON/BLOCK2/ WS,TS,PS,CS,DS,CMASS,EMASS,TO,PO,RH,DO,CO,S,ST,HOB ABF12740
      CHARACTER*6 UNITS(2)/'ms','du'/                                   ABF12750
                                                                        ABF12760
C     Enter main loop:                                                  ABF12770
      DO 8000 N=1,NRADII                                                ABF12780
      IF(INDEX.EQ.0) CALL SCREEN('          time history')              ABF12790
      RI=RADII(N)/S                                                     ABF12800
      IF(RI.LT.R(1).OR.RI.GT.R(NR)) WRITE(6,1)                          ABF12810
1     FORMAT(/' WARNING: The following data are EXTRAPOLATED, that is,'/ABF12820
     ,10X,'for a position not within the data base range'/)             ABF12830
                                                                        ABF12840
C     Locate neighbouring column numbers:                               ABF12850
      DO 101 K=1,NR                                                     ABF12860
  101 IF(R(K).GE.RI) GO TO 102                                          ABF12870
      K=NR                                                              ABF12880
  102 IF(K.EQ.1) K=2                                                    ABF12890
      K2=K                                                              ABF12900
      K1=K-1                                                            ABF12910
                                                                        ABF12920
C     Collect neighbouring time-of-arrival data:                        ABF12930
      X1(1)=T1(K1)                                                      ABF12940
      Y1(1,1)=P1(K1)                                                    ABF12950
      Y1(2,1)=D1(K1)                                                    ABF12960
      Y1(3,1)=U1(K1)                                                    ABF12970
      X2(1)=T1(K2)                                                      ABF12980
      Y2(1,1)=P1(K2)                                                    ABF12990
      Y2(2,1)=D1(K2)                                                    ABF13000
      Y2(3,1)=U1(K2)                                                    ABF13010
                                                                        ABF13020
C     Collect neighbouring field data:                                  ABF13030
      IF(NJ(K1).EQ.0) GO TO 104                                         ABF13040
      DO 103 J=1,NJ(K1)                                                 ABF13050
      L=J1(K1)+J-1                                                      ABF13060
      X1(J+1)=T(L)                                                      ABF13070
      Y1(1,J+1)=P(K1,J)                                                 ABF13080
      Y1(2,J+1)=D(K1,J)                                                 ABF13090
      Y1(3,J+1)=U(K1,J)                                                 ABF13100
  103 CONTINUE                                                          ABF13110
  104 IF(NJ(K2).EQ.0) GO TO 106                                         ABF13120
      DO 105 J=1,NJ(K2)                                                 ABF13130
      L=J1(K2)+J-1                                                      ABF13140
      X2(J+1)=T(L)                                                      ABF13150
      Y2(1,J+1)=P(K2,J)                                                 ABF13160
      Y2(2,J+1)=D(K2,J)                                                 ABF13170
      Y2(3,J+1)=U(K2,J)                                                 ABF13180
```

```
  105 CONTINUE                                                          ABF13190
  106 L1=NJ(K1)+2                                                       ABF13200
      X1(L1)=T2(K1)                                                     ABF13210
      Y1(1,L1)=P2(K1)                                                   ABF13220
      Y1(2,L1)=D2(K1)                                                   ABF13230
      Y1(3,L1)=U2(K1)                                                   ABF13240
      IF(K1.EQ.1.AND.L1.EQ.2) L1=1                                      ABF13250
      L2=NJ(K2)+2                                                       ABF13260
      X2(L2)=T2(K2)                                                     ABF13270
      Y2(1,L2)=P2(K2)                                                   ABF13280
      Y2(2,L2)=D2(K2)                                                   ABF13290
      Y2(3,L2)=U2(K2)                                                   ABF13300
                                                                        ABF13310
C     Interpolate time of arrival:                                      ABF13320
      CALL LINT(RI,TOA,R,T1,NR)                                         ABF13330
      IF(TOA.LT.T1(K1).AND.T1(K1)-TOA.GT.T1(K2)-T1(K1)) WRITE(6,2)      ABF13340
      IF(TOA.GT.T1(K2).AND.TOA-T1(K2).GT.T1(K2)-T1(K1)) WRITE(6,2)      ABF13350
2     FORMAT(' WARNING: Extrapolation is considered excessive'/)        ABF13360
      X(1)=0.                                                           ABF13370
                                                                        ABF13380
C     Interpolate peak data:                                            ABF13390
      FRAC=(TOA-X1(1))/(X2(1)-X1(1))                                    ABF13400
      YI(1,1)=Y1(1,1)+FRAC*(Y2(1,1)-Y1(1,1))                            ABF13410
      YI(2,1)=Y1(2,1)+FRAC*(Y2(2,1)-Y1(2,1))                            ABF13420
      YI(3,1)=Y1(3,1)+FRAC*(Y2(3,1)-Y1(3,1))                            ABF13430
                                                                        ABF13440
C     Interpolate history data:                                         ABF13450
      IF(TOA.LE.0..OR.TOA.GE.X2(L2)) L1=1                               ABF13460
      IF(L1.EQ.1.OR.L2.EQ.1) GO TO 111                                  ABF13470
      J=2                                                               ABF13480
      L=1                                                               ABF13490
  107 IF(X2(J).GT.TOA) GO TO 108                                        ABF13500
      IF(J.EQ.L2) GO TO 111                                             ABF13510
      J=J+1                                                             ABF13520
      GO TO 107                                                         ABF13530
  108 IF(L1.EQ.2) GO TO 110                                             ABF13540
      DO 109 I=2,L1-1                                                   ABF13550
      IF(X1(I).LT.TOA) GO TO 109                                        ABF13560
      IF(X1(I).LT.X2(J)) GO TO 109                                      ABF13570
      IF(X1(I).NE.X2(J)) PRINT*,'WARNING(FF4A): TIMES NOT EQUAL'        ABF13580
      L=L+1                                                             ABF13590
      X(L)=X1(I)-TOA                                                    ABF13600
      YI(1,L)=Y1(1,I)+FRAC*(Y2(1,J)-Y1(1,I))                            ABF13610
      YI(2,L)=Y1(2,I)+FRAC*(Y2(2,J)-Y1(2,I))                            ABF13620
      YI(3,L)=Y1(3,I)+FRAC*(Y2(3,J)-Y1(3,I))                            ABF13630
      J=J+1                                                             ABF13640
      IF(J.EQ.L2.OR.X2(J).GT.X1(L1)) GO TO 110                          ABF13650
  109 CONTINUE                                                          ABF13660
                                                                        ABF13670
C     Finalize the interpolation:                                       ABF13680
  110 LENGTH=L+1                                                        ABF13690
      CALL LINT(RI,TOD,R,T2,NR)                                         ABF13700
      X(LENGTH)=TOD-TOA                                                 ABF13710
      FRAC=(TOD-X1(L1))/(X2(L2)-X1(L1))                                 ABF13720
      YI(1,LENGTH)=Y1(1,L1)+FRAC*(Y2(1,L2)-Y1(1,L1))                    ABF13730
      YI(2,LENGTH)=Y1(2,L1)+FRAC*(Y2(2,L2)-Y1(2,L1))                    ABF13740
      YI(3,LENGTH)=Y1(3,L1)+FRAC*(Y2(3,L2)-Y1(3,L1))                    ABF13750
      GO TO 200                                                         ABF13760
  111 LENGTH=1                                                          ABF13770
      IF(YI(1,1).GT.1.) GO TO 200                                       ABF13780
```

```
      YI(1,1)=1.
      YI(2,1)=1.
      YI(3,1)=0.

C     Output interpolated data:
  200 IF(INDEX.EQ.1) GO TO 9001
      CALL FF4OUT(RADII(N),TOA,X,YI,Y,LENGTH,NDM,UNITS)
      IF(IFILE.NE.6) WRITE(6,3)
3     FORMAT(/' Data output to disk file')

C     Check for interpolation:
      IPEAK=0
      IF(LENGTH.GT.1) CALL INTERP(X,Y,LENGTH,UNITS,1)

C     Check for integration:
      IF(LENGTH.GT.1) CALL IMPULS(X,Y,LENGTH,UNITS)

 8000 CONTINUE
      INDEX=0
      RETURN

 9001 INDEX=LENGTH
      RETURN
      END


*.FF4OUT
************************************************************************
* Subprogram to output time history data (free field).

      SUBROUTINE FF4OUT(RO,TOA,X,YI,Y,LENGTH,NDM,UNITS)
      REAL*4 X(NDM),YI(3,NDM),Y(NDM)
      COMMON/BLOCK1/ IUNITS,IRATIO,IFILE,ITYPE,IATMOS,ISCALE,IBURST,IB2,
     ,IDATA,ITIME,IFORM,IPEAK
      COMMON/BLOCK2/ WS,TS,PS,CS,DS,CMASS,EMASS,TO,PO,RH,DO,CO,S,ST,HOB
      CHARACTER*1 UTEMP(2)/'C','F'/
      CHARACTER*2 UDIST(2)/' m','ft'/
      CHARACTER*3 UPRESS(2)/'kPa','psi'/
      CHARACTER*5 UVEL(2)/'m/ms ','ft/ms'/
      CHARACTER*6 UDEN(2)/' kg/m3','lb/ft3'/,UNITS(2)

C     Expand:
      DO 101 I=1,LENGTH
      CALL EXPAND(YI(1,I),YI(2,I),YI(3,I),PVM,SSR,HOP,DEN,DYP,TOP,END,
     ,TEMP,PVML,ENT,WORK,DUM,I)

C     Finalize:
      X(I)=X(I)*ST
      IF(IDATA.EQ.5) Y(I)=HOP
      IF(IDATA.EQ.6) Y(I)=DYP
      IF(IDATA.EQ.7) Y(I)=TOP
      IF(IDATA.EQ.10) Y(1)=PVM
      IF(IDATA.EQ.11) Y(I)=DEN
      IF(IDATA.EQ.12) Y(I)=TEMP
      IF(IDATA.EQ.13) Y(I)=SSR
      IF(IDATA.EQ.14) Y(I)=ENT
      IF(IDATA.EQ.15) Y(I)=END
      IF(IDATA.EQ.16) Y(I)=WORK
      IF(IRATIO.EQ.2.AND.IDATA.EQ.5) Y(I)=Y(I)*PO
      IF(IRATIO.EQ.2.AND.IDATA.EQ.6) Y(I)=Y(I)*PO
```

```
      IF(IRATIO.EQ.2.AND.IDATA.EQ.7) Y(I)=Y(I)*PO                       ABF14390
      IF(IRATIO.EQ.2.AND.IDATA.EQ.10) Y(I)=Y(I)*CO                      ABF14400
      IF(IRATIO.EQ.2.AND.IDATA.EQ.11) Y(I)=Y(I)*DO                      ABF14410
      IF(IRATIO.EQ.2.AND.IDATA.EQ.12) Y(I)=Y(I)*(273.15+TO)-273.15      ABF14420
      IF(IRATIO.EQ.2.AND.IDATA.EQ.13) Y(I)=Y(I)*CO                      ABF14430
  101 CONTINUE                                                          ABF14440
      TOA=TOA*ST                                                        ABF14450
                                                                        ABF14460
C     Branch to output:                                                 ABF14470
      IF(IDATA.EQ.5) GO TO 500                                          ABF14480
      IF(IDATA.EQ 6) GO TO 600                                          ABF14490
      IF(IDATA.EQ.7) GO TO 700                                          ABF14500
      IF(IDATA.EQ.10) GO TO 1000                                        ABF14510
      IF(IDATA.EQ.11) GO TO 1100                                        ABF14520
      IF(IDATA.EQ.12) GO TO 1200                                        ABF14530
      IF(IDATA.EQ.13) GO TO 1300                                        ABF14540
      IF(IDATA.EQ.14) GO TO 1400                                        ABF14550
      IF(IDATA.EQ.15) GO TO 1500                                        ABF14560
      IF(IDATA.EQ.16) GO TO 1600                                        ABF14570
      PRINT*,' ERROR(FF4): INVALID CALL'                                ABF14580
      RETURN                                                            ABF14590
                                                                        ABF14600
C     Output hydrostatic overpressure time history:                     ABF14610
  500 IF(IRATIO.EQ.1) WRITE(IFILE,1) RO,UDIST(IUNITS),TOA               ABF14620
      IF(IRATIO.EQ.2) WRITE(IFILE,2) RO,UDIST(IUNITS),TOA,UPRESS(IUNITS)ABF14630
      WRITE(IFILE,3) (X(I),Y(I),I=1,LENGTH)                             ABF14640
    1 FORMAT(' Hydrostatic overpressure at distance=',F9.3,1X,A2/       ABF14650
     ,' Time of arrival, TO=',F9.3,' ms'/                               ABF14660
     ,' T-TO(ms)  P(ratio)')                                            ABF14670
    2 FORMAT(' Hydrostatic overpressure at distance=',F9.3,1X,A2/       ABF14680
     ,' Time of arrival, TO=',F9.3,' ms'/                               ABF14690
     ,' T-TO(ms)     P(',A3,')')                                        ABF14700
    3 FORMAT(F9.3,F10.3)                                                ABF14710
      UNITS(2)=' '                                                      ABF14720
      IF(IRATIO.EQ.2) UNITS(2)=UPRESS(IUNITS)                           ABF14730
      RETURN                                                            ABF14740
                                                                        ABF14750
C     Output dynamic pressure time history:                             ABF14760
  600 IF(IRATIO.EQ.1) WRITE(IFILE,4) RO,UDIST(IUNITS),TOA               ABF14770
      IF(IRATIO.EQ.2) WRITE(IFILE,5) RO,UDIST(IUNITS),TOA,UPRESS(IUNITS)ABF14780
      WRITE(IFILE,6) (X(I),Y(I),I=1,LENGTH)                             ABF14790
    4 FORMAT(' Dynamic pressure at distance=',F9.3,1X,A2/               ABF14800
     ,' Time of arrival, TO=',F9.3,' ms'/                               ABF14810
     ,' T-TO(ms)  P(ratio)')                                            ABF14820
    5 FORMAT(' Dynamic pressure at distance=',F9.3,1X,A2/               ABF14830
     ,' Time of arrival, TO=',F9.3,' ms'/                               ABF14840
     ,' T-TO(ms)     P(',A3,')')                                        ABF14850
    6 FORMAT(F9.3,F10.3)                                                ABF14860
      UNITS(2)=' '                                                      ABF14870
      IF(IRATIO.EQ.2) UNITS(2)=UPRESS(IUNITS)                           ABF14880
      RETURN                                                            ABF14890
                                                                        ABF14900
C     Output total head overpressure time history:                      ABF14910
  700 IF(IRATIO.EQ.1) WRITE(IFILE,7) RO,UDIST(IUNITS),TOA               ABF14920
      IF(IRATIO.EQ.2) WRITE(IFILE,8) RO,UDIST(IUNITS),TOA,UPRESS(IUNITS)ABF14930
      WRITE(IFILE,9) (X(I),Y(I),I=1,LENGTH)                             ABF14940
    7 FORMAT(' Total head overpressure at distance=',F9.3,1X,A2/        ABF14950
     ,' Time of arrival, TO=',F9.3,' ms'/                               ABF14960
     ,' T-TO(ms)  P(ratio)')                                            ABF14970
    8 FORMAT(' Total head overpressure at distance=',F9.3,1X,A2/        ABF14980
```

```
     ,' Time of arrival, TO=',F9.3,' ms'/                               ABF14990
     ,' T-TO(ms)    P(',A3,')')                                         ABF15000
9     FORMAT(F9.3,F10.3)                                                ABF15010
      UNITS(2)=' '                                                      ABF15020
      IF(IRATIO.EQ.2) UNITS(2)=UPRESS(IUNITS)                           ABF15030
      RETURN                                                            ABF15040
                                                                        ABF15050
C     Output particle velocity time history:                            ABF15060
 1000 IF(IRATIO.EQ.1) WRITE(IFILE,10) RO,UDIST(IUNITS),TOA              ABF15070
      IF(IRATIO.EQ.2) WRITE(IFILE,11) RO,UDIST(IUNITS),TOA,UVEL(IUNITS) ABF15080
      WRITE(IFILE,12) (X(I),Y(I),I=1,LENGTH)                            ABF15090
10    FORMAT(' Particle velocity at distance=',F9.3,1X,A2/              ABF15100
     ,' Time of arrival, TO=',F9.3,' ms'/                               ABF15110
     ,' T-TO(ms)  V(Mach#)')                                            ABF15120
11    FORMAT(' Particle velocity at distance=',F9.3,1X,A2/              ABF15130
     ,' Time of arrival, TO=',F9.3,' ms'/                               ABF15140
     ,' T-TO(ms)  V(',A5,')')                                           ABF15150
12    FORMAT(F9.3,F10.3)                                                ABF15160
      UNITS(2)=' '                                                      ABF15170
      IF(IRATIO.EQ.2) UNITS(2)=UVEL(IUNITS)                             ABF15180
      RETURN                                                            ABF15190
                                                                        ABF15200
C     Output gas density time history:                                  ABF15210
 1100 IF(IRATIO.EQ.1) WRITE(IFILE,13) RO,UDIST(IUNITS),TOA              ABF15220
      IF(IRATIO.EQ.2) WRITE(IFILE,14) RO,UDIST(IUNITS),TOA,UDEN(IUNITS) ABF15230
      WRITE(IFILE,15) (X(I),Y(I),I=1,LENGTH)                            ABF15240
13    FORMAT(' Density at distance=',F9.3,1X,A2/                        ABF15250
     ,' Time of arrival, TO=',F9.3,' ms'/                               ABF15260
     ,' T-TO(ms)  D(ratio)')                                            ABF15270
14    FORMAT(' Density at distance=',F9.3,1X,A2/                        ABF15280
     ,' Time of arrival, TO=',F9.3,' ms'/                               ABF15290
     ,' T-TO(ms) D(',A6,')')                                            ABF15300
15    FORMAT(F9.3,F10.3)                                                ABF15310
      UNITS(2)=' '                                                      ABF15320
      IF(IRATIO.EQ.2) UNITS(2)=UDEN(IUNITS)                             ABF15330
      RETURN                                                            ABF15340
                                                                        ABF15350
C     Output temperature time history:                                  ABF15360
 1200 IF(IRATIO.EQ.1) WRITE(IFILE,16) RO,UDIST(IUNITS),TOA              ABF15370
      IF(IRATIO.EQ.2) WRITE(IFILE,17) RO,UDIST(IUNITS),TOA,UTEMP(IUNITS)ABF15380
      WRITE(IFILE,18) (X(I),Y(I),I=1,LENGTH)                            ABF15390
16    FORMAT(' Temperature at distance=',F9.3,1X,A2/                    ABF15400
     ,' Time of arrival, TO=',F9.3,' ms'/                               ABF15410
     ,' T-TO(ms)  T(ratio)')                                            ABF15420
17    FORMAT(' Temperature at distance=',F9.3,1X,A2/                    ABF15430
     ,' Time of arrival, TO=',F9.3,' ms'/                               ABF15440
     ,' T-TO(ms)      T(',A1,')')                                       ABF15450
18    FORMAT(F9.3,F10.3)                                                ABF15460
      UNITS(2)=' '                                                      ABF15470
      IF(IRATIO.EQ.2) UNITS(2)=UTEMP(IUNITS)                            ABF15480
      RETURN                                                            ABF15490
                                                                        ABF15500
C     Output sound speed time history:                                  ABF15510
 1300 IF(IRATIO.EQ.1) WRITE(IFILE,19) RO,UDIST(IUNITS),TOA              ABF15520
      IF(IRATIO.EQ.2) WRITE(IFILE,20) RO,UDIST(IUNITS),TOA,UVEL(IUNITS) ABF15530
      WRITE(IFILE,21) (X(I),Y(I),I=1,LENGTH)                            ABF15540
19    FORMAT(' Sound speed at distance=',F9.3,1X,A2/                    ABF15550
     ,' Time of arrival, TO=',F9.3,' ms'/                               ABF15560
     ,' T-TO(ms)  A(ratio)')                                            ABF15570
20    FORMAT(' Sound speed at distance=',F9.3,1X,A2/                    ABF15580
```

```
     ,' Time of arrival, TO=',F9.3,' ms'/
     ,' T-TO(ms)  A(',A5,')')
21    FORMAT(F9.3,F10.3)
      UNITS(2)=' '
      IF(IRATIO.EQ.2) UNITS(2)=UVEL(IUNITS)
      RETURN

C     Output entropy change time history:
 1400 WRITE(IFILE,22) RO,UDIST(IUNITS),TOA,(X(I),Y(I),I=1,LENGTH)
22    FORMAT(' Entropy change at distance=',F9.3,1X,A2/
     ,' Time of arrival, TO=',F9.3,' ms'/
     ,' T-TO(ms) DS(J/K/g)'/
     ,(F9.3,F10.6))
      UNITS(2)='J/K/g'
      RETURN

C     Output energy density time history:
 1500 WRITE(IFILE,23) RO,UDIST(IUNITS),TOA,(X(I),Y(I),I=1,LENGTH)
23    FORMAT(' Energy density at distance=',F9.3,1X,A2/
     ,' Time of arrival, TO=',F9.3,' ms'/
     ,' T-TO(ms)   E(J/cc)'/
     ,(F9.3,F10.3))
      UNITS(2)='J/cc'
      RETURN

C     Output available work time history:
 1600 WRITE(IFILE,24) RO,UDIST(IUNITS),TOA,(X(I),Y(I),I=1,LENGTH)
24    FORMAT(' Available work at distance=',F9.3,1X,A2/
     ,' Time of arrival, TO=',F9.3,' ms'/
     ,' T-TO(ms)   W(J/cc)'/
     ,(F9.3,F10.6))
      UNITS(2)='J/cc'
      RETURN
      END


*.VAPOUR
*************************************************************************
* Subprogram to interpolate for atmospheric water vapour pressure.

      SUBROUTINE VAPOUR(P,TO,RH)
      REAL*4 T(82)/-15.,-14.,-13.,-12.,-11.,-10.,-9.,-8.,-7.,-6.,-5.,
     ,-4.,-3.,-2.,-1.,0.,1.,2.,3.,4.,5.,6.,7.,8.,9.,10.,11.,12.,13.,14.,
     ,15.,16.,17.,18.,19.,20.,21.,22.,23.,24.,25.,26.,27.,28.,29.,30.,
     ,31.,32.,33.,34.,35.,36.,37.,38.,39.,40.,41.,42.,43.,44.,45.,46.,
     ,47.,48.,49.,50.,51.,52.,53.,54.,55.,56.,57.,58.,59.,60.,61.,62.,
     ,63.,64.,65.,66./
      REAL*4 SVP(82)/1.436,1.560,1.691,1.834,1.987,2.149,2.326,2.514,
     ,2.715,2.931,3.163,3.410,3.673,3.956,4.258,4.579,4.926,5.294,5.685,
     ,6.101,6.543,7.013,7.513,8.045,8.609,9.209,9.844,10.518,11.231,
     ,11.987,12.788,13.634,14.530,15.477,16.477,17.535,18.650,19.827,
     ,21.068,22.377,23.756,25.209,26.739,28.349,30.043,31.824,33.695,
     ,35.663,37.729,39.898,42.175,44.563,47.067,49.692,52.442,55.324,
     ,58.34,61.50,64.80,68.26,71.88,75.65,79.60,83.71,88.02,92.51,97.20,
     ,102.09,107.20,112.51,118.04,123.80,129.82,136.08,142.60,149.38,
     ,156.43,163.77,171.38,179.31,187.54,196.09/
      INTEGER*4 NPTS/82/

C     Check that temperature is in range:
      IF(TO.LT.-15..OR.TO.GT.66.) PRINT*,'WARNING(VAPOUR): Temperature',
```

```
     ,' is out of range.'

C     Interpolate for saturation vapour presure:
      CALL LINT(TO,SVPO,T,SVP,NPTS)

C     Compute vapour pressure, in kPa:
      P=SVPO*101.325/760.*RH/100.
      RETURN

C     Reference:  CRC Handbook, 65th edition, D-192.
      END


*.ATMOS
************************************************************************
* Subprogram to assign ambient atmospheric conditions at given altitude.

      SUBROUTINE ATMOS(ZO)
      PRINT*,' SORRY, BALANCE OF THIS OPTION IS NOT AVAILABLE YET.'
      STOP
      END


*.SCREEN
************************************************************************
* Subprogram to partition output.

      SUBROUTINE SCREEN(KEY)
      COMMON/BLOCK1/ DUMMY1(2),IFILE
      CHARACTER*1 KEY(21)
      DATA INDEX/0/
      INDEX=INDEX+1

C     Check for initialization:
      IF(KEY(1).EQ.'Z') GO TO 100
      IF(INDEX.EQ.1) GO TO 101

C     Offer option to clear terminal screen:
  090 IF(IFILE.EQ.6.AND.KEY(21).NE.'t') WRITE(6,1)
1     FORMAT(/' READY.  (Clear screen, then hit enter/return key)')
      IF(IFILE.EQ.6.AND.KEY(21).NE.'t') READ(5,*,END=9001)
      IF(INDEX.EQ.1) RETURN

C     Insert flag in disk file:
      WRITE(IFILE,2) KEY
2     FORMAT(/T53,21A1)
      RETURN

C     Reset index:
  100 INDEX=0
      RETURN

C     Begin new page in disk file:
  101 IF(IFILE.NE.6) WRITE(IFILE,3) KEY
3     FORMAT(//'1',21A1,T53,'           keyword')
      GO TO 090

 9001 REWIND 5
      RETURN
      END
```

```
*.EVENT
**********************************************************************
* Subprogram to summarize the event.

      SUBROUTINE EVENT
      COMMON/BLOCK1/ IUNITS,IRATIO,IFILE,ITYPE,IATMOS,ISCALE,IBURST,IB2,
     ,IDATA,ITIME,IFORM,IPEAK
      COMMON/BLOCK2/ WS,TS,PS,CS,DS,CMASS,EMASS,TO,PO,RH,DO,CO,S,ST,HOB
      CHARACTER*1 UTEMP(2)/'C','F'/
      CHARACTER*2 UDIST(2)/' m','ft'/,UMASS(2)/'kg','lb'/
      CHARACTER*3 UPRESS(2)/'kPa','psi'/
      CHARACTER*5 UVEL(2)/'m/ms ','ft/ms'/
      CHARACTER*9 TYPE(6)/'TNT      ','ANFO     ','Pentolite',
     ,'hexogen  ','of gas   ','nuclear  '/
      CHARACTER*21 CONFIG(3)/'free field      event',
     ,'surface burst   event','height of burst event'/
      IF(HOB.GT.0.) GO TO 100
      CALL SCREEN(CONFIG(IBURST))

      IF(ITYPE.GT.4) GO TO 090
      IF(CMASS.LT.1.E5) WRITE(IFILE,1) CMASS,UMASS(IUNITS),TYPE(ITYPE)
      IF(CMASS.GE.1.E5) WRITE(IFILE,2) CMASS,UMASS(IUNITS),TYPE(ITYPE)
1     FORMAT(' Charge=',F9.3,1X,A2,' of ',A9)
2     FORMAT(' Charge=',E14.7,1X,A2,' of ',A9)
      GO TO 091
  090 IF(ITYPE.EQ.5) WRITE(IFILE,7) CMASS,UDIST(IUNITS),TYPE(ITYPE)
      IF(ITYPE.EQ.6) WRITE(IFILE,8) CMASS,TYPE(ITYPE)
7     FORMAT(' Charge=',F9.3,' cubic ',A2,1X,A9)
8     FORMAT(' Charge=',F9.3,' kT ',A9)
  091 IF(EMASS.NE.CMASS) WRITE(IFILE,3) EMASS,UMASS(IUNITS)
3     FORMAT(' TNT-equivalent mass=',E14.7,1X,A2)
      IF(IATMOS.EQ.1) WRITE(IFILE,4)
      IF(IATMOS.EQ.3) WRITE(IFILE,5)
4     FORMAT(' Standard sea-level atmosphere:')
5     FORMAT(' Atmosphere as follows:')
      WRITE(IFILE,6) TO,UTEMP(IUNITS),PO,UPRESS(IUNITS),CO,UVEL(IUNITS)
6     FORMAT(
     ,' Temperature            TO=',F10.5,1X,A2/
     ,' Pressure               PO=',F10.5,1X,A3/
     ,' Computed sound speed   CO=',F10.7,1X,A5)
      WRITE(IFILE,9) S,ST
9     FORMAT(' Scale factors: S,ST=',F7.3,',',F7.3)
      RETURN

  100 WRITE(IFILE,10) HOB,UDIST(IUNITS)
10    FORMAT(/' Height of burst=',F10.5,1X,A2)
      RETURN
      END


*.INPUT
**********************************************************************
* Subprogram to input source data.

      SUBROUTINE INPUT(R,T1,P1,D1,U1,T2,P2,D2,U2,T,J1,NJ,P,D,U,NR,NT,
     ,NRM,NTM,NDM)
      COMMON/BLOCK1/ DUM1(2),IFILE,DUM2(3),IBURST
      REAL*4 R(NRM),T1(NRM),P1(NRM),D1(NRM),U1(NRM),T2(NRM),P2(NRM),
```

```
     ,D2(NRM),U2(NRM),T(NTM),P(NRM,NDM),D(NRM,NDM),U(NRM,NDM),G/1.4/    ABF17390
      INTEGER*4 NJ(NRM),J1(NRM),LFILE/0/                                ABF17400
      CHARACTER*4 CODE(3)                                               ABF17410
                                                                        ABF17420
C     Set input file number:                                            ABF17430
      IF(IBURST.NE.2) JFILE=8                                           ABF17440
      IF(IBURST.EQ.2) JFILE=9                                           ABF17450
      IF(JFILE.EQ.LFILE) RETURN                                         ABF17460
      WRITE(6,1) JFILE                                                  ABF17470
1     FORMAT(' Reading from file number ',I1,'...')                     ABF17480
      LFILE=JFILE                                                       ABF17490
                                                                        ABF17500
C     Input title and index records:                                    ABF17510
      READ(JFILE,2,END=9001) NR,KIND,CODE                               ABF17520
2     FORMAT(/2I2,64X,3A4)                                              ABF17530
      IF(CODE(1).NE.'RECO') GO TO 9002                                  ABF17540
      IF(KIND.NE.3) GO TO 9003                                          ABF17550
      IF(NR+1.GT.NRM) GO TO 9004                                        ABF17560
      NR=NR+1                                                           ABF17570
                                                                        ABF17580
C     Input radius vector:                                              ABF17590
      READ(JFILE,3,END=9005) (R(K),K=2,NR)                              ABF17600
3     FORMAT(10F8.0)                                                    ABF17610
                                                                        ABF17620
C     Input flow field data:                                            ABF17630
      DO 101 K=1,NR                                                     ABF17640
      NJ(K)=0                                                           ABF17650
  101 CONTINUE                                                          ABF17660
      NT=0                                                              ABF17670
  102 READ(JFILE,4,END=9005) TJ,K1,K2,(P1(K),D1(K),U1(K),K=K1,K2)       ABF17680
4     FORMAT(F8.0,2I4,8F8.0/(10F8.0))                                   ABF17690
      IF(TJ.EQ.9999.) GO TO 104                                         ABF17700
      IF(NT.EQ.NTM) GO TO 9006                                          ABF17710
      NT=NT+1                                                           ABF17720
      T(NT)=TJ                                                          ABF17730
      IF(K1.LE.0.OR.K1.GT.NR.OR.K2.LT.K1.OR.K2.GT.NR) GO TO 9007        ABF17740
      DO 103 K=K1,K2                                                    ABF17750
      L=K+1                                                             ABF17760
      IF(NJ(L).EQ.NDM) GO TO 9008                                       ABF17770
      NJ(L)=NJ(L)+1                                                     ABF17780
      IF(NJ(L).EQ.1) J1(L)=NT                                           ABF17790
      P(L,NJ(L))=P1(K)                                                  ABF17800
      D(L,NJ(L))=D1(K)                                                  ABF17810
      U(L,NJ(L))=U1(K)/SQRT(G)                                          ABF17820
  103 CONTINUE                                                          ABF17830
      GO TO 102                                                         ABF17840
  104 R(1)=P1(1)                                                        ABF17850
      T1(1)=D1(1)                                                       ABF17860
      P1(1)=2.*G/(G+1.)*(U1(1)**2-1.)+1.                                ABF17870
      D1(1)=(U1(1)**2/((G-1.)/(G+1.)*(U1(1)**2-1.)+1.))                 ABF17880
      U1(1)=SQRT(G)/((1.-(G-1.)/(G+1.))*(U1(1)**2-1.)/U1(1))            ABF17890
                                                                        ABF17900
C     Input time-of-arrival data:                                       ABF17910
      READ(JFILE,3,END=9009) (T1(K),K=2,NR)                             ABF17920
      READ(JFILE,3,END=9009) (P1(K),K=2,NR)                             ABF17930
      READ(JFILE,3,END=9009) (D1(K),K=2,NR)                             ABF17940
      READ(JFILE,3,END=9009) (U1(K),K=2,NR)                             ABF17950
      READ(JFILE,3,END=9009) (T2(K),K=2,NR)                             ABF17960
      READ(JFILE,3,END=9009) (P2(K),K=2,NR)                             ABF17970
      READ(JFILE,3,END=9009) (D2(K),K=2,NR)                             ABF17980
```

```
      READ(JFILE,3,END=9009) (U2(K),K=2,NR)
      DO 105 K=2,NR
      U1(K)=U1(K)/SQRT(G)
      U2(K)=U2(K)/SQRT(G)
  105 CONTINUE

C     Conclude:
      WRITE(6,5)
5     FORMAT(' Source data input is complete'/)
      IF(IFILE.NE.6) WRITE(IFILE,6) CODE(2),CODE(3)
6     FORMAT(/' Source data code= ',2A4)
      REWIND JFILE
      RETURN

C     Report errors:
 9001 WRITE(6,9901)
 9901 FORMAT('0ERROR(INPUT9001): PREMATURE END OF FILE')
      STOP 1
 9002 WRITE(6,9902) CODE(1)
 9902 FORMAT('0ERROR(INPUT9002): INVALID FILE TYPE',A5)
      STOP 1
 9003 WRITE(6,9903) KIND
 9903 FORMAT('0ERROR(INPUT9003): INVALID FILE KIND',I3)
      STOP 1
 9004 WRITE(6,9904) NR,NRM
 9904 FORMAT('0ERROR(INPUT9004): FILE TOO LARGE',2I5)
      STOP 1
 9005 WRITE(6,9905)
 9905 FORMAT('0ERROR(INPUT9005): PREMATURE END OF FILE')
      STOP 1
 9006 WRITE(6,9906) NTM
 9906 FORMAT('0ERROR(INPUT9006): FILE TO LARGE',I5)
      STOP 1
 9007 WRITE(6,9907) K1,K2,NR
 9907 FORMAT('0ERROR(INPUT9007): INVALID COLUMN INDICES',3I5)
      STOP 1
 9008 WRITE(6,9908) K,NDM
 9908 FORMAT('0ERROR(INPUT9008): FILE TOO LARGE',2I5)
      STOP 1
 9009 WRITE(6,9909)
 9909 FORMAT('0ERROR(INPUT9009): PREMATURE END OF FILE')
      STOP 1
      END


*.LINT
************************************************************************
* Subprogram to handle general linear interpolation.

      SUBROUTINE LINT(XO,YO,X,Y,NPTS)
      REAL*4 X(NPTS),Y(NPTS)

      YO=9999.
      IF(NPTS.LT.2) RETURN

      IX=0
      IF(X(NPTS).LT.X(1)) IX=1

      IEXT=0
      IF(IX.EQ.0.AND.XO.LT.X(1)) IEXT=1
```

```
      IF(IX.EQ.1.AND.XO.GT.X(1)) IEXT=2
      IF(IX.EQ.0.AND.XO.GT.X(NPTS)) IEXT=3
      IF(IX.EQ.1.AND.XO.LT.X(NPTS)) IEXT=4
      IF(IEXT.EQ.0) GO TO 100

      WRITE(6,1)
1     FORMAT(' NOTE: Extrapolation, or apparent extrapolation')
      IF(IEXT.EQ.1.AND.X(1)-XO.GT.X(2)-X(1)) WRITE(6,2)
      IF(IEXT.EQ.2.AND.XO-X(1).GT.X(1)-X(2)) WRITE(6,2)
      IF(IEXT.EQ.3.AND.XO-X(NPTS).GT.X(NPTS)-X(NPTS-1)) WRITE(6,2)
      IF(IEXT.EQ.4.AND.X(NPTS)-XO.GT.X(NPTS-1)-X(NPTS)) WRITE(6,2)
2     FORMAT(' WARNING: Extrapolation is considered excessive')

  100 DO 101 N=1,NPTS
      IF(IX.EQ.0.AND.XO.LE.X(N)) GO TO 102
      IF(IX.EQ.1.AND.XO.GE.X(N)) GO TO 102
  101 CONTINUE
      N=NPTS
  102 IF(N.EQ.1) N=N+1
      SLOPE=(Y(N)-Y(N-1))/(X(N)-X(N-1))
      YO=Y(N-1)+SLOPE*(XO-X(N-1))
      RETURN
      END


*.EXPAND
************************************************************************
* Subprogram to expand P,D,U data (compute other physical parameters).

      SUBROUTINE EXPAND(P,D,U,PVM,SSR,HOP,DEN,DYP,TOP,END,TEMP,PVML,ENT,
     ,WORK,M,IM)
      COMMON/BLOCK2/ DUM(2),PS,CS,DS
      REAL*4 M,MM,GAMMA/1.4/,RAIR/.28705/

      G=(GAMMA-1.)/(GAMMA+1.)
      G1=2./GAMMA
      G2=(GAMMA-1.)/2.
      G3=GAMMA/(GAMMA-1.)
      G4=(GAMMA+1.)/2.
      G5=2.*GAMMA/(GAMMA+1.)
      G6=1./(GAMMA-1.)

      HOP=P-1.
      M=SQRT(HOP/(G+1.)+1.)
      PVM=U
      DEN=D
      IF(IM.EQ.1) PVM=(1.-G)*(M*M-1.)/M
      IF(IM.EQ.1) DEN=M*M/(G*(M*M-1.)+1.)
      SSR=SQRT((HOP+1)/DEN)
      DYP=DEN*DS*(PVM*CS*1000.)**2/2./(PS*1000.)
      MM=G1*DYP/(HOP+1.)
      IF(MM.LE.1.) TOP=(HOP+1.)*(G2*MM+1.)**G3-1.
      IF(MM.GT.1.) TOP=(HOP+1.)*MM*G4**G3/(G5-G/MM)**G6-1.
      END=(HOP*(PS*1000.)/(GAMMA-1.)+DYP*DS*(CS*1000.)**2/GAMMA)/1.E6

      TEMP=SSR**2
      PVML=PVM/SSR
      WORK=((TOP+1.)/(GAMMA-1.)*(1.-(TOP+1.)**((1.-GAMMA)/GAMMA))+
     ,(1.-(TOP+1.)**(1./GAMMA)))*PS*1000./1.E6
      ENT=RAIR/(GAMMA-1.)*ALOG((HOP+1.)/DEN**GAMMA)
```

```
                                                                        ABF19190
      RETURN                                                            ABF19200
      END                                                               ABF19210
                                                                        ABF19220
                                                                        ABF19230
*.INTERP                                                                ABF19240
************************************************************************ABF19250
* Subprogram to offer interpolation option.                             ABF19260
                                                                        ABF19270
      SUBROUTINE INTERP(X,Y,N,UNITS,LINEAR)                             ABF19280
      REAL*4 X(N),Y(N)                                                  ABF19290
      CHARACTER*6 UNITS(2),HOLD                                         ABF19300
      COMMON/BLOCK1/ ID1(2),IFILE,ID2(3),IBURST,ID3,IDATA,ID4(2),IPEAK  ABF19310
                                                                        ABF19320
C     Check for interpolation or extrapolation:                         ABF19330
  100 WRITE(6,1)                                                        ABF19340
1     FORMAT(/' Do you want to interpolate?'/                           ABF19350
     ,' 1=yes, normal interpolation'/                                   ABF19360
     ,' 2=yes, inverse      "       ')                                  ABF19370
      READ(5,*,END=9001) INDEX                                          ABF19380
      IF(INDEX.NE.1.AND.INDEX.NE.2) GO TO 100                           ABF19390
      IF(INDEX.EQ.2) HOLD=UNITS(1)                                      ABF19400
      IF(INDEX.EQ.2) UNITS(1)=UNITS(2)                                  ABF19410
      IF(INDEX.EQ.2) UNITS(2)=HOLD                                      ABF19420
      IF(UNITS(1).NE.'      ') WRITE(6,2) UNITS(1)                      ABF19430
      IF(UNITS(1).EQ.'      ') WRITE(6,7)                               ABF19440
2     FORMAT(' Where (input one value at a time, units=',1X,A6,')')     ABF19450
7     FORMAT(' Where (input one value at a time)')                      ABF19460
      IF(IBURST.EQ.3.AND.IPEAK.EQ.1.AND.INDEX.EQ.2) WRITE(6,10)         ABF19470
10    FORMAT(' (Be careful interpolating near transition, because the'/ ABF19480
     ,       ' inverse function may be double-valued there.  Check the'/ABF19490
     ,       ' RR profile for a ''lift'' just before transition, e.g.)')ABF19500
  101 CONTINUE                                                          ABF19510
      READ(5,*,END=9002) XO                                             ABF19520
                                                                        ABF19530
C     Interpolate and ouptput result:                                   ABF19540
      IF(LINEAR.EQ.1) GO TO 103                                         ABF19550
      IF(INDEX.EQ.1.AND.(XO.LT.X(1).OR.XO.GT.X(N))) GO TO 200           ABF19560
      IF(INDEX.EQ.2.AND.(XO.LT.Y(1).OR.XO.GT.Y(N))) GO TO 202           ABF19570
C     IF(INDEX.EQ.1) CALL TINT01(XO,YO,X,Y,N)                           ABF19580
C     IF(INDEX.EQ.2) CALL TINT01(XO,YO.Y,X,N)                           ABF19590
  104 IF(IFILE.NE.6.AND.INDEX.EQ.1) WRITE(IFILE,5) XO,YO                ABF19600
      IF(IFILE.NE.6.AND.INDEX.EQ.2) WRITE(IFILE,5) YO,XO                ABF19610
5     FORMAT(F9.3,F10.6,' (interpolated)')                              ABF19620
  102 IF(UNITS(1).NE.'      ') WRITE(6,4) YO,UNITS(2),UNITS(1)          ABF19630
      IF(UNITS(1).EQ.'      ') WRITE(6,9) YO,UNITS(2)                   ABF19640
4     FORMAT(' Result=',F12.6,1X,A6//' Where else (',A6,')')            ABF19650
9     FORMAT(' Result=',F12.6,1X,A6//' Where else')                     ABF19660
      GO TO 101                                                         ABF19670
  103 IF(INDEX.EQ.1) CALL LINT(XO,YO,X,Y,N)                             ABF19680
      IF(INDEX.EQ.2) CALL LINT(XO,YO,Y,X,N)                             ABF19690
      GO TO 104                                                         ABF19700
                                                                        ABF19710
C     Extrapolate and output result:                                    ABF19720
  200 WRITE(6,6)                                                        ABF19730
6     FORMAT(' WARNING: Extrapolation')                                 ABF19740
      IF(XO.GT.X(N)) GO TO 201                                          ABF19750
      X1=X(1)                                                           ABF19760
      X2=X(2)                                                           ABF19770
      Y1=Y(1)                                                           ABF19780
```

```
      Y2=Y(2)                                                           ABF19790
      GO TO 204                                                         ABF19800
  201 X1=X(N-1)                                                         ABF19810
      X2=X(N)                                                           ABF19820
      Y1=Y(N-1)                                                         ABF19830
      Y2=Y(N)                                                           ABF19840
      GO TO 204                                                         ABF19850
  202 WRITE(6,6)                                                        ABF19860
      IF(XO.GT.Y(N)) GO TO 203                                          ABF19870
      X1=Y(1)                                                           ABF19880
      X2=Y(2)                                                           ABF19890
      Y1=X(1)                                                           ABF19900
      Y2=X(2)                                                           ABF19910
      GO TO 204                                                         ABF19920
  203 X1=Y(N-1)                                                         ABF19930
      X2=Y(N)                                                           ABF19940
      Y1=X(N-1)                                                         ABF19950
      Y2=X(N)                                                           ABF19960
  204 IF(XO.LT.X1.AND.(X1-XO).GT.2.*(X2-X1)) WRITE(6,3)                 ABF19970
      IF(XO.GT.X2.AND.(XO-X2).GT.2.*(X2-X1)) WRITE(6,3)                 ABF19980
3     FORMAT(' WARNING: Extrapolation is considered excessive')         ABF19990
      SLOPE=(Y2-Y1)/(X2-X1)                                             ABF20000
      YO=Y1+SLOPE*(XO-X1)                                               ABF20010
      IF(IFILE.NE.6.AND.INDEX.EQ.1) WRITE(IFILE,8) XO,YO                ABF20020
      IF(IFILE.NE.6.AND.INDEX.EQ.2) WRITE(IFILE,8) YO,XO                ABF20030
8     FORMAT(F9.3,F10.3,' (extrapolated)')                              ABF20040
      GO TO 102                                                         ABF20050
                                                                        ABF20060
 9001 REWIND 5                                                          ABF20070
      RETURN                                                            ABF20080
 9002 REWIND 5                                                          ABF20090
      IF(INDEX.NE.2) RETURN                                             ABF20100
      HOLD=UNITS(1)                                                     ABF20110
      UNITS(1)=UNITS(2)                                                 ABF20120
      UNITS(2)=HOLD                                                     ABF20130
      RETURN                                                            ABF20140
      END                                                               ABF20150
                                                                        ABF20160
                                                                        ABF20170
*.IMPULS                                                                ABF20180
************************************************************************ABF20190
* Subprogram to integrate pressure impulse.                             ABF20200
                                                                        ABF20210
      SUBROUTINE IMPULS(X,Y,L,UNITS)                                    ABF20220
      REAL*4 X(L),Y(L)                                                  ABF20230
      CHARACTER*6 UNITS(2)                                              ABF20240
      COMMON/BLOCK1/ IDUM1(2),IFILE,IDUM2(5),IDATA                      ABF20250
                                                                        ABF20260
C     Check for integration:                                            ABF20270
      WRITE(6,1)                                                        ABF20280
1     FORMAT(/' Do you want integrate?  1=yes')                         ABF20290
      READ(5,*,END=9001) INDEX                                          ABF20300
      IF(INDEX.NE.1) RETURN                                             ABF20310
  099 WRITE(6,2)                                                        ABF20320
2     FORMAT(/' For how long after time of arrival?  (ms)'/             ABF20330
     ,         ' 0=integrate all data')                                 ABF20340
      READ(5,*,END=9001) TIME                                           ABF20350
      IF(TIME.LT.0.) GO TO 099                                          ABF20360
      IF(TIME.EQ.0.) TIME=X(L)                                          ABF20370
      CALL LINT(TIME,YTIME,X,Y,L)                                       ABF20380
```

```
C     Integrate:
      XL=0.
      YL=Y(1)
      AREA=0.
      WRITE(6,10) XL,YL,AREA
10    FORMAT(/' T-TO(ms)      data  integral'/F9.3,2F10.3)
      DO 101 I=2,L
      IF(X(I).GT.TIME) GO TO 102
      AREA=AREA+(X(I)-XL)*(Y(I)+YL)/2.
      WRITE(6,11) X(I),Y(I),AREA
11    FORMAT(F9.3,2F10.3)
      XL=X(I)
      YL=Y(I)
  101 CONTINUE
  102 AREA=AREA+(TIME-XL)*(YTIME+YL)/2.
      WRITE(6,11) TIME,YTIME,AREA

C     Output results:
      IF(IDATA.NE.5.AND.IDATA.NE.6.AND.IDATA.NE.7) GO TO 103
      IF(UNITS(2).EQ.' ') WRITE(6,3) AREA
      IF(UNITS(2).NE.' ') WRITE(6,4) AREA,UNITS(2)
3     FORMAT(/' Pressure impulse=',F9.3,' atm-ms')
4     FORMAT(/' Pressure impulse=',F9.3,1X,A3,'-ms')
      IF(IFILE.EQ.6) RETURN
      IF(UNITS(2).EQ.' ') WRITE(IFILE,3) AREA
      IF(UNITS(2).NE.' ') WRITE(IFILE,4) AREA,UNITS(2)
      WRITE(IFILE,5) TIME
5     FORMAT(' at T-TO=',F9.3,' ms')
      RETURN
  103 IF(IDATA.NE.15.AND.IDATA.NE.16) GO TO 104
      IF(IDATA.EQ.15) WRITE(6,6) AREA,UNITS(2)
      IF(IDATA.EQ.16) WRITE(6,7) AREA,UNITS(2)
      WRITE(6,8)
6     FORMAT(/' Energy integral=',F9.6,1X,A4,'-ms')
7     FORMAT(/' Work integral=',F9.6,1X,A4,'-ms')
8     FORMAT(/' (this integral has the dimensions of pressure impulse)')
      IF(IFILE.EQ.6) RETURN
      IF(IDATA.EQ.15) WRITE(IFILE,6) AREA,UNITS(2)
      IF(IDATA.EQ.16) WRITE(IFILE,7) AREA,UNITS(2)
      WRITE(IFILE,5) TIME
      RETURN
  104 WRITE(6,9) AREA,UNITS(2)
9     FORMAT(/' Integral=',F10.4,1X,A6,'-ms')
      IF(IDATA.EQ.10.OR.IDATA.EQ.13.AND.UNITS(2).NE.' ') WRITE(6,10)
12    FORMAT(' (note that the result above is NOT a dispalacement)')
      IF(IFILE.NE.6) WRITE(IFILE,8) AREA,UNITS(2)
      IF(IFILE.NE.6) WRITE(IFILE,5) TIME
      RETURN

 9001 REWIND 5
      RETURN
      END


*.INTEGR
***********************************************************************
* Subprogram to integrate radial (wave) profiles.

      SUBROUTINE INTEGR(X,Y,L,UNITS)
```

```
      REAL*4 X(L),Y(L)                                                  ABF20990
      CHARACTER*2 UDIST(2)/' m','ft'/                                   ABF21000
      CHARACTER*6 UNITS(2)                                              ABF21010
      COMMON/BLOCK1/ IUNITS,IDUM1,IFILE,IDUM2(3),IBURST,IDUM3,IDATA     ABF21020
                                                                        ABF21030
C     Check for integration:                                            ABF21040
      WRITE(6,1)                                                        ABF21050
1     FORMAT(/' Do you want integrate?  1=yes')                         ABF21060
      READ(5,*,END=9001) INDEX                                          ABF21070
      IF(INDEX.NE.1) RETURN                                             ABF21080
  099 WRITE(6,2) UDIST(IUNITS)                                          ABF21090
2     FORMAT(/' Between the shock front and what radius?  (',A2,')'/    ABF21100
     ,        ' 0=integrate all data')                                  ABF21110
      READ(5,*,END=9001) RLIM                                           ABF21120
      IF(RLIM.LT.0..OR.RLIM.GE.X(1)) GO TO 099                          ABF21130
      IF(RLIM.EQ.0.) RLIM=X(L)                                          ABF21140
      CALL LINT(RLIM,YRLIM,X,Y,L)                                       ABF21150
                                                                        ABF21160
C     Integrate:                                                        ABF21170
      XL=X(1)                                                           ABF21180
      YL=Y(1)                                                           ABF21190
      AREA=0.                                                           ABF21200
      FACTOR=8.*ARSIN(1.)                                               ABF21210
      IF(IBURST.EQ.2) FACTOR=FACTOR/2.                                  ABF21220
      WRITE(6,10) UNITS(IUNITS),XL,YL,AREA                              ABF21230
10    FORMAT(/'    R(',A2,')      data  integral'/F9.3,2F10.3)          ABF21240
      DO 101 I=2,L                                                      ABF21250
      IF(X(I).LT.RLIM) GO TO 102                                        ABF21260
      AREA=AREA+FACTOR*(XL-X(I))*(YL*XL**2+Y(I)*X(I)**2)/2              ABF21270
      WRITE(6,11) X(I),Y(I),AREA                                        ABF21280
11    FORMAT(F9.3,2F10.3)                                               ABF21290
      XL=X(I)                                                           ABF21300
      YL=Y(I)                                                           ABF21310
  101 CONTINUE                                                          ABF21320
  102 AREA=AREA+FACTOR*(XL-RLIM)*(YL*XL**2+YRLIM*RLIM**2)/2.            ABF21330
      WRITE(6,11) RLIM,YRLIM,AREA                                       ABF21340
                                                                        ABF21350
C     Output results:                                                   ABF21360
      IF(IDATA.NE.15.AND.IDATA.NE.16) GO TO 103                         ABF21370
      IF(IDATA.EQ.15) WRITE(6,3) AREA,UNITS(2),UDIST(IUNITS)            ABF21380
      IF(IDATA.EQ.16) WRITE(6,4) AREA,UNITS(2),UDIST(IUNITS)            ABF21390
      WRITE(6,5) X(1)-RLIM,UDIST(IUNITS)                                ABF21400
      IF(UNITS(2).EQ.'J/cc'.AND.UDIST(IUNITS).EQ.' m') WRITE(6,6)       ABF21410
3     FORMAT(/' Total energy=',F12.3,1X,A4,'-',A2,'**3')                ABF21420
4     FORMAT(/' Total work=',F12.3,1X,A4,'-',A2,'**3')                  ABF21430
5     FORMAT(' over the leading',F9.3,1X,A2,' of wave')                 ABF21440
6     FORMAT(' (note: J/cc- m**3 equals megaJoules)')                   ABF21450
      IF(IFILE.EQ.6) RETURN                                             ABF21460
      IF(IDATA.EQ.15) WRITE(IFILE,3) AREA,UNITS(2),UDIST(IUNITS)        ABF21470
      IF(IDATA.EQ.16) WRITE(IFILE,4) AREA,UNITS(2),UDIST(IUNITS)        ABF21480
      WRITE(IFILE,5) X(1)-RLIM,UDIST(IUNITS)                            ABF21490
      RETURN                                                            ABF21500
  103 IF(IDATA.NE.5.AND.IDATA.NE.6.AND.IDATA.NE.7) GO TO 104            ABF21510
      IF(UNITS(2).EQ.' ') WRITE(6,7) AREA,UDIST(IUNITS)                 ABF21520
      IF(UNITS(2).NE.' ') WRITE(6,8) AREA,UNITS(2),UDIST(IUNITS)        ABF21530
7     FORMAT(/' Volume integral=',E13.5,' atm-',A2,'**3')               ABF21540
8     FORMAT(/' Volume integral=',E13.5,1X,A3,'-',A2,'**3')             ABF21550
      WRITE(6,5) X(1)-RLIM,UDIST(IUNITS)                                ABF21560
      IF(IFILE.EQ.6) RETURN                                             ABF21570
      IF(UNITS(2).EQ.' ') WRITE(IFILE,7) AREA,UDIST(IUNITS)             ABF21580
```

```
      IF(UNITS(2).NE.' ') WRITE(IFILE,8) AREA,UNITS(2),UDIST(IUNITS)    ABF21590
      WRITE(6,5) X(1)-RLIM,UDIST(IUNITS)                                ABF21600
      RETURN                                                            ABF21610
  104 IF(UNITS(2).EQ.' ') WRITE(6,9) AREA,UDIST(IUNITS)                 ABF21620
      IF(UNITS(2).NE ' ') WRITE(6,8) AREA,UNITS(2),UDIST(IUNITS)        ABF21630
    9 FORMAT(/' Integral=',E13.5,1X,A6,'-',A2)                          ABF21640
      WRITE(6,5) X(1)-RLIM,UDIST(IUNITS)                                ABF21650
      IF(IFILE.EQ.6) RETURN                                             ABF21660
      IF(UNITS(2).EQ.' ') WRITE(IFILE,9) AREA,UDIST(IUNITS)             ABF21670
      IF(UNITS(2).NE.' ') WRITE(IFILE,8) AREA,UNITS(2),UDIST(IUNITS)    ABF21680
      WRITE(6,5) X(1)-RLIM,UDIST(IUNITS)                                ABF21690
      RETURN                                                            ABF21700
                                                                        ABF21710
 9001 REWIND 5                                                          ABF21720
      RETURN                                                            ABF21730
      END                                                               ABF21740
                                                                        ABF21750
                                                                        ABF21760
*.FFR                                                                   ABF21770
************************************************************************ABF21780
* Subprogram to handle free field reflection (HOB/ground surface case). ABF21790
                                                                        ABF21800
      SUBROUTINE FFR(R,T1,P1,NR,NRM,R2,T2,P2,D2,U2,V2,NR2,RD)           ABF21810
      PRINT*,' SORRY, BALANCE OF THIS OPTION IS NOT AVAILABLE YET.'     ABF21820
      STOP                                                              ABF21830
      END                                                               ABF21840
      SUBROUTINE RR0                                                    ABF21850
      RETURN                                                            ABF21860
      END                                                               ABF21870
      SUBROUTINE RR3                                                    ABF21880
      RETURN                                                            ABF21890
      END                                                               ABF21900
      SUBROUTINE RR4                                                    ABF21910
      RETURN                                                            ABF21920
      END                                                               ABF21930
```

Appendix B

DATA BASE FILE 8 (FREE FIELD TNT)

What follows is a listing of UVic/AIRBLAST data base file number 8 (free field TNT reconstruction). This data file is being supplied as file 'FFT DATA' on magnetic tape.

```
FREE FIELD TNT                                               /airblast data   31Dec86
99 3                                                                        RECDGCCG7Z1C
 1.03000 1.06000 1.09000 1.12000 1.15000 1.18000 1.21000 1.24000 1.28000 1.31000
 1.35000 1.39000 1.43000 1.47000 1.51000 1.55000 1.59000 1.64000 1.68000 1.73000
 1.78000 1.83000 1.88000 1.93000 1.98000 2.04000 2.09000 2.15000 2.21000 2.27000
 2.33000 2.40000 2.47000 2.53000 2.60000 2.68000 2.75000 2.83000 2.91000 2.99000
 3.07000 3.15000 3.24000 3.33000 3.42000 3.52000 3.62000 3.72000 3.82000 3.93000
 4.04000 4.15000 4.26000 4.38000 4.50000 4.63000 4.75000 4.89000 5.02000 5.16000
 5.30000 5.45000 5.60000 5.76000 5.92000 6.08000 6.25000 6.42000 6.60000 6.79000
 6.97000 7.17000 7.37000 7.57000 7.78000 8.00000 8.22000 8.45000 8.68000 8.92000
 9.17000 9.42000 9.68000 9.9500010.2300010.5100010.8000011.1000011.4100011.73000
12.0500012.3900012.7300013.0800013.4500013.8200014.2000014.6000014.95000
 0.57555    3    3 9.59698 3.64037 2.53167
 0.65673    4    5 6.34667 2.69724 2.08872 7.09461 2.93404 2.19754
 0.73621    6    7 5.43506 2.42569 1.85985 6.23490 2.81410 1.96859
 0.81100    7    9 4.29467 2.04087 1.66621 5.08014 2.31156 1.78094 6.02137 2.92449
 1.89203
 0.90914    8   12 3.22389 1.66292 1.33426 3.77745 1.87074 1.44742 4.23725 2.23760
 1.56042 5.07886 2.67739 1.70258 5.82610 3.03059 1.81180
 1.00272    9   13 2.80612 1.50603 1.17948 3.10882 1.62779 1.28636 3.60532 1.99388
 1.40676 3.90745 2.17421 1.45087 4.84241 2.68939 1.60708
 1.10196   10   15 2.26974 1.29430 0.97854 2.65237 1.45330 1.12054 2.84460 1.68343
 1.15508 3.54684 2.07191 1.33698 4.06097 2.37179 1.41950 4.60751 2.67689 1.51681
 1.20503   11   17 1.97534 1.17206 0.85104 2.11002 1.23424 0.88925 2.62709 1.61740
 1.05978 2.91619 1.80156 1.13571 3.28512 2.03854 1.22852 3.73362 2.30359 1.33091
 4.38771 2.62720 1.45799
 1.31219   11   18 1.61052 1.01301 0.64917 1.73410 1.07285 0.70042 1.99013 1.30436
 0.81942 2.25632 1.45087 0.92051 2.55857 1.64087 1.02266 2.83008 1.83264 1.09969
 3.20075 2.06370 1.18487 3.67264 2.31376 1.27324
 1.43865   12   20 1.45522 0.94224 0.54086 1.60438 1.01491 0.62515 1.74806 1.18897
 0.71010 1.94377 1.30432 0.78634 2.14466 1.45016 0.85326 2.45747 1.65689 0.95744
 2.77814 1.88019 1.04791 3.06702 2.02342 1.11971 3.55923 2.33100 1.21745
 1.55712   12   21 1.25062 0.84560 0.41718 1.35030 0.89732 0.48461 1.48194 1.05668
 0.56209 1.57722 1.10478 0.60904 1.73506 1.21761 0.67328 1.91114 1.35497 0.75533
 2.17033 1.53366 0.84526 2.45190 1.71973 0.94168 2.79642 1.91299 1.02433 3.21495
 2.17469 1.12988
 1.67900   13   23 1.19344 0.82156 0.37271 1.27248 0.90460 0.43011 1.39224 1.01060
 0.49032 1.48580 1.07658 0.54469 1.62672 1.18742 0.60826 1.81937 1.33673 0.69892
 1.96257 1.45525 0.73914 2.22388 1.60835 0.82509 2.48778 1.79040 0.91280 2.77676
 1.96622 0.99007 3.19363 2.19194 1.09587
 1.80538   13   24 1.08184 0.76242 0.29352 1.17050 0.81026 0.35128 1.22543 0.92255
 0.38197 1.31295 0.96916 0.43480 1.41404 1.05208 0.48665 1.54599 1.17505 0.55602
 1.69365 1.28471 0.61823 1.87404 1.41937 0.68902 2.04913 1.52526 0.75697 2.36091
 1.73870 0.87302 2.53317 1.84417 0.90559 2.83504 2.02800 0.97943
 1.93733   13   25 1.01295 0.72742 0.22225 1.06211 0.75593 0.26311 1.09687 0.85233
 0.28377 1.17042 0.89278 0.34277 1.24412 0.94838 0.39490 1.36702 1.05131 0.44719
 1.46199 1.14343 0.49502 1.60710 1.26170 0.57009 1.77081 1.36499 0.64625 1.91575
 1.47972 0.69570 2.13685 1.62443 0.77372 2.37744 1.76884 0.84194 2.64577 1.93087
 0.91973
 2.07266   13   27 0.95390 0.69687 0.17159 0.98622 0.71694 0.19643 1.01656 0.77056
 0.22666 1.06070 0.83217 0.26286 1.13417 0.87295 0.30312 1.20770 0.93999 0.34778
 1.28005 1.02684 0.38299 1.36990 1.10408 0.44055 1.50144 1.21154 0.51249 1.64253
 1.29538 0.57757 1.81959 1.44359 0.64081 1.93971 1.52296 0.68255 2.17439 1.66571
 0.76691 2.41474 1.81881 0.83716 2.68023 1.96710 0.91384
 2.22662   15   28 0.95763 0.70204 0.17846 0.99679 0.79603 0.18590 1.03133 0.81565
 0.21735 1.08453 0.87046 0.24825 1.14827 0.94169 0.30424 1.22863 1.02150 0.35745
 1.30443 1.09574 0.39216 1.42449 1.16849 0.44672 1.55526 1.27502 0.51408 1.66557
 1.35962 0.56153 1.83939 1.46736 0.63276 1.96900 1.56316 0.67364 2.22289 1.71478
```

```
0.76266 2.41116 1.82842 0.81239
2.36870  16   29 0.94720 0.76754 0.14044 0.97195 0.78182 0.15836 1.01338 0.82928
0.19937 1.04612 0.86844 0.22633 1.12284 0.95788 0.27375 1.20052 1.01675 0.31035
1.27537 1.07825 0.36172 1.36372 1.14036 0.41078 1.47682 1.24366 0.46936 1.58735
1.31979 0.51322 1.75073 1.42641 0.58040 1.86619 1.50849 0.62146 2.05655 1.62660
0.68985 2.26386 1.75051 0.75938
2.53264  18   31 0.95469 0.79469 0.15161 0.97909 0.82627 0.16337 1.02476 0.88712
0.19479 1.07483 0.93952 0.22494 1.14346 0.99736 0.27126 1.22161 1.04848 0.32066
1.28216 1.11520 0.34841 1.39318 1.19680 0.40667 1.53088 1.29167 0.47114 1.63593
1.36937 0.51662 1.79397 1.47134 0.58791 1.91768 1.55256 0.62742 2.15446 1.69369
0.71281 2.30259 1.78413 0.75885
2.68390  19   31 0.94669 0.80665 0.13296 0.98143 0.84938 0.15098 1.02116 0.90577
0.18614 1.05987 0.94472 0.21114 1.13600 0.99410 0.25871 1.19140 1.05401 0.29088
1.26130 1.11429 0.33010 1.38286 1.19687 0.39143 1.46215 1.25460 0.43428 1.58227
1.34483 0.49076 1.73178 1.43998 0.55321 1.84306 1.51187 0.59075 2.00026 1.61257
0.64440
2.85544  20   33 0.91376 0.80712 0.09608 0.95934 0.85604 0.13343 0.99822 0.90513
0.15572 1.03378 0.92806 0.17814 1.07604 0.96282 0.20491 1.14489 1.03688 0.24969
1.23643 1.10490 0.30776 1.30298 1.15120 0.34771 1.39117 1.22001 0.38464 1.51550
1.30688 0.44508 1.61378 1.37414 0.49002 1.76439 1.47256 0.55144 1.91402 1.56350
0.60791 2.10570 1.67997 0.67628
3.01262  21   34 0.89601 0.81528 0.06536 0.96206 0.87456 0.11751 0.99117 0.90057
0.13607 1.03016 0.93332 0.16799 1.06813 0.98674 0.19230 1.13009 1.03540 0.23663
1.20388 1.08405 0.28140 1.27372 1.14523 0.31180 1.37177 1.21473 0.36524 1.48600
1.29402 0.42494 1.58546 1.35795 0.46623 1.73146 1.45465 0.53293 1.84830 1.52992
0.56967 2.02724 1.63716 0.63991
3.18992  23   35 0.94462 0.87015 0.10008 0.96600 0.89142 0.11384 1.01498 0.94377
0.15153 1.05541 0.98532 0.17901 1.11871 1.02870 0.21425 1.17303 1.07193 0.24660
1.24624 1.13400 0.28730 1.32427 1.18887 0.33131 1.42205 1.25585 0.38399 1.52485
1.32683 0.43013 1.65369 1.41046 0.48818 1.75165 1.47299 0.52764 1.90972 1.57026
0.58588
3.36997  24   36 0.91855 0.85529 0.07490 0.96277 0.90521 0.10729 1.00063 0.94484
0.12682 1.03307 0.97182 0.14863 1.10122 1.02431 0.19544 1.15806 1.07284 0.22887
1.22270 1.12200 0.26863 1.28572 1.16825 0.30534 1.36475 1.22576 0.34239 1.48254
1.30260 0.40258 1.56860 1.36097 0.44244 1.70442 1.44734 0.50183 1.84261 1.53183
0.55502
3.55490  25   37 0.90923 0.85747 0.05010 0.96660 0.92140 0.09731 0.99064 0.94314
0.11243 1.03290 0.97851 0.13988 1.07064 1.01187 0.16518 1.13111 1.06045 0.20735
1.19461 1.10850 0.24621 1.27801 1.16417 0.29396 1.35328 1.21987 0.33151 1.44861
1.28460 0.38196 1.54246 1.34587 0.42470 1.67086 1.42807 0.48577 1.77643 1.49373
0.52202
3.74371  26   38 0.88591 0.86578 0.01862 0.95201 0.91672 0.07596 0.98439 0.94546
0.09987 1.02156 0.97852 0.12775 1.05566 1.00943 0.14959 1.11801 1.05603 0.19290
1.16702 1.09085 0.22085 1.24046 1.14495 0.26112 1.32464 1.20376 0.31279 1.39800
1.25448 0.34733 1.50050 1.32190 0.39723 1.62072 1.39867 0.45209 1.71700 1.46067
0.48984
3.93543  27   39 0.87210 0.86107 0.00007 0.95106 0.92248 0.07214 0.97893 0.94917
0.09165 1.01557 0.97998 0.11842 1.04750 1.00785 0.13961 1.11798 1.05791 0.18224
1.16084 1.09196 0.21353 1.22034 1.13367 0.24618 1.28501 1.18060 0.28577 1.38348
1.24715 0.33427 1.47242 1.30493 0.37680 1.58667 1.37909 0.43095 1.67600 1.43701
0.47048
4.13058  28   40 0.87590 0.86980-0.00359 0.94860 0.92808 0.06714 0.97142 0.94936
0.07926 1.01308 0.98408 0.10483 1.03867 1.00374 0.12729 1.10396 1.05347 0.17029
1.14731 1.08467 0.20005 1.20545 1.12782 0.23079 1.28213 1.18102 0.27313 1.36018
1.23292 0.31491 1.44829 1.29106 0.36382 1.53896 1.35147 0.40438 1.64554 1.41920
0.45164
4.34858  30   42 0.92136 0.91395 0.02854 0.96926 0.95132 0.07371 0.99948 0.97624
0.09421 1.02749 1.00082 0.11660 1.06258 1.02635 0.13662 1.13345 1.07890 0.18119
1.19533 1.12261 0.22024 1.26381 1.16936 0.25957 1.32689 1.21260 0.29473 1.41352
1.27158 0.34205 1.49549 1.32488 0.37976 1.60502 1.39496 0.43011 1.69158 1.44950
```

```
0.46504
4.54881   31   43 0.91568 0.91345 0.02442 0.96806 0.95421 0.07062 0.99873 0.98073
0.08965 1.02912 1.00316 0.10661 1.06252 1.02948 0.12955 1.12969 1.07745 0.17466
1.17096 1.10732 0.20239 1.24324 1.15736 0.24593 1.30831 1.20290 0.28113 1.39391
1.25982 0.32494 1.47453 1.31235 0.36526 1.56061 1.36814 0.40475 1.66922 1.43624
0.45270
4.75172   32   44 0.91991 0.92007 0.01921 0.97181 0.96177 0.05911 0.98903 0.97509
0.07377 1.02720 1.00491 0.09984 1.05990 1.02947 0.12498 1.11196 1.06710 0.16130
1.16682 1.10538 0.19536 1.21642 1.14074 0.22289 1.29762 1.19698 0.26943 1.35911
1.23801 0.30041 1.44032 1.29182 0.34162 1.54225 1.35707 0.39324 1.62972 1.41262
0.43061
4.97784   33   45 0.91957 0.92201 0.01322 0.95971 0.95332 0.04635 0.98570 0.97565
0.06879 1.02337 1.00392 0.09721 1.04818 1.02288 0.11156 1.10412 1.06242 0.14970
1.16169 1.10378 0.18535 1.20636 1.13568 0.21024 1.28610 1.18961 0.25773 1.33648
1.22400 0.28586 1.42274 1.28097 0.33095 1.51396 1.34010 0.37431 1.59629 1.39223
0.41088
5.20659   34   46 0.90951 0.91744 0.00285 0.95571 0.95435 0.04442 0.98418 0.97630
0.06611 1.02036 1.00327 0.09020 1.04741 1.02301 0.10606 1.10011 1.06084 0.14200
1.14836 1.09632 0.17308 1.19517 1.12886 0.20258 1.25009 1.16691 0.23385 1.32632
1.21835 0.27551 1.40066 1.26716 0.31716 1.47331 1.31467 0.35072 1.57644 1.38038
0.40009
5.43760   35   47 0.90673 0.91916-0.00426 0.95062 0.95241 0.03215 0.97689 0.97255
0.05134 1.01874 1.00293 0.08187 1.04253 1.02088 0.09735 1.08974 1.05604 0.13156
1.13673 1.08915 0.16133 1.18458 1.12240 0.19114 1.24042 1.16142 0.22448 1.30694
1.20596 0.26190 1.37198 1.24939 0.29557 1.44667 1.29818 0.33138 1.55371 1.36648
0.38619
5.65005   36   47 0.91513 0.92687-0.00241 0.95339 0.95578 0.02810 0.98433 0.97862
0.05362 1.01808 1.00366 0.07836 1.04199 1.02278 0.09307 1.08787 1.05539 0.12678
1.13317 1.08727 0.15650 1.18627 1.12475 0.18984 1.23787 1.15986 0.22060 1.29959
1.20169 0.25532 1.37686 1.25294 0.29590 1.45083 1.30118 0.33349
5.90805   38   48 0.95008 0.95417 0.02313 0.97898 0.97598 0.04412 1.01352 1.00266
0.07194 1.02904 1.01430 0.08063 1.07518 1.04720 0.11608 1.12114 1.08023 0.14648
1.16683 1.11191 0.17449 1.22289 1.15057 0.20714 1.28059 1.18970 0.24048 1.35399
1.23850 0.27982 1.42161 1.28272 0.31521
6.14689   39   49 0.95139 0.95625 0.02074 0.97900 0.97814 0.04173 1.01233 1.00251
0.06916 1.02770 1.01395 0.07758 1.07758 1.05008 0.11332 1.12138 1.08080 0.14188
1.16286 1.10992 0.16900 1.22241 1.15082 0.20558 1.27903 1.18895 0.23666 1.34431
1.23248 0.27338 1.41201 1.27701 0.30715
6.38826   40   50 0.94761 0.95537 0.01554 0.97973 0.97930 0.04020 1.00187 0.99568
0.05518 1.02717 1.01470 0.07232 1.07248 1.04690 0.10689 1.11672 1.07785 0.13580
1.16016 1.10843 0.16299 1.21748 1.14775 0.19953 1.26834 1.18210 0.22700 1.33185
1.22456 0.26447 1.40144 1.27033 0.30030
6.63139   41   51 0.95026 0.95817 0.01242 0.98155 0.98122 0.03657 1.00882 1.00172
0.05843 1.02701 1.01500 0.07061 1.07166 1.04660 0.10235 1.11318 1.07615 0.13205
1.15570 1.10582 0.15869 1.21299 1.14500 0.19363 1.26114 1.17775 0.22178 1.32933
1.22328 0.25934 1.39370 1.26562 0.29231
6.89947   42   52 0.94104 0.95209 0.00156 0.98119 0.98190 0.03483 1.00898 1.00219
0.05513 1.02589 1.01448 0.06780 1.07223 1.04770 0.10121 1.11284 1.07637 0.12827
1.15179 1.10342 0.15370 1.20762 1.14179 0.18798 1.25868 1.17646 0.21882 1.32043
1.21771 0.25307 1.38268 1.25873 0.28587
7.17037   43   53 0.93997 0.95226-0.00313 0.97934 0.98107 0.03117 1.00104 0.99686
0.04756 1.02617 1.01535 0.06404 1.06764 1.04494 0.09512 1.10913 1.07407 0.12207
1.14483 1.09905 0.14618 1.19713 1.13505 0.17915 1.24920 1.17034 0.20928 1.30942
1.21060 0.24408 1.37029 1.25088 0.27679
7.42015   44   54 0.94375 0.95547-0.00088 0.98048 0.98219 0.02868 1.00764 1.00220
0.05078 1.02997 1.01837 0.06485 1.06962 1.04660 0.09367 1.10778 1.07340 0.12018
1.14534 1.09971 0.14451 1.19903 1.13657 0.17833 1.24736 1.16931 0.20745 1.30235
1.20623 0.23870 1.36566 1.24796 0.27257
7.69502   45   55 0.94223 0.95467-0.00474 0.98088 0.98297 0.02685 1.00909 1.00356
0.04700 1.02885 1.01794 0.06252 1.06688 1.04493 0.09123 1.10491 1.07183 0.11661
```

```
 1.14176 1.09750 0.14041 1.19407 1.13340 0.17369 1.24134 1.16559 0.20228 1.29891
 1.20404 0.23496 1.35734 1.24270 0.26612
 7.97134   46   56 0.94757 0.95901-0.00398 0.98338 0.98524 0.02650 1.01134 1.00553
 0.04881 1.02865 1.01805 0.06006 1.06845 1.04645 0.08916 1.10710 1.07358 0.11600
 1.14524 1.10003 0.14055 1.18979 1.13068 0.16896 1.24154 1.16579 0.20022 1.29251
 1.19998 0.22910 1.35252 1.23962 0.26156
 8.25063   47   56 0.95731 0.96651 0.00370 0.98401 0.98605 0.02464 1.01133 1.00578
 0.04642 1.02890 1.01863 0.05771 1.06915 1.04717 0.08795 1.10523 1.07244 0.11208
 1.14128 1.09755 0.13715 1.18735 1.12921 0.16598 1.23614 1.16235 0.19529 1.29611
 1.20246 0.23009
 8.55510   48   57 0.94976 0.96141-0.00541 0.98251 0.98519 0.02163 1.01164 1.00637
 0.04375 1.02770 1.01789 0.05574 1.06743 1.04610 0.08492 1.10218 1.07055 0.10892
 1.13807 1.09552 0.13298 1.18304 1.12646 0.16130 1.23790 1.16362 0.19520 1.28907
 1.19795 0.22445
 8.83750   49   58 0.95359 0.96439-0.00392 0.98469 0.98715 0.02165 1.01132 1.00627
 0.04194 1.02728 1.01784 0.05352 1.06563 1.04507 0.08250 1.10565 1.07314 0.11002
 1.14051 1.09737 0.13215 1.18336 1.12676 0.15970 1.23252 1.16017 0.19085 1.28906
 1.19804 0.22300
 9.14577   50   59 0.96114 0.97023 0.00007 0.98467 0.98726 0.01966 1.01320 1.00785
 0.04255 1.03083 1.02057 0.05433 1.07018 1.04843 0.08294 1.10485 1.07267 0.10800
 1.13807 1.09578 0.12953 1.17877 1.12378 0.15585 1.22997 1.15855 0.18695 1.28248
 1.19377 0.21782
 9.43191   50   60 0.93864 0.95395-0.02074 0.96494 0.97309 0.00237 0.98833 0.99012
 0.02173 1.01347 1.00826 0.04041 1.03339 1.02256 0.05479 1.06934 1.04794 0.08122
 1.10732 1.07455 0.10785 1.13545 1.09411 0.12611 1.18648 1.12912 0.15890 1.22856
 1.15770 0.18397 1.28328 1.19437 0.21659
 9.74420   51   61 0.94326 0.95741-0.01764 0.96853 0.97591 0.00347 0.99038 0.99181
 0.02075 1.01690 1.01088 0.04177 1.03643 1.02480 0.05610 1.07029 1.04876 0.08059
 1.10523 1.07323 0.10512 1.14055 1.09772 0.12780 1.18597 1.12884 0.15780 1.22865
 1.15782 0.18321 1.27866 1.19140 0.21254
10.05885   52   61 0.94493 0.95887-0.01818 0.96954 0.97685 0.00245 0.99130 0.99262
 0.02089 1.01763 1.01148 0.04106 1.04124 1.02834 0.05818 1.06837 1.04754 0.07847
 1.10694 1.07451 0.10452 1.14703 1.10224 0.13259 1.18798 1.13030 0.15767 1.23550
 1.16249 0.18696
10.37472   53   62 0.94622 0.96001-0.01840 0.96966 0.97710 0.00223 0.99176 0.99304
 0.01953 1.01783 1.01177 0.03975 1.04035 1.02784 0.05627 1.06947 1.04840 0.07749
 1.10656 1.07431 0.10357 1.15086 1.10496 0.13381 1.18944 1.13136 0.15756 1.23459
 1.16197 0.18551
10.69173   54   63 0.94978 0.96275-0.01656 0.97695 0.98243 0.00619 0.99602 0.99623
 0.02100 1.01808 1.01207 0.03901 1.04860 1.03374 0.06148 1.07770 1.05422 0.08290
 1.11034 1.07703 0.10457 1.15005 1.10447 0.13145 1.19318 1.13399 0.15889 1.23625
 1.16312 0.18538
11.03443   55   64 0.95540 0.96690-0.01393 0.98005 0.98480 0.00797 0.99583 0.99622
 0.02014 1.01916 1.01293 0.03824 1.04941 1.03438 0.06078 1.08026 1.05611 0.08326
 1.10858 1.07587 0.10244 1.15044 1.10482 0.13121 1.19225 1.13339 0.15730 1.23962
 1.16542 0.18647
11.35467   56   65 0.95762 0.96864-0.01243 0.98113 0.98569 0.00736 1.00334 1.00167
 0.02505 1.02218 1.01514 0.03925 1.05145 1.03591 0.06141 1.08362 1.05851 0.08448
 1.11374 1.07952 0.10474 1.15045 1.10486 0.12945 1.19608 1.13603 0.15841 1.24406
 1.16845 0.18785
11.70090   57   65 0.95846 0.96937-0.01254 0.98498 0.98851 0.00936 1.00600 1.00363
 0.02661 1.02236 1.01535 0.03840 1.05186 1.03623 0.06114 1.08760 1.06136 0.08640
 1.11550 1.08078 0.10541 1.15299 1.10664 0.13001 1.19640 1.13630 0.15742
12.02388   58   66 0.96374 0.97324-0.00927 0.98833 0.99101 0.01085 1.01019 1.00670
 0.02892 1.02911 1.02016 0.04263 1.05835 1.04090 0.06490 1.08985 1.06297 0.08739
 1.12077 1.08446 0.10790 1.16042 1.11179 0.13428 1.19806 1.13745 0.15763
12.37319   59   67 0.96760 0.97611-0.00669 0.98834 0.99110 0.00966 1.01075 1.00714
 0.02836 1.03282 1.02288 0.04483 1.05910 1.04146 0.06411 1.09198 1.06448 0.08741
 1.13312 1.09303 0.11600 1.16692 1.11625 0.13797 1.20686 1.14346 0.16242
12.72323   59   68 0.95194 0.96480-0.02182 0.97255 0.97977-0.00368 0.99209 0.99381
```

```
 0.01237 1.01515 1.01035 0.03017 1.04005 1.02804 0.04926 1.06603 1.04635 0.06843
 1.09690 1.06796 0.09036 1.13247 1.09261 0.11444 1.17044 1.11869 0.13948 1.20957
 1.14533 0.16342
13.09811   60   68 0.95569 0.96761-0.01929 0.97543 0.98187-0.00257 0.99573 0.99651
 0.01399 1.01584 1.01089 0.03000 1.04333 1.03039 0.05108 1.07178 1.05043 0.07216
 1.09761 1.06848 0.08995 1.13413 1.09379 0.11457 1.17623 1.12269 0.14237
13.44798   61   69 0.95872 0.96982-0.01773 0.98337 0.98766 0.00335 1.00150 1.00068
 0.01798 1.01959 1.01359 0.03237 1.04698 1.03300 0.05296 1.07774 1.05463 0.07573
 1.10455 1.07334 0.09361 1.13635 1.09534 0.11521 1.17559 1.12228 0.14062
13.82259   62   70 0.96433 0.97396-0.01391 0.98486 0.98877 0.00402 1.00575 1.00374
 0.02070 1.02360 1.01647 0.03472 1.05145 1.03618 0.05568 1.07808 1.05490 0.07491
 1.10732 1.07528 0.09510 1.14475 1.10117 0.12056 1.18146 1.12630 0.14401
14.19732   63   71 0.96550 0.97485-0.01361 0.98849 0.99141 0.00622 1.00931 1.00632
 0.02280 1.02760 1.01934 0.03689 1.05318 1.03744 0.05603 1.08161 1.05739 0.07624
 1.11856 1.08311 0.10291 1.15181 1.10604 0.12471 1.18735 1.13032 0.14748
14.57202   64   71 0.97219 0.97970-0.00853 0.99241 0.99425 0.00837 1.01139 1.00783
 0.02365 1.03312 1.02329 0.04029 1.05414 1.03814 0.05598 1.08764 1.06164 0.08026
 1.11990 1.08406 0.10247 1.15451 1.10791 0.12554
14.94663   64   72 0.95833 0.96970-0.02140 0.97676 0.98303-0.00531 0.99519 0.99627
 0.01001 1.01449 1.01007 0.02525 1.03908 1.02752 0.04463 1.06514 1.04591 0.06369
 1.09255 1.06508 0.08318 1.12301 1.08623 0.10403 1.16133 1.11260 0.12935
15.32124   65   73 0.96318 0.97324-0.01763 0.98371 0.98805 0.00009 1.00124 1.00063
 0.01470 1.01788 1.01250 0.02754 1.04353 1.03070 0.04737 1.07195 1.05070 0.06805
 1.09641 1.06779 0.08519 1.13413 1.09392 0.11111 1.16627 1.11601 0.13171
15.72073   66   73 0.96696 0.97600-0.01478 0.98658 0.99014 0.00192 1.00517 1.00346
 0.01703 1.02263 1.01592 0.03072 1.04778 1.03373 0.04982 1.07399 1.05215 0.06890
 1.10716 1.07528 0.09222 1.13989 1.09791 0.11459
16.12009   67   74 0.97027 0.97842-0.01249 0.99097 0.99331 0.00487 1.00901 1.00623
 0.01967 1.03043 1.02147 0.03628 1.05017 1.03543 0.05072 1.08186 1.05767 0.07389
 1.11332 1.07957 0.09634 1.14567 1.10190 0.11789
16.51944   68   75 0.97614 0.98267-0.00798 0.99468 0.99600 0.00743 1.01335 1.00935
 0.02228 1.03490 1.02465 0.03911 1.06264 1.04421 0.06014 1.08572 1.06038 0.07634
 1.11560 1.08116 0.09721 1.15052 1.10524 0.12048
16.94380   68   75 0.96257 0.97289-0.02047 0.98027 0.98568-0.00500 0.99745 0.99801
 0.00897 1.01622 1.01140 0.02426 1.03912 1.02765 0.04204 1.06601 1.04659 0.06195
 1.08922 1.06284 0.07824 1.12417 1.08710 0.10262
17.34319   69   76 0.96691 0.97606-0.01709 0.98684 0.99041-0.00024 1.00434 1.00294
 0.01453 1.02094 1.01477 0.02746 1.04425 1.03128 0.04524 1.06961 1.04913 0.06405
 1.10194 1.07169 0.08707 1.13070 1.09162 0.10656
17.76732   70   77 0.97148 0.97938-0.01361 0.98992 0.99263 0.00222 1.00832 1.00580
 0.01726 1.02728 1.01928 0.03197 1.05100 1.03606 0.04991 1.07575 1.05344 0.06797
 1.10575 1.07436 0.08926 1.14028 1.09822 0.11267
18.19151   71   77 0.97704 0.98340-0.00926 0.99507 0.99634 0.00599 1.01231 1.00865
 0.01977 1.03373 1.02387 0.03666 1.05851 1.04136 0.05543 1.08263 1.05826 0.07268
 1.11472 1.08058 0.09533
18.59067   71   78 0.96573 0.97525-0.01950 0.98552 0.98950-0.00231 1.00130 1.00080
 0.01056 1.01790 1.01264 0.02389 1.04073 1.02883 0.04161 1.06459 1.04564 0.05951
 1.09369 1.06598 0.08048 1.12307 1.08637 0.10038
19.03976   72   79 0.97027 0.97854-0.01628 0.98800 0.99129-0.00071 1.00446 1.00308
 0.01297 1.02188 1.01548 0.02658 1.04327 1.03063 0.04308 1.07031 1.04965 0.06304
 1.09823 1.06914 0.08296 1.12894 1.09042 0.10413
19.46384   73   79 0.97480 0.98181-0.01225 0.99203 0.99420 0.00233 1.01001 1.00704
 0.01700 1.03036 1.02151 0.03315 1.05494 1.03886 0.05168 1.07796 1.05502 0.06817
 1.10444 1.07347 0.08663
19.91280   74   80 0.97970 0.98535-0.00832 0.99619 0.99718 0.00548 1.01434 1.01014
 0.01987 1.03530 1.02501 0.03640 1.05842 1.04132 0.05372 1.08204 1.05787 0.07095
 1.11317 1.07953 0.09280
20.36180   74   81 0.96840 0.97722-0.01867 0.98672 0.99040-0.00292 1.00231 1.00157
 0.01003 1.01838 1.01302 0.02286 1.03940 1.02791 0.03911 1.06383 1.04513 0.05736
 1.09212 1.06491 0.07784 1.12168 1.08542 0.09819
```

```
20.81081   75   81 0.97409 0.98133-0.01406 0.99136 0.99374 0.00062 1.00742 1.00522
 0.01384 1.02693 1.01909 0.02943 1.04897 1.03467 0.04628 1.07065 1.04991 0.06215
 1.09747 1.06864 0.08114
21.25983   76   82 0.97889 0.98479-0.01028 0.99570 0.99685 0.00411 1.01160 1.00820
 0.01687 1.03257 1.02309 0.03340 1.05509 1.03899 0.05032 1.07797 1.05504 0.06702
 1.10758 1.07566 0.08796
21.73363   76   83 0.97003 0.97842-0.01827 0.98421 0.98862-0.00592 1.00109 1.00071
 0.00820 1.01632 1.01156 0.02037 1.03685 1.02613 0.03626 1.06062 1.04288 0.05406
 1.08737 1.06161 0.07377 1.11372 1.07993 0.09187
22.18251   77   83 0.97516 0.98212-0.01413 0.99143 0.99380-0.00001 1.00657 1.00463
 0.01231 1.02352 1.01669 0.02574 1.04724 1.03347 0.04430 1.06774 1.04789 0.05923
 1.09397 1.06621 0.07799
22.65633   78   84 0.97937 0.98515-0.01085 0.99546 0.99669 0.00289 1.01051 1.00744
 0.01519 1.03087 1.02190 0.03134 1.05255 1.03722 0.04777 1.07860 1.05549 0.06713
 1.10485 1.07378 0.08555
23.13013   79   84 0.98417 0.98860-0.00680 0.99866 0.99899 0.00547 1.01571 1.01114
 0.01910 1.03500 1.02484 0.03410 1.05782 1.04093 0.05136 1.08400 1.05927 0.07062
23.62889   79   85 0.97442 0.98160-0.01546 0.98994 0.99275-0.00218 1.00561 1.00395
 0.01091 1.02224 1.01579 0.02411 1.04407 1.03124 0.04108 1.06438 1.04553 0.05602
 1.09325 1.06572 0.07700
24.10268   80   86 0.97832 0.98441-0.01224 0.99513 0.99647 0.00200 1.00991 1.00703
 0.01413 1.02925 1 02077 0.02948 1.05014 1.03553 0.04533 1.07560 1.05340 0.06411
 1.10085 1.07101 0.08197
24.60136   81   86 0.98502 0.98923-0.00677 0.99879 0.99909 0.00480 1.01489 1.01057
 0.01782 1.03413 1.02423 0.03286 1.05705 1.04039 0.05025 1.08162 1.05762 0.06813
25.10001   81   87 0.97528 0.98223-0.01550 0.99039 0.99308-0.00232 1.00533 1.00377
 0.01003 1.02300 1.01634 0.02418 1.04446 1.03153 0.04084 1.06403 1.04530 0.05533
 1.09099 1.06416 0.07486
25.59868   82   87 0.97954 0.98530-0.01204 0.99546 0.99671 0.00152 1.00967 1.00686
 0.01313 1.02884 1.02049 0.02857 1.05065 1.03589 0.04517 1.07450 1.05264 0.06280
26.09747   83   88 0.98582 0.98981-0.00672 1.00099 1.00067 0.00601 1.01654 1.01175
 0.01863 1.03371 1.02394 0.03202 1.05606 1.03970 0.04895 1.08046 1.05681 0.06673
26.62111   83   89 0.97715 0.98359-0.01450 0.99160 0.99396-0.00190 1.00608 1.00431
 0.01004 1.02426 1.01724 0.02465 1.04292 1.03044 0.03902 1.06550 1.04633 0.05597
 1.09073 1.06397 0.07403
27.11989   84   89 0.98059 0.98606-0.01160 0.99661 0.99754 0.00206 1.01111 1.00790
 0.01394 1.02975 1.02114 0.02873 1.05036 1.03569 0.04439 1.07381 1.05216 0.06170
27.64343   85   90 0.98943 0.99241-0.00407 1.00328 1.00231 0.00737 1.01832 1.01303
 0.01959 1.03855 1.02736 0.03548 1.05877 1.04161 0.05057 1.08410 1.05935 0.06910
28.16702   85   90 0.98007 0.98569-0.01249 0.99446 0.99601 0.00002 1.00806 1.00572
 0.01120 1.02557 1.01817 0.02523 1.04582 1.03250 0.04087 1.06880 1.04865 0.05789
28.69064   86   91 0.98432 0.98875-0.00884 0.99834 0.99879 0.00308 1.01299 1.00924
 0.01500 1.03079 1.02188 0.02894 1.05227 1.03704 0.04539 1.07548 1.05333 0.06254
29.23921   86   91 0.97720 0.98363-0.01520 0.99041 0.99312-0.00367 1.00514 1.00364
 0.00850 1.02153 1.01531 0.02174 1.03961 1.02812 0.03577 1.06067 1.04295 0.05150
29.78786   87   92 0.98148 0.98671-0.01163 0.99572 0.99692 0.00055 1.01004 1.00714
 0.01239 1.02778 1.01975 0.02647 1.04669 1.03311 0.04099 1.07001 1.04950 0.05838
30.31155   88   93 0.98740 0.99096-0.00663 1.00165 1.00116 0.00534 1.01693 1.01204
 0.01773 1.03564 1.02532 0.03252 1.05558 1.03937 0.04753 1.07936 1.05605 0.06497
30.86008   88   93 0.97961 0.98537-0.01361 0.99367 0.99546-0.00141 1.00744 1.00529
 0.00999 1.02421 1.01722 0.02343 1.04399 1.03121 0.03880 1.06495 1.04596 0.05437
31.43355   89   94 0.98462 0.98897-0.00927 0.99850 0.99891 0.00249 1.01259 1.00896
 0.01409 1.03150 1.02239 0.02909 1.05017 1.03557 0.04322 1.07495 1.05296 0.06162
31.98219   90   94 0.99121 0.99370-0.00374 1.00469 1.00333 0.00758 1.02087 1.01484
 0.02060 1.03918 1.02782 0.03487 1.06013 1.04258 0.05064
32.55566   90   95 0.98280 0.98766-0.01108 0.99629 0.99733 0.00040 1.01100 1.00783
 0.01259 1.02643 1.01879 0.02477 1.04588 1.03255 0.03980 1.06851 1.04846 0.05663
33.12914   91   95 0.98838 0.99167-0.00639 1.00278 1.00197 0.00567 1.01652 1.01176
 0.01692 1.03528 1.02506 0.03165 1.05487 1.03888 0.04646
33.70273   91   96 0.98120 0.98652-0.01278 0.99478 0.99625-0.00110 1.00741 1.00527
```

```
 0.00935 1.02456 1.01747 0.02317 1.04295 1.03048 0.03735 1.06448 1.04564 0.05345
34.27623   92   96 0.98630 0.99018-0.00835 0.99908 0.99933 0.00245 1.01361 1.00969
 0.01425 1.03261 1.02317 0.02942 1.05148 1.03650 0.04376
34.87469   93   97 0.99289 0.99490-0.00283 1.00543 1.00386 0.00758 1.02173 1.01546
 0.02074 1.04028 1.02860 0.03525 1.06056 1.04289 0.05046
35.47318   93   97 0.98472 0.98905-0.01002 0.99777 0.99839 0.00118 1.01184 1.00843
 0.01271 1.02952 1.02099 0.02679 1.04780 1.03390 0.04079
36.04662   94   98 0.99181 0.99413-0.00399 1.00444 1.00316 0.00666 1.01963 1.01397
 0.01890 1.03710 1.02635 0.03255 1.05754 1.04076 0.04801
36.64517   94   99 0.98436 0.98879-0.01048 0.99667 0.99761 0.00004 1.01017 1.00725
 0.01117 1.02515 1.01789 0.02305 1.04587 1.03255 0.03921 1.06508 1.04606 0.05359
37.26866   95   99 0.98963 0.99257-0.00608 1.00292 1.00208 0.00518 1.01740 1.01239
 0.01691 1.03451 1.02452 0.03037 1.05147 1.03649 0.04385
37.86717   95   99 0.98314 0.98792-0.01173 0.99583 0.99701-0.00087 1.00822 1.00586
 0.00947 1.02345 1.01668 0.02163 1.03924 1.02786 0.03500
38.49072   96   99 0.98844 0.99172-0.00725 1.00221 1.00157 0.00444 1.01535 1.01093
 0.01587 1.02975 1.02116 0.02815
39.11417   97   99 0.99403 0.99572-0.00241 1.00576 1.00411 0.00857 1.02001 1.01425
 0.02079
39.73759   97   99 0.98650 0.99033-0.00826 0.99858 0.99897 0.00301 1.01021 1.00727
 0.01312
40.36098   98   99 0.99266 0.99474-0.00166 1.00264 1.00188 0.00707
41.00935   98   99 0.98519 0.98939-0.00785 0.99639 0.99741 0.00194
41.63287   99   99 0.98981 0.99271-0.00366
42.28127   99   99 0.98353 0.98820-0.00920
9999.       1    1 1.00000 0.50000 3.08999
 0.52234 0.54489 0.57555 0.60718 0.63173 0.67393 0.70912 0.74535 0.79198 0.82057
 0.86922 0.90914 0.98149 1.03532 1.10196 1.14748 1.20503 1.28763 1.34939 1.43865
 1.51728 1.59726 1.67900 1.77676 1.87808 1.98223 2.07266 2.19551 2.30489 2.43385
 2.53264 2.70096 2.83817 2.97742 3.11868 3.27956 3.42474 3.61128 3.78183 3.95475
 4.17005 4.34858 4.54881 4.75172 4.97784 5.20659 5.43760 5.69283 5.92970 6.21256
 6.47645 6.76503 7.03476 7.32912 7.64910 7.97134 8.27397 8.62557 8.97948 9.33643
 9.7200610.1073610.4965210.9118711.3546711.7752312.2482212.6982313.1980613.69776
14.1723314.7468115.2962615.8455616.4195917.0436717.6425618.2663618.9150519.58859
20.2869621.0103621.7086922.4817523.2548224.0528124.8506825.6984126.5463027.46892
28.3914629.3140430.2866231.2839432.3312833.4034734.4757235.6477436.64517
10.5319010.33957 9.59698 8.95894 8.33144 7.74382 7.31747 7.00768 6.51864 6.42672
 6.04401 5.82610 5.26311 4.95248 4.60751 4.53878 4.38771 3.95143 3.83531 3.55923
 3.42251 3.30253 3.19363 2.97723 2.80996 2.72184 2.68023 2.56088 2.47262 2.32693
 2.30259 2.17447 2.13454 2.06999 2.03311 1.98430 1.93115 1.85341 1.81542 1.77748
 1.71494 1.69158 1.66922 1.62972 1.59629 1.57644 1.55371 1.52073 1.49396 1.46919
 1.45389 1.42831 1.40665 1.39451 1.37450 1.35252 1.34199 1.32982 1.31315 1.30237
 1.28186 1.27274 1.26507 1.25720 1.24406 1.23706 1.22368 1.21292 1.20455 1.19504
 1.18914 1.18013 1.16895 1.16496 1.15864 1.15008 1.14563 1.14217 1.13520 1.13136
 1.12743 1.11984 1.11616 1.11102 1.10628 1.10220 1.09922 1.09608 1.09307 1.09005
 1.08532 1.08334 1.08051 1.07753 1.07538 1.07206 1.07004 1.06727 1.06508
 3.85407 3.83928 3.64037 3.64535 3.46118 3.44163 3.36122 3.29944 3.13337 3.17528
 3.08330 3.03059 2.88712 2.79740 2.67689 2.67333 2.62720 2.44873 2.43896 2.33100
 2.28288 2.22875 2.19194 2.10014 2.02063 1.98628 1.96710 1.91101 1.86522 1.79461
 1.78413 1.71695 1.69637 1.66186 1.64219 1.61620 1.58686 1.54321 1.52153 1.49973
 1.46310 1.44950 1.43624 1.41262 1.39223 1.38038 1.36648 1.34617 1.32955 1.31407
 1.30445 1.28828 1.27450 1.26674 1.25391 1.23962 1.23289 1.22498 1.21410 1.20703
 1.19353 1.18751 1.18243 1.17720 1.16845 1.16378 1.15482 1.14759 1.14195 1.13554
 1.13154 1.12544 1.11783 1.11512 1.11080 1.10495 1.10190 1.09953 1.09474 1.09210
 1.08939 1.08416 1.08162 1.07806 1.07478 1.07195 1.06988 1.06770 1.06561 1.06350
 1.06021 1.05883 1.05685 1.05477 1.05327 1.05095 1.04953 1.04759 1.04606
 2.68283 2.65199 2.53167 2.42391 2.35879 2.20531 2.12367 2.06247 1.96230 1.94279
 1.86030 1.81180 1.68112 1.60506 1.51681 1.49855 1.45799 1.33570 1.30153 1.21745
 1.17403 1.13489 1.09587 1.02358 0.96291 0.92979 0.91384 0.86724 0.83169 0.77097
 0.75885 0.70414 0.68601 0.65621 0.63882 0.61544 0.58950 0.55056 0.53003 0.51119
```

```
 0.47779 0.46504 0.45270 0.43061 0.41088 0.40009 0.38619 0.36671 0.35118 0.33555
 0.32612 0.31065 0.29694 0.28877 0.27589 0.26156 0.25462 0.24654 0.23569 0.22811
 0.21413 0.20785 0.20254 0.19728 0.18785 0.18291 0.17341 0.16588 0.15983 0.15292
 0.14847 0.14187 0.13361 0.13065 0.12594 0.11952 0.11617 0.11356 0.10828 0.10543
 0.10242 0.09654 0.09371 0.08974 0.08606 0.08289 0.08057 0.07811 0.07576 0.07342
 0.06966 0.06809 0.06585 0.06349 0.06178 0.05914 0.05753 0.05531 0.05359
 0.53736 0.57555 0.62345 0.67393 0.70912 0.79198 0.85935 0.92945 1.05731 1.17037
 1.37438 1.59726 2.11832 2.21105 2.32073 2.43385 2.53264 2.66694 2.76940 2.90746
 3.04789 3.18992 3.33376 3.48010 3.63011 3.81999 3.97407 4.17005 4.32865 4.50861
 4.67010 4.89534 5.10236 5.26930 5.50112 5.73565 5.86478 6.12512 6.36624 6.60922
 6.87695 7.10252 7.37457 7.64910 7.92508 8.25063 8.53163 8.83750 9.14577 9.47976
 9.8166510.1559010.4721510.8384411.1820611.5771211.9492112.3731912.7732113.22307
13.6228814.0723914.5470415.0215415.4960415.9703816.4944817.0187117.5427718.11667
18.6655319.2642519.8629020.4615521.1101521.7585822.4318723.1051823.8034424.50163
25.2496026.0226726.7956827.5935828.4164029.2641430.1120831.0096731.9323332.90472
33.8523434.8746935.8721236.9195338.0168339.1141740.2612341.4582842.50571
 9.15563 7.84434 6.83633 5.74380 5.34088 4.16061 3.48754 2.94023 2.28255 1.87683
 1.47977 1.20674 0.94059 0.94285 0.89090 0.89441 0.86279 0.86289 0.87160 0.86584
 0.87189 0.85829 0.85860 0.85928 0.86644 0.86045 0.86174 0.86578 0.86797 0.86984
 0.87390 0.87779 0.88313 0.88176 0.88529 0.88985 0.91256 0.91576 0.91801 0.91845
 0.92105 0.92257 0.92433 0.92660 0.92861 0.93047 0.93231 0.93409 0.93573 0.93756
 0.93934 0.94085 0.94239 0.94412 0.94545 0.94682 0.94850 0.95002 0.95138 0.95266
 0.95397 0.95509 0.95628 0.95773 0.95890 0.95972 0.96098 0.96203 0.96319 0.96395
 0.96504 0.96611 0.96697 0.96781 0.96877 0.96960 0.97044 0.97126 0.97199 0.97275
 0.97352 0.97424 0.97499 0.97557 0.97622 0.97684 0.97751 0.97815 0.97872 0.97927
 0.97985 0.98040 0.98093 0.98142 0.98193 0.98245 0.98215 0.98182 0.98214
 3.48728 3.13763 2.84420 2.51169 2.38460 1.99514 1.75892 1.55705 1.29950 1.13000
 0.95357 0.82430 0.68991 0.69110 0.66674 0 73674 0.71804 0.73932 0.74465 0.77666
 0.79954 0.80610 0.81279 0.81550 0.82845 0.84793 0.85375 0.86261 0.87080 0.87715
 0.88348 0.88977 0.89577 0.89735 0.90358 0.90851 0.92636 0.92943 0.93216 0.93428
 0.93708 0.93870 0.94091 0.94304 0.94479 0.94662 0.94841 0.95005 0.95146 0.95317
 0.95469 0.95591 0.95722 0.95865 0.95970 0.96083 0.96217 0.96333 0.96440 0.96542
 0.96639 0.96728 0.96819 0.96927 0.97016 0.97077 0.97172 0.97250 0.97338 0.97395
 0.97475 0.97554 0.97617 0.97680 0.97750 0.97811 0.97872 0.97932 0.97985 0.98040
 0.98096 0.98149 0.98203 0.98245 0.98293 0.98337 0.98386 0.98432 0.98473 0.98513
 0.98555 0.98595 0.98633 0.98668 0.98705 0.98742 0.98721 0.98697 0.98720
 2.55715 2.38984 2.19075 1.99902 1.87675 1.61709 1.43224 1.26257 1.00298 0.81824
 0.56677 0.37946 0.16422 0.15368 0.09597 0.08709 0.06106 0.05727 0.04576 0.04889
 0.04015 0.01625 0.01513 0.01367 0.00595-0.00791-0.00875-0.01208-0.01361-0.01458
-0.01773-0.02028-0.02407-0.02174-0.02369-0.02647-0.00891-0.01123-0.01263-0.01236
-0.01656-0.01714-0.01773-0.01878-0.01955-0.01996-0.02054-0.02097-0.02127-0.02165
-0.02201-0.02211-0.02234-0.02261-0.02261-0.02251-0.02278-0.02278-0.02275-0.02253
-0.02252-0.02224-0.02200-0.02203-0.02185-0.02141-0.02129-0.02102-0.02089-0.02036
-0.02021-0.01999-0.01963-0.01931-0.01904-0.01871-0.01840-0.01811-0.01775-0.01746
-0.01716-0.01682-0.01656-0.01615-0.01583-0.01549-0.01525-0.01496-0.01463-0.01427
-0.01401-0.01370-0.01342-0.01312-0.01282-0.01257-0.01163-0.01072-0.01049
```

Appendix C

DATA BASE FILE 9 (SURFACE BURST TNT)

What follows is a listing of UVic/AIRBLAST data base file number 9 (surface burst TNT reconstruction). This data file is being supplied as file 'SBT DATA' on magnetic tape.

```
SURFACE BURST TNT                                             /airblast data   31Dec86
99 3                                                                      RECOGCNJI77J
 1.03000 1.06000 1.09000 1.12000 1.15000 1.18000 1.21000 1.24000 1.28000 1.31000
 1.35000 1.39000 1.43000 1.47000 1.51000 1.55000 1.59000 1.64000 1.68000 1.73000
 1.78000 1.83000 1.88000 1.93000 1.98000 2.04000 2.09000 2.15000 2.21000 2.27000
 2.33000 2.40000 2.47000 2.53000 2.60000 2.68000 2.75000 2.83000 2.91000 2.99000
 3.07000 3.15000 3.24000 3.33000 3.42000 3.52000 3.62000 3.72000 3.82000 3.93000
 4.04000 4.15000 4.26000 4.38000 4.50000 4.63000 4.75000 4.89000 5.02000 5.16000
 5.30000 5.45000 5.60000 5.76000 5.92000 6.08000 6.25000 6.42000 6.60000 6.79000
 6.97000 7.17000 7.37000 7.57000 7.78000 8.00000 8.22000 8.45000 8.68000 8.92000
 9.17000 9.42000 9.68000 9.9500010.2300010.5100010.8000011.1000011.4100011.73000
12.0500012.3900012.7300013.0800013.4500013.8200014.2000014.6000014.95000
 0.57281    6    7 9.28127 3.04820 2.6619210.99442 3.92314 2.75044
 0.65680    7    9 6.10830 2.24827 2.16787 7.40588 2.57989 2.31839 8.70633 3.32111
 2.42663
 0.73775    9   12 5.48908 2.08313 1.94489 5.94702 2.37092 2.05449 7.37581 3.10936
 2.22228 8.42906 3.55770 2.32805
 0.81558   10   13 4.31197 1.75332 1.74304 5.06270 2.11347 1.87237 5.63462 2.49478
 1.94562 7.04597 3.13032 2.10297
 0.90720   11   15 3.38554 1.47518 1.45322 3.72218 1.58749 1.52284 4.72111 2.26105
 1.73651 5.48134 2.61647 1.85765 6.35592 3.05094 1.97067
 1.00304   12   17 2.74609 1.27033 1.18874 3.09484 1.39145 1.31726 3.61404 1.81678
 1.45909 4.20915 2.11928 1.58236 4.98749 2.53786 1.71479 5.78930 2.94088 1.83255
 1.10404   13   19 2.31172 1.12335 1.03541 2.63317 1.23984 1.15765 3.01902 1.59775
 1.24194 3.42663 1.82977 1.36802 4.02738 2.09957 1.50653 4.70404 2.53569 1.63083
 5.42433 2.93153 1.71762
 1.20017   13   20 1.89396 0.97428 0.82838 2.13136 1.06004 0.93358 2.40943 1.30442
 1.04633 2.61436 1.48254 1.12479 2.92722 1.67172 1.20711 3.48297 1.96388 1.33643
 3.98710 2.25327 1.43816 4.67265 2.63531 1.55525
 1.31181   14   22 1.71640 0.90814 0.70858 1.85052 0.95829 0.79755 2.05310 1.16353
 0.91298 2.34800 1.37303 1.00816 2.67361 1.56697 1.10628 2.92844 1.73512 1.15406
 3.40189 2.01180 1.28458 3.95128 2.33790 1.39761 4.56676 2.66799 1.50445.
 1.43975   14   23 1.42575 0.79543 0.54894 1.51268 0.82979 0.62143 1.64875 0.88747
 0.69937 1.79485 1.10210 0.77311 2.04901 1.26738 0.85767 2.28827 1.40214 0.95549
 2.51899 1.57540 1.02640 2.84383 1.78615 1.11283 3.31870 2.08096 1.23257 3.80088
 2.34018 1.33665
 1.56183   15   25 1.29682 0.74337 0.46795 1.39787 0.78431 0.53126 1.49581 0.92800
 0.59055 1.70200 1.09113 0.68689 1.78830 1.14999 0.74608 2.03233 1.28826 0.83079
 2.29848 1.49197 0.93587 2.58877 1.69003 1.00901 2.96740 1.94232 1.12543 3.26943
 2.10153 1.19202 3.74910 2.39264 1.28187
 1.67588   15   26 1.16677 0.68932 0.35904 1.22742 0.71475 0.40408 1.33152 0.76185
 0.45659 1.43961 0.94147 0.55580 1.52965 1.01452 0.59450 1.69414 1.13121 0.66998
 1.89040 1.26933 0.75685 2.10272 1.42681 0.84608 2.36044 1.59802 0.93554 2.65681
 1.80983 1.00685 3.01740 1.99796 1.09917 3.53643 2.30596 1.23114
 1.80521   15   27 1.07200 0.64885 0.28119 1.12365 0.67103 0.31227 1.15026 0.68236
 0.33972 1.25913 0.83479 0.42858 1.33439 0.91706 0.46566 1.45822 0.99403 0.54730
 1.59376 1.08293 0.61315 1.74305 1.22451 0.69020 1.94783 1.36314 0.76851 2.21499
 1.54636 0.86753 2.40163 1.68392 0.91662 2.69069 1.86248 1.00165 3.03732 2.07408
 1.08740
 1.93860   16   29 1.02948 0.63036 0.24129 1.06091 0.64407 0.27195 1.11603 0.75283
 0.30686 1.16932 0.81153 0.35149 1.26050 0.88355 0.41076 1.35542 0.96461 0.47930
 1.48590 1.06879 0.55341 1.65296 1.20148 0.61777 1.81812 1.31306 0.69346 1.94443
 1.43617 0.74453 2.24596 1.60722 0.85764 2.45950 1.75326 0.90958 2.81898 1.96943
 1.01244 3.09543 2.12375 1.07666
 2.07810   17   30 1.01214 0.62277 0.22220 1.04118 0.68306 0.25472 1.09599 0.77484
 0.29443 1.12796 0.81614 0.31559 1.23090 0.88071 0.39744 1.30492 0.93881 0.43158
 1.43099 1.06358 0.50221 1.54186 1.15357 0.55894 1.66132 1.24348 0.62030 1.87580
 1.41147 0.70607 2.02204 1.50117 0.75289 2.28870 1.68932 0.84695 2.56203 1.84390
```

```
0.93199 2.85585 2.02131 1.00227
2.22108   18   31 0.97175 0.60836 0.18841 1.01141 0.73165 0.22505 1.05425 0.77500
0.26453 1.11276 0.81947 0.30054 1.16141 0.86385 0.32727 1.24977 0.96554 0.38189
1.33249 1.03938 0.44073 1.45777 1.12142 0.51019 1.61573 1.25825 0.57209 1.76685
1.35243 0.64704 1.89144 1.45341 0.69403 2.14097 1.61690 0.77823 2.34272 1.73945
0.83709 2.58677 1.88906 0.90661
2.36843   20   33 0.99551 0.74391 0.20788 1.03320 0.77718 0.22935 1.08500 0.82286
0.26800 1.12042 0.87361 0.30022 1.21423 0.96189 0.36422 1.29118 1.01626 0.39279
1.42851 1.13047 0.47637 1.52456 1.21720 0.52626 1.65029 1.30678 0.58675 1.83018
1.44001 0.65762 2.05441 1.57488 0.74078 2.21952 1.67467 0.79366 2.48544 1.84248
0.87184 2.76363 2.00335 0.94464
2.53403   21   34 0.97792 0.74725 0.18279 1.00098 0.75980 0.21605 1.05430 0.80617
0.25214 1.10959 0.88688 0.28031 1.16652 0.94517 0.31470 1.25354 1.01689 0.37192
1.35131 1.11530 0.43814 1.44197 1.17912 0.47038 1.57756 1.28939 0.53231 1.74642
1.39693 0.60626 1.87528 1.47921 0.65325 2.11808 1.63773 0 74631 2.29772 1.75107
0.80011 2.49247 1.86565 0.85364
2.68768   23   35 0.98814 0.76970 0 18447 1.03256 0.84246 0.21573 1.07771 0.89319
0.25004 1.12657 0.93287 0.29198 1.20679 1.01038 0.33835 1.29721 1.08461 0.38328
1.40530 1.16507 0.43969 1.54344 1.27551 0.50553 1.62642 1.33286 0.54267 1.80678
1.44670 0.61917 1.99746 1.57618 0.69326 2.15637 1.67711 0.74320 2.40230 1.81993
0.82017
2.84659   24   36 0.97599 0.80922 0.17042 1.00390 0.84906 0.19887 1.06461 0.88543
0.23836 1.09758 0.93651 0.25380 1.19699 1.02407 0.31783 1.25889 1.07703 0.35266
1.36906 1.17036 0.41511 1.47194 1.23727 0.46250 1.60569 1.32896 0.52409 1.76085
1.43979 0.59702 1.88915 1.52226 0.64007 2.07872 1.63880 0.70457 2.27874 1.75817
0.76584
3.02508   26   37 0.99895 0.84607 0.17453 1.03835 0.88890 0.20389 1.09105 0.95847
0.24011 1.15201 1.01089 0.27499 1.22765 1.06736 0.32173 1.33336 1.14895 0.38327
1.40505 1.20360 0.42309 1.53467 1.30075 0.48294 1.67054 1.38729 0.54260 1.80979
1.47984 0.59242 1.99457 1.59402 0.66307 2.16582 1.70155 0.71918
3.19066   27   38 0.98586 0.85657 0.15707 1.02491 0.91544 0.18574 1.08525 0.96250
0.22735 1.12645 1.00002 0.25314 1.19911 1.06508 0.30263 1.30972 1.14185 0.36469
1.39051 1.19991 0.40542 1.50578 1.28754 0.45846 1.59400 1.34859 0.49416 1.76677
1.45935 0.56617 1.91325 1.55180 0.62214 2.09705 1.66585 0.68815
3.36011   28   39 0.97395 0.87649 0.14324 1.01467 0.91119 0.17115 1.06397 0.95509
0.20873 1.09659 0.99881 0.22880 1.18867 1.06543 0.28444 1.27401 1.12652 0.33390
1.36546 1.20009 0.38025 1.45273 1.25751 0.42185 1.56931 1.33896 0.47399 1.71472
1.43353 0.53838 1.86479 1.52959 0.59422 2.02935 1.62944 0.65350
3.55009   29   40 0.96648 0.87897 0.13373 1.00062 0.91412 0.15954 1.05797 0.97356
0.19392 1.09508 1.00166 0.21729 1.17613 1.06400 0.26914 1.23299 1.11250 0.30137
1.33617 1.18274 0.35614 1.44023 1.25754 0.40788 1.54559 1.32858 0.45996 1.67486
1.41614 0.51512 1.80322 1.49688 0.56696 1.96977 1.60025 0.62829
3.74317   31   42 0.99226 0.92589 0.14294 1.05440 0.97421 0.18440 1.08393 1.00310
0.20111 1.14605 1.05298 0.23795 1.21321 1.10343 0.27857 1.31679 1.17915 0.33848
1.37397 1.22108 0.36376 1.49000 1.29801 0.42166 1.60804 1.37703 0.47808 1.75398
1.47028 0.53830 1.88778 1.55419 0.58830 2.04048 1.64605 0.64156
3.93895   32   43 0.98471 0.92528 0.12589 1.04135 0.97176 0.16619 1.07883 1.00097
0.19017 1.13911 1.05486 0.22741 1.20770 1.10811 0.26821 1.29362 1.16788 0.31909
1.35705 1.21295 0.35049 1.47176 1.29168 0.40961 1.56696 1.35453 0.44853 1.69217
1.43604 0.50238 1.80785 1.50866 0.54993 1.96510 1.60393 0.60834
4.12063   33   44 0.98535 0.93182 0.11893 1.03122 0.96862 0.14993 1.07636 1.01255
0.17854 1.13571 1.05837 0.21919 1.19550 1.10110 0.25513 1.28614 1.16551 0.30962
1.34885 1.21327 0.34023 1.45836 1.28631 0.39615 1.54232 1.34350 0.43381 1.66086
1.41908 0.48503 1.78807 1.49832 0.53715 1.92970 1.58608 0.58833
4.34208   34   45 0.97763 0.93239 0.10343 1.01290 0.96698 0.12921 1.06586 1.00798
0.16656 1.10745 1.04236 0.19556 1.16797 1.08732 0.23143 1.24372 1.14494 0.27599
1.32504 1.20013 0.31755 1.40039 1.25313 0.35696 1.49933 1.31838 0.40684 1.61113
1.38976 0.45740 1.72846 1.46444 0.50521 1.86659 1.54963 0.55905
4.54790   35   46 0.97301 0.93962 0.09696 1.01037 0.96870 0.12451 1.05746 1.00853
0.15382 1.10707 1.04652 0.18600 1.16659 1.08987 0.22103 1.23158 1.13817 0.26220
```

```
1.30734 1.19026 0.30418 1.38599 1.24514 0.34126 1.48886 1.31334 0.39444 1.58490
1.37545 0.43787 1.69867 1.44825 0.48644 1.81599 1.52062 0.53086
4.75794   36   47 0.97535 0.94459 0.09451 1.01046 0.97596 0.11795 1.05516 1.01122
0.14779 1.09960 1.04465 0.17646 1.15535 1.08660 0.21044 1.21686 1.13071 0.24419
1.29643 1.18695 0.29473 1.36833 1.23633 0.33157 1.47116 1.30406 0.38324 1.56545
1.36595 0.42674 1.67864 1.43736 0.47587 1.78778 1.50570 0.51621
4.97115   37   48 0.97210 0.94934 0.08326 1.00457 0.97635 0.10546 1.05087 1.01047
0.13710 1.08499 1.03891 0.15646 1.13750 1.07733 0.19403 1.20298 1.12502 0.23403
1.28455 1.18116 0.28060 1.35988 1.23217 0.32045 1.42618 1.27663 0.35105 1.54191
1.35197 0.40697 1.65652 1.42542 0.45719 1.75518 1.48683 0.49826
5.20675   38   49 0.97320 0.95447 0.07652 1.00055 0.97415 0.09513 1.04748 1.01301
0.13088 1.07472 1.03432 0.14642 1.13153 1.07614 0.18450 1.19360 1 12022 0.22415
1.26754 1.17154 0.26751 1.32572 1.21093 0.29546 1.41534 1.27164 0.34096 1.51404
1.33545 0.39094 1.61447 1.40054 0.43505 1.71492 1.46329 0.47670
5.42507   39   49 0.97373 0.95542 0.07324 1.00022 0.97993 0.09067 1.04476 1.01364
0.12671 1.07183 1.03527 0.14171 1.12827 1.07607 0.17833 1.19043 1.11960 0.21619
1.26461 1.17079 0.25750 1.31957 1.20922 0.28657 1.40398 1.26523 0.33336 1.50213
1.32971 0.38143 1.59753 1.39060 0.42642
5.66591   40   50 0.97230 0.96032 0.06939 0.99598 0.97960 0.08480 1.03584 1.00931
0.11210 1.07081 1.03567 0.13531 1.12698 1.07645 0.17041 1.18628 1.11835 0.20863
1.25555 1.16682 0.25133 1.31096 1.20469 0.27879 1.39146 1.25884 0.32325 1.46810
1.30904 0.35923 1.56934 1.37374 0.40682
5.91006   41   51 0.97387 0.96402 0.06434 0.99152 0.97827 0.07530 1.03514 1.01090
0.10691 1.06982 1.03716 0.12934 1.12247 1.07505 0.16544 1.18233 1.11732 0.20144
1.24994 1.16386 0.24336 1.29531 1.19576 0.26612 1.38070 1.25259 0.31176 1.45117
1.29887 0.34495 1.54694 1.36051 0.39055
6.13674   42   52 0.97542 0.96558 0.05989 0.99835 0.98504 0.07462 1.03556 1.01333
0.10163 1.06945 1.03793 0.12376 1.12244 1.07590 0.16162 1.17950 1.11649 0.19918
1.24762 1.16342 0.23889 1.28966 1.19293 0.25879 1.37567 1.25013 0.30572 1.44864
1.29787 0.34100 1.54486 1.35990 0.38741
6.38655   43   53 0.97612 0.96917 0.05620 0.99690 0.98604 0.07205 1.03326 1.01271
0.09846 1.06548 1.03656 0.12088 1.11975 1.07580 0.15596 1.17409 1.11399 0.18946
1.23922 1.15938 0.22954 1.28350 1.18968 0.25137 1.36686 1.24477 0.29792 1.43129
1.28716 0.32867 1.52230 1.34612 0.37317
6.63823   44   54 0.97594 0.97068 0.05350 0.99718 0.98728 0.06825 1.04138 1.01914
0.10096 1.07124 1.04226 0.11924 1.11885 1.07593 0.14989 1.17309 1.11446 0.18586
1.23639 1.15801 0.22545 1.28374 1.19020 0.24983 1.35849 1.23999 0.29049 1.42135
1.28152 0.32095 1.51047 1.33900 0.36465
6.89193   45   55 0.97653 0.97263 0.05032 0.99964 0.98980 0.06698 1.04050 1.02081
0.09687 1.06973 1.04197 0.11489 1.11452 1.07437 0.14634 1.16929 1.11271 0.18032
1.23219 1.15581 0.21850 1.27846 1.18734 0.24268 1.34864 1.23429 0.28112 1.41256
1.27633 0.31315 1.49945 1.33228 0.35635
7.17028   46   56 0.97722 0.97389 0.04707 1.00080 0.99283 0.06256 1.03990 1.02113
0.09247 1.06272 1.03840 0.10645 1.11297 1.07413 0.14042 1.16604 1.11112 0.17409
1.22740 1.15304 0.21163 1.27085 1.18298 0.23674 1.34075 1.22959 0.27358 1.40151
1.26947 0.30625 1.48804 1.32552 0.34800
7.43085   47   57 0.97808 0.97664 0.04407 1.00091 0.99357 0.05948 1.03916 1.02166
0.08878 1.06713 1.04232 0.10623 1.11309 1.07482 0.13782 1.16747 1.11253 0.17249
1.22215 1.15033 0.20628 1.26722 1.18103 0.23083 1.33538 1.22628 0.26926 1.41291
1.27733 0.30987 1.47862 1.31997 0.34162
7.69382   48   57 0.97991 0.97863 0.04141 1.00461 0.99728 0.05989 1.04385 1.02580
0.08775 1.07629 1.04918 0.11039 1.11420 1.07596 0.13514 1.16596 1.11211 0.16828
1.22087 1.14988 0.20378 1.27924 1.18921 0.23722 1.34167 1.23097 0.27051 1.40474
1.27248 0.30274
7.98102   49   58 0.98061 0.98021 0.03873 1.00641 0.99939 0.05715 1.04255 1.02558
0.08455 1.07520 1.04892 0.10602 1.11199 1.07509 0.13057 1.16240 1.11024 0.16323
1.21605 1.14693 0.19813 1.27435 1.18629 0.23019 1.32449 1.21999 0.25689 1.40226
1.27116 0.29810
8.27026   50   59 0.98244 0.98230 0.03866 1.00992 1.00251 0.05829 1.04404 1.02710
0.08437 1.07361 1.04844 0.10398 1.11423 1.07717 0.13046 1.15872 1.10801 0.15951
```

```
 1.21522 1.14668 0.19421 1.26771 1.18237 0.22487 1.33160 1.22497 0.26042 1.39303
 1.26536 0.29260
 8.56139   51   60 0.98574 0.98530 0.03747 1.01245 1.00481 0.05770 1.04443 1.02800
 0.08016 1.07549 1.05027 0.10234 1.11597 1.07860 0.12937 1.16241 1.11085 0.15930
 1.21279 1.14554 0.19050 1.26994 1.18416 0.22360 1.33560 1.22782 0.26060 1.39336
 1.26574 0.29011
 8.83198   52   61 0.98794 0.98715 0.03613 1.01292 1.00551 0.05565 1.04968 1.03220
 0.08216 1.08448 1.05675 0.10639 1.11896 1.08102 0.12886 1.16424 1.11260 0.15823
 1.22115 1.15147 0.19293 1.27433 1.18730 0.22432 1.33600 1.22825 0.25883 1.40348
 1.27248 0.29439
 9.14947   53   61 0.98850 0.98813 0.03383 1.01421 1.00716 0.05457 1.04824 1.03141
 0.07937 1.08657 1.05848 0.10639 1.11603 1.07931 0.12517 1.16221 1.11141 0.15395
 1.21591 1.14815 0.18812 1.26825 1.18342 0.21877 1.32870 1.22366 0.25265
 9.44640   54   62 0.99166 0.99111 0.03477 1.01997 1.01146 0.05641 1.05280 1.03488
 0.07987 1.08698 1.05917 0.10405 1.12811 1.08801 0.13179 1.17830 1.12266 0.16492
 1.21681 1.14893 0.18704 1.27360 1.18714 0.22156 1.32735 1.22291 0.24998
 9.74605   55   63 0.99310 0.99233 0.03362 1.02194 1.01309 0.05557 1.05173 1.03451
 0.07730 1.09066 1.06209 0.10483 1.12678 1.08732 0.12912 1.17631 1.12147 0.16103
 1.21456 1.14756 0.18353 1.27265 1.18663 0.21840 1.32598 1.22218 0.24721
10.07096   56   64 0.99598 0.99465 0.03328 1.02150 1.01316 0.05336 1.05150 1.03470
 0.07439 1.09094 1.06249 0.10276 1.13097 1.09042 0.13013 1.17567 1.12119 0.15923
 1.21919 1.15080 0.18474 1.26936 1.18467 0.21427 1.32571 1.22213 0.24469
10.37309   57   64 1.00104 0.99863 0.03691 1.03271 1.02145 0.06005 1.05971 1.04067
 0.08061 1.09404 1.06483 0.10356 1.13286 1.09187 0.12991 1.18252 1.12596 0.16203
 1.23093 1.15886 0.19116 1.28197 1.19318 0.22103
10.69820   58   65 1.00503 1.00182 0.03844 1.03157 1.02085 0.05833 1.06148 1.04209
 0.07981 1.09627 1.06651 0.10392 1.13397 1.09273 0.12858 1.18021 1.12454 0.15829
 1.23277 1.16028 0.19066 1.28602 1.19599 0.22217
11.02341   58   66 0.98523 0.98768 0.02092 1.00899 1.00477 0.03949 1.03753 1.02522
 0.06068 1.06501 1.04470 0.08064 1.10144 1.07024 0.10597 1.13707 1.09501 0.13019
 1.18176 1.12572 0.15829 1.23455 1.16159 0.19022 1.28956 1.19848 0.22306
11.34850   59   67 0.98894 0.99047 0.02244 1.01307 1.00790 0.04068 1.04212 1.02861
 0.06262 1.07115 1.04913 0.08397 1.11035 1.07657 0.11108 1.15219 1.10553 0.13851
 1.18926 1.13098 0.16123 1.23860 1.16442 0.19146 1.28971 1.19865 0.22114
11.69668   60   67 0.99298 0.99358 0.02394 1.01738 1.01110 0.04310 1.04559 1.03118
 0.06382 1.07854 1.05445 0.08779 1.11252 1.07820 0.11013 1.15631 1.10847 0.14007
 1.19257 1.13334 0.16231 1.24176 1 16664 0.19186
12.02170   61   68 0.99405 0.99448 0.02382 1.01907 1.01243 0.04284 1.04550 1.03127
 0.06184 1.08072 1.05609 0.08771 1.11728 1.08161 0.11222 1.16009 1.11121 0.14071
 1.19782 1.13700 0.16401 1.24581 1.16942 0.19328
12.39271   62   69 0.99977 0.99870 0.02647 1.02300 1.01533 0.04451 1.05335 1.03691
 0.06687 1.08307 1.05785 0.08815 1.11805 1.08229 0.11156 1.16173 1.11240 0.14101
 1.20439 1.14151 0.16788 1.25053 1.17263 0.19490
12.74052   63   70 1.00499 1.00253 0.02950 1.03153 1.02152 0.04973 1.05783 1.04018
 0.06914 1.09031 1.06303 0.09263 1.12141 1.08468 0.11337 1.16170 1.11246 0.13938
 1.20735 1.14355 0.16789 1.26024 1.17918 0.19973
13.08806   63   70 0.98872 0.99090 0.01570 1.01077 1.00679 0.03237 1.03792 1.02616
 0.05373 1.06426 1.04477 0.07325 1.09859 1.06887 0.09719 1.13300 1.09274 0.12037
 1.17118 1.11896 0.14487 1.21771 1.15061 0.17360
13.45872   64   71 0.99211 0.99348 0.01745 1.01520 1.01005 0.03579 1.03884 1.02688
 0.05342 1.06821 1.04767 0.07439 1.10048 1.07024 0.09716 1.14139 1.09856 0.12535
 1.17485 1.12153 0.14589 1.21943 1.15182 0.17351
13.82922   65   72 0.99398 0.99493 0.01800 1.01720 1.01156 0.03600 1.04066 1.02829
 0.05373 1.07285 1.05098 0.07716 1.10583 1.07400 0.09986 1.14670 1.10227 0.12763
 1.18790 1.13047 0.15407 1.22503 1.15563 0.17572
14.19948   66   73 0.99976 0.99914 0.02138 1.02578 1.01777 0.04201 1.05027 1.03513
 0.06002 1.08163 1.05716 0.08257 1.11042 1.07724 0.10189 1.14779 1.10307 0.12716
 1.19183 1.13317 0.15517 1.23950 1.16541 0.18458
14.56970   66   73 0.98671 0.98980 0.01003 1.00730 1.00464 0.02687 1.03191 1.02217
 0.04541 1.05792 1.04056 0.06504 1.08999 1.06305 0.08815 1.11965 1.08368 0.10790
```

```
 1.15879 1.11064 0.13324 1.20049 1.13909 0.15980
14.94005   67   74 0.99127 0.99318 0.01294 1.01153 1.00771 0.02897 1.03505 1.02443
 0.04722 1.06282 1.04405 0.06762 1.09304 1.06521 0.08878 1.13116 1.09165 0.11517
 1.16626 1.11579 0.13786 1.20551 1.14252 0.16178
15.33309   68   75 0.99423 0.99536 0.01436 1.01485 1.01011 0.03086 1.03767 1.02634
 0.04816 1.06759 1.04743 0.07000 1.09922 1.06955 0.09244 1.13625 1.09519 0.11746
 1.17578 1.12232 0.14309 1.21570 1.14945 0.16774
15.72624   69   75 0.99698 0.99737 0.01601 1.02175 1.01507 0.03546 1.04661 1.03269
 0.05408 1.07573 1.05317 0.07524 1.10431 1.07312 0.09502 1.13915 1.09723 0.11810
 1.17979 1.12509 0.14458
16.11932   69   76 0.98645 0.98984 0.00676 1.00681 1.00444 0.02326 1.02935 1.02049
 0.04071 1.05500 1.03863 0.05989 1.08452 1.05934 0.08096 1.11793 1.08259 0.10443
 1.14962 1.10447 0.12485 1.18983 1.13195 0.15059
16.51236   70   77 0.99183 0.99374 0.01054 1.01055 1.00714 0.02544 1.03488 1.02445
 0.04427 1.06102 1.04289 0.06343 1.08916 1.06261 0.08355 1.12384 1.08670 0.10739
 1.15759 1.10996 0.12918 1.19714 1.13696 0.15425
16.92834   71   77 0.99460 0.99576 0.01191 1.01477 1.01018 0.02789 1.03857 1.02709
 0.04599 1.06603 1.04644 0.06612 1.09530 1.06692 0.08688 1.12966 1.09076 0.11031
 1.16824 1.11728 0.13588
17.34431   72   78 1.00041 0.99995 0.01609 1.02225 1.01553 0.03315 1.04584 1.03225
 0.05086 1.07228 1.05085 0.07005 1.10215 1.07172 0.09118 1.13798 1.09653 0.11544
 1.17565 1.12236 0.13970
17.76015   72   79 0.98767 0.99084 0.00478 1.00645 1.00430 0.02014 1.02810 1.01971
 0.03655 1.05202 1.03663 0.05502 1.08067 1.05676 0.07587 1.11477 1.08051 0.09957
 1.14660 1.10248 0.12053 1.18185 1.12660 0.14308
18.17596   73   79 0.99271 0.99449 0.00829 1.01239 1.00855 0.02402 1.03322 1.02336
 0.04039 1.05944 1.04189 0.05965 1.08863 1.06235 0.08084 1.12187 1.08544 0.10368
 1.15594 1.10891 0.12636
18.59171   74   80 0.99545 0.99647 0.01016 1.01688 1.01178 0.02721 1.04100 1.02890
 0.04560 1.06817 1.04805 0.06575 1.09605 1.06753 0.08519 1.12817 1.08981 0.10750
 1.16620 1.11595 0.13248
19.03053   75   80 1.00378 1.00245 0.01617 1.02582 1.01816 0.03378 1.04872 1.03438
 0.05067 1.07544 1.05316 0.07058 1.10816 1.07597 0.09359 1.13573 1.09504 0.11165
19.46922   75   81 0.99093 0.99326 0.00524 1.01056 1.00732 0.02124 1.03111 1.02194
 0.03706 1.05581 1.03939 0.05580 1.08282 1.05834 0.07520 1.11433 1.08026 0.09728
 1.14815 1.10359 0.11982
19.90794   76   82 0.99577 0.99677 0.00853 1.01385 1.00969 0.02333 1.03728 1.02633
 0.04117 1.06302 1.04448 0.06069 1.08929 1.06286 0.07930 1.11890 1.08343 0.09951
 1.15776 1.11019 0.12587
20.36971   77   82 1.00204 1.00128 0.01349 1.02345 1.01654 0.03053 1.04474 1.03161
 0.04684 1.07063 1.04983 0.06573 1.10249 1.07206 0.08856 1.12872 1.09023 0.10597
20.80856   77   83 0.99023 0.99283 0.00327 1.00816 1.00566 0.01802 1.02787 1.01969
 0.03326 1.05192 1.03668 0.05147 1.07856 1.05538 0.07103 1.10860 1.07631 0.09216
 1.14011 1.09809 0.11316
21.27029   78   84 0.99484 0.99616 0.00659 1.01241 1.00871 0.02082 1.03232 1.02285
 0.03618 1.05991 1.04232 0.05703 1.08427 1.05938 0.07442 1.11302 1.07938 0.09435
 1.15051 1.10524 0.11981
21.73203   79   84 1.00043 1.00017 0.01106 1.02069 1.01461 0.02713 1.04257 1.03011
 0.04388 1.06718 1.04743 0.06196 1.09746 1.06859 0.08388 1.12580 1.08824 0.10288
22.19363   79   85 0.98975 0.99253 0.00180 1.00701 1.00487 0.01567 1.02649 1.01874
 0.03099 1.04937 1.03491 0.04836 1.07479 1.05277 0.06704 1.10450 1.07349 0.08814
 1.13588 1.09519 0.10936
22.65524   80   85 0.99507 0.99635 0.00573 1.01192 1.00838 0.01911 1.03149 1.02229
 0.03448 1.05698 1.04028 0.05380 1.08224 1.05799 0.07210 1.11381 1.07995 0.09416
23.14011   81   86 1.00099 1.00059 0.01029 1.02122 1.01501 0.02637 1.04209 1.02979
 0.04258 1.06633 1.04685 0.06043 1.09416 1.06630 0.08048 1.12204 1.08566 0.09937
23.62486   81   87 0.99067 0.99321 0.00141 1.00681 1.00476 0.01459 1.02635 1.01865
 0.03006 1.04817 1.03409 0.04666 1.07353 1.05190 0.06518 1.10070 1.07086 0.08450
 1.13257 1.09294 0.10625
24.10950   82   87 0.99526 0.99651 0.00499 1.01200 1.00846 0.01824 1.03327 1.02357
```

```
 0.03502 1.05568 1.03938 0.05195 1.08156 1.05753 0.07063 1.11110 1.07809 0.09144
24.59413   83   88 1.00100 1.00062 0.00925 1.02103 1.01490 0.02530 1.04171 1.02954
 0.04119 1.06651 1.04700 0.05967 1.09252 1.06518 0.07826 1.12228 1.08584 0.09872
25.10178   83   88 0.99168 0.99396 0.00122 1.00771 1.00541 0.01436 1.02555 1.01811
 0.02848 1.04983 1.03527 0.04705 1.07128 1.05035 0.06267 1.09920 1.06984 0.08254
25.58627   84   89 0.99656 0.99746 0.00514 1.01489 1.01054 0.01976 1.03440 1.02439
 0.03509 1.05833 1.04127 0.05307 1.08625 1.06082 0.07347 1.11184 1.07862 0.09121
26.09399   85   89 1.00463 1.00323 0.01142 1.02254 1.01598 0.02569 1.04309 1.03053
 0.04160 1.06665 1.04711 0.05897 1.09271 1.06533 0.07766
26.60176   85   90 0.99388 0.99555 0.00226 1.00972 1.00687 0.01511 1.02760 1.01958
 0.02920 1.05122 1.03627 0.04733 1.07529 1.05317 0.06499 1.10305 1.07253 0.08465
27.13261   86   91 0.99828 0.99870 0.00572 1.01729 1.01226 0.02101 1.03705 1.02627
 0.03635 1.06023 1.04261 0.05395 1.08572 1.06046 0.07223 1.11362 1.07986 0.09176
27.64030   86   91 0.99044 0.99310-0.00113 1.00573 1.00403 0.01164 1.02363 1.01677
 0.02587 1.04594 1.03256 0.04306 1.06721 1.04752 0.05859 1.09355 1.06592 0.07750
28.17111   87   92 0.99559 0.99679 0.00298 1.01203 1.00853 0.01641 1.03153 1.02237
 0.03168 1.05437 1.03850 0.04908 1.08020 1.05661 0.06807 1.10539 1.07416 0.08569
28.70187   88   92 1.00272 1.00189 0.00863 1.02031 1.01442 0.02277 1.03990 1.02830
 0.03792 1.06207 1.04391 0.05451 1.08729 1.06157 0.07267
29.23264   88   93 0.99340 0.99523 0.00076 1.00888 1.00629 0.01348 1.02707 1.01922
 0.02789 1.04800 1.03402 0.04396 1.07407 1.05233 0.06323 1.09917 1.06984 0.08102
29.78659   89   93 0.99844 0.99884 0.00472 1.01656 1.01176 0.01948 1.03552 1.02520
 0.03425 1.05798 1.04104 0.05121 1.08164 1.05763 0.06846
30.31744   90   94 1.00537 1.00379 0.01018 1.02182 1.01550 0.02331 1.04384 1.03109
 0.04032 1.06502 1.04599 0.05605 1.09174 1.06467 0.07530
30.87126   90   94 0.99694 0.99777 0.00307 1.01263 1.00897 0.01589 1.03188 1.02263
 0.03101 1.05236 1.03709 0.04667 1.07658 1.05409 0.06448
31.42516   91   95 1.00268 1.00188 0.00765 1.01973 1.01402 0.02136 1.03896 1.02765
 0.03630 1.05929 1.04197 0.05152 1.08494 1.05993 0.07009
32.00208   91   95 0.99420 0.99582 0.00038 1.00932 1.00661 0.01294 1.02668 1.01895
 0.02664 1.04837 1.03429 0.04337 1.07242 1.05118 0.06121
32.55598   92   96 1.00080 1.00054 0.00579 1.01755 1.01247 0.01933 1.03525 1.02503
 0.03316 1.05711 1.04045 0.04970 1.08036 1.05674 0.06664
33.13293   92   96 0.99287 0.99487-0.00095 1.00748 1.00530 0.01107 1.02355 1.01674
 0.02390 1.04367 1.03098 0.03943 1.06924 1.04896 0.05865
33.70985   93   97 0.99745 0.99815 0.00262 1.01432 1.01018 0.01647 1.03243 1.02303
 0.03073 1.05303 1.03758 0.04642 1.07558 1.05340 0.06287
34.28679   94   98 1.00481 1.00340 0.00861 1.02125 1.01511 0.02178 1.04098 1.02908
 0.03714 1.06434 1.04552 0.05475 1.08808 1.06213 0.07182
34.86374   94   98 0.99666 0.99758 0.00177 1.01187 1.00844 0.01424 1.03067 1.02179
 0.02916 1.05089 1.03607 0.04462 1.07316 1.05171 0.06096
35.46367   95   99 1.00320 1.00226 0.00705 1.01908 1.01357 0.01988 1.03816 1.02709
 0.03473 1.05932 1.04200 0.05071 1.08037 1.05675 0.06604
36.04047   95   99 0.99536 0.99666 0.00043 1.00971 1.00691 0.01222 1.02786 1.01980
 0.02665 1.04834 1.03427 0.04245 1.06683 1.04727 0.05692
36.64055   96   99 1.00219 1.00154 0.00594 1.01808 1.01286 0.01889 1.03658 1.02598
 0.03349 1.05165 1.03661 0.04618
37.26369   96   99 0.99424 0.99586-0.00088 1.00814 1.00578 0.01075 1.02431 1.01729
 0.02467 1.04110 1.02917 0.03886
37.86375   97   99 1.00024 1.00015 0.00434 1.01523 1.01084 0.01812 1.02921 1.02077
 0.03009
38.48674   97   99 0.99240 0.99455-0.00143 1.00580 1.00412 0.01103 1.01883 1.01340
 0.02227
39.10977   98   99 0.99651 0.99749 0.00378 1.01032 1.00734 0.01573
39.73289   98   99 0.99043 0.99314-0.00099 1.00181 1.00128 0.00895
40.35606   99   99 0.99458 0.99611 0.00302
9999.       1    1 1.00000 0.40000 3.72300
 0.41855 0.43726 0.46268 0.48895 0.50925 0.54395 0.57281 0.60256 0.64108 0.66469
 0.70470 0.73775 0.79783 0.84257 0.89781 0.93561 0.98359 1.05283 1.10404 1.17834
 1.24446 1.31181 1.38088 1.46359 1.54932 1.63765 1.71434 1.81835 1.91127 2.02177
```

```
 2.10637 2.25031 2.36843 2.48845 2.61041 2.76655 2.89495 3.05788 3.22434 3.39433
 3.56754 3.72550 3.90319 4.12063 4.30500 4.51027 4.75794 4.97115 5.20675 5.44499
 5.70635 5.99230 6.24057 6.53316 6.82832 7.14867 7.43085 7.78198 8.11428 8.47161
 8.83198 9.19502 9.58456 9.9779010.3963210.8144011.2557011.7199112.1608912.67098
13.1344013.6670614.1994814.7548715.3099715.8882116.5123617.1363517.7601518.43004
19.1229219.7925120.5314021.2702922.0551822.8168323.6248624.4556725.3094326.20941
27.0864628.0557429.0249629.9942931.0328132.0943933.1560134.2867935.27905
15.3880415.1007614.0441212.9401712.3146111.4964710.9944210.29331 9.56829 9.42723
 9.01357 8.42906 7.51487 7.20549 6.54352 6.51482 6.26689 5.69711 5.42433 4.97173
 4.76571 4.56676 4.37903 4.16745 3.81186 3.70096 3.61790 3.46771 3.23418 3.07119
 3.03393 2.83161 2.76363 2.67360 2.60175 2.44388 2.39700 2.32769 2.22603 2.15072
 2.09797 2.05741 2.01779 1.92970 1.90162 1.86303 1.78778 1.75518 1.71492 1.69465
 1.65488 1.61113 1.58795 1.56581 1.54321 1.49947 1.47862 1.45523 1.43783 1.42117
 1.40348 1.38904 1.36662 1.35043 1.33492 1.31118 1.30045 1.28910 1.28133 1.26783
 1.25890 1.25127 1.23950 1.23186 1.21945 1.21076 1.19714 1.18876 1.18185 1.17338
 1.16845 1.16197 1.15602 1.15051 1.14407 1.13975 1.13257 1.12819 1.12303 1.11764
 1.11492 1.11094 1.10528 1.10282 1.09782 1.09420 1.09201 1.08808 1.08554
 4.32313 4.31479 4.09707 4.13030 3.98677 3.98121 3.92314 3.83604 3.64116 3.71764
 3.65034 3.55770 3.40091 3.33686 3.14946 3.19288 3.14030 2.96845 2.93153 2.77730
 2.74358 2.66799 2.62352 2.55510 2.42118 2.39026 2.35925 2.30173 2.19266 2.13544
 2.12448 2.03286 2.00335 1.96183 1.92777 1.85444 1.83134 1.79664 1.74499 1.70430
 1.67766 1.65598 1.63454 1.58608 1.57041 1.54870 1.50570 1.48683 1.46329 1.45134
 1.42770 1.40143 1.38739 1.37389 1.36003 1.33278 1.31997 1.30530 1.29431 1.28375
 1.27248 1.26324 1.24883 1.23837 1.22830 1.21281 1.20577 1.19830 1.19318 1.18425
 1.17833 1.17326 1.16541 1.16030 1.15198 1.14614 1.13696 1.13129 1.12660 1.12085
 1.11749 1.11307 1.10901 1.10524 1.10083 1.09787 1.09294 1.08992 1.08636 1.08264
 1.08076 1.07800 1.07408 1.07238 1.06891 1.06639 1.06487 1.06213 1.06036
 3.35780 3.32076 3.18328 3.03353 2.99567 2.82593 2.75044 2.64149 2.52398 2.50034
 2.43020 2.32805 2.15921 2.09919 1.99550 1.95912 1.90651 1.80491 1.71762 1.60824
 1.55619 1.50445 1.45428 1.39597 1.29340 1.25994 1.23440 1.18728 1.11115 1.05564
 1.04053 0.96996 0.94464 0.91046 0.88254 0.81774 0.79836 0.77052 0.72520 0.69277
 0.66751 0.64973 0.63098 0.58833 0.57438 0.55411 0.51621 0.49826 0.47670 0.46567
 0.44439 0.41975 0.40647 0.39362 0.38036 0.35383 0.34162 0.32684 0.31604 0.30559
 0.29439 0.28516 0.27068 0.26010 0.24985 0.23399 0.22674 0.21922 0.21369 0.20438
 0.19818 0.19285 0.18458 0.17917 0.17033 0.16410 0.15425 0.14826 0.14308 0.13694
 0.13329 0.12838 0.12394 0.11981 0.11496 0.11170 0.10625 0.10291 0.09896 0.09483
 0.09273 0.08965 0.08527 0.08335 0.07946 0.07666 0.07491 0.07182 0.06982
 0.43101 0.46268 0.49567 0.53692 0.55826 0.61776 0.66469 0.72108 0.79783 0.86078
 0.96426 1.05283 1.24446 1.46359 1.84471 2.03582 2.13476 2.25031 2.33876 2.42818
 2.54925 2.64124 2.76655 2.87881 3.00870 3.14048 3.27509 3.42874 3.60248 3.74317
 3.90319 4.10228 4.26798 4.43530 4.62364 4.85445 5.02978 5.24633 5.48496 5.68612
 5.93059 6.15749 6.40746 6.65931 6.91315 7.19190 7.49637 7.78198 8.06982 8.40438
 8.74156 9.03582 9.33197 9.6768010.0476810.3266310.6749611.0698811.4644711.88249
12.2768012.7173413.1807313.6902214.1532114.6854115.1250815.6105916.1655416.72038
17.2519117.8294518.4300419.0305319.6539620.3004520.9471121.5935522.2628922.97847
23.7171324.4556725.2171326.0247526.8556427.6864528.5403329.4403730.3174431.28667
32.2328633.2252334.2406335.2790536.3636237.4713938.5790639.7790540.79462
13.4563711.6182910.64153 9.01604 8.64760 6.80962 6.04980 5.05461 4.05404 3.42856
 2.77874 2.34598 1.75902 1.35223 1.05472 0.98661 0.90104 0.89560 0.89124 0.97132
 0.96443 0.95667 0.96442 0.95756 0.95259 0.95660 0.95400 0.95646 0.95056 0.95635
 0.95325 0.95635 0.96022 0.95966 0.95984 0.95974 0.96175 0.96387 0.96335 0.96368
 0.96589 0.96774 0.96938 0.96865 0.96981 0.97026 0.96758 0.96931 0.96983 0.97073
 0.97124 0.97189 0.97314 0.97432 0.97522 0.98205 0.98267 0.98315 0.98364 0.98416
 0.98448 0.98521 0.98543 0.98026 0.98106 0.97821 0.98206 0.98228 0.98289 0.98326
 0.98379 0.98428 0.98486 0.98502 0.98550 0.98605 0.98630 0.98671 0.98702 0.98747
 0.98784 0.98806 0.98840 0.98878 0.98904 0.98937 0.98958 0.98993 0.99017 0.99047
 0.99072 0.99099 0.99126 0.99147 0.99166 0.99190 0.99136 0.99022 0.98995
 3.92822 3.55785 3.34172 2.96876 2.88167 2.42968 2.23289 1.96396 1.67773 1.48851
 1.28109 1.13519 0.92419 0.76591 0.64136 0.61150 0.57314 0.57391 0.65221 0.73095
 0.73987 0.73561 0.75645 0.79828 0.81050 0.82029 0.82841 0.86523 0.86861 0.88504
```

```
 0.89975 0.90617 0.91479 0.92012 0.93052 0.93377 0.94211 0.94793 0.94813 0.95422
 0.95838 0.96014 0.96438 0.96549 0.96784 0.96893 0.96914 0.97106 0.97250 0.97392
 0.97491 0.97566 0.97714 0.97855 0.97957 0.98468 0.98551 0.98618 0.98667 0.98727
 0.98763 0.98828 0.98855 0.98498 0.98568 0.98371 0.98659 0.98680 0.98729 0.98761
 0.98802 0.98841 0.98886 0.98900 0.98937 0.98981 0.99002 0.99033 0.99057 0.99091
 0.99119 0.99135 0.99161 0.99189 0.99208 0.99233 0.99249 0.99275 0.99293 0.99314
 0.99332 0.99352 0.99372 0.99387 0.99401 0.99418 0.99380 0.99299 0.99280
 3.22872 3.05358 2.87768 2.66937 2.56795 2.30754 2.12095 1.91629 1.66880 1.49007
 1.23681 1.05459 0.74695 0.50345 0.25416 0.22284 0.10241 0.10244 0.10288 0.17058
 0.16826 0.16925 0.15306 0.15289 0.15012 0.13937 0.13533 0.12618 0.12458 0.11398
 0.11124 0.10221 0.09407 0.09028 0.08565 0.08072 0.07546 0.06973 0.06636 0.06308
 0.05782 0.05341 0.04924 0.04735 0.04398 0.04126 0.03493 0.03165 0.02961 0.02717
 0.02524 0.02351 0.02136 0.01923 0.01740 0.02243 0.02081 0.01929 0.01787 0.01646
 0.01538 0.01393 0.01298 0.00731 0.00624 0.00302 0.00477 0.00435 0.00359 0.00308
 0.00249 0.00195 0.00135 0.00115 0.00070 0.00020-0.00002-0.00036-0.00060-0.00094
-0.00120-0.00132-0.00153-0.00176-0.00188-0.00206-0.00212-0.00230-0.00238-0.00249
-0.00257-0.00265-0.00273-0.00277-0.00278-0.00282-0.00222-0.00116-0.00088
```

Appendix D

SAMPLE CONSOLE RECORD

What follows is a listing of the 'console record' of a computer terminal session with the UVic/AIRBLAST display package, a session in which a specific explosion was simulated. The lines marked by an arrow are the user's responses to requests for information put to him by the executing program.

In the example provided here, a 1 kT nuclear device is detonated on the ground surface in a (uniform) atmosphere of temperature 4.5°C, pressure 99 kPa and relative humidity 31%.

The user asks for a summary of all data at a point on the ground surface 150 meters from the charge centre. The user then asks for a time history of pressures at this position. A profile of peak flow velocity is also requested, and a value interpolated at ground range 150 m.

Output data were sent to a disk file which the user has called 'FILE ONE', for subsequent examination. The user might have asked that the be displayed at the terminal. A listing of the disk file which was created is given in the next appendix (appendix E).

```
→ airblast
  LOAD AB
  FILE 8 DISK FFT DATA
  FILE 9 DISK SBT DATA


  START
  EXECUTION BEGINS...

                                                          UVic AIRBLAST/20feb87
  Do you want program information?  1=yes
  ?
→
  What UNITS would you like to use?
  1=metric  [2=British]
  ?
→ 1
  Do you want pressures, etc. NORMALIZED?
  1=yes, normalized to ambient atmospheric values
  2=no, in absolute units (pressures in kPa, for example)
→ 1
  ?
  Where do you want the output data SENT TO?
  1=this terminal  2=a disk file (for printing and/or plotting)
  ?
→ 2
  What would you like to NAME the file?
  (IBM CMS:  7 characters maximum, 1st alphabetic; "filedef" if)
  (a FILEDEF exists, otherwise FILEID will be "FILE yourname A")
→ one

  Explosive type?
  1=TNT  2=ANFO  3=Pentolite  4=hexogen  5=gaseous  6=nuclear
  ?
→ 6
  Nuclear data will be derived from TNT data.

  One kiloton of TNT (=907184.86 kg) releases the same amount of
  energy as the standard 1 kiloton nuclear device, though a much
  greater portion of this energy gets into the blast wave in the
  TNT case.  We are using an adjustment factor of 52%

  Over the range of middle distances, the adjusted TNT data give
  a good approximation to the standard nuclear event.

  What energy yield?  (kilotons, TNT equivalent energy yield)
  ?
→ 1
  TNT-equivalent mass= 0.4717361E+06 kg

  Ambient atmosphere?
  1=standard sea level  [2=altitude dependent]  3=user defined
  ?
→ 3
  Air temperature? (C)
  ?
→ 4.5
  Air pressure? (kPa)
```

```
   ?
→  99
   Relative humidity? (%)
   ?
→  31
   Computed sound speed= 0.33418 m/ms

   Scales relative to 1 kg TNT in standard sea level air=     78.450 for distance
                                                              79.885 for time

   Charge-ground configuration?
   1=free field  2=surface burst  [3=height of burst]
   (spherical)   (hemispherical)  (reflected free field)
   ?
→  2
   Reading from file number 9...
   Source data input is complete


   What kind of data would you like to see?
   (data codes= 0,1,...,18, or 99 for menu)
   ?
→  99

    0=all data at one point       10=particle velocities
    1=shock front radii           11=gas densities
    2=times of shock arrival      12=temperatures
    3=shock front velocities      13=sound speeds
    4=particle trajectories       14=entropy changes

    5=hydrostatic pressures       15=energy densities
    6=dynamic pressures           16=available work densities
    7=total head pressures        17=energy integrals
    8=positive durations          18=work integrals
    9=pressure impulses
                                  99=repeat menu
   ?
→  0
   At what distance from ground zero? ( m)
   (between  78.450 and 1172.828)
   ?
→  150

   For more data, enter a time (ms after time of arrival); or 0,
   for a time history of all pressures, maximum T-TO=   1.405 ms.
   ?
→  0

   Do you want other data?  99=yes (repeat menu)
   or you may enter the data code (0 through 18)
   ?
→  10
   In what format would you like the data output?
   1=spatial profiles  2=time histories
   (3=guide to contouring in R-T plane)
   ?
→  1
   Profile of PEAK data values only?
   1=yes  2=no, full wave profiles
   ?
```

```
→1

  Data output to disk file

  Do you want to interpolate?
  1=yes, normal interpolation
  2=yes, inverse      "
  ?
→ 1
  Where (input one value at a time, units=  m     )
  ?
→ 150
  Result=    1.191396

  Where else ( m     )
  ?
→

  Do you want other data?  99=yes (repeat menu)
  or you may enter the data code (0 through 18)
  ?
→

  Restart?  1=yes
  ?
→

  User terminated.  Bye.
```

Appendix E

SAMPLE OUTPUT FILE

What follows is a listing of the disk file defined by the user in the data retrieval session described in the previous appendix. Keywords with the output data so that other programs may access the data for plotting, etc.

```
1UVic AIRBLAST/20feb87                                           keyword

                                                   surface burst   event
 Charge=    1.000 kT nuclear
 TNT-equivalent mass= 0.4717361E+06 kg
 Atmosphere as follows:
 Temperature              TO=   4.50000  C
 Pressure                 PO=  99.00000 kPa
 Computed sound speed  CO= 0.3341789 m/ms
 Scale factors: S,ST= 78.450, 79.885

 Source data code= GCNJI77J

                                                              point data
 Distance from charge centre= 150.000 m
 Time of shock front arrival= 114.546 ms
 Shock front velocity, Mach #   1.944              649.725 m/s

 Peak hydrostatic overpressure ratio=  3.243       321.100 kPa
  "   dynamic pressure ratio        =  2.567       254.179 kPa
  "   total head overpressure ratio =  6.415       635.049 kPa
  "   density ratio                 =  2.583         3.209 kg/m3
  "   temperature ratio             =  1.643       182.950 C
  "   sound speed ratio             =  1.282       428.312 m/s
  "   particle velocity, local Mach #  0.930       398.203 m/s

 Peak excess energy  = 1.081749 J/cc
  "   available work = 0.496080  "
 Entropy change      = 0.083778 J/K/g

 Positive duration (static overpressure)=   1.298 ms
 Static overpressure impulse=    1.067 atm-ms

 T-TO(ms)  Pstatic  Density  Velocity     Pd      Ptot    Energy    Work
    0.000    3.243    2.583    1.192   2.567    6.415    1.082    0.496
    0.128    2.161    2.179    0.914   1.274    3.630    0.677    0.214
    0.242    1.550    1.907    0.726   0.703    2.325    0.464    0.107
    0.371    1.119    1.689    0.571   0.386    1.530    0.323    0.054
    0.505    0.759    1.489    0.422   0.186    0.951    0.211    0.024
    0.644    0.502    1.335    0.300   0.084    0.588    0.136    0.010
    0.787    0.303    1.207    0.193   0.031    0.334    0.080    0.004
    0.935    0.181    1.126    0.120   0.011    0.192    0.047    0.001
    1.100    0.090    1.063    0.062   0.003    0.093    0.023    0.000
    1.254    0.017    1.012    0.012   0.000    0.017    0.004    0.000
    1.405   -0.040    0.971   -0.029   0.001   -0.039   -0.010    0.000

 Pressure, density and flow velocity are normalized to ambient
 values.  Static and total pressures are overpressures.  Energy
 is excess energy density.  It and available work are in J/cc.

                                                            peak profile
 Peak particle velocity:
    R( m)  V(Mach#)
   78.450     2.879
   80.804     2.815
   83.157     2.784
```

85.511	2.670
87.864	2.545
90.218	2.471
92.571	2.371
94.925	2.308
97.278	2.217
100.416	2.119
102 770	2.099
105.908	2.041
109.046	1.955
112.184	1.814
115.322	1.763
118.460	1.651
121.598	1.646
124.736	1.602
128.658	1.497
131.796	1.444
135.719	1.352
139.641	1.308
143.564	1.265
147.486	1.223
151.409	1.174
155.331	1.088
160.038	1.060
163.961	1.038
168.668	0.999
173.375	0.935
178.082	0.888
182.789	0.877
188.280	0.816
193.772	0.795
198.479	0.766
203.970	0.743
210.246	0.689
215.738	0.673
222.014	0.649
228.290	0.612
234.566	0.583
240.842	0.563
247.118	0.547
254.178	0.531
261.239	0.495
268.299	0.484
276.144	0.467
283.989	0.435
291.834	0.420
299.679	0.402
308.309	0.393
316.938	0.374
325.568	0.354
334.197	0.342
343.611	0.332
353.025	0.321
363.224	0.299
372.638	0.288
383.621	0.276
393.819	0.267
404.802	0.258
415.785	0.248
427.552	0.241

```
  439.320     0.228
  451.872     0.220
  464.424     0.211
  476.976     0.197
  490.313     0.191
  503.649     0.185
  517.770     0.180
  532.676     0.173
  546.797     0.167
  562.487     0.163
  578.177     0.156
  593.867     0.151
  610.341     0.144
  627.600     0.139
  644.859     0.130
  662.903     0.125
  680.946     0.121
  699.774     0.116
  719.387     0.112
  738.999     0.108
  759.396     0.105
  780.578     0.101
  802.544     0.097
  824.510     0.094
  847.260     0.090
  870.795     0.087
  895.115     0.084
  920.219     0.080
  945.323     0.078
  971.996     0.076
  998.669     0.072
 1026.126     0.070
 1055.153     0.067
 1084.179     0.065
 1113.990     0.063
 1145.371     0.061
 1172.828     0.059
  150.000  1.191396 (interpolated)
```

51601