PCBoom Version 7 Technical Reference

Technical Reference Authors (affiliations at time of contribution)
Joel B. Lonzaga, NASA Langley Research Center
Juliet A. Page, Volpe National Transportation Systems Center
Robert S. Downs, Volpe National Transportation Systems Center
Sophie R. Kaye, Volpe National Transportation Systems Center
Meghan J. Shumway, Volpe National Transportation Systems Center
Alexandra Loubeau, NASA Langley Research Center
William J. Doebler, NASA Langley Research Center

Code Developers (affiliations at time of contribution)
Joel B. Lonzaga, NASA Langley Research Center (current developer of PCBoom and author of Enhanced Burgers Solver)
Juliet A. Page, Volpe National Transportation Systems Center (developer and author of PCBoom methodologies)
Sophie R. Kaye, Volpe National Transportation Systems Center (current developer)
Robert S. Downs, Volpe National Transportation Systems Center (current developer)
Aric R. Aumann, NASA Langley Research Center (current developer)
John M. Morgenstern, Lockheed Martin (developer of Turbo)
Joseph A. Salamone, III, Gulfstream (developer of LNTE)
Trevor A. Stout, Pennsylvania State University (developer of the turbulence FIR filters in FiltView)
Clifton B. Wilmer, KBR (developer of SonicBAT software)
Christopher M. Hobbs, KBR (developer of FiltView)
Kevin A. Bradley, KBR (contributor to SonicBAT-Run-Fort code)
Victor W. Sparrow, Pennsylvania State University (developer of the turbulence filter subroutine within WCON and FiltView)
Lance L. Locey, Pennsylvania State University (developer of the turbulence filter subroutine within WCON)
Kenneth J. Plotkin (deceased), Wyle Laboratories, Inc (developer and author of Legacy PCBoom methodologies)

December 2022

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NTRS Registered and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counter-part of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at http://www.sti.nasa.gov

- E-mail your question to help@sti.nasa.gov

- Phone the NASA STI Information Desk at 757-864-9658

- Write to:
  NASA STI Information Desk
  Mail Stop 148
  NASA Langley Research Center
  Hampton, VA 23681-2199

NASA/TM–20220007177
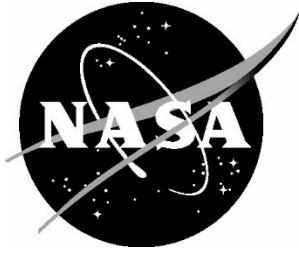
PCBoom Version 7 Technical Reference

Technical Manual Authors (affiliations at time of contribution)
Joel B. Lonzaga, NASA Langley Research Center
Juliet A. Page, Volpe National Transportation Systems Center
Robert S. Downs, Volpe National Transportation Systems Center
Sophie R. Kaye, Volpe National Transportation Systems Center
Meghan J. Shumway, Volpe National Transportation Systems Center
Alexandra Loubeau, NASA Langley Research Center
William J. Doebler, NASA Langley Research Center

Code Developers (affiliations at time of contribution)
Joel B. Lonzaga, NASA Langley Research Center (current developer of PCBoom and author of Enhanced Burgers Solver)
Juliet A. Page, Volpe National Transportation Systems Center (developer and author of PCBoom methodologies)
Sophie R. Kaye, Volpe National Transportation Systems Center (current developer)
Robert S. Downs, Volpe National Transportation Systems Center (current developer)
Aric R. Aumann, NASA Langley Research Center (current developer)
John M. Morgenstern, Lockheed Martin (developer of Turbo)
Joseph A. Salamone, III, Gulfstream (developer of LNTE)
Trevor A. Stout, Pennsylvania State University (developer of the turbulence FIR filters in FiltView)
Clifton B. Wilmer, KBR (developer of SonicBAT software)
Christopher M. Hobbs, KBR (developer of FiltView)
Kevin A. Bradley, KBR (contributor to SonicBAT-Run-Fort code)
Victor W. Sparrow, Pennsylvania State University (developer of the turbulence filter subroutine within WCON and FiltView)
Lance L. Locey, Pennsylvania State University (developer of the turbulence filter subroutine within WCON)
Kenneth J. Plotkin (deceased), Wyle Laboratories, Inc (developer and author of Legacy PCBoom methodologies)

December 2022

Acknowledgements

Dedication

This PCBoom Technical Reference is dedicated to Dr. Kenneth Jay Plotkin for his decades of contributions to the advancement of sonic boom research and foundational development of the PCBoom software.

Table of Contents

List of Figures

1   Introduction

PCBoom is a suite of sonic boom propagation programs that applies full three-dimensional ray tracing based on geometrical acoustics. It predicts sonic boom ground waveforms and footprints from supersonic flight vehicles performing arbitrary maneuvers using a variety of built-in or user-supplied nearfield vehicle source definitions. It also computes loudness metrics, ground signature locations, and sonic boom propagation times. The nomenclature "PC" in PCBoom denotes that the programs were historically targeted to run on a personal computer.

PCBoom provides a valuable contribution toward sonic boom research as illustrated by the following examples:
- PCBoom allows for quick "what if" type predictions for a multitude of sonic boom scenarios, including low boom design iteration work.
- The detailed flight planning capabilities allow for flight conditions and waypoints to be provided to pilots to generate booms needed for research.
- PCBoom allows for post-flight analysis of as-flown trajectories with measured weather data.
- PCBoom has been used in sonic boom damage claim cases to determine claim veracity.

1.1   About PCBoom

PCBoom has its roots in the NASA sonic boom program written by Thomas [1] in the early 1970s. Subsequent development consisted of adding focus boom prediction capability [2] and extending the original code to handle full maneuvers as well as a variety of aircraft source inputs [3, 4]. Three-dimensional geometrical acoustics ray tracing algorithms were added [5] while Thomas' original flat Earth layered ray equations have been retained as an optional feature. A historical timeline of the code development is presented in Section 1.1.1.

The current version of PCBoom determines the ray paths using either the three-dimensional ray tracing equations of geometrical acoustics, or the flat Earth layered ray equations used by Thomas. The waveforms are predicted from a Burgers equation, which is numerically solved using solvers with varying levels of approximation.

1. Lossless Burgers solver:
    This solver is integrated in the main program, called FOBoom, and ignores atmospheric absorption resulting in a very fast calculation.
2. Legacy Burgers solver:
    This solver resides outside of FOBoom and accounts for atmospheric absorption. The solver can be run with or without a graphical user interface (GUI).
3. Enhanced Burgers solver:
    Recently developed at NASA, this solver is integrated into FOBoom and accounts for atmospheric absorption. It is significantly more efficient than the Legacy Burgers solver. It also accounts for full wind effects [6]. Doppler effects are included via frequency shifting based on the motion of the medium relative to the ground reference frame. The difference in propagation path length through the medium relative to the reference frame due to winds (convective effect) is accounted for by scaling the absorption coefficients [6].

The functionality of PCBoom is provided by the main program, called FOBoom (compiled with the name pcboom7xy where x and y are numbers), and post-processing modules. FOBoom contains the algorithms

for ray path determination and waveform modeling using either the Lossless Burgers solver or Enhanced Burgers solver. In addition to FOBoom, there are modules within PCBoom that can be used to process the data generated by FOBoom or analyze information in advance of running FOBoom. The modules are used to integrate certain propagation effects such as turbulence, model propagation emanating from the top of a supersonic aircraft, or visualize interactive signatures and footprints.

The FOBoom executable (pcboom7xy) is run using a Command Prompt on a Windows Operating machine or a Terminal on either a Mac or Linux machine. The executable requires an argument, which is the name of an input file containing a series of keywords. The keywords control the input and output operations within FOBoom and provide instructions to the program for reading the initial aircraft signatures, flight conditions along its trajectory, and atmospheric profiles. The program extracts the initial signature along a specified direction and propagates it through the specified atmosphere until it reaches the specified altitude. The predicted waveform, ray path, propagation time, and other information are written as output files.

PCBoom 7 includes the following capabilities:
- Specification of a vehicle nearfield signature as either an *F*-function, a nearfield pressure distribution $\Delta p/p_\infty$ via data from a computational fluid dynamics (CFD) solution, a simple form from a library of aircraft, or a blunt hypersonic body
- Calculation of booms along rays specified by the user, or across the full width of the primary sonic boom carpet
- Specification of arbitrary aircraft maneuvers in either local Cartesian coordinates or in geographic latitude and longitude
- Calculation of shock structures, either as a simple Taylor structure or via molecular relaxation absorption processes
- Ray tracing through a 3D windy atmosphere over either a flat Earth or an ellipsoidal Earth using the WGS-84 global reference system
- Calculation of spectra and a variety of loudness metrics for ground booms
- Calculation of the effect of terrain or finite ground impedance on boom signature reflection for receivers located above the ground
- Calculation of the effects of turbulence on sonic booms
- Calculation of secondary booms including tracing of "Over the Top" ray paths
- Prediction of focused signatures from waveforms using a 2D Lossy Nonlinear Tricomi Equation (LNTE), and of the secondary post-boom signatures a distance away from the focal points
- Calculation with a launch vehicle mode, including the effect of an underexpanded rocket plume
- Interactive visualization of computed sonic boom footprints, waveforms and ray paths using provided supplemental tools
- Calculation of sonic boom waveforms below the surface of the ocean

1.1.1 PCBoom History

The algorithms present in PCBoom have been improved over decades as the physical understanding of sonic boom phenomena and computing technology have evolved. Since the earliest research in the 20th century, NASA has been at the forefront of sonic boom research and the development of computer modeling. Given the number of organizations interested in predicting sonic booms from arbitrary flight conditions, PCBoom has enjoyed a number of sponsors and code developers. Table 1-1 outlines some of the key elements of the PCBoom historical code development timeline.

Table 1-1 PCBoom historical code development timeline.

Year(s)	Researcher *Entity*	Description of Activity	Funding Source	Version(s)
1972	Charles L. Thomas *NASA*	Sonic boom propagation with winds and the Waveform Parameter Method	NASA	Initial source
1992-1995	Kenneth J. Plotkin *Wyle Laboratories, Inc.*	PC-based version, launch vehicle and plume modeling, importing trajectory data from external sources	US Air Force Research Lab, WPAFB	PCBoom3
1995-2013	Kenneth J. Plotkin and Juliet A. Page *Wyle Labs*	Carlson model, focal zone contouring, 3D ellipsoidal Earth model, Crows Turbulence Model, Lossy Burgers Equation, Middleton–Carlson–Hayes waveform steepening, Over the Top and underwater booms	United States Air Force & NASA	PCBoom versions 4, 5, 6.6
2009-2012	Joseph Salamone *Pennsylvania State Univ. & Gulfstream*	Lossy Nonlinear Tricomi Focal zone solver	NASA	6.6.6b
2015-2020	Juliet A. Page *Volpe*	Cylinder mode expansion	NASA & US Department of Transportation Volpe Center	6.8b
2017-2018	Joel B. Lonzaga *NASA*	Efficient inline (Enhanced) Lossy Burgers solver, integrated into FOBoom; Enabled PCBoom to be run on Windows, Mac OS X, and Linux machines	NASA	6.7.1
2015-2018	Clifton Wilmer, Christopher Hobbs *KBR Wyle;* John Morgenstern, *Lockheed Martin*; Trevor Stout, *Pennsylvania State University*	Classical and FIR filter Turbulence Models, sBOOM interface, high-resolution signatures in postprocessors	NASA	7.0.0
2019-2020	Joel B. Lonzaga *(NASA)*, Juliet Page *(Volpe), KBR Wyle*	Harmonization of versions 6.7.1.x, 6.8b, 7.0.0	NASA	7.1.0
2021-2022	Joel B. Lonzaga *NASA*	Integration of KZK turbulence filters with the Enhanced Burgers solver	NASA	7.2.0 7.3.0

1.2 About this Technical Reference

The use of PCBoom requires a general understanding of the physics of sonic boom generation and propagation. This Technical Reference provides information on sonic boom theory and its implementation in PCBoom 7. Supplemental information about sonic booms is available in Maglieri et al. [7]. For information on how to use the PCBoom code, see the PCBoom User's Guide [8] or its latest version.

2   PCBoom System Architecture

Sonic boom calculations can be divided into the following specific elements:

1) ***Prediction or specification of the pressure disturbance in the vicinity of the vehicle***. For supersonic slender bodies, the source function can be calculated directly from vehicle geometry via the equal-area rule. In that scenario, it is required that the radial distance is large enough that the disturbance is a locally axisymmetric acoustic wave, but small compared to atmospheric gradients. PCBoom can also generate simple "N-wave" *F*-functions from Carlson's model [9], based on vehicle weight, length, and shape category. The source function can also be obtained empirically from wind tunnel or flight test measurements or extracted from CFD flow-field calculations, making use of Equation (1) to convert pressure disturbance to *F*-function form. In all input cases, conversions are applied internally to produce *F*-function forms of source characteristics.

2) ***Calculation of linear geometrical acoustic propagation to large distances, accounting for atmospheric gradients***. This is accomplished by the method of geometrical acoustics. The computed linear ray field depends only on the vehicle kinematics and atmospheric properties. The amplitude of the propagating acoustic disturbance is governed by the change in area of ray tubes (bundles of differentially separated rays) and local acoustic impedance. Section 3.2 characterizes the application of linear acoustic principles to sonic boom.

3) ***Calculation of the nonlinear steepening of the boom signature as it propagates***. One technique employs a cumulative "advance" or "age" parameter, which is computed as part of the geometrical acoustics solution. This age parameter represents the differences in arrival time, at various distances along a ray, between an acoustic wave with amplitude small enough that deviation from local sound speed is negligible, and a weak nonlinear wave of finite strength $\delta p/p$, as described in Section 3.3.1.1. The nonlinear distortion of each part of the boom signature consists of an advance proportional to its original strength times the age parameter. Signature aging and steepening are described in Section 3.3.

The three elements above are the standard components of sonic boom calculation programs. Figure 2-1 shows the typical computational flow. Note that elements (1) and (2) are carried out independently, and their results merge only in the nonlinear signature calculation (3).
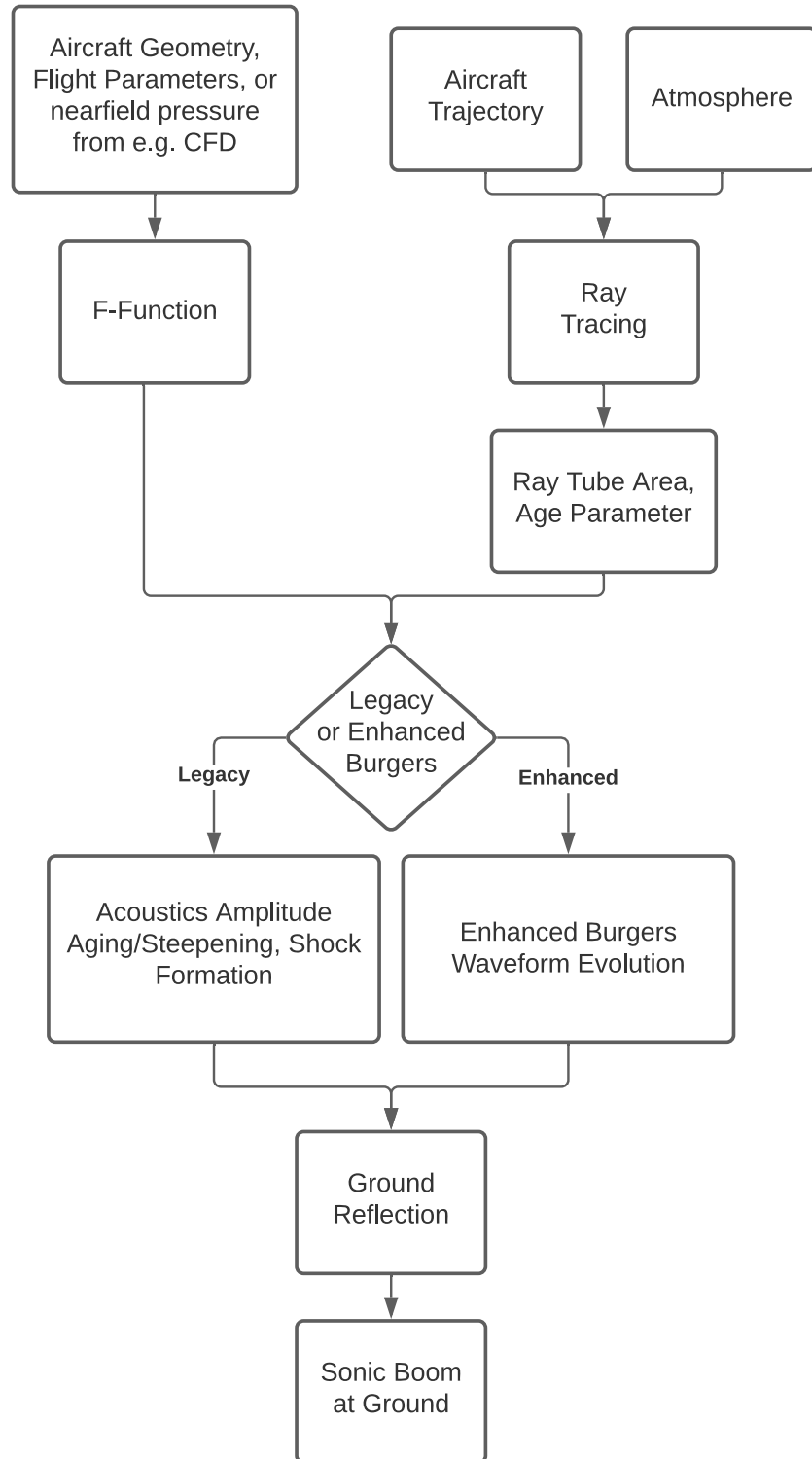
Figure 2-1: Logical Flow of Sonic Boom Calculations.

There are additional elements to sonic boom computation which must be considered under certain circumstances:

Prediction and Computation of Focal Zone Signatures. Focal zones exist where the ray tube area vanishes, and the geometrical acoustic solution is singular. Diffraction effects cause such focused superbooms to be finite, and validated local solutions are available. Focus solutions must be locally applied to focal zones using the adjacent geometrical acoustics calculation as a boundary condition. PCBoom's application of focal zone theory for undertrack focusing has been validated by flight tests [2]. Section 3.3.4 contains the theory and PCBoom computational technique for focal zones.

Computing finite shock thickness. Shock waves have a finite thickness, dependent on shock amplitude and atmospheric conditions. This thickness is very small compared to overall signature length, and in the lossless Burgers solver this thickness is neglected, with sonic boom shocks treated as zero-thickness jumps. Note that legacy PCBoom methods, include via PCBFoot, the capability to apply the simple shock-thickening scheme described in Section 3.3.1.3. Shock structure affects the upper frequency content of booms and is important in loudness calculations. Sufficient empirical N-wave data are available to empirically define nominal shock rise times, which can be used with the lossless Burgers solver to apply to nominal thin-shock ground boom waveforms in the form of a simplified shock thickness technique as described in Section 3.3.

Shock structure is controlled by a combination of nonlinear steepening and molecular relaxation processes, and may be computed from the Burgers equation. Two full Burgers equation solvers are included in PCBoom, and one of which is the Enhanced Burgers solver that is contained within FOBoom. The other method is implemented as a postprocessor with a graphical interface (PCBurg) or as a batch calculation tool (HeadlessBurgers). The simplified finite shock thickness technique and the Burgers solver methods of signature evolution are described in Section 3.3.

Propagation of Sonic Booms through turbulence. While most sonic boom propagation physics is deterministic, turbulence effects are stochastic processes that affect waveforms passing through turbulent regions, most notably the atmospheric boundary layer, consisting of the last few thousand feet of propagation. In the context of sonic boom modeling in PCBoom, turbulence effects are applied as a statistical variation about the deterministic solution, and three different turbulence models are included. A detailed discussion of these turbulence models is contained in Section 3.3.3.

Receiver Module and Ground Effects. Once the sonic boom reaches the vicinity of the ground, the effects of reflections based on plane waves propagating over varying impedance may be computed for receiver locations on the ground or at finite heights above the ground. Loudness and other important sonic boom metric calculation methods, including sonic boom penetration below water in littoral and deep regions, are discussed in Section 3.4.

2.1   Modules Overview

The modules that make up PCBoom are shown in Figure 2-2. These consist of the main PCBoom program, FOBoom, as well as pre- and post-processors. Each module performs a discrete portion of sonic boom analysis, and combinations of modules are used to complete analyses depending on analysis goals. For more information on the functionality of these modules, see the PCBoom User's Guide [8].
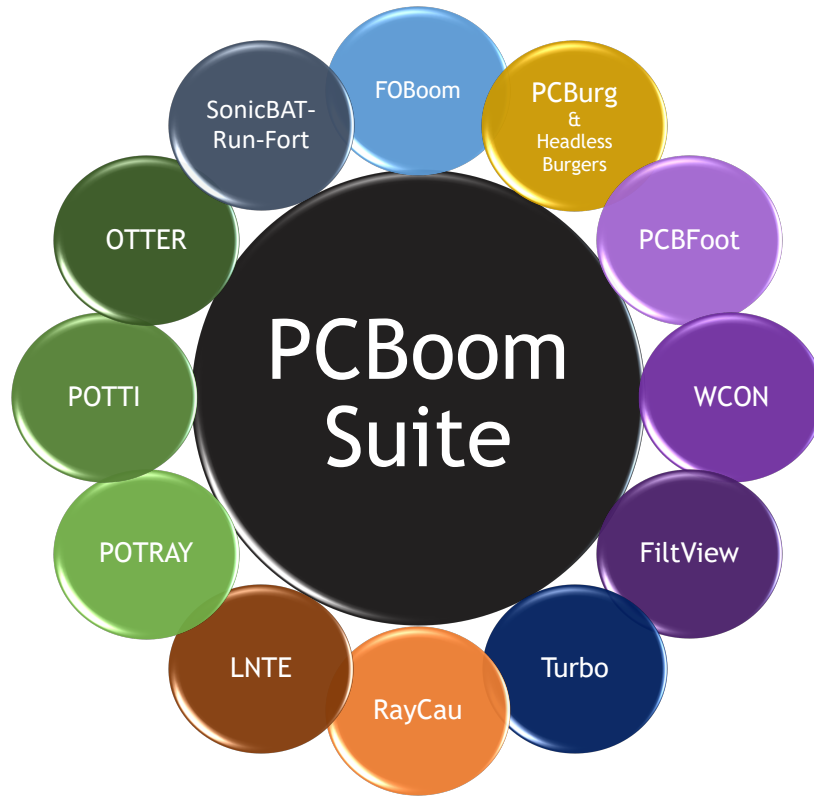
Figure 2-2: PCBoom Modules.

3   Overview of Sonic Boom Propagation for PCBoom

An aircraft traveling at supersonic speed generates a disturbance in the form of shock waves. Near the aircraft, these shock waves form a detailed pressure signature, which relates to every detail of the aircraft's geometry. A sketch of this disturbance is provided in Figure 3-1. As this disturbance propagates away from the aircraft, its shape tends to stretch and distort toward a simpler "N-wave", with a bow shock associated with the front of the aircraft, a linear expansion to a negative overpressure, then a second shock related to the recompression at the end of the aircraft. This N-wave, sketched at the bottom of Figure 3-1, is heard at ground level as a sonic boom. The magnitude and duration of the boom depend on the size, shape, and weight of the aircraft, as well as flight parameters and atmospheric conditions along the ray path of the sonic boom.
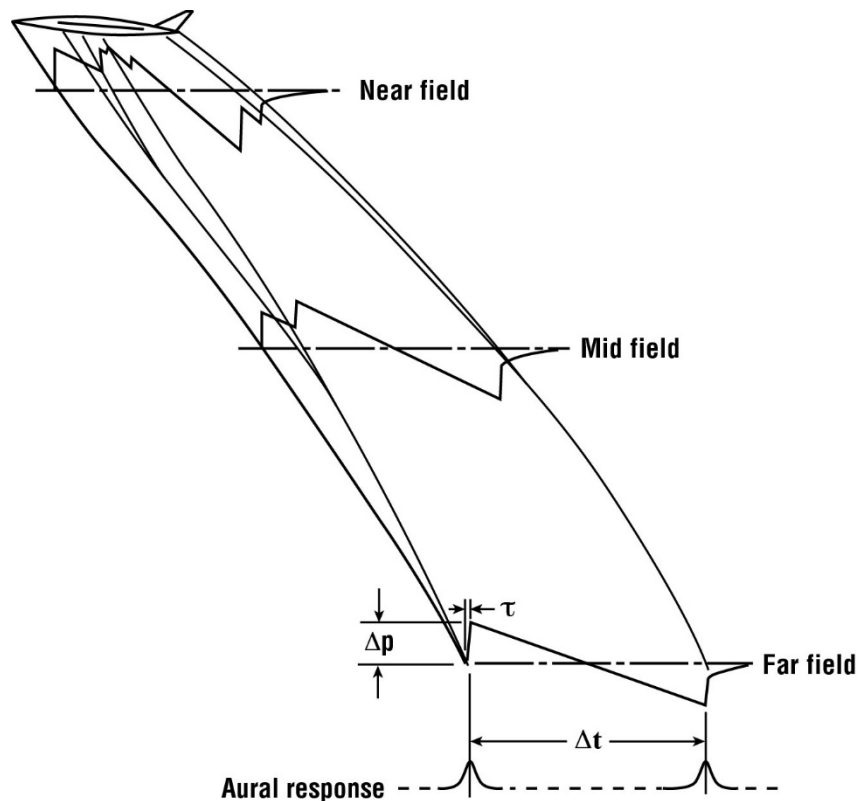
Figure 3-1: Sonic Boom Generation, Propagation, and Signature Evolution (*Source: NASA-SP-622* [7]).

The atmospheric conditions between the aircraft and the ground play an important role in the nature of the sonic boom that reaches the ground – both in terms of the boom location and the characteristics of the signature itself. Many of the diagrams in this section were originally developed by Maglieri and Plotkin [10] while others were refined by Maglieri et al. [7], and are used here with permission. Changes in the atmospheric temperature, wind speed, and wind direction serve to vary the speed at which the waves propagate, thereby causing rays to bend and curve. The impact of this is illustrated in Figure 3-2, where the aircraft is flying out of the page, toward the viewer. In the lateral direction, rays that travel from the aircraft directly to the ground form the primary sonic boom carpet. Other rays, either those originating at higher azimuths relative to the vehicle, or those rays that reflect off the ground, travel upward then curve back down to the ground, forming the secondary boom carpet. Figure 3-3 illustrates these sonic boom carpet regions for a typical supersonic operation.
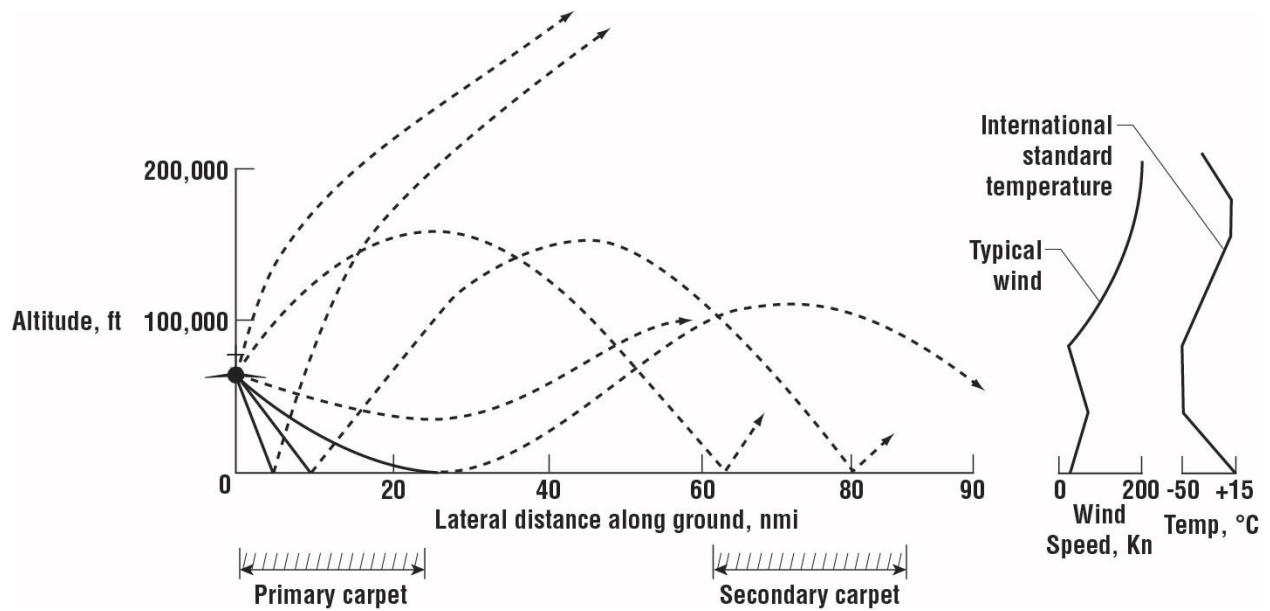
Figure 3-2: Sonic Boom Propagation Paths (*Source: NASA-SP-622* [7]).
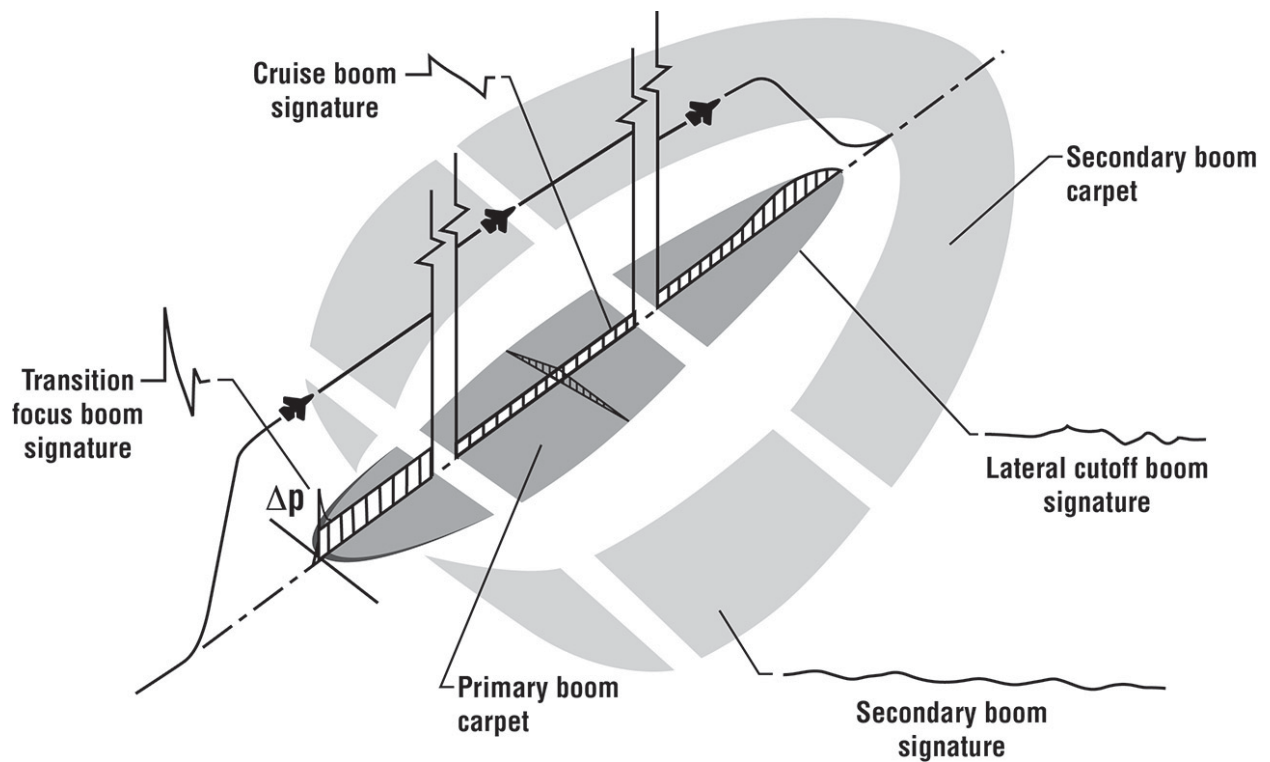
Figure 3-3: Sonic Boom Carpet Regions (*Source: NASA-SP-622* [7]).

Figure 3-1 shows the sonic boom under a supersonic aircraft. The rays are shown curved to represent the distortion due to sound speed gradients in the atmosphere. Sonic booms are generated on both sides as well, so that the rays shown in Figure 3-1 are a two-dimensional cut of a three-dimensional wave cone such as that shown in Figure 3-4. This figure shows a pure cone, but actual wave cones are distorted by atmospheric gradients. Note that the boom wave cone intersects the ground in a hyperbolic

pattern; N-waves are sketched at this intersection in Figure 2-4. This hyperbolic intersection line is sometimes called an isolabe. The boom moves with the aircraft, and the shaded area on the ground to the rear represents areas that have experienced a boom. The area over which the isolabe sweeps the ground during a supersonic flight is often referred to as the boom carpet. Because of the gradients in real atmospheres, the cone is usually distorted such that the isolabe does not extend indefinitely to either side, but stops at some lateral cutoff distance. The distance from the left cutoff to the right cutoff is referred to as the carpet width.
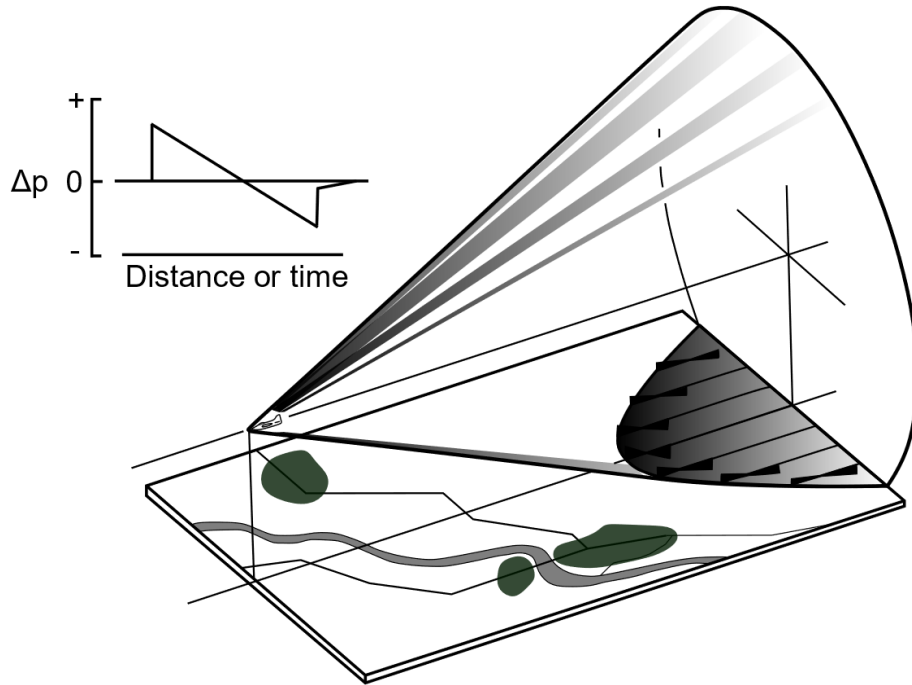
Figure 3-4: Sonic Boom Waves and N-Wave Signature (*Source: Reproduced from Carlson and Maglieri* [11] *H. W. Carlson and D. J. Maglieri, "Review of Sonic-Boom Generation Theory and Prediction Methods," The Journal of the Acoustical Society of America, vol. 51, no. 2, pp. 675-685, 1972. https://doi.org/10.1121/1.1912901., with the permission of the Acoustical Society of America*).

Aircraft Mach number has surprisingly little effect on the amplitude of the boom but does affect its ground location. The half angle of the wave cone is given by the Mach angle, which is inversely related to the Mach number. At high Mach numbers, the wave cone is very slender, trailing further behind the aircraft, while it is broad at low supersonic Mach numbers.

Figure 3-1 and Figure 3-4 present a sonic boom from an aircraft-fixed viewpoint. This viewpoint shows the aircraft and the wave patterns as they exist at a given time and is useful when considering steady flight. While the wave cone exists as shown at a given time, it is actually generated over a period of time. A maneuvering aircraft generates a distorted wave cone. When the aircraft is maneuvering, or when the interest lies at specific points on the ground, it is better to adopt a viewpoint fixed to the ground and analyze the boom in terms of rays rather than wavefronts. Figure 3-5 shows a sonic boom from both wave and ray viewpoints. The aircraft is flying toward the right (rather than to the left as in Figure 3-4), and the wave cone and isolabe are shown to the left of the aircraft. On the other hand, a ray cone is shown toward the right. The ray on the cone is perpendicular to the wave, and represents the path of sonic boom energy that is generated at a given time. The boom generated at time "t" in Figure 3-5 will propagate along rays and reach the ground at a later time. The hyperbolic line where the ray cone intercepts the ground is referred to as an isopemp. The isopemp is always forward of the position at

which the aircraft is located when the boom is generated. It is common to refer to this distance from the aircraft position to the isopemp, measured along the flight path, as the "forward throw" of the boom. The ray cone is narrow at low Mach numbers (long forward throw) and broad at high Mach numbers (small forward throw).

It is important to recognize that once the aircraft generates the vertex of its Mach cone at a given time, the corresponding ray cone and its isopemp are established and unaffected by subsequent maneuvers. The Mach cone is made up of a collection of boom elements from a number of earlier ray cones. For a steady flight through a uniform atmosphere, the wave cone and the Mach cone are the same. Figure 3-4 and the left side of Figure 3-5 are valid only if the aircraft has been flying at a steady speed; had it been maneuvering or flying with a varying speed, the wave cones would be distorted.
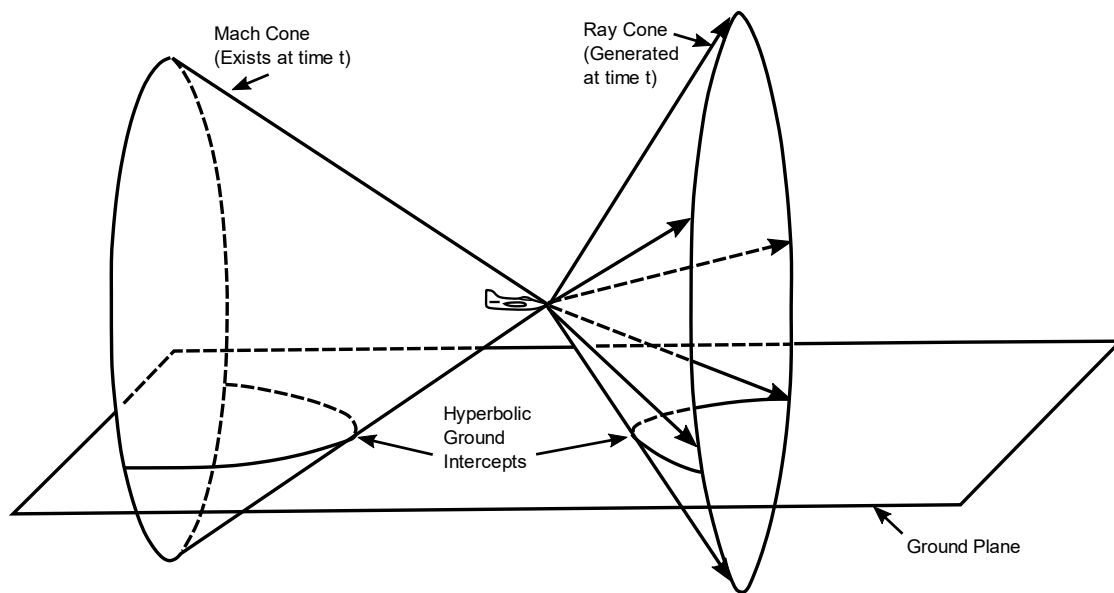
Figure 3-5: Wave and Ray Viewpoints for Steady Flight (*Source: Plotkin* [12]*, reprinted with permission*).

If the ray cones are computed at regular intervals for an aircraft in a level, steady flight, the isopemps are a sequence of identical forward-facing hyperbolic curves following the flight track. This is illustrated in Figure 3-6, where the isopemps are plotted at six-second intervals for such a level flight. In this figure, the aircraft is flying toward the top of the figure (as indicated by the direction of the arrow). The flight track is the vertical line, and the first trajectory point analyzed is marked by a "+". The forward throw of the first trajectory point is the distance from the "+" to the first isopemp.

If the aircraft makes a turn, then the isopemps will be laid out in a curved pattern. Because of the forward throw, the flight track will lie toward the inside of a curve connecting the centers of the isopemps. If the turn is sharp enough, the flight track can even have a smaller radius of curvature than a line connecting the cutoff edges of the isopemps. This is illustrated in Figure 3-7 for a segment of a turn where the aircraft is initially headed north (beginning of trajectory marked by a "+") and turns to the right. In this figure, the isopemps are overlapping toward the inside. Overlapping isopemps indicate the formation of a focused boom.

Steady flight
Mach 1.2, 10 kft AGL

4 nmi
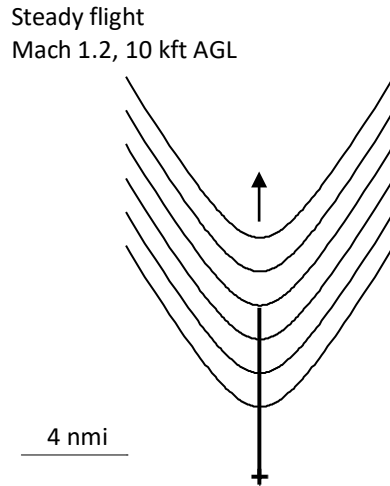
Figure 3-6: Sequence of Isopemps for Steady Level Flight (Top Down, 2-D View).

4g level turn
Mach 1.2, 10 kft AGL

4 nmi
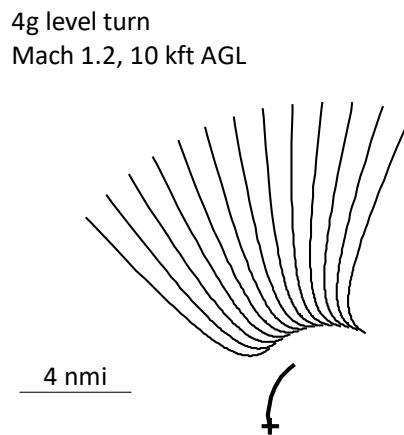
Figure 3-7: Sequence of Isopemps for Steady Supersonic Turn (Top Down, 2-D View).

Changes in Mach number can also cause focusing, which is typically illustrated by an accelerating aircraft. Figure 3-8 shows ray cones at two successive times for an accelerating aircraft. The acceleration causes a broadening of the ray cones and consequent reduction in the forward throw. As sketched in Figure 3-8, the isopemps intersect at points off to the side. At an earlier time, the isopemps intersected along the centerline. The intersection of isopemps represents a position where boom energy is concentrated or focused, as shown in Figure 3-9 for an aircraft accelerating to the east. Focus boom waveforms are U-waves rather than N-waves, both of which are sketched in Figure 3-10. Locations near and after a focus can have a combination of N-waves and U-waves, as shown in Figure 3-11. A detailed description of focusing can be found in Section 3.3.4.

13
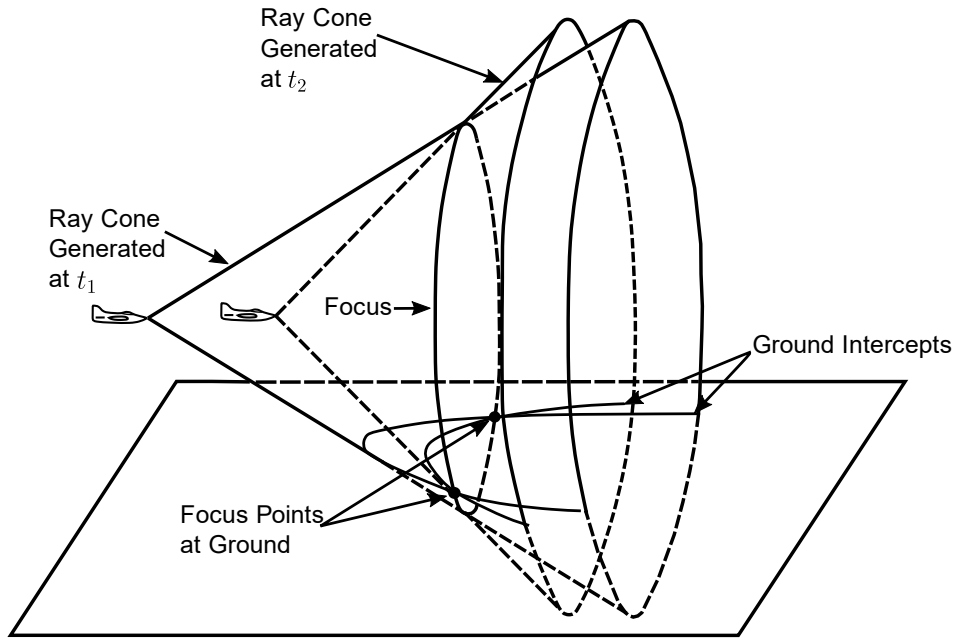
Figure 3-8: Acceleration Focus in Three Dimensions (*Source: Plotkin* [3]*, reprinted with permission*).

0.28 g level acceleration, 10 kft MSL
Ground at 2800 ft MSL
Standard atmosphere

4 nmi
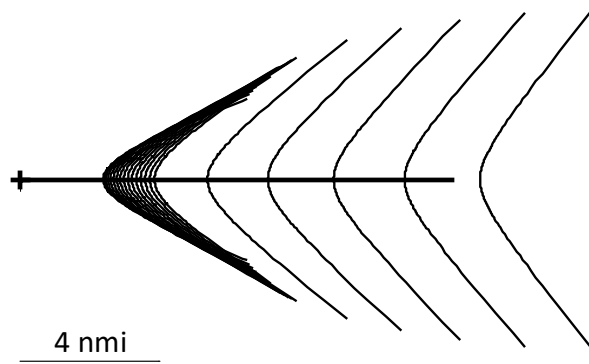
Figure 3-9: Sequence of Isopemps for an Accelerating Aircraft.
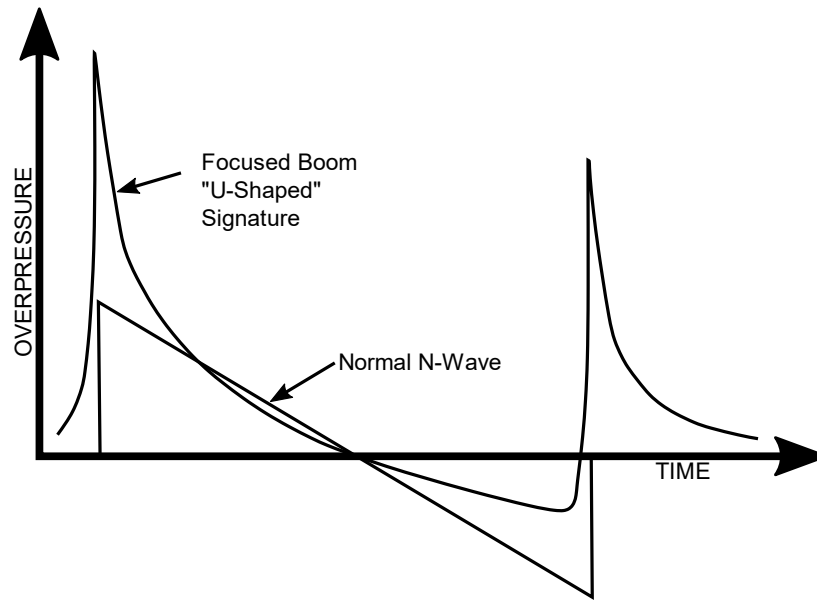
Figure 3-10: Focused and Nonfocused Boom Signatures (*Source: Plotkin* [12]*, reprinted with permission*).
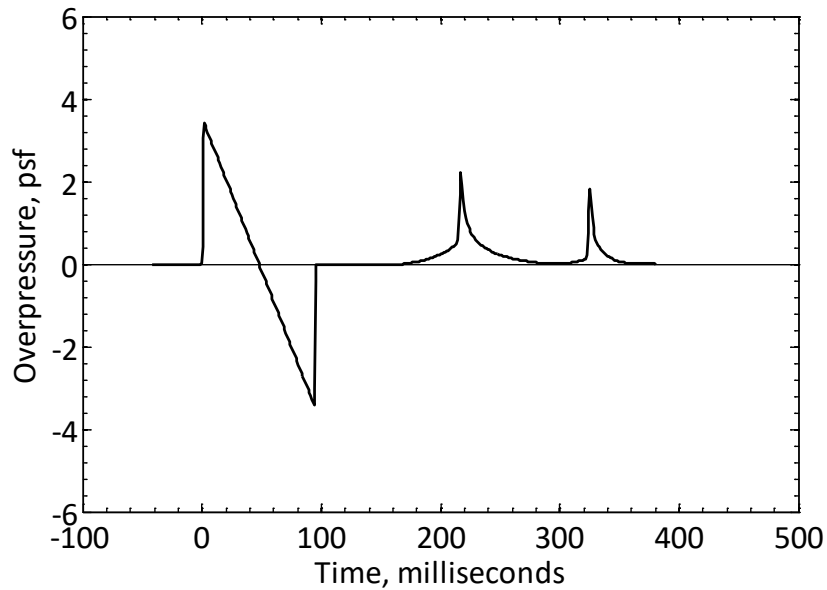
Figure 3-11: Combined N-wave and U-wave Near a Focus.

3.1 Vehicle Source Definition

As mentioned above, one needs to describe the vehicle source characteristics in order to conduct a sonic boom analysis. Depending on the level of analysis and complexity of the vehicle, there are several ways to describe the vehicle model in PCBoom including:

- Source Strength from Linearized Aerodynamics
- Source Strength from Measurement Data and CFD Solutions
- Launch Vehicle Plumes based on thrust and drag

3.1.1 Source Strength from Linearized Aerodynamics

Historically, sonic boom signature prediction has been established by Whitham [13] and Walkden [14] and is based on an *F*-function and equivalent area distribution. The linearized flow pressure field about a supersonic vehicle is given by

$$\delta p(\tau; \theta) = \frac{\gamma p_0 M^2}{(2\beta r)^{1/2}} F(\tau; \theta)$$

(1)

where

$$\begin{aligned}
\delta p &= \; p - p_o, \text{ pressure disturbance} \\
\tau &= \; x - \beta r, \text{ a distance parameter} \\
\theta &= \; \text{azimuthal angle in aircraft-fixed coordinates} \\
\gamma &= \; \text{ratio of specific heats} \\
p_o &= \; \text{ambient pressure} \\
M &= \; \text{Mach number} \\
\beta &= \; \sqrt{M^2 - 1} \\
r &= \; \text{radial coordinate} \\
F &= \; \text{source } F\text{-function.}
\end{aligned}$$

The source *F*-function can be expressed as

$$F(\tau; \theta) = \frac{1}{2\pi} \int_0^\tau \frac{A_e''(x; \theta)}{(\tau - x)^{1/2}} dx$$

(2)

where

$$x = \quad \text{axial coordinate parallel to flight path.}$$

A_e is the cross-sectional area of an equivalent body of revolution at the given azimuth θ and is given by[*]

$$A_e(x; \theta) = A_v(x; \theta) \frac{\beta}{\rho u^2} \int_0^x L(\xi; \theta) d\xi$$

(3)

where

$$\begin{aligned}
A_v(x; \theta) &= \quad \text{the normal projection of the area cut by a plane tangent to} \\
&\qquad \text{a Mach cone at the azimuthal angle } \theta \\
L &= \quad \text{the component of lift per unit length in the } \theta \text{ direction} \\
\rho &= \quad \text{ambient density} \\
u &= \quad \text{flight speed.}
\end{aligned}$$

For a slender axisymmetric body, $A_v(x; \theta)$ is the normal cross-sectional area at an axial station x. For an aircraft in a level flight or in a coordinated turn (i.e., lift is normal to the wingspan, "head up"), $L(x; \theta) = L(x) \cos \theta$, where $L(x)$ is the lift per unit length and θ is zero in the "head down" direction. The total lift is the integral of $L(x)$ along the length of the vehicle.

---

[*] Notation in this document is not entirely self-consistent. τ has been chosen to match the notation for this distance in diverse original sonic boom work. "A" is elsewhere used to denote ray tube area.

The function F was first introduced by Whitham [13, 15] and is referred to as the Whitham F-function. This was extended to wing-body combinations by Walkden [14] as Equation (3). A more detailed discussion of Equations (1) through (3) is contained in References [10, 12, 16, 17]. Equation (1) gives the initial source pressure signature as a simple scaling of the F-function.

PCBoom can be directly given an F-function or a pressure distribution $\delta p(x)$. When an F-function input is used, a different F-function should be provided for each azimuth and for each flight condition if maneuvers are involved. The program can accept a single F-function and adjust for azimuth and flight parameters according to scaling laws from Carlson's simplified model [9]. That model is a rough approximation, since it only adjusts the amplitude of the F-function and not its shape. The program can also generate simple "N-wave" F-functions from Carlson's model, based on vehicle weight, length, and shape category. Carlson's shape factor curves for various aircraft types are predefined within the PCBoom code and accessed via aircraft name. These operating modes are described in more detail in the PCBoom User's Guide, e.g., see Ref. [8].

The most general linearized flow source input is via aircraft geometry A_v and lift L with the F-Function computed from Equations (2) and (3). The F-function at a given azimuth θ is given by

$$F(\tau) = \frac{1}{2\pi} \int_0^\tau \frac{A''(x)}{(\tau - x)^{\frac{1}{2}}} dx. \tag{4}$$

$A(x)$ is one of the two elements on the righthand side of Equation (3). In general, $A(x)$ will be provided as values A_i at a sequence of axial stations x_i, with values between stations estimated from polynomial fits. Consider $A''(x)$ in the interval $[x_i, x_{i+1}]$ to be given by

$$A''(x) = C_i + D_i x + E_i x^2 \cdots. \tag{5}$$

Then, $F_i(\tau) = \int_{x_i}^{x_{i+1}} \frac{A''(x)}{(\tau - x)^{1/2}} dx$, the contribution of this interval to $F(\tau)$, can be expressed by

$$F_i(\tau) = C_i F_0(x_i, x_{i+1}, \tau) + D_i F_1(x_i, x_{i+1}, \tau) + E_i F_2(x_i, x_{i+1}, \tau) + \cdots \tag{6}$$

where

$$F_0(x_i, x_{i+1}, \tau) = \frac{1}{2\pi} \int_{x_i}^{x_{i+1}} \frac{1}{\sqrt{(\tau - x)}} dx, \tag{7}$$

$$F_1(x_i, x_{i+1}, \tau) = \frac{1}{2\pi} \int_{x_i}^{x_{i+1}} \frac{x}{\sqrt{(\tau - x)}} dx, \tag{8}$$

$$F_2(x_i, x_{i+1}, \tau) = \frac{1}{2\pi} \int_{x_i}^{x_{i+1}} \frac{x^2}{\sqrt{(\tau - x)}} dx, \tag{9}$$

etc.

Equations (7), (8), and (9) are straightforward to evaluate in closed form. Final expressions for F_0, F_1, and F_2 are embodied as statement functions in the program, providing up to a cubic fit to A''. It is possible to have a discontinuity in A' at a position x_i. This can occur when area versus distance has a sharp change in slope, or if the lift distribution has a jump. In that case, A'' is a delta function at x_i, and the contribution to the F-function is easily shown [12, 16] to be

$$\Delta F_i(x_i, \tau) = \Delta A_i' \frac{H(\tau - x_i)}{(\tau - x_i)^{1/2}} \tag{10}$$

where

$\quad F(\tau) = \quad$ is computed by summing the elements of Equation (10) over all intervals x_i from the aircraft bow through τ

$\quad \Delta A_i' = \quad$ the jump in A' across x_i

$\quad H = \quad$ the Heaviside step function.

When $F(\tau)$ is evaluated for τ at each input x_i, additional points up to one-half the body length beyond the last geometrically specified x_i need to be included. This extension is necessary to account for the tail wave recompression. The F-function may be scaled via Equation (1) to local flight conditions to provide a δp pressure distribution at an initial reference distance. One may also prescribe a normalized pressure distribution $\delta p / p_o$.

3.1.2 Source Strength from Measurement Data and CFD Solutions

The concept of using wind tunnel, flight test measurement, or CFD solutions instead of traditional area rule has been noted above. Obtaining and preparing such data are a significant endeavor. The equal-area rule does not give the actual pressure field close to the vehicle but provides an asymptotic result, valid at distances that are large compared to the vehicle length.

Area rule source strength is given by the Whitham–Walkden theory [13, 14, 15]. Note that in Equation (1), δp is a function of x and r, and Equation (1) states that the dependence separates in τ, r with a shape $F(\tau)$ (with no explicit dependence on r) and an amplitude proportional to $1/\sqrt{r}$. This is valid at large r. At small r, F is not independent of r, so this simple relation may not apply.

Equation (1) is a far-field acoustic form. Regardless of the linearity of the flow near the vehicle, if $\delta p / p_o$ is known at some radius and azimuth in the far field, then $F(\tau, \theta)$ is known and $\delta p / p_o$ can be computed at any other far-field radius[†] [18].

The critical issue is that the known $\delta p / p_o$ (either from wind tunnel, aircraft flight test probing, or CFD data) must be at a far-field radius. The far-field effect may be tested for conformity to the $1/\sqrt{r}$

[†] Since $\delta p / p_0$ and F differ only by a scale and radius factor, they are virtually interchangeable.

asymptotic behavior. If there is no conformity, then the θ and r dependence cannot be treated separately, and the far-field assumption is violated.

3.1.3 Launch Vehicle Plumes based on Thrust and Drag

As part of the Carlson N-wave *F*-function model [9], PCBoom accounts for the effect of underexpanded rocket exhaust plumes on the boom. A starting signature is defined by a Carlson N-wave *F*-function for the vehicle itself, immediately followed by an *F*-function for the plume. The Carlson parameters are based on a shape factor corresponding to Tiegerman's hypersonic boom theory with drag [19] corresponding to the engine's plume drag, and length from the Jarvinen-Hill universal plume model [20]. The engine thrust and plume drag are required input parameters.

3.2 Acoustic Propagation using Geometrical Acoustics

Sonic boom prediction consists of the nonlinear propagation of a nearfield aircraft pressure signature within a few aircraft body lengths along acoustic rays. The method is illustrated in Figure 3-12 where a nearfield signature propagates along a ray path. The shape of the acoustic signature is given by the *F*-function [15], $F(\tau)$, which is based on the aircraft's configuration and aerodynamic loads (Section 3.1). The acoustic rays are traced by the method of geometrical acoustics [21], which yields the position of the boom as well as its amplitude.

Figure 3-13 illustrates the coordinates of rays from their origin at the aircraft. Rays emanate from a ray cone that is orthogonal to the Mach cone. For a given flight path and atmospheric profile, there are two independent variables: the time t at the aircraft, and the azimuthal angle ϕ that identifies the ray of interest on that ray cone. The definition of ϕ is shown in Figure 3-13, with 0° being down in a vertical plane containing the flight path. The original Thomas code calculated the boom for a single combination of t, ϕ.
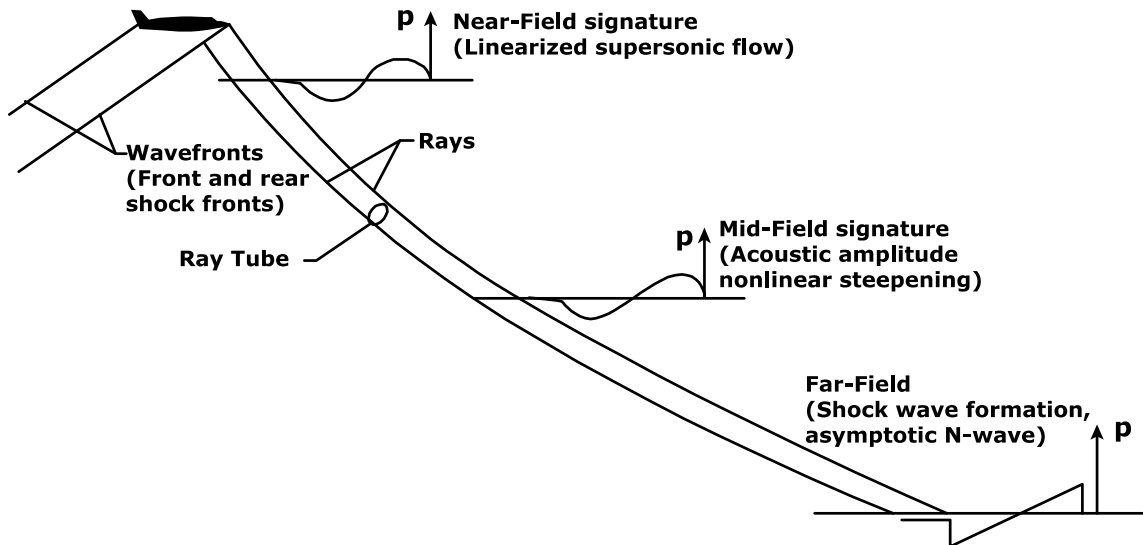
Figure 3-12: Schematic of Sonic Boom Propagation (*Source: NASA-SP-622* [7]).
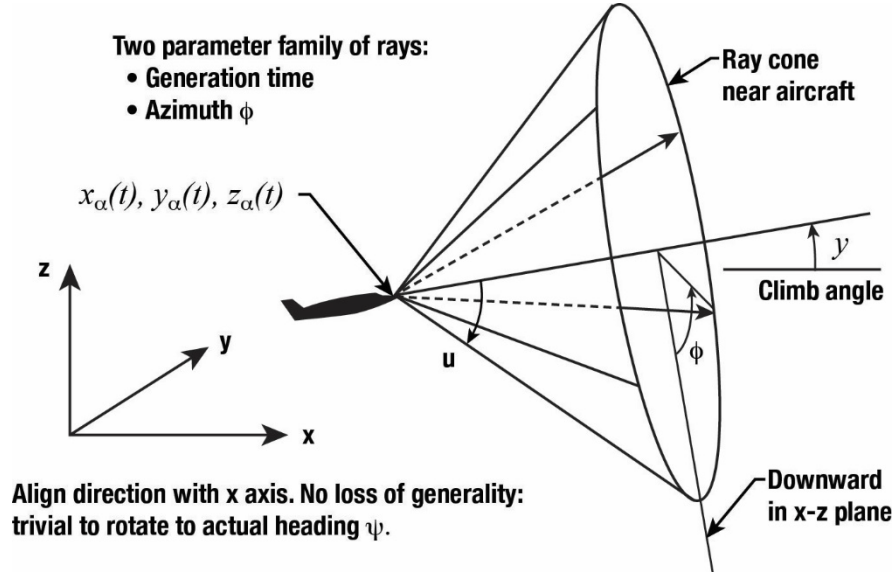
Figure 3-13: Initial Ray Cone and Coordinates (*Source: NASA-SP-622* [7]).

3.2.1 Acoustics and the Eikonal

A general description of ray tracing of sonic booms through a real atmosphere may be found in Maglieri et al., Chapter 6 [7]. The specific implementation in PCBoom is as follows. Consider a sound field in which the acoustic pressure at frequency ω can be written as

$$p(\vec{r}, t) = P(\vec{r})e^{i\omega\left(t - \frac{W(\vec{r})}{a_\infty}\right)}$$

(11)

where

$$P(\vec{r}) = \text{a spatial amplitude function}$$
$$W(\vec{r}) = \text{a phase function}$$
$$a_\infty = \text{a constant reference sound speed.}$$

The phase function $W(r)$ yields the loci of $\vec{r}$ called the wavefronts when $W(\vec{r})$ is constant.

The ray-theoretic acoustic equations can be obtained by linearizing (small amplitude acoustic perturbations assumed) the conservation equations of fluid mechanics but allowing for arbitrary gradients of the unperturbed quantities. If Equation (11) and similar expressions for velocity and density perturbations are substituted, one can solve for $P(\vec{r}), W(\vec{r})$ and the acoustic velocity, density, amplitude as well as the phase functions. In general, the resulting equations are difficult to solve and are frequency dependent. However, if the limit $\omega \to \infty$ is taken, such that the acoustic wavelength is small compared to the gradient scales of unperturbed flow quantities, we obtain [21]

$$\left|\vec{\nabla}W\right|^2 = \frac{q^2}{c_0^2}$$

(12)

and

$$\vec{\nabla} \cdot \left(\frac{P^2 a_\infty}{\rho_0 q a_0^2}\vec{V_s}\right) = 0$$

(13)

where

$$q = a_\infty - \vec{V}_s \cdot \vec{\nabla}W$$
$$\vec{V}_s = a_o \cdot \hat{n} + \vec{V}_0$$
$$\hat{n} = \frac{\vec{\nabla}W}{|\vec{\nabla}W|} \text{ is the unit vector normal to wavefronts}$$
$$\rho_0 = \text{local fluid density}$$
$$a = \text{sound speed}$$
$$\vec{V}_0 = \text{background flow velocity.}$$

Equation (12) is known as the eikonal equation. It shows that sound propagates along rays with velocity $\vec{V}_s$ and that successive wavefronts (surfaces of $W(\vec{r})$ constant) can be constructed using Huygens' principle. Equation (13) has the form of a conservation equation along rays. Following the development described by Pierce [22], it may be integrated to give the following acoustic amplitude relation

$$\frac{p^2 V_s A}{\rho_0 q a_0^2} = const \tag{14}$$

where

$$A = \text{the normal cross-sectional area of a ray tube (bundle of rays, analogous to a}$$
$$\text{fluid dynamic stream tube being a bundle of streamlines).}$$

Equation (14) is commonly referred to as the Blokhintzev invariant. Note that in the presence of wind, the ray trajectories, along which acoustic energy is transported, are not necessarily normal to the constant phase wavefront surfaces.

Ray paths are computed by direct numerical integration of the eikonal. The difference form for a horizontally stratified atmosphere over a flat Earth is given in Reference [18]. A simple linear predictor/corrector technique is used in PCBoom: an initial ray unit vector is taken as normal to the Mach cone at the aircraft, which is moving relative to the local wind. A ray tube is constructed from four corner rays, as sketched in Figure 3-14. The rays are separated by small but finite angle $\Delta\phi$ (azimuth ϕ about the flight path is relative to a vertical plane) and time Δt along the trajectory. The initial ray angles at the different time steps account for the differences in the flight path angle, heading, and Mach number, so that the ray tube implicitly accounts for maneuver effects. The integration time steps along all four rays are coordinated such that a given number of steps correspond to a constant phase on all four rays.
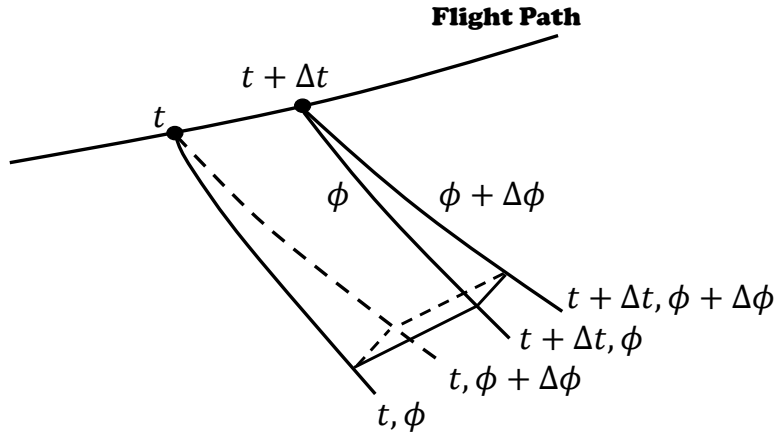
Figure 3-14: Ray Tube Outlined by Four Corner Rays Δt and $\Delta\phi$ apart (*Source: Plotkin* [12]*, reprinted with permission*).

An approximation to the cross-sectional area of a ray tube is obtained by first taking the cross-product of vectors diagonally across the tube ($\{t, \phi\}$ to $\{t + \Delta t, \phi + \Delta\phi\}$ and $\{t, \phi + \Delta\phi\}$ to $\{t + \Delta t, \phi\}$ in Figure 3-14). This gives a quantity proportional to the area of the constant phase surface. The dot product of this quantity and the local unit vector of the t, ϕ ray is then formed to yield a quantity proportional to the normal cross-sectional area.

3.2.2 Stratified Atmospheric Effects including Winds

Traditionally, sonic boom propagation is computed for a horizontally stratified atmosphere over a flat Earth. This is generally an acceptable approximation for primary downward-propagating booms, where propagation distances are short relative to Earth's curvature and typical vertical atmospheric gradients. There has been recent interest in long range sonic boom propagation, both for an analysis of boom carpet edges from high altitude flight and for an analysis of over-the-top booms. If long distances are involved, the effect of the Earth's curvature is not necessarily negligible. PCBoom has the capability to accommodate a full three-dimensional atmosphere and a nonflat Earth [5].

A nearfield signature may be represented in acoustic form as

$$\delta p = p_0 \frac{F(\tau)}{\sqrt{B}} \qquad (15)$$

where

$\qquad B = \quad$ the Blokhintzev invariant, an amplitude parameter combining the scaling factors in Equation (1).

As long as one can compute ray paths and B, subsequent propagation and steepening effects apply. The crux of the extension is implementing an appropriate ray tracing scheme.

A full 3-D ray tracing in an Earth fixed geocentric (EFG) coordinate system, illustrated in Figure 3-15, has been implemented in PCBoom. Latitude λ is the elevation angle of a line that is normal to the local surface of the ellipsoid, and longitude θ is the angle east of the x axis. The piercing point of the x axis is the prime meridian at the equator. Aircraft location and flight parameters, as sketched in Figure 3-15, are generally defined in terms of local flat Earth, i.e., λ, θ (or local tangent plane x, y that can be related

22

to λ, θ) and altitude h above the ellipsoid (or above the geoid, which has a known offset to the ellipsoid). Calculation of EFG coordinates from λ, θ, and h is explicit and straightforward. Once an initial ray position and vector are obtained, the ray path may be computed in Cartesian EFG coordinates. To enable the calculation of λ, θ, and h from EFG coordinates, an efficient, robust algorithm developed by Sofair [23, 24] is available and has been coded into PCBoom. A more detailed description of this implementation is described in Plotkin et al. [5].
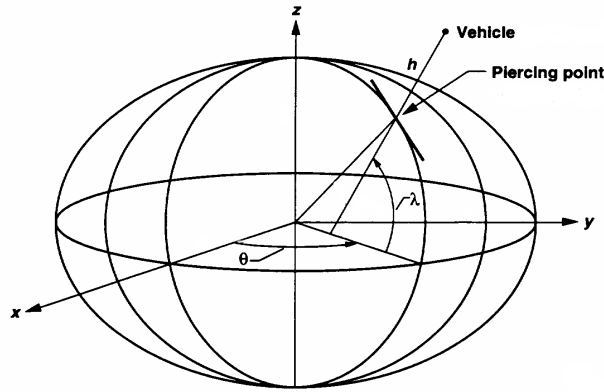
Figure 3-15: Ellipsoidal Earth and Coordinate Systems (*Source: Plotkin, Page, and Haering* [5]*, reprinted with permission*).

3.2.3 Secondary "Over the Top" Booms

Secondary ground booms originate from upward propagating rays (see Figure 3-2) that turn in the upper atmosphere and eventually intersect the ground. These have been associated with many supersonic flight programs [7], most notably the Concorde [25]. Such booms are also referred to as "Over-the-Top" booms, or OTT booms. Figure 3-16 illustrates three kinds of ray paths that are usually considered when discussing primary and OTT booms. The primary boom is the direct path from the aircraft to the ground. The existence of any OTT boom requires upper air wind and/or sound speed such that the maximum effective horizontal propagation speed is greater than the aircraft's flight speed. The criterion is essentially the same as for the cutoff of primary booms. Given this condition, there are two categories of OTT booms. Type I emanates from the top of the aircraft and refracts downward. Type II reflects from the ground then comes back down as a skip. Secondary sonic booms observed in New England from Concorde operations were Type II [25].
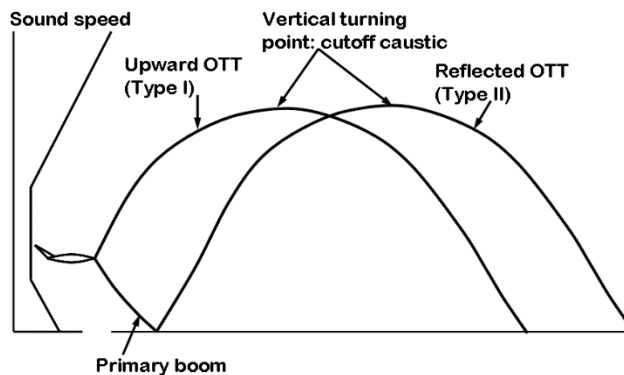
Figure 3-16: Primary Boom Ray Path and Two Types of Over the Top (OTT) Ray Paths
(*Source: Plotkin, Page, and Haering* [5]*, reprinted with permission*).

It is common to see sketches like Figure 3-16 with a primary zone noted around the primary boom's ground intercept, a secondary zone indicated around the OTT boom ground intercepts, and a no-boom gap in between. This situation can happen but is not necessarily universal. Consider the ray cone of an aircraft in level flight, as shown in Figure 3-17: Rays at the bottom may reach the ground as primary booms. Here the ground intercept will be hyperbolic between lateral cutoff angles. Rays at the top are potential Type I OTT booms, given appropriate upper air conditions. Next consider the same aircraft but in a dive steep enough that all rays on the cone are directed downward. In Figure 3-18, all rays may reach the ground in a full ellipse with no lateral cutoff, and are all primary booms [5].
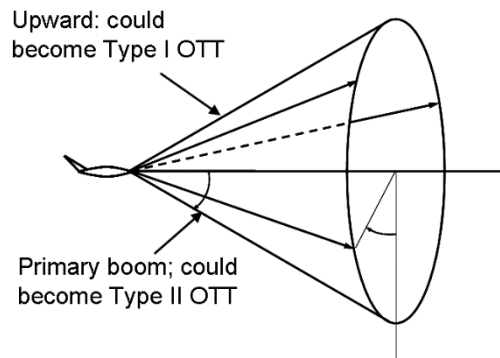
Figure 3-17: Ray Cone in Level Flight (*Source: Plotkin, Page, and Haering* [5]*, reprinted with permission*).
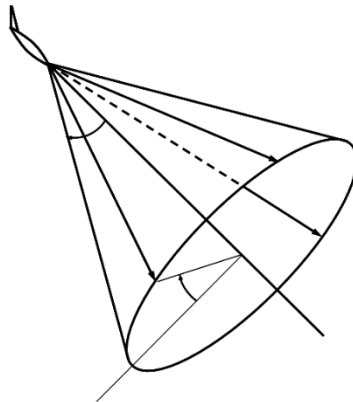
Figure 3-18: Ray Cone in Diving Flight (*Source: Plotkin, Page, and Haering* [5]*, reprinted with permission*).

Finally, compare these two sketches with rays in a refracting atmosphere, sketched in Figure 3-19. On the left are vertical cuts through the ray cones of Figure 3-17 and Figure 3-18. On the right is a vertical cut through a ray cone from an aircraft in level flight with strong downward refraction. The lower ray is primary. The upper ray could be considered to be a Type I OTT boom, as sketched in Figure 3-17. But it could also be considered primary, as sketched in Figure 3-18.
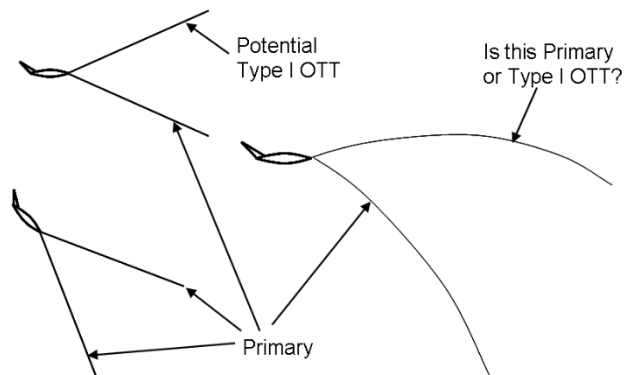
Figure 3-19: Sketch of Straight Rays in Level and Diving Flight, and Curved Rays under Inversion Conditions
(*Source: Plotkin, Page, and Haering* [5]*, reprinted with permission*).

Whether a ray may be considered as primary or secondary depends on its relationship to other rays. The three-dimensional nature of booms, with rays emanating from a full cone, must be recognized. The potential for a given ray to reach the ground, either as primary or OTT, depends on its initial vertical angle. Rays at various ϕs around the cone are at different vertical angles. Under the aircraft, higher Mach numbers cause rays to be directed more downward, so that they are more likely to reach the ground. Above the aircraft, higher Mach numbers cause rays to be directed more upward, making them less likely to turn back toward the ground. To the side, rays are emitted at shallower vertical angles and may do neither.

The rays sketched in Figure 3-16 represent conditions where rays at $\phi = 0°$ will reach the ground and return as Type II OTT booms, whereas rays at $\phi = 180°$ will be turned downward as apparent Type I OTT booms. At other ϕ values, however, the initial inclination of the rays will be closer to the horizontal. They may or may not reach the ground. If they do not reach the ground, they can be trapped in a channel. Additional examples can be found in Reference [5].

Knowledge of OTT booms is somewhat limited compared with primary sonic boom due to comparatively limited observations and measurements [7]. Propagation of OTT booms involves both the upper and lower levels of the atmosphere and may involve propagation distances greater than 100 miles. OTT booms typically have small peak overpressures when they reach the ground and are characterized by low frequency contents and attenuated high frequency energy.

OTT ground boom signatures are also distinct from primary boom signatures. Maglieri et al. [7] described the characteristics of secondary booms in the context of measurements reported by Rickley and Pierce [25]. One measured Concorde secondary boom signal is shown in Figure 3-20(a) and exhibits multiple periods of disturbances over a total of around 90 seconds. Note that the maximum peak-to-peak overpressure is approximately 0.2 psf, whereas a primary boom from the Concorde might reach 2.0 psf [7]. Zooming in on the three regions labeled A, B, and C in Figure 3-20(b) to Figure 3-20(d), respectively, no sharply rising shocks are observed, indicating that the spectral composition of the secondary boom is characterized by low frequencies. Because the fundamental frequencies of these secondary booms appear to be on the order of a few Hz, they may be inaudible or may manifest more likely as rumbles or vibration [7].

(a) Overall time pressure history

(b) Expanded pressure time history A.

(c) Expanded pressure time history B.

(d) Expanded pressure time history C.
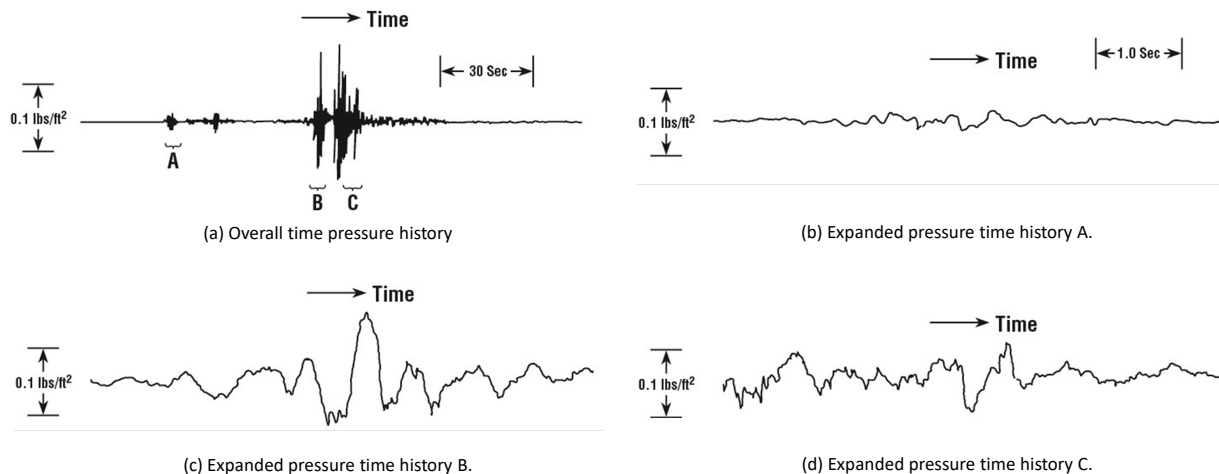
Figure 3-20: Characteristics of Secondary Sonic Booms (*Source: NASA-SP-622* [7]).

At higher Mach numbers under an atmospheric condition, it is possible for ray angles to be sufficiently steep so that the rays do not return to the ground as secondary booms. As with the determination of lateral cutoff, aircraft trajectory and atmospheric conditions determine which rays reach the ground. Functionally, this analysis can be considered in separate steps: evaluating the ray paths to determine which ones reach the ground; evaluation of the primary and secondary boom carpets; and calculation of the signature evolution along the ray paths. The ground bounce for Type II and turning points for Types I and II over-the-top rays require additional considerations.

Focusing of sonic booms can occur due to aircraft maneuvers or atmospheric gradients. A common consequence of modeling propagation using the geometric acoustics is for the ray tube area to vanish at a focus. The focus is distributed along a line referred to as a caustic [5]. This phenomenon is described in greater detail in Section 3.3.4; Figure 3-34 is one example showing sonic boom focusing at the caustic line associated with a steady flight at a speed below the cutoff Mach number. As that figure shows, rays refract away from the ground at the caustic altitude.

An analogy to the Mach cutoff caustic line can be drawn to the point where upward propagating OTT rays turn back toward the ground; this so-called turning point as well as the caustic passage are described in detail in Reference [5] and are summarized here. As with a Mach cutoff caustic line, the turning point also involves a caustic passage where the ray tube area approaches zero. These regions push the limits of the geometric approximation. If the ray tube area analytically approaches zero, the pressure amplitude approaches infinity, which is not physically realistic.

Other factors to consider are diffraction effects due to the caustic acting like an edge, and finite boom amplitude limits through the caustic. The maximum boom at the caustic is distorted into the form denoted "maximum focus U" in Figure 3-34. The distortion is a result of diffraction at the caustic edge: diffraction effects are frequency dependent where the amplitudes of the high-frequency components are affected differently than the lower-frequency components in the incoming waveform. Nonlinear propagation effects should be accounted for, following the mathematical description of the nonlinear Tricomi equation (e.g., see the review by Plotkin [12]). To summarize, while propagation modeling (using the same techniques as those used for nonfocused sonic booms) could be used to calculate the signatures entering the caustic passage, those methods will not capture all the important effects through the focal zone.

26

To account for the important physical processes affecting signature evolution in the caustic passage, it is necessary to restart propagation and signature aging after the passage using the post-focus signature as described by Plotkin et al. [5]. Post-focus propagation could continue using the same method as used for booms approaching the focal zone or for primary booms. Boom propagation models such as Zephyrus [26] and TRAPS [27] can both handle propagation of OTT booms, but these models approximate post-focus signatures as U-waves generated by applying a Hilbert transform to the incoming signature through the caustic. Thus, the actual focusing is not calculated and energy loss through the caustic is not rigorously accounted for. As an illustrative example of signature modeling through a focus, the work of Kandil and Khasdeo [28] shows post-focus signatures modeled via a numerical solution of the conservative lossless nonlinear Tricomi equation, including cases where incoming waves were N-waves and modeled after Concorde source characteristics.

Another propagation issue is encountered when modeling the ground bounce for Type II OTT booms. Interaction of a boom with a ground surface will affect the amplitude and phase of the reflected boom signature, as described in greater detail in Section 3.4.4. Similar to modeling OTT booms through focal zones, propagation modeling involving a ground bounce needs to account for ground reflection effects.

3.3 Signature Evolution

Signature evolution through the atmosphere was presented graphically in Figure 3-1. With PCBoom, there are two different methods encompassing three fundamental run streams for computing the evolution of the waveform as it propagates through the atmosphere:

- Lossless signature evolution with application of a post processed simple shock thickness model, referred to in this technical reference as the lossless Burgers solver lossless Burgers

- Burgers evolution computed as a postprocessor, with or without the visualization features for interactive examination of the signature evolution, referred to in this technical reference as the Legacy Burgers solver or Legacy Burgers

- Burgers solver with incorporation of Doppler shift and absorption coefficient scaling due to the convective index of the wind, referred to in this technical reference as the Enhanced Burgers solver or Enhanced Burgers

3.3.1 FOBoom Lossless Signature Evolution

One important facet of sonic boom propagation is the aging and steepening of signatures. Each portion or phase of a signature advances proportionally to its amplitude. The phase speed is given by the local sound speed plus the perturbed increment δa to the local sound speed and the acoustic velocity δu. Thus, it can be represented by

$$\text{phase speed} = a_0 + \delta a + \delta u = a_0\left(1 + \frac{\gamma + 1}{2\gamma}\frac{\delta p}{p_0}\right) \tag{16}$$

where

$\delta p =$ the acoustic overpressure
$p_0 =$ the ambient pressure.

Each element of the waveform advances proportionally to its original pressure, as sketched in Figure 3-21.
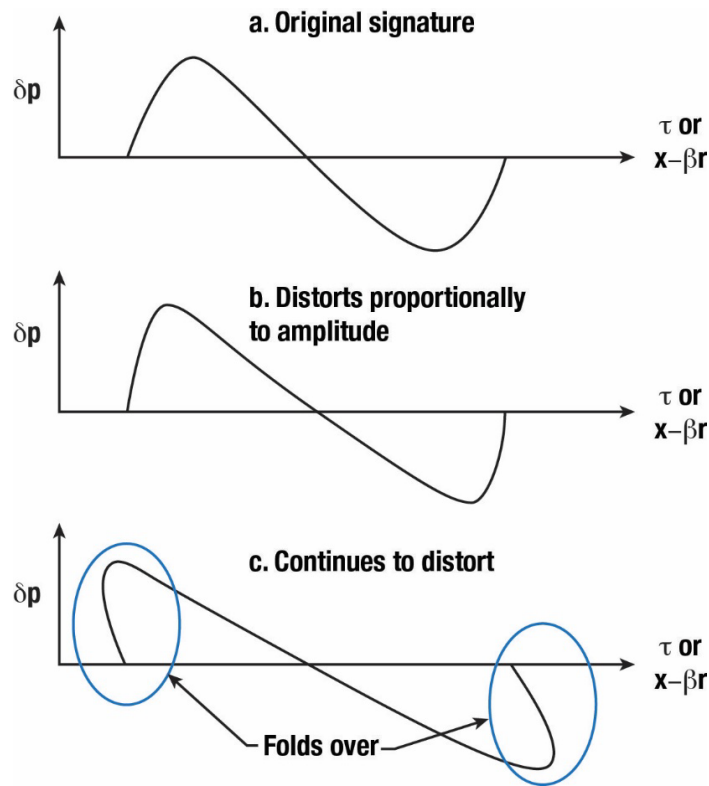
Figure 3-21: Evolution and Steepening of Sonic Boom Signatures (*Source: NASA-SP-622* [7]).

Figure 3-21 shows portions of the wave folding over. This does not happen in reality. When a portion of the wave approaches vertical, the gradient is strong enough that loss mechanisms become significant, and a shock forms, with a balance between steepening and losses.

The phase speed given by Equation (16) represents the isentropic speed of a quasilinear wave. When a shock forms, its propagation speed is governed by the Rankine–Hugoniot equations. For weak shocks, this speed is the average of the isentropic speeds corresponding to δp just before and just after the shock. This leads to a simple "area balance" construction for insertion of shocks. The aging process is implemented in two steps: an "age parameter" that defines how much the wave has steepened, and a general area balance procedure.

3.3.1.1 The Age Parameter

Sonic boom propagation may be summarized by the linear solution [15] provided in Equation (15) and is repeated here as

$$\delta p = p_0 \frac{F(\tau)}{\sqrt{B}}$$

(17)

for convenience. The arrival time t of a linear acoustic wave element τ at distance *s* along a ray is

$$t = \tau + \int_0^s ds/a_0. \tag{18}$$

For a weak, nonlinear wave with a phase speed given by Equation (16), the arrival time becomes

$$t = \tau + \int_0^s \frac{ds}{a_0 \left(1 + \dfrac{\gamma + 1}{2\gamma} \dfrac{\delta p}{p_0}\right)}, \tag{19}$$

which can be approximated by

$$t = \tau + \int_0^s \frac{ds}{a_0} - \int_0^s \frac{\gamma + 1}{2\gamma} \frac{\delta p}{p_0} \frac{ds}{a_0} \tag{20}$$

by virtue of $\frac{\gamma+1}{2\gamma} \frac{\delta p}{p_0}$ being small compared to 1. Substituting Equation (17) into (20) yields

$$t = \tau + \int_0^s \frac{ds}{a_0} - F(\tau)\Lambda(s) \tag{21}$$

where

$$\Lambda(s) = \frac{\gamma + 1}{2\gamma} \int_0^s \frac{ds}{a_0\sqrt{B}} \tag{22}$$

is known as the age parameter [29]. The signature evolution sketched in Figure 3-21 depends only on this parameter (which is a function of s along a given ray) and the *F*-function of each element.

Equation (21) represents the results of isentropic propagation, and leads to multiple-valued regions, which are equivalent to characteristics crossing. The crossing represents shock formation at which a shock is fit at the angle bisecting the characteristics before and after it. Once Equation (21) predicts an infinite slope at a point, that point is replaced with a shock whose speed is the average of the isentropic speeds ahead of and behind it. This shock traveling at the average of the isentropic speeds on either side uses the "area balance" rule for shock fitting. The folded-over construction sketched in Figure 3-21 demonstrates this property including the overlaps in the bottom sketch. A shock is inserted into each overlap region, positioned such that the overlap area ahead of it equals the area behind. The overlap regions are then deleted. The process can be intricate for complex signatures, but a systematic algorithm, described in the next section, is available and has been implemented in PCBoom.

3.3.1.2 Area Balance and Middleton–Carlson–Hayes Method
A general signature folding and shock fitting method was developed by Middleton and Carlson [30] used in the ARAP [29] and TRAPS [27] codes. The key to this method is a function that is the integral of the original *F*-function, so that the area balance of the *F*-function corresponds to crossings in the integral function. The process and rules for handling multiple crossings are described by Middleton and Carlson, who give a step-by-step procedure for its application.

Integrate area of original F-function:

$$S_0 = \int_0^x F(\tau)d\tau$$

Integrate area of aged (strained) F-function:

$$\tau_1 = \tau \text{ - } F(\tau)\Lambda(s)$$

$$S_0 = \int_0^x F(\tau)d\tau_1$$
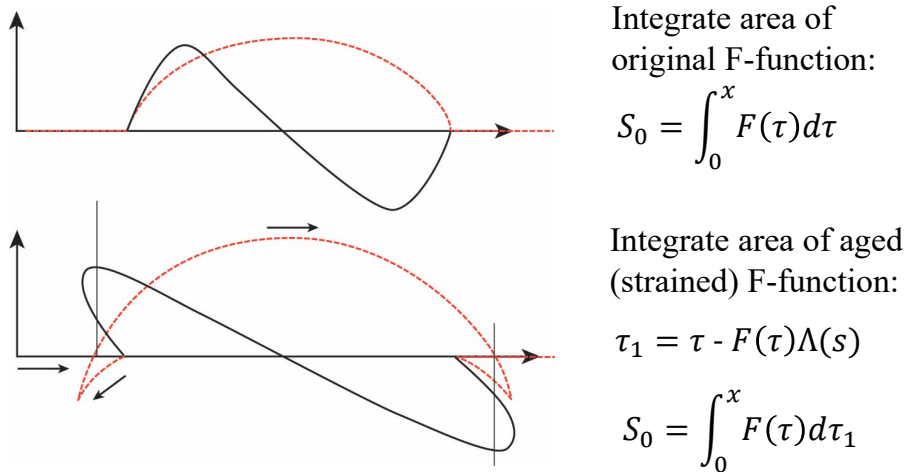
Figure 3-22: Middleton–Carlson Aging Method (*Source: NASA-SP-622* [7]).

Figure 3-22 illustrates the key elements of signature folding. A function S_0 is computed as the integral of the *F*-function. It is aged and steepened along with the *F*-function, and then develops folded regions. Each region can be determined to be "forward running" or "backward running." The crossing points of overlapping forward running segments define shock locations. The formation of shocks in more complex signatures is generally treated by the area balance rule: perform the construction defined by Equation (21), then resolve multiple-valued (multivalued) regions by noting that the area of the aged *F*-function must equal that of the original. A more brute force method identifies multivalued regions and then iteratively locates shocks within each region by computing the areas of the overlapping lobes before and after the shock. A similar method consists of drawing lines at the slope across the original *F*-function and iterating them up and down until the area is balanced.

Middleton and Carlson [30] developed a more direct method by working with the integral of the *F*-function

$$I(\tau) = \int_0^\tau F(\tau)d\tau \tag{23}$$

where

τ = the aged coordinate from Equation (21)
I = *F*-function integral.

Area balancing dictates that I is conserved as shocks form. Because the coordinate τ is strained according to the age parameter and *F*-function value, it can evolve in the forward or backward direction. Middleton and Carlson wrote I in a piecewise manner, between successive maxima and minima of τ. Segments may overlap with shocks located at the intersections of overlapping segments. Regions with multiple overlapping segments (representing coalescence) can occur. Rules are provided for the selection of the proper shock in each case.

Hayes et al. [29] noted that it was not necessary to consider the integral I in a piecewise fashion, where they retained the notation for the original τ and used a strained version representing the difference as given by

$$\tau_1 = \tau - \Lambda(s)F(\tau). \tag{24}$$

The equivalent of Equation (23) at propagation distance $s = 0$ is

$$S_0(\tau) = \int_{-\infty}^{\tau} F(\tau')d(\tau') \tag{25}$$

where

$-\infty = $ the beginning of the indefinite integral far enough forward so that any steepening is covered.

At a propagation distance s, Hayes et al. [29] then showed that the equivalent to Equation (23) is

$$S(\tau_1) = S_0(\tau) - \frac{(\tau_1 - \tau)^2}{2\Lambda(s)}. \tag{26}$$

Equation (26) yields the same result as the Middleton and Carlson's procedure, but has the benefit of not having to identify the reversal points while performing the integration. The reversal points are readily found by testing the difference between successive values of τ_1.

Computing S as described by Hayes et al. [29] is simpler than the multiple I process provided by Middleton and Carlson. However, subsequent description of identifying shocks by Hayes et al. is more complex, as it relies on a graphical interpretation and does not provide rules for multiple overlap regions. As a result, the original ARAP program [29] did not fully automate the shock insertion process, but rather output the overlap information and left it to the user to insert the shocks. In contrast, Middleton and Carlson provide a complete procedure using the following steps:
- Compute the curve given by Equation (26);
- Remove backward-running segments;
- Identify all intersections between the remaining segments;
- Insert shocks at each intersection;
- For multiple shocks within a segment, follow a hierarchal procedure defined in [30] to identify the proper shock(s).

The lossless Burgers solver implements aging by using Equations (24) through (26) and applying the Middleton and Carlson's shock fitting procedure with Hayes formulation. The amplitude is then scaled via the Blokhintzev invariant. This procedure permits a rapid calculation of the signature at any position along the ray. The age and amplitude parameters may optionally be written to an output file and used to compute booms for other configurations at the same flight and atmospheric conditions.

3.3.1.3 Simple Shock Thickness Model
Traditional sonic boom theory (also referred to as thin or weak shock theory) treats shock waves as infinitesimally thin pressure jumps, governed by the Rankine–Hugoniot conditions. The detailed mechanism within a shock, first studied by Taylor [31], is a balance between nonlinear steepening and thermoviscous absorption, which does not account for the absorption due to the molecular vibrational relaxation. This balance is governed by the Burgers equation. This is a second-order, nonlinear expansion of the Navier–Stokes equations. Although it is applicable to plane (or locally plane) wave

31

propagation, it can easily be adapted for nonplane wave propagation (as described in Section 3.3.2). It was first published by Bateman [32], and its properties were studied extensively by Burgers [33]. There is extensive literature of its properties and methods of solution, notably in the papers by Hamilton and Blackstock [34], Hopf [35], and Cole [36].

For the classical thermoviscous absorption, the Burgers equation can be written as

$$\frac{\partial P}{\partial t} + \frac{\gamma + 1}{2\gamma} \frac{a_\infty}{p_\infty} P \frac{\partial P}{\partial \xi} = \frac{2}{3} \gamma \eta \frac{\partial^2 P}{\partial \xi^2}$$

(27)

where

$$P(\xi, t) = \text{pressure signature}$$
$$\xi = x - a_\infty t = \text{wave-fixed coordinate}$$
$$x = \text{length coordinate in propagation direction}$$
$$t = \text{time}$$
$$a_\infty = \text{ambient sound speed}$$
$$p_\infty = \text{ambient pressure}$$
$$\gamma = \text{ratio of specific heats}$$
$$\eta = \text{kinematic viscosity.}$$

The first term, $\frac{\partial P}{\partial t}$, is the substantial derivative in the momentum equation. The second term is equivalent to the steepening that, for sonic booms, is addressed by the Whitham's rule [15]. The term on the right represents thermoviscous losses. As a second derivative, it represents losses that are proportional to frequency squared. The solution to Equation (27) for a steady shock is [35, 36]

$$p(t) = \frac{\Delta p}{2}[1 + \tanh(t/T)]$$

(28)

where

$$\Delta p = \text{the shock jump}$$
$$T = \text{is rise time determined by the maximum slope.}$$

T is proportional to the coefficient of viscosity and inversely proportional to Δp, i.e., $T \propto \eta / \Delta p$. The hyperbolic tangent term in Equation (28) is sometimes referred to as the Taylor shock structure.

Flight test data in the 1960s showed thatsonic boom rise times were three to five orders of magnitude greater than predicted from classical viscosity and exhibited great variability under turbulent atmospheric conditions. A turbulent scattering theory was developed, which qualitatively agreed with the flight test data [37].

Figure 3-23 is a compendium of rise times from measured shock waves, turbulence thickening theory, and molecular relaxation absorption theory. The measurements around 10 psf (lb/ft^2) are blast waves; the rest are sonic booms. All fit an inverse Δp relationship but with a wide range of effective viscosity values. The appropriate curve to take for nominal rise times is through the lower portion of flight test data, with longer rise times considered an additional effect from turbulence. The nominal rise time at 1 psf is 1 ms, so the rise time relation to use in Equation (28) is

$$T = 1 \text{ ms} / \Delta p$$

(29)

where

$$\Delta p = \quad \text{shock jump in psf.}$$
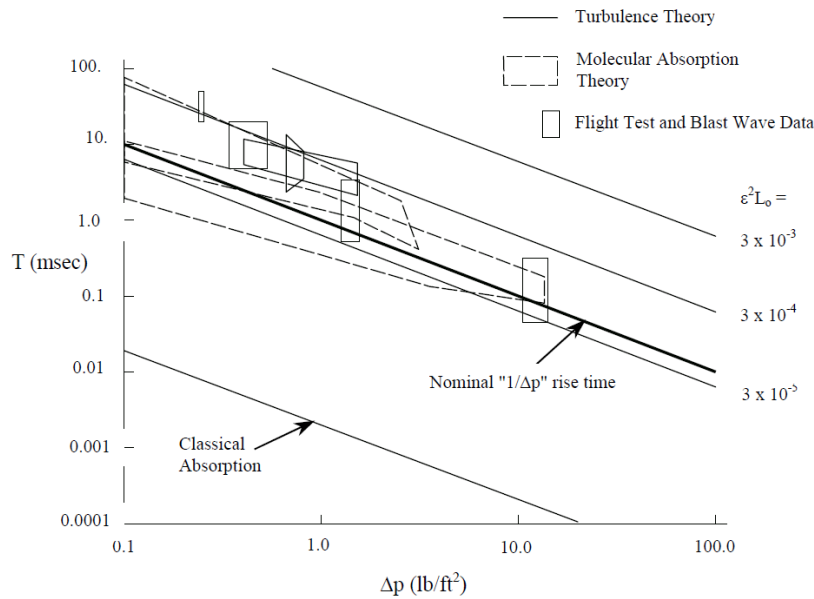
Figure 3-23: Shock Wave Rise Time, Theories, and Applied Model (*Source: Adapted from Plotkin* [12]*, reprinted with permission*).

This turbulent scattering theory did not entirely explain the long rise times under calm conditions. Further developments of the sonic boom theory to include molecular vibrational relaxation absorption [38] show that this effect can be several orders of magnitude greater than the classical thermoviscous effects.

The lossless Burgers solver in the main ray tracing and signature aging module, FOBoom, calculates signatures according to thin shock theory. The signature and footprint post processing module PCBFoot applies the Taylor shock structure given by Equation (28) to each shock in the signature.

3.3.2 Burgers Solvers

In addition to the lossless Burgers equation discussed in Section 3.3.1, the sonic boom waveform evolution may be more accurately predicted in PCBoom using either of the two lossy Burgers equation solvers incorporated into the model. The Legacy Burgers Solver (Sec 3.3.2.1) is the solver that has been included in prior releases of PCBoom and implemented in the PCBurg and HeadlessBurgers modules. The Enhanced Burgers Solver (Sec 3.3.2.2) is a recent update to PCBoom and is implemented within the FOBoom module using the keyword BURGERS followed by a line of relevant input parameters for the Solver. The Enhanced Burgers Solver is significantly faster than the Legacy Burgers Solver. It also includes additional physical effects that are not included in the Legacy Burgers Solver as outlined in the following sections.

3.3.2.1 Legacy Burgers Solver

The governing equation for the Legacy Burgers Solver is given [39] by

$$\frac{\partial p}{\partial t} + \frac{\gamma + 1}{2\gamma}\frac{c_o}{p_o}p\frac{\partial p}{\partial \xi} = [\alpha(\omega)p^T]^{-T} \qquad (30)$$

where

$p =$ acoustic pressure
$\gamma =$ the ratio of specific heats of gases
$c_o =$ the sound speed
$p_o =$ atmospheric pressure
$\xi =$ $s - c_o t$, the coordinate moving with the boom directed along the propagation path with length s
$\omega =$ angular frequency
$a =$ the total frequency-dependent atmospheric absorption
$[...]^T =$ Fourier Transform operator
$[...]^{-T} =$ inverse Fourier Transform operator.

Note that Equation (30) is written in a frame moving with the wave. The wave is moving along the propagation direction and evolves with time. When considering the waveform, it can be viewed as a function of ξ at a particular time t. It can also be viewed as a function of time as it passes a particular ray path length s. The second perspective is useful because that is how sonic booms are usually measured: a microphone is placed at some location. As part of ξ, t represents a coordinate along the waveform. By itself, as the other parameter of p, t represents the time the wave has propagated. Care must be taken to not confuse the two roles of t.

Although the Burgers equation used by the Legacy Burgers solver has been historically represented by Equation (30), the actual equation used in this solver has been adapted to appropriately account for 3D effects. Equation (30) only applies to 1D propagation, as it does not account for the geometrical spreading effects, whereas sonic boom propagation is inherently 3D. To account for the spreading effects, Equation (30) is modified to include the ray tube area discussed in prior sections. Specifically, at each discrete propagation step within PCBoom, the acoustic pressure is multiplied by the factor

$$B^{(L)} = \sqrt{\frac{c_{n,r}^2\, p_o\, c_o\, A_r}{c_n^2\, p_{o,r}\, c_{o,r}\, A}} \qquad (31)$$

where

$c_n =$ $c_o + \overrightarrow{v_o}\cdot\hat{n}$
$A =$ the ray tube area
subscript $r =$ represents values at the reference (initial) time step.

More detail on the implementation is given in Section 5.1.

3.3.2.1.1   Absorption Effects in the Legacy Burgers Solver

The absorption α in Equation (30) is the sum of the classical-rotational absorption and the vibrational relaxation of nitrogen and oxygen molecules present in the atmosphere. This is a further generalization compared to Equation (27), which only includes attenuation by classical mechanisms. Relaxation effects are more complex than a simple f^2 behavior of classical absorption, but it is well defined by current ANSI/ASA standard [38].

The units of α must be consistent with Equation (30); as written, α should be in units of nepers per second to denote energy loss. In PCBoom, the absorption from the ANSI/ASA standards, given in nepers per meter, is multiplied by the local sound speed, resulting in the absorption unit nepers per second. It should be noted that α can be complex, with the real part (always negative) being attenuation and the imaginary part being dispersion. This is the methodology utilized in PCBoom modules PCBurg and HeadlessBurgers. For general absorption, where $\alpha(\omega)$ is not a simple function, numeric solution of Equation (30) is necessary. There are time domain solutions for the steady case [40, 41]. Implementation details of absorption effects in PCBurg and HeadlessBurgers can be found in Section 5.1.

3.3.2.2 Enhanced Burgers Solver
The Enhanced Burgers solver is integrated into the main program (FOBoom) and is significantly more efficient than the Legacy Burgers solver. Additional physical effects are included and discussed in the next sections. Another benefit of the Enhanced Burgers Solver is that it does not need an anti-Gibbs phenomenon filter, which is sometimes needed in the Legacy Burgers solver to suppress the occurrence of undesirable Gibbs oscillations.

The governing equation for the Enhanced Burgers Solver, which employs the propagation path s to step along the ray paths, is given by [6]

$$\frac{\partial U}{\partial s} = \frac{i\omega\tilde{\beta}}{4\pi} \int U(\omega')U(\omega - \omega')d\omega' - \alpha_t(\omega)U(\omega) \tag{32}$$

where

$\quad U = \quad$ Fourier transform of the scaled acoustic pressure u

$\quad \omega = \quad 2\pi f$, the angular frequency

$\quad \alpha_t = \quad$ Doppler shifted and convected absorption coefficient

$\quad \tilde{\beta} = \quad \frac{\beta p_i^{(p)}B\Omega\chi}{\rho_o c_o^3}$, a scaled coefficient of nonlinearity

$\quad \beta = \quad$ 1.2, the coefficient of nonlinearity of air

$\quad \chi = \quad c_o/v$, the convective index

$\quad c_o = \quad$ quiescent sound speed

$\quad v = \quad \|\overrightarrow{v_r}\|$, the ray velocity magnitude

$\quad \Omega = \quad 1 - \overrightarrow{v_o} \cdot \left(\frac{\hat{n}}{c_o + \overrightarrow{v_o} \cdot \hat{n}}\right)$, the Doppler shift

$\quad \overrightarrow{v_o} = \quad$ wind velocity vector

$\quad \hat{n} = \quad$ wavefront-normal unit vector

$\quad B = \quad \sqrt{\frac{\rho_o c_o \chi \Omega A_r}{\rho_{o,r} c_{o,r} \chi_r \Omega_r A}}$, Blokhintzev scaling factor that includes the wind effects.

The scaled acoustic pressure satisfies $p = p_r^{(p)}Bu$, where $p_r^{(p)}$ is the peak overpressure at a reference point denoted by the subscript r. As with linear geometrical acoustics, the Blokhintzev factor measures the combined effects of spreading and inhomogeneity in the medium. Additional details on $\tilde{\beta}$ as well as B can be found in Lonzaga [6].

In the presence of wind, the Legacy and Enhanced Burgers equations are not equivalent due to different treatments of the full effects of the motion of the medium relative to the ground, which is the reference frame used by PCBoom and many other sonic boom codes. The Legacy Burgers solver incorporates wind effects using an approximation in the calculation of the Blokhintzev amplitude factor and age parameter

by replacing the ray speed with the effective sound speed [6] component in the direction of the ray propagation.

3.3.2.2.1 Moving Air Mass Absorption Effects in the Enhanced Burgers Solver

The Enhanced Burgers Solver takes the motion of the air relative to the ground into consideration by applying a Doppler shift to the frequencies and scaling the absorption coefficients by the convective index of the wind. The convective index quantifies the effect with which the path length traversed by the ray through the medium is different from the path length seen by the observer. The Doppler effect may be observed as a shift in the frequency seen by an observer at rest with the ground reference frame.

The solution to the second term on the right-hand side of Equation (32) is

$$U(s, \omega) = U(s_o, \omega)e^{-a(s,\omega)}$$

(33)

where

$$a(s, \omega) = \int_{s_o}^{s} \alpha_t(s', \omega)ds'$$

is the integrated absorption coefficient and s_o is the initial coordinate along the ray path. The Doppler shifted and convected absorption coefficient derived by Lonzaga [6] is given by

$$\alpha_t(s, \omega) = \chi\alpha_t'(s, \Omega\omega)$$

(34)

where

$$\alpha_t'(\omega) = \omega^2 A_{cr} + \sum_v \frac{A_{vr}^{(v)}(\omega^2\omega_v - i\omega^3)}{\omega^2 + \omega_v^2}, \text{ the quiescent absorption coefficient}$$

$$A_{cr} = A_{cr}(\omega, p_o, T), \text{ classical rotational absorption coefficient}$$

$$A_{vr}^{(v)} = A_{vr}^{(v)}(\omega, p_o, T), \text{ vibrational relaxation absorption coefficient, where } v$$
represents relaxation processes from either diatomic nitrogen or oxygen with their corresponding relaxation frequencies ω_v.

3.3.2.2.2 Moving Air Mass Effects on the Age Parameter

The solution to the first term in the right-hand side of Equation (32) is

$$u(s, \tau) = u_0(\tau + u_0\Lambda(s)).$$

The Enhanced Burgers Solver's age parameter Λ is given by

$$\Lambda(s) = \int_{s_o}^{s} \tilde{\beta}ds'$$

where

$$s = \text{the ray coordinate along the propagation path}$$
$$s_o = \text{the starting coordinate along the path.}$$

3.3.3 Turbulence Effects

Atmospheric turbulence is caused by fine scale variations in the atmosphere due to wind and/or temperature inhomogeneities. The sonic boom is focused and defocused due to these inhomogeneities, resulting in signature distortion. Turbulence has two distinctive effects on sonic booms: distortion in the form of random perturbations appear behind the shock waves, and shock rise times are increased randomly. Atmospheric turbulence affects booms generally within the atmospheric boundary layer (ABL). For overland propagation, the ABL height and strength of the turbulence are usually related to the time of day and the strength of solar radiation. In early morning, there is minimal turbulence, but in the afternoon, the turbulence is at its peak. Principally the major effect of turbulence on N-wave shaped sonic booms is peaking and rounding of the front and tail shocks. This affects both the amplitude and the rise time. Hence it is imperative to include the effects of turbulence on signatures in order to adequately capture the effects realistic atmospheric conditions have on sonic boom waveforms.

Three atmospheric turbulence effect models are incorporated into PCBoom. The first generation model uses a 3D turbulent scattering model developed by S. C. Crow [42] in the late 1960s. Crow's technique quantifies the random distortions, and this theory was shown to qualitatively agree with available flight test data [43, 44]. The second method, more recently developed at The Pennsylvania State University (Penn State), is a 2D turbulence model that uses finite impulse response (FIR) filters. The Penn State scattering model is based on numerical simulations of sonic booms propagating through statistically derived instances of the ABL. PCBoom incorporates both Crow's method (described in Section 3.3.3.1), as well as the Penn State method (described in Section 3.3.3.2). The Penn State method was extended in conjunction with the NASA Sonic Booms in Atmospheric Turbulence (SonicBAT) flight tests [45], and a new set of turbulence filters is provided by the FiltView module (described in Section 3.3.3.2.2). Finally, a "classical" turbulence model utilizing fully 3-D, linear acoustic propagation with turbulent temperature and gust variation was developed at Lockheed Martin Corp., also in conjunction with the SonicBAT flight tests, and is described in Section 3.3.3.3. It was developed as an efficient alternative to the 2D turbulence model by Penn State, which is computationally inefficient.

3.3.3.1 Crow's Method

For conventional supersonic aircraft, sonic booms are N-waves consisting of two nearly equal shocks separated by a linear expansion. Propagation through the real atmosphere causes some distortion of the sonic boom signature. Figure 3-24 shows samples of N-wave sonic booms measured under two different atmospheric conditions [46]. Part (a) shows relatively clean N-waves measured under calm conditions, and part (b) shows distorted N-waves measured under turbulent, gusty conditions. Two types of distortion are seen. The first is that perturbations appear behind the shocks. Crow [42], who called these perturbations a "spiky fine structure," showed that the perturbations were caused by turbulent scattering in the lower atmosphere. The second distortion is that the shock rise times tend to be longer under turbulent conditions.

A detail seen in Figure 3-24 is that the distortion associated with the tail shocks strongly resembles the distortion associated with the bow shocks. This is generally cited as evidence of frozen turbulence, i.e., the time scale of changes in the turbulence is slow compared to the time between the bow and tail shocks, therefore, each shock sees essentially the same turbulence. In the 2004 Shaped Sonic Boom Experiment, it was found that the spiky fine structure for front and rear shocks of N-wave booms was virtually identical [43, 47].
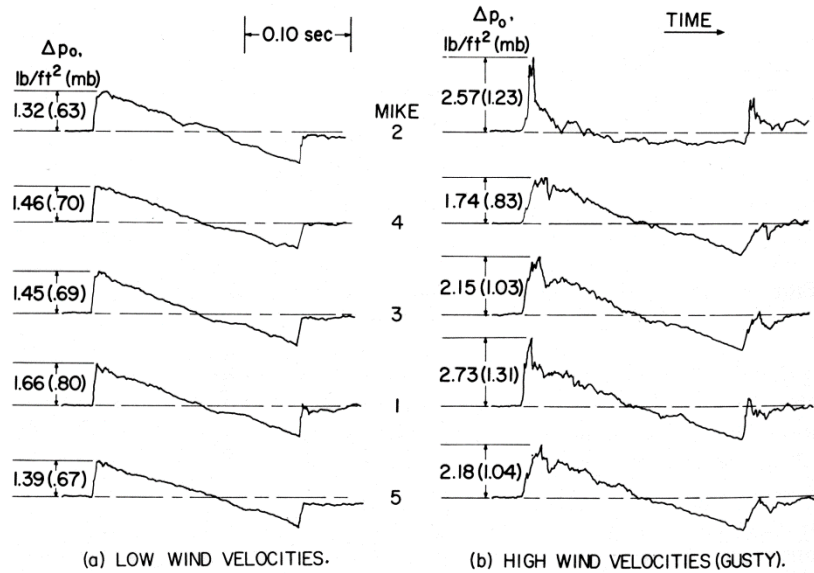
Figure 3-24: Sonic Booms Measured Under Calm and Turbulent Conditions
(*Source: NASA TN D-3655* [46]).

When a wave passes through a region of turbulence, interaction of the wave with eddies causes the generation of secondary scattered waves, as sketched in Figure 3-25 for a wave propagating downward and to the left. The properties of the scattered waves are given by Green's function that accounts for turbulence properties, wavelength, etc. In traditional scattering theory [48], the incident wave is taken as a continuous harmonic wave, and Green's function is integrated over the turbulent volume through which the wave has passed. It is common to present scattering results in the frequency domain in terms of scattering cross-sections and the angles at which sound is scattered.
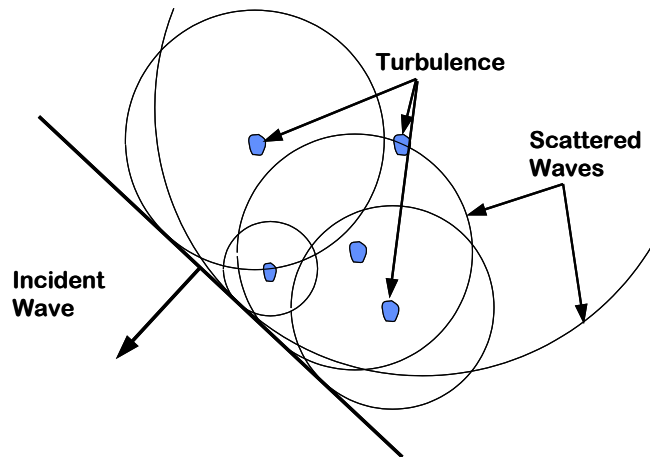
Figure 3-25: Scattering of Sound by Turbulence (*Source: Plotkin, Maglieri, and Sullivan* [43]*, reprinted with permission*).

Attempts to apply traditional frequency domain scattering to sonic booms tended to be unsuccessful, hampered by the thin nature of shock waves. Thus, Crow worked in the time domain and formally represented the shock as a step function. He also took the primary independent variable to be the distance behind the shock rather than the traditional scattering angle. Because the shock is a line, the scattering volume for a receiver point a given distance behind the shock reduces to a paraboloid of revolution, with the point being the focus and the shock being the directrix. This is illustrated in Figure

3-26 for a receiver point a distance h behind the shock. Crow called this the "paraboloid of dependence."
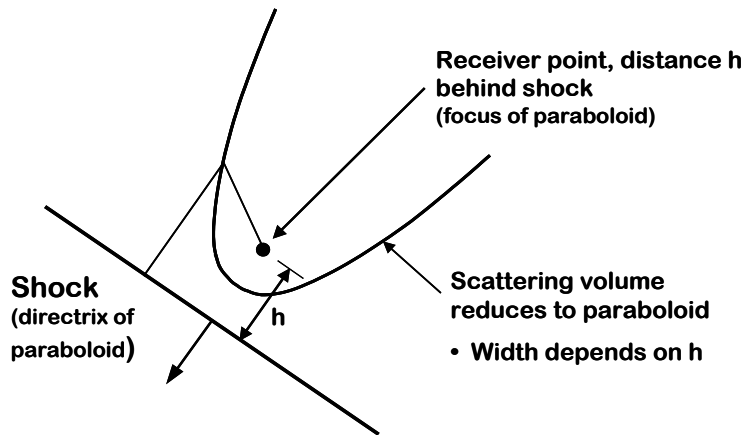
Figure 3-26: Scattering Volume for Thin Shock: Paraboloid of Dependence
(*Source: Plotkin, Maglieri, and Sullivan* [43]*, reprinted with permission*).

The paraboloid perspective is very useful because it illustrates the scales of turbulence that matter the most: those that match the diameter of the paraboloid. All scales matter for a given h, but larger scales dominate at regions further from the front. For receiver points at larger h, the paraboloid is wider, and larger scales are involved. This phenomenon can be used to deduce the frequency content of scattered waves [49], although Crow did not pursue that detail. It does explain why the spiky fine structure becomes smoother further behind the shock.

After developing the scattering relations associated with the paraboloid, Crow prepared a model for the turbulent ABL. He based this on a flat plate boundary layer, scaling the thickness and velocity profile to ABL dimensions and using a reasonable value of turbulent intensity. The turbulent spectrum was taken to correspond to the Kolmogorov inertial subrange. The scale size in the ABL increases with distance from the ground, so that for the dimensions Crow examined, the paraboloid diameter was always within the outer scale of turbulence, i.e., in the 5/3 law Kolmogorov range. Crow's final result was

$$\langle p^2 \rangle^{1/2} = \left(\frac{h_c}{h}\right)^{7/12} \tag{35}$$

where

$h_c =$ the single parameter that resulted from integration over the scattering paraboloid.

If a sonic boom signature is plotted as a function of time rather than distance, there is a corresponding $t_c = h_c/a$, where $a =$ speed of sound. The numerical value of t_c in milliseconds is within about 10 percent of the numerical value of h_c in feet, and, considering the approximate nature of any estimate of turbulence, it is common to use them interchangeably. A demonstration of the application of this methodology to the Shaped Sonic Boom Demonstrator (SSBD) measurements [47] is provided in Figure 3-27 for three different turbulence levels ranging from $t_c = 0.1$ to 0.4 ms.
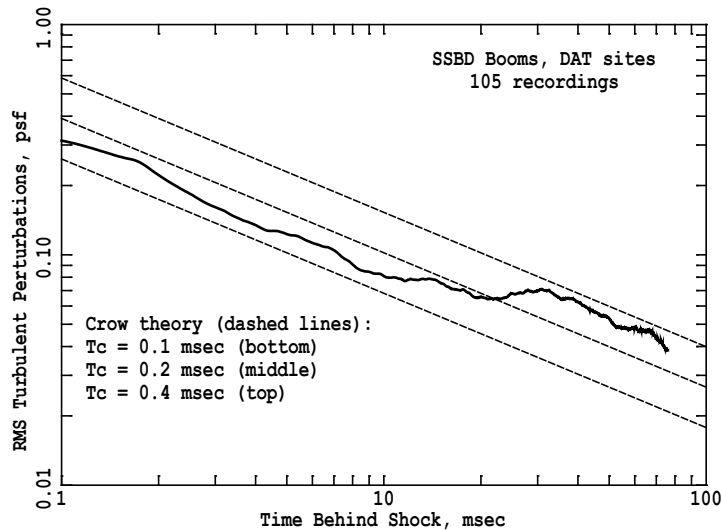
Figure 3-27: RMS Values of SSBD Rear Shock Spiky Fine Structure
(*Source: Plotkin, Maglieri, Sullivan* **[43]***, reprinted with permission*).

3.3.3.2 FIR Filters

3.3.3.2.1 FIR Filters Developed Using Nonlinear Progressive Wave Equation (NPE)

During 2007-2009, a new paradigm for including turbulence in numerical simulations of sonic boom propagation in realistic atmospheres was established. This method involves simulated propagation of sonic boom signatures through an inhomogeneous atmosphere. The simulations include the effects of nonlinear propagation and a realistic turbulence model. The initial attempts at using this method were presented [50] at the AIAA/CEAS Aeroacoustics Meeting in Spring 2008 and in the Ph.D. dissertation [51] of Locey.

The model is incorporated into PCBoom by way of post processing functions. Specifically, all resource-intensive calculations are performed ahead of time and incorporated into the post processing functions. The post processing functions, including turbulence effects, are applied to booms already propagated without turbulence.

Figure 3-28 depicts how a post processing function, or filter, is obtained. Above the ABL, the signature is relatively unaffected by turbulence and therefore called the "clean waveform." However, near the Earth's surface the effects of turbulence are evident. Thus, only the propagation path through the ABL need be modeled for turbulence. The clean signature above the ABL and the ground signature are used to construct the filter. Once built, the filter can be quickly applied by convolving it with any appropriate sonic boom waveform. That sonic boom waveform has then been "turbulized." Also seen in Figure 3-28 is a small sample of the atmosphere extracted to show the domain of the computer code used to establish the post processing/filter functions. This numerical method is called the Nonlinear Progressive wave Equation (NPE) and was first used for sonic booms by Piacsek [52]. However, that initial model did not include a realistic atmospheric turbulence spectrum. In 2008, a realistic von Karman model for turbulence was incorporated into the NPE [50].

40
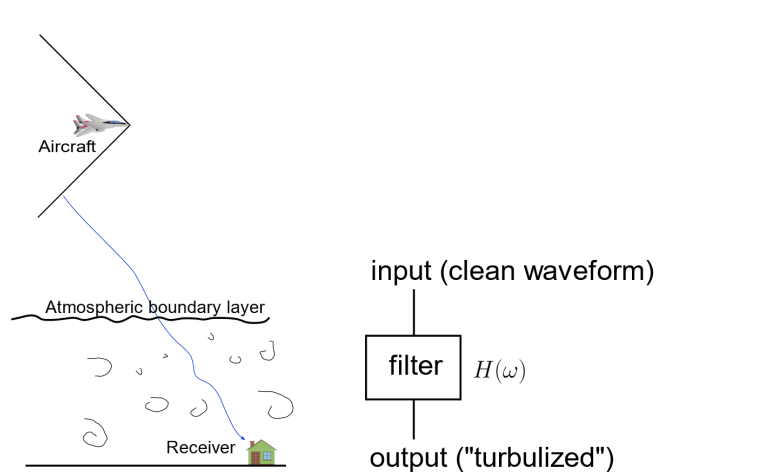
Figure 3-28: Overview of FIR Filter Method for Atmospheric Turbulence
(*Source: Locey and Sparrow* **[53]**, *reprinted with permission*).

User inputs are tied to values of the turbulent energy dissipation rate ε per unit mass of air. Since it is difficult to measure ε, instead one usually measures C_V^2 and C_T^2, the velocity structure function parameter and the temperature structure function parameter, respectively [54]. In the current NPE code, only temperature fluctuations are used, for simplicity, to model the inhomogeneities of the atmosphere.

The user inputs for strength of turbulence in PCBoom are tied to pre-established values of C_T^2 which have been used in the NPE precalculations. Although C_T^2 varies slightly with height, this is ignored in the NPE calculations. Within the NPE, the turbulent atmospheric inhomogeneities are modeled with the Fourier modes approach of Blanc-Benon et al. [55]. See Locey's documentation [50, 51] for additional details.

The FIR filters implemented in WCON were computed based on the output of five different NPE runs, each with different turbulent realizations and an averaged turbulence filter as described in Table 3-1. The PCBoom postprocessor WCON was adapted to apply the resulting average or one of five individual realizations of turbulence based on the FIR filters as shown in Figure 3-29. Two assumptions in the filter methodology of Locey are that energy lost between the clean waveform (at altitude) and the ground waveform is negligible, and that the sonic boom at altitude corresponds to the same sonic boom on the ground [51]. Applying these NPE-based FIR filters to clean ground waveforms requires the same assumptions. Subsequent FIR filter development using KZKFourier simulations [56] followed a similar methodology, though the clean waveforms were created by propagating through the atmospheric boundary layer without turbulence.

The FIR filters are based on analysis of turbulent distortions that occur in the first 50 ms after the passage of a shock. This encompasses the shorter region near the shock where turbulence effects are strongest. The NPE-based FIR filters were not faired smoothly to zero at the end of the 50 ms, so there is a slight irregularity at the end of the 50 ms. It has been found that this irregularity does not have an effect on loudness metrics such as PL. The user is cautioned against application of the FIR filters for signatures that extend less than 50 ms beyond the final shock. Note that the KZKFourier-based FIR filters developed by Stout [56] (see Section 3.3.3.2.2) included short tapers to the front and back of the filter coefficients to ensure that output waveforms taper to zero at temporal edges.

Table 3-1 FIR turbulence filters implemented in WCON.

WCON Option	Description
1	PSU 1 -Average turbulence
2	PSU 2 -Big Spike (PSU Run 5, Filter 30, high turbulence)
3	PSU 3 -Rounded (PSU Run 5, Filter 144, high turbulence)
4	PSU 4 -Small Spike (PSU Run 5, Filter 400, low turbulence)
5	PSU 5 -Tiny Round (PSU Run 5, Filter 200, low turbulence)
6	PSU 6 -Little Spike, Little Round Combo (PSU Run 5, Filter 250, low turbulence)
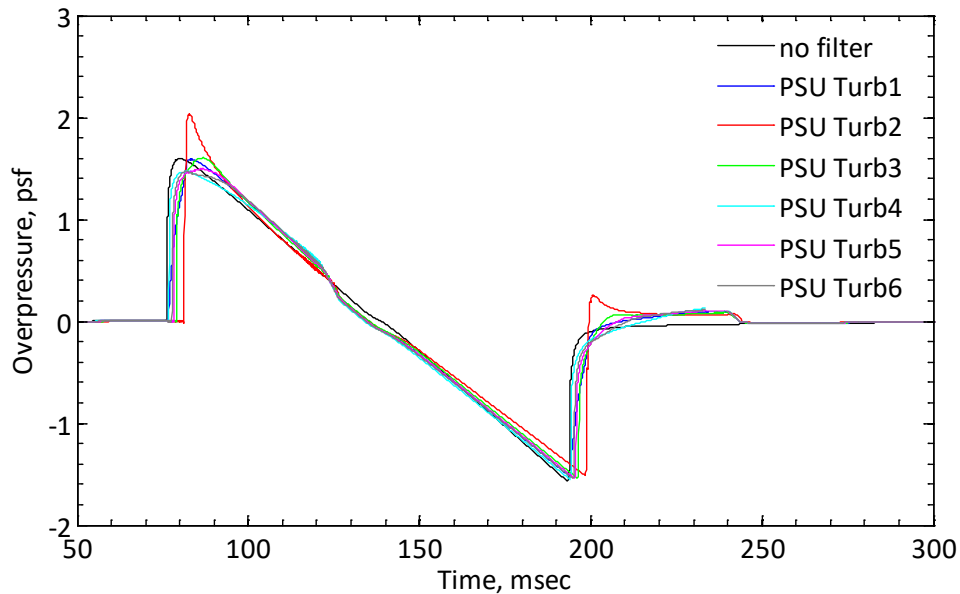
Figure 3-29: Modeled F-18 Ground Signature with Turbulence Filters Applied.

3.3.3.2.2 FIR Filters Developed Using KZK Equation

The Penn State FIR filter approach to modeling turbulence effects [51] was extended in conjunction with the NASA SonicBAT [45] effort. Material in this section of the PCBoom Technical Manual was adapted/reproduced from Chapters 3 and 8 of the SonicBAT report by Bradley et al. [45], and the work by Stout [56].

To model the random focusing and defocusing of acoustic energy in sonic boom waveforms via atmospheric turbulence, the nonlinear Khokhlov-Zabolotskaya-Kuznetsov (KZK) propagation equation was used. This equation is more physically descriptive than the Burgers equation and includes effects of transverse diffraction and propagation in more than one dimension, as well as vector and scalar turbulence. Penn State developed an algorithm referred to as the KZKFourier engine to solve an augmented KZK equation to numerically model propagation of the sonic boom through turbulence. The KZK implementation of Stout [56] is better behaved at domain boundaries than the time-domain solution of the NPE. The augmented KZK equation in two or three dimensions solved by KZKFourier is given by References [45, 56] and can be written as

$$\frac{\partial p}{\partial z} = \frac{c_0}{2} \int_{-\infty}^{\tau} \nabla_{\perp} p \, d\tau' + \frac{\delta}{2c_0^3} \frac{\partial^2 p}{\partial \tau^2} + \frac{\beta p}{\rho_0 c_0^3} \frac{\partial p}{\partial \tau} + \frac{c'}{c_0^2} \frac{\partial p}{\partial \tau} + \frac{\boldsymbol{u_z}}{c_0^2} \frac{\partial p}{\partial \tau}$$
$$- \frac{1}{c_0} (\boldsymbol{u_\perp} \cdot \nabla_{\perp} p) + \sum_v \frac{c'_v}{c_0^2} \int_{-\infty}^{\tau} \frac{\partial^2 p}{\partial \tau'^2} e^{-\frac{(\tau - \tau\prime)}{t_v}} \, d\tau' \qquad (36)$$

where

$p(z, y, \tau) =$ pressure waveform

$z =$ propagation direction coordinate

$y =$ transverse coordinate perpendicular to z

$\tau =$ retarded time, defined as $t - \frac{z}{c_0}$

$t =$ time coordinate

$c_0 =$ ambient sound speed

$\rho_0 =$ ambient density

$c' =$ sound speed change due to temperature fluctuations

$\beta =$ nonlinearity coefficient

$\delta =$ sound diffusivity

$u =$ wind fluctuation, subscript denotes vector component

$c'_v =$ incremental sound speed change due to relaxation process

$t_v =$ relaxation time.

The retarded time coordinate system follows the mean propagation of the wave. For a two-dimensional system, the transverse Laplacian $\nabla_{\perp}^2$ is equivalent to $\partial^2/\partial y^2$, and $\boldsymbol{u_\perp} \cdot \nabla_{\perp} p$ is equivalent to $\boldsymbol{u_y} \, \partial p/\partial y$. From left to right, the terms on the right-hand side of Equation (36) describe the physical processes of diffraction, thermoviscous absorption, nonlinearity, scalar turbulence (temperature fluctuations), vector turbulence (wind fluctuations) in the propagation direction, vector turbulence in the transverse direction, and molecular relaxation absorption [45, 56].

The approach to turbulence modeling in the KZKFourier engine follows the Random Fourier Modes (RFM) method formulated for both temperature fluctuations (Chevret et al. [57]) and wind fluctuations (Blanc-Benon et al. [55]). Implementing this method avoids the cost of full-fidelity CFD simulation over the entire ABL. The RFM method requires several atmospheric turbulence parameters as inputs, which were measured or inferred as part of SonicBAT [45, 56]. FIR filter production runs of the KZKFourier engine used sonic anemometer and balloon launch measurements of four turbulence quantities from SonicBAT, together with the approximate turbulence model of Ostashev and Wilson [58], and Wilson [59], to generate inputs for RFM [45, 56].

The numerical turbulence modeling method described in References [45, 56] creates frozen, i.e., time-invariant, isotropic fields and requires only the turbulent energy spectrum as a function of wavenumber for the calculation. The fields are computed by sampling the energy spectra and adding wavenumber components together. Amplitudes are prescribed by the spectra with components having uniformly randomized direction and phase. For scalar turbulence, 800 modes are summed in this way, and 8,000 modes for vector turbulence. Modes are logarithmically spaced in wavenumber from about 1.0×10^{-5} m^{-1} to 3.1×10^4 m^{-1}, depending on the ABL height.

The 3D von Karman energy spectrum was used to model energy spectra of real atmospheric turbulence. To construct the spectra of both scalar and vector turbulence, the KZKFourier engine uses the methods of Wilson [59] and Ostashev and Wilson [58], which model modified von Karman spectra using the outer length scale of turbulence, and a scalar variance of either a single velocity component or the

temperature. Values of those parameters were derived from measured or assumed atmospheric turbulence quantities [45, 56].

The theory underpinning the KZKFourier turbulence model is given by References [45, 56]. For scalar turbulence, the variance $\sigma_T{}^2$ and outer length scale L_T suitable for the von Karman model are given by Equations (6.50) and (6.51) in Ostashev and Wilson [58] and, for convenience, are presented here as

$$\frac{\sigma_T^2}{T_*^2} = \frac{4.0}{\left[1 + 10\left(-\frac{h}{L_{mo}}\right)\right]^{2/3}} \tag{37}$$

and

$$\frac{L_T(h)}{h} = 2.0 \frac{1 + 7.0\left(-\frac{h}{L_{mo}}\right)}{1 + 10\left(-\frac{h}{L_{mo}}\right)} \tag{38}$$

where

$$
\begin{array}{rl}
h = & \text{height above ground} \\
T_* = & \text{the surface-layer temperature scale} \\
L_{mo} = & \text{the Monin-Obukhov length scale defined in Equation (39).}
\end{array}
$$

The Monin-Obukhov length scale is define as

$$L_{mo} = -\frac{z_i u_*^3}{\kappa w_*^3} \tag{39}$$

where

$$
\begin{array}{rl}
u_* = & \text{the friction velocity} \\
w_* = & \text{the mixed-layer velocity scale} \\
z_i = & \text{the atmospheric boundary-layer height} \\
\kappa = & 0.4, \text{ von Karman's constant.}
\end{array}
$$

Turbulence production due to wind shear and buoyancy due to solar heating of the ground are important mechanisms for atmospheric vector turbulence. Buoyancy mechanisms can dominate in much of the ABL when conditions are favorable for convection such as a clear, sunny day [45]. The KZKFourier engine models vector turbulence produced by both shear and buoyancy mechanisms. Two sets of quantities characterizing shear and buoyancy turbulence production are, respectively, given in Wilson [59] and in Ostashev and Wilson [58] as

$$
\begin{aligned}
\sigma_s^2 &= 3.0u_*^2 \\
L_s &= 1.8h
\end{aligned} \tag{40}
$$

and

$$
\begin{aligned}
\sigma_b^2 &= 0.35w_*^2 \\
L_b &= 0.23z_i
\end{aligned} \tag{41}
$$

where

$$
\begin{array}{rl}
\text{the subscript } s = & \text{denotes shear-driven turbulence production} \\
\text{the subscript } b = & \text{denotes buoyancy-driven turbulence production}
\end{array}
$$

z_i = the capping inversion height
u_* = the friction velocity
w_* = the mixed-layer velocity scale.

The theories for buoyancy and shear production can be combined by adding the corresponding von Karman energy spectra. The combined variance of vector turbulence is given by the relation [45, 56]

$$\sigma^2_{\text{vector}} = 3.0 u_*^2 + 0.35 w_*^2. \tag{42}$$

To develop turbulence FIR filters using the method developed by Locey and Sparrow [50, 53], KZKFourier simulations were completed at 96 different conditions [45, 56]. Filter production simulations were run using the NASA Pleiades supercomputing cluster [60, 61]. Simulated conditions for the filter production runs comprised a four-dimensional matrix of the following parameters: turbulence strength, ABL height, propagation angle, and relative humidity. The four parameters were varied using the discrete values in Table 3-2. All parameter combination values were tested, for a total of 96 sets of production simulation conditions.

Table 3-2 Parameter values used in KZKFourier production simulations
(*Source: NASA/CR–2020–220509. Bradley et al.* [45]).

Parameter	Values
Turbulence Strength	Low, medium, and high as defined in Table 3-3
Atmospheric boundary-layer height	400 m, 1000 m, 1600 m, 2200 m
Ray elevation angle	21°, 26°, 33°, 42°
Relative humidity	15%, 60%

Table 3-3 Atmospheric turbulence quantities used for each turbulence condition considered
(*Source: NASA/CR–2020–220509. Bradley et al.* [45]).

	u_* (m/s)	w_* (m/s)	T_* (K)	σ_{vector} (m/s)
Low turbulence	0.1	0.44	-0.5	0.31
Medium turbulence	0.35	1.55	-0.5	1.10
High turbulence	0.6	2.66	-0.5	1.89

The atmospheric values in Table 3-3 correspond to conditions during the SonicBAT flight tests [45, 56]. The RMS vector turbulence for the "medium" condition is near the mean of the RMS vector turbulence strengths measured at NASA Kennedy Space Flight Center (KSC) and below the mean measured at NASA Armstrong Flight Research Center (AFRC). The "low" condition magnitude is below all the recorded values but is close to the minimum of 0.46 m/s recorded at AFRC in the early morning. The "high" condition magnitude approaches the maximum value recorded at AFRC of 2.31 m/s. These conditions do not span all possible turbulence states of the atmosphere, and only one value of $-z_i/L_{mo}$ was used to characterize convection strength. Additional guidance for applicability of these conditions is given in References [45, 56].

The initial condition at the top of the ABL used in the KZKFourier production simulations [45, 56] was a motor glider measurement of an F-18 boom waveform at AFRC, when the relative humidity measured by the weather tower was about 15%. The KZKFourier engine "clones" the starting waveform into a planar soundwave, which is distorted randomly by turbulence as it propagates. To quantify variability in

simulated ground waveforms due to turbulence, many realizations of KZKFourier simulations were performed at each of the 96 sets of conditions from Table 3-2. Approximately 18,000 waveforms were produced at each condition.

For each of the 96 conditions, three FIR filters were developed [45, 56] to represent the "mean", "mean minus standard deviation" (-std), and "mean plus standard deviation" (+std) effects of turbulence with respect to particular metrics. Specifically, FIR filters were developed for the following metrics[3]: PL, P_{max}, ASEL, BSEL, CSEL, ESEL, and ISBAP. Here, the mean and standard deviation refer to the metric's statistical values for the total population of turbulized waveforms (around 18,000) at a particular set of atmospheric conditions. The utility of the filters is that they can be convolved with a boom signature modeled without turbulence (i.e., a clean waveform) to approximate metric levels under the corresponding turbulent conditions. This approach of using pre-computed filters does not require additional computationally expensive sets of numerical simulations to estimate metrics from different clean waveforms. Note that these filters were developed using an N-wave starting signature. Stout [56] conducted a modeling study in which shaped boom signatures were numerically propagated through a turbulent ABL using the KZKFourier code, and compared metrics from those simulations with metrics determined using FIR filters developed from N-wave signatures. Errors in mean and standard deviation turbulence FIR filters were generally within 1 dB, though applicability of these FIR filters to shaped boom signatures has yet to be demonstrated beyond this model-to-model comparison for one atmospheric turbulence condition.

3.3.3.3 Classical Turbulence Model: Turbo

In addition to FIR filters in the FiltView module, the classical turbulence module Turbo was developed in conjunction with the SonicBAT effort. Turbo was developed by John Morgenstern of Lockheed Martin Corp., and material in this section of the PCBoom Technical Reference was adapted/reproduced from chapters 4 and 7 of the SonicBAT report by Bradley et al. [45].

In classical sonic boom modeling, predictions of ground sonic boom can be accurately made using a combination of linear acoustic ray tracing and nonlinear aging. To predict loudness levels, modeling shock rounding is also important. Shock rounding is a function of thermoviscous dissipation (classical absorption) and molecular relaxation as well as a function of shock scattering in atmospheric turbulence. Burgers methods are named for the equation they solve, which adds the effects of classical absorption and molecular relaxation to classical sonic boom modeling. The classical formulation of linear geometrical acoustics is well established, and acoustic rays can be traced through each turbulent field desired. The Turbo methodology [45] builds upon the modified linear theory used for calm atmosphere sonic boom prediction by repropagating sonic booms through the ABL with random variations in wind and temperature due to turbulence superimposed on the mean winds and temperature. The approach is intended to capture variations in amplitude and focusing from refraction plus rounding and distortions from scattered multipath arrival times. Propagation path and propagation time both affect the turbulent

---

[3] PL is Stevens Mark VII perceived level [80], Pmax is the maximum overpressure, ASEL is the A-weighted sound exposure level [78], BSEL is the B-weighted sound exposure level [78], CSEL is the C-weighted sound exposure level [78], ESEL is the E-weighted sound exposure level [78], and ISBAP is the Indoor Sonic Boom Annoyance Predictor [86]; see also Section 3.5 on Metrics.

signature results. For each PCBoom ray selected (described using coordinates in terms of time at which the ray originated from the aircraft, T_{ac}, and ϕ), Turbo will repropagate a grid of rays starting from the top of the ABL with grid times and azimuth angles before/after and on either side of the PCBoom ray.

Input to Turbo includes ground results from either the modified linear theory (FOBoom) or Burgers predictions of ground sonic boom carpet signatures propagated through mean atmospheric conditions. A second PCBoom run, orchestrated by SonicBAT-Run-Fort, is executed to output ray starting location, vector direction, and time at the top of the ABL. Turbo reads ray starting conditions (x, y, z, ray unit vectors, and time) from PCBoom output at the top of the ABL, but does not read signatures from this height. The signatures are read from ground solutions (FOBoom, HeadlessBurgers, or sBOOM), along with their ground intercept location and time. Mean atmospheric temperatures and winds are read from the PCBoom atmospheric input or from the .age file along with ABL/ground amplitude and aging ratios. For the milliseconds of propagation difference between zero turbulence and turbulent multipath arrival times, nonlinear aging differences can be neglected, so Turbo saves computational time by not repeating nonlinear signature aging calculations [45].

Ray propagation in Turbo continues until the ground elevation is crossed or a time step limit is reached. The exact ray-ground intersection is then calculated between the last two steps. After all ray propagations are complete, each ray tube is checked for intersections with all the signature times requested at that azimuth. At the ground, quadrilateral ray tubes (regular parallelograms without turbulence) are cut into triangular tubes which are referred to as tritubes. This approach to resolving ray tube intersections increases area resolution and accounts for the scenario in which quadrilaterals get folded over upon themselves. Triangle cuts are done between the closer corners of the ray tubes [45].

Individual turbulence-distorted signatures are calculated in Turbo via superposition in space and time of repropagated rays. Following propagation, a signature summation step is invoked to sum one or more ground signatures, one for each tritube intersection found. First, each tritube's area ratio relative to PCBoom's nonturbulent calculation is used to adjust the signature amplitude contributed by each tri-tube. Second, scaled signatures are shifted in time to match the tritube intersection time. Then, all intersections are interpolated to a common time scale and summed into a single signature. Turbo includes the capability to specify multiple output time instances (similar to a ground microphone array) to calculate statistical variations from a large number of signatures in a more efficient manner [45].

The atmospheric modeling implemented in Turbo has typical features found in other codes, with some differences [45]. The Turbo methodology is formulated in Cartesian coordinates. It uses invariant (frozen) Fourier modes of different lengths and random directions to model the turbulence with a von Karman distribution (which follows a Kolmogorov distribution up to the maximum L_0 scale size) and logarithmic spacing of mode lengths. In the RFM method of Blanc-Benon et al. [55, 57], the vector modes have a random velocity direction and a random direction of variation that is normal to the direction of the velocity. Rays are most bent by a mode when traveling in the direction of the variation, which is normal to the mode velocity. If traveling normal to the velocity but not in the direction of the variation, the mode becomes effectively longer (and thereby weaker in gradient) until its effective mode length becomes infinite (zero gradient strength) when the ray is traveling normal to the variation and velocity [45]. In Turbo, every velocity mode has variations in both directions normal to the velocity. Mode velocities vary in a 2-D square lattice in an alternating checker pattern. Gradients are created in two dimensions by each mode, each with their own random phase and normal to the random direction of the mode velocity.

47

Since Turbo models two directions of variation, the model includes the possibility to vary the turbulence strength evenly in each direction (50%/50%) or to use a random strength in each direction from 50%/50% to 25%/75%. Documentation by Bradley et al. [45] and the PCBoom User's Guide [8] provide additional detail. Turbo also includes the possibility to require the phase origin of all z-direction modes to be located at the top of the ABL, making the onset of turbulence continuous with the atmosphere above the ABL, or to generate a random z phase for each mode like the random x and y phases.

Turbo uses integrated forms of the RFMs to get average turning gradients and displacements over discrete time steps [45]. The propagation scheme implemented in Turbo will accurately calculate non-turbulent propagation to the ground with step sizes as large as 10 seconds. Accurate propagation through turbulent conditions requires step sizes of 0.1 to 0.01 seconds. Each propagation step is calculated as circular arcs from integration of the RFM gradients, incorporating both ray turning and displacement. While more computationally expensive than ray turning alone, it also allows mean wind displacements and turning to be assessed at the same time. Turbo also checks the convergence on every step and iterates based on an error estimate dependent on height, since ray turning accuracy at larger distances has a much larger effect on final position.

The turbulence propagation implemented in Turbo handles temperature, x, y, z winds and all their cross-derivatives. Therefore, many more evaluations are performed in Turbo than are required by legacy propagation modeling in FOBoom (temperature, along with x and y derivatives only in the z direction). Mean variations are calculated and simply added with turbulent variations in summation, making their accommodation inexpensive [45].

The Turbo turbulence model is based on several input parameters: minimum length scale of turbulence, maximum length scale of turbulence, and the number of Fourier modes, which are spaced in constant logarithmic steps. Turbo also includes an optional parameter to allow the representation of larger scales than the mode of maximum strength by definition. Modes greater than this can be significant but have diminishing strength according to the von Karman spectrum equations of Cotte and Blanc-Benon [62]. The strength of the turbulence is quantified using temperature and velocity structure parameters C_T^2 and C_v^2. User inputs for structure parameters are required at two altitudes to produce a logarithmic variation in turbulence strength versus altitude. As described in References [45, 56], turbulence structure constants can be derived from atmospheric measurements or from other turbulence parameters using

$$C_T^2 = \frac{3\Gamma\left(\frac{5}{6}\right)\sigma_T^2}{\pi^{1/2}L_0^{2/3}}$$

(43)

and

$$C_v^2 = \frac{3\Gamma\left(\frac{5}{6}\right)\sigma_{\text{vector}}^2}{\pi^{1/2}L_0^{2/3}}$$

(44)

where

$\Gamma =$ the Gamma function
$L_0 =$ the outer length scale associated with velocity or temperature fluctuations.

3.3.4 Focusing

Focal zones occur when a combination of aircraft maneuvers and atmospheric gradients give rise to concave wavefronts and the subsequent propagation distance is sufficiently long for the focal point to be reached. A wavefront with constant curvature will behave similarly to optical focusing through a lens. An aircraft in motion, however, will not always generate such perfectly focusing waves. In such cases, curvature varies such that different portions of the wavefront (generated at different times at the aircraft) have different curvature. The focus point thus moves in space and traces out a line in 2D. This line (or surface in 3D) represents a focal zone. Mathematically, this zone represents an envelope of rays called a caustic.

The location and shape of a caustic is defined by geometrical rays. A determination of signatures at a focus requires more geometrical information than just the location. The concept that a real focus is usually distributed over a caustic surface, rather than a point, suggests that focus amplification would depend on the extent of the caustic surface as compared to the original wavefront. A more quantitative viewpoint is that the caustic forms a boundary to the wave system within which diffraction effects play a key role in the focused signature. The discussion of caustic location and geometry in Section 3.3.4.1 is therefore directed toward defining location and curvature, considering both atmospheric conditions as well as vehicle motion. The curvature of the caustic surface, an important parameter in the signature itself, is described in the treatment of focus behavior found in Section 3.3.4.2.

3.3.4.1 Focus Geometry

There are a number of conceivable ray configurations that can lead to focus. Within FOBoom, however, only those that can be produced by a maneuvering vehicle in a smoothly varying atmosphere are of interest. This section reviews the ray configurations associated with vehicle maneuvers and normal atmospheric gradients, and identifies the kind of foci, which can occur as well as discusses which ones are implemented in FOBoom.

3.3.4.1.1   Steady Maneuvers in a Uniform Atmosphere

Consider a supersonic aircraft in a uniform atmosphere. In steady flight, wavefronts form Mach cones with straight rays emanating normal to the cone surface. Figure 3-30 illustrates a Mach cone and the four-ray construction of a ray tube employed by FOBoom.

The ray tube area is the product of a lateral and a longitudinal component. The longitudinal component is constant, while width increases proportionally to r. Laterally-separated rays diverge linearly, while longitudinally-separated rays are equidistant at all radii.
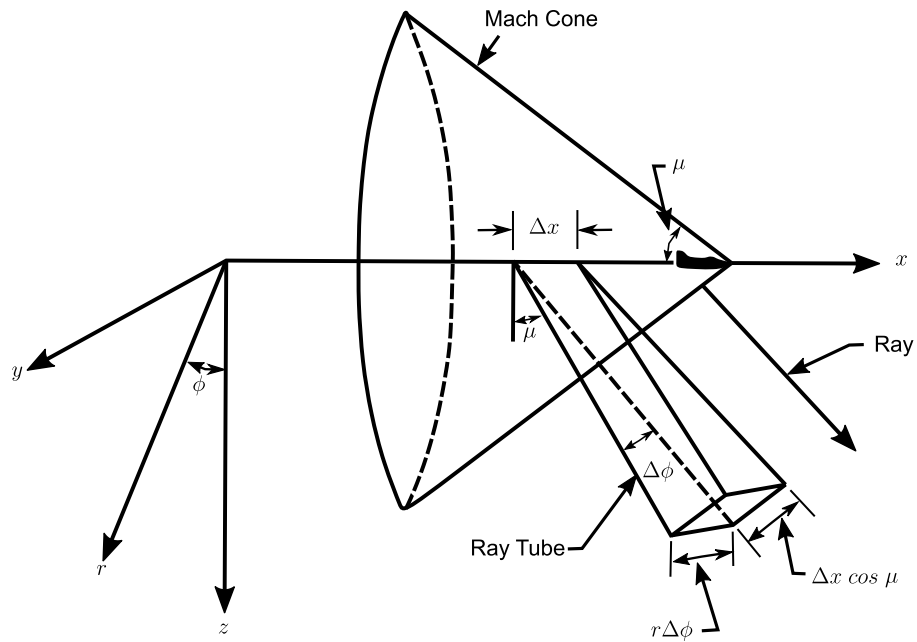
**Figure 3-30: Mach Cone, Ray, and Ray Tube for Steady Level
Supersonic Flight in a Uniform Atmosphere.**

Now consider a supersonic aircraft executing a turn. Laterally-adjacent rays near this plane still diverge at a constant rate, just as they would during steady flight. However, longitudinally-separated rays are no longer parallel. Figure 3-31 illustrates the convergent rays inside the turn, while omitting the divergent waves outside the turn. Note that differentially adjacent rays intersect to form an envelope of rays, known as a caustic. The caustic represents a boundary beyond which rays do not pass. For a steady turn at finite Mach numbers, the radius of the caustic is nonzero. A perfectly spherical focus (analogous to focusing by a lens) can only be achieved by an infinite Mach number.
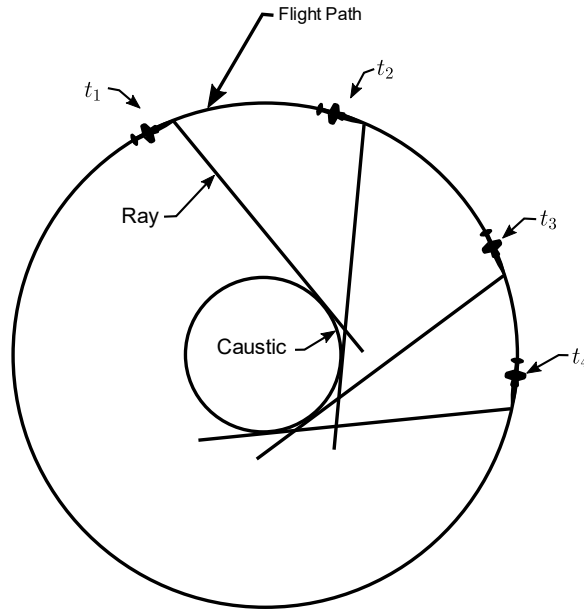
Figure 3-31: Caustic Geometry for a Steady Turn in a Uniform Atmosphere (*Source: Plotkin and Cantril, "Prediction of Sonic Boom at a Focus,"* **[2]***; reprinted by permission of the American Institute of Aeronautics and Astronautics, Inc.***).**

Figure 3-32 illustrates the behavior of rays and wavefronts at a caustic. The rays that graze the caustic are reflected back, causing the wavefront to cusp. Because the caustic geometry represents an edge of the wave field, diffraction plays an important role in the determination of the focused signature. Different portions of the curved wavefront focus at different locations on the caustic, so that the focus at a given point represents a differential segment of wavefront converging. Focus of a finite segment of wavefront to a single point (such as for a cylindrical lens) is a higher order case, occurring only under certain unsteady flight conditions. See Section 3.3.4.1.4 for more details.
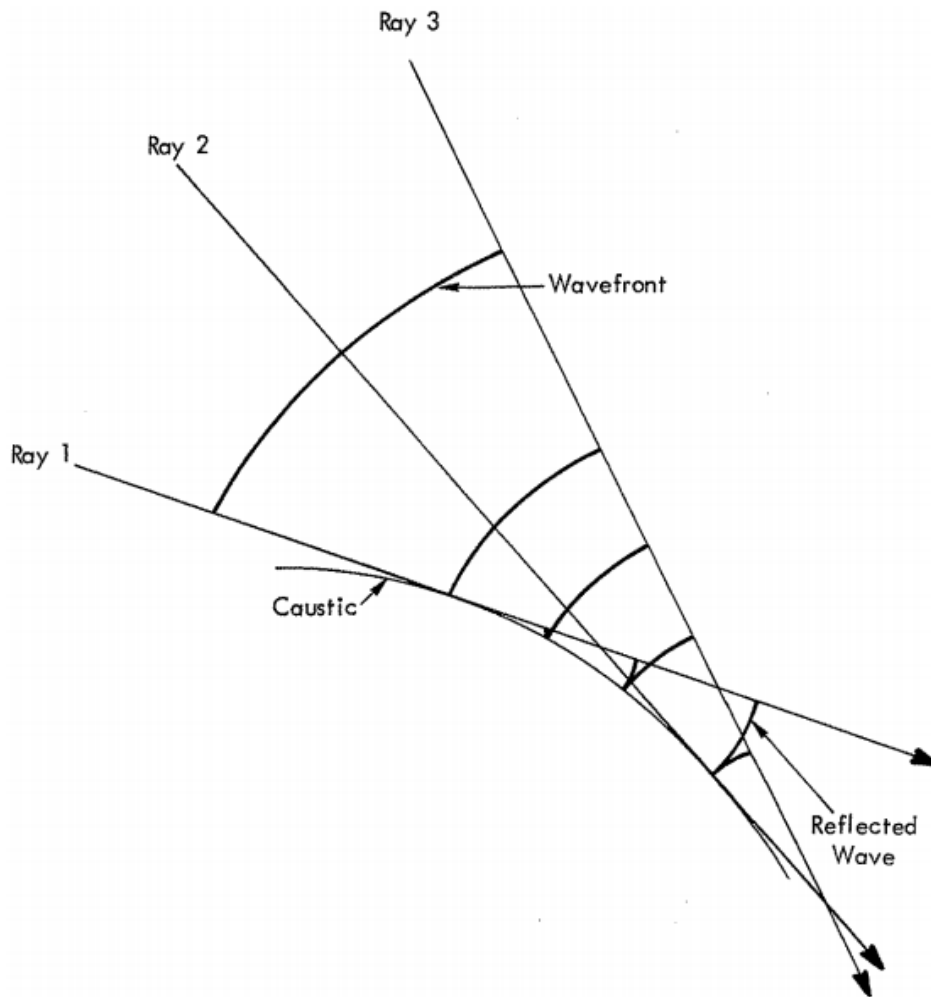
Figure 3-32: Ray Geometry and Wavefront Folding (Reflection) at a Caustic (*Source: Plotkin and Cantril, "Prediction of Sonic Boom at a Focus,"* [2]; *reprinted by permission of the American Institute of Aeronautics and Astronautics, Inc.*).

Near the caustic, differentially adjacent rays converge linearly to zero at a focal point. This convergence will be much stronger than the radial lateral spreading at points closer to the caustic than the flight track. The ray tube area will thus be proportional to S, the ray distance to the focus. If the caustic has radius R_c, a coordinate Y normal to the caustic will be given by $Y = \frac{S^2}{2R_c}$ for $Y \ll R_c$. For a general curved caustic, Y and S are related in terms of the local radius of curvature. Close to the caustic, where the ray tube area $A \propto S \propto \sqrt{Y}$, the only fixed physical dimension available for scaling is R_c. This identifies the lateral scale implicit to diffraction, which is defined by the geometrical ray boundary.

3.3.4.1.2   Refraction by Simple Atmospheric Gradients

Consider steady level supersonic flight in a horizontally stratified atmosphere, with temperature increasing with decreasing altitude. When the rays reach an altitude where the speed of sound is equivalent to the aircraft speed, they begin to curve upward. The lateral and/or longitudinal location at which this occurs is referred to as cutoff. Figure 3-33 illustrates the general curvature of rays, including upward curvature beginning at cutoff. Since successive rays exhibit shape similarity and are displaced in the direction of flight, a caustic is created at cutoff altitude, as shown in Figure 3-34.

a. Under Flight Track

b. Lateral Plane
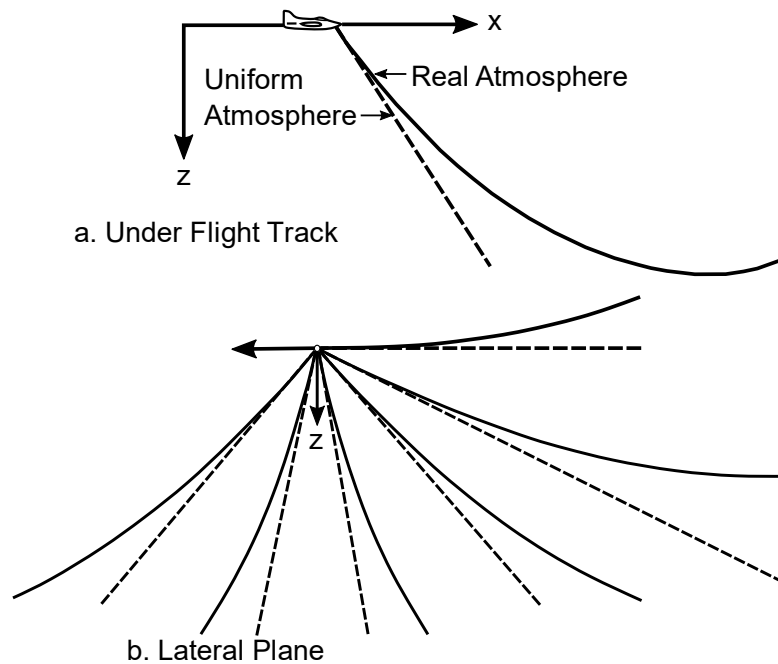
Figure 3-33 Ray Curvature in a Real Atmosphere. (*Source: Plotkin* [12]*, reprinted with permission*).
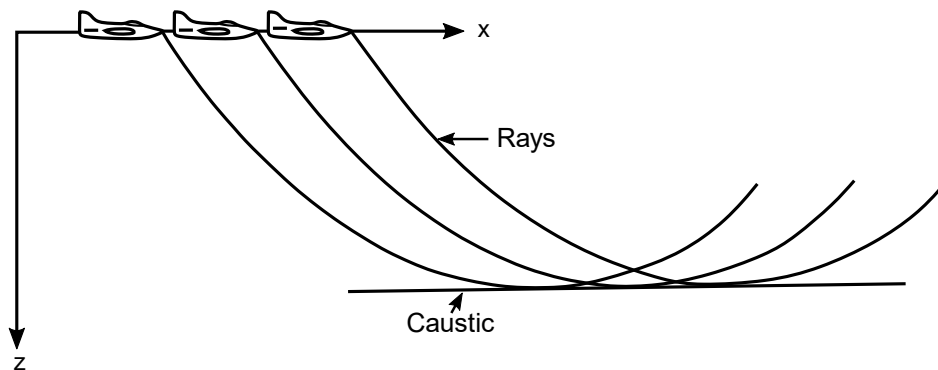
Figure 3-34: Caustic Due to Atmospheric Gradient (*Source: Plotkin and Cantril, "Prediction of Sonic Boom at a Focus,"* [2]*; reprinted by permission of the American Institute of Aeronautics and Astronautics, Inc.*).

For a small altitude range near the caustic, the speed of sound gradient can be assumed constant. The rays are then arcs with radius R_r, tangent to the horizontal caustic. The ray height Y above the caustic is then $Y = \dfrac{S^2}{2R_r}$ for $Y \ll R_r$. For locations much closer to the caustic than to the aircraft, the ray tube area behaves as $A \propto S \propto \sqrt{Y}$. The geometrical acoustics solution near this caustic in ray coordinates is, thus, the same as for the maneuver caustic. The two cases are topologically similar, and a simple

distortion of coordinates can convert one to the other. Diffraction solutions for both geometries are identical for $Y \ll R_c$. They are interchangeable and can be applied to a general case of curved rays and caustics by using the relative radius of curvature between ray and caustic, given by $\frac{1}{R_r} - \frac{1}{R_c}$.

3.3.4.1.3   Focal Conditions from Unsteady Maneuvers

The focal conditions discussed above are created via steady maneuvers and simple atmospheric gradients. Other cases of interest are induced by various unsteady maneuvers.

Acceleration — The turn focus shown in Figure 3-31 arises because the ray angle changes with time while the flight path remains steady. Consider the acceleration of an aircraft in a level flight, as shown in Figure 3-35. As the Mach number increases with position, the Mach angle decreases, and the angle between ray and flight path increases. A caustic surface forms at some distance from the vehicle. The acceleration and rate of change of acceleration must fit certain conditions in order to produce a circular caustic as in Figure 3-31. If these are not matched exactly, the acceleration maneuver produces a noncircular caustic which must be described in terms of its local radius of curvature.
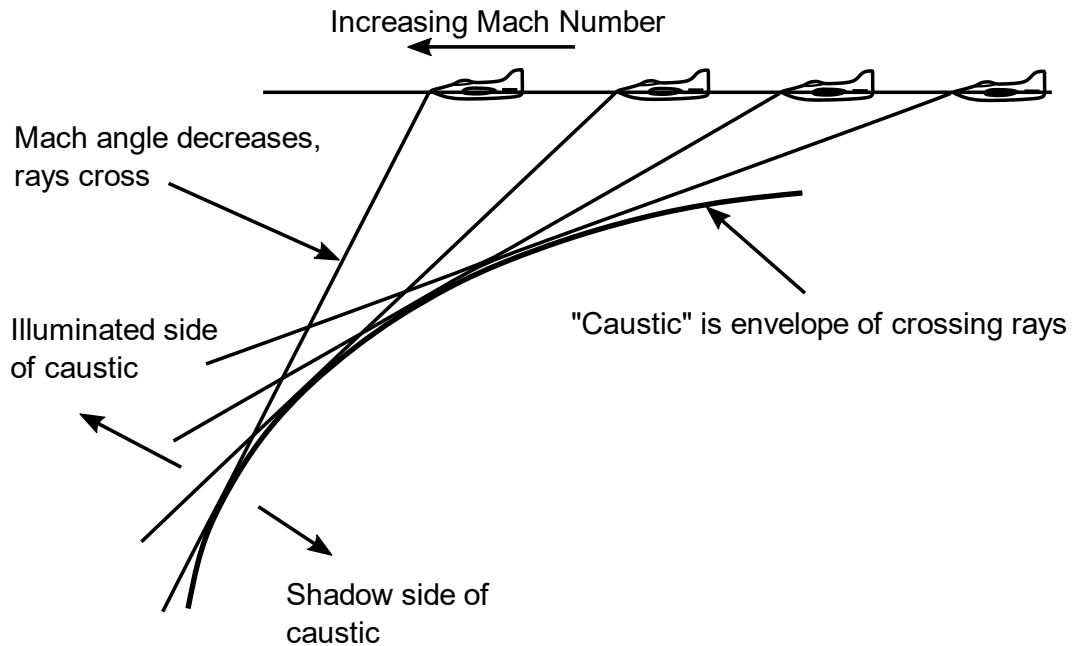
Figure 3-35: Caustic Due to Acceleration (*Source: NASA-CR-218871* [63]).

Pushover/pullup — A pushover or pullup maneuver describes an aircraft changing its pitch angle exactly equivalent to a turn, except that motion is in the vertical plane, such as a dive maneuver. Figure 3-36 demonstrates a pushover maneuver that causes a focus below the vehicle. Conversely, a pullup maneuver creates a focus above the vehicle.

Figure 3-36: Caustic Due to Pushover (*Source: NASA-CR-218871* [63]).

Combined Maneuvers — Combinations of turns and acceleration give rise to a caustic of varying curvature. Combinations of turns and pushovers/pullups will produce nonsymmetrical caustics.

3.3.4.1.4   Complex Maneuvers: Cusped Caustics

Steady turn and cutoff maneuvers lead to caustics of constant finite curvature. Unsteady maneuvers, such as acceleration, lead to caustics of varying curvature. A carefully coordinated acceleration and/or unsteady turn can, in principle, produce a cylindrical wavefront that focuses at a single point. In reality, such point focus conditions last for a short time and are usually preceded and followed by imperfect focus conditions. A point focus forms an end point of a caustic with a simple, finite curvature. Since a caustic is the boundary of a group of rays, it cannot end in midfield. Just as a simple caustic would follow a perfect focus, one must precede it. Thus, a perfect focus is located between two simple caustics.

Figure 3-37 illustrates a hypothetical perfect point focus. The aircraft has maneuvered in an appropriate way such that the wavefront consists of two plane sections (A—B and C—D) with a portion of cylinder between (B—C). The cylindrical segment focuses perfectly at 0 where the two sheets of the focus are joined. Construction of rays beyond the focus reveals two caustics, 0E and 0F. The wavefront is shown at two later times. Rays through 0 from the segment B—C are straight lines through 0, other rays are parallel to either BF or CE.
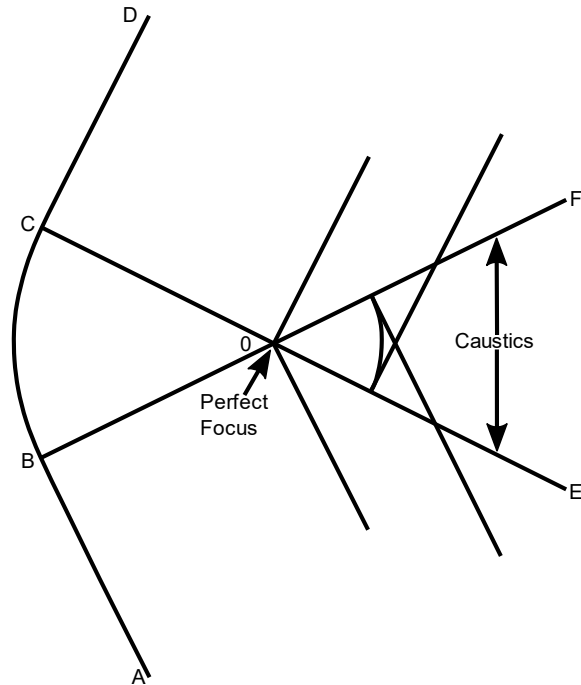
Figure 3-37: Perfect Focus of a Hypothetical Wavefront (*Source: Plotkin and Cantril, "Prediction of Sonic Boom at a Focus,"* [2]*; reprinted by permission of the American Institute of Aeronautics and Astronautics, Inc.*).

Most often, conditions for a perfect focus are satisfied only for an infinitesimal period of time during maneuver transitions, such as turn entry. Figure 3-38 shows the ray and wave pattern for a turn entry. The point focus has an infinitesimally small angle and the two sheets of the caustic (shown as the focus line and the pseudofocus line) meet at a cusp, referred to as a super focus. A super focus represents a perfect focus (infinite curvature) of an infinitesimal wave element. The finite wave element focusing at a point in Figure 3-37 is a higher order case than a cusped caustic in Figure 3-38. The diffractive limit for this case is analogous to diffraction by a slit. The larger the angle, the greater the wave element relative to diffraction edge effects, and the greater the amplification. For a 360° wave element, there would be a perfect convergence with no diffraction, with amplitude limited only by molecular relaxation absorption processes.
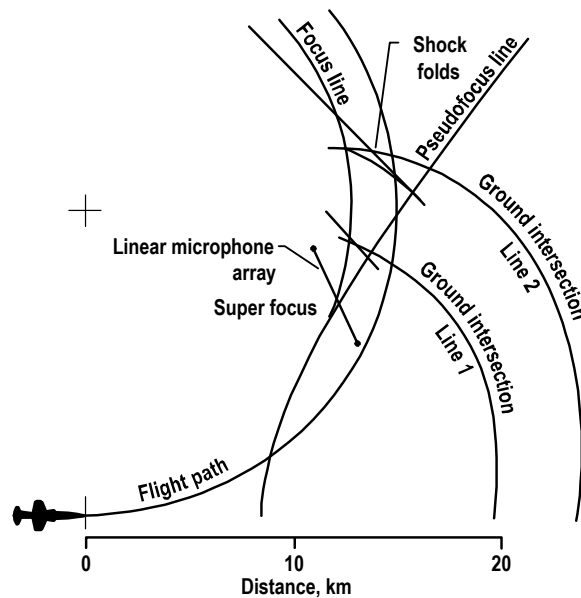
Figure 3-38: Shock Wave Footprint on the Ground in a Turn Entry Maneuver (*Source: NASA-SP-622* [7]).

The distinction between these three cases is important. The smooth caustics pictured in the previous sections represent a continuous focusing distributed over a line; the cusped caustic (Figure 3-38) has an infinitesimal, perfect focus at a point; and the perfect focus (Figure 3-37) has a finite wave element focusing to a point. The character of diffraction at the focus depends on the precise geometry. A wavefront leading to a smooth caustic, for example, cannot be approximated by a series of elements with piecewise constant curvature. Reference [64] includes additional illustrations of focusing ray geometries.

3.3.4.1.5   General Nonsymmetrical Focus

The cases in the previous sections are two-dimensional, in which the rays can be identified by a single parameter: time on the trajectory. In three dimensions, rays are identified by both trajectory time and azimuthal angle. Thus, differentially adjacent rays that cross may be separated by azimuthal differences as well as a time difference.

Figure 3-39 illustrates an aircraft making a steady turn in a uniform atmosphere. Note that this maneuver forms a caustic surface inside the turn. At an altitude below the flight altitude, the caustic has a circular horizontal cross section. Two rays are shown separated by a small distance $R_T \Delta \Psi$ along the trajectory. They intercept the caustic at points separated by $\tilde{R}_c \Delta \Psi$, where $\tilde{R}_c$ is the radius of the horizontal cross section of the caustic surface. From the construction, it is clear that the two rays are skew and will never cross. A later ray which would cross one of these would have to be at a slightly smaller azimuth angle ϕ (measured from the vertical) on the ray cone. The intersection of these two rays would be below the caustic circle shown.

57
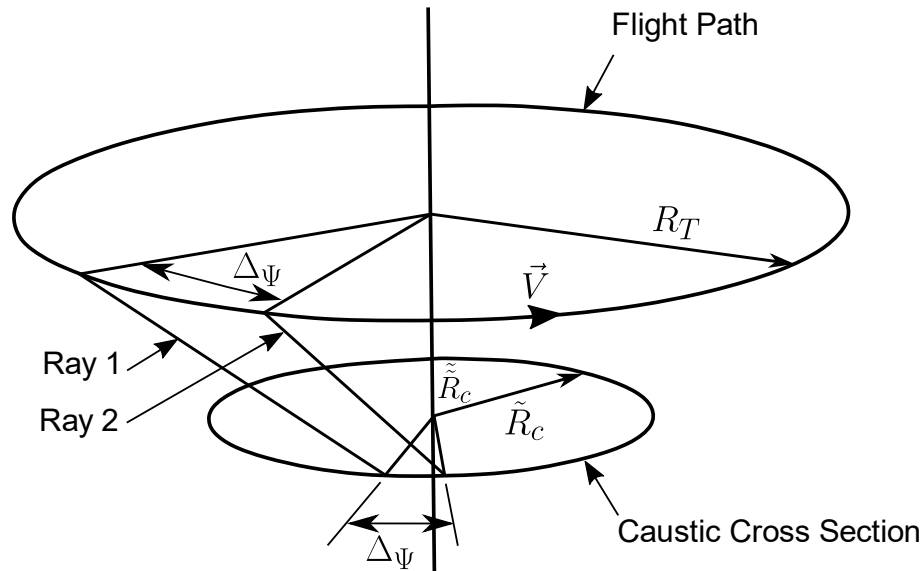
Figure 3-39: Two Successive Rays at Fixed Azimuth in a Steady Turn.

In order to trace out a path on the caustic, the relation between ϕ and t for crossing rays must be known. A focus is located by looking for ray tube area going to zero. However, practical ray tube area calculations do not trace every ray in a ray tube. Presuming the ray function to be sufficiently well behaved, a simple cross-sectional shape is assumed, and a few key rays sufficient to define an area are traced. One procedure is to trace four rays to produce a quadrilateral ray tube cross section. Figure 3-40 shows the usual four rays and an area cross section looking up the ray tube toward the aircraft. The rays are numbered for identification. Ray 2 is separated from Ray 1 by $\Delta\phi$, Ray 3 from Ray 1 by Δt, etc. These separations are differences in independent variables defining the ray, not physical dimensions.

a. Ray Identification b. Ray Tube Cross Section
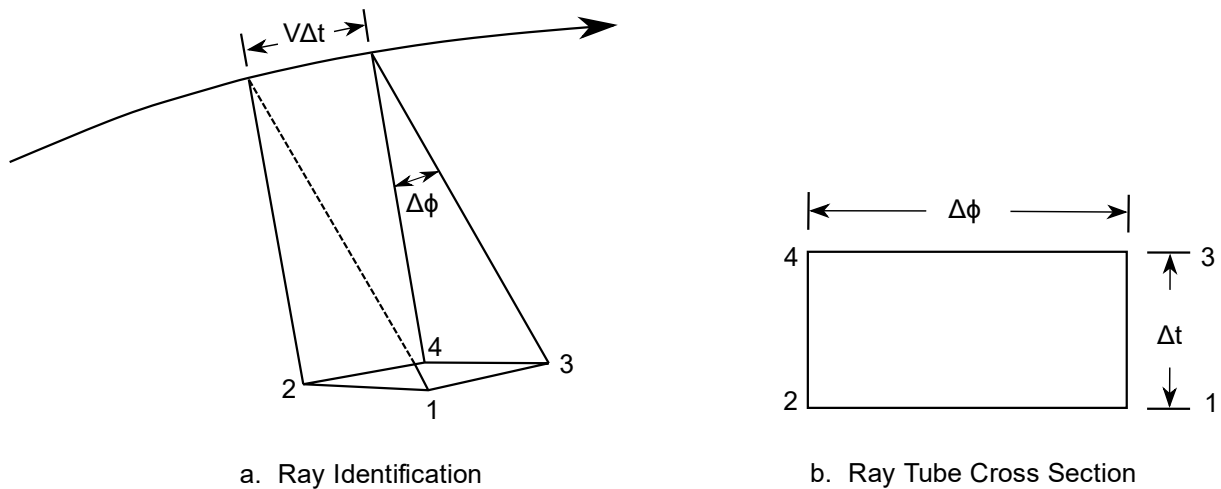
Figure 3-40: Parallelogram Ray Tube Defined by Four Rays.

Analytic expressions for the differential ray tube area can be utilized, or four distinct rays can be numerically traced in a finite difference scheme. For realistic aircraft maneuvers resulting in a focus, ray tube areas of zero indicate ray tubes collapsing to a line (or a point for a higher order focus). In a plane of symmetry, Rays 1 and 3 cross each other as do Rays 2 and 4, as shown in Figure 3-41(a).

Outside of a plane of symmetry, an initially rectangular ray tube does not necessarily remain rectangular as it focuses. Consider the two rays in Figure 3-39 to be Rays 1 and 3. At the focus circle, they do not cross. Figure 3-41(b) illustrates the ray cross section in this case. Ray 3 is displaced toward the center of the turn relative to Ray 1. There is a ray at $t + \Delta t$ which crosses Ray 1, indicated as Ray 5. For $\Delta\phi$, sufficiently small compared to gradient scales, the value of $\partial\phi$ corresponding to the ϕ difference between Rays 1 and 5 can be obtained by linear interpolation between Rays 3 and 4.

Figure 3-41 shows how $\delta\phi/\Delta t$ for crossing rays can be obtained at a known focal point. This defines a single variable family of rays, $\delta\phi$ as a function of Δt, which are to be followed in order to trace a focal surface. The curvature of the caustic traced by these rays defines the diffraction properties of the focus.

a. Focus in a Plane of Symmetry b. Focus not in a Plane of Symmetry
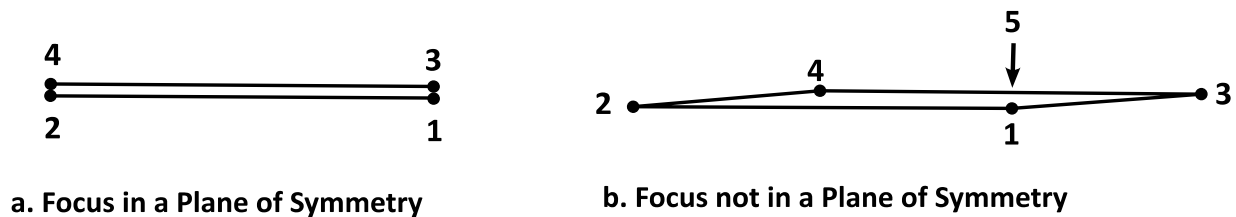
Figure 3-41: Ray Tube Cross Section at a Focus.

59

3.3.4.2 Sonic Boom Signatures at a Focus

3.3.4.2.1   Behavior of a Signal at a Focus

Figure 3-42 sketches signatures at a cutoff caustic, in ray/atmosphere (a) and wave (b) frames of reference. A normal N-wave increases in amplitude as it approaches the focal region along the ray, reaching maximum amplitude at the caustic. Since diffraction plays a role in limiting the maximum amplitude, high frequencies are amplified more than low, resulting in a U-shaped maximum focus signature. A low frequency evanescent wave propagates into the shadow zone below the caustic, and a high frequency post-focus U-wave emerges along the ray beyond the focus. The incoming N-wave is the incident wave, the high frequency post-focus U-wave is "reflected" off of the caustic "soft surface", and the low frequency evanescent wave penetrates the "reflective" caustic. The maximum focus U-wave is limited by a combination of diffraction and nonlinear effects.
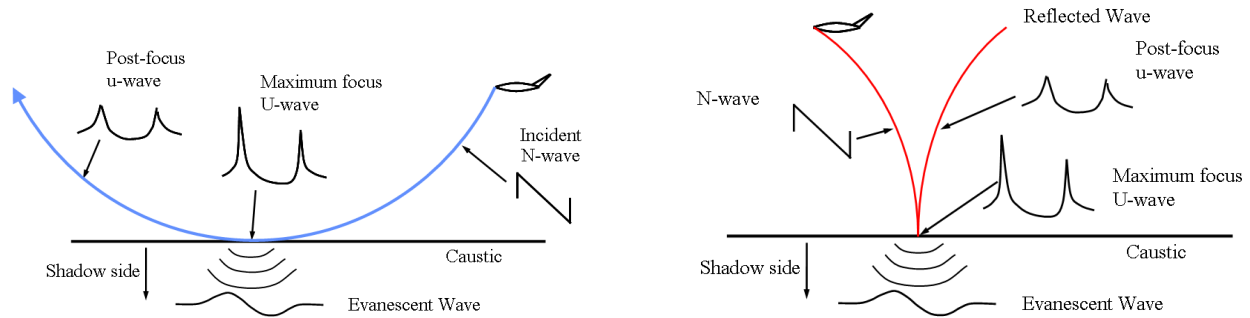
Figure 3-42: Focus at a Cutoff Caustic. a. Atmosphere fixed frame b. Wave fixed frame
(*Source: Plotkin, Page, and Haering* [5]*; reprinted with permission*).

Figure 3-43 is in the same reference frame as Figure 3-42b and is oriented to the perspective of observers on the ground. From left to right, the aircraft is transitioning from subsonic to supersonic speed. This image demonstrates the development of the on-track prefocus, focus, and post-focus carpet boom regions resulting from accelerated flight. The sonic boom pressure signatures that would be observed at various positions along the ground within each region are illustrated. Also shown is the development of the bow shock and focus caustic line and their intersection with the ground as the aircraft accelerates. In the prefocus region ahead of the focus caustic line, the pressure disturbances are perceived as low rumbles that increase in amplitude as the focus is approached. Within the focus region, which may be on the order of 100 m – 200 m in width, the intense U-shaped focus signatures are perceived as a loud bang-bang. In the post-focus region, three or four shocks may be experienced at intensities less than at the focus, perceived as a bang-bang pop-pop. The time between experiencing the two groups of shocks increases as the carpet boom region is approached. In the carpet boom region, the U-wave disappears due to atmospheric refraction and an observer experiences only the lower level bang-bang of the carpet N-wave.
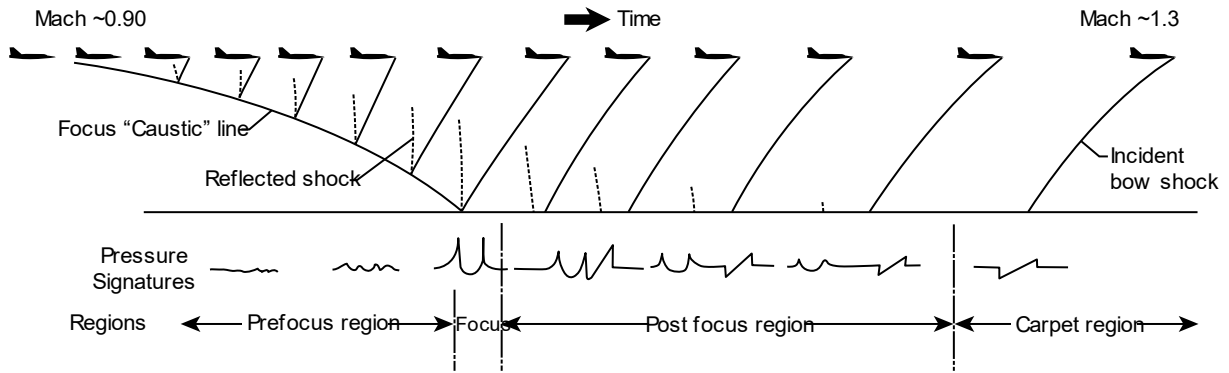
Figure 3-43: Pressure signatures along ground track for aircraft in transition flight (*Source: NASA-SP-622* [7]).

3.3.4.2.2 Scaling Law at a Focus

In Figure 3-42a, the region above the caustic is hyperbolic, the caustic is parabolic, and the region below the caustic is ellipsoidal. The flow is governed by the nonlinear Tricomi equation, which may be written as

$$\phi_{yy} - (y/R + \phi_x)\phi_{xx} = 0 \tag{45}$$

where

$\phi =$ the perturbation velocity potential
$R =$ a measure of the relative curvature between a ray and the caustic.

The similarity properties of this equation give rise to a major scaling law, first given by Guiraud [65]. If the overpressure at a reference location y_{ref} is C_{ref}, then the maximum overpressure at the focus is given by

$$\frac{C_{P_{max}}}{C_{P_{ref}}} \propto \left[\frac{y_{ref}}{RC_{P_{ref}}}\right]^{\frac{1}{5}}. \tag{46}$$

An amplification factor, therefore, can be defined as a function of reference conditions and R. Note that as $C_{P_{ref}}$ increases, the amplification factor decreases. The constant of proportionality must be obtained from theoretical solutions to Equation (45), or from experiment.

3.4 Footprint/Geometric Considerations

3.4.1 Lateral Cutoff

The initial ray cone was illustrated with a linear taper. While that is a reasonable approximation for short propagation distances, in reality, such a cone is usually distorted due to the effects of gradients in real atmospheres. In particular, isopemps do not extend indefinitely to either side, but stop at a lateral "cutoff" distance as a result of atmospheric ray refraction. The flight path perpendicular distance from left cutoff to right cutoff is referred to as the carpet width. Lateral cutoff can be characterized as the point where ray paths become tangent to the ground, driven largely by atmospheric temperature gradients causing the rays to refract away from the ground [7]. The ray path diagram of Figure 3-2 shows

rays emanating at azimuth angles below the aircraft as solid lines, with the ray at lateral cutoff reaching the ground at an approximate lateral distance of 25 nmi from the aircraft. Rays that originate at even higher angles will follow paths that turn up (away from the ground) without intersecting the ground. The lateral cutoff locations define the primary boom carpet. At ground locations beyond the lateral edge, geometrical acoustics predicts that sonic boom carpet waveforms should not occur, though acoustic rumbles and other distorted waveforms have been observed [7, 66].

Atmospheric conditions affect the lateral cutoff location and, hence, the carpet width as described by Maglieri et al. [7]. Specifically, deviations in temperature from a standard profile and winds are recognized as two key factors affecting the lateral cutoff. In general, temperatures lower than the standard level on the ground will increase the lateral extent of the carpet, while ground temperatures higher than standard will decrease the carpet width. Considering the effects of 2D profiles on carpet width, the general observation is that tailwinds increase the carpet width, whereas headwinds decrease the carpet width. Crosswinds tend to increase the lateral cutoff distance on the downwind side and decrease the lateral cutoff distance on the upwind side of the carpet.

3.4.2 Contouring

Visualizing variation in boom level across sonic boom carpets can be facilitated with contour plots. A key feature of the WCON module is to draw lines of equal metric levels over the plotted footprints in the WCON GUI. For each metric type computed by PCBFoot, WCON finds contour lines by treating metrics at each point in the footprint like elevation over a surface. Specifically, WCON maps points with coordinates defined by T_{ac}, ϕ onto a rectangular grid. For each of the six contour levels that WCON considers, the grid is methodically searched for grid squares situated over the respective levels. Vectors forming the shapes of the contours are formed by tracking metric gradients to identify grid faces through which contours enter or exit.

The WCON contour algorithm has similarities to ACM Algorithm 671 [67]. However, contour vectors returned by WCON are not smoothed, in contrast to the ACM algorithm. To avoid extrapolating metric levels outside of the predicted footprint, WCON also includes an option to not draw contours outside of the isopemp space.

3.4.3 Propagation over Terrain

The primary output of PCBoom is the boom at the ground (signature and location). Prior versions used only a single ground elevation for the entire carpet. The current version of PCBoom allows specification of varying terrain via an external file. Terrain data are supplied in the form of geo-referenced ground elevations with format specification described in the PCBoom 7.1 User's Guide [8]. As the program traces the primary ray on each tube, it checks the ground elevation below the current end. When the ray crosses below local ground elevation, the ray time step is linearly interpolated to obtain the point where the ray exactly intercepts the ground.

Figure 3-44 is an example of a carpet boom footprint over a mountainous region in the US near the Nevada and Utah border [68]. The impact of varying ground impedance on the boom signature itself is described in more detail in Section 3.4.4.
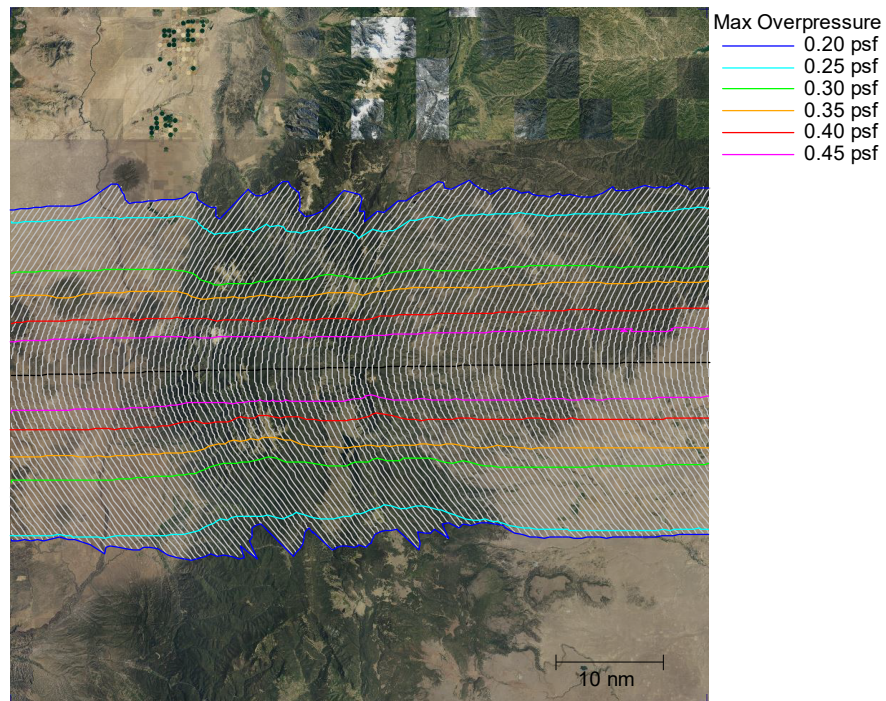
Figure 3-44: Example of Sonic Boom Propagation over Mountainous Terrain (*See also Rachami et al.* [68]).

Note that automatic determination of the lateral cutoff azimuth angle does not account for varying terrain. Specifically, if the FOBoom ray tracing input block has NPHIS set to zero, FOBoom will determine the range of azimuth angles to use based on a uniform ground elevation specified in the ALTITUDES keyword section. In that situation, ray tracing and propagation modeling checks terrain data for local ground elevation to determine if a ray has turned up before intersecting the ground, but the extreme azimuth angles may not represent the full range of rays that would have intersected the ground. More detail is given in Section 4.7.3.3.

3.4.4 Ground Impedance Effects and Elevated Receivers

Sonic boom propagation calculations are done using the free field assumption. Measurements of booms are traditionally (and properly) made at the ground, usually on a hard surface, so that direct and reflected waves constructively add. If the ground is perfectly hard, then the amplitude of the boom on the ground will be twice that of the free-field boom. Ground surfaces are rarely perfectly hard or smooth, so it is common to use a reflection factor slightly less than 2. The reflection factor is a PCBoom input parameter. A value of 1.9 is recommended for nominally hard ground. Water is nonporous and acoustically hard, so one should consider a reflection factor closer to 2.0 for boom impacts on water. If the free-field boom is of interest, then a factor of 1.0 can be used.

There are two situations where a simple reflection factor is not appropriate. One is for shallow elevation angle impact on soft ground, where the actual ground impedance must be accounted for. The other is for overwater analysis, where the boom below the water surface is of interest. Penetration of a sonic boom into water is discussed in Section 3.4.6.

In general, sound waves that reflect from the ground will constructively or destructively interfere with those propagating directly from the source. Most real ground types are partially reflecting, and, as such, the amplitude and phase of the reflected wave are also modified though interaction with the ground

63

surface. Physical characteristics of the ground type such as tortuosity and porosity affect the amount of attenuation of the reflected wave [69]. Furthermore, such ground interactions are frequency dependent and also vary with the geometry of the sound source and receiver relative to the ground surface. Effective flow resistivity, σ_{eff}, quantifies how material characteristics of the ground affect sound wave reflection. The scale of σ_{eff} is such that sound absorbing materials have low values, and highly reflective surfaces have high values. Typical examples of specific flow resistivity values are provided in Table 3-4, which shows that the scale of σ_{eff} varies several orders of magnitude depending on the ground type.

Table 3-4 Typical specific flow resistivity values.

Ground Cover	Specific Flow Resistivity (Rayls/m)
Snow Covered Ground	30
Forest Floor	50
Grassy Field	225
Roadside Dirt	650
Packed Sand	1,650
Hard Packed Dirt	3,000
Exposed Dirt/Rock	6,000
Asphalt	30,000
Water	1,000,000

Effective flow resistivity can be measured *in situ* using an acoustic point source and a system of two microphones at different heights. Procedures for that measurement are described by ANSI standard S1-18 [69], and details of a real-world implementation of σ_{eff} on pavement types can be found in a report by Hastings et al. [70].

Modeling ground reflections using specific flow resistivity and receiver height is implemented in PCBoom in the PCBFoot postprocessor. The ground properties are defined as a complex ground impedance in the form of flow resistivity whose details are discussed in the next few paragraphs. The PCBoom implementation of ground effects modeling depends on the incoming wave properties and local geometry and surface properties. The implementation treats the signal as a plane wave that negates the need for the details of the arbitrary ray paths and source location. The ground effect on the incident wave is implemented using the theory outlined in ANSI standard S1-18 [69]. The modeling method implemented in PCBFoot is outlined in the next paragraph.

Starting with the time history of the incident wave, the PCBoom implementation uses two identical, complex waveforms along with their corresponding Fourier transforms. One waveform represents the direct wave to the receiver at height h_r above the ground – the direct path waveform as depicted in Figure 3-45, and the other waveform represents the wave that is reflected from the ground – the reflected waveform.
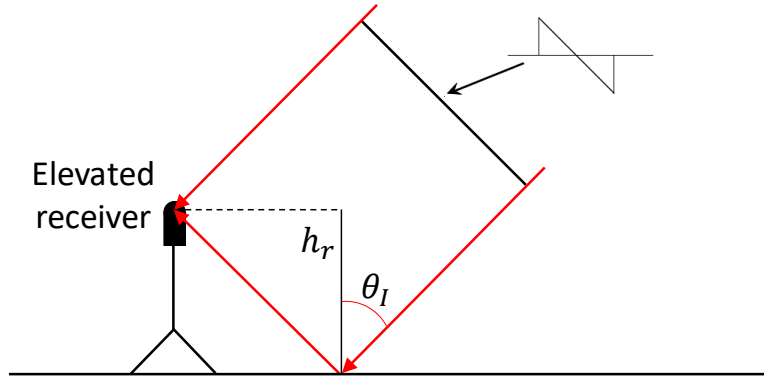
Figure 3-45: Plane Wave Ground Model Geometry.

The ground effect is applied as two corrections to the reflected waveform. The first is a phase shift equal to the time delay from the extra distance, r_2, the wave travels to the receiver. The equivalent phase shift as a function of frequency for a given distance is

$$dp(f) = \cos\left(2\pi f \frac{r_2}{c}\right) - i \sin\left(2\pi f \frac{r_2}{c}\right) \tag{47}$$

where

$$c = \text{ the speed of sound}$$

and

$$r_2 = 2h_r \cos(\theta_I) \tag{48}$$

where

$$h_r = \text{ Receiver height above local ground}$$
$$\theta_I = \text{ Ray incidence angle.}$$

This phase shift is added to the reflected waveform.

The second correction comes from the one-parameter model of the ground whose effective flow resistivity is represented by σ_{eff} at an angle θ_I as a function of frequency and is given by

$$R(f) = \frac{\cos\theta_I - z(f)}{\cos\theta_I + z(f)} \tag{49}$$

where $z(f)$, the reciprocal of the specific acoustic impedance ratio, is given by [69]

$$z(f) = \frac{1}{\left(1 + 9.08\left(\frac{f}{\sigma_{eff}}\right)^{-0.75}\right) + i\left(11.9\left(\frac{f}{\sigma_{eff}}\right)^{-0.73}\right)}. \tag{50}$$

The product of this correction and the reflected waveform gives the amplitude and phase as a result of the ground impedance. The inverse Fourier transform of the sum of the direct and reflected waveforms gives the pressure waveform at the receiver. The application of this methodology in the program is only valid when the reflected wave's arrival time delay is small enough to allow the direct and reflected

signatures to fit in the computational array. A modeled example of ground effect on an elevated microphone is given in Figure 3-46. The effect of interference between the reflected and direct rays can be observed in the interrupted rise of the bow and tail shocks.
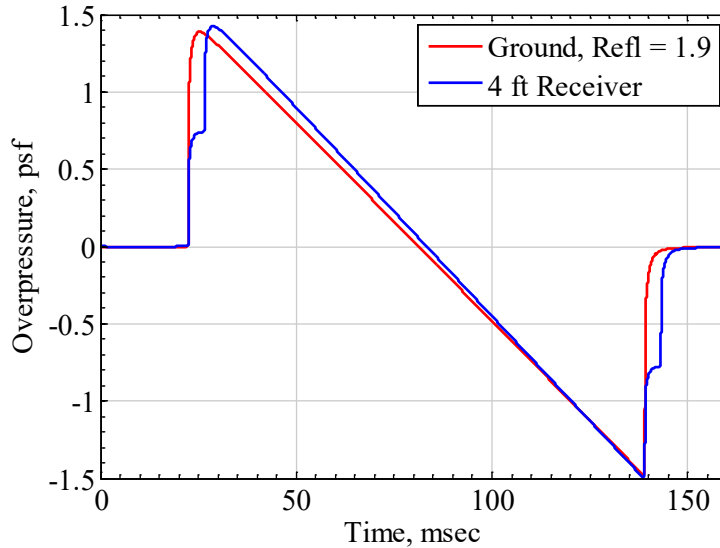
Figure 3-46: Comparison of Ground Waveform Modeling using Simple Reflection Factor = 1.9 and Waveform at Reciever Elevated 4 ft above Ground.

3.4.5 Footprint Signature Assembly

The modeling paradigm applied in FOBoom considers propagation of rays originating at distinct points in time (T_{ac}, or time at aircraft). For a simple case of steady level flight over flat ground with constant atmospheric conditions, the resulting ground isopemps form a regularly spaced series of identical hyperbolic lines as illustrated in Figure 3-6.

For a more complicated maneuvering flight, it is possible for isopemps to overlap or cross each other. The top half of Figure 3-47 shows isopemps from an aircraft in a supersonic turn from the top of the figure toward the right side of the figure. There are regions where the rays from later points in the trajectory land near the edge of the isopemps from early points in the trajectory. One such point is marked in Figure 3-47 with a red circle. A ground observer would experience two booms, one from each isopemp. Because the aircraft is traveling faster than the speed at which acoustic rays propagate, the boom from the later trajectory time can actually reach the ground before the boom from the earlier trajectory time.

Following the propagation modeling in FOBoom, the postprocessor PCBFoot can be used to assemble a boom footprint. One key aspect of that assembly is using ground boom arrival times from each ray to form ground signatures, which may be composed of more than one boom signature. The ground signature plot in the bottom part of Figure 3-47 corresponds to the red circle in the carpet above and shows a distinct N wave (from the near-center ray originating at the later trajectory point) and a U-shaped post-focus wave from a high azimuthal emission angle originating at the earlier trajectory point.

66
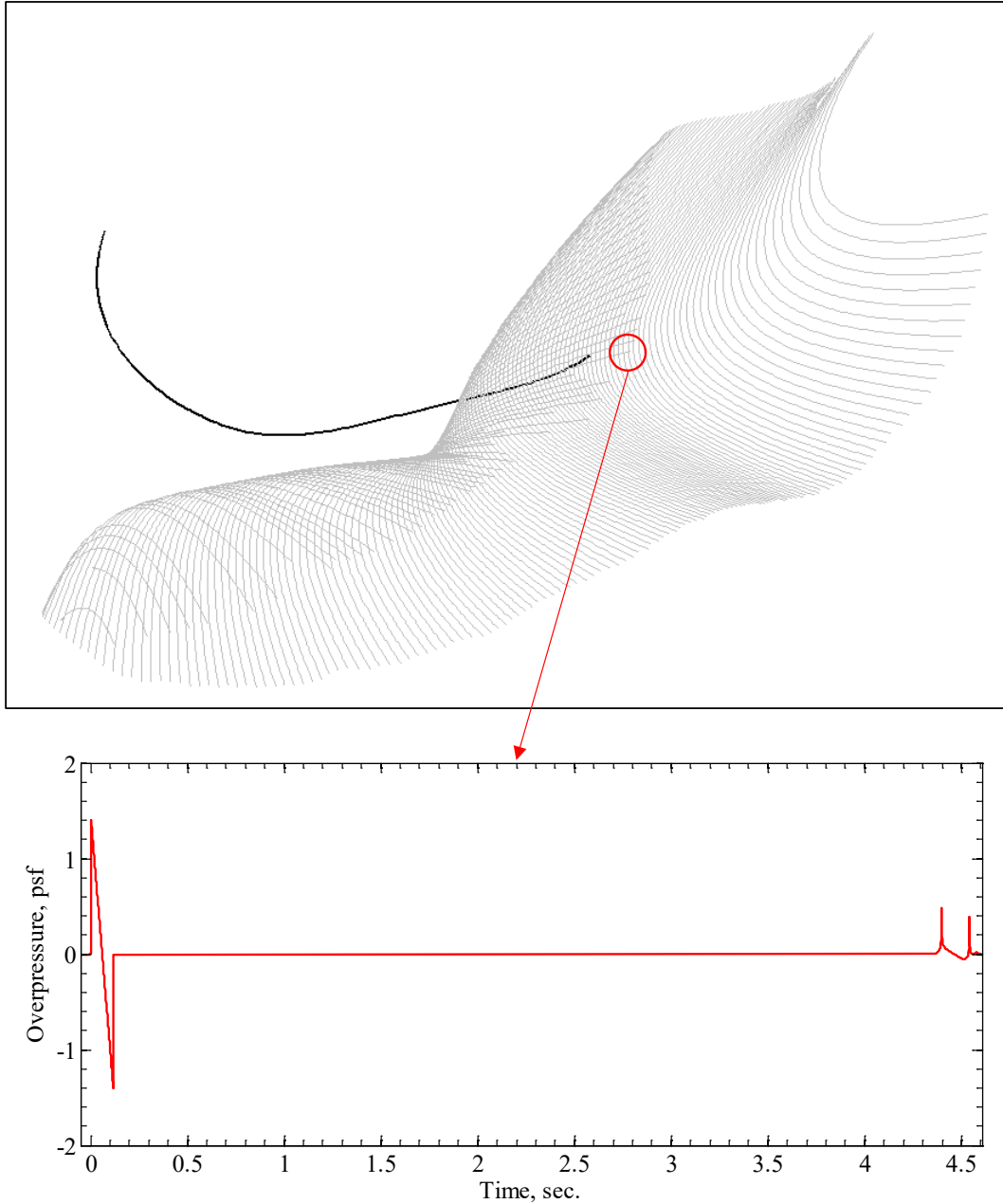
Figure 3-47: Sonic Boom Carpet for Maneuvering Flight (top) and Combined Ground Signature Showing Contributions from Two Ground Booms (bottom).

3.4.6 Underwater Penetration of Sonic Booms

Signatures below the surface of the ocean may be computed using the theory of Sparrow and Ferguson [71]. Building on earlier work by Sawyers [72] and Cook [73], they determined that the frequency component ω for a boom from an aircraft in level flight at Mach number M would penetrate into the ocean as

$$P(z) = P(0)e^{-z\mu k_0} \tag{51}$$

where

$$z = \quad \text{depth}$$

$$k_0 = \frac{|\omega|}{c_{air}M}$$

$$c_{air} = \text{sound speed in air}$$

$$\mu = \sqrt{1 - \frac{c_{air}^2 M^2}{c_{water}^2}}$$

$$c_{water} = \text{sound speed in water.}$$

Equation (51) is implemented in WCON, the footprint and signature viewing module of PCBoom. When a boom is displayed, pressing the down cursor moves the receiver position to successive depths below the water surface. Equation (51) is applied to each frequency component of the Fourier transform of the surface boom signature, similar to that used for the absorption part of the Burgers equation.

Sparrow and Ferguson's model applies to penetration into a deep calm ocean. There are two other, more complex cases of interest: a wavy ocean, and a shallow ocean near a shoreline. PCBoom contains a code to write a surface boom file that interfaces to Cheng's wavy ocean model [74]. However, Cheng's software is not available as part of the PCBoom package.

The second more complex case of interest is a boom impacting a shallow water region along a coastline. Garrelick has developed a model for such littoral regions [75]. This application requires details of the geometry of the shoreline and the ocean bottom, and is beyond the scope of PCBoom. PCBoom may be run to obtain the surface boom, which can then be used as input to Garrelick's model.

3.5 Metrics

Sonic boom metrics are calculated from pressure waveforms in the following modules: PCBFoot, WCON, FiltView, PCBurg, and HeadlessBurgers. All of these modules use a common set of subroutines for metric calculations, though, as described in Section 3.3.3, only a subset of metrics is calculated by FiltView. Metrics reported by PCBoom include ASEL, BSEL, CSEL, DSEL, ESEL and PL. Except for the CSEL, these metrics along with ISBAP are the six metrics anticipated to be used in development of single-event noise dose-response curves in future sonic boom community response tests [76, 77]. As will be described below, ISBAP can be calculated from ASEL, CSEL, and PL in a straightforward manner.

One of the metrics is P_{max}, which is the maximum overpressure in the signature reported in units of pounds per square foot (psf). It is also known as the peak overpressure. This is the traditional physical metric for N-wave sonic booms. WCON additionally reports P_{min}, which is the minimum overpressure in the signature.

L_{pk} is the peak overpressure expressed in terms of decibels relative to $P_{ref} = 20\ \mu\text{Pa}$ and is given by

$$L_{pk} = 20 \log_{10}\left(\frac{P_{max}}{P_{ref}}\right). \tag{52}$$

ZSEL is the sound exposure level (SEL) [78] with equal weight given to all frequencies, i.e., Z-weighted sound exposure level. ZSEL is reported in units of dB.

ASEL, BSEL, CSEL, DSEL, and ESEL are sound exposure levels calculated from acoustic energy spectra to which frequency-weighting networks [79] have been applied. Specifically, ASEL is the A-weighted sound exposure level, BSEL is the B-weighted sound exposure level, etc. All weighted SEL metrics are reported in terms of dB.

PL is a loudness metric based on Stevens Mark VII Perceived Level of Loudness [80]. Application of Mark VII loudness to sonic booms produced by aircraft is described in detail by Shepherd and Sullivan [81]. In calculating PL, one-third-octave band sound energy levels are normalized by a time constant of 70 ms, which is the critical time of the human auditory system. The PCBoom implementation of PL uses the ANSI-preferred base-10 system [82] to calculate one-third-octave band spectra using an FFT-based approach. Calculation of PL follows the procedure prescribed by Jackson and Leventhall [83]. For an ideal N wave with shocks separated by a time period longer than 70 ms, the human ear will perceive two distinct events. Because spectra are calculated using the full signature for this two-event case, the sound pressure levels are reduced by 3.0 dB to account for perception of two distinct events [84]. That approach produces metric levels consistent with the PL metric algorithm in the NASA Loudness Code for Asymmetric Sonic Booms (LCASB) [85].

Indoor Sonic Boom Annoyance Predictor [86] (ISBAP) has been described as a hybrid metric, in that it uses weighted SELs to apply a low-frequency correction to PL. ISBAP was developed empirically from NASA laboratory subjective studies and is defined as

$$ISBAP = PL + 0.4201(CSEL - ASEL) \tag{53}$$

4 Main PCBoom Program (FOBoom)

The main program in the PCBoom suite of sonic boom propagation modules is FOBoom. FOBoom computes ray paths and propagates boom signatures through the specified atmosphere over the user-specified azimuthal acoustic emission angles for a defined flight trajectory using either the three-dimensional ray tracing equations based on geometrical acoustics, or the flat-Earth layered ray equations used by Thomas for ray path determination. FOBoom predicts sonic boom waveforms from a Burgers equation, which is numerically solved using methods described in Section 3.3.2.

This section describes the link between the physics described in Section 2 and the PCBoom architecture described in Section 2 for the FOBoom module. The core of the FOBoom computations is described in Section 4.2. The sections that follow describe additional computations, which are invoked by specific keywords.

4.1 Summary of Inputs

Many fundamental parameters are defined within the FOBoom input file including atmosphere, altitude, terrain, ray tracing, nearfield signature or vehicle, and trajectory information. These parameters are input through a sequence of keywords within the FOBoom input file and supplemental files. The complete list of available keywords and input requirements for each keyword are described in the PCBoom User's Guide [8].

4.2 Base Calculations

This section describes the link between the physics described in Section 2 and the PCBoom architecture described in Section 2 for the core computations within the FOBoom module. Where appropriate, the various input parameters, over which the user has control, are described.

4.2.1 Geometry and Coordinate Systems

PCBoom has the capability to work with trajectory points provided in either local Cartesian coordinates (x, y, z) or geographic coordinates (latitude, longitude, altitude). Conversion between Earth-fixed Cartesian and geographic coordinates follows the raytracing mode selected (see Section 4.7.1), including nonflat Earth models used in some Schulten-based [87] raytracing methods, as detailed below:
- raytracing modes LEGACY and SCHULFLAT use a flat Earth model,
- raytracing mode SCHULSPHERE uses a spherical Earth model with radius of 20,888,146.39 ft (6366.71 km) and the assumption that the arc length corresponding to 1 minute is equal to 1 nautical mile,
- raytracing mode SCHULELLIPSE uses an ellipsoidal Earth model following the WGS-84 global reference system.

4.2.1.1 Flat Earth coordinate conversions
For the flat Earth mode, conversion from Cartesian coordinates to geographic coordinates is

$$\text{lat} = \text{lat}_0 + \frac{y}{R_E}$$
$$\text{lon} = \text{lon}_0 + \left(\frac{x}{R_E \cos(\text{lat})}\right) \tag{54}$$
$$\text{alt} = z$$

where

$$\text{lat} = \text{latitude (radians)}$$

$$\begin{aligned}
\text{lon} &= \quad \text{longitude (radians)} \\
\text{alt} &= \quad \text{altitude (ft)} \\
\text{lat}_0 &= \quad \text{latitude reference point (radians)} \\
\text{lon}_0 &= \quad \text{longitude reference point (radians)} \\
x, y, z &= \quad \text{Cartesian coordinates (ft)} \\
R_E &= \quad \text{Radius of the Earth based on spherical approximation} \\
& \qquad \text{(20,888,146.39 ft).}
\end{aligned}$$

For a geomode format trajectory, the coordinate reference point is the first trajectory point. Otherwise, the origin is specified by the longlat keyword. For the flat Earth mode, conversion from geographic coordinates (latitude, longitude, altitude) to Cartesian coordinates (x, y, z) is done using

$$\begin{aligned}
x &= (\text{lon} - \text{lon}_0)R_E \cos(\text{lat}) \\
y &= (\text{lat} - \text{lat}_0)R_E \\
z &= \text{alt.}
\end{aligned} \tag{55}$$

4.2.1.2 Spherical Earth coordinate conversions

For spherical Earth mode, conversion from Earth-Fixed Geocentric (EFG) Cartesian coordinates to geographic coordinates is given by

$$\begin{aligned}
\text{lat} &= \sin^{-1}\left(\frac{z_q}{r_q}\right) \\
\text{lon} &= \tan^{-1}\left(\frac{y_q}{x_q}\right) \\
\text{alt} &= r_q - R_E
\end{aligned} \tag{56}$$

where

$$\begin{aligned}
\text{lat} &= \quad \text{latitude (radians)} \\
\text{lon} &= \quad \text{longitude (radians)} \\
\text{alt} &= \quad \text{altitude} \\
x_q, y_q, z_q &= \quad \text{EFG Cartesian coordinates} \\
r_q &= \quad \text{Radial position, } \sqrt{x_q^2 + y_q^2 + z_q^2} \\
R_E &= \quad \text{Radius of the Earth based on spherical approximation} \\
& \qquad \text{(20,888,146.39 ft).}
\end{aligned}$$

For spherical Earth mode, conversion from geographic coordinates to EFG Cartesian coordinates is

$$\begin{aligned}
x_q &= r_q \cos(\text{lat}) \cos(\text{lon}) \\
y_q &= r_q \cos(\text{lat}) \sin(\text{lon}) \\
z_q &= r_q \sin(\text{lat}).
\end{aligned} \tag{57}$$

4.2.1.3 Ellipsoidal Earth coordinate conversions

For ellipsoidal Earth mode, conversion from EFG Cartesian coordinates to geographic coordinates is via the method of Sofair [23, 24]. Details of that method are not repeated here; implementation in PCBoom follows Reference [24] with the following notes.
- The WGS-84 system is used with values
  - Semimajor axis $a = 20,925,646.325459316372871$ ft
  - Semiminor axis $b = 20,855,486.595144357532263$ ft

- To protect against numerical errors, the argument to the inverse sine function used in calculating latitude is coerced to the range [-1,1] if necessary
- The $q > 0$ criterion from [24] is not enforced in PCBoom

For ellipsoidal Earth mode, conversion from geographic coordinates to EFG Cartesian coordinates follows ellipse geometry equations. The first step is to convert geodetic latitude to geocentric latitude, with special treatment near the poles using

$$
\begin{aligned}
\text{lat}' &= \tan^{-1}\left(\left(\frac{b}{a}\right)^2 \tan(\text{lat})\right), |\text{lat}| < 89° \\
\text{lat}' &= \text{sign}(\text{lat})\cos^{-1}\left(\left(\frac{a}{b}\right)^2 \cos(\text{lat})\right), |\text{lat}| \geq 89°
\end{aligned}
\tag{58}
$$

where

$$
\begin{aligned}
\text{lat}' &= \text{geocentric latitude (radians)} \\
\text{sign}(\text{lat}) &= \text{Function which returns the sign of the geodetic latitude} \\
a &= \text{Semi-major axis of reference ellipse} \\
b &= \text{Semi-minor axis of reference ellipse.}
\end{aligned}
$$

The radius of Earth $R_{E,c}$ at the current latitude is obtained using

$$
R_{E,c} = \sqrt{\frac{(a^2 \cos(\text{lat}))^2 + (b^2 \sin(\text{lat}))^2}{(a\cos(\text{lat}))^2 + (b\sin(\text{lat}))^2}}.
\tag{59}
$$

The coordinates in geocentric polar geometry at zero longitude are

$$
\begin{aligned}
x' &= R_{E,c}\cos(\text{lat}') \\
z' &= R_{E,c}\sin(\text{lat}').
\end{aligned}
\tag{60}
$$

The corresponding geocentric point in space at zero longitude is

$$
x_0 = x' + \text{alt} \times \cos(\text{lat}).
\tag{61}
$$

EFG Cartesian coordinates are then given by

$$
\begin{aligned}
x_q &= x_0 \cos(\text{lon}) \\
y_q &= x_0 \sin(\text{lon}) \\
z_q &= z' + \text{alt} \times \sin(\text{lat}).
\end{aligned}
\tag{62}
$$

4.2.1.4 Coordinate conversions in trajectory projection

The incremental projection of trajectory coordinates in PCBoom is based on a conversion between longitude/latitude and Cartesian (x, y) with an (x_0, y_0) origin and is given by

$$
\begin{aligned}
\text{lat} &= \text{lat}_0 + \Delta y \, \frac{180}{\pi \times R_E} \\
\text{lon} &= \text{lon}_0 + \Delta x \left(\frac{180}{\pi \times R_E \cos\left(\frac{\pi \times \text{lat}}{180}\right)} \right)
\end{aligned}
\tag{63}
$$

where

$$
\begin{aligned}
\text{lat} &= \text{latitude (decimal degrees)} \\
\text{lon} &= \text{longitude (decimal degrees)} \\
\Delta y &= y_1 - y_0 \\
\Delta x &= x_1 - x_0 \\
\text{lat}_0 &= \text{origin corresponding to } (x_0, y_0) \text{ decimal degrees} \\
\text{lon}_0 &= \text{origin corresponding to } (x_0, y_0) \text{ decimal degrees} \\
R_E &= \text{Radius of the Earth based on spherical approximation} \\
&\quad (20{,}888{,}146.39 \text{ ft)}.
\end{aligned}
$$

4.2.2 Signature Aging (Lossless Burgers Solver)

The legacy signature aging is, in essence, a lossless Burgers solver and is computed via the Middleton–Carlson method [30] also reiterated by Hayes, Haefeli and Kulsrud [29] as described in Section 3.3.1.

The FOBoom implementation is as follows:
1. Obtain the starting *F*-function
2. Propagate the rays away from the aircraft and calculate the ray tube areas
3. Compute caustic geometry if a focus condition is encountered
4. Calculate the Age parameter integrand and the Blokhintzev parameter for each point along the ray, using ray tube areas from step 2 as input
5. Evolve the signature along the ray by aging the signature (using the starting *F*-function) and applying the amplitude factor (based on the Blokhintzev parameter)
6. Calculate focus signatures as needed using the Gill and Seebass' focused shock wave solution

Step 1. The starting *F*-function is dependent on the source modeling mode as is described in Section 3.1. A user can prescribe pressure, *F*-function, or specify a particular aircraft type. Internally, all options result in transposition of starting signatures into an *F*-function, which is used in the calculation of signature aging. It is possible to scale the starting *F*-function to other conditions (weight, length, atmospheric conditions) for some MODES (see Section 4.5).

There are 8 modes in FOBoom Legacy for defining the vehicle and used in determining the *F*-function. They are controlled by a numeric flag, which is defined by the MODE keywords in the FOBoom input file (see the PCBoom User's Manual for more information on available modes and keywords [8]); each of which is explained in Table 4-1.

<div align="center">Table 4-1 FOBoom starting signature modes.</div>

Flag	Description	Modes
0	Pressure input directly	MODE/THOMAS MODE/CFDDPP
1	*F*-Function input directly	MODE/FFUNC MODE/CARLSON /ACNAME MODE/CARLSON/AXISYMMETRIC MODE/CARLSON/BASIC MODE/CARLSON/LAUNCH MODE/CARLSON/LOOKUP MODE/SEEB
2	Obtain general *F*-function	Derived using user-supplied area and lift distribution routines
3	DPP read directly from CFD cylinder	MODE/CYLINDER MODE/CYLINDER_ASYMM

		MODE/CYLINDERHIRES
4	Hypersonic blunt body *F*-function	MODE/TIEGERMAN
5	Mach interpolated CFD, regular	MODE/CYLINDERM
6	Mach interpolated CFD, irregular	MODE/CYLINDERMF
7	G-load interpolated non uniform X CFD	MODE/CYLINDERX MODE/CYLINDERXHIRES
8	Non uniform X CFD	MODE/CYLINDERVARYX

<u>Flag 0</u>: Original Thomas form, where pairs of $\Delta p/p_\infty$ and X are read. This mode is applicable at only one azimuth and flight condition. Supporting input data are number of points in the signature (NX), $\Delta p/p_\infty$, axial distance (X), aircraft length (l) and model length (ML). The input $\Delta p/p_\infty$ is multiplied by the local atmospheric pressure at the trajectory altitude to obtain the starting pressure signature. In this mode the *F*-function is determined by scaling the simple input via lift factor (K_L) and shape factor (K_s) parameters and then converted to a pressure signature (Δp) at the user specified starting radius.

<u>Flag 1</u>: This is a simple *F*-function mode. $\Delta p/p_\infty$ and X pairs are read, but $\Delta p/p_\infty$ is considered to be the *F*-function under the aircraft at flight condition (Mach and pressure at altitude) for steady level flight. The input *F*-function is scaled to other loads, flight parameters, and azimuths by use of Carlson's formulae. Supporting input data are l, ML, aircraft weight (IWT), reference pressure ($PVFFN$), and reference Mach number ($EMREF$). The index of the Carlson shape factor curve (IUN) is input by the user. The *F*-function is converted to the initial pressure signature based on atmospheric parameters at the user specified starting radius.

This flag is also utilized for Carlson *F*-function mode. The *F*-function is taken to be Carlson's simplified equivalent N-wave shape. Supporting data are l, IUN, and IWT. Weight is read as a floating-point value in kilopounds and the integer quantity IWT is extracted from this. Carlson's method then results in the determination of a shape factor (K_s). K_s is then converted into the starting *F*-function using Carlson's formulae as illustrated in Figure 4-1 using the vehicle length, l. For more information on the Carlson mode options, see Section 4.5.
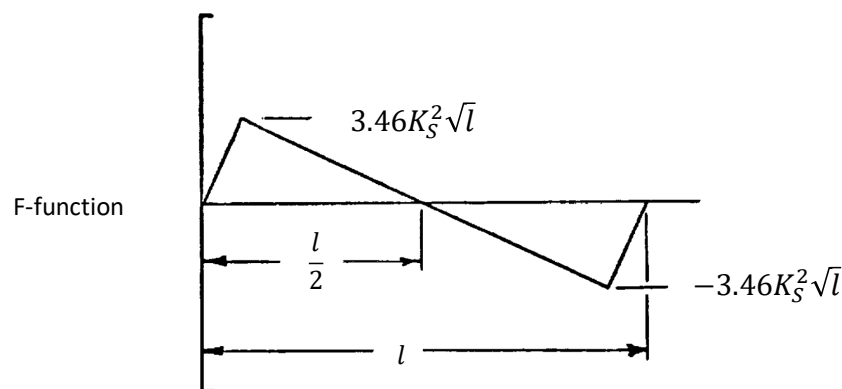
Figure 4-1: Carlson's *F*-function based on Shape Factor (*Source: NASA TP 1122* [9]).

<u>Flag 3</u>: Input data are in the form of a CFD solution on a cylinder. These are read and processed into multipole source distributions and equivalent *F*-functions at the starting radius. Subroutine NERDAT assembles $\Delta p/p_\infty$ at the required azimuth, and scales this to the starting radius.

MODE/CYLINDER_ASYMM allows for asymmetric aircraft cylinder starting signatures. A weighted angle cubic spline routine is used to interpolate the provided azimuthal distribution to the desired azimuth used in the propagation calculation. The method is cubic through two points with the derivatives given by weighting the straight-line angles, not the slopes. A description of that interpolation algorithm is given in Appendix A. This mode requires the same axial distribution of points down the cylinder across all input azimuth angles.

<u>Flag 4</u>: This mode sets up the *F*-function for hypersonic blunt bodies using Tiegerman's theory. When using this mode for launch vehicles, thrust, plume drag, and angle of attack may be prescribed by the user. This mode has the following three extensions.
1. "Launch vehicle mode": IUN values from 9-16 can be used to point to Carlson curves 1-8, respectively and trigger a trailing plume computation.
2. By using an IUN value of 17, an axisymmetric shape factor can be directly supplied by the user via a NEWLOAD data line.
3. By using an IUN value of 18, the shape factor can be looked up from a table supplied by the user.

<u>Flag 5</u>: This mode is for multi-Mach CFD cylinder input. It requires that all cylinders have the same x and azimuth mesh. A two-way linear interpolation on Mach and azimuth is computed to get the starting $\Delta p/p_\infty$ pressure distribution. This is then scaled to the starting pressure distribution at the cylinder radius based on the input trajectory and atmospheric parameters.

<u>Flag 6</u>: This mode is for a multi-Mach CFD cylinder mode that allows for uneven axial and azimuth mesh. A two-way linear interpolation on Mach and azimuth and axial station is computed to get the starting $\Delta p/p_\infty$ pressure distribution. This is then scaled to the starting pressure distribution at the cylinder radius based on the input trajectory altitude and atmospheric parameters.

<u>Flag 7</u>: This mode includes G-load interpolation and nonuniform axial spacing cylinder input. The $\Delta p/p_\infty$ is obtained as a function of azimuth angle by selecting the closest available azimuth angle and G-load number from the input CFD solutions. It is then scaled to the starting pressure distribution at the cylinder radius based on the input trajectory G-load, altitude, and atmospheric parameters.

<u>Flag 8</u>: This mode includes a single cylinder with different axial spacing at each azimuth and selects the closest azimuth angle for propagation. The starting $\Delta p/p_\infty$ is scaled to the starting pressure distribution at the cylinder radius based on the input trajectory altitude and atmospheric parameters.

For all modes with *F*-function inputs, the *F*-function amplitudes are scaled to local conditions so as to be Δp. When needed for input or output in FOBoom, length scales are multiplied by 1/M to convert to distance along a ray.

Step 2. The location at which signature aging starts is dependent on the MODE keyword. Calculations start at either the defined input cylinder radius (MODE / CYLINDER* or CFDDPP) or at user input ROVERL (radius divided by length) with default one body length (MODE / CARLSON, FFUNC, THOMAS, TIEGERMAN or SEEB).

Four rays corresponding to corners of the ray tube are propagated away from the aircraft source using geometrical acoustics as described in Section 3.2. The nature of the propagation is dependent on the atmospheric inputs (ATMOS) and the raytracing geometric mode (RAYTRACING / LEGACY, SCHULELLIPSE, SCHULSPHERE or SCHULFLAT). During the ray tracing, the rays are examined for cutoff. If any one of the four rays cut off in LEGACY mode, there is no ray tube beyond that point, and the area calculation halts. The Schulten ray tracing modes, which are used for over-the-top booms, continue tracing rays which propagate upwards. If a focus is detected (by the ray tube area vanishing), the point is marked so that caustic tracing is initiated in the main program. If not, signature aging calculation follows the method for a standard sonic boom. Conditions of encountering sonic boom cutoff or having no focus for an auxiliary ray in a focus case (i.e., the caustic has run into the ground) are marked with IFLAG being set to 1 or 2, respectively. If a focus is detected but it is above ZMAX (the user-specified altitude above which focus booms are not considered), IFLAG is set to 4.

For the case of propagating the first ray with focus below ZMIN (the user-specified altitude below which focus booms are not considered), there is no focus at the ground and FOBoom will evolve the signature to the ground point and output it to file. Continued signature evolution past the ground bounce point may be computed using the OTTER postprocessor.

Ray tube areas are computed using the cross product between the diagonals (opposite corners) of the primary and three incremental rays as described in Figure 3-14 and implemented as Equation (91). FOBoom utilizes the original formula from the starting version of the Thomas code [1]. If the ray tube areas are zero or negative, the point is identified as a caustic.

Step 3. The legacy focusing method based on empirical N-wave based methods is described in Section 4.2.3. A supplemental method is to use the LNTE postprocessor (Section 6.2) for calculation of the signature aging and evolution in the vicinity of the focal zone.

Step 4. The Age parameter and the Blokhintzev parameter are computed for each point along the primary ray as described in Section 3.2.1. These parameters are output to the .age or .ott files (enabled by the OTT keyword or OUTPUTS / AGE keyword option).

Step 5. Signature evolution is conducted at each step along the primary ray based on the formulation presented in Section 3.3.1.2. The method is implemented using a fine point spacing (based on the signature length / 1000), since the integral function S_0 (Figure 3-22) is quadratic in the advancing direction. Straight-line regions of $F(\tau)$ are generally curved in s, although they remain straight in the aged F-function. The input F-function signature is interpolated to finer spacing before application of the aging. When the final aged signature (with inserted shocks) is obtained, points are culled down to those points, which are from the original input signature, plus any additional points defining the inserted shocks.

The fine point spacing is accomplished by redistributing the input F-function to a larger array and then interpolating fill points. A leading and trailing buffer is applied that is long enough to encompass the biggest stretch in the signature. Current implementation uses five times the starting signature length. When the input F-function is mapped to the larger array, the original starting point indices are retained. The FOBoom logic creates fill points using approximately 1/1000 of the signature length. The tolerance for determining overlap of the segments (for shock formation) is the signature length divided by 1000. The effective length of a boom signature is computed via the Sparrow 5%-95% rule [63], which defines the effective signature length as the length between the signature points that correspond to 5% of the

total acoustic energy of the signal and 95% of the total acoustic energy of the signal. The 5% and 95% points are determined by integrating the *F*-function numerically from the beginning to each index along the signature. Integration is accomplished using the trapezoid rule. Evaluation of the cumulative energy at each axial point along the signature identifies the 5% and 9% locations, from which the effective signature length is obtained.

Step 6. Focal zones can be created by arbitrary dynamic, maneuvering flight and from the diffraction of rays due to atmospheric gradients. Modeled focus signatures in those focal zones are based on the method developed by Plotkin and Cantril [2] as described in Section 4.2.3. Propagation of focused signatures using the specialized tool LNTE is described in Section 6.2.

4.2.3 Focusing at a Smooth Caustic

It is possible to encounter focusing conditions for maneuvering flight. A focus is identified when the ray tube area approaches zero. When a focal zone is identified, PCBoom uses a scaling law that relates maximum overpressure to 1) the atmospheric reference conditions, 2) the measure of the relative curvature between a ray and the caustic and 3) a constant of proportionality.

Within FOBoom, the logic of tracing a caustic happens when the ray tube area calculations indicate a focus condition has been encountered. The nominal focus location is refined via interpolation along a curved ray. The trajectory position is moved so as to provide initial points for three auxiliary ray tubes. After all four tubes (and their focal point positions) are computed, the caustic curvature is obtained.

The signature calculation method described in Section 4.2.3.1 is applied to all altitudes requested in the input file (keyword ALTITUDES). The caustic tracing process is described in Section 4.2.3.2.

4.2.3.1 Signature Prediction at a Focus

Similar to the procedure employed in signature aging (Section 4.2.2) the incoming signature is examined and the shocks identified. Shock locations are identified based on the rise time with a threshold of 0.1% of the signature length. The signature is then redistributed onto a larger array with even spacing. For legacy calculations using simple N-waves a 100-point array is used. For inputs based on CFD or cylinder input files, the number of points used is commensurate to the starting pressure signature and is based on the average incremental spacing over the nonzero length of the starting signature. The signature is examined, multiple shocks within each spacing element are combined by selecting the maximum value, and a buffer space before and after the signature is added. The buffer space allows array space for the next step.

The Gill–Seebass canonical solution (Section 3.3.4.2.2) is then applied to each identified shock. This takes place in two stages. First, the amplitude of each shock and the signature behind it (up to the next shock) is multiplied by the amplitude scale factor. Each segment is treated relative to the pressure just ahead of its shock. In the initial development of FOBoom's focus methodology [2], it was shown that if the finite thickness of shock waves is accounted for, then theoretical and experimental values of the constant of proportionality between the overpressure at the reference location and the maximum overpressure at the focus (Equation (46)) used to determine the amplification factor are in excellent agreement.

If needed, a linear ramp is added to the end of the signal to return to zero pressure. The waveform is also adjusted in the data array to recover the unused leading space, allowing for two zero-pressure points at the beginning.

Determination of the edges of focal zones (Equation (69)) is implemented in FOBoom using equivalent N-wave shock strength. For a symmetric N-wave, the whole signature is uniformly scaled. However, the applicability of this simplified focusing model should be used with caution, as a cumulative DC offset error can occur for a complex asymmetric signature. Therefore, it is recommended that the LNTE module be employed to compute propagation of shaped booms or non-N-wave configurations in focal zones.

4.2.3.1.1 Numeric Solution of a Focus Signature

Gill and Seebass [88] have obtained solutions to the Tricomi equation for a step function incident wave, representing a thin shock wave. They began with a dimensional form of the nonlinear Tricomi equation in wave-fixed coordinates given by

$$\left[\frac{Y}{R} + \frac{\gamma + 1}{U}\Phi_x\right]\Phi_{xx} + \Phi_{yy} = 0 \tag{64}$$

where

$$U = \text{ the flow speed.}$$

In this model, the speed of sound varies linearly, and equals U at $Y = 0$. R in Equation (64) is twice the actual radius of curvature of the rays.

By applying Guiraud's similitude in the form of

$$\Phi = \frac{US^{5/3}}{(\gamma + 1)R^{2/3}}\phi$$

$$X = Sx \tag{65}$$

$$Y = R^{1/3}S^{2/3}y,$$

the nondimensional form

$$(y + \phi_x)\phi_{xx} - \phi_{yy} = 0 \tag{66}$$

is obtained. The linear characteristics of this equation are p and $q = x \pm y^{2/3}$, and the initial condition is specified as a function of p on $q = -1$ that yields

$$\phi_x(p) = -\mu y^{-1/4}F(p) \tag{67}$$

where

$$\mu = \text{ an amplitude parameter.}$$

In view of the preceding transformation, the solution should be independent of μ. Gill and Seebass [88] found that this was the case for values of μ from 0.00001 through 0.1. Retaining this parameter was useful to their numerical scheme, and a μ value of 0.1 was used.

The Gill and Seebass [88] computational approach was to introduce a coordinate strain transformation, which reduced it to the linear Tricomi equation. Linear solutions, in terms of Airy functions, were constructed and then transformed back to x,y coordinates. The procedure is analogous to Whitham's rule [15], except that two dimensions are involved. This numerical solution works well for N-waves and was subsequently validated by a flight test [89]. A more recent numeric solution by Auger and Coulouvrat [90] confirmed that this procedure is appropriate for N-waves, but demonstrated that the shocks in more complex signatures interact, reducing the peak focus overpressure.

4.2.3.1.2   Application to Focused Booms

Figure 4-2 shows the Gill–Seebass solution [88] with a shock fitted by area balance. Also shown is the initial $\mu = 0.1$ step function wave at $y = 1$. For this wave, x may be taken as p, the coordinate along the ray path. The focused signature $Cp = (-2/U)\,\phi_x$ is obtained by using the scaling transformation to convert back to physical coordinates. This figure also gives u for a step function signature. Sonic boom signatures can be considered as a series of shocks followed by expansion regions. The Gill and Seebass computation can be performed on a series of superimposed elements. However, because the slope of the expansion will change with scaling, a universal function cannot be obtained.

Since the higher frequencies associated with the shock wave focus more strongly, it can be argued that the shape of u is dominated by the jump and does not depend greatly on the signature behind it. The solution for $F(p)$, a shock followed by an expansion, is provided in Figure 4-3 where the difference between $u(x)$ and $F(p)$ is the same as in Figure 4-2. If this behavior is assumed, then the solution presented by Gill and Seebass [88] can be adapted to an arbitrary signature by a simple procedure in which the scaling is carried out based on the reference shock strength. The reference signature is adjusted in strength so that the shock is $u = -0.1$. Equation (64) is then applied to this modified u, using the reference value of $C_p(p)$ at the shock.
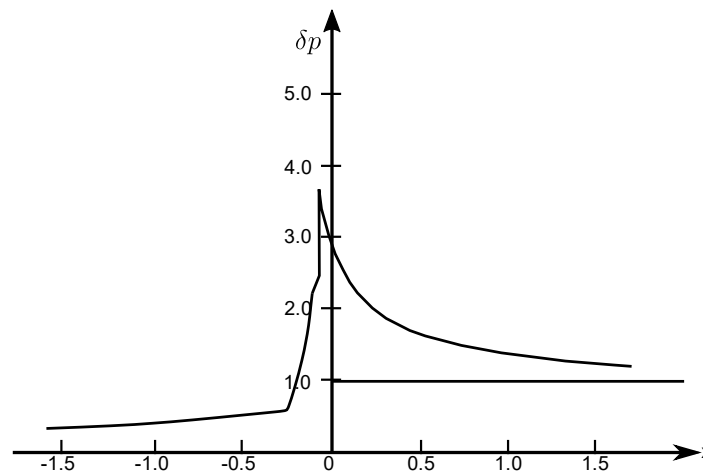
Figure 4-2: Gill–Seebass Solution for Focused Shock Wave (Unit Step Function) (*Source: Plotkin and Cantril, "Prediction of Sonic Boom at a Focus,"* [2]*; reprinted by permission of the American Institute of Aeronautics and Astronautics, Inc.*).
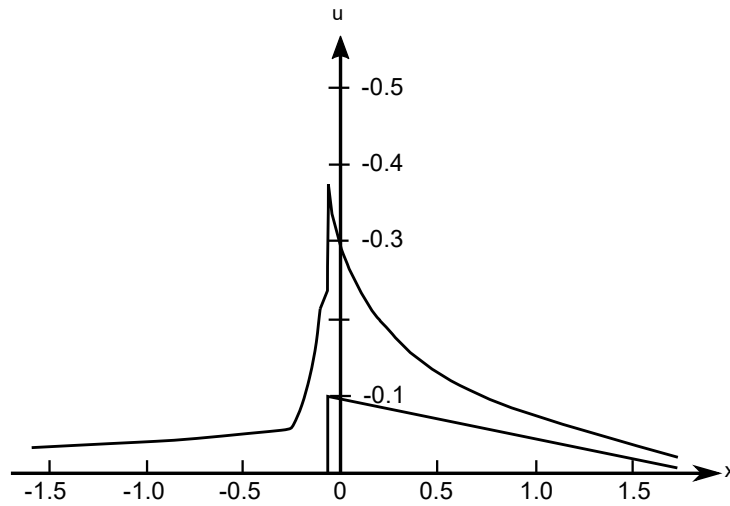
Figure 4-3: Solution for Step Function Applied to Leading Shock followed by an Expansion
(*Source: Plotkin and Cantril, "Prediction of Sonic Boom at a Focus,"* [2]; *reprinted by permission of the American Institute of Aeronautics and Astronautics, Inc.*).

4.2.3.1.3   Extent of Focal Zones

The extent of the focal zone normal to the focus is a parameter of practical significance. Precise definition of such a dimension is like defining a boundary-layer thickness, in that a somewhat arbitrary decision must be made to establish the edge. A simple definition may be obtained by comparing the geometrical incoming wave to the finite real focused signal, where Figure 4-4 is a sketch of what C_p vs. Y might look like. At large Y, the signal follows the curve corresponding to $C_p(p) \propto Y^{-1/4}$. At $Y = 0$, C_p is a finite value given by Equation (46). The transition between the two is included, although not to scale.
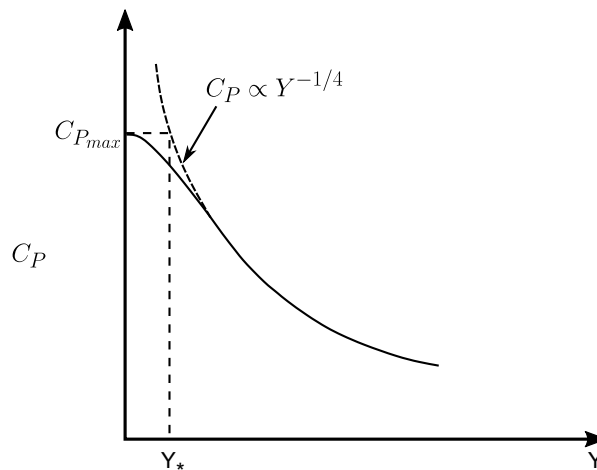
Figure 4-4: Definition of Focal Zone Width Y_*.

The location Y_* shown in Figure 4-4 corresponds to the point where the geometrical acoustic solution equals the focused signature at $Y = 0$. This gives a measure of the distance at which the geometrical solution gives pressures larger than the maximum true solution. For $Y > Y_*$, the geometrical solution

may be considered correct in magnitude, but incorrect in location, by an amount up to Y_*. The focal zone extent is, thus, given by

$$C_{p_{max}}Y_*^{1/4} = C_p(p)Y^{1/4} \tag{68}$$

where $C_{p_{max}}$ is $C_p(x)$ from Equation (46) using $u(x)$ for the maximum focus location. Inserting Equation (46) and solving for Y_* results in

$$Y_* = \left(\frac{\mu^{4/5}}{-u}\right)^4 \left[\frac{(\gamma+1)}{2}RC_p(p)Y^{1/4}\right]^{4/5}. \tag{69}$$

Note that Y_* increases for smaller u, so that the focal zone for signature elements near the focal zone will be smaller than for the lower pressure elements away from it. This is consistent with qualitative estimates of diffraction based on the wave parameter discussed earlier.

4.2.3.2 Tracing a Caustic

The crux of the focus calculation is obtaining the relative curvature R between the ray and the caustic. The local ray curvature is obtained by fitting a circle to three ray points around the focus. A similar process is performed for the caustic. This requires computing additional focal points. The method employed is illustrated in Figure 4-5. When at Point 1, the aircraft generates Focus Point 1 on the caustic. Aircraft trajectory parameters are used to move the aircraft to Points 2 and 3 to obtain the two additional focal points required to compute the caustic curvature.
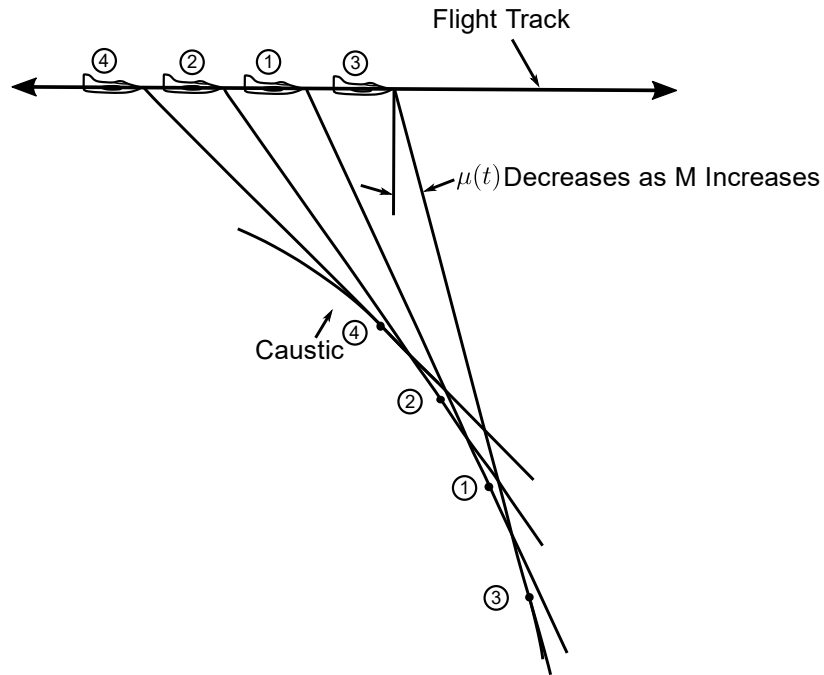
Figure 4-5: Tracing of Caustic Acceleration Focus (*Source: Adapted from Plotkin [12]; reprinted with permission*).

It is possible that Points 1, 2, and 3 straddle a cusp. Thus, Point 4 is computed. The arrival time sequence of a boom at the four points on a smooth caustic should be either 4, 2, 1, 3 or 3, 1, 2, 4. If this sequence test is passed, smooth caustic analysis proceeds.

This caustic tracing process has the following computational requirements in FOBoom:
- Spacing between trajectory Points 1, 2, 3, and 4 must be specified. Spacing must be large enough to obtain a reasonable base for accurate differences between focal points.
- Accurate calculation requires second derivatives of Mach number, flight path angle, and heading. Only first derivatives are necessary to account for the effect of maneuvers on nonfocusing booms.
- It is necessary to save ray quantities through this caustic tracing process for subsequent boom calculations.
- Control of ray tube azimuthal width and ray integration step size are needed, since these can have an effect on focal point precision.

In addition to these requirements, there are practical considerations concerning the relation between ray geometry, focal zone size, and the ground:
- Ray tracing must be carried below the ground. This is necessary to trace Point 3 in Figure 4-5. Below ground ray tracing is also necessary to avoid missing a focus if, for example, Focus 1 is very slightly below the ground. The "below the ground" extension must be larger than the expected focal zone size. The "below the ground" extension must be larger for Point 3 than for Point 1. Otherwise, the caustic tracing will fail.
- When testing for cutoff location (the point at which rays begin refracting upward), care must be taken to determine whether this point occurs above or below the ground. A nonfocusing boom on Ray 1 must not be inadvertently discarded if it turns upward below the ground. Auxiliary rays tracing a caustic can be important even if they cut off above the ground.
- A focus sufficiently far above the ground will result in small "postfocus" boom at the ground. It is worth having an upper altitude limit (ZMAX), above which computational effort will not be expended to trace such a focus if it is not of interest.
- For the symmetric case, only time varies between the four points. In asymmetrical cases, proper tracing of the caustic will involve variation of azimuth ϕ, as described in Section 3.3.4.1.5. The rate of azimuthal variation is determined from Focal Point 1.

4.3 *Keyword BURGERS*: Burgers Solvers

The BURGERS keyword in FOBoom indicates that a solution to a Burgers equation is needed that fully accounts for both the nonlinear and atmospheric absorption effects. There are two Burgers solvers available with varying levels of approximation. The first one is the Legacy Burgers solver, which is the first Burgers solver that was implemented in PCBoom. It is implemented in two standalone postprocessing programs, PCBurg (Section 5.1) and HeadlessBurgers (Section 5.1.3), and can be run with or without a graphical user interface to produce output for a single ray or multiple rays (HeadlessBurgers only). The other solver is the Enhanced Burgers solver. It is integrated into FOBoom (i.e., it is not a postprocessor module) and is significantly more efficient than the Legacy Burgers solver. It also accounts for wind effects including the Doppler and convective effects and can be used for a single ray or multiple rays.

If the BURGERS keyword is followed by two numerical entries in the succeeding line (see the PCBoom User's Guide for details [8]), the Enhanced Burgers solver will be invoked. Otherwise, FOBoom will generate the files required to run the Legacy Burgers solver via PCBurg or HeadlessBurgers.

4.3.1 Legacy Burgers Solver

The Legacy Burgers solver is used by two standalone post processing programs: PCBurg and HeadlessBurgers. PCBurg is a postprocessor employing a graphical user interface to view the evolution of a sonic boom waveform from an initial altitude to the ground while HeadlessBurgers is a postprocessor used to implement the Legacy Burgers solver in batch processing mode. HeadlessBurgers and PCBurg contain the same physics. Both postprocessors take in ray paths, age parameters, and signature *F*-functions generated by the main PCBoom program. Details of the implementation of the Legacy Burgers Solver in those modules are given in Section 5.1.

The Legacy Burgers Solver requires the following parameters defined along the ray (as calculated by the FOBoom module): starting signature, propagation time, sound speed, ambient temperature, ambient pressure, and the Blokhintzev parameter. These are written by FOBoom to a pair of files. The "ssg" starting signature file contains the *F*-function at the beginning of each ray. The "age" age parameter file contains the ray quantities as indicated, plus the age parameter so that the PCBurg and HeadlessBurgers solvers can calculate a thin shock signature at any time along the ray or even at the ground. The parameters in the age file are written at 1000-foot intervals from the flight altitude to the ground. Generation of the ssg and age files is triggered by the BURGERS keyword in the input file or by a value of 16 in the command line Boolean option codes.

4.3.2 Enhanced Burgers Solver

4.3.2.1 Implementing Equations
The approach used in the Enhanced Burgers solver follows the approach developed by Lonzaga [6]. As mentioned in Section 3.3.2, the Enhanced Burgers solver uses the Burgers equation given by Equation (32). Since there is no known analytical solution, this differential equation is typically solved using a split-step approach [26, 91] similar to that used in the Legacy Burgers solver discussed in Section 4.3.1. The approach used in the Enhanced Burgers solver treats the nonlinear and absorption terms separately at each spatial step. The nonlinear effects are accounted for in the time domain while the absorption effects are accounted for in the frequency domain. (When the split-step approach is not employed, the Burgers equation is solved using a finite-difference numerical scheme.)

Using the split-step, pseudospectral approach, Equation (32) can be rewritten into two components. The advantage of this approach is that each component lends itself to an analytical solution. In the time domain, the first component is just a differential equation whose discretized solution in the spatial domain is

$$u_i(\tau) = u_{i-1}(\tau + u_{i-1}\Lambda_i) \tag{70}$$

where
$\quad\quad u_i =$ the waveform at step s_i
$\quad\quad \Lambda_i =$ the age parameter [22, 92, 93].

The second component of Equation (32) gives rise to a differential equation whose discretized solution can be expressed by

$$U_i(\omega) = U_{i-1}(\omega)e^{-a_i(\omega)}$$

(71)

where

$$U_i(\omega) = U(s_i, \omega)$$
$$a_i(\omega) = \text{an integrated absorption coefficient.}$$

Recall that $U_i(\omega)$ is the Fourier transform of $u_i(\tau)$. Both the age parameter and the integrated absorption coefficients are approximated to second order using

$$\Lambda_i = \frac{[\tilde{\beta}(s_i) + \tilde{\beta}(s_{i-1})]}{2}\delta s$$

(72)

and

$$a_i(\omega) = \frac{[\alpha_t(s_i, \omega) + \alpha_t(s_{i-1}, \omega)]}{2}\delta s$$

(73)

where

$$\delta s = s_i - s_{i-1}; \text{ the step size}$$
$$\tilde{\beta} = \text{effective coefficient of nonlinearity.}$$

4.3.2.2 Effective Coefficient of Nonlinearity, Blokhintzev Factor, and Absorption

The steepening and subsequent stretching of a waveform is dictated by the magnitude of the age parameter Λ, which is dependent on the effective coefficient of nonlinearity $\tilde{\beta}$. This coefficient is calculated using

$$\tilde{\beta} = \frac{\beta p_o^{(p)}B\Omega\chi}{\rho_o c_o^3}$$

(74)

where

$$\beta = 1.2, \text{ the coefficient of nonlinearity}$$
$$p_o^{(p)} = \text{the initial peak overpressure}$$
$$B = \text{the Blokhintzev scaling factor}$$
$$\Omega = c_o/c_n, \text{ the Doppler effect, where } c_n \text{ is the effective sound speed}$$
$$\chi = c_o/v, \text{ the convective index, where } v \text{ is the ray speed}$$
$$\rho_o = \text{the atmospheric density}$$
$$c_o = \text{the sound speed.}$$

Not only is the Blokhintzev scaling factor necessary in calculating the effective coefficient of nonlinearity, it is also required in calculating the acoustic pressure amplitude $p(s, \tau) = p_o^{(p)}B(s)u(s, \tau)$. It can be calculated using

$$B = \sqrt{\frac{\rho_o c_o \chi \Omega A_r}{\rho_{o,r} c_{o,r} \chi_r \Omega_r A}}$$

(75)

where

$$A = \text{the ray tube area}$$
$$\text{subscript } r = \text{represents values at the reference (initial) point along the ray path.}$$

Note that the parameters without the susbscript r all depend on the step index. In comparison, the Blokhintzev parameter $B^{(L)}$ used in the Legacy Burgers solution can be expressed as Equation (31) repeated here as

$$B^{(L)} = \sqrt{\frac{p_o c_o c_{n,r}^2 A_r}{p_{o,r} c_{o,r} c_n^2 A}}. \tag{76}$$

$B^{(L)}$ is identical to B only if the wind velocity is along or opposite the propagation direction.

The total absorption coefficients $\alpha_t(\omega)$ in Equation (73) can be calculated using

$$\alpha_t(s_i, \omega) = \chi \left(\Omega^2 \omega^2 A_{cr} + \sum_v \frac{A_{vi}^{(v)}(\Omega^2 \omega^2 \omega_v - i\Omega^3 \omega^3)}{\omega_v^2 + \Omega^2 \omega^2} \right) \tag{77}$$

where
A_{cr} = classical-rotational absorption function
$A_{vi}^{(v)}$ = vibrational-relaxation absorption function.

Both A_{cr} and $A_{vi}^{(v)}$ are dependent on the ambient temperature and pressure, and can be determined from Robinson [26] and Bass et al. [94]. The total absorption coefficient already accounts for the full wind effects. In contrast, the Legacy Burgers solver uses the absorption coefficients $\alpha(\omega)$ as shown in Equation (30). Such an absorption coefficient $\alpha(\omega)$ can be obtained from Equation (77) but with the convective index χ and Doppler shift Ω set to unity, ignoring the Doppler and convective effects of wind.

4.3.2.3 Development and Implementation of the Enhanced Burgers Algorithm
An algorithm that achieves an efficiency improvement over the Legacy Burgers solver by 2-3 orders of magnitude [6] has been implemented in FOBoom. The algorithm employs a combination of the analytical solutions, Equations (70) and (71), and a spatial step size that is a compromise between numerical accuracy and computational efficiency. This step size is generally larger than that in the Legacy Burgers solver.

In the Legacy Burgers solver and most other sonic boom propagation codes [26, 95], the step size used to advance along the ray path is taken to be proportional to the propagation distance just before a shock forms. Such a shock-formation distance δs is inversely proportional to the maximum slope of the waveform in the previous step along the ray path. In a discretized waveform, this shock-formation distance is

$$\delta s \sim \Delta\tau / (\Delta p)_{max}. \tag{78}$$

This shock formation distance can be converted into a propagation time by the relationship [91, 96] $dt = ds/v_r$ while marching is done along the ray path with the step size $\Delta t = \int_{s_o}^{s} ds/v_r$. Note that v_r here is just the local sound speed if there is no wind. The step size obtained based on Equation (78) becomes small as $(\Delta p)_{max}$ increases. This usually happens when the peak overpressure is large, such as for waveforms generated by conventional supersonic aircraft. The step size is further diminished by increasing the waveform sampling rate, i.e., decreasing the time interval $\Delta\tau$.

85

The step size in the Enhanced Burgers solver is not based on the relationship between the shock formation distance and the pressure slope. Instead, it is based on the integrated absorption coefficients $a_i(\omega)$ given by Equation (73). If the step size at a particular step is small such that the largest value of $a_i(\omega)$ is nearly zero, then $U_i(\omega)$ would be nearly identical to $U_{i-1}(\omega)$. Thus, that particular step would be unnecessary and can be skipped with minimal impact on the accuracy of the solution. Consequently, here the step size is chosen such that the largest value of $a_i(\omega)$ is sufficiently large while keeping the commutation of the nonlinear and absorption operators negligible. Not only does this step size increase the efficiency, it also suppresses the formation of Gibbs oscillations by ensuring that the atmospheric absorption effect is significant enough to attenuate the high frequency components.

In waveforms with strong shocks, the step size used in the Enhanced Burgers solver can be larger than the shock-formation distance. In this situation, the nonlinear solution given by Equation (71) becomes multivalued and is subsequently converted into a single-valued solution by using a procedure similar to the Middleton–Carlson method [30]. Such a procedure is based on the well-known Rankine–Hugoniot jump condition, which leads to the equal-area rule used to determine shock positions [22, 34, 91]. The additional computations needed to implement the equal area rule don't outweigh the improvement in computational efficiency gained by using the larger step size.

4.4 *Keyword HYSTATMO, OTT, OTTRANGE, ATMOS*: Atmospheric Handling

Input of atmospheric conditions to PCBoom can happen in several different formats. Following file-parsing operations, atmospheric data processing is done as described in this section. For any altitude at which atmospheric properties are internally requested by FOBoom, linear interpolations of pressure, temperature (via local speed of sound), x-wind and y-wind velocity, and relative humidity are performed between those conditions at the altitudes specified by the user. If the requested altitude is above the highest altitude in the atmospheric profile(s), the value of relevant atmospheric quantity is set to the value from the top of the atmospheric profile. If the requested altitude is below the lowest altitude in the atmospheric profile(s), the value of relevant atmospheric quantity is set to the value from the bottom of the atmospheric profile.

An extra "below ground" point is added to each of the atmospheric profiles to facilitate caustic curvature calculations when a focus occurs near the ground. In this context, ground level is taken as the lowest level in the atmospheric profile. Rather than extrapolating based on slopes of atmospheric conditions near the ground or standard lapse rates, the atmospheric pressure, temperature, x-wind and y-wind velocity, and relative humidity values are held constant from the lowest profile point to the below-ground point. The general below-ground point is located 10,000 ft below the supplied atmospheric profile. If keyword HYSTATMO is used, the below-ground point is the lowest of:
- −2,000 ft,
- distance below ground altitude at which raytracing stops, ZBELOW, and
- altitude below which foci are ignored, ZMIN.

If ray tracing over a spherical or ellipsoidal Earth model is specified, atmospheric properties are cast in the corresponding three-dimensional reference frame. To do so, height relative to local ground is determined and then atmospheric properties are determined following a layered atmosphere assumption. In addition, wind vectors are rotated in an Earth-centered, Earth-Fixed Greenwich (EFG) reference frame.

PCBoom internally forms atmospheric pressure profiles from the hydrostatic equation under one of two conditions: if an atmospheric pressure profile is not supplied in some form as user input, or if the keyword HYSTATMO is used. In those scenarios, the atmospheric pressure at ground level (p_{0g}) and the temperature profile are used to determine the atmospheric pressure profile using established methods (see, for example, Anderson [97]) as described below.

The differential form of the hydrostatic equation is

$$dp = -\rho g_o dh \tag{79}$$

where

g_o = sea-level gravity
h = geopotential altitude.

Using the ideal gas assumption and dividing Equation (79) by that equation of state ($p = \rho RT$) results in

$$\frac{dp}{p} = -\frac{\rho g_0 dh}{\rho RT} = -\frac{g_0}{R}\frac{dh}{T} \tag{80}$$

where

R = specific gas constant for dry air
T = absolute temperature.

Integrating both sides of Equation (80) from a base height h_i to arbitrary height h results in the general expression

$$p(h) = p_i \exp\left(-\frac{g_0}{R}\int_{h_i}^{h}\frac{dh}{T(h)}\right) \tag{81}$$

for atmospheric pressure p at h following a temperature profile $T(h)$. Further simplifications can be made to Equation (81) if assumptions are made about the form of $T(h)$, e.g., isothermal or gradient layers [97]. PCBoom uses supplied temperature profiles to evaluate Equation (81) and does not assume a functional form for $T(h)$. Rather, the integral is evaluated numerically using a trapezoidal approximation. In that numerical evaluation, dh is approximated as a finite height difference: $\Delta h = $ 1,000 ft generally and $\Delta h = $ 100 ft when the HYSTATMO keyword is used. The temperature profile is linearly interpolated onto a set of altitudes determined by Δh prior to numerical integration. The first segment of the integration uses supplied base pressure p_{0g} as p_i and subsequent segments build on integrated pressure from the previous segment. As a final step, atmospheric pressure at the below-ground point is linearly extrapolated from the integrated pressure profile.

A special atmospheric handling step is done in OTT mode. To avoid inconsistencies associated with iterative procedures at the ground, atmospheric temperature and wind profiles are held constant near the ground. In this context, the distance over which properties are held constant is variable, and is determined as

$$z_{break} = a_{ground}t_{step} \tag{82}$$

where

a_{ground} = speed of sound at ground level

$$t_{step} = \text{ time step specified for integration along a ray}$$
$$z_{break} = \text{ vertical distance.}$$

The vertical distance z_{break} will be sufficiently large for any ray, including a completely vertical ray, to propagate a complete time step toward the ground through constant atmospheric conditions.

The optional keyword OTTRANGE specifies a range of trajectory times to control the points written by PCBoom to OTT output files. By default, PCBoom sets variables OTTMIN and OTTMAX to $\pm 10^{35}$ seconds to effectively allow all trajectory times through. Those internal values are replaced with user-specified values if provided.

4.5 *Keyword MODE/CARLSON*: N-wave *F*-function Source Description

A simplified method for the calculation of sonic boom characteristics for preliminary engineering studies or the preparation of environmental impact statements was developed by Carlson [9]. This method is employed by use of the MODE and CARLSON keywords in the PCBoom input file. The MODE and CARLSON keywords result in the internal computation of a simple N-wave *F*-function for each flight condition and azimuth angle.

There are five options to prescribe the *F*-function in FOBoom based on Carlson's method, each described in the following subsections. Carlson established a process by which a shape factor may be utilized for the *F*-function based on operational and state parameters.

The shape factor is a table look-up and linear interpolation based on the shape factor formulae as shown in Figure 4-6 where M = Mach number, W = vehicle weight (kg, PCBoom converts input file klb to kg), γ (flight path angle, degrees), θ (azimuth angle, degrees), p_v (atmospheric pressure at aircraft altitude, Pa) and l (aircraft length, meters – PCBoom converts input file value from ft to meters). The horizontal axis parameter is the Lift factor (K_L) while the vertical axis is the shape factor K_S. The shape factor K_S is then converted into the starting *F*-function (see Section 4.2.2). The PCBoom shape factor is limited to the domain shown in Figure 4-6 (except for the shuttle formula), it will not extrapolate K_L below 0.0 or above 0.05.

4.5.1 CARLSON / ACNAME

A number of existing aircraft parameters have been preset in PCBoom (see the PCBoom User's Guide [8]). When using the CARLSON/ACNAME keywords, the indication of a preset aircraft name in the FOBoom input file references an internal data table that is prepopulated with the following information:
- Carlson Shape Factor Curve
- Weight
- Length
- Thrust (only applicable to the Titan launch vehicle)
- Drag (only applicable to the Titan launch vehicle)

The starting signature is prepared via Carlson's simplified model and scaled to local conditions. The shape factor curves are provided in Table 4-2. Parameters are implemented as a data statement within FOBoom and interpolated linearly, without extrapolation.

The upper chart plots K_S (vertical axis, from .04 to .20) against K_L (horizontal axis, 0 to .05), with lines labeled:

- Large Fighter — F-101
- Small Fighter — F-104
- Medium bomber — B-58, SR-71
- Large bomber — B-70

$$K_L = \frac{\sqrt{M^2 - 1} \; W \; cos \; \gamma \; cos \; \theta}{1.4 p_v M^2 l^2}$$

$$K_S \approx 0.74\sqrt{K_L + 0.027}$$

The lower chart plots K_S (vertical axis, from .04 to .20) against K_L (horizontal axis, 0 to .05), with lines labeled:

- Shuttle Orbiter
- Fixed-wing fighters — F-15, F-16
- Variable-sweep airplanes — B-1, F-111, F-14
- Concorde

$$K_L = \frac{\sqrt{M^2 - 1} \; W \; cos \; \gamma \; cos \; \theta}{1.4 p_v M^2 l^2}$$

**Figure 4-6: Carlson Representative Aircraft Shape Factors (K$_S$)
as a function of Lift Parameter (K$_L$) (*Source: NASA TP 1122. Carlson* [9]).**

Table 4-2 Shape Factors (K$_S$) for various Shape Factor (IUN) Curves in FOBoom
(*Data Source: NASA TP 1122. Carlson* [9]).

IUN	Lift Factor (K$_L$)					
	0	.01	.02	.03	.04	.05
1 Large Fighter	.080	.100	.120	.141	.160	.181
2 Small Fighter	.064	.087	.116	.128	.139	.179
3 Medium Bomber	.070	.088	.107	.130	.153	.174
4 Large Bomber	.056	.075	.101	.125	.149	.171
5 Fixed Wing Fighters	.078	.099	.122	.144	.165	.184
6 Variable Sweep Airplanes	.080	.099	.117	.134	.152	.169
7 Concorde	.050	.079	.108	.131	.151	.168
8 Shuttle Orbiter	.140	.158	.172	.184	.196	.205

4.5.2 CARLSON / AXISYMMETRIC

CARLSON/AXISYMMETRIC keywords read the shape factor (KS) from the TRAJECTORY section of the input file under the NEWLOAD sub-keyword. This results in a 4-point *F*-function and N-wave vehicle. If the optional vehicle thrust or drag options are specified, a 6-point *F*-function and corresponding starting pressure signature are developed. The plume thrust is determined as described by the Tiegerman's method [19] (Section 4.5.4).

4.5.3 CARLSON / BASIC

CARLSON/BASIC keywords allow the user to prescribe an aircraft length and weight while utilizing an existing Carlson shape curve. The available shape curves are provided in Figure 4-6, where M = Mach number, W = weight (kg), γ = flight path angle (deg), Θ = ray path angle (deg), p_v is the atmospheric pressure at the vehicle altitude (Pa) and l is the aircraft characteristic length, normally the fuselage length (m).

4.5.4 CARLSON / LAUNCH

CARLSON/LAUNCH keywords exercise Tiegerman's method [19] for determining the 4-point *F*-function. The user may also change drag at trajectory points by prescribing it in the TRAJECTORY keyword section of FOBoom input. The positive pressure pulse in the symmetric four-point N-wave starting signature is defined by

$$dp/p = 0.6079 \frac{\left(\frac{D}{2\pi p_o}\right)^{3/8}}{r_o^{3/4}} \tag{83}$$

where

$$r_o = \text{starting radius in feet}$$
$$D = \text{plume drag in klbs}$$
$$p_o = \text{ambient pressure at the flight altitude (psf).}$$

The signature length is computed as

$$X = 1.042 \times M \frac{\left(\frac{D}{2\pi p_o}\right)^{3/8}}{r_o{}^{1/4}} \tag{84}$$

where

$M =$ flight Mach number.

4.5.5 CARLSON / LOOKUP

The CARLSON / LOOKUP feature allows a user to define their own shape factor curve in an external file. The user's defined input is handled similarly to the preprogrammed Carlson shape factor curves shown in Figure 4-6 and Table 4-2.

4.6 *Keyword TRAJECTORY:* Trajectory Kinematics

The aircraft motion along the prescribed trajectory is advanced in time using the TADVNCE (and NEWDERS, if applicable) keyword(s) based on the following prescribed aircraft state parameters:
- Position (for information on coordinate systems, see Section 4.2.1),
- Mach number including first and second derivatives,
- Heading angle including first and second derivatives, and
- Flight path angle including first and second derivatives.

Each new trajectory point is computed based on a third order Taylor expansion. The new Mach number, heading, flight path angle, and their first derivatives at the new point are computed using Taylor expansions through the second derivatives.

The TRAJECTORY keyword TADVNCE projects the current trajectory forward using the current values of the first and second derivatives. The NEWDERS keyword reads and uses the updated first and second derivatives for the subsequent trajectory point computations.

4.7 *Keyword RAYTRACING:* Fundamental Ray Tracing Physics

The PCBoom module, FOBoom, conducts ray tracing and signature evolution. The ray tracing outputs are utilized by the Enhanced Burgers and Legacy solvers. The specifics of the implementation in PCBoom are described below.

4.7.1 Ray Tracing

4.7.1.1 Implementing Equations

There are different ray tracing modes available in PCBoom. These are the LEGACY, SCHULFLAT, SCHULELLIPSE, and SCHULSPHERE modes. The LEGACY mode refers to the Thomas ray tracing algorithm [1] developed in the 1970s. This algorithm assumes that the Earth is flat and that the atmospheric properties are stratified. While the LEGACY mode is still functional or backward compatible in the current version of FOBoom, the underlying algorithm used is that of the SCHULFLAT ray tracing mode.

Unlike the LEGACY mode, the Schulten ray tracing modes require derivatives of atmospheric conditions in three dimensions relative to the EFG coordinate systems used. The single atmosphere profile input to PCBoom is used to construct a 3D shell atmosphere with the single profile normal to the local surface. As the name suggests, the SCHULFLAT mode [5] approximates the Earth as flat. It can be used in sonic boom propagation involving primary booms, in which the horizontal distance traveled is small, resulting in negligible effects of the Earth's surface curvature. The SCHULSPHERE and the SCHULELLIPSE modes [5]

are used for long-range sonic boom propagation, such as those involving secondary booms, in which acoustic rays propagate upward and refract back down from the stratosphere or thermosphere. In these situations, propagation distances are large enough that the effects of the Earth's curvature may no longer be negligible.

Regardless of the ray tracing modes, there is only one set of ray tracing equations used in PCBoom to determine the ray paths. Here, the ray paths are indicated by the three Cartesian components x_α of the ray position vector $\vec{x}$, where $\alpha = 1, 2, 3$ correspond to the x, y and z coordinates, respectively. In this section, any Greek letter used as a subscript or index of a quantity indicates a vector quantity, and terms with a repeated Greek index are summed over the values of that index (from 1 to 3). In range-dependent, windy atmospheres, the ray tracing equations are the coupled differential equations [6, 22, 34]

$$\frac{dx_\alpha}{dt} = c_o \frac{q_\alpha}{\|\vec{q}\|} + w_\alpha \tag{85}$$

and

$$\frac{dq_\alpha}{dt} = -\|\vec{q}\| \frac{\partial c_o}{\partial x_\alpha} - q_\gamma \frac{\partial w_\gamma}{\partial x_\alpha} \tag{86}$$

where

$c_o =$ the sound speed
$q_\alpha =$ the slowness vector $\vec{q}$, determined by Equation (86)
$w_\alpha =$ the Cartesian components of the wind velocity vector $\vec{w}$.

Equations (85) and (86) are similar in form to the 2D relations used by Thomas [1], and are a natural choice for the 3D extension, in which the sound speed and wind velocity vary in all three dimensions. These equations are implemented using the same numerical integration algorithm [5] as in the LEGACY Thomas code, but with added complexities due to the 3D gradients of the wind velocity and sound speed.

In SCHULELLIPSE or SCHULSPHERE mode, the cartesian coordinates are converted to the coordinates suitable for the ellipsoidal or spherical model of the Earth (see Section 4.2.1). In the spherical model, the coordinates used are (r, λ, θ) where r is the altitude from a local surface of the Earth, λ is the latitude, and θ is the longitude. The Earth's atmosphere is set up as a simple shell with a single profile that is normal to the local surface. A grid of profiles, available from 4D weather forecasting systems, is easily extended to a profile lookup table for (r, λ, θ) for a full 3D spherical shell atmosphere. From this lookup table, the full 3D gradient components needed in Equation (86) are computed. In SCHULELLIPSE mode, the Earth is treated as ellipsoidal accounting for its minor and major axes in determining the altitude. The determination of the atmospheric profiles is done using the same method as that used in the SCHULELLIPSE mode.

A special scheme is needed for reflection of rays at the ground in either the SCHULELLIPSE or SCHULSPHERE mode. Such a scheme ensures the coordination of the origins and timing of the four rays in a ray tube. It also ensures that these rays remain in phase while having different step sizes at their reflection points. Lateral cutoff limits in the horizontally stratified model are easy to obtain using admittance ellipses similar to the process used in TRAPS [27]. However, this process is no longer possible in the 3D formulation because a 3D atmosphere does not have the spatial homogeneity of a traditional 2D layered atmosphere. Here, admittance ellipses are used for general guidance only, as it is necessary to trace sample rays at all azimuths to establish which rays would reach the ground as either primary or secondary booms.

4.7.1.2 Numerical Implementation

In an arbitrary atmosphere, the coupled differential equations, Equation (85) and Equation (86), cannot be solved analytically. Consequently, the ray paths are determined numerically by approximating these equations to first-order Taylor expansions as

$$x_\alpha(t_{i+1}) \approx x_\alpha(t_i) + \Delta t \left(c_o(t_i) \frac{q_\alpha(t_i)}{\|\vec{q}(t_i)\|} + w_\alpha(t_i) \right) \tag{87}$$

and

$$q_\alpha(t_{i+1}) \approx q_\alpha(t_i) - \Delta t \left(\|\vec{q}(t_i)\| \frac{\partial c_o}{\partial x_\alpha}(t_i) + q_\gamma(t_i) \frac{\partial w_\gamma}{\partial x_\alpha}(t_i) \right) \tag{88}$$

where

$\Delta t = t_{i+1} - t_i$, the step size in the time coordinate.

As these equations show, the determination of $x_\alpha = x_\alpha(t)$ requires the determination of $q_\alpha = q_\alpha(t)$. The ratio $q_\alpha / \|\vec{q}\|$ gives the unit vector n_α normal to the wavefront. The initial slowness vector is approximated by

$$q_\alpha = \frac{n_\alpha}{c_o} \tag{89}$$

in which the wind effects are neglected. Since the magnitude of the unit vector is unity, the magnitude $\|\vec{q}\|$ of the initial slowness vector is then equal to the reciprocal of the sound speed. In general, $\|\vec{q}\|$ is obtained using

$$\|q\| = \sqrt{q_\alpha q_\alpha}. \tag{90}$$

4.7.2 Ray Tube Area Calculation

Ray tube area is a quantity required in solving either the lossless or lossy Burgers equation to compute sonic boom waveforms [6, 91]. It quantifies the geometrical spreading of acoustic rays as they propagate away from the source location. In typical sonic booms, the ray tube area increases with increasing acoustic propagation distance, accompanied by a reduction in the acoustic pressure. It must be computed as it largely determines the acoustic pressure.
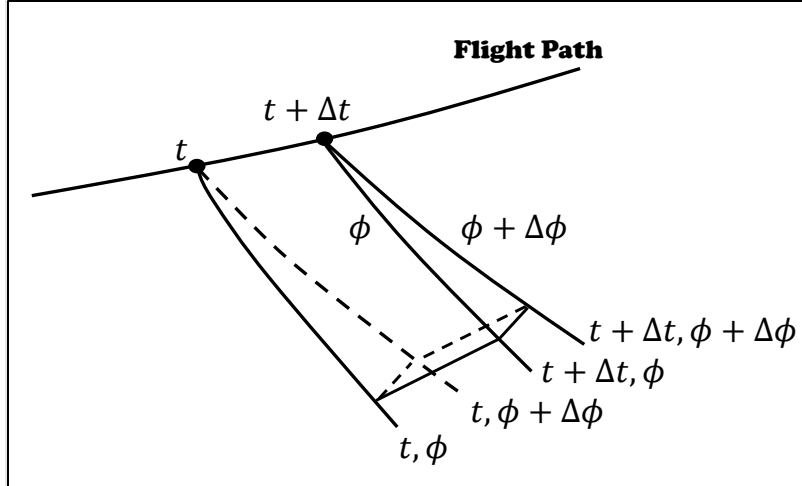
Figure 4-7: Ray Tube Outlined by Four Corner Rays Δt and $\Delta \phi$ apart (*Source: Plotkin* [12]*; reprinted with permission*).

The ray tube area is calculated using an abstract ray bundle consisting of four rays as shown in Figure 4-7. It is the cross-sectional area subtended by the four rays and whose unit vector normal to its plane is parallel to the propagation direction. The primary ray is determined by the pair of t and ϕ parameters, where t is the time at which the ray was emitted and ϕ is the azimuthal angle of the ray around the aircraft. The second ray is generated at the same instant of time but with a slightly different azimuthal angle, $\phi + \Delta \phi$. The third and fourth rays are generated by repeating the procedure at a slightly different time, $t + \Delta t$. The initialization of the rays accounts for the bank angle of the aircraft.

The ray paths corresponding to the four rays discussed in the previous paragraph are calculated using Equations (85) and (86). These can be denoted by $x_\alpha^{(j)}$ where the superscript $j = 1, 2, 3, 4$ identifies a ray from the set of four rays. For instance, the ray path for the primary ray is denoted by $x_\alpha^{(1)}$, the secondary ray by $x_\alpha^{(2)}$, etc. The ray tube area is computed from the Jacobian [96] of the transformation from the ray coordinates to the Cartesian coordinates for a single ray. The most general expression of the Jacobian is given by

$$J = \left\| \frac{\partial x_\alpha}{\partial \phi} \times \frac{\partial x_\alpha}{\partial t_a} \right\| \cdot \frac{\partial x_\alpha}{\partial s} \tag{91}$$

where

$$s = \text{ray path length}$$
$$\text{symbol} \times \text{denotes the cross product operator}$$
$$\text{symbol} \cdot \text{denotes the dot product operator.}$$

Equation (91) is adapted in PCBoom using the four rays, as is done by most sonic boom propagation codes, and the Jacobian is aptly called the ray tube area. The partial derivatives in Equation (91) are approximated by the finite differences

$$\frac{\partial x_\alpha}{\partial \phi} \approx \frac{\triangle x_\alpha}{\triangle \phi}, \text{ at constant } t \text{ and } t_a$$

$$\approx \frac{x_\alpha^{(1)}(t_i) - x_\alpha^{(2)}(t_i)}{\triangle \phi}, \tag{92}$$

$$\frac{\partial x_\alpha}{\partial t_a} \approx \frac{\triangle x_\alpha}{\triangle t_a}, \text{ at constant } t \text{ and } \phi$$

$$\approx \frac{x_\alpha^{(1)}(t_i) - x_\alpha^{(3)}(t_i)}{\triangle t_a}, \tag{93}$$

and

$$\frac{\partial x_\alpha}{\partial t} \approx \frac{\triangle x_\alpha}{\triangle t}, \text{ at constant } \phi \text{ and } t_a$$

$$\approx \frac{x_\alpha^{(1)}(t_{i+1}) - x_\alpha^{(1)}(t_i)}{\triangle t}. \tag{94}$$

Note that $x_\alpha^{(4)}$ are not used in these equations as the ray tube area determination only requires three rays. The ray tube area is overdetermined by the use of the fourth ray. To make use of $x_\alpha^{(4)}$ in PCBoom, $x_\alpha^{(1)}$ in Equation (94) is replaced by $x_\alpha^{(4)}$, without significantly changing the value of the ray tube area if the rays are close enough from each other.

The ray tube area A at each time step t_i is then computed using

$$A = \frac{1}{N} \begin{vmatrix} \dfrac{\partial x_1}{\partial t} & \dfrac{\partial x_2}{\partial t} & \dfrac{\partial x_3}{\partial t} \\[2mm] \dfrac{\partial x_1}{\partial \phi} & \dfrac{\partial x_2}{\partial \phi} & \dfrac{\partial x_3}{\partial \phi} \\[2mm] \dfrac{\partial x_1}{\partial t_a} & \dfrac{\partial x_2}{\partial t_a} & \dfrac{\partial x_3}{\partial t_a} \end{vmatrix} \tag{95}$$

where

$$N = n_1(\partial x_1/\partial t) + n_2(\partial x_2/\partial t) + n_3(\partial x_3/\partial t)$$
$$n_\alpha = \text{the components of the unit vector along the propagation direction}$$
$$[\,] = \text{matrix determinant.}$$

Note that the partial derivatives in Equation (95) are approximated by those in Equations (92), (93), and (94). The use of N, which has a unit of speed, arises from the use of $\partial x_\alpha/\partial t$ in Equation (95), rather than $\partial x_\alpha/\partial s$ required for the Jacobian in Equation (91).

4.7.3 Lateral Cutoff

As described in Section 3.4.1, lateral cutoff occurs where rays refract away from the ground before reaching it. Lateral cutoff location is a function of aircraft trajectory, atmospheric conditions, and local ground elevation. PCBoom ray tracing inputs can include either a discrete list of azimuth angles (a phi list) to model, or an azimuthal angle increment. In the course of propagation modeling using a ϕ list, PCBoom can detect when rays refract away from the ground and not output ground boom results at those azimuth angles. When a PCBoom input file specifies a ϕ increment and NPHIS = 0 (i.e., no ϕ list), an internal algorithm is used to determine the lateral cutoff angles and then construct a list of ϕ values between the left and right lateral cutoff angles. PCBoom then models propagation from each azimuth angle in that list to construct a carpet.

PCBoom begins by calculating the cutoff angle for a nominal no-wind atmosphere at the ground elevation specified in the PCBoom input file. That cutoff angle, ϕ_{cut}, is found as

$$\phi_{cut} = \cos^{-1}\left(\frac{\cos\delta_{cut} + \sin\mu\sin\gamma}{\cos\mu\cos\gamma}\right) \tag{96}$$

where

$\delta_{cut} = $ the vertical angle of the cutoff ray found from Equation (97)
$\mu = $ the Mach angle found from Equation (98)
$\gamma = $ the flight path angle from the aircraft trajectory.

The vertical angle of the cutoff ray and the Mach angle are determined, respectively, using

$$\delta_{cut} = \sin^{-1}\left(\frac{a_{ground}}{a_0}\right) \tag{97}$$

and

$$\mu = \sin^{-1}\left(\frac{1}{M}\right) \tag{98}$$

where

$a_0 = $ the speed of sound at the aircraft.

The no-wind cutoff angle is used as input to an iterative routine to find the lateral cutoff angle. Details of the method used depend on the ray tracing mode specified in the PCBoom input file and are outlined in the following subsections.

4.7.3.1 Lateral Cutoff Determination – Legacy Ray Tracing Modes

If LEGACY ray tracing mode is specified in the PCBoom input file, the lateral cutoff angle algorithm begins by calculating the wave vector at the aircraft using actual winds specified with the atmospheric input. The wave vector $\vec{w}$ are calculated as

$$\vec{w} = \vec{n}\left(\frac{a_0}{a_0 + V_{0x}n_1 + V_{0y}n_2}\right) \tag{99}$$

where

the subscript $0 = $ conditions at the aircraft
$V_{0x} = $ x-component of wind at the aircraft
$V_{0y} = $ y-component of wind at the aircraft.

The components of the wave normal unit vector are given by

$$n_1 = \frac{M_x}{M}\sin\mu - \frac{M_y}{M_h}\cos\mu\sin\phi + \frac{M_xM_z}{M_hM}\cos\mu\cos\phi, \tag{100}$$

$$n_2 = \frac{M_y}{M}\sin\mu + \frac{M_x}{M_h}\cos\mu\sin\phi + \frac{M_yM_z}{M_hM}\cos\mu\cos\phi, \tag{101}$$

and

$$n_3 = \frac{M_z}{M}\sin\mu - \frac{M_h}{M}\cos\mu\cos\phi. \tag{102}$$

The components of the three-dimensional Mach vector (M_x, M_y, M_z), and the projection M_h of the Mach vector onto a horizontal plane are given, respectively, by

$$M_x = M \cos \gamma \sin \psi, \tag{103}$$

$$M_y = M \cos \gamma \cos \psi, \tag{104}$$

$$M_z = M \sin \gamma, \tag{105}$$

and

$$M_h = \frac{M_x^2 + M_y^2}{\sqrt{M_x^2 + M_y^2}} \tag{106}$$

where

$\gamma =$ the flight path angle
$\psi =$ the aircraft heading.

The wave vector is then used to check whether Snell's law allows for existence of rays at altitudes below the flight altitude using

$$\left(a_0 - V_x w_x - V_y w_y\right)^2 - a^2\left(w_x^2 + w_y^2\right) < 0 \tag{107}$$

where

$a =$ speed of sound at intermediate altitudes between flight altitude and ground level.

The azimuth angle is incremented from the starting angle found from Equation (96) in steps of 1°. The incremental search direction is set by first checking whether the Snell's law condition for the ray at the initial azimuth angle is met at some altitude above the ground. Once that condition is found, the cutoff angle search is backed off by 1°, and then continued in steps of 0.1°. Similarly, once the cutoff condition is found again using 0.1° steps, the cutoff angle search is backed off by 0.1°, and then continued in steps of 0.01° to further refine the lateral cutoff angle. As a final step, an "edge" buffer of 0.5° is subtracted to produce the final lateral cutoff angle, the implications of which are described in the following paragraph. The process is executed for both left and right lateral cutoff angles.

Propagation modeling of rays at extreme azimuth angles has shown the corresponding sensitivity of ground intersection locations and waveform metrics [98, 99] for different ray tracing codes. Depending on atmospheric conditions, azimuth angle increments as small as 0.1° can result in significantly different waveform metrics due to sensitivity of propagation path length and ray tube area when the rays are nearly horizontal in the vicinity of lateral cutoff. For a particular set of ray tracing parameters in FOBoom, one can find the lateral cutoff angle by manually incrementing a user-specified list of azimuth angles and examining FOBoom output files to find the highest azimuth angle for which FOBoom returns a ground signature. Doing so produces lateral cutoff angles around 0.5° higher than those found using the internal algorithm due to its edge buffer. However, results from such high azimuth angles must be evaluated critically as increased sensitivity and finite numerical precision can lead to unrealistic ray paths or ground signatures [98]. When modeling full carpets using lateral cutoff angles found by FOBoom, the edge buffer helps to protect against that type of result, especially due to the finite ray tube

width (DPHI, default value 0.5°) used by FOBoom. Users should be aware that inclusion by PCBoom of the edge buffer may contribute to discrepancies between measured and modeled lateral cutoff locations, and that that carpet width is affected by the buffer more for wider carpets than narrow carpets as illustrated in Figure 4-8.
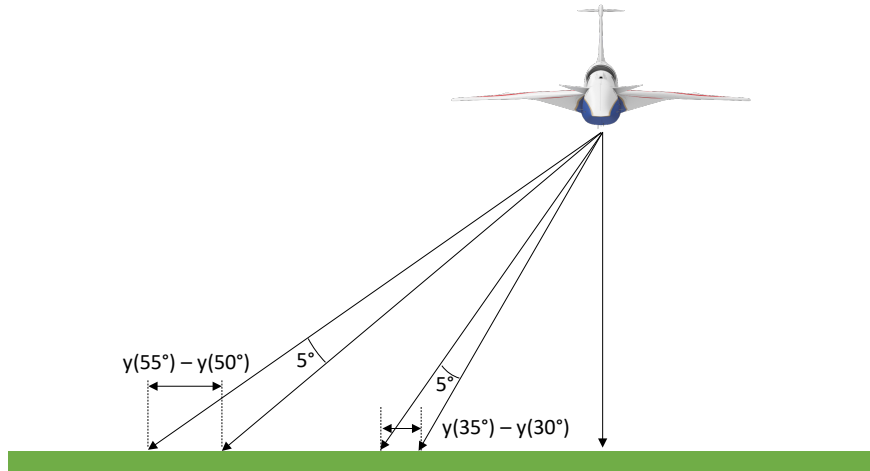
Figure 4-8: Differences in Lateral Ground Intersection Points for Pairs of Rays Separated by 5°.

4.7.3.2 Lateral Cutoff Determination – Schulten Ray Tracing Modes

When the PCBoom input file specifies a Schulten ray tracing mode, a search algorithm similar to the LEGACY ray tracing mode (Section 4.7.3.1) is used to find the lateral cutoff angles. However, rather than using wave vectors and Snell's Law to evaluate whether a particular ray is above or below lateral cutoff, the specified Schulten mode is used to execute ray tracing at candidate lateral cutoff angles. The minimum height reached is checked against the ground elevation to determine if the azimuth angle should be incremented. Using that condition, the search in steps of 1°, 0.1°, and 0.01° is done in the same manner as described for non-Schulten ray tracing modes. Similarly, a 0.5° "edge" buffer is included in the final lateral cutoff angle; see preceding section for additional details.

4.7.3.3 Lateral Cutoff Determination with Terrain Files

Lateral cutoff rays are handled differently when nonzero ground elevation is specified via terrain file versus using the ALTITUDES keyword. When the internal FOBoom algorithm is used to find the lateral cutoff angle by setting NPHIS = 0, the ground elevation used to do so is the bottom altitude in the ALTITUDES keyword section, regardless of whether a terrain file is also specified. If a terrain file is specified using the TERRAIN keyword, an output altitude of 0.0 signals FOBoom to use the local ground altitude [8]. This also causes FOBoom to find the lateral cutoff angle using a ground elevation of 0.0 ft above mean sea level (MSL), and then apply that lateral cutoff angle when ray tracing to local ground elevations from the terrain file.

An example of this is illustrated in Figure 4-9, in which a synthetic terrain file was generated with uniform ground elevation 2,400 ft above MSL. When PCBoom is run with the terrain file, the lateral cutoff angle found by the internal algorithm corresponds to the ray which describes lateral cutoff at sea level. For a ground elevation of 2,400 ft, rays at slightly higher azimuth angles will intersect the ground compared with the scenario where ground elevation is at sea level. In applying the sea-level lateral cutoff angle to locations with terrain above sea level, the effect is typically to underestimate the carpet

width. For the example in Figure 4-9 (standard atmosphere, steady level flight at 35,000 ft and M = 1.4), the lateral cutoff location is underpredicted by approximately 10,000 ft when a flat terrain file specifying ground elevation 2,400 ft is used compared with specifying that same ground elevation using the ALTITUDES keyword. This is a potential area of improvement for a future version of PCBoom; when using terrain files with PCBoom 7.1, users should be aware of the implication for footprints modeled by setting NPHIS = 0.
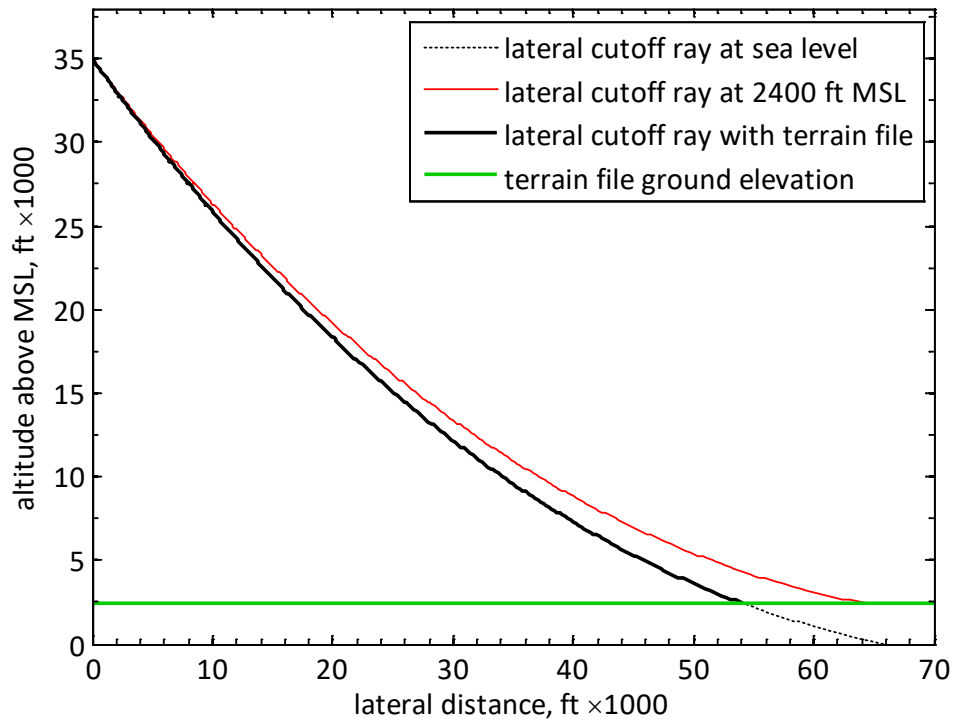
Figure 4-9: Ray Paths at Lateral Cutoff as Determined by FOBoom Internal Algorithm.

5   Additional Calculation and Visualization Options

5.1   PCBurg

The PCBurg module is an implementation of propagation that includes nonlinear steepening and absorption from molecular relaxation based on a lossy Burgers equation, which more accurately accounts for how the atmosphere affects the boom signature. Molecular relaxation is particularly important when analyzing loudness metric results due to the effect of shock structures on the upper frequency content of the booms. The PCBurg module contains a GUI to view the signature as it evolves from the flight altitude to the ground. It is separate from and slower than the Enhanced Burgers propagation that can be invoked in the FOBoom input file.

Computation of Pmax, ASEL, BSEL, CSEL, DSEL, ESEL, ZSEL, and PL metrics is available in PCBurg. Note that before computing the metrics, waveforms are zero padded to the next power of 2 in array length if the number of points is not a power of 2. This is to conform to the signal processing requirements of the FFT algorithms. Also, note that no window is applied to force the ends of the signature to zero, nor is the mean value subtracted from the waveform.

5.1.1 Summary of Inputs

The program reads FOBoom output files (.age and .ssg), which contain the ray paths, meteorological parameters, and the starting signature *F*-function derived from the vehicle source definition (Section 3.1). PCBurg can also read in caustic information such as the ray elevation angle and diffraction boundary-layer thickness from a file generated by the RayCau module (.foc).

5.1.2 Calculations

The Legacy Burgers solver numerically solves a differential equation similar to that given by Equation (30). The solution to Equation (30) must be scaled for three reasons: 1) the equation does not account for geometrical spreading, 2) it does not include altitude-dependent atmospheric density and sound speed, and 3) it does not account for the presence of winds. To obtain an accurate solution, the Legacy Burgers tools solve Equation (30) using a pseudospectral scheme described by Anderson [100] at every step along a ray path using a propagation time step Δt. In this scheme, the nonlinear steepening effect is accounted for in the time domain while the absorption effect is taken into consideration in the frequency domain. The solution is then scaled to account for the position-dependent amplitude caused by the geometrical spreading, atmospheric density, sound speed, and wind velocity.

5.1.2.1 Numerical Solution Methodology

Equation (30) can be approximated by the first-order time difference form as

$$\Delta p = -\Delta t \frac{\gamma + 1}{2\gamma} \frac{c_o}{p_o} p \frac{\partial p}{\partial \xi} + \Delta t [\alpha(\omega) p^T]^{-T} \tag{108}$$

where

$\Delta t =$ a finite propagation time increment.

Using Equation (108) and denoting the *i*th step with subscript *i*, $p_{i+1} = p_i + \Delta p$ and $t_{i+1} = t_i + \Delta t$ where p_i is the acoustic pressure at t_i and p_{i+1} is the acoustic pressure at time t_{i+1}. In the widely used Anderson algorithm [100] the mixed time-frequency domain process is generally described as alternating time and frequency steps, with the assumption that the terms can be solved independently

over sufficiently small steps. However, use of time domain for one term and frequency domain for the other is just a matter of technique, and does not imply any kind of alternation.

As noted earlier, $p(\xi, t)$ contains t twice: once as part of the wave coordinate $\xi = x - c_o t$ and once as the time dependence t along the ray path. At various points within the legacy Burgers solver, the boom waveform is represented as a function of distance along a ray (corresponding to ξ) and sometimes as a function of time past a point along a ray. For propagation and evolution of the waveform, the primary perspective is that of time. It is therefore convenient to view the dependence of p on ξ as a function of t rather than of x. Treating p as a function of t, the absorption term poses no problems. One simply takes the Fourier transform of p as a function of t and directly applies α, which is a function of frequency. The nonlinear term is slightly more complex. Note that at a fixed ray path length , $d\xi = -c_o dt$. Thus, we can use the transformation $\partial p / \partial \xi$ to $- (1/c_o)(\partial p / \partial t)_x$, from which we recognize that the divisor $-c_o$ occurs. If p is written as $p(t)$, the minus sign is reversed. Thus, Equation (108) can be rewritten as

$$\Delta p = -\Delta t \frac{\gamma + 1}{2\gamma} \frac{p}{p_o} \left(\frac{\partial p}{\partial t} \right)_x + \Delta t [\alpha(\omega) p^T]^{-T}. \tag{109}$$

A consideration is that Equation (30) is written for propagation of a plane wave in a uniform atmosphere. This is generalized to a locally plane wave (geometrical acoustics approximation) [21] if the position-dependent amplitude is accounted for. In the theory upon which PCBurg and HeadlessBurgers are based, the linear acoustic solution is

$$\frac{p}{p_o} = \frac{F(\tau)}{\sqrt{B}} \tag{110}$$

where

$\quad F(\tau) =$ the F-function
$\quad\quad \tau = t - \int_0^s c_o ds'$
$\quad\quad\quad \tau$ is a time-oriented equivalent of ξ, and is essentially the variable transform used when changing perspective from $p(\xi)$ to $p(t)$
$\quad\quad B =$ Factor incorporating the effects of ray tube area and acoustic impedance, and the scaling factors in Equation (1).

Note that in Equation (110), which is a rearrangement of Equation (15), the amplitude dependence along the ray is governed by $p_o / \sqrt{B}$. Within FOBoom, that combined factor is represented by the variable "Blok", which is labeled as "Ampl" in .age files. The amplitude dependence along a ray is then handled by a multiplication of Blok at each step. The Blok parameter at step i combines the local ambient pressure $p_{o,i}$ and the leading factor in Equation (1) with either factor B or factor $B^{(L)}$ as defined in Equation (31). Expressing Blok in terms of $B^{(L)}$ requires a rearrangement of reference and local ambient pressures that eventually yields

$$\text{Blok}_i = \frac{\gamma M^2}{\sqrt{2(M^2 - 1)r}} p_{0,i} \sqrt{\frac{c_{n,r}^2 \, p_{0,r} c_{0,i} A_r}{c_{n,i}^2 \, p_{0,i} c_{0,r} A_i}} = \frac{\gamma M^2}{\sqrt{2(M^2 - 1)r}} p_{0,r} B^{(L)}. \tag{111}$$

By incorporating the Blok multiplier, Equation (109) has been implemented in PCBurg and HeadlessBurgers. There are three important considerations for implementation: two are mathematical-numerical and one is practical. The first mathematical consideration is that the time step Δt in

Equation (109) must be small enough that the two terms on the right-hand side are small. Note that this requirement is not as strict for the second term as for the first. If only the second term were present, the solution for p at a given frequency is an exponential decay. The difference expression for the second term can therefore be improved by replacing $\alpha(\omega)\Delta t$ with $\exp(\alpha(\omega)\Delta t) - 1$.

The second mathematical consideration is that, at locations near the aircraft, the amplitude is high, and nonlinear steepening is strong. As a result, a very small Δt would be needed to advance the propagation to avoid multivalued solutions. It may not be necessary to begin accounting for the absorption at the starting point, as nonlinear effects dominate over absorption effects. The code is thus organized such that the propagation is solved using the thin shock theory near the aircraft, and the Legacy Burgers solver is used starting a user-specified distance away from the aircraft.

The practical consideration is that numeric processing of the Burgers equation in the Legacy Burgers solver is comparatively computationally intensive. Thin-shock computation of a boom footprint is fairly fast, between a few seconds and a few minutes. Calculation of the Burgers equation can take tens of seconds or even minutes on a single ray. That level of detail may not be required on all rays. The Burgers calculation is therefore performed as a postprocessor to FOBoom via PCBurg or HeadlessBurgers (Section 5.1.3).

5.1.2.2 Burgers Equation Modeling Workflow

Propagation modeling with molecular relaxation via Burgers Equation modeling is done in PCBurg following the numerical methodology discussed in Section 5.1.2.1. The numerical solution of Equation (30) is done in PCBurg via discretization as in Equation (108). The main propagation loop in PCBurg operates in increments according to rows of data in the .age file. Each increment is further subdivided into time steps. At each time step, the basic procedure is to:

1. Calculate how long the time step will be and get atmospheric and propagation parameter values at the beginning and end of the time step (Section 5.1.2.2.2),
2. apply molecular relaxation to the sonic boom signature (Section 5.1.2.2.3),
3. apply nonlinear steepening to the sonic boom signature (Section 5.1.2.2.4),
4. (optionally) apply anti-Gibbs-phenomenon filter (Section 5.1.2.2.5).

When the time-step loop reaches the end of a .age file increment, PCBurg calculates metrics from the pressure signature and updates the plot window (Section 5.1.2.2.6).

5.1.2.2.1 Preliminary steps

PCBurg starts in legacy propagation mode by default. Thin shock aging, as described in Section 4.2.2, is used to model the signature from the flight altitude to the second altitude in the .age file (generally the first integer multiple of 1,000 ft below the starting point). At that point, the user may invoke the Burgers computation for the rest of the propagation path, or continue to use the Middleton–Carlson–Hayes methodology to age the signature through altitudes specified in the .age file down to a desired altitude before transitioning to Burgers computations. Burgers computations will continue in 1,000 ft increments for the remainder of the propagation path, once the Burgers mode is invoked. After Burgers calculations are invoked by the user, but before starting Burgers calculations, the signature is evenly resampled to 51,200 samples per second (sps) as default. The user can specify via command line options the following sampling rates (in sps):

- 10,000
- 25,600
- 51,200

- 102,400.

The sampling algorithm first linearly interpolates the input signature onto an evenly spaced time array. Next, a smoothing routine is applied so that the shocks are not pure single point jumps. Fourier series representations of functions with sharp jumps can exhibit spurious overshoot and oscillations known as Gibbs phenomenon. A smoothing filter is applied to the starting signature to remove pure single value shocks. This smoothing filter was added to PCBurg to help avoid the creation of the Gibbs phenomena artifacts in the starting signature [6]. The smoothing routine uses a 9-point moving average with Hann-window weighting. For a point p_i in the original pressure signature, the smoothed point $p_{i,new}$ is

$$p_{i,new} = \frac{1}{\sum w(j)} \sum_{j=-4}^{4} \big(p(i+j)w(j)\big)$$

(112)

where

$$w(j) = \quad \text{9-point Hann window function, } \frac{1}{2}\left(1 - \cos\left(\pi\frac{j+4}{4}\right)\right), -4 \leq j \leq 4.$$

5.1.2.2.2 Time step initiation

The discrete time step Δt in Equation (108) follows from the differential time dt in Equation (30). Within PCBurg, the nominal time advance step Δt_{nom} used in the propagation calculations is capped at 1,000 times the time increment in the evenly-sampled pressure signature, and is thus dependent on the prescribed signature resampling rate. The nominal time step is also divided by the maximum absolute value of the signature pressure. That is, the nominal time step factor is made shorter for stronger overpressures.

The actual time step used (Δt) varies and is determined according to the logic described in Section 5.1.2.2.4. At each time step, the atmospheric pressure, temperature, and relative humidity are determined via linear interpolation on time of values in the .age file and assumed constant over the time step. Note that atmospheric temperature is not included in the .age file, and is back-calculated in PCBurg from the local sound speed.

5.1.2.2.3 Molecular Relaxation

The molecular relaxation modeling in PCBurg is done following the methodology described in Section 3.3.2.1.1. The molar concentration of water vapor is calculated from atmospheric temperature, pressure, and relative humidity using the formulas found in ANSI/ASA S1.26-2014 [38]. The second term on the right side of Equation (109) is evaluated by the subroutine RELAX, which uses an exponential form. Complex air absorption from ANSI/ASA standard S1.26-2014 is used.

5.1.2.2.4 Nonlinear Steepening

Nonlinear steepening of the pressure signature is done in PCBurg following the methodology described in this section. The first term on the right side of Equation (109), taking the Blok multiplier into consideration, is evaluated by subroutine STEEP. Routine STEEP implements the first term through the application of Whitham's rule and advances each sampled point. Following the steepening, the signature is resampled to the original user-specified sampling frequency. Effects of molecular relaxation and nonlinear steepening are then added using Equation (109).

PCBurg advances the signature and displays one 1,000-ft layer at a time. Internally, it computes the waveform evolution based on a number of smaller Δt steps across each layer. Δt is calculated at each step using the following procedure:

- The steepest segment in the signature is found, by dividing each pressure change by the sample interval.
- The shock formation time for that segment is computed, using Whitham's steepening rule.
- The shock formation time is multiplied by a factor " dtfact", which is always less than 1.0 and determines the step size Δt. The default value of dtfact is 0.05.
- After each Δt step, the time remaining in the interval is computed. When the remaining time is less than the calculated Δt, a final step of the remainder is taken to finish the layer.

An implicit assumption is that the time step is small enough so the signature does not fold over during the time step, as illustrated in Figure 3-21. Toward that end, the time step is set to be a small fraction (variable dtfact = 0.05) of the shock formation time from Whitham's rule steepening.

5.1.2.2.5 Optional Anti-Gibbs-Phenomenon Filtering

An optional smoothing filter can be applied as the signature is evolved to remove ringing artifacts resulting from the Gibbs phenomenon. That filter follows a two-step procedure and is invoked after each call to the nonlinear steepening subroutine:

1. Step through the pressure signature examining 4-point windows. Calculate the mean of each window. If any group of four points crosses the local mean value three times (three consecutive changes in sign of the slope), those points are marked as following a zigging path.
2. For any points p_i marked as zigging, apply a 3-point smoothing using

$$p_{i,\text{new}} = \frac{1}{4}(p_{i-1} + 2p_i + p_{i+1}). \tag{113}$$

Points in the pressure signature not marked as zigging are retained without modification.

5.1.2.2.6 Post-propagation Steps

Following each propagation segment, several quantities are calculated and displayed. The displayed rise time is calculated as maximum overpressure divided by the maximum positive slope, restricted to the front half of the pressure signature. The displayed thickness corresponds to the time that transpires between 10% of the maximum overpressure and 90% of the maximum overpressure. Various acoustic metrics are calculated according to the methods described in Section 3.5.

If propagation modeling has reached the ground level indicated in the .age file and the user enters keypress 'g', a ground reflection factor is applied. This is done by multiplying each point in the pressure signature by 1.9 (default) or the value specified with command-line option GR. The metrics are recalculated, and the plot window updated with the final pressure signature. Note that invoking 'g' is irreversible.

5.1.3 HeadlessBurgers

The HeadlessBurgers tool contains the same algorithms as PCBurg, but does not include the graphical user interface, allowing for its use via batch calculation methods. There are a few differences between the HeadlessBurgers and PCBurg:

1. The HeadlessBurgers does not use the command-line TW or PW inputs, but rather does propagation modeling for all T_{AC} and ϕ values found in the .age file.
2. The HeadlessBurgers enters the Burgers equation propagation section of the algorithm at the starting point 1,000 ft below the aircraft, rather than allowing the user to select an altitude

point at which to begin Burgers equation modeling. Just as in PCBurg, the Middleton–Carlson–Hayes method is used to age the signature to the starting point.

3. Default sampling rates for PCBurg and HeadlessBurgers are 51,200 Hz and 102,400 Hz respectively.

5.2 PCBFoot

The PCBFoot module applies artificial aging to legacy FOBoom calculations to approximate atmospheric absorption. PCBFoot also organizes the FOBoom footprint and signature outputs into structured files for use in WCON. When the GROUND keyword is in use, PCBFoot computes the signatures for elevated microphones by applying the effects of ground impedance to the sonic boom signal.

5.2.1 Summary of Inputs

The principal input file for PCBFoot is the .out file generated by FOBoom as it sequentially analyzes the user defined trajectory. The .out file is organized according to the analysis trajectory point sequence and a sweep across the calculation azimuths at each trajectory point. Additional input files include the .gmp file for cases to apply ground impedance effects, and .org file when longlat or geomode are used in FOBoom, and an optional user-generated HighRes file for specifying the sampling rate for PCBFoot generated signature files.

5.2.2 Calculations

Finite rise times for legacy calculations are obtained by applying Taylor shock absorption (Section 3.3.1.3) to pressure signatures, from which metrics at ground receivers are calculated. Note that, although this is done by default, a user option exists to skip the shock-thickening step. Focus handling in PCBFoot utilizes the single-shock solution as described in Section 3.3.4.2. If specified, a ground reflection factor is applied and metrics (Pmax, ASEL, BSEL, CSEL, DSEL, ESEL, ZSEL, PL) are computed as described in the User's Guide [8].

Ground boom and focus signatures (including supplemental information not necessarily at the ground altitude) are written to the .out file as computed by FOBoom, and are tagged as:
- "GROUND" – signature at the ground altitude
- "LAST GEOMETRIC SIGNATURE" – last calculation by legacy propagation before focusing algorithms are applied.
- "FOCUSED BOOM SIGNATURE" – signature within the focal zone

In the case of focusing, the .out file also contains information about the radius of caustic curvature and the focus coordinates (geometric or longitude/latitude and altitude). PCBFoot interprets the focal zone signatures depending on whether the focus point is above or below the ground. In the case where the focus is below the ground and the ground is within the focal zone (Equation 50), the signature type (displayed in WCON) is set to "Maximum Focus Boom". If the focus is above the ground and the ground is within the focal zone, the focused boom signature is used and the signature type is set to "Maximum Focus Boom". If the focus is above the ground but the ground is outside of the focal zone, the signature is zeroed out and the signature type is set to "No Boom".

The ground impedance methodology described in ANSI S1.18-1999 [69] Appendix B has been implemented in PCBFoot. Details of that method as implemented in PCBFoot are described in Section 3.4.4. The signal is assumed to be at the ground. The output is the reflected signal in time, at the receiver. The angle of incidence is defined by direction cosines of the incoming primary ray. The method

takes a Fourier transform of the input signal. It applies the ground impedance to the Fourier coefficients of the bounced signal and an additional phase change representing the time delay for the bounce. This is then summed with the transform of the direct signal. The inverse transform is then applied to obtain the signature at the receiver.

5.3 WCON

The WCON module is an interactive sonic boom footprint and signature display module. This module interpolates boom signatures (including secondary post-focus u-waves) between ray ends provided by PCBFoot in order to allow users to retrieve boom signatures at desired locations.

Ground effects on the pressure waveform received at a user-specified elevation above ground level can also be viewed in WCON. WCON also includes algorithms to compute and view turbulence effects using several different methods and to compute booms that have penetrated beneath a water surface. Additionally, the spectrum and residual shock spectrum of the booms can be displayed.

Pmax, ASEL, BSEL, CSEL, DSEL, ESEL, ZSEL, PL, and Lpk metrics are available in WCON. Note that before computing metrics, the waveforms are zero padded to the next power of 2 in array length if the number of points is not a power of 2. This is to conform to the signal processing requirements of the FFT algorithms. Also, note that no window is applied to force the ends of the waveform to zero, nor is the mean value subtracted from the waveform.

5.3.1 Summary of Inputs

WCON primarily reads output files from PCBFoot and FOBoom (.qwk, .sig, .ind, .ens, .asc, .org) that contain the footprint, metric data, isopemp data, signature data, shock indices, incident ray angles, position and flight condition information, centerline boom location, and overpressure information. The SonicBAT-Run-Fort config file is also used when accessing the FiltView module within WCON for incorporating turbulence effects. A full list of accepted inputs is described in the PCBoom User's Guide [8].

5.3.2 Calculations

WCON reads in the isopemps and maps the vector coordinates (time and ϕ indices, each pair of which defines a cell) into a physical coordinate system using linear interpolation of the boom intercept coordinates across the cells. Constant-metric contours are calculated at user-configurable levels with no smoothing applied, and contour lines are plotted in the WCON GUI.

Within WCON, two methods of estimating turbulence effects on ground waveforms are implemented. From the WCON signature display for a selected ground point, a user can invoke calculations of a root-mean-square (RMS) envelope on a waveform via Crow's method (Section 3.3.3.1), or apply NPE-based FIR filters (Section 3.3.3.2.1). Alternatively, the FiltView module (5.3.3) can be called from the WCON display to estimate turbulence effects using KZK-based FIR Filters.

5.3.2.1 WCON Turbulence: Crow's Method
Crow's method is implemented as a method of defining the range of effects turbulence might have on a particular waveform, see Equation (35). In that equation, the h_c parameter is related to the turbulence strength and is supplied as a user defined input. WCON actually uses a time parameter t_c and time behind a shock t in place of the distance-based parameters in Equation (35). The rationale for this is that

the numerical value of t_c in milliseconds is within about 10 percent of the numerical value of h_c in feet, and (considering the approximate nature of any estimate of turbulence) it is common to use them interchangeably [39]. The time parameter is calculated as

$$t_c = \frac{S \times th}{3}$$ (114)

where

$S =$ Turbulence strength factor; $S = 0.5$ for user-input 'c' and $S = 1.0$ for user-input 'C'

$th =$ Shock thickness, based on maximum slope and calculated as $\frac{p_{max}}{\left(\frac{dp}{dt}\right)_{max}}$.

Perturbations to the waveform are calculated employing first-order centered finite differences to calculate the first derivative of the acoustic pressure. The pressure perturbations are smoothed using a three-point center-weighted smoothing function given by

$$p_{i,\text{new}} = \frac{1}{4}(p_{i-1} + 2p_i + p_{i+1})$$ (115)

As a final step, the upper and lower components of the RMS envelope are calculated by adding and subtracting the pressure perturbations from the input waveform.

5.3.2.2 WCON Turbulence: NPE-based FIR Filters

To apply the turbulence effects via NPE-based FIR filters in WCON, the user selects an appropriate filter from the list in Table 3-1. For a filter F of length $n = 1200$ points and a pressure signature P with m points, the convolution of P and F has $m + n - 1$ points. The convolution w is calculated as

$$w(k) = \sum_{j=j_1}^{j_2} P(j)F(k - j + 1)$$ (116)

where

$j_1 = \max(1, k + 1 - n)$
$j_2 = \min(k, m)$.

The convolution is then normalized to ensure that the output waveform has the same total energy as the input signature. That normalization is done according to

$$w_{\text{scaled}}(k) = w(k)\sqrt{\frac{\sum_j P^2(j)}{\sum_j w^2(j)}}.$$ (117)

The scaled, convolved waveform is plotted in WCON and represents a signature at the ground propagated through a turbulent ABL.

5.3.3 FiltView

The FiltView module, which is called from the WCON signature display, convolves a set of KZK-based FIR filters with the selected WCON signature in order to represent the effects of turbulent atmosphere on the input signature. The filters were developed by Stout and Sparrow at PSU [45, 56], and the theoretical underpinning is described in Section 3.3.3.2.2.

Prior to launching FiltView, WCON resamples the pressure signature at an even rate of 51.2 kHz. That pressure signature is passed by file to FiltView, along with the following inputs:

1. user-specified turbulence level (low, medium, high),
2. user-specified atmospheric boundary-layer height (ABLH),
3. user-specified relative humidity, and
4. ray elevation angle θ (from WCON), which is assumed to be constant across the ABL to the ground.

A set of 2,016 FIR filters was precomputed using simulations of propagation through turbulence; the matrix of conditions used for those simulations is given by Table 3-2. Those 2,016 filters are the product of 7 metrics (ISBAP, PL, Pmax, ASEL, BSEL, CSEL, ESEL), 3 filter types (mean, mean + 1 standard deviation, mean − 1 standard deviation), and 96 simulation conditions. Those 96 simulation conditions are the combination of parameters in Table 3-2. FiltView performs a filter number lookup by calculating the sum of the following bits.

1. Turbulence strength:
    a. If low, start sum at 1
    b. If medium, start sum at 2
    c. If high, start sum at 3
2. Atmospheric boundary-layer height (ABLH):
    a. If ABLH > 6233.6 ft, add 36
    b. If 4265.1 ft < ABLH ≤ 6233.6 ft, add 24
    c. If 2296.6 ft < ABLH ≤ 4265.1 ft, add 12
    d. If ABLH ≤ 2296.6 ft, add 0
3. Ray elevation angle, θ:
    a. If θ > 37.5°, add 0
    b. If 29.5° < θ ≤ 37.5°, add 3
    c. If 23.5° < θ ≤ 29.5°, add 6
    d. If θ ≤ 23.5°, add 9
4. Relative humidity:
    a. If RH > 35%, add 48

That sum is then used to select a filter number between 1 and 96. For a filter of length F of n points (generally 3200 or 3201) and a pressure signature P with m points, the convolution of P and F has $m + n - 1$ points. As with the NPE-based FIR filters, the convolution w is calculated using

$$w(k) = \sum_{j=j_1}^{j_2} P(j)F(k - j + 1) \tag{118}$$

where

$$
\begin{aligned}
j_1 &= \max(1, k + 1 - n) \\
j_2 &= \min(k, m).
\end{aligned}
$$

The convolution is then the waveform at the ground having propagated through a turbulent ABL.

5.4 Turbo

The classical turbulence code within the Turbo module is a fully 3D, linear acoustic propagation allowing turbulent temperature and gust variation versus altitude, along with mean temperature and wind variations [45]. The Turbo module is compatible with SonicBAT-Run-Fort and HeadlessBurgers to generate results that superimpose the separate (no interactions) effects of nonlinear aging and rounding as well as mean and turbulent atmospheric variations.

5.4.1 Summary of Inputs

Turbo primarily reads input and output files from FOBoom (.dat, .out, .age, ablh.out) that contain signature data, aging details, and atmospheric data at different altitudes, along with the traced rays, and information about the ray states at the altitude corresponding to the input atmospheric boundary-layer height. An additional input file (-tin.txt) specifies turbulence modeling parameters and ray gridding parameters.

5.4.2 Calculations

The internal Turbo workflow can be categorized into three main areas: (1) atmospheric data input and turbulent perturbation modeling, (2) ray propagation modeling, and (3) summation of ground signatures. The theoretical approach that Turbo utilizes is summarized in Section 3.3.3.3 of this document. Additional details can be found in Chapters 4 and 7 of the SonicBAT report by Bradley et al. [45].

5.4.2.1 Turbulence Modeling

Profiles of mean (i.e., nonturbulent) atmospheric quantities are read by Turbo from PCBoom output files, specifically the x- and y-wind speeds, temperature, pressure, and relative humidity. Turbo also requires information characterizing the atmospheric turbulence within the mixed layer. Quantities that a user provides in the Turbo -tin.txt file include:

1. z_i, the height of the mixed layer;
2. l_0, the inner scale of the turbulence;
3. L_0, the outer scale of the turbulence;
4. C_T^2, the temperature structure parameter for the thermal turbulence; and
5. C_v^2 , the velocity structure parameter for the kinetic turbulence.

The inner scale of the turbulence can be conceptualized as the smallest turbulent eddy size and the scale at which energy is dissipated by viscosity. It is typically on the order of millimeters. The outer scale of the turbulence can be conceptualized as the largest turbulent eddy size present in the atmosphere, and can be on the order of tens and hundreds of meters. The cascade of energy from larger scales to smaller scales defines the turbulent energy spectrum in the inertial subrange. The temperature structure parameter C_T^2 and velocity structure parameter C_v^2 control the amplitude of turbulent spectra for the thermal and velocity portions of the turbulence, respectively. As noted by Bradley et al. [45], C_T^2 and C_v^2 can vary separately, as in the case of a windy day with little sun, for example.

Turbulence modeling in Turbo is implemented using the Random Fourier Modes approach of Blanc-Benon et al. [55]. Speed of sound fluctuations in the propagation medium are calculated from either a temperature fluctuation or a velocity fluctuation, each of which is derived from corresponding turbulent spectra [45]. Random temperature fields T' and random velocity fields v' at points $\boldsymbol{x}$ can be represented as sums of N random Fourier modes as

$$T' = \sum_{i=1}^{N} \mathcal{T}'(\boldsymbol{K}) \cos(\boldsymbol{K} \cdot \boldsymbol{x} + \varphi_i) \tag{119}$$

and

$$v' = \sum_{i=1}^{N} \mathcal{V}'(\boldsymbol{K}) \cos(\boldsymbol{K} \cdot \boldsymbol{x} + \varphi_i) \tag{120}$$

where

$\qquad \mathcal{T}'(\boldsymbol{K}) = \quad$ Amplitude of the thermal turbulence modes

$$\mathcal{V}'(\boldsymbol{K}) = \quad \text{Amplitude of the velocity turbulence modes}$$
$$\boldsymbol{K} = \quad \text{Turbulence wavevector with random direction}$$
$$\varphi = \quad \text{Random phase.}$$

Turbo calculates turbulence strengths at wavenumbers K for temperature fluctuations using

$$\frac{\mathcal{T}'(K)}{C_T^2} = \sqrt{\frac{0.792}{1.91} K^2 \frac{1}{L_0^{2/3}} \left(K^2 + \frac{1}{L_0^2}\right)^{-11/6} e^{-K^2/K_m^2}} \tag{121}$$

and for velocity fluctuations using

$$\frac{\mathcal{V}'(K)}{C_v^2} = \sqrt{\frac{11}{6} \frac{0.792}{1.91} K^4 \frac{1}{L_0^{2/3}} \left(K^2 + \frac{1}{L_0^2}\right)^{-17/6} e^{-K^2/K_m^2}} \tag{122}$$

where
$$\mathcal{T}'(K) = \quad \text{Amplitude of the thermal turbulence modes}$$
$$\mathcal{V}'(K) = \quad \text{Amplitude of the velocity turbulence modes}$$
$$K = \quad \text{Turbulence wavenumber}$$
$$K_m = \quad 5.92/l_0.$$

The change in local speed of sound c' is calculated from

$$\frac{c'(K)}{C_T^2} = \frac{\mathcal{T}'(K)}{C_T^2} \frac{1}{T_0 a_0} \tag{123}$$

where
$$T_0 = \quad \text{Standard temperature, 288.16 K}$$
$$a_0 = \quad \text{Speed of sound at standard temperature.}$$

Equations (121) and (122) are based on thermal turbulent energy spectra, $G(K)$, and kinetic turbulent energy spectra, $E(K)$, respectively. Those equations follow the modified von Karman spectra described by Cotte and Blanc-Benon [62]. To form Equations (121) and (122), the thermal turbulent energy spectra and kinetic turbulent energy spectra are cast in terms of the structure parameters, making use of the relations $C_T^2 = 1.91\overline{T'^2}L_0^{-2/3}$ and $C_v^2 = 1.91\overline{V_x'^2}L_0^{-2/3}$ [62].

The Random Fourier Modes approach [55, 57] implemented in Turbo utilizes vector modes, which have a random velocity direction and a random direction of variation that is normal to the direction of the velocity. Note that, currently, only turbulent velocity (gust) modes (C_v^2) are enabled in Turbo, not turbulent temperature modes (C_T^2).

5.4.2.2 Propagation Modeling
Turbo implements linear acoustic ray tracing and calculates both ray turning and displacement to capture variations in amplitude and focusing from refraction, plus rounding and distortions from scattered multipath arrival times. For each signature azimuth where a turbulent assessment is requested, the mean atmospheric conditions of temperature, x-winds and y-winds as a function of altitude above sea level are read along with ray starting location, vector direction, and time at the top of the ABL.

1. Propagation steps are calculated by estimating the local ray path between discrete steps i and $i + 1$ as circular arcs. For the mean flow (no turbulent variations) in a stratified atmosphere, the speed of sound c, x and y wind components w_x and w_y vary only with z. The radius of curvature with respect to the propagation direction is calculated as

$$R_c = \frac{\bar{c}}{\dfrac{dc}{dW} + \dfrac{dw_x}{dW} + \dfrac{dw_y}{dW}} \tag{124}$$

where
$$\bar{c} = \quad \text{Average speed of sound between steps } i \text{ and } i + 1$$
$$W = \quad \text{Propagation direction.}$$

Ray angles relative to $x - y$ planes at the start and end of the step (denoted by subscripts 0 and 1, respectively) are calculated as

$$\alpha_0 = \sin^{-1}(-\hat{r}_z), \text{ and}$$
$$\alpha_1 = \alpha_0 - \frac{\bar{c}\Delta t}{R_c} \tag{125}$$

where
$$\hat{r} = \quad \text{Ray unit vector; subscript } z \text{ indicates component in } z \text{ direction}$$
$$\Delta t = \quad \text{Time step between steps } i \text{ and } i + 1.$$

The ray vector at the end of the step is then given by

$$\hat{r}_1 = \left(\frac{W_{z,1}}{W_{z,0}} \hat{r}_{x,0}, \frac{W_{z,1}}{W_{z,0}} \hat{r}_{y,0}, \sin^{-1} \alpha_1 \right). \tag{126}$$

For propagation through turbulent conditions, unit vector turning gradients due to wind variations from the random Fourier modes are calculated at each step. More detail is given in Reference [45].

2. For integration of the Fourier gradients, each time step location is calculated from an average of speed of sound and winds, and an integration of turbulent variations in speed of sound and velocity (gusts). The change in direction due to these effects is averaged over the time step and resolved as a constant curvature arc during the time step.

3. The average speed of sound and winds are calculated at the end points, and the variations in speed of sound and gusts are integrated along a straight-line approximation of the arc.

4. Ray propagation continues until the ground elevation is crossed or a built-in time step limit is reached.

5. The exact ray-ground intersection is then calculated between the last two steps. After all ray propagations are complete, the ground intersection point for each ray tube is checked for intersections with rays from all the originating times (NSIGTIMES, see References [8, 45]) requested at the azimuth of interest as described in Section 5.4.2.3.

For each pair of (T_{ac}, ϕ) associated with a PCBoom nonturbulent ray, Turbo propagates a grid of rays from the top of the ABL. Unlike the time and azimuth increments used by FOBoom to assemble a ray tube, the grid of rays assembled by Turbo uses regular distance increments aligned with the local direction of isopemp at the originating point, and along the flight direction. Typical distances used are

on the order of a few feet. The grid of rays assembled by Turbo extends before and after the T_{ac} of interest, and to either side (along the isopemp) of the ϕ of interest. Successive grids of rays can be used to model a virtual microphone array on the ground.

Because of the methodology implemented, Turbo is referred to as a repropagation model utilizing turbulence and mean atmospheric conditions. As described in Reference [45] and Section 3.3.3.3, differences in propagation time between the nonturbulent and turbulent cases are small, on the order of milliseconds. As such, Turbo neglects nonlinear aging differences and saves computational time by not repeating the nonlinear signature aging calculations. Scaling of ground signatures for rays propagated through turbulent conditions is described in Section 5.4.2.3.

5.4.2.3 Signature Summation

The final step in Turbo workflow is a signature summation step, in which one or more ground signatures are summed. The outline of this procedure is as follows.
1. Ray tubes, which are initially arranged in regular parallelograms based on time steps and phi steps, are cut into triangles (so-called tritubes) for greater resolution per ray calculation. The cuts are made along the corners that were closer before turbulence.
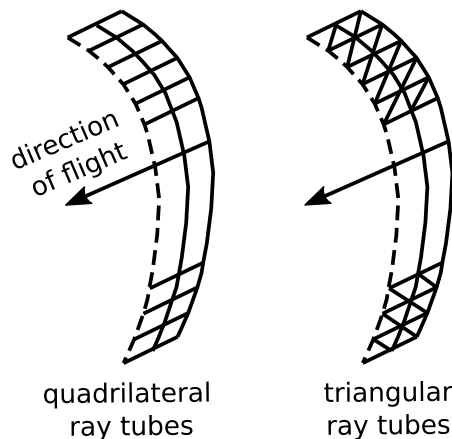
Figure 5-1: Illustration of Quadrilateral Ray Tubes and the Triangular Ray Tubes in Turbo
(*Source: Adapted from Figure 115 in NASA/CR–2020–22050,. Bradley et al.* [45]).

2. The ground intersection point of each tritube is checked for intersection with ground points of interest. Signature superposition is based on the number of tritube intersections found with the desired ground point. For each tritube, an area ratio is calculated between the tritube and PCBoom's nonturbulent calculation. The area ratios are used to scale the signature amplitude contributed by each tube.
3. Scaled signatures are shifted in time to match the tritube intersection times.
4. Signatures from all intersecting tritubes are interpolated to a common time scale (defined by the sampling rate specified in the -tin.txt file) and summed into a single signature.
   a. To avoid potential problems with interpolation, ground signatures read into Turbo are adjusted to remove multivalued points by stretching such segments by 10^{-6} ms. Similarly, shocks are rounded if:
      i. The maximum slope > 0.3/TIMEMIN, where TIMEMIN = 2 ms,
      ii. or time step < 0.000003 ms

b. For linear, discontinuous signatures (i.e., not from lossy Burgers calculations), Turbo uses linear interpolation.

c. For rounded signatures from either HeadlessBurgers or sBOOM, Turbo uses a weighted angle cubic fit to interpolate the signatures. A description of that interpolation algorithm is given in Appendix A.

5. Summed signatures from each originating trajectory time and azimuth angle are written to the tur.out file. That file follows the format of the FOBoom .out file; thus, PCBFoot can be used as a post-processor.

6   Specialized Tools

6.1   RayCau

The RayCau graphical user interface visualizes the caustic and ray(s) near a focus. The initial display shows the flight parameters corresponding to the caustic's ground intercept, and the caustic tangent ray, all at an azimuth angle of 0°. Other azimuth angles, rays, and focus points can also be visualized upon keyboard command. The output contains command data for use by PCBurg or HeadlessBurgers in preparing a signature for input to LNTE.

6.1.1 Summary of Inputs

RayCau reads the trajectory and focus information from the FOBoom .out file, including location of focus points. Ray and atmospheric information are read from the FOBoom .age file, and the origin in Cartesian coordinates from the FOBoom .org file.

6.1.2 Calculations

Much of the data visualized in the GUI and output by RayCau are simply passed through from FOBoom output, although RayCau does perform a few novel computations for both visualization and output for PCBurg or HeadlessBurgers.

RayCau transforms the Cartesian coordinates of the local tangent plane to geographical coordinates for flat Earth using the method described in Equation (63). The focus ground intercept is computed by linearly interpolating (or extrapolating) using the last two focus points. For the example illustrated in Figure 6-1, the caustic ground intercept is found from linear interpolation between focus points 2 and 3. The caustic radius of curvature is determined by fitting focus points to a circle with center (a_m, b_m) and radius r_m, where the subscript m refers to the "Modified Least Squares Method" by Umbach and Jones [101]. The elevation angle, θ, of the ray ground intercept is calculated for each azimuth angle using

$$\theta = \cos^{-1}\left(\frac{z_g - b_m}{r_m}\right)$$

(127)

where

$$
\begin{aligned}
z_g &= \text{ground elevation}\\
b_m &= \text{vertical coordinate of caustic circle}\\
r_m &= \text{radius of circle fitted to focus points.}
\end{aligned}
$$
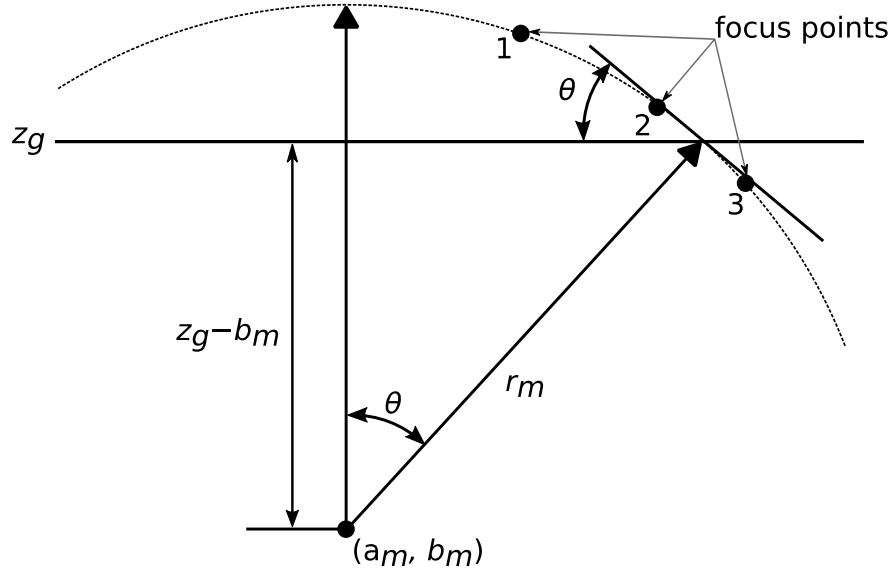
Figure 6-1: Illustration of Caustic Circle Fitted to Focus Points by RayCau.

So that the RayCau GUI can identify which of the rays modeled in FOBoom is closest to one diffraction boundary-layer thickness, δ, from the caustic ground intercept, the shortest distance from each ray to the focus ground intercept is found using the Pythagorean theorem. The RayCau GUI then draws the ray whose shortest distance is closest to δ.

6.2   LNTE

A method of solving the nonlinear Tricomi equation, which governs the evolution of signatures in the vicinity of a caustic, has been developed by Salamone [102] and incorporated into PCBoom. This solver computes the signature at the focus as well as in the vicinity of the focal zone, including in the evanescent and post-focus regions. Salamone uses a numerical algorithm similar to Auger and Coulouvrat [90], except that the numerical implementation includes an extra step to solve for the newly derived atmospheric loss mechanisms. The algorithm is referred to as the Lossy Nonlinear Tricomi Equation (LNTE).

Figure 6-2 shows the region associated with the Tricomi equation in a wave-fixed reference frame. The red line denotes a ray tangent to the caustic line where the maximum focus occurs. The arrows on the right indicate the notional edge of the diffraction zone at $\overline{z} = 1$. FOBoom defines the "Last Geometric Signature" at distance S^*, representing its ray-oriented boundary condition. In FOBoom, the Gill–Seebass [88] solution for a step shock is applied to each shock in the last geometric signature, scaling the amplitude and shape according to Guiraud's similitude [65]. For LNTE, the boundary condition is the full incident signature at the top edge of the region, indicated by the N-wave in Figure 6-2.
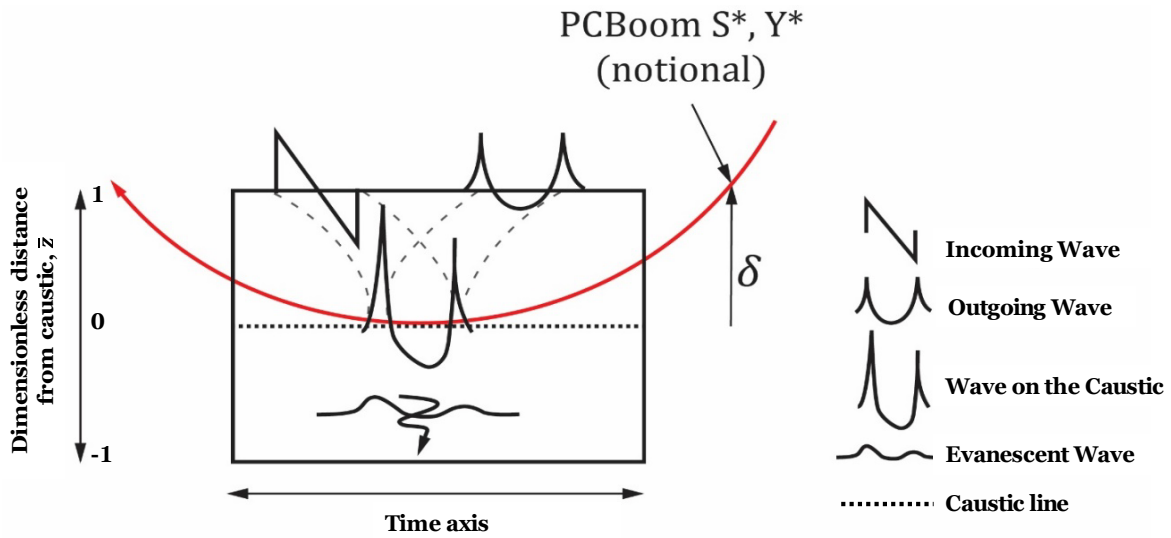
Figure 6-2: Focal Zone as Defined by the Tricomi Solution Region, with Tangent Ray Shown
(*Source: Adapted from Auger and Coulouvrat* [90] *and Salamone* [102]*, with permissions, and NASA-CR-218871* [63]).

Figure 6-3 shows the focal zone and its relation to the overall boom ray field. The region governed by the LNTE is shown in the lower left, along with a "Caustic tangent ray" that is equivalent to the red ray shown in Figure 6-2. A second ray, denoted "δ tangent ray", is tangent to the top of the Tricomi region. The boundary condition for LNTE is the "Incoming signature" shown along the δ tangent ray. This is different from the "Last Geometric Signature" boundary condition that FOBoom uses internally for the Gill–Seebass [88] solution.
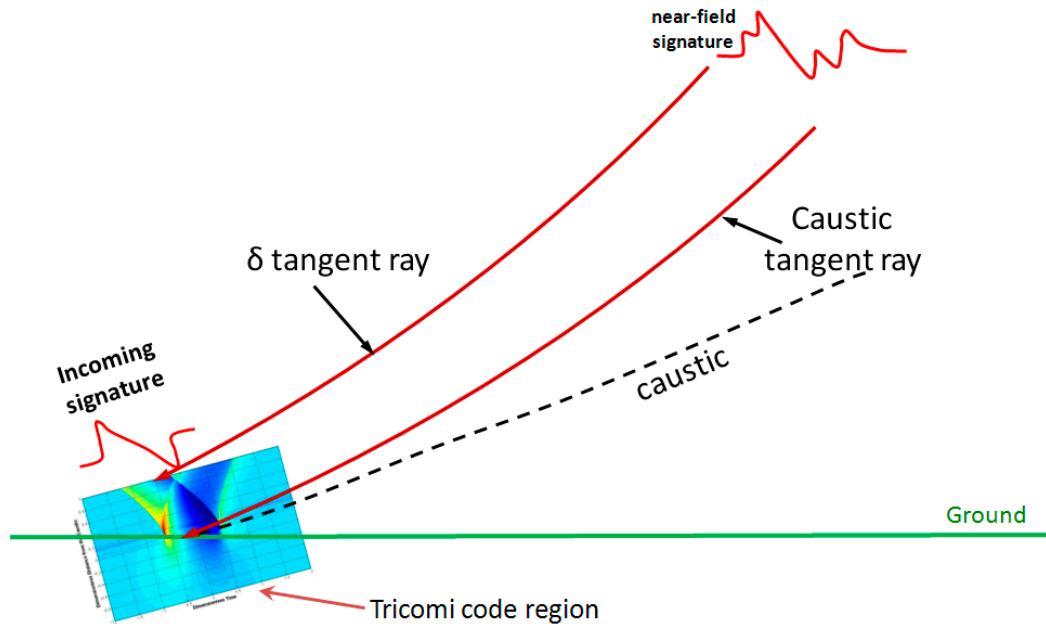
Figure 6-3: Rays, Caustic, and Focal zone (*Source: NASA-CR-218871* [63]).

116

6.2.1 Summary of Inputs

Using the –pb, command line option, LNTE writes its own input file (automatically named <BurgersOutputFileName>_tricomi.inp) after reading in the signature output file (.txt) from PCBurg or HeadlessBurgers [63]. The LNTE input file consists of three header rows followed by a single column of signature pressure data in psf. The first row consists of the following text: "Tricomi format input file created from PCBoom Burgers output file: <BurgersOutputFileName.txt>". The second row contains the following data:

- Characteristic acoustic frequency, f_{ac} (Hz),
- Total radius of curvature, R_{tot} (meters),
- Temperature (Kelvin),
- Atmospheric pressure (kPa),
- Relative humidity (%),
- x-wind component (m/s),
- y-wind component (m/s),
- dimensionless distance below the caustic $\bar{z}_{min}$ (ratio),
- dimensionless distance above the caustic $\bar{z}_{max}$ (ratio), and
- number of z-axis layers (points).

The characteristic acoustic frequency (f_{ac}) is automatically computed from the incoming waveform using the Sparrow 5%-95% rule [63], which defines the effective signature length as the length spanning points that correspond to 5% of the total acoustic energy of the signal and 95% of the total acoustic energy of the signal. The total radius of curvature (R_{tot}) is extracted from the PCBurg or HeadlessBurgers output file. The value is computed in FOBoom, which are passed to the Burgers solver, using [2]

$$\overrightarrow{k_{rel}} = \left[1 - \frac{\overrightarrow{k_r} . \overrightarrow{k_c}}{\overrightarrow{k_c} . \overrightarrow{k_c}}\right] \overrightarrow{k_c} \tag{128}$$

where

$$\overrightarrow{k_{rel}} = \text{the difference between the caustic curvature and the normal}$$
component of the ray curvature
$$\overrightarrow{k_r} = \text{the ray curvature}$$
$$\overrightarrow{k_c} = \text{the caustic curvature}$$

The radius of curvature is calculated as

$$R_{tot} = 1/|\overrightarrow{k_{rel}}|. \tag{129}$$

The temperature, atmospheric pressure, relative humidity, and wind components are the ambient values at the caustic that are also extracted from the PCBurg or HeadlessBurgers output file. The atmospheric profile is defined in FOBoom and passed on to the Burgers solver.

The $\bar{z}$ bounds default respectively to −1 and 1. The number of z-axis data layers defaults to 4,000 with a default time domain sample rate of 32 kHz in order obtain sufficient grid convergence while reducing the Gibbs-like oscillations in the solution near a steep pressure increase (see Figure 3.6 in [102]). If the

user chooses to increase the default incoming waveform resampling rate, the number of z-axis layers must be increased accordingly to reduce the amplitude of said oscillations.

The third row of the LNTE input file contains the number of points in the incoming wave (points), as well as its sampling rate (Hz). Both data are extracted from the Burgers solver output file.

The column of signature pressure data that follows is copied directly from the PCBurg or HeadlessBurgers output file.

6.2.2 Calculations

A numerical solution is generated from the LNTE module when it is provided an incoming waveform, acoustic parameters, and an atmospheric profile. The LNTE is given by

$$\frac{\partial^2 p}{\partial \bar{z}^2} - \bar{z}\frac{\partial^2 p}{\partial \bar{t}^2} + \frac{\beta}{\varepsilon^2 \rho_0 c_0^2}\frac{\partial^2 p^2}{\partial \bar{t}^2} + \left[\frac{\bar{\alpha}}{\varepsilon^2} + \sum_{v=1}^{2}\frac{\bar{\theta}_v/\varepsilon^2}{1 + \bar{\tau}_v\frac{\partial}{\partial \bar{t}}}\right]\frac{\partial^3 p}{\partial \bar{t}^3} = 0 \tag{130}$$

where

p = the acoustic pressure
$\bar{z}$ = the dimensionless distance from the caustic in the direction normal to the caustic
$\bar{t}$ = the dimensionless retarded time
β = the coefficient of nonlinearity
ε = the dimensionless diffraction parameter
ρ_0 = the ambient density
c_0 = the ambient speed of sound
$\bar{\alpha}$ = the dimensionless absorption coefficient
$\bar{\theta}_v$ = the dimensionless dispersion coefficient for the v-th relaxation component
$\bar{\tau}_v$ = the dimensionless relaxation time for the v-th relaxation component.

The first two terms in the LNTE (Equation (130)) account for diffraction, the next term accounts for nonlinearity, and the final term represents thermoviscous and molecular relaxation effects. The code iteratively solves for this equation from an initial waveform by adding an unsteady pressure term and a pseudotime variable as shown in

$$\frac{\partial^2 p}{\partial \sigma \partial \bar{t}} = \frac{\partial^2 p}{\partial \bar{z}^2} - \bar{z}\frac{\partial^2 p}{\partial \bar{t}^2} + \frac{\beta}{\varepsilon^2 \rho_0 c_0^2}\frac{\partial^2 p^2}{\partial \bar{t}^2} + \left[\frac{\bar{\alpha}}{\varepsilon^2} + \sum_{v=1}^{2}\frac{\bar{\theta}_v/\varepsilon^2}{1 + \bar{\tau}_v\frac{\partial}{\partial \bar{t}}}\right]\frac{\partial^3 p}{\partial \bar{t}^3}. \tag{131}$$

Note that the pseudotime variable is unrelated to the dimensionless time variable, $\bar{t}$. The dimensionless time is given by $\bar{t} = f_{ac}t$, which represents the time delay, scaled by the characteristic acoustic frequency, f_{ac}. The dimensionless vertical distance, $\bar{z}$, in the above equations refers to a ratio of the physical distance, z, from the caustic to the diffraction boundary-layer thickness, in the normal direction and is calculated using

$$\bar{z} = z \left[\frac{R_{tot}c_0^2}{2f_{ac}^2} \right]^{-\frac{1}{3}}. \tag{132}$$

According to this relationship, if the pressure variable in Equation (130) is normalized by a characteristic acoustic pressure, P_{ac}, the coefficient in the third term of Equation (130) must be presented as

$$\frac{\mu}{2} = \frac{\beta P_{ac}}{\varepsilon^2 \rho_0 c_0^2} = \frac{\beta P_{ac}}{\rho_0 c_0^2} \left[\frac{R_{tot}f_{ac}}{2c_0} \right]^{\frac{2}{3}}. \tag{133}$$

The variable μ in Equation (133) represents the magnitude of the nonlinear effects relative to the diffraction effects. P_{ac} is typically set to the peak amplitude value of the incoming waveform at $\bar{z} = 1$.

6.2.2.1 Burgers-LNTE Interface

The ray path geometry must be understood in order to locate the focus delta ray. Figure 6-4 shows the caustic line (light blue) and the focus delta ray (dark blue). The latter ray passes through a point at an orthogonal distance δ (diffraction boundary-layer thickness, orange in Figure 6-4) from the caustic line ground intercept and is associated with new Mach conditions. Along the focus delta ray in the region of the blue square in Figure 6-4, geometric ray acoustic theory, including the Burgers equation [33], is no longer appropriate, and a lossy Tricomi equation solver, such as the LNTE module, is required to solve for the focus delta ray's acoustic pressure and remaining path to the ground by accounting for diffraction, reflection, and propagation behavior at the caustic edge. This point will be referred to as the Burgers–LNTE interface point. The pressure signature must be extracted from PCBurg at the Burgers–LNTE interface for input to LNTE. Figure 6-4 illustrates the exact coordinates of the extraction point.
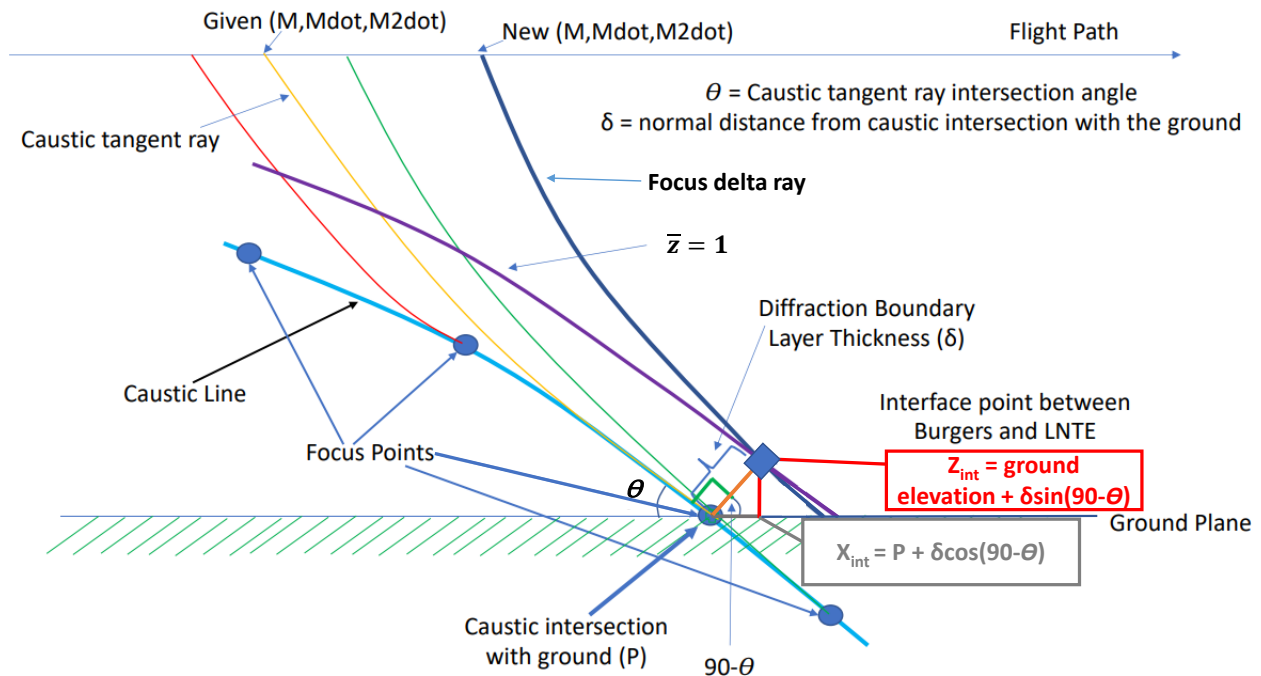
Figure 6-4: Geometric Interface between Ray Tracing and LNTE (*Source: Adapted from Rallabhandi* [103]).

6.3 POTRAY, POTTI, OTTER

Analysis of over-the-top booms can be considered in separate steps: evaluation of the ray paths to determine which ones reach the ground; evaluation of the primary and secondary boom carpets; and calculation of the signature evolution along the ray paths. The PCBoom modeling suite uses a self-standing routine called POTRAY (Propagate Over the Top Rays) to evaluate OTT rays and determine whether OTT paths that reach the ground exist. POTRAY also finds the primary boom cutoff. That routine is typically used in conjunction with another tool to plot the results and locate rays of interest: POTTI (Plot Over The Top Isopemps). Once that information is known, signature evolution parameters (age, Blok, and ray tube area) and OTT ray paths can be modeled in FOBoom. OTT ray paths are plotted and signatures along rays are calculated in a separate module called OTTER (Over The Topper).

6.3.1 POTRAY

POTRAY is a pre-processor for OTT analyses and computes sonic boom ray paths. Rays at all azimuth angles are classified by POTRAY, depending on whether paths include ground bounce points and whether ray ends are below the ground or above the ground:

> Type 1: ground reflection (bounce)

> Type 2: ray end below the ground (final ground intercept)

> Type 3: ray end above the ground - no ground intercept

Rays that refract away from the ground due to atmospheric conditions are type 3.

6.3.1.1 Summary of Inputs

The general inputs to POTRAY are trajectory information and atmospheric information, in either Thomas-style ATT format or a balloon file. Note that while POTRAY can read a PCBoom trajectory file with first and second derivatives of Mach number, heading, and flight path angle, those derivative quantities are not used by POTRAY. Trajectory and atmosphere filenames and additional information such as azimuth angle range and raytracing parameters are input through a namelist. The complete list of input requirements for POTRAY is described in the PCBoom User's Guide [8].

6.3.1.2 Calculations

As part of its input routine, POTRAY checks whether the ATT-format atmosphere files contain a pressure profile. If not, a pressure profile is generated via numerical integration of the hydrostatic equation from the base level, as described in Section 4.4. After data are read from the atmosphere file, POTRAY extends profiles to 10,000 ft below the lowest level in the atmosphere file. This is done by linearly extrapolating profiles of temperature and pressure while holding x- and y- wind constant. The exception is that for BALLOON input, temperature profiles are also held constant below the lowest supplied level.

The components of the initial ray vector $\vec{n}$ are determined as

$$n_1 = \frac{V_x}{V}\sin\mu - \frac{V_y}{V_h}\cos\mu\sin\phi + \frac{V_x V_z}{V_h V}\cos\mu\cos\phi \tag{134}$$

$$n_2 = \frac{V_y}{V}\sin\mu + \frac{V_x}{V_h}\cos\mu\sin\phi + \frac{V_y V_z}{V_h V}\cos\mu\cos\phi \tag{135}$$

$$n_3 = \frac{V_z}{V}\sin\mu - \frac{V_h}{V}\cos\mu\cos\phi \tag{136}$$

where

$$\mu = \quad \text{Mach angle, } \sin^{-1}(1/M)$$

$$\phi = \quad \text{Aircraft azimuth angle.}$$

The components of the three-dimensional velocity vector (V_x, V_y, V_z), the projection V_h of the velocity vector onto a horizontal plane, and the total velocity magnitude are given by

$$V_x = a_0 M \cos\gamma \sin\psi \tag{137}$$

$$V_y = a_0 M \cos\gamma \cos\psi \tag{138}$$

$$V_z = a_0 M \sin\gamma \tag{139}$$

$$V_h = \sqrt{V_x^2 + V_y^2} \tag{140}$$

$$V = \sqrt{V_h^2 + V_z^2} \tag{141}$$

where

$\psi =$ the aircraft heading

$\gamma =$ aircraft climb angle.

The starting ray is then recast from local coordinates to an Earth-fixed coordinate system. From that starting point, ray tracing is done following the method described by Schulten [87]. POTRAY uses the same predictor/corrector algorithm as PCBoom for ray tracing. The ray position is governed by Equation (85) with wave normal vector described by Equation (86); see also Equations (8) and (7) in Schulten [87]. The numerical implementation in POTRAY follows this procedure:

1. Predict ray position at point $i + 1$ using Equation (87), which is a first-order approximation to Equation (85).
2. Calculate atmospheric parameters at predicted point
3. Average the atmospheric parameters at points i and $i + 1$, and re-compute ray point using the average atmospheric parameters (corrector step)
4. Use Equation (88), which is a first-order approximation to Equation (86) to calculate the wave normal vector at point $i + 1$. To do so, derivative quantities of wind and sound speed are estimated using first-order forward finite divided differences.

If a ray propagation step crosses the ground elevation (i.e., goes underground), linear interpolation of the ray path altitude is used to determine the ground point and reflect the ray at the interpolated ground point. That time step is split based on the fraction

$$F_z = \frac{z_g - z_{i-1}}{z_i - z_{i-1}} \tag{142}$$

where

$z_g =$ The ground elevation.

That fractional step is also used to linearly interpolate longitude and latitude to determine coordinates of the ground bounce point. Calculation of the reflected wave normal vector $\vec{q}_r$ is done using

$$\vec{q}_r = \vec{q} - 2(\vec{q} \cdot \hat{g})\hat{g} \tag{143}$$

where

$\hat{g} =$ Ground normal unit vector.

The ray tracing algorithm in POTRAY proceeds until one of the following conditions is met:

1. the ray has reflected the user-specified maximum number of times from the ground, then passes the ground and reaches the below-the-ground limit,
2. the ray passes out of the atmosphere based on atmosphere file specified altitude limit,
3. the ray is stuck in a channel, or
4. the dimensional limit is reached (40,003), which is the maximum number of steps POTRAY will trace.

If the number of times the ray path passes the flight altitude while traveling downward is greater than the bounce count plus two, the ray is determined to be stuck in a channel. As a final step, ray paths and isopemps are written to a file for use with POTTI.

6.3.2 POTTI

The POTTI module is an interactive display program that achieves the visualization of secondary sonic boom carpet locations computed in POTRAY and allows selection of individual rays for further examination of the sonic boom signature.

6.3.2.1 Summary of Inputs

The POTTI module reads outputs from POTRAY (.hit, .ray) that contain isopemp data, ray propagation information, and summary information for a single point. If POTRAY was run in trajectory mode, an index file with extension .fil will be created. POTTI can read the index to plot isopemps from successive trajectory points.

6.3.2.2 Calculations

POTTI is largely a file parsing and plotting tool, making use of calculations done by POTRAY. The header line from .hit files contains azimuth angle range and angle increment used by POTRAY. POTTI uses that information to initialize arrays of ground intersection coordinates. The actual coordinates as well as ground arrival times are then read from data blocks in the .hit file. Rays which do not reach the ground have latitude, longitude coordinates set to values of 999.

The end result of POTRAY/POTTI analysis is identification by the user of trajectory times and azimuths that produce secondary boom rays of interest.

6.3.3 OTTER

The OTTER module propagates boom signatures along the rays after running PCBoom using the OTT keyword. The OTTER module advances and ages the signature and visualizes the boom waveform along the secondary ray path using a lossless Burgers solver similar to the one in FOBoom. The full ray paths and initial *F*-functions (derived from the user-provided source characteristics) are computed by FOBoom (triggered by keyword OTT) and written to the appropriate OTTER input files.

6.3.3.1 Summary of Inputs

The OTTER module reads .ott and .ots output files from FOBoom. The .ott file contains ray coordinates, atmospheric properties, and amplitude and age parameters at every time step along each ray. It is similar to the .age file associated with BURGERS mode, but contains more quantities. The independent variable is time along the ray, rather than altitude. The .ots file contains the *F*-function at the start of each ray. It is the same as the .ssg file associated with BURGERS mode, but is given a different type to avoid confusion if both types of runs are done for the same base case.

6.3.3.2 Calculations

Inputs to OTTER include amplitude and age parameters from ray starting points up to endpoint conditions described in Section 6.3.1.2. The interactive OTTER display allows the user to position a cursor at a selected point. Internally, OTTER models signature evolution from the starting point up to the user-selected point using parameter values from the input files. Basic signature aging in OTTER follows the Middleton–Carlson–Hayes method, which is based on the area balance technique described by Middleton and Carlson [30], reiterated and formalized by Hayes, Haefeli and Kulsrud [29]. That method is the same as is implemented in FOBoom (see Section 4.2.2) with the theoretical underpinning described in Section 3.3.1. If no focal points are encountered between the starting point and the user-selected point, output of the Middleton–Carlson–Hayes method is scaled to dimensional pressure via multiplication with the Blok parameter.

If focal zone(s) are encountered, OTTER models focus and post-focus signatures in a manner generally similar to PCBoom. In that instance, the signature evolution algorithm follows these steps:

1. Determination of focal zone edges is done using equivalent N-wave shock strength, in the same manner as FOBoom. That methodology is described in more detail in Section 4.2.3.1.3.
2. Signature aging from the starting point up to the edge of the focal zone is done using the Middleton–Carlson–Hayes method.
3. The last geometric signature from the previous step is used as input to the Plotkin and Cantril [2] focus signature algorithm described in Section 4.2.3. Two differences from the FOBoom focal point algorithm are that FOBoom can handle high-resolution input signatures (the focal point modeling routine used by OTTER is currently limited to signatures with 200 points), and OTTER does not include the logic necessary to handle overlapping shocks in the input signature. The maximum focus signature is generated in this step.
4. Within a distance along the ray of $10X_*$ (see Equation (144)) ahead of the focus, the last geometric signature is linearly blended with the maximum focus signature and scaled so that $p_{\max}$ matches the maximum overpressure of the signature which came out of the Middleton–Carlson–Hayes algorithm. Distance along the ray is given by

$$X_* = \sqrt{2RY_*} \tag{144}$$

where

$$R = \quad \text{Caustic curvature}$$
$$Y_* = \quad \text{dimension of focal zone normal to ray, see also Equation (69).}$$

5. The postfocus edge of the focal zone is defined to be symmetric with the start of the focal zone. The nominal postfocus signature p_{post} is calculated from the maximum focus signature p_{foc} and the last geometric signature p_{LGS} using

$$p_{\text{post}} = p_{\text{foc}} - \frac{1}{3.78} \frac{\max(p_{\text{foc}})}{\max(p_{\text{LGS}})} p_{\text{LGS}}. \tag{145}$$

The concept behind step 5 follows the method described in Section 4.2.3.1. FOBoom constructs the maximum focus boom as a mix of the approaching boom and the analytical solution of Gill–Seebass (GS) [88], applying the scaling described by Plotkin and Cantril [2] in which GS is a spike of 2.78 times the geometric shock amplitude, plus the original shock. The maximum focus boom is thus GS plus $1/(1 + 2.78)$ times the approaching boom strength. At the edge of the focal zone, the nominal post-focus signature is the GS part of the maximum focus boom, which can be obtained by subtracting a signature with amplitude $1/3.78$ times the approaching boom from the maximum focus boom.

6. The maximum focus signature is linearly blended through the second half of the focal zone with the postfocus signature.
7. Beyond the focal zone, the postfocus signature is aged using the Middleton–Carlson–Hayes method.

7   Additional Processing Options

The PCBoom modules can be run either through the command line or in a batch script. The SonicBAT-Run-Fort module offers the following additional batch processing options.

7.1   SonicBAT-Run-Fort

The SonicBAT-Run-Fort (SBRF) module is a batch processing tool that can sequence multiple modules (such as FOBoom, HeadlessBurgers, and Turbo), compile results, and prepare files for further processing in additional PCBoom modules such as WCON. While these individual tools can be run separately by the user for specific rays and trajectory points, it is sometimes advantageous to examine a full footprint for multiple trajectory points and azimuthal angles in concert. SBRF orchestrates this process by running the tools as needed and assembling FOBoom .out files for lossy propagation and turbulence calculations.

Note that a propagation tool called sBOOM[4], which uses a time-domain Burgers solution algorithm, can be used within SBRF as the primary propagation module in place of FOBoom and HeadlessBurgers. sBOOM uses a time-domain Burgers solution algorithm, which is faster than PCBoom's Anderson algorithm. However, sBOOM is distributed separately from the PCBoom suite. sBOOM version 2.01 and version 2.84 are compatible with PCBoom version 7.1.

7.1.1 Summary of Inputs

The inputs required for the SBRF module are the same as the files that are required when running each module individually, plus a .txt file to specify which modules will be run, and a configuration file. See the PCBoom User's Guide [8] for a complete list of inputs.

7.1.2 Calculations

Within SBRF, no mathematical calculations are made. It is principally a method for utilizing user prescribed instructions, running the FOBoom modules in an orchestrated manner, and creating suitable output for other downstream postprocessors.

---

[4] sBOOM is available upon request from the NASA Software catalog, https://software.nasa.gov/software/LAR-18012-1

Appendix A Weighted angle cubic interpolation

PCBoom uses a common subroutine to interpolate pressure signatures in some scenarios. Specifically, this subroutine is used in FOBoom when CYLINDER or CYLINDERHIRES modes are used, and in Turbo. This subroutine uses cubic interpolation (quadratic at endpoints), and derivatives are calculated by weighting line segment angles instead of slopes. This subroutine interpolates an input pressure signature P defined at M stations corresponding to locations X, to a set of N output stations at locations XI. The procedure is as follows:

1. For all line segments connecting discrete pairs of points at $X(i)$ and $X(i + 1)$, the increments dx and dp, the line segment angles θ, and the distances between points L are calculated from

$$dx_i = X_{i+1} - X_i$$
$$dp_i = P_{i+1} - P_i$$
$$\theta_i = \tan\left(\frac{dp_i}{dx_j}\right)$$
$$L_i = \sqrt{dx_i^2 + dp_i^2}$$

2. Weighted points $V1$ are calculated as

$$V1_i = \frac{L_i\theta_{i-1} + L_{i-1}\theta_i}{L_i + L_{i-1}}$$

3. Interpolation constants are calculated as

$$A_i = \frac{dx_i(\tan V1_{i+1} + \tan V1_i) - 2(dp_i)}{dx_i^3}$$
$$B_i = \frac{3(dp_i) - dx_i(\tan V1_{i+1} + 2\tan V1_i)}{dx_i^2}$$
$$C_i = \tan V1_i$$
$$D_i = P_i$$

Endpoints are handled differently using

$$A_1 = 0 \qquad\qquad\qquad A_{M-1} = 0$$
$$B_1 = \frac{P_1 - P_2 + dx_1\tan V1_2}{dx_1^2} \qquad B_{M-1} = \frac{P_M - P_{M-1} + dx_{M-1}\tan V1_{M-1}}{dx_{M-1}^2}$$
$$C_1 = \frac{2(P_2 - P_1) - dx_1\tan V1_2}{dx_1} \qquad C_{M-1} = \tan V1_{M-1}$$
$$D_1 = P_1 \qquad\qquad\qquad D_{M-1} = P_{M-1}, D_M = P_M$$

4. Finally, the input function is interpolated to points XI_j. In each region for which $XI_j \le X_i$, interpolated pressure signature values PI are calculated as

$$PI_j = A_{i-1}(XI_j - X_{i-1})^3 + B_{i-1}(XI_j - X_{i-1})^2 + C_{i-1}(XI_j - X_{i-1}) + D_{i-1}$$

Appendix B Glossary
A select group of terms used in this reference are described below.

ABL	Atmospheric Boundary Layer
ABLH	Atmospheric Boundary-Layer Height
AGL	Above Ground Level
carpet	The area on the ground exposed to the direct downward propagating sonic boom rays from the aircraft, excluding upward propagating and secondary booms. The width of the carpet is dependent on aircraft trajectory, and the atmospheric properties.
CFD	Computational Fluid Dynamics
.dat	FOBoom input file
dB	Decibel
EFG	Earth-Fixed Geocentric
FiltView	Finite impulse response filter viewer
FIR	Finite impulse response
FOBoom	Main propagation code in the PCBoom suite, which includes focused boom capability
Focused Boom	When isopemps overlap in time and space, their intersection represents a position where boom energy is concentrated, or focused, and a "superboom" exists
Footprint	The area on the ground exposed to the sonic boom
HeadlessBurgers	Legacy Burgers equation propagation module without a graphical user interface
Hz	Hertz
ISBAP	Indoor Sonic Boom Annoyance Predictor
KZK	Khokhlov-Zabolotskaya-Kuznetsov
KZKFourier	Khokhlov-Zabolotskaya-Kuznetsov Fourier
L	Aircraft Length, in ft
LNTE	Lossy Nonlinear Tricomi Equation
Lpk or Peak Level	Peak sound pressure level, in dB
ML	Model Length, in ft
Molecular relaxation	In PCBoom, molecular relaxation refers to the absorptive effects of oxygen and nitrogen molecular relaxation (based on relative humidity aloft) on sonic boom propagation
MSL	Mean Sea Level
NPE	Nonlinear Progressive Wave Equation
OTT	Over the top
OTTER	Over-the-Topper ray plotting program
PCBoom	Personal Computer sonic Boom program
PCBurg	PCBoom legacy Burgers equation module with graphical interface
PCBFoot	PCBoom Footprint assembly module
Phi, ϕ	Ray angle relative to the aircraft fixed coordinate system, also referred to as Azimuth angle, in degrees. 0 degrees is directly below the aircraft.
PL or PLdB	Stevens Mark VII Perceived Level, in dB

Pmax	The maximum peak overpressure in the signature. It is presented in units of pounds per square foot (psf).
POTRAY	Propagate Over-the-Top Rays module
POTTI	Plot Over-the-Top Isopemps interactive visualizer module
RayCau	Caustic Ray visualization module
RMS	Root mean square
SELA or ASEL	A-Weighted Sound Exposure Level, in dB(A)
SELB or BSEL	B-Weighted Sound Exposure Level, in dB(B)
SELC or CSEL	C-Weighted Sound Exposure Level, in dB(C)
SELD or DSEL	D-Weighted Sound Exposure Level, in dB(D)
SELE or ESEL	E-Weighted Sound Exposure Level, in dB(E)
SELZ or ZSEL	Z-Weighted (flat) Sound Exposure Level, in dB
Signature	An aircraft traveling at supersonic speed generates a disturbance in the form of shock waves. Near the aircraft, the shocks form a detailed pressure signature, which relates to the detail of the aircraft's geometry. As it propagates through the atmosphere, the signature evolves.
SonicBAT-Run-Fort (SBRF)	Tool developed during the SonicBAT program that automatically calls and runs other Fortran programs (e.g., HeadlessBurgers, sBOOM, Turbo) and assembles their outputs suitable for use with other PCBoom modules such as WCON and FiltView
Turbo	Classical turbulence modeling module employing linear acoustic re-propagation through atmospheres with turbulent temperature and gust variations
Waveform	See "Signature". "Waveform" and "signature" are used interchangeably throughout this guide.
WCON	Interactive footprint, contours, and signature display module

Appendix C References

[1]   C. L. Thomas, "Extrapolation of Sonic Boom Pressure Signatures by the Waveform Parameter Method," NASA TN D-6832, June 1972.

[2]   K. J. Plotkin and J. M. Cantril, "Prediction of Sonic Boom at a Focus," Wyle Laboratories Research Report WR 75-7., Also AIAA Paper 76-2, January 1976., https://doi.org/10.2514/6.1976-2.

[3]   K. J. Plotkin, "PCBoom3 Sonic Boom Prediction Model, Version 1.0e," Wyle Research Report WR 95-22E, October 1998.

[4]   K. J. Plotkin and F. Grandi, "Computer Models for Sonic Boom Analysis: PCBoom4, CABoom, BooMap, CORBoom," Wyle Research Report WR 02-11, June 2002.

[5]   K. J. Plotkin, J. A. Page and E. A. Haering Jr., "Extension of PCBoom to Over-The-Top Booms, Ellipsoidal Earth, and Full 3-D Ray Tracing," AIAA Paper 2007-3677, May 2007. https://doi.org/10.2514/6.2007-3677.

[6]   J. B. Lonzaga, "Recent Enhancements to NASA's PCBoom Sonic Boom Propagation Code," AIAA Paper 2019-3386, AIAA Aviation 2019 Forum. https://doi.org/10.2514/6.2019-3386, 2019.

[7]   D. J. Maglieri, P. J. Bobbitt, K. J. Plotkin, K. P. Shepherd, P. G. Coen and D. M. Richwine, "Sonic Boom: Six Decades of Research," NASA SP-2014-622, December 2014.

[8]   J. A. Page, J. B. Lonzaga, M. J. Shumway, S. R. Kaye, R. S. Downs, A. Loubeau and W. J. Doebler, "PCBoom Version 7.1 User's Guide," NASA/TM–20205007703, October 2020.

[9]   H. W. Carlson, "Simplified Sonic Boom Prediction," NASA Technical Paper 1122, 1978.

[10]   D. J. Maglieri and K. J. Plotkin, "Chapter 10: Sonic Boom," in *Aeroacoustics of Flight Vehicle: Theory and Practice, Volume 1: Noise Sources*, NASA Reference Publication 1258, H.H. Hubbard, ed., August 1991.

[11]   H. W. Carlson and D. J. Maglieri, "Review of Sonic-Boom Generation Theory and Prediction Methods," *The Journal of the Acoustical Society of America,* vol. 51, no. 2, pp. 675-685, 1972. https://doi.org/10.1121/1.1912901.

[12]   K. J. Plotkin, "Review of Sonic Boom Theory," AIAA Paper 89-1105, April 1989. https://doi.org/10.2514/6.1989-1105.

[13]   G. B. Whitham, "The Flow Pattern of a Supersonic Projectile," *Communications on Pure and Applied Mathematics,* vol. 5, no. 3, pp. 301-348, 1952. https://doi.org/10.1002/cpa.3160050305.

[14] F. Walkden, "The Shock Pattern of a Wing-Body Combination, Far From the Flight Path," *Aeronautical Quarterly,* vol. 9, no. 2, pp. 164-194, 1958. https://doi.org/10.1017/S0001925900001372.

[15] G. B. Whitham, "On the Propagation of Weak Shock Waves," *Journal of Fluid Mechanics,* vol. 1, no. 3, pp. 290-318, 1956. https://doi.org/10.1017/S0022112056000172.

[16] A. R. Seebass, "Sonic Boom Theory," *Journal of Aircraft,* vol. 6, no. 3, pp. 177-184, May-June 1969. https://doi.org/10.2514/3.44032.

[17] K. J. Plotkin and L. C. Sutherland, "Sonic Boom Considerations for High-Speed Civil Transport Aircraft," Wyle Research Report WR 87-4, June 1987.

[18] C. L. Thomas, "Extrapolation of Wind-Tunnel Sonic Boom Signatures Without Use of a Whitham F Function," in *NASA SP-255, Third Conference on Sonic Boom Research*, Schwartz, I.R., (Ed.), January 1971.

[19] B. Tiegerman, "Sonic Booms of Drag-Dominated Hypersonic Vehicles," Ph.D. Dissertation, Cornell University, Ithaca, NY, August 1975.

[20] P. O. Jarvinen and J. A. F. Hill, "Universal model for Underexpanded Rocket Plumes in Hypersonic Flow," Proceedings of the 12th JANNAF Liquid Meeting, Las Vegas, Nevada, 1970.

[21] D. I. Blokhintzev, "The Propagation of Sound in an Inhomogeneous and Moving Medium I, Acoustics of a Nonhomogeneous Moving Medium (translated as NACA TM-1399)," *The Journal of the Acoustical Society of America,* vol. 18, no. 2, pp. 322-328, 1946.

[22] A. Pierce, Acoustics: An Introduction to Its Physical Principles and Applications, Melville, NY: Acoustical Society of America, 1994.

[23] I. Sofair, "Improved Method for Calculating Exact Geodetic Latitude and Altitude," *Journal of Guidance, Control, and Dynamics,* vol. 20, no. 4, pp. 824-826, 1997. https://doi.org/10.2514/2.4119.

[24] I. Sofair, "Improved Method for Calculating Exact Geodetic Latitude and Altitude Revisited," *Journal of Guidance, Control, and Dynamics,* vol. 23, no. 2, p. 369, 2000. https://doi.org/10.2514/2.4534.

[25] E. J. Rickley and A. D. Pierce, "Detection and Assessment of Secondary Sonic Booms in New England," FAA-AEE-80-22, May 1980.

[26] L. D. Robinson, "Sonic Boom Propagation through an Inhomogeneous, Windy Atmosphere," Ph.D. Dissertation, University of Texas, Austin, TX, 1991.

[27]   A. D. Taylor, "The TRAPS Sonic Boom Program," NOAA Technical Memorandum ERL ARL-87, July 1980.

[28]   O. Kandil and N. Khasdeo, "Parametric Investigation of Sonic Boom Focusing using Solution of Nonlinear Tricomi Equation," AIAA Paper 2006-0415, 2006. https://doi.org/10.2514/6.2006-415.

[29]   W. D. Hayes, R. C. Haefeli and H. E. Kulsrud, "Sonic Boom Propagation in a Stratified Atmosphere, With Computer Program," NASA CR-1299, April 1969.

[30]   W. D. Middleton and H. W. Carlson, "A Numerical Method for Calculating Near-Field Sonic-Boom Pressure Signatures," NASA TN D-3082, November 1965.

[31]   G. I. Taylor, "The Conditions Necessary for Discontinuous Motion in Gases," *Proceedings of the Royal Society London A,* vol. 84, no. 571, pp. 371-377, October 1910. https://doi.org/10.1098/rspa.1910.0081.

[32]   H. Bateman, "Some Recent Researches on the Motion of Fluids," *Monthly Weather Review,* vol. 43, no. 4, pp. 163-170, 1915. https://doi.org/10.1175/1520-0493(1915)43<163:SRROTM>2.0.CO;2.

[33]   J. M. Burgers, "Mathematical Examples Illustrating Relations Occurring in the Theory of Turbulent Fluid Motion," *Transactions of the Royal Netherlands Academy of Sciences,* vol. 17, pp. 1-53, 1939.

[34]   M. F. Hamilton and D. T. Blackstock, Nonlinear Acoustics, Academic Press, 1998.

[35]   E. Hopf, "The Partial Differential Equation ut + uux = μxx," *Communications on Pure and Applied Mathematics,* vol. 3, no. 3, pp. 201-230, 1950. https://doi.org/10.1002/cpa.3160030302.

[36]   J. D. Cole, "On a Quasi-linear Parabolic Equation Occurring in Aerodynamics," *Quarterly of Applied Mathematics,* vol. 9, no. 3, pp. 225-236, 1951. https://doi.org/10.1090/qam/42889.

[37]   K. J. Plotkin and A. R. George, "Propagation of Weak Shock Waves Through Turbulence," *Journal of Fluid Mechanics,* vol. 54, no. 3, pp. 449-467, 1972. https://doi.org/10.1017/S0022112072000795.

[38]   American National Standards Institute, "Methods for Calculation of the Absorption of Sound by the Atmosphere," ANSI/ASA S1.26, 2014.

[39]   J. A. Page, K. J. Plotkin and C. Wilmer, "PCBoom Version 6.6 Technical Reference and User Manual," Wyle Laboratories Inc, Wyle Research Report, WR 10-10, 2010.

[40] J. Kang, "Nonlinear Acoustic Propagation of Shock Waves Through the Atmosphere with Molecular Relaxation," Ph.D. Dissertation, The Pennsylvania State University, State College, PA, May 1991.

[41] T. A. Gionfriddo, "Quantification of Sonic Boom Signature Distortions from Propagation Through Atmospheric Turbulence," MS Thesis, The Pennsylvania State University, State College, PA, December 1992.

[42] S. C. Crow, "Distortion of Sonic Bangs by Atmospheric Turbulence," *Journal of Fluid Mechanics,* vol. 37, no. 3, pp. 529-563, 1969. https://doi.org/10.1017/S0022112069000711.

[43] K. J. Plotkin, D. J. Maglieri and B. M. Sullivan, "Measured Effects of Turbulence on the Loudness and Waveforms of Conventional and Shaped Minimized Sonic Booms," AIAA Paper 2005-2949, May 2005. https://doi.org/10.2514/6.2005-2949.

[44] G. Kamali and A. D. Pierce, "Time Dependence of Variances of Sonic Boom Waveforms," *Nature,* vol. 234, pp. 30-31, November 1971. https://doi.org/10.1038/234030a0.

[45] K. A. Bradley, C. M. Hobbs, C. B. Wilmer, V. W. Sparrow, T. A. Stout, J. M. Morgenstern, K. H. Underwood, D. J. Maglieri, R. A. Cowart, M. T. Collmar, H. Shen and P. Blanc-Benon, "Sonic Booms in Atmospheric Turbulence (SonicBAT): The Influence of Turbulence on Shaped Sonic Booms," NASA/CR-2020-220509, 2020.

[46] D. A. Hilton, V. Huckel and D. J. Maglieri, "Sonic Boom Measurements during Bomber Training Operations in the Chicago Area," NASA TN D-3655, 1966.

[47] K. J. Plotkin, E. A. Haering Jr., J. E. Murray, D. J. Maglieri, J. Salamone, B. M. Sullivan and D. Schein, "Ground Data Collection of Shaped Sonic Boom Experiment Aircraft Pressure Signatures," AIAA Paper 2005-0010, January 2005. https://doi.org/10.2514/6.2005-10.

[48] V. I. Tatarskii, "Propagation of Waves through Turbulence," Dover, New York, 1961.

[49] K. J. Plotkin, "The Effect of Turbulence on the Loudness of Minimized Sonic Boom Signatures," NASA High-Speed Research Sonic Boom Workshop, NASA CP-3172, May 1992.

[50] L. L. Locey, V. W. Sparrow and A. A. Piacsek, "Sonic Boom Post Processing to Include Atmospheric Turbulence Effects," AIAA Paper 2008-3035, May 2008. https://doi.org/10.2514/6.2008-3035.

[51] L. L. Locey, "Sonic Boom Postprocessing Functions to Simulate Atmospheric Turbulence Effects," Ph.D. Dissertation, The Pennsylvania State University, State College, PA, 2008.

[52] A. A. Piacsek, "Atmospheric Turbulence Conditions Leading to Focused and Folded Sonic Boom Wave Fronts," *The Journal of the Acoustical Society of America,* vol. 111, no. 1, pp. 520-529, 2002. https://doi.org/10.1121/1.1377631.

[53] L. L. Locey and V. W. Sparrow, "Modeling Atmospheric Turbulence as a Filter for Sonic Boom Propagation," *Noise Control Engineering Journal,* vol. 55, pp. 495-503, November 2007. https://doi.org/10.3397/1.2817810.

[54] S. Bradley, "Atmospheric Remote Sensing, Chap 2," CRC Press, Boca Raton, FL, 2008.

[55] P. Blanc-Benon, B. Lipkens, L. Dallois, M. F. Hamilton and D. T. Blackstock, "Propagation of Finite Amplitude Sound through Turbulence: Modeling with Geometrical Acoustics and the Parabolic Approximation," *The Journal of the Acoustical Society of America,* vol. 111, no. 1, pp. 487-498, 2002. https://doi.org/10.1121/1.1404378.

[56] T. A. Stout, "Simulation of N-Wave and Shaped Supersonic Signature Turbulent Variations," Ph.D. Dissertation, The Pennsylvania State University, State College, PA, 2018.

[57] P. Chevret, P. Blanc-Benon and D. Juvé, "A Numerical Model for Sound Propagation through a Turbulent Atmosphere near the Ground," *The Journal of Acoustical Society of America,* vol. 100, no. 6, pp. 3587-3599, 1996. https://doi.org/10.1121/1.417224.

[58] V. E. Ostashev and D. K. Wilson, Acoustics in Moving Inhomogeneous Media, Boca Raton: CRC Press, September 2015.

[59] D. K. Wilson, "A Turbulence Spectral Model for Sound Propagation in the Atmosphere that Incorporates Shear and Buoyancy Forcings," *The Journal of the Acoustical Society of America,* vol. 108, no. 5, pp. 2021-2038, November 2000. https://doi.org/10.1121/1.1311779.

[60] R. Biswas, W. Thigpen, R. Hood, R. Ciotti, P. Mehrotra, C. Henze, J. Parks and B. Biegel, "Pleiades: NASA's First Petascale Supercomputer," in *Contemporary High Performance Computing: From Petascale toward Exascale*, edited by J. Vetter, Chapman and Hall, 2013.

[61] R. Biswas and P. Mehrotra, "NASA Advanced Computing Environment for Science and Engineering," in *26th International Conference on Parallel Computational Fluid Dynamics Parallel CFD2014*, 2014.

[62] B. Cotte and P. Blanc-Benon, "Estimates of the Relevant Turbulent Scales for Acoustic Propagation in an Upward Refracting Atmosphere," *Acta Acustica United with Acustica,* vol. 93, pp. 944-958, 2007.

[63] J. A. Page, K. J. Plotkin, C. Hobbs, V. W. Sparrow, J. Salamone, R. Cowart, H. R. Welge, J. Ladd, D. J. Maglieri and A. Piacsek, "Superboom Caustic Analysis and Measurement Program (SCAMP) Final Report," NASA/CR-2015-218871, August 2015.

[64] R. O. Onyeonwu, "A Numerical Study of the Effects of Aircraft Maneuvers on the Focusing of Sonic Booms," UTIAS Report No. 192, November 1973.

[65] J. P. Guiraud, "Acoustique Geometrique, Bruit Ballistique des Avions Supersonique et Focalisation," *J. Mecanique,* vol. 4, pp. 215-267, 1965.

[66] L. J. Cliatt II, M. A. Hill and E. A. Haering Jr., "Mach Cutoff Analysis and Results from NASA's Farfield Investigation of No-boom Thresholds," 22nd AIAA/CEAS Aeroacoustics Conference, AIAA Paper 2016-3011, 2016. https://doi.org/10.2514/6.2016-3011.

[67] A. Preusser, "Algorithm 671: FARB-E-2D: Fill Area with Bicubics on Rectangles - A Contour Plot Program," *ACM Transactions on Mathematical Software,* vol. 15, no. 1, pp. 79-89, March 1989. https://doi.org/10.1145/62038.69651.

[68] J. Rachami, J. A. Page, L. Zhou and B. Kim, "Environmental Modeling of Advanced Vehicles in NextGen," AIAA Paper 2009-6981, 2009. https://doi.org/10.2514/6.2009-6981.

[69] American National Standards Institute, "Template Method for Ground Impedance," S1.18, 1999.

[70] A. L. Hastings, A. S. Rapoza, S. R. Kaye and D. Flynn, "Quieter Pavement Project – Death Valley National Park: Interim Report for One-Year Post-Treatment Conditions," DOT-VNTSC-NPS-20-04, June 2020.

[71] V. W. Sparrow and T. J. Ferguson, "Penetration of Shaped Sonic Boom Noise into a Flat Ocean," AIAA Paper 97-0486, 1997. https://doi.org/10.2514/6.1997-486.

[72] K. N. Sawyers, "Underwater Sound Pressure from Sonic Booms," *The Journal of the Acoustical Society of America,* vol. 44, no. 2, pp. 523-524, 1968. https://doi.org/10.1121/1.1911114.

[73] R. K. Cook, "Penetration of a Sonic Boom into Water," *The Journal of the Acoustical Society of America,* vol. 47, no. 5 pt 2, pp. 1430-1436, 1970. https://doi.org/10.1121/1.1912052.

[74] H. K. Cheng, C. J. Lee, M. M. Hafez and W. H. Guo, "Sonic Boom Propagation and Submarine Impact: A Study of Computational and Theoretical Issues," AIAA Paper 96-0755, 1996. https://doi.org/10.2514/6.1996-755.

[75] J. Garrelick, "The Effect of a Coastline on the Underwater Penetration of Sonic Booms," *The Journal of the Acoustical Society of America,* vol. 111, no. 1 pt 2, pp. 610-613, 2002. https://doi.org/10.1121/1.1414702.

[76] A. Loubeau and J. Page, "Human Perception of Sonic Booms from Supersonic Aircraft," *Acoustics Today,* vol. 14, no. 3, pp. 23-30, 2018.

[77] J. Page and A. Loubeau, "Overall Vehicle System Noise: Sonic Boom," *CEAS Aeronautical Journal,* vol. 10, pp. 335-353, 2019. https://doi.org/10.1007/s13272-019-00379-0.

[78] American National Standards Institute, "Acoustical Terminology," S1.1, 1994.

[79] American National Standards Institute, "Design Response of Weighting Networks for Acoustical Measurements," S1.42, 2001.

[80] S. S. Stevens, "Perceived Level of Noise by Mark VII and Decibels (E)," *The Journal of the Acoustical Society of America,* vol. 51, no. 2, pp. 575-601, 1972. https://doi.org/10.1121/1.1912880.

[81] K. Shepherd and B. M. Sullivan, "A Loudness Calculation Procedure Applied to Shaped Sonic Booms," NASA-TP-3134, 1991.

[82] American National Standards Institute, "Specification for Octave Band and Fractional Octave-Band Analog and Digital Filters," S1.11, 2004.

[83] G. M. Jackson and H. G. Leventhall, "Calculation of the Perceived Level of Noise (PLdB) using Stevens' Method (Mark VII)," *Applied Acoustics,* vol. 6, no. 1, pp. 23-24, 1973. https://doi.org/10.1016/0003-682X(73)90027-3.

[84] C. R. Bolander, D. F. Hunsaker, H. Shen and F. L. Carpenter, "Procedure for the Calculation of the Perceived Loudness of Sonic Booms," AIAA Paper 2019-2091, January 2019. http://doi.org/10.2514/6.2019-2091.

[85] NASA Langley Research Center, "Loudness Code for Asymmetric Sonic Booms, Software Package, Ver. 1.0.0.1, Reference Number LAR-16954-1," Hampton, VA.

[86] A. Loubeau, "Evaluation of the Effect of Aircraft Size on Indoor Annoyance Caused by Sonic Booms," *Journal of the Acoustical Society of America,* vol. 136, pp. 2223-2224, 2014. https://doi.org/10.1121/1.4900073.

[87] J. B. H. M. Schulten, "Computation of Aircraft Noise Propagation through the Atmospheric Boundary Layer," National Aerospace Laboratory NLR TP 97374, December 1997.

[88] P. M. Gill and A. R. Seebass, "Non-Linear Acoustic Behavior at a Caustic: An Approximate Analytical Solution," Aeroacoustics: Fan, STOL, and Boundary Layer Noise; Sonic Boom; Aeroacoustics Instrumentation, 1975. https://doi.org/10.2514/5.9781600865121.0353.0386.

[89] M. Downing, N. Zamot, C. Moss, D. Morin, E. Wolski, S. Chung, K. J. Plotkin and D. J. Maglieri, "Controlled Focused Sonic Booms from Maneuvering Aircraft," *Journal of the Acoustical Society of America,* vol. 104, no. 1, pp. 112-121, July 1998. https://doi.org/10.1121/1.423261.

[90] T. Auger and F. Coulouvrat, "Numerical Simulation of Sonic Boom Focusing," *AIAA Journal,* vol. 40, no. 9, pp. 1726-1734, 2002. https://doi.org/10.2514/2.1877.

[91] J. B. Lonzaga, R. M. Waxler, J. D. Assink and J. D. Talmadge, "Modelling Waveforms of Infrasound Arrivals from Impulsive Sources Using Weakly Non-Linear Ray Theory," *Geophysical Journal International,* vol. 200, no. 3, pp. 1347-1361, 2015. https://doi.org/10.1093/gji/ggu479.

[92] W. D. Hayes and H. L. Runyan Jr, "Sonic-Boom Propagation through a Stratified Atmosphere," *The Journal of the Acoustical Society of America,* vol. 51, no. 2C, pp. 695-701, 1972. https://doi.org/10.1121/1.1912903.

[93] G. B. Whitham, "The Propagation of Weak Spherical Shocks in Stars," *Communications on Pure and Applied Mathematics,* vol. 6, no. 3, pp. 397-414, 1953. https://doi.org/10.1002/cpa.3160060305.

[94] H. E. Bass, L. C. Sutherland, A. J. Zuckerwar, D. T. Blackstock and D. M. Hester, "Atmospheric absorption of sound: Further developments," *The Journal of the Acoustical Society of America,* vol. 97, no. 1, pp. 680-683, 1995. https://doi.org/10.1121/1.412989.

[95] R. O. Cleveland, M. F. Hamilton and D. T. Blackstock, "Time-Domain Modeling of Finite-Amplitude Sound in Relaxing Fluids," *The Journal of the Acoustical Society of America,* vol. 99, no. 6, pp. 3312-3318, 1996. https://doi.org/10.1121/1.414983.

[96] L. M. Brekhovskikh and O. A. Godin, Acoustics of Layered Media II: Point Sources and Bounded Beams, Berlin Heidelberg: Springer Science & Business Media, 1999. https://doi.org/10.1007/978-3-662-03889-5.

[97] J. D. Anderson Jr, Introduction to Flight, New York: McGraw Hill, 2016.

[98] R. S. Downs, K. S and J. Page, "PCBoom Propagation Modeling for the Third AIAA Sonic Boom Prediction Workshop," *Journal of Aircraft, Advance online publication,* pp. 1-12, https://doi.org/10.2514/1.C036434.

[99] G. Carrier, P. Malbéqui and P.-E. Normand, "Analysis and Comparison of the Results of Two Ray Tracing-Based Sonic Boom Propagation Codes Applied to the SBPW3 Test Cases," AIAA Paper 2021-0231, 2021. https://doi.org/10.2514/6.2021-0231.

[100] M. D. Anderson, "The Propagation of a Spherical N-Wave in an Absorbing Medium and its Diffraction by a Circular Aperture," Applied Research Laboratories Technical Report ARL-74-25, The University of Texas at Austin, August 1974.

[101] D. Umbach and K. N. Jones, "A Few Methods for Fitting Circles to Data," *IEEE Transactions on Instrumentation and Measurement,* vol. 52, no. 6, pp. 1881-1885, 2003. https://doi.org/10.1109/TIM.2003.820472.

[102] J. A. Salamone, "Solution of the Lossy Nonlinear Tricomi Equation With Application to Sonic Boom Focusing," Ph.D. Dissertation, The Pennsylvania State University, State College, PA, 2013.

[103] S. K. Rallabhandi, "Propagation Analysis Using sBOOM for the Third AIAA Sonic Boom Prediction Workshop," *Journal of Aircraft, Advance online publication,* pp. 1-11, https://arc.aiaa.org/doi/full/10.2514/1.C036325.