INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA

UMI®
800-521-0600

Downloaded from SHAREOK. Persistent Handle URL: https://hdl.handle.net/11244/5906

University of Oklahoma

Graduate College

The Sobolev Approximation in a Spherical Shell Expanding With Constant Radial Velocity

A Dissertation Submitted to the Graduate Faculty

In Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

Thomas E. Vaughan

Norman, Oklahoma

2000

UMI Number: 9957003

UMI®

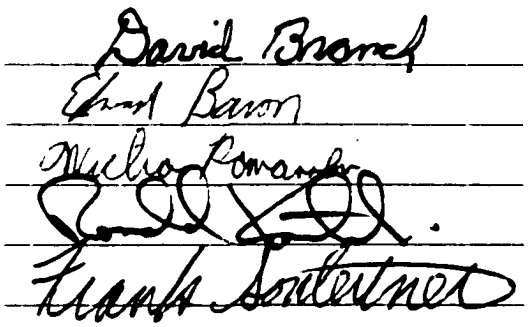
UMI Microform 9957003
Copyright 2000 by Bell & Howell Information and Learning Company.
All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

Bell & Howell Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

The Sobolev Approximation in a Spherical Shell Expanding With Constant Radial Velocity

A Dissertation Approved for the
Department of Physics and Astronomy

BY



Copyright © by THOMAS E. VAUGHAN

All rights reserved.

Contents

1 Introduction 1
- 1.1 Classification of Supernovae . . . 2
- 1.2 Motivation for a Simple Model . . . 8

2 Beam–Line Resonance in an Atmosphere with Power-Law Expansion 11
- 2.1 General Description of the Model . . . 11
  - 2.1.1 Power-Law Expansion . . . 12
  - 2.1.2 Line Opacity and Resonance . . . 12
- 2.2 An Unbounded Atmosphere . . . 13
  - 2.2.1 Impact-Parameter Representation of the Beam . . . 13
  - 2.2.2 Common-Resonance-Point Representation . . . 19
  - 2.2.3 Optical Depth and the Resonance Region for $a = 0$ . . . 27
- 2.3 A Bounded Atmosphere with $a = 0$ . . . 48

3 Sobolev Method for the Constant-Speed Radial Wind 66
- 3.1 Integration of the Transfer Equation . . . 67
- 3.2 Source Function for a Two-Level Atom . . . 70
- 3.3 The Line Profile . . . 72
  - 3.3.1 Geometry . . . 72
  - 3.3.2 Pure Scattering . . . 76
  - 3.3.3 Isothermal Shell . . . 83
- 3.4 Instrumental Resolution . . . 84

4 Conclusion 88

A Source Code 92
- A.1 Resonance Region Around Finite-Width Line . . . 92
  - A.1.1 C++ Header (Interface-Description) Files . . . 92
  - A.1.2 C++ Implementation Files for Library (Reusable) Components . 93
  - A.1.3 C++ Main-Program Files . . . 98
- A.2 Observed Profile of Delta-Function Line . . . 101
  - A.2.1 C++ Header (Interface-Description) Files . . . 101

A.2.2 C++ Implementation Files for Library (Reusable) Components . 103
A.2.3 C++ Main-Program Files 109
A.2.4 Perl Scripts 122

Abstract

A simple model of line radiative transfer in a circumstellar shell surrounding a supernova is presented. An examination of optical depth in an atmosphere with power-law expansion and a single source of opacity highlights certain difficulties arising as the power-law index approaches zero. The Sobolev approximation is shown, however, to be applicable to the case of a constant-radial-velocity wind so long as the intrinsic line profile of the opacity source is a Dirac delta function. Sample line profiles for various geometries are presented along with the source code used to generate the profiles. The simple model predicts the shape of spectral features with a characteristic size of about 10–100 km s^{-1} in the observed spectrum of a core-collapse supernova that shows evidence for circumstellar interaction.

Chapter 1

Introduction

I present a model for the formation of narrow features observed in the spectrum of a supernova (SN) of Type IIn. A SN is an explosion that destroys a star or transforms it into something exotic (like a neutron star). This explosion is bright. The visible-light luminosity typically rises to a peak that occurs from about a week to about a month after the beginning of the explosion. The peak optical luminosity ranges from about $10^9 L_\odot$ (for some Type II SN) to about $10^{10} L_\odot$ (for a typical Type Ia SN), where $L_\odot$ is the luminosity of the Sun.

An observation of light from a SN is either *spectroscopic* or *photometric*. A spectroscopic observation is represented as brightness versus wavelength, and the wavelength resolution is high (say, 10 Å). A photometric observation measures the brightness averaged across a well-defined, broad (say, 1000-Å) wavelength range, or passband [Bessell, 1990]. A time sequence of photometric observations in the same passband is known as a *light curve*.

A SN model is typically either static or dynamic. A static model (as of Lentz et al. [1999]) produces a high-resolution synthetic spectrum that can be compared with a spectroscopic observation of a SN. A dynamic model (as of Höflich and Khokhlov [1996])

produces synthetic light curves. The model presented in this dissertation is of the static type, and unlike a static model that aims to fit every feature over the wavelength range of interest, the model under consideration here only addresses certain features thought to be formed reasonably independently of other features in the spectrum.

1.1 Classification of Supernovae

A SN is classified on the basis of its optical spectrum; Filippenko [1997] has provided a recent review of the optical spectra of supernovae (SNe). At its root, the classification system for SNe hinges, from historical precedent, on the presence or absence of spectral features associated with hydrogen [Minkowski, 1941]. A Type I SN (or SN I) shows no spectroscopic evidence for the presence of hydrogen; the spectrum of a SN II, however, does imply the presence of hydrogen.

Only the spectrum of a SN Ia shows a deep absorption feature (near 6150 Å) due to blueshifted lines (laboratory-frame Si II $\lambda\lambda 6347, 6371$) of silicon. In the standard theoretical framework, a SN Ia results from the thermonuclear fusion explosion of a white dwarf star that accumulates matter from a companion star. A white dwarf is supported against its own gravity not by the familiar pressure resulting from the temperature of a gas but rather by electron degeneracy pressure resulting from the close proximity of neighboring electrons in the high-density material of the white dwarf. An object supported against gravity by electron degeneracy pressure has a maximum mass and, more importantly, becomes *smaller in size* as it accumulates mass. The white dwarf, as it accumulates mass, undergoes quasistatic contraction until the atomic nuclei in its constituent material (carbon and oxygen) fuse. The resultant thermonuclear explosion completely unbinds the star and provides the kinetic energy of the ejected material. Conditions near the center of the SN Ia drive thermonuclear reactions to produce ra-

dioactive ^{56}Ni. Away from the center of the explosion, thermonuclear reactions produce a substantial amount of silicon as well as other intermediate-mass elements. Because the SN Ia begins as a compact object, the fractional change in volume per unit time is huge at early times during the explosion, and so the temperature drops rapidly. However, not all of the nuclear potential energy is deposited thermally on the fusion time scale and dissipated in the expansion. The radioctive decay of ^{56}Ni (whose half-life is about a week and whose decay product, ^{56}Co, also radioactively decays with a half-life of more than two months) provides a delayed input of energy that accounts for the great luminosity of the SN Ia.

All of the SN Ib, SN Ic, and SN II are associated with a different explosion model: core-collapse. A massive star (initially greater than about 8–10 solar masses) will, over the course of its evolution, fuse nuclei in its core and in concentric spherical layers surrounding the core. The process must come to an end soon after the star begins to fuse silicon into iron because although fusion of nuclei lighter than iron is an exothermic process, fusion of iron and heavier elements is endothermic. As nuclear burning in the silicon shell surrounding the core produces iron, the mass, density, and temperature of the core grow. Eventually, the temperature becomes so great that iron nuclei are photodissociated, and then electrons begin to disappear from the core; each disappearing electron combines with a proton in a nucleus in order to form a neutron and a neutrino. Because the pressure is provided by electron degeneracy—even at temperatures great enough so that an individual photon can break apart an iron nucleus—this reaction causes the core (now more than a solar mass) to collapse. The collapse continues until the core's density overshoots that of the nucleus of an atom. In what has been the standard model for many years, the core then rebounds to produce a shock wave that drives away the outer portion of the star. The nuclear-density core remains as a neutron star (or possibly collapses to form a black hole). Recently, pervasive evidence

for asymmetry in core-collapse SNe and the difficulty of getting core-bounce models actually to produce a simulated explosion together suggest that some other mechanism, like bipolar jets emanating from the neutron star or black hole, may actually produce the explosion. In any event, the spectrum of a SN Ib—as if the progenitor were, before core collapse, stripped of its outermost hydrogen envelope—shows little or no evidence for hydrogen. Further, the relatively bland spectrum of a SN Ic—as if the progenitor were, before core collapse, stripped of both its hydrogen *and helium* envelopes—shows little or no evidence for hydrogen and helium.

A SN I for which there exists a good optical spectrum is usually labeled "SN Ia", "SN Ib", or "SN Ic", but a SN II—even one with a good optical spectrum—is usually classified spectroscopically as simply "SN II". Figure 1.1 shows optical spectra of SN II 1992H [Clocchiatti et al., 1996]. At maximum light and for at least a few weeks after maximum light, the spectrum of a typical SN II is characterized by wide features (with doppler width of more than 10^4 km s^{-1}). Usually a few *P-Cygni* profiles [Mihalas, 1978], each of which results from the scattering of light off of a particular atomic transition in the ejecta above the region in which the continuum of the spectrum is formed, can be identified. A P-Cygni profile has a blueward absorption trough and a redward emission feature. After the spectrum has been adjusted in order to remove the relative line-of-sight velocity component between the observer and the SN, the emission part of a P-Cygni profile peaks near the rest-frame wavelength of the transition that gives rise to the profile. A P-Cygni profile in a typical SN II is broad enough to imply that the profile forms in the ejecta.

Recently, "SN IIn", a peculiar spectroscopic subclass of SN II, has been defined. A SN IIn has narrow emission lines, probably indicative of slowly expanding circumstellar gas with which the rapidly expanding ejected material interacts. Figure 1.2, copied from a preprint of Leonard et al. [1999], shows an optical spectrum of the SN IIn 1998S;

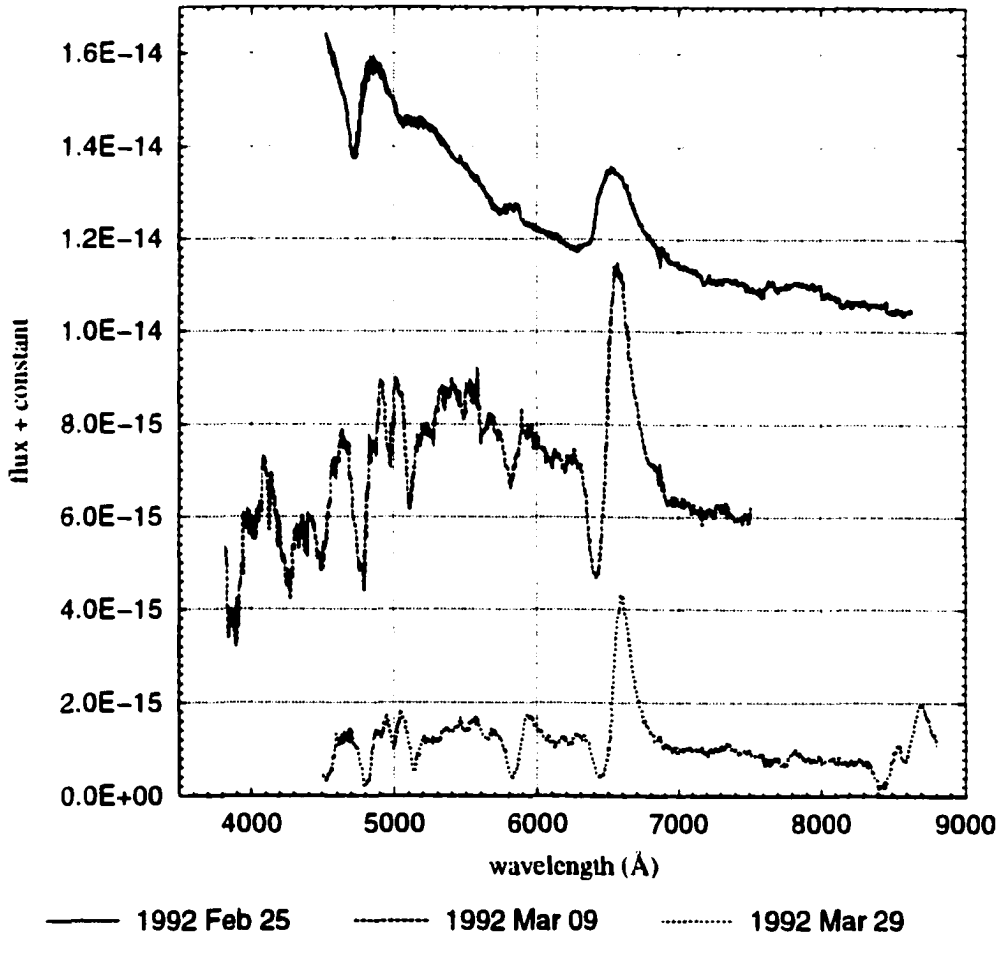


Figure 1.1: Spectra of SN II 1992H starting (at top) about two weeks after maximum light show broad features characteristic of a typical SN II. The time sequence is arranged from top to bottom. Maximum light in the B passband occured around 1992 Feb 13.

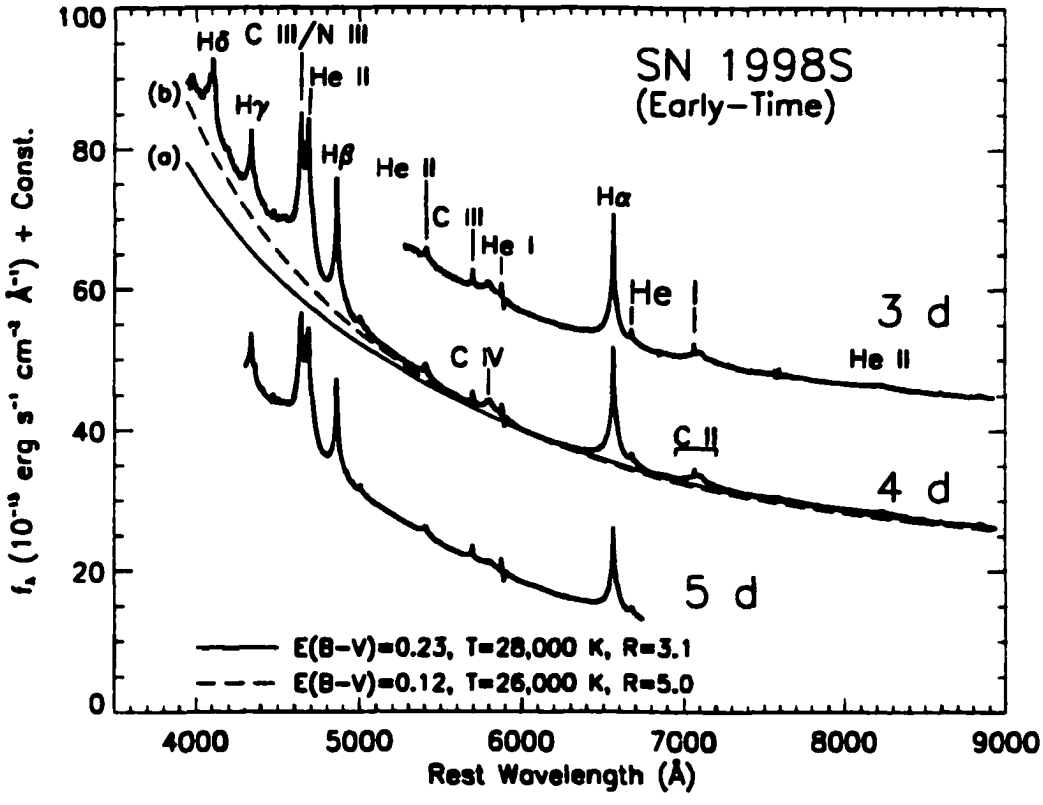


Figure 1.2: Early time spectra, observed three to five days after discovery, of SN IIn 1998S show narrow emission features characteristic of its "IIn" designation.

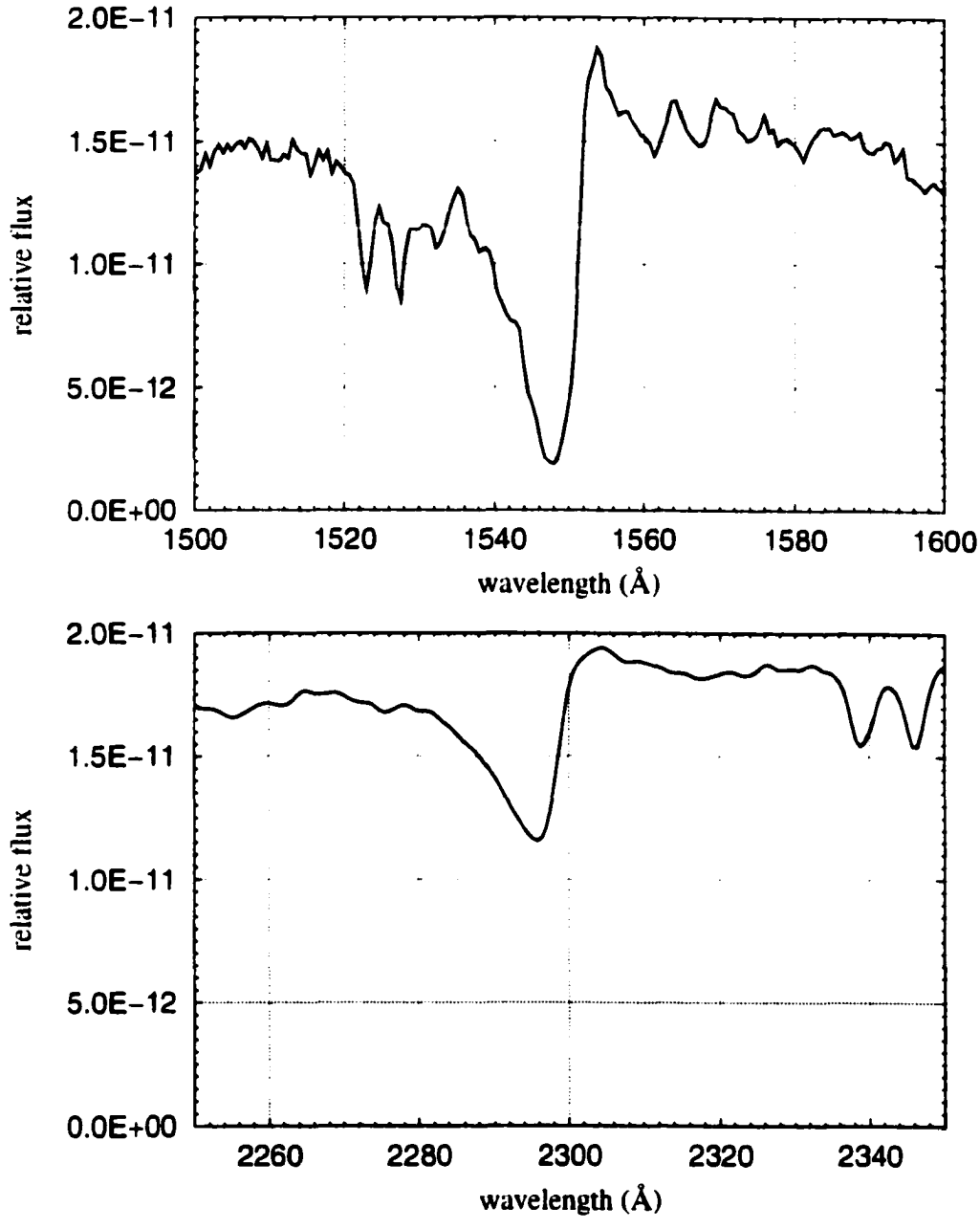


Figure 1.3: The Hubble Space Telescope was used to obtain very high-resolution ultraviolet spectra of SN IIn 1998S. Here are two sections of the same spectrum on 1998 Mar 16, around maximum light.

compare with Figure 1.1. Many an emission feature in the spectra of SN1998S has a component narrow enough to be unresolved in the figure. Particularly interesting, however, are ultraviolet spectra taken by the Hubble Space Telescope. Figure 1.3 shows such a spectrum around maximum light, several days after the spectra in Figure 1.2. Note that the strongest features in Figure 1.3 are as narrow as can be resolved by the detector (which at these wavelengths cannot resolve features due to a radial wind component slower than about 300 km s^{-1}), and there appear to be features unresolved even at this scale. Nevertheless each of the top and bottom plots displays a prominent P-Cygni profile that suggests scattering in an envelope that is expanding around 10^3 km s^{-1} (as compared with the characteristic speed of 10^4 km s^{-1} for the ejecta).

Although the SN II has two photometric subclasses (SN II-L and SN II-P), distinguished by the shape of the optical light curve, the SN IIn exhibits a light curve that does not fit well into the photometric classification scheme. The optical light curve of a SN IIn declines, in comparison with other SNe II, very slowly just after maximum light, probably because of light emitted by the interaction of the ejected material and the circumstellar wind.

1.2 Motivation for a Simple Model

Radio observations of SNe II [Weiler et al., 1992] and of SNe Ib and Ic [Van Dyk et al., 1993] imply that a wind blown off from the progenitor of a core-collapse SN may interact with the ejecta. As indicated in the previous section, recent observations of SN IIn 1998S have revealed narrow spectral features that suggest the presence of a circumstellar wind. High-resolution spectroscopic observations of these features provide the opportunity to test models of line formation in the wind.

A circumstellar wind is accelerated away from a star by radiation pressure. A small

mass element in the wind will, after initial acceleration near the surface of the star, asymptotically approach a terminal radial velocity. The wind expansion speed (up to about 10^3 km s^{-1}) is much smaller than the expansion speed (more than 10^4 km s^{-1}) of the ejecta. Because the width of an observed spectral feature is related by the doppler effect to the speed of the radial expansion, a line formed in the wind will appear much narrower than a line formed in the ejecta. In the simplest wind model, the luminosity and mass-loss rate of the star were constant around the time during which a spherically symmetric wind (now interacting with light from the SN) was initially accelerated, and the wind under consideration exists in a shell that expands at the terminal speed.

Especially for SNe Ia, simple, spherically symmetric models [Branch et al., 1983, Jeffery et al., 1992, Fisher et al., 1997, Millard et al., 1999, Hatano et al., 1999] of line formation in the SN ejecta have taken advantage of homologous expansion, the proportionality between the speed of an ejected mass element and its distance from the center of the explosion. A large, isotropic velocity gradient at every point in the homologously expanding ejecta provides an ideal context for application of the Sobolev [1960] approximation to the mathematical representation of line radiative transfer. If, in a frame of reference at rest with respect to atmospheric material, the atmosphere is considered to be transparent except for a few individual wavelengths (lines) at which the atmosphere is somewhat opaque, then in the Sobolev approximation a monochromatic light beam will interact with the atmosphere only at one location in the atmosphere for each line. A model derived from the Sobolev approximation is very computationally efficient and so allows for rapid identification of the dominant, relatively wide features in a SN spectrum; the spectrum-fitting process gives insight into the physical structure of the SN ejecta. Can such a model be extended or changed in order to give insight into the formation of the very narrow lines that appear to form in a circumstellar shell?

The primary difficulty in adapting to this task a model based on the Sobolev method

is that for a constant-radial-velocity wind, the velocity gradient becomes zero along the radial direction. Nevertheless, I show that the Sobolev method can be used in order to produce synthetic line profiles so long as the intrinsic width of the line is vanishingly small. Although this discussion covers only the case in which the shell has a single line (or multiple lines spaced far enough apart in wavelength space), extension to the case of line blending, in which light scatters off of more than one line before leaving the shell, is straight-forward.

Chapter 2

Beam–Line Resonance in an Atmosphere with Power-Law Expansion

In order numerically to calculate a line profile as seen by a distant observer, one must describe, in terms of the model's geometry, the propagation of a monochromatic light beam through the line-forming region. Only then can a representative sample of beams, that are directed toward the distant observer, be combined in order to produce a synthetic profile.

2.1 General Description of the Model

The model under consideration is a spherical shell of radially expanding gas whose opacity comes only from narrow lines and whose expansion speed is independent of radial coordinate. For the development of the initial material related to the propagation of a beam through an expanding atmosphere, however, I begin by discussing the more

general case of power-law expansion. This provides a context in which one can see the mathematical difficulties introduced by the assumption of a constant-speed wind.

2.1.1 Power-Law Expansion

Consider a spherical shell containing a gaseous medium that is expanding radially and has a radial-power-law velocity field with

$$v_a(r) = \left[\frac{r}{r_0}\right]^a v_0, \tag{2.1}$$

where $v_a(r)$ is the magnitude of the atmosphere's radial velocity at distance r from the center of symmetry; $v_a(r_0) = v_0$; and the power-law index for the velocity field is a. The inner and outer boundaries of the atmosphere must be chosen such that everywhere in the shell $v_a(r) \ll c$, where c is the speed of light; otherwise a time-independent treatment is not valid. Atmospheric expansion speeds must be much smaller than the speed of light because the geometric analysis presented here assumes that the size of the atmosphere and shape of the velocity profile do not change significantly over the time required for a beam to traverse the atmosphere. Moreover, calculations are greatly simplified by the first-order approximation for terms involving v_0/c. For $a < 0$ the velocity constraint applies to the inner boundary, for which $r = R_I$; for $a > 0$, to the outer boundary, for which $r = R_O$.

2.1.2 Line Opacity and Resonance

For points near the origin of a comoving frame—a frame of reference whose origin is stationary with respect to the atmosphere near that origin—the opacity as a function of wavelength as measured in the comoving frame is, in the model under consideration, a superposition of narrow (non-overlapping) profiles, each corresponding to an atomic (or,

in principle, molecular) line transition, which I shall call a "line". For a monochromatic specific intensity beam, which I shall call a "beam", whose wavelength, as measured by a comoving observer on the beam path, lies between sufficiently separated line profiles, the atmosphere in the neighborhood of the observer is transparent to the beam. In the opposite extreme, a comoving observer who finds the beam's wavelength to be at the center of a line profile occupies a *resonance point* for the beam and the line, and surrounding the observer is a *resonance region* in which the beam interacts with the line.

2.2 An Unbounded Atmosphere

As a beginning, let us ignore inner and outer boundaries of the line-forming region. Consider a region of space filled with a gaseous medium with bulk motion that is radial. Assume that the characteristic thermal speed for a gas particle is small in comparison with the local mean radial velocity and that the atmosphere is transparent except for a single line, whose wavelength dependence in the comoving frame is given by a normalized distribution function, ϕ, centered at λ'.

2.2.1 Impact-Parameter Representation of the Beam

Consider a beam that propagates through this atmosphere. Because of spherical symmetry, the beam's trajectory may be fully described by its impact parameter, p, and one may without loss of generality describe the geometry of the system in terms of the plane that contains the center of the velocity field and the beam path.

In Figure 2.1, the beam is represented by a thick line and propagates to the right. The beam path coordinate, $\bar{s}$, is zero at C, which on the beam is the closest point to O, the center of the radial velocity field. In this diagram, $\bar{s}$ is positive for points on the

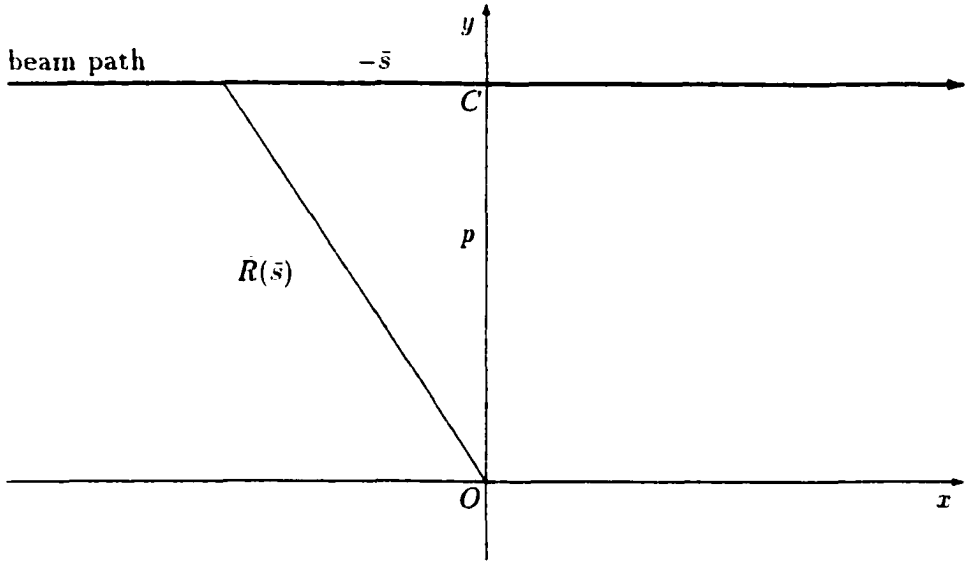


Figure 2.1: The beam path is represented by the horizontal line at top. The positive distance to a point at negative beam path parameter $\bar{s}$ is indicated by $-\bar{s}$. On the y-axis at distance p from the center O of the velocity field, the point of the beam's closest approach to O is C. On the incoming (left) side of C, $\bar{s}$ is negative; on the outgoing (right) side of C, $\bar{s}$ is positive. The distance from O to the point at $\bar{s}$ is $\bar{R}(\bar{s})$.

beam to the right of C; negative, to the left of C. The distance from O to the point at $\tilde{s}$ on the beam is $\tilde{R}(\tilde{s}) = \sqrt{p^2 + \tilde{s}^2}$.

Let the wavelength that a comoving observer at C ascribes to the beam be ℓ. Although a comoving observer at C is not stationary with respect to O, there is no component of atmospheric velocity along the beam at C, and so, to first order in velocity, ℓ is the same wavelength observed for the beam by any (non-comoving) observer stationary with respect to O.

At this point I introduce the unitless quantities, $\tilde{u} = \tilde{s}/r_0$ and $\tilde{w} = p/r_0$, which will simplify many of the expressions that follow.

Let $\tilde{\Lambda}(\tilde{u})$ be the wavelength that a comoving observer at a point corresponding to $u = \tilde{s}/r_0$ on the beam ascribes to the beam. According to the first-order Doppler correction,

$$\tilde{\Lambda}(\tilde{u}) = \left[1 + \frac{r_0 \tilde{u}}{\tilde{R}(r_0 \tilde{u})} \frac{v_a(\tilde{R}(r_0 \tilde{u}))}{c}\right] \ell = \left[1 + \tilde{f}_a(\tilde{u}) \frac{v_0}{c}\right] \ell, \tag{2.2}$$

where

$$\tilde{f}_a(\tilde{u}) = \left[\tilde{w}^2 + \tilde{u}^2\right]^{[a-1]/2} \tilde{u} \tag{2.3}$$

is proportional to the fractional difference between the comoving wavelength at $\tilde{u} = \tilde{s}/r_0$ and ℓ, the comoving wavelength at C. So long as $a \geq 0$, $\tilde{f}_a(\tilde{u})$ increases monotonically with $\tilde{u}$. In particular this means that for both a constant-speed wind ($a = 0$) and a homologously expanding atmosphere ($a = 1$), the wavelength of a beam becomes ever longer, in the comoving frame of reference, as the beam traverses the atmosphere.

For the case in which $\tilde{w} = p/r_0 = 1$, Figure 2.2 illustrates the beam's wavelength as perceived by comoving observers along the beam path. The horizontal axis represents $\tilde{u} = \tilde{s}/r_0$, the beam path parameter in units of r_0. The vertical axis represents $\tilde{f}_a(\tilde{u})$. For each curve, a positive slope indicates local redshifting of the beam; a negative slope indicates local blueshifting. The curves for $a < 0$ show that the beam blueshifts,

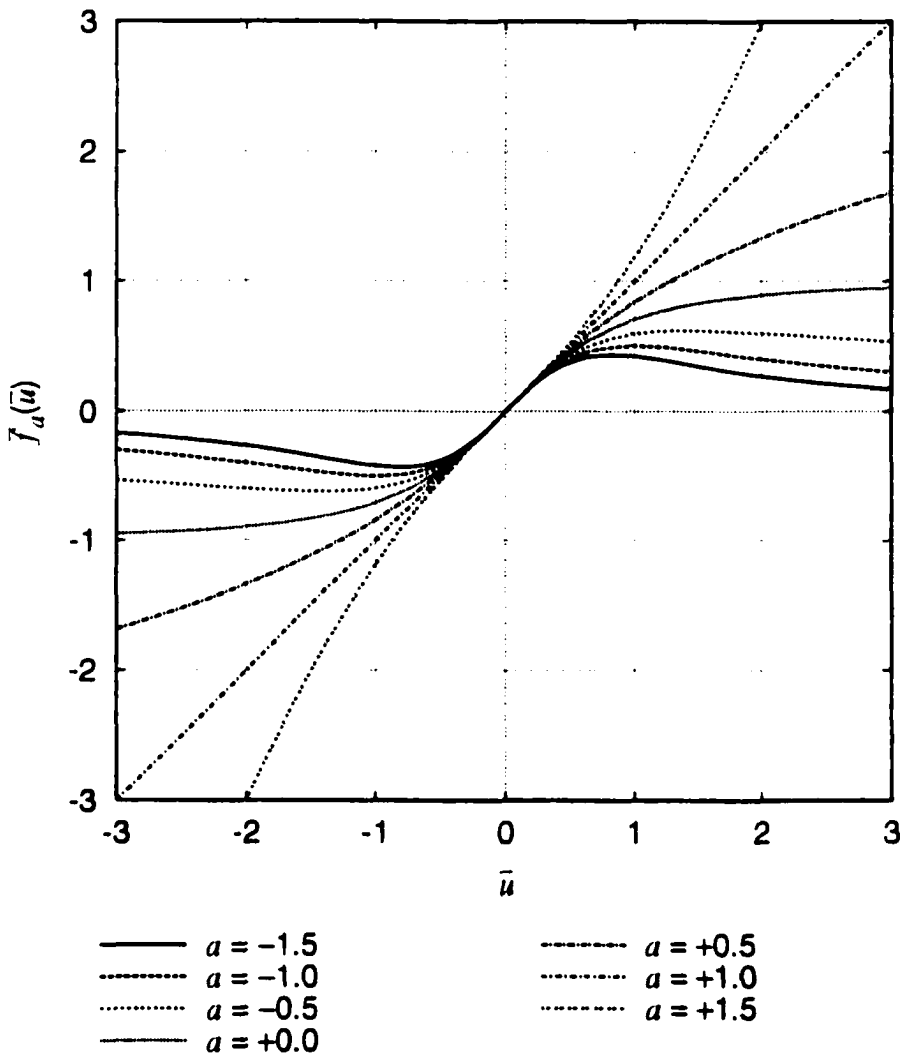


Figure 2.2: For each of several values of a, $\bar{f}_a(\bar{u})$, which is proportional to the fractional deviation of the comoving wavelength from ℓ, is plotted against $\bar{u} = \bar{s}/r_0$.

redshifts, and then blueshifts as it passes through the atmosphere. The curve for $a = 0$ shows that the beam always redshifts, but it asymptotically approaches a constant wavelength at large distances. The curves for $a > 0$ show that the beam always redshifts as it passes through the atmosphere and that the beam obtains every wavelength in an unbounded atmosphere. Note that the curve for $a = 1$ is a straight line, and so the beam redshifts linearly with increasing u in the case of homologous expansion.

For a given velocity field, to hold $\bar{w} = p/r_0$ fixed is to specify a particular trajectory, but many distinct monochromatic beams share the same trajectory. One must also fix ℓ in order to specify a particular beam. As the beam travels through the atmosphere it may appear to a comoving observer on the beam to be at the central wavelength, λ', for the opacity distribution function, ϕ. Any such observer is located at a resonance point for the beam (determined by $\bar{w}$ and ℓ) and the line (determined by ϕ).

For a given beam and a given line, Figure 2.2, in which a resonance point would be identified by the intersection of (1) the curve corresponding to a particular value of a and (2) the horizontal line corresponding to a particular value of λ', implies that there will be zero, one, or two resonance points in an unbounded expanding atmosphere with a power-law velocity profile. Specifically, for $a < 0$ there will be zero, one, or two resonance points; for $a = 0$, zero or one resonance point; and for $a > 0$, exactly one resonance point. Below we shall consider as specific examples $a = -1$, $a = 0$, and $a = 1$, which together cover the three qualitatively distinct types of atmosphere that the model under consideration describes.

In order to calculate the optical depth of the resonance region later, we effectively invert $\bar{\Lambda}$ by finding a function $\bar{U}_a$ such that the beam path coordinate $\bar{u}$ for a particular comoving beam wavelength λ is given by $\bar{U}_a(\, L(\ell, \lambda)\,)$, where

$$L(\lambda_1, \lambda_2) = \left[\frac{\lambda_2}{\lambda_1} - 1\right] \frac{c}{v_0} \tag{2.4}$$

is proportional to the fractional difference between λ_2 and λ_1.

For $a < 0$, in which case there are as many as two values of $\bar{u}$ that correspond to a given comoving wavelength, we define two functions, $\bar{U}_{a-}$ and $\bar{U}_{a+}$, such that $\bar{U}_{a-}(\, L(\ell, \lambda)\,)$ is one of the two values of $\bar{u}$ for which the beam has comoving wavelength λ, and $U_{a+}(\, L(\ell, \lambda)\,)$ is the other of the two values. Then

$$\bar{\Lambda}(\, \bar{U}_{a-}(\, L(\ell, \lambda)\,)\,) = \bar{\Lambda}(\, \bar{U}_{a+}(\, L(\ell, \lambda)\,)\,) = \lambda. \tag{2.5}$$

The solution for $a = -1$ is

$$\bar{U}_{-1\pm}(z) = \frac{1 \pm \sqrt{1 - [2z\bar{w}]^2}}{2z}. \tag{2.6}$$

where $\bar{U}_{-1+}(z)$ is defined for $0 < |z| \leq 1/2\bar{w}$, and $\bar{U}_{-1-}(z)$ is defined for $|z| \leq 1/2\bar{w}$. Moreover,

$$\bar{U}_{-1-}\left(\pm\frac{1}{2\bar{w}}\right) = \bar{U}_{-1+}\left(\pm\frac{1}{2\bar{w}}\right) = \pm\bar{w}, \tag{2.7}$$

respectively, as is evident from Figure 2.2, in which the curve for $a = -1$ has an extremum at each of $\bar{u} = s/r_0 = s/p = \pm 1$. Finally, $\bar{U}_{-1+}(0)$ is undefined, but $\bar{U}_{-1-}(0) = 0$, and $\bar{U}_{-1-}(z)$ is well defined for z in the neighborhood of 0.

For any $a \geq 0$ one needs only a single function $\bar{U}_a$ such that

$$\bar{\Lambda}(\, \bar{U}_a(\, L(\ell, \lambda)\,)\,) = \lambda. \tag{2.8}$$

For $a = 0$, $\bar{U}_0(z)$ is defined only for $|z| < 1$, and

$$\bar{U}_0(z) = \frac{\bar{w}z}{\sqrt{1 - z^2}}. \tag{2.9}$$

Just as for $a < 0$, a corresponding value of $\bar{u}$ will only exist for λ sufficiently close to

ℓ because there is a maximum atmospheric velocity difference between a point on the beam path and point C in Figure 2.1. For $a > 0$, $\tilde{U}_a(z)$ is defined for any value of z. The simplest example corresponds to $a = 1$, for which

$$\tilde{U}_1(z) = z. \tag{2.10}$$

2.2.2 Common-Resonance-Point Representation

Now that the basic variation of comoving wavelength with beam path parameter has been presented, the objective becomes to examine the region surrounding a resonance point. Figure 2.3 locates on the y–axis a resonance point P for a line centered at λ' and a monochromatic beam, whose trajectory is identified by y–coordinate ρ and direction-angle θ. By varying θ, one may consider many beams that share a common resonance point. The new beam path parameter, s, increases in the direction of beam propagation and gives the distance to the resonance point, P. The impact parameter in this representation is $p = \rho \sin\theta$. The distance from O to a point at s on the beam path is

$$R(s) = \sqrt{\rho^2 + s^2 + 2\rho s \cos\theta}. \tag{2.11}$$

The comoving beam wavelength at C is now

$$\ell = \left[1 - \frac{v_a(\rho)}{c}\cos\theta\right]\lambda', \tag{2.12}$$

and as before we introduce unitless quantities, $w = \rho/r_0$ and $u = s/r_0$, that will simplify expressions that follow. So, the comoving beam wavelength at a point corresponding to u on the beam is now

$$\Lambda(u) = \left[1 + \frac{r_0 u + \rho\cos\theta}{R(r_0 u)} \; \frac{v_a(\,R(r_0 u)\,)}{c}\right]\ell. \tag{2.13}$$

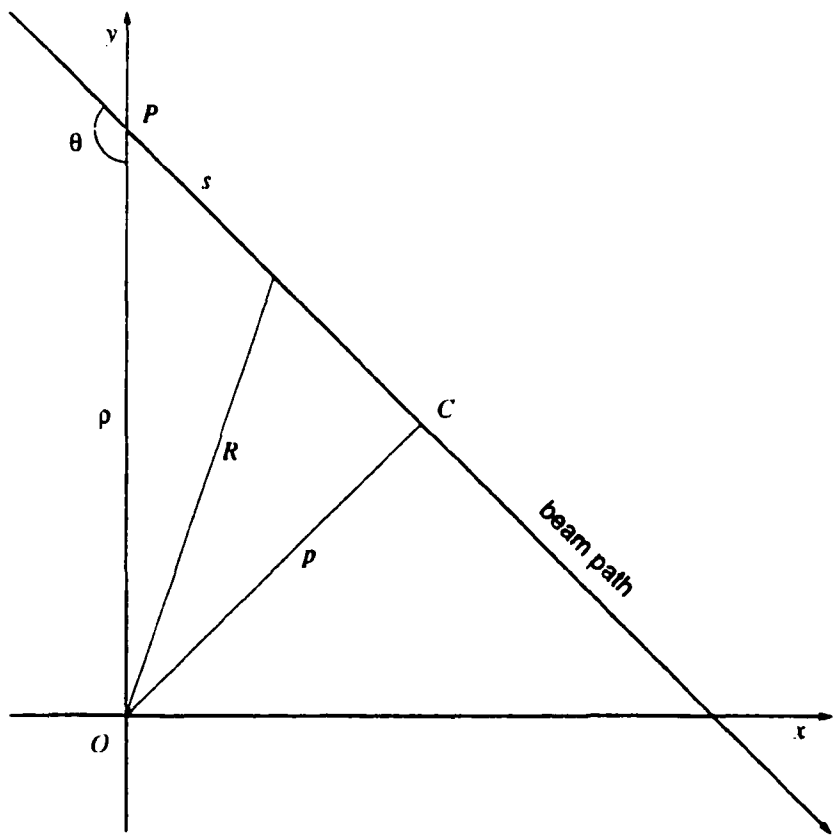


Figure 2.3: The beam travels from the upper left to the lower right. Points O and C and distance p have exactly the same meaning as in Figure 2.1. The beam has comoving wavelength λ' at the resonance point P on the positive y–axis. The direction angle θ measures the angular separation at P between the incoming beam and the negative y–direction. The distance between the resonance point and the the origin O is ρ. The distance between a point on the beam and P is s for points on the outgoing side of P and $-s$ for points on the incoming side of P. The distance between O and a point on the beam is $R(s)$.

Plugging in the power law for velocity and only keeping first-order terms in v_0 yields

$$\Lambda(u) = \left[1 + f_a(u)\frac{v_0}{c}\right]\lambda', \tag{2.14}$$

where

$$f_a(u) = [u + w\cos\theta][w^2 + u^2 + 2wu\cos\theta]^{[a-1]/2} - w^a\cos\theta. \tag{2.15}$$

For the case in which $w = \rho/r_0 = 1$, Figure 2.4 displays some curves that correspond to $a = -1$. The horizontal axis represents $u = s/r_0$. The vertical axis represents $f_{-1}(u)$. Each curve corresponds to a particular value of the direction angle, θ. Note that the curve corresponding to 90° is just the same as the $a = -1$ curve in Figure 2.2. At $\theta = 90°$, the beam passes through the resonance point at impact parameter distance from the center of the velocity field; the resonance point is situated mid-way between the point of maximum blueshift and the point of maximum redshift for the beam. As θ decreases from 90°, the resonance point moves closer to the point of maximum redshift until $\theta = 45°$, where the resonance point is the point of maximum redshift for the beam. So, at $\theta = 45°$, the beam is blueshifted in either direction away from the resonance point. For $0° < \theta < 45°$ the beam encounters the resonance point as its comoving wavelength asymptotically blueshifts toward a constant value. As θ approaches 0°, the beam passes ever closer to the center of the velocity field, where velocities are very large, and so the maximum redshift and maximum blueshift of the beam become arbitrarily large. The case for $90° < \theta < 180°$ is very similar to the one just described, but the resonance point moves through the point of maximum blueshift; each corresponding curve is symmetric about the origin with a curve in Figure 2.4.

In a manner similar to that of Figure 2.4, Figure 2.5 displays some curves that correspond to the $a = 0$ case. The vertical axis now corresponds to $f_0(u)$. Again note that the curve corresponding to 90° is just the same as the $a = 0$ curve in Figure 2.2. As

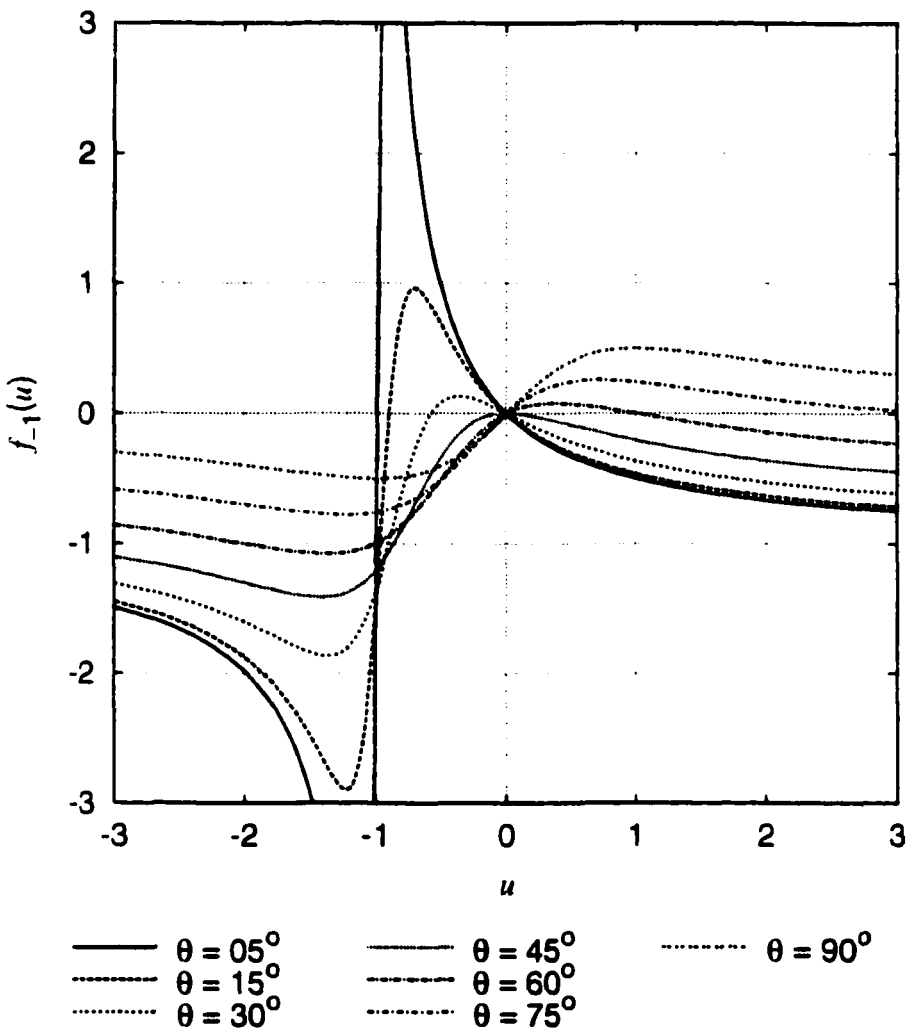


Figure 2.4: For each of several values of θ, $f_{-1}(u)$, which is proportional to the fractional difference between λ' and the comoving wavelength at $u = s/r_0$, is plotted against u.

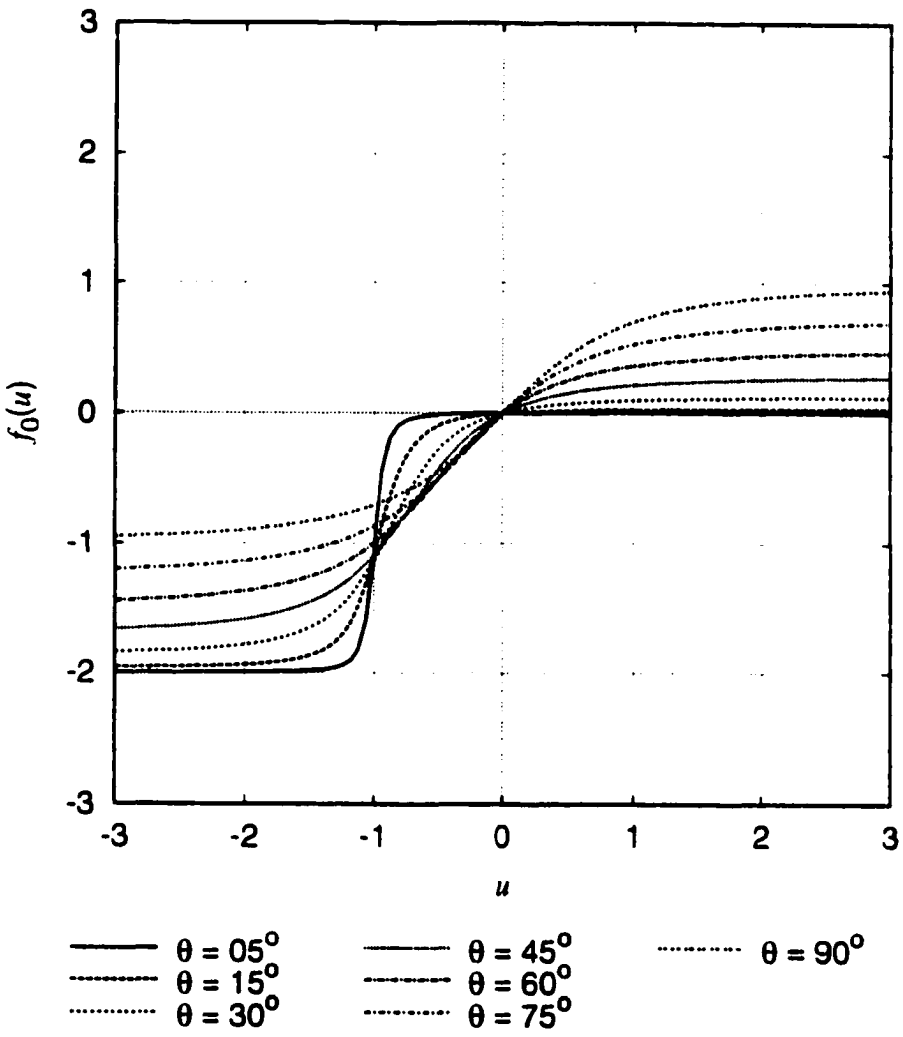


Figure 2.5: Like Figure 2.4, but for $a = 0$.

θ decreases toward 0°, the comoving wavelength at the resonance point asymptotically approaches the maximum redshift limit for the beam. As θ increases toward 180°, the comoving wavelength at the resonance point asymptotically approaches the maximum blueshift limit for the beam.

For $a = 1$, $f_1(u) = u$, and therefore $\Lambda(u) = [1 + uv_0/c]\lambda'$, are independent of w and θ, just as $\bar{f}_1(u)$, and therefore $\bar{\Lambda}(u)$, are independent of $\bar{w}$. The beam redshifts uniformly as it traverses the atmosphere.

Now $\Lambda(u)$ can be inverted to produce the functions, $U_{a\pm}$ and U_a, analogous to $\bar{U}_{a\pm}$ and $\bar{U}_a$ above. $U_a(\, L(\lambda', \lambda)\,)$ is the value of the beam path parameter u such that the comoving wavelength at the corresponding point on the beam is λ.

For $a = -1$,

$$U_{-1\pm}(z) = \left[\frac{1 \pm \sqrt{1 - \{2[wz + \cos\theta]\sin\theta\}^2}}{2[wz + \cos\theta]} - \cos\theta\right] w. \tag{2.16}$$

$U_{-1+}(z)$ is defined for

$$0 < |wz + \cos\theta| \leq \frac{1}{2\sin\theta}, \tag{2.17}$$

and $U_{-1-}(z)$ is defined for

$$-\frac{1}{2\sin\theta} - \cos\theta < wz \leq \frac{1}{2\sin\theta} - \cos\theta. \tag{2.18}$$

Figure 2.6 indicates the range of values of z for which $U_{-1\pm}(z)$ are defined. The horizontal axis corresponds to θ, the beam path direction angle, and the vertical axis corresponds to wz, a normalized wavelength parameter. U_{-1-} is defined for all points between the top and bottom curves; U_{-1+} is, too, except for points along the central curve.

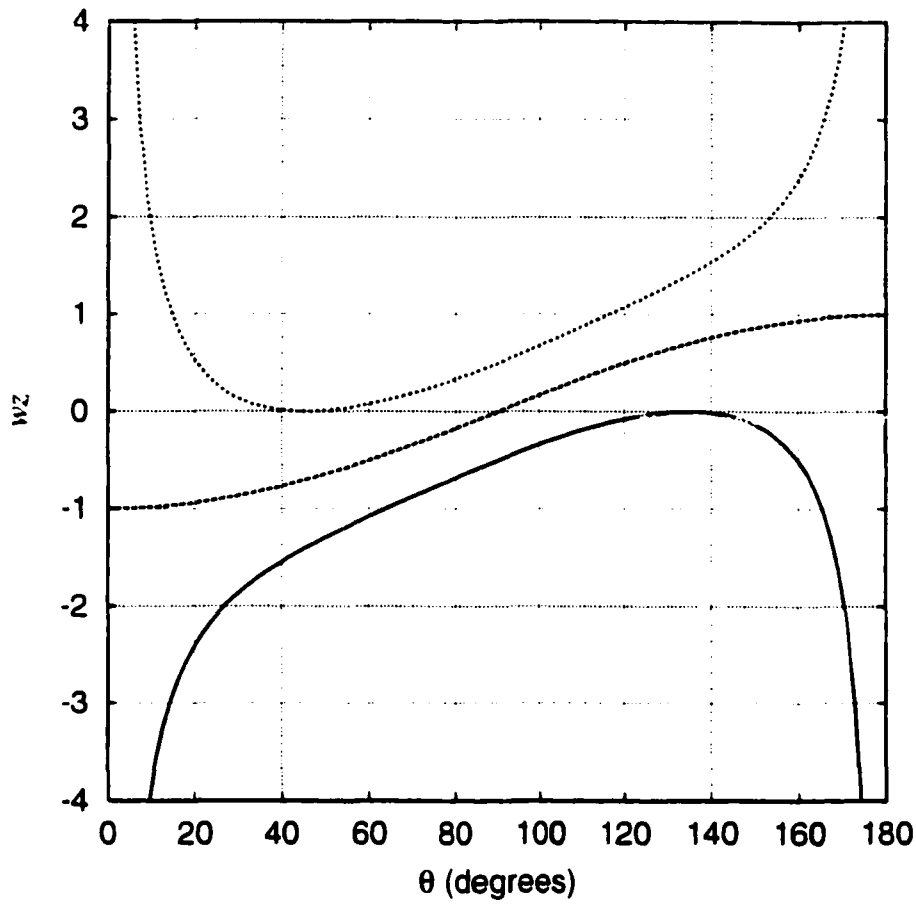


Figure 2.6: The highest contour represents the upper limit of the product wz such that $U_{-1_\pm}(z)$ are defined; the lowest contour represents the lower limit such that they are defined. For points along the central contour, U_{-1_+} is undefined, and so its domain is composed of two disjoint sets.

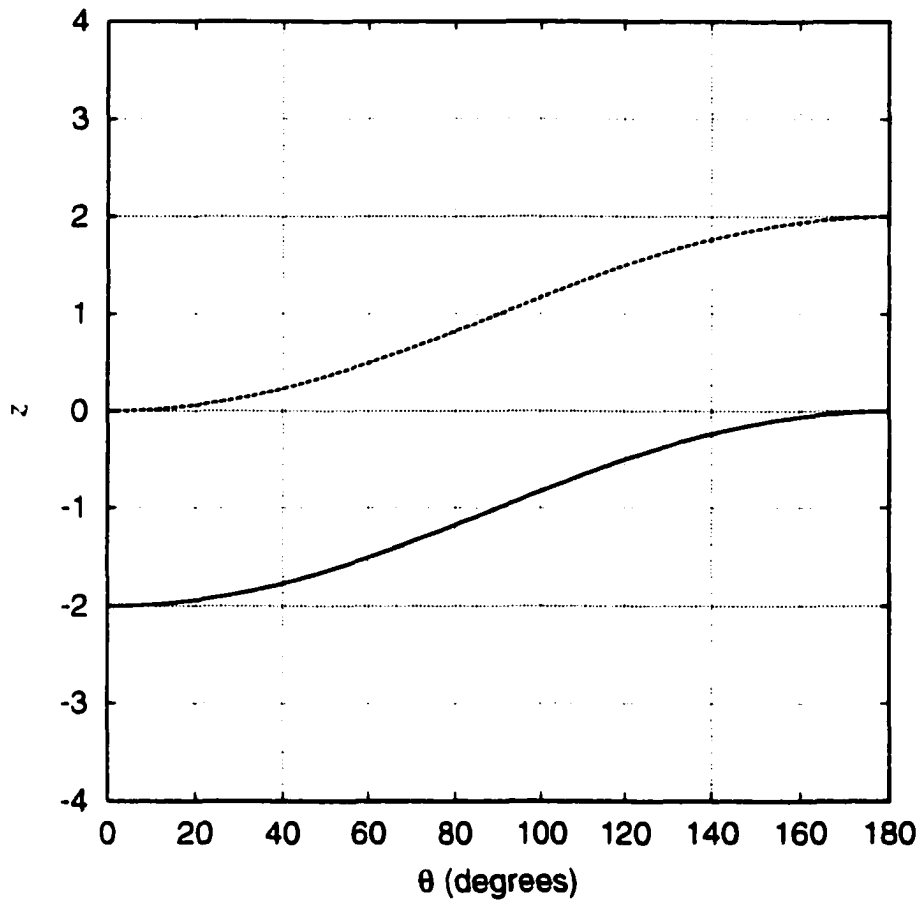


Figure 2.7: The higher contour represents the upper limit of z such that $U_0(z)$ is finite; the lower contour represents the lower limit.

$U_0(z)$ is expressed in a manner that will simplify optical depth integrals below.

$$U_0(z) = \begin{cases} -\infty & \text{if } z \leq -1 - \cos\theta \\ \left[\frac{[z+\cos\theta]\sin\theta}{\sqrt{1-[z+\cos\theta]^2}} - \cos\theta\right] w & \text{if } -1 - \cos\theta < z < 1 - \cos\theta \\ +\infty & \text{if } z \geq 1 - \cos\theta \end{cases} . \tag{2.19}$$

Similar to Figure 2.6, Figure 2.7 indicates the range of values of z for which $U_0(z)$ has finite value. The vertical axis corresponds in this case to a simpler normalized wavelength parameter, z. U_0 is finite for all points between the upper and lower curves.

For $a = 1$,

$$U_1(z) = z, \tag{2.20}$$

and so, regardless of θ (and w), $U_1(z)$ is defined for every z.

2.2.3 Optical Depth and the Resonance Region for $a = 0$

We shall now examine the optical depth of the resonance region in a constant-speed wind atmosphere, for which $a = 0$.

Integrated Line Opacity

Assume that the integrated line opacity at radial coordinate r can be approximated by

$$K_b(r) = \left[\frac{r}{r_0}\right]^b K_0, \tag{2.21}$$

a radial power law. The power-law index is b, and the integrated line opacity at radial coordinate r_0 is K_0. Then the total optical depth of the resonance region can be written

as

$$\tau \;=\; \int_{-\infty}^{+\infty} K_b(\,R(s)\,)\,\phi(\,\Lambda(s/r_0)\,)\,\mathrm{d}s \;=\; r_0 \int_{-\infty}^{+\infty} K_b(\,R(r_0 u)\,)\,\phi(\,\Lambda(u)\,)\,\mathrm{d}u, \qquad (2.22)$$

an integral over the entire beam path. The assumption of constant mass-loss rate for a constant-speed wind gives a radial power-law index of -2 for the atmospheric density. To assume that the integrated line opacity follows the matter density is to assume that $b = -2$. This assumption is correct only if the level populations associated with the source of opacity are independent of radial coordinate. Nevertheless, the assumption is made here as a rough approximation so that expressions below may be evaluated explicitly.

Line Profile

We may evaluate the integral directly if we approximate the comoving line profile as a rectangular impulse of width 2ϵ and height $1/2\epsilon$. Then

$$\phi(\lambda) = \begin{cases} 0 & \text{if } |\lambda - \lambda'| > \epsilon \\ 1/2\epsilon & \text{otherwise} \end{cases} . \qquad (2.23)$$

and $\epsilon = \sigma\sqrt{3}$ gives a distribution with standard deviation σ. This is of course an unrealistic shape for the comoving line profile, and this shape is not a useful approximation unless the actual line width is very small. Nevertheless, the discontinuity at each edge of this artificial comoving line profile produces, in the figures that follow, easily recognizable features, typically of the cusp sort. The features, though unrealistic, do allow one to develop an intuitive grasp of the model and to verify the validity of the expressions derived below.

An approximate expression for the optical depth of the resonance region becomes

$$\tau = \frac{T}{\varepsilon} \int_{U_0(-\epsilon)}^{U_0(+\epsilon)} \frac{\mathrm{d}u}{w^2 + 2wu\cos\theta + u^2}. \tag{2.24}$$

where

$$T = \frac{r_0 K_0}{2}. \tag{2.25}$$

and

$$\varepsilon = \frac{\epsilon}{\lambda'} \frac{c}{v_0}. \tag{2.26}$$

In the $w = 1$ case, Figure 2.8 displays for each of several values of ε a contour that represents the boundary of the line in physical space. Each axis represents, in units of r_0, the distance from the resonance point, and the plane of the graph contains the center of the velocity field. For a beam that travels on a non-radial trajectory and has its incoming side on the left, a contour corresponds to $U_0(-\varepsilon)$ on the left half ($u_x < 0$) of the graph and to $U_0(\varepsilon)$ on the right half ($u_x > 0$) of the graph; here, u_x is the horizontal distance from the resonance point in units of r_0, and u_y similarly corresponds to vertical distance. As w changes, the entire figure maintains its shape, but the scale is directly proportional to w. On its way toward the resonance point, a beam that crosses the contour enters the line; if the incoming beam does not cross the contour, then the incoming beam is entirely within the line. On its way from the resonance point, a beam that crosses the contour departs from the line; if the outgoing beam does not cross the contour, then the outgoing beam is entirely within the line.

At large distance from the resonance point, each contour converges to an incoming asymptote and to an outgoing asymptote. The incoming asymptote corresponds to a critical angle, $\theta_- = \arccos(\varepsilon - 1)$, and the outgoing asymptote corresponds to a critical angle, $\theta_+ = \arccos(1 - \varepsilon)$. By definition, for a collection of beams that share the same

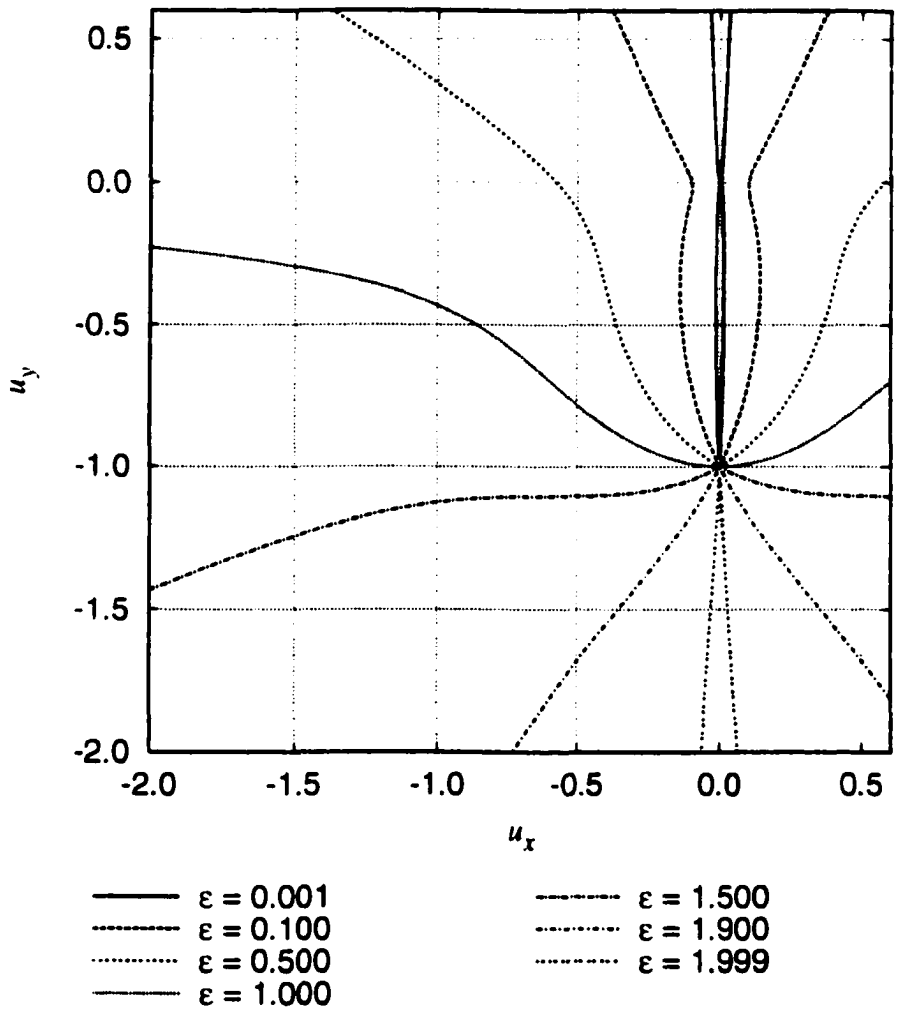


Figure 2.8: The horizontal axis represents u_x, the horizontal distance from the resonance point in units of r_0. Similarly, the vertical axis represents u_y, the vertical distance from the resonance point in units of r_0. For $u_x < 0$, contours of $U_0(-\varepsilon)$ are displayed, and for $u_x > 0$, contours of $U_0(\varepsilon)$ are displayed. This figure corresponds to the $w = 1$ case.

resonance point, every beam is in the line at least near the resonance point. For $\theta < \theta_+$, the entire outgoing beam is in the line; for $\theta > \theta_-$, the entire incoming beam is in the line. For $\varepsilon < 1$, there is no beam entirely within the line because $\theta_+ < \theta_-$. When $\varepsilon = 1$, $\theta_- = \theta_+ = 90°$, and so only the beam with $\theta = 90°$ is entirely within the line. For $1 \leq \varepsilon < 2$, the line is sufficiently wide so that $0° < \theta_- \leq \theta_+ \leq 180°$, and every beam with $\theta_- \leq \theta \leq \theta_+$ is entirely within the line. For $\varepsilon \geq 2$, every beam that shares the resonance point is entirely within the line.

Approximate Expression for Total Optical Depth

Evaluation of the integral leads to an approximate expression for τ when $a = 0$:

$$\tau = \frac{T}{\varepsilon w \sin\theta} \left[g(\cos\theta + \varepsilon) - g(\cos\theta - \varepsilon) \right], \tag{2.27}$$

where

$$g(x) = \begin{cases} +\pi/2 & \text{if } x > 1 \\ -\pi/2 & \text{if } x < -1 \\ \arcsin(x) & \text{otherwise} \end{cases} . \tag{2.28}$$

Figure 2.9 illustrates the relationship between the resonance region's normalized total optical depth τ/T and the normalized line width ε. Note that because of symmetry, a curve corresponding to θ corresponds as well to $180° - \theta$, that the total optical depth of the resonance region becomes large as the beam's direction becomes radial, and that each curve converges at $\varepsilon = 0$ to the total optical depth of the resonance region for a line with a Dirac-delta-function profile. For $\theta = 90°$ the optical depth increases as the normalized line width ε grows from zero (corresponding to the delta-function profile) until the optical depth reaches a maximum at $\varepsilon = 1$; at this point, the line is minimally broad enough so that the beam is entirely in the line. As ε increases beyond unity,

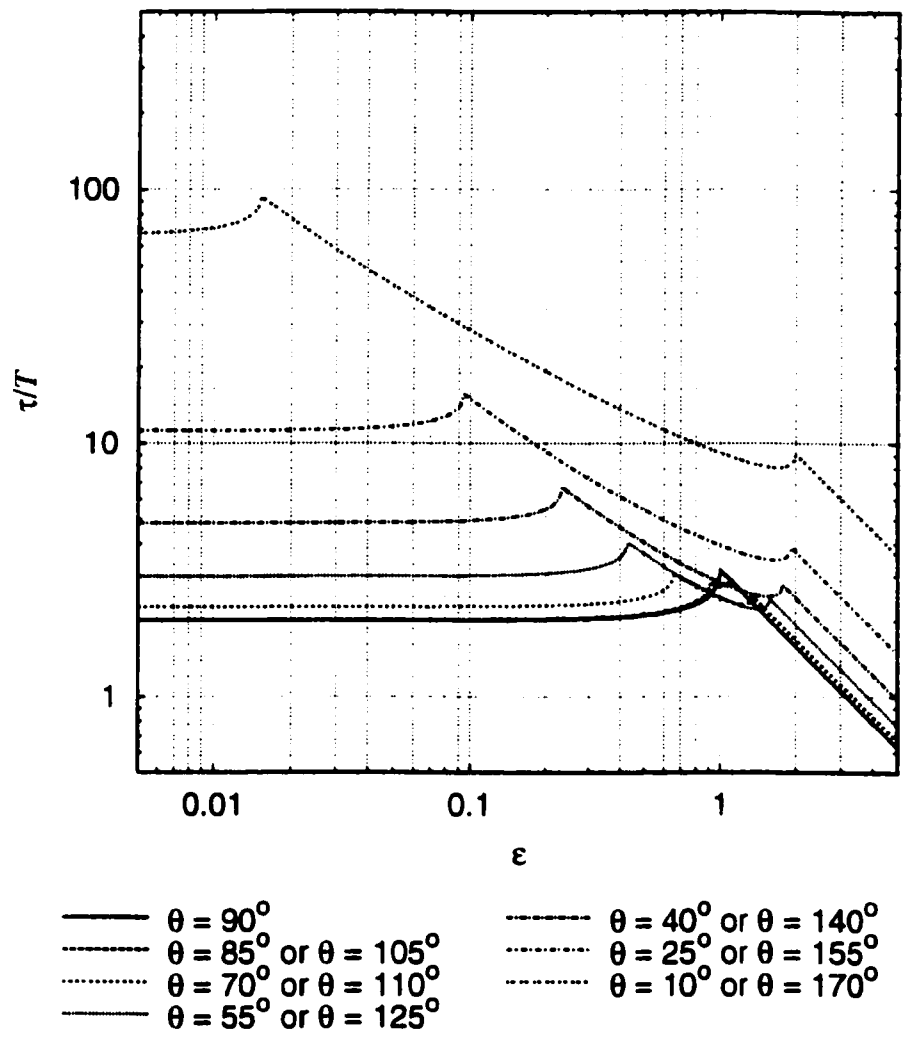


Figure 2.9: The normalized total optical depth τ/T of the resonance region is plotted against the normalized line width ε with $a = 0$ and $w = 1$, and for various values of θ.

the total optical depth diminishes because the decreasing height of the comoving line profile is no longer offset by increase in the (now unit) fraction of the beam that is in the line. In fact, the total optical depth τ becomes proportional to $1/\varepsilon$, and so the curve becomes a straight line with negative unit slope on the logarithmic plot. For $\theta < 90°$ ($\theta > 90°$) the optical depth increases as ε grows from zero until the optical depth reaches a maximum that corresponds to the line's being minimally wide enough so that the outgoing (incoming) beam is entirely in the line. As the line width increases further, the total optical depth diminishes at first but then reaches a second, lower, local maximum that corresponds to the minimum line width such that all points on the beam are in the line. Finally, as for $\theta = 90°$, further increase in the line width brings down the total optical depth along with the height of the comoving line profile.

Figure 2.10 represents the resonance region's total optical depth as a function of the beam's direction angle for values of ε up to unity. For every value of ε, the curve diverges to infinite optical depth as the beam's trajectory becomes radial. This makes sense because $K_{-2}(r)$ becomes arbitrarily large as r decreases, and the impact parameter p for the beam decreases to zero as the trajectory becomes radial; the beam passes through a region of arbitrarily large opacity for a trajectory sufficiently near a radial orientation.

Consider first the curves corresponding to the smallest two values of ε. Note that for both of $\varepsilon = 0.001$ and $\varepsilon = 0.01$, the total optical depth is essentially the same. (For a model in which a supernova is surrounded by a constant-velocity wind, useful values of ε lie between about 0.001 and 0.01). For a very narrow line ($\varepsilon \ll 1$),

$$\tau = \frac{T}{w}\left\{\frac{2}{\sin^2\theta} + \left[\frac{1+2\cos^2\theta}{3\sin^6\theta}\right]\varepsilon^2\right\}. \tag{2.29}$$

The optical depth has practically reached the limit corresponding to a delta-function

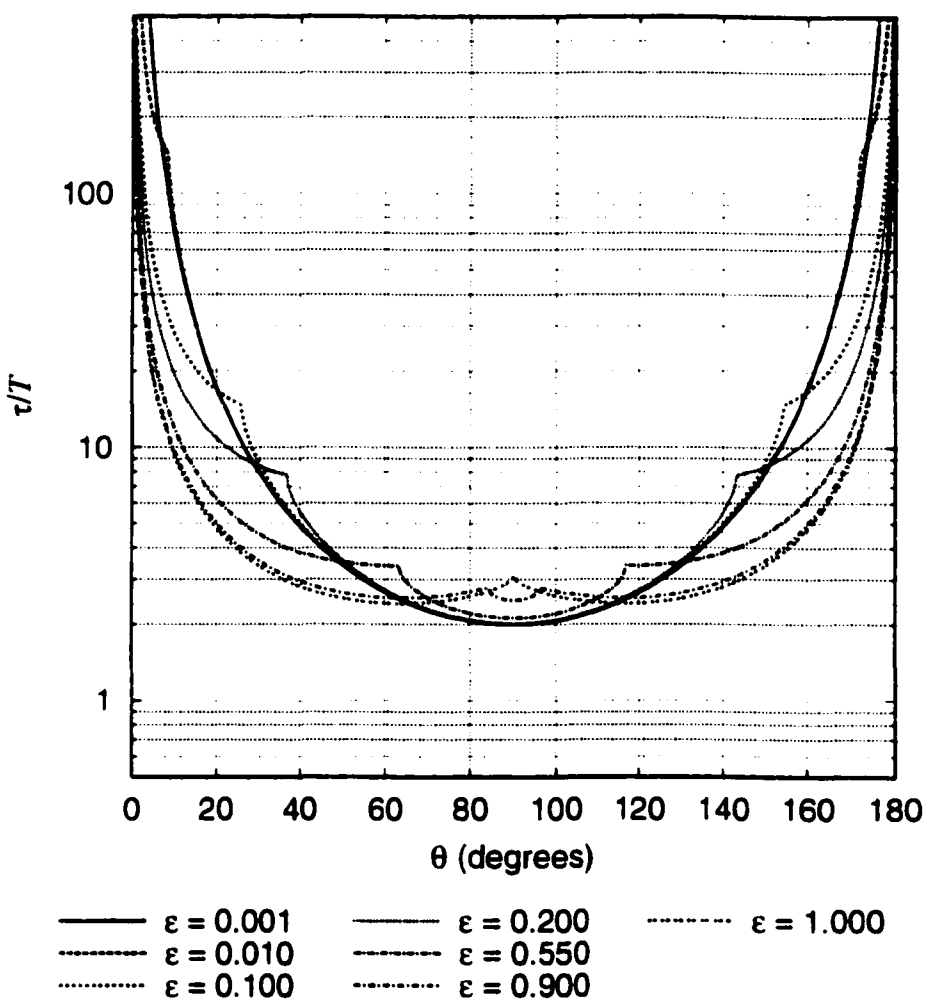


Figure 2.10: Like Figure 2.9 but normalized total optical depth of the resonance region as a function of θ and for various values of the normalized line width ε.

line profile, at least for θ larger than about 10° and smaller than about 170°, where the narrowest-line curves begin to separate from one another. Note that as θ approaches 90°, τ/T approaches 2, and so because of symmetry, the optical depth at the resonance point is T. Therefore, $T = r_0 K_0/2$ is just the optical depth at the resonance point for a line with a delta-function profile and a beam with a direction angle of 90° in an unbounded constant-velocity atmosphere, and for $w = 1$. In this case, T is also clearly the minimum optical depth at the resonance point; any value of θ other than 90° corresponds to an optical depth larger than T.

Next consider each curve corresponding to a value of ε less than unity. For $\theta = 90°$, at a certain point on the outgoing (incoming) side of the beam, the beam leaves (enters) the line. As θ decreases (increases), that point moves farther from the resonance point. For $\theta < \theta_+$, the outgoing beam never leaves the line, and so the optical depth accumulated on the outgoing side of the beam suddenly decreases as, with decreasing θ, the outgoing beam, no longer increasing in fractional path length inside the line, retreats more rapidly from the center of the opacity distribution. (For $\theta > \theta_-$, the incoming beam is always in the line, and so the optical depth accumulated on the incoming side of the beam suddenly decreases as, with increasing θ, the incoming beam, no longer increasing in fractional path length inside the line, retreats more rapidly from the center of the opacity distribution.) So $\theta = \theta_+$ ($\theta = \theta_-$) corresponds to the left (right) cusp in each of the curves for $\varepsilon < 1$; for example, the left (right) cusp corresponding to $\varepsilon = 0.55$ is expected to occur at about $\arccos(1 - \varepsilon) \approx 63°$ (or $\arccos(\varepsilon - 1) \approx 117°$), as shown in the figure. The curve corresponding to the smallest value of ε has cusps that are above the top edge of the figure.

Next, consider the curve corresponding to $\varepsilon = 1$, which is the minimum value such that the entire beam is in the line for $\theta = 90°$. As θ decreases (increases), the outgoing (incoming) side of the beam, completely within the line, retreats more rapidly from the

center of the opacity distribution, and so its contribution to the optical depth decreases. At first, the contribution from the incoming (outgoing) side decreases as well because the beam's entry (departure) point into (from) the line rapidly moves closer to the resonance point from infinite distance. As θ decreases (increases) further, however, the beam's trajectory eventually passes near enough to the center of the opacity distribution so that the optical depth contributed by the incoming (outgoing) beam grows much more rapidly than the optical depth contributed by the outgoing (incoming) beam diminishes.

Figure 2.11 is just like Figure 2.10 except that curves are plotted for $\varepsilon \geq 1$ instead of $\varepsilon \leq 1$. Consider first the curves corresponding to $1 < \varepsilon < 2$. Because $\varepsilon > 1$, the beam is entirely within the line for $\theta_- < \theta < \theta_+$. For θ within this range, $\tau = \pi T/\varepsilon w \sin\theta \propto \csc\theta$. As θ decreases below θ_- (increases above θ_+), we expect, in correspondence with the left (right) cusp in the figure, a sudden decrease in total optical depth as the incoming (outgoing) beam enters (departs from) the line at finite distance from the resonance point. In consideration of the curves corresponding to $\varepsilon \geq 2$, we find that the the beam is always entirely within the line; so there are no cusps, and each such curve is given by a function proportional to $\csc\theta$.

For θ approaching zero, Figure 2.12 shows the divergence of curves representative of Figures 2.10 and 2.11. Note that as θ falls below about 10°, the curve for $\varepsilon = 0.001$ is distinguishable from the curve for $\varepsilon = 0.01$. Each curve's linear nature on the left side of the figure arises because for small θ,

$$\tau \simeq \frac{T}{\varepsilon w \theta}\left[\frac{\pi}{2} - g(1-\varepsilon)\right]. \tag{2.30}$$

The small-θ dependence of $\log(\tau/T)$ on $\log\theta$ is linear, and the slope is -1, because τ/T is proportional to θ^{-1}.

Figure 2.13 shows the dependence of τ on w for small ε. As w decreases, the

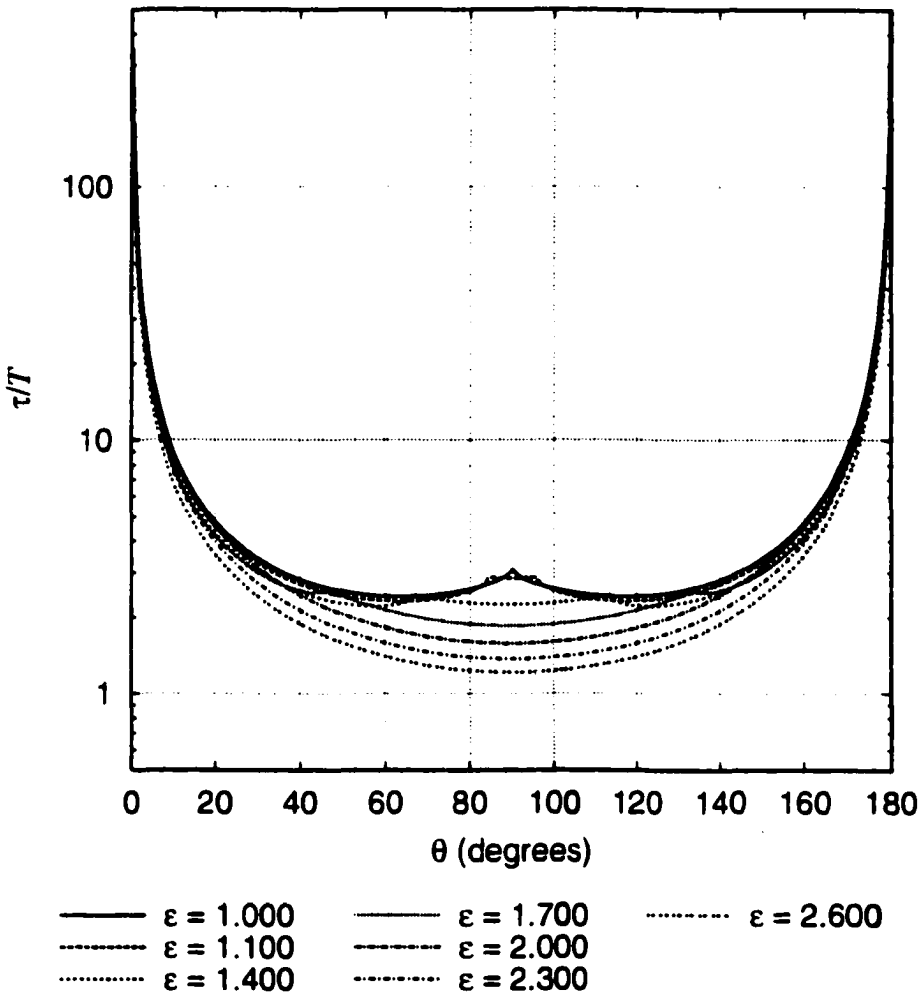


Figure 2.11: Like Figure 2.10 but for $\varepsilon \geq 1$.

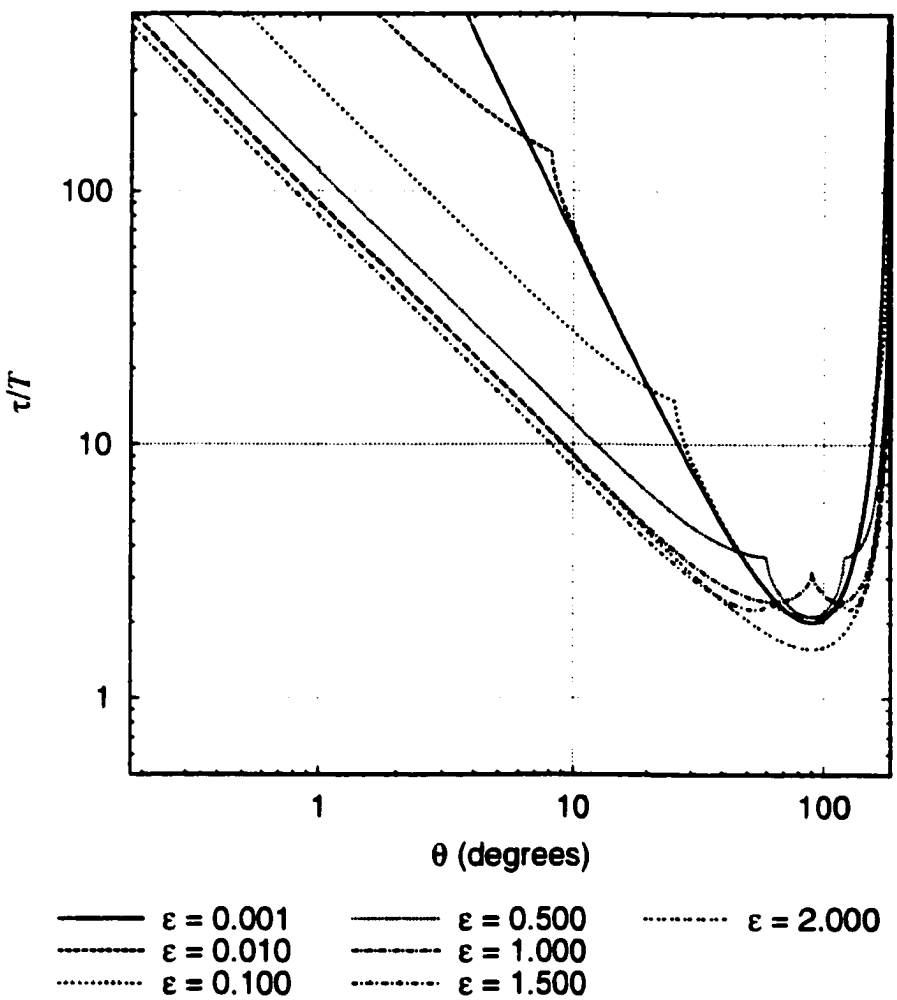


Figure 2.12: Like Figures 2.10 and 2.11 but with a logarithmic scale for the angular axis.

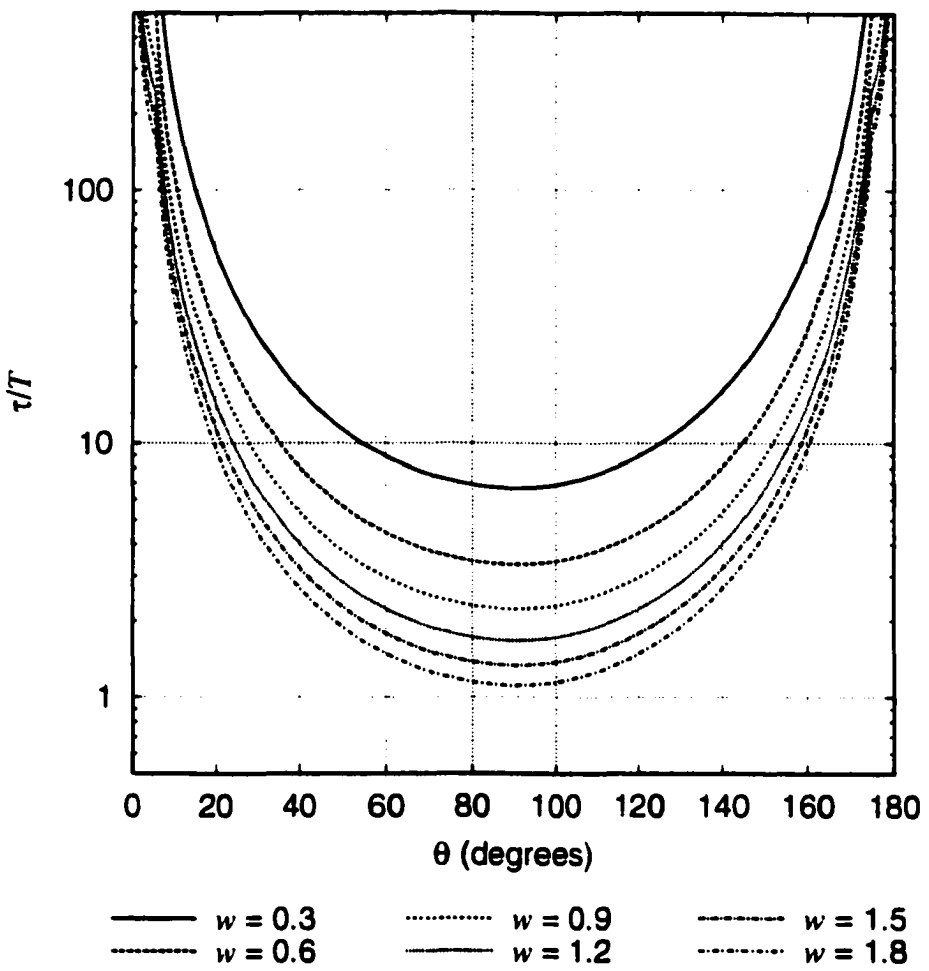


Figure 2.13: Like Figure 2.10 but with $\varepsilon = 0.005$ and for several values of w.

resonance point moves closer to the center of the velocity field, where the integrated line opacity $K_{-2}(r)$ is larger. Basically, one of the curves for very small ε in Figure 2.10 is multiplied by $1/w$ in order to produce the curves in Figure 2.13.

Accumulation of Optical Depth and the Shape of the Resonance Region

At a point corresponding to $u = s/r_0$, the optical depth of the atmosphere along the incoming beam path is

$$t(u) = \begin{cases} 0 & \text{if } u \le U_0(-\varepsilon) \\ \frac{T}{\varepsilon w \sin\theta}\left[\arctan\left(\frac{u}{w\sin\theta} + \cot\theta\right) - g(\cos\theta - \varepsilon)\right] & \text{if } U_0(-\varepsilon) < u \le U_0(+\varepsilon) \\ \tau & \text{if } u > U_0(+\varepsilon) \end{cases} . \tag{2.31}$$

This expression allows us to visualize the shape of the resonance region by plotting a contour of constant optical depth for all beams arriving at a particular resonance point. Figure 2.14 displays a contour of constant optical depth $\tau/T = 0.1$ for each of the values of ε used in Figure 2.8, and the scale is the same as in that figure. For $\tau/T = 0.1$, every contour of constant optical depth is almost identical to the corresponding contour of constant $U_0(-\varepsilon)$.

Figure 2.15 displays contours of constant optical depth $\tau/T = 1$. The most significant deviation from the contours of Figure 2.8 occurs for θ near 90° and ε near unity.

In practice we are interested in narrow lines; that is, small values of ε (between 0.001 and 0.01). For $\varepsilon = 0.005$ and $w = 1$ Figure 2.16 shows contours of constant optical depth for various values of t/T. The contours are essentially identical on a scale large enough to include on the same graph both the center of the velocity field at $(0, -1)$ and the resonance point at the origin. For a narrow line, the resonance region is physically

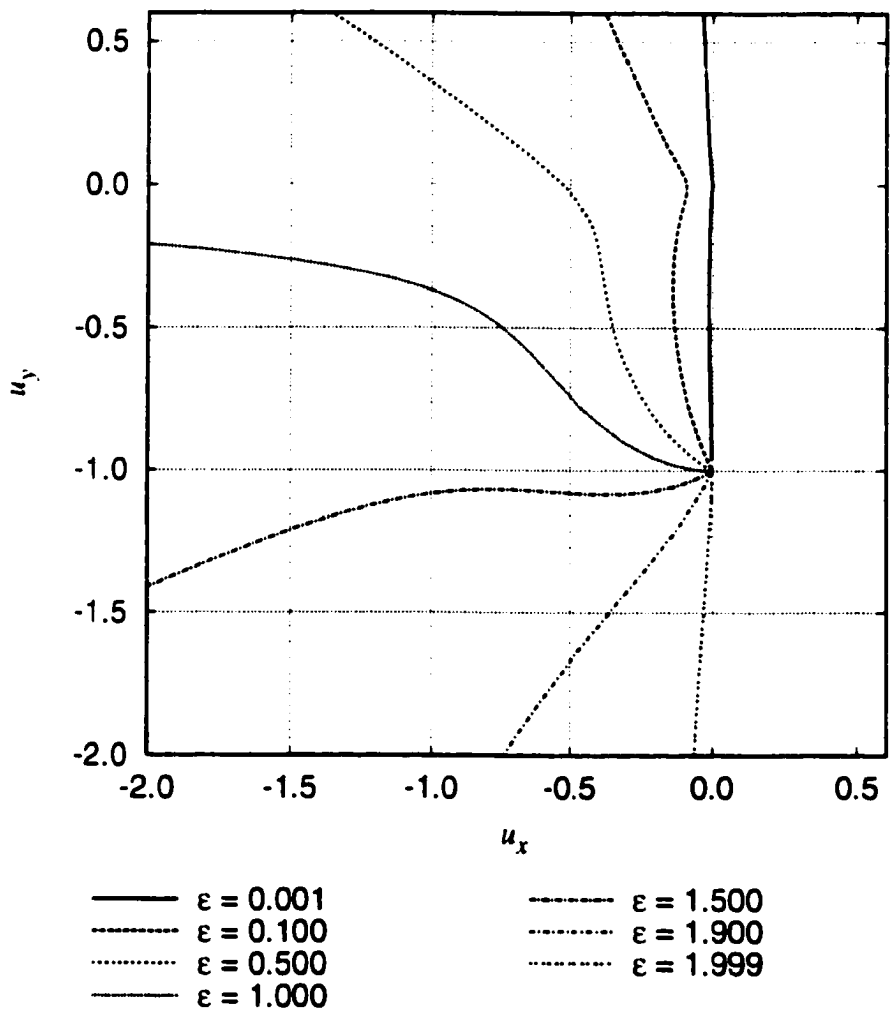


Figure 2.14: For the incoming beam in an unbounded atmosphere, a contour of constant optical depth $\tau/T = 0.1$ (for $w = 1$) is plotted for each of several values of ε. The resonance point is at the origin, and the center of the velocity field is at $(0, -1)$. Compare with Figure 2.8.

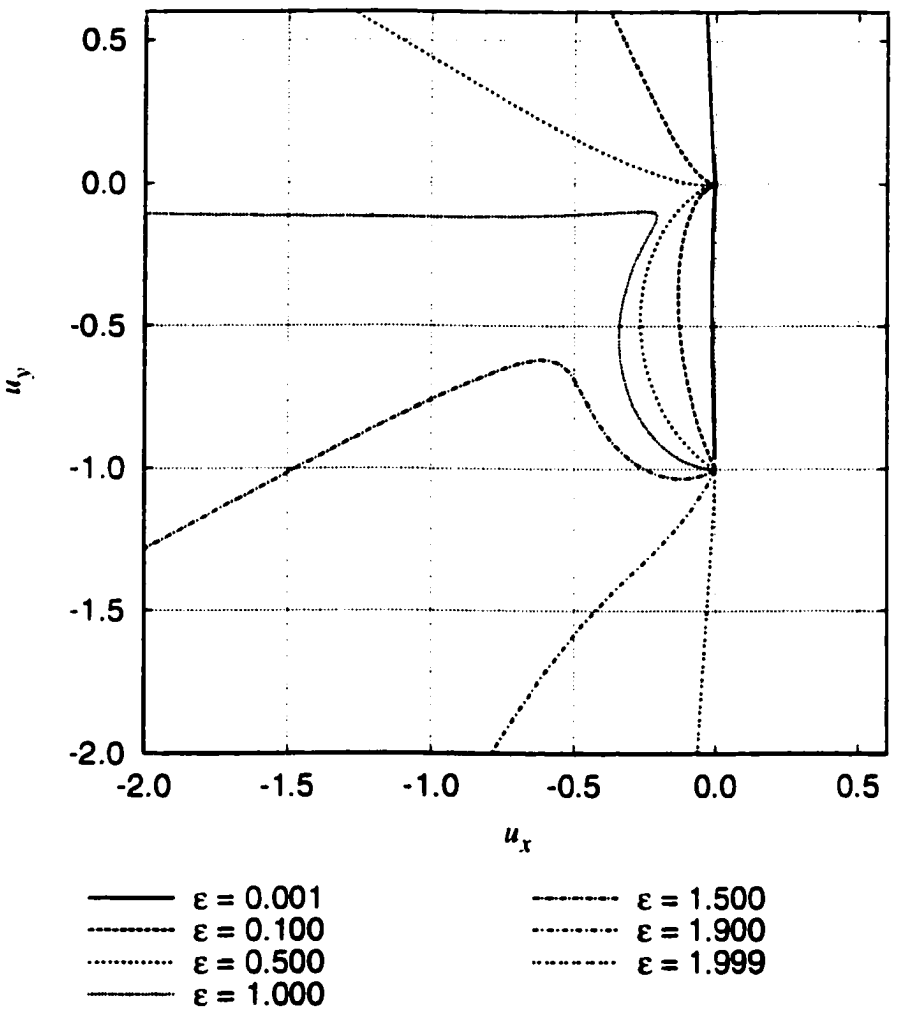


Figure 2.15: Like Figure 2.14, but for $\tau/T = 1$.

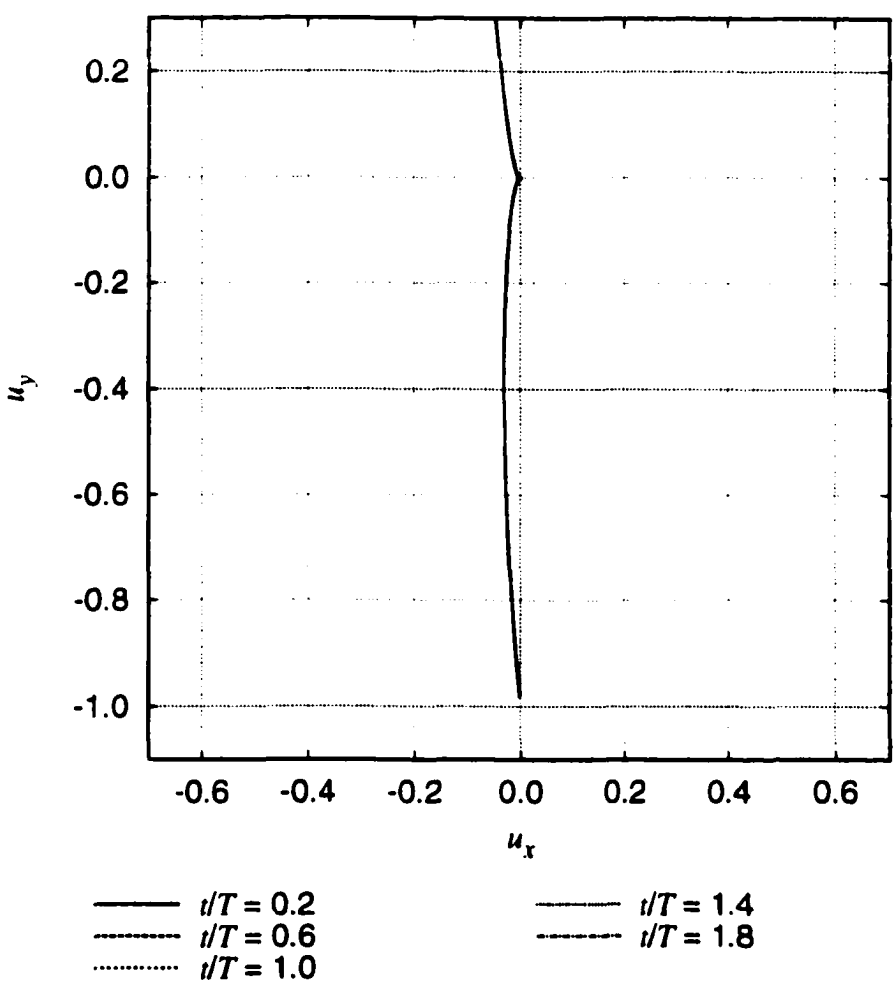


Figure 2.16: For the incoming beam in an unbounded atmosphere, a contour of constant optical depth (for $\varepsilon = 0.005$ and $w = 1$) is plotted for each of several values of normalized optical depth t/T. The resonance point is at the origin, and the center of the velocity field is at $(0, -1)$. Although several contours are plotted, they overlap completely on this scale.

narrow and is elongated in the radial direction.

Figure 2.17 shows a magnified view of the vicinity of the resonance point. Note that for values of optical depth t greater than T the contour passes through the resonance point because the incoming beam does not accumulate the specified optical depth before it reaches the resonance point. For a delta-function line profile, if $\theta = 90°$, and $w = 1$, then T is the optical depth of an incoming beam at the resonance point. The optical depth at the resonance point is minimum for a beam with $\theta = 90°$, and so increasing optical depth for a contour causes the contour first to pass through the resonance point at $\theta = 90°$. For larger values of optical depth, a contour passes through the resonance point at an angle farther from 90°.

For $t/T = 1$ Figure 2.18 illustrates the shape of the resonance region for several values of w. Remember that w is just the distance of the resonance region from the center of the velocity field in units of r_0, and so the center of the velocity field is located at $(0, -w)$ in this figure. The effect of increasing w is strongly to elongate and weakly to widen the resonance region.

For the very same contours as in Figure 2.18, Figure 2.19 presents a magnified view of the region near the resonance point. Note that even though the total optical depth for any direction angle decreases with increasing w, the contour for $w = 0.2$ more closely approaches the resonance point than does the contour for $w = 0.6$. This merely reflects that for small w the resonance point is near the center of the distribution of integrated line opacity, and so most of the optical depth is accumulated very near the resonance point.

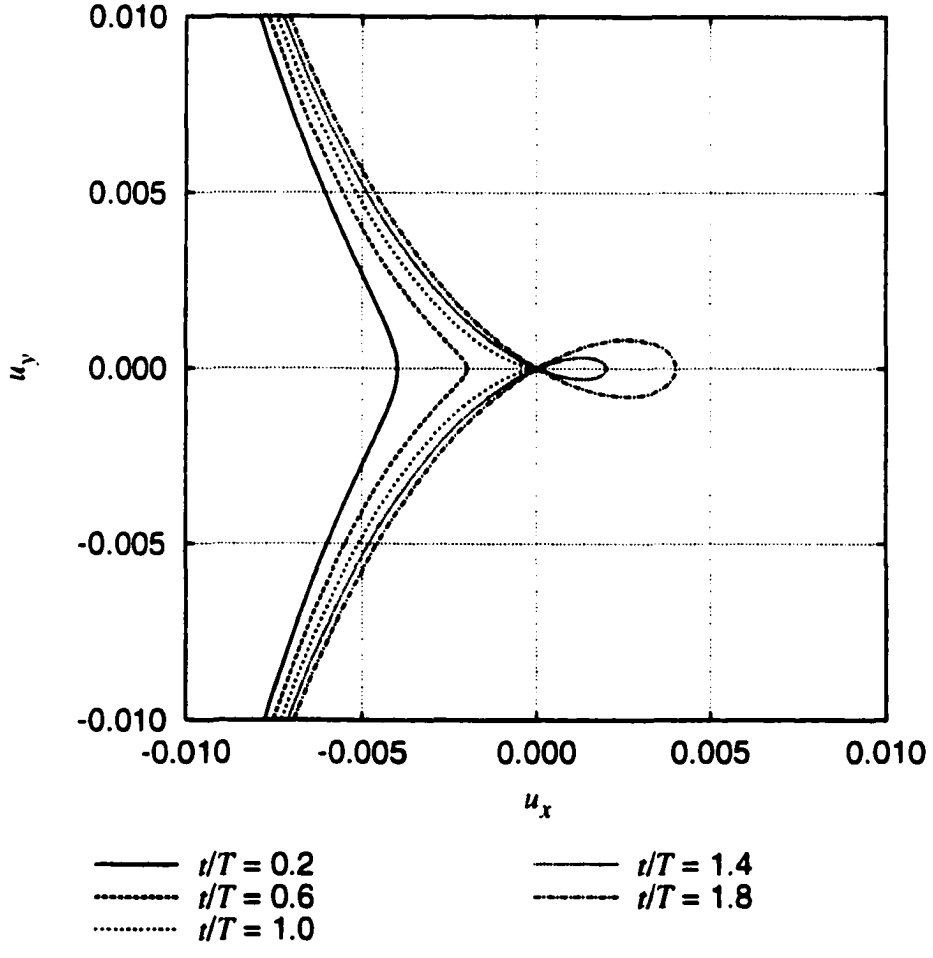


Figure 2.17: Like Figure 2.16, but a close-up view near the resonance point.

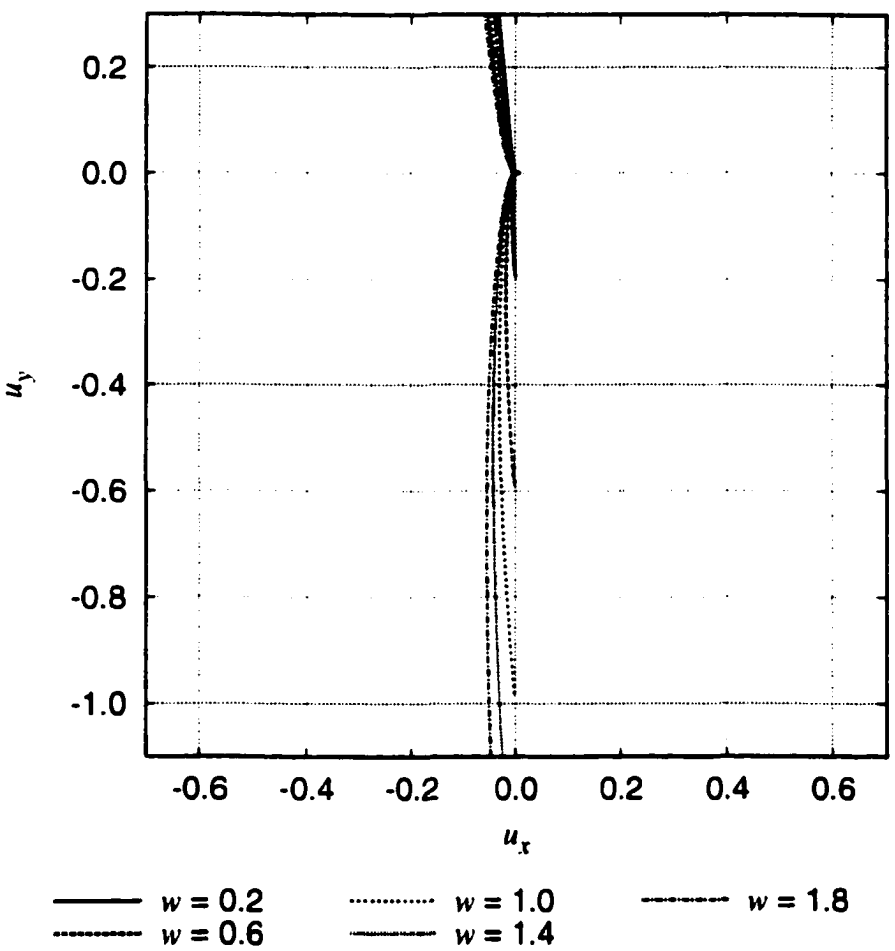


Figure 2.18: Like Figure 2.16, but with each contour corresponding to $t/T = 1$ for a particular value of w. The center of the velocity field is now located at $(0, -w)$.

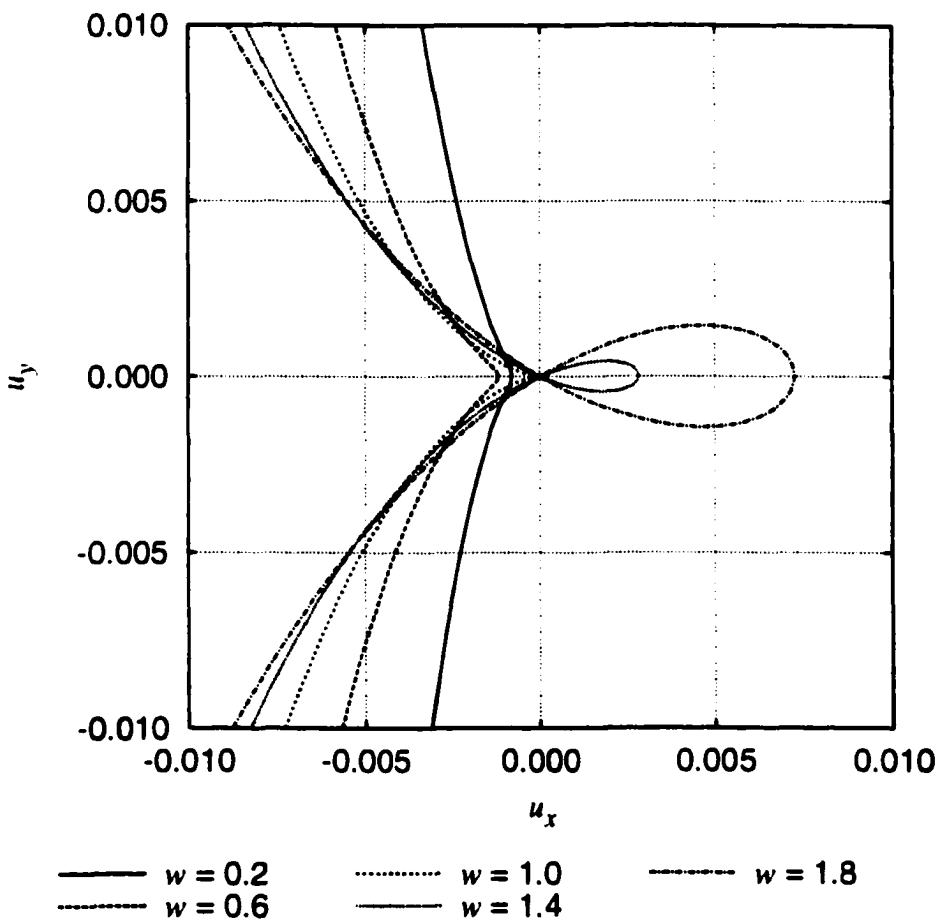


Figure 2.19: Like Figure 2.18, but a close-up view near the resonance point.

2.3 A Bounded Atmosphere with $a = 0$

Let $w_I = R_I/r_0$ and $w_O = R_O/r_0$ represent, respectively, the inner and outer boundary radii in units of r_0. Then the location of the resonance point must be such that $w_I < w < w_O$. At the resonance point, the inner boundary subtends an angle,

$$\theta_I = \arcsin\left(\frac{w_I}{w}\right). \tag{2.32}$$

For $\theta_I \leq \theta \leq 180°$ the incoming beam does not accumulate optical depth until it is within the outer boundary, and so every contour of constant optical depth must lie between the outer boundary and the resonance point. For $0° \leq \theta < \theta_I$ the incoming beam is considered to originate on the inner boundary, and so every contour of constant optical depth must lie between the inner boundary and the resonance point.

The total optical depth of the resonance region becomes

$$\tau = \frac{T}{\varepsilon w \sin\theta}\left[\arctan\left(\frac{\Upsilon(+\varepsilon) + w\cos\theta}{w\sin\theta}\right) - \arctan\left(\frac{\Upsilon(-\varepsilon) + w\cos\theta}{w\sin\theta}\right)\right]. \tag{2.33}$$

where

$$\Upsilon(z) = \begin{cases} u_- & \text{if } U_0(z) < u_- \\ u_+ & \text{if } U_0(z) > u_+ \\ U_0(z) & \text{otherwise} \end{cases}, \tag{2.34}$$

$$u_- = \begin{cases} \sqrt{w_I^2 - w^2\sin^2\theta} - w\cos\theta & \text{if } 0 \leq \theta < \theta_I \\ -\sqrt{w_O^2 - w^2\sin^2\theta} - w\cos\theta & \text{if } \theta_I \leq \theta \leq \pi \end{cases}, \tag{2.35}$$

and

$$u_+ = \begin{cases} \sqrt{w_O^2 - w^2\sin^2\theta} - w\cos\theta & \text{if } 0 \leq \theta < \pi - \theta_I \\ -\sqrt{w_I^2 - w^2\sin^2\theta} - w\cos\theta & \text{if } \pi - \theta_I \leq \theta \leq \pi \end{cases}. \tag{2.36}$$

Like Figure 2.9, Figure 2.20 illustrates the relationship between the resonance re-

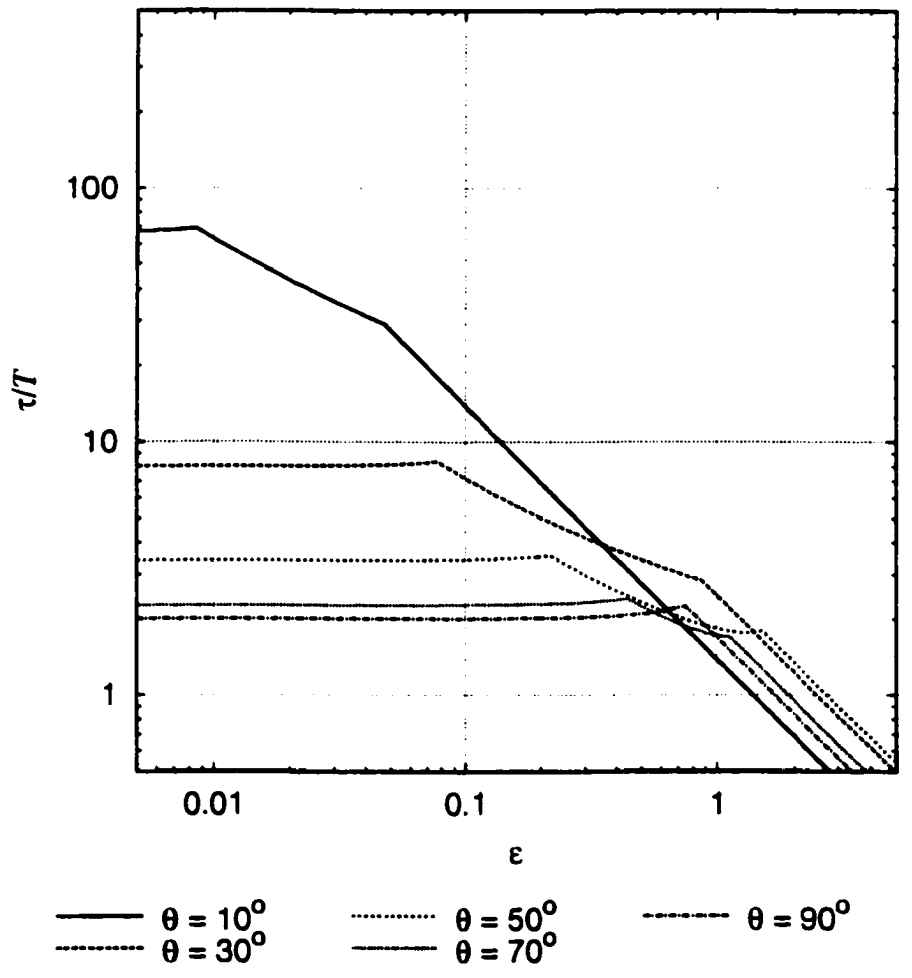


Figure 2.20: Like Figure 2.9, but with atmospheric boundaries, $w_{\mathrm{I}} = 0.5$ and $w_{\mathrm{O}} = 1.5$.

gion's normalized total optical depth τ/T and the normalized line width ε. The atmospheric boundaries have been chosen such that $w_I = 0.5$ and $w_O = 1.5$. As in the case of an unbounded atmosphere, a cusp in one of the curves still corresponds to a minimum linewidth such that the beam is always in the line in either the incoming or outgoing direction, or in both directions at once for $\theta = 90°$. In the bounded case, however, "entirely in the line" has changed in its meaning to imply that every comoving observer on the beam path *but only in the bounded volume* sees the beam within the line. In the unbounded case the cusp is situated at a value of ε such that $\theta_+ = \theta$ for the left cusp, and $\theta_- = \theta$ for the right cusp. In the bounded case for $\theta > \theta_I$, the cusp is situated at a value of ε such that $U(\varepsilon)$ identifies a point on the outer boundary for the left cusp, and $U(-\varepsilon)$ identifies a point on the outer boundary for the right cusp. For $\theta < \theta_I$ the left cusp is found in the same way, but the right cusp is situated at the value of ε such that $U(-\varepsilon)$ identifies a point on the inner boundary. Note that because the path length through the atmosphere now depends (discontinuously) on θ, for large values of ε the dependence of τ/T on θ is not so straight-forward as it was in the unbounded case. In particular, for large ε, the total optical depth does simply become larger as θ becomes smaller.

Like Figure 2.10, Figure 2.21 represents the resonance region's total optical depth as a function of the beam's direction angle for values of ε up to the critical value

$$\varepsilon_I = \sqrt{1 - \left[\frac{w}{w_O}\right]^2}, \tag{2.37}$$

which is the minimum value of ε such that the beam is entirely within the line for the bounded case. In this figure, $\varepsilon_I \approx 0.745$ because $w = 1$, and $w_O = 1.5$. The most striking difference from Figure 2.10 is that the curves corresponding to the larger values of ε do not appear to diverge for θ approaching 0 and 180. In fact, there is no divergence

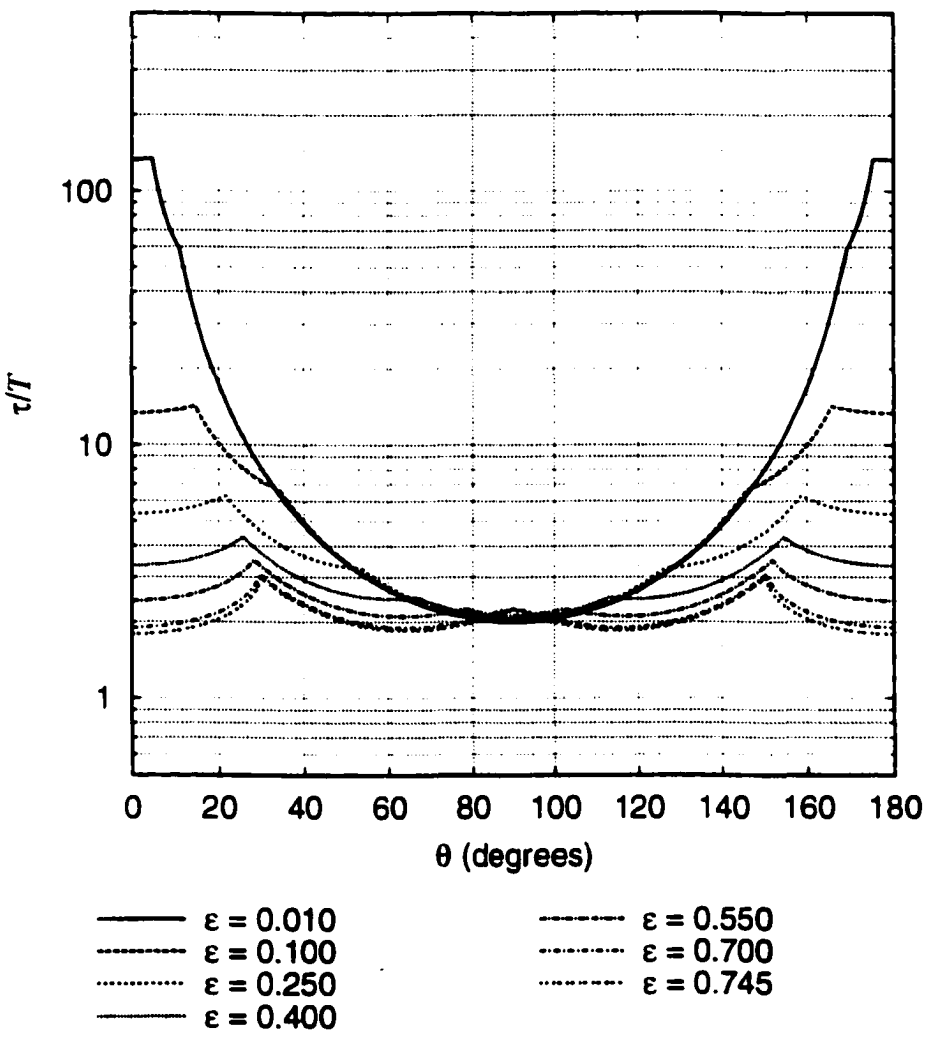


Figure 2.21: Like Figure 2.10, but with atmospheric boundaries, $w_{\mathrm{I}} = 0.5$ and $w_{\mathrm{O}} = 1.5$.

for ε = constant > 0. For $\varepsilon < \varepsilon_1$, the total optical depth has a local minimum at $\theta = 90°$, just as for $\varepsilon < 1$ in the unbounded case. As θ decreases (increases) from 90°, the total optical depth increases until the outgoing (incoming) beam is entirely in the line; then there is a cusp because the optical depth contributed by the outgoing (incoming) beam suddenly begins to decrease. In the unbounded case the angle corresponding to this cusp is θ_+ (θ_-), but in the bounded case, the angle is larger (smaller) because of the atmospheric outer boundary. As θ becomes still smaller (larger), the total optical depth inevitably begins to rise again as the opacity sampled by the incoming (outgoing) beam increases. Eventually, however, the total optical depth reaches a maximum value at the maximum (minimum) angle such that the beam segment between the inner boundary and the resonance point is entirely within the line. This corresponds to the left-most (right-most) cusp. As ε increases toward

$$\varepsilon_2 = \sqrt{1 - \left[\frac{w_1}{w}\right]^2}, \tag{2.38}$$

the cusp angle increases (decreases) toward θ_1 (or $180° - \theta_1$). When $\varepsilon = \varepsilon_2$, the line is minimally broad enough so that the entire beam segment between the inner boundary and the resonance point is in the line for a beam tangent to the inner boundary (that is, for $\theta = \theta_1$). In this example we have $\varepsilon_2 \approx 0.866$, and so the cusp angle continues to increase (decrease) for each subsequently larger value of ε in Figure 2.21.

Figure 2.22 is just like Figure 2.21 except that curves are plotted for $\varepsilon \geq \varepsilon_1$ instead of $\varepsilon \leq \varepsilon_1$. The left-most (right-most) cusp is located at the same angle, 30° (or 150°), for every curve except the one corresponding to $\varepsilon_1 \approx 0.745 < \varepsilon_2$. Every other curve corresponds to $\varepsilon \geq \varepsilon_2 \approx 0.866$; for each of these the line width ε is large enough so that every beam path crossing the inner boundary is completely within the line between the inner boundary and the resonance point. As the line width increases from ε_1, two

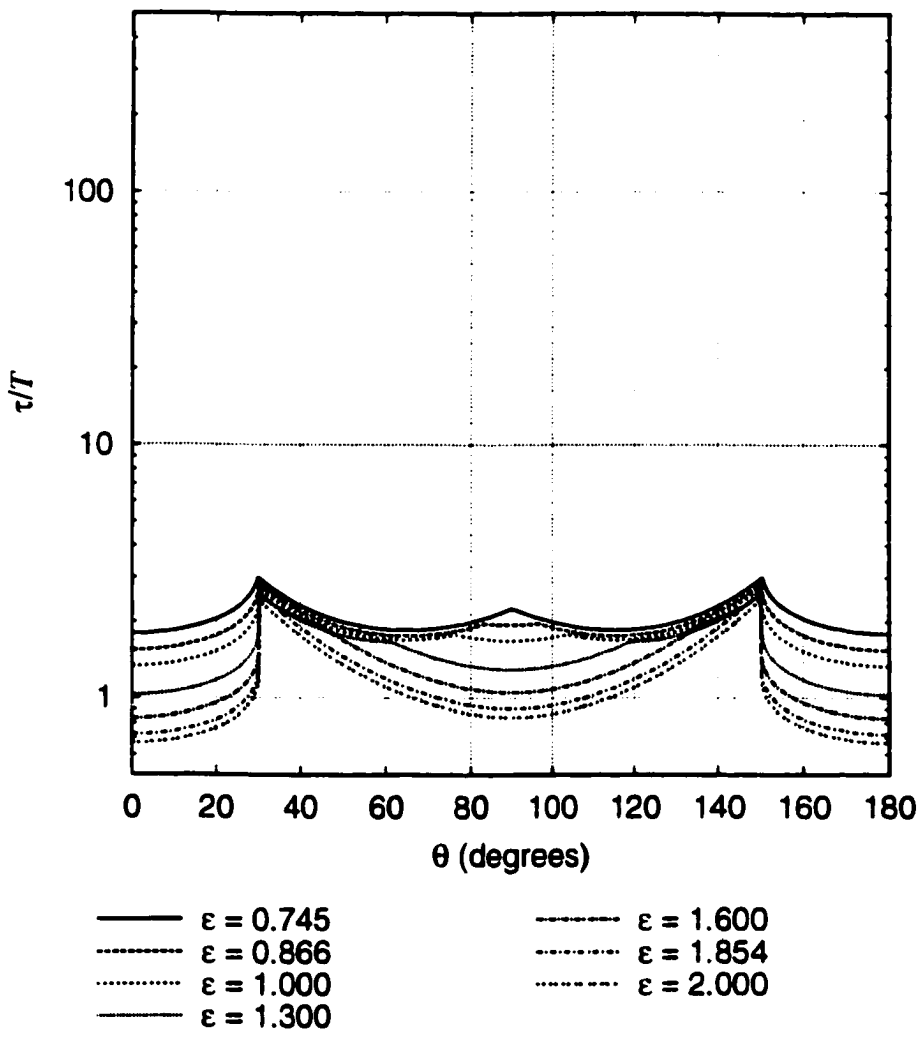


Figure 2.22: Like Figure 2.11, but with atmospheric boundaries, $w_{\mathrm{I}} = 0.5$ and $w_{\mathrm{O}} = 1.5$.

central cusps appear. Each of these corresponds to an angle at which the beam is just barely entirely within the line. The central cusps move outward with increasing ε until ε becomes

$$\varepsilon_3 = \sqrt{1 - \frac{w_I^2}{w^2}} + \sqrt{1 - \frac{w_I^2}{w_O^2 + 4\sqrt{[w_O^2 - w_I^2][w^2 - w_I^2]} + 4[w^2 - w_I^2]}}, \tag{2.39}$$

which, in this case, is approximately 1.854. At this point the line is wide enough so that a beam tangent to the inner boundary is just barely entirely within the line. In fact, ε_3 is the minimum line width such that every beam is entirely within the line because at this line width the beam segment between the inner boundary and the resonance point is certainly within the line.

For an unbounded atmosphere, the total optical depth of the resonance region becomes arbitrarily large—regardless of ε—as the beam direction becomes radial. For a bounded atmosphere, however, the total optical depth is finite for every value of θ so long as ε is positive. Figure 2.23 illustrates that even for small ε and small θ, the total optical depth of the resonance region is finite.

Like Figure 2.13, Figures 2.24 and 2.25 show the angular dependence of τ for several values of w and for $\varepsilon = 0.005$. In Figure 2.24, the resonance point's location w varies from a value just slightly greater than $w_I = 0.5$ to

$$w_1 = w_O \sqrt{[\alpha - 1] \frac{\sqrt{\varepsilon^2[\alpha^2 - 1] + 1} - \varepsilon\alpha}{\alpha\sqrt{\varepsilon^2[\alpha^2 - 1] + 1} - \varepsilon[\alpha^2 - 0.5]}}, \tag{2.40}$$

where

$$\alpha = \frac{w_O^2 + w_I^2}{w_O^2 - w_I^2}. \tag{2.41}$$

When the resonance point is located at w_1, each of the two direction angles at which the outbound beam becomes entirely within the line is the same as the direction angle at

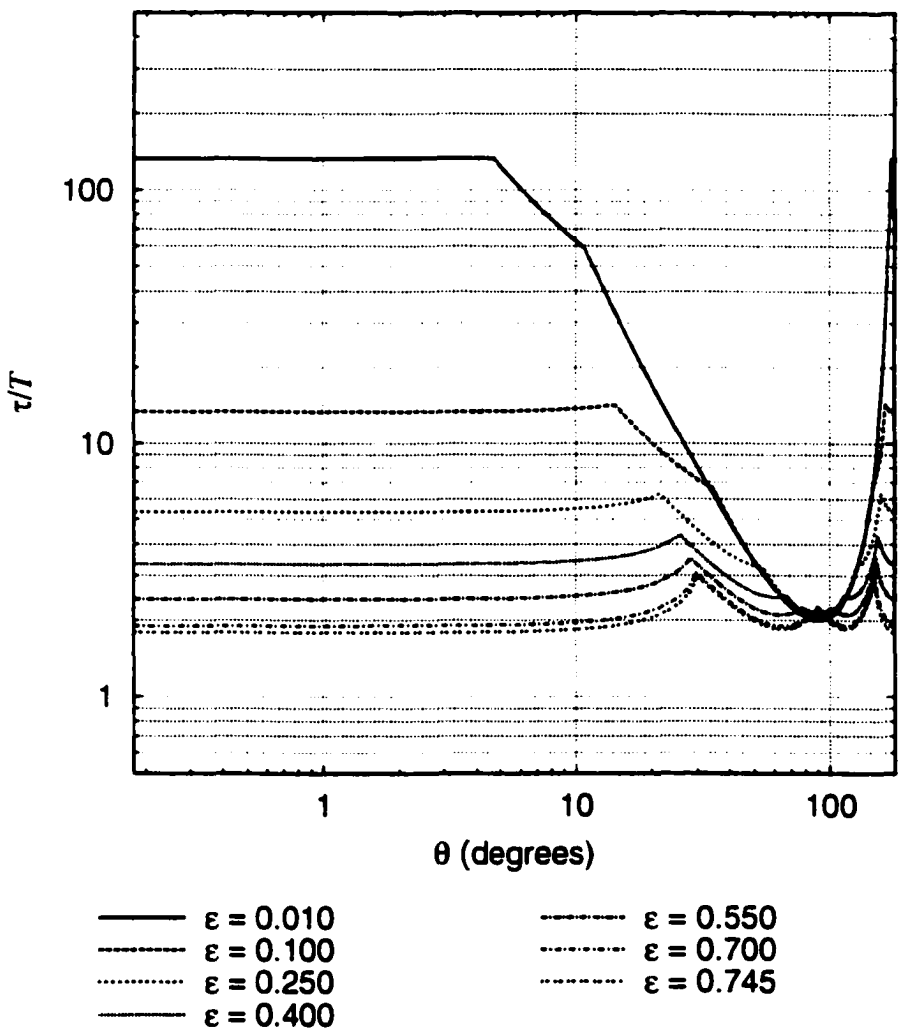


Figure 2.23: Like Figure 2.12, but with atmospheric boundaries, $w_{\mathrm{I}} = 0.5$ and $w_{\mathrm{O}} = 1.5$.

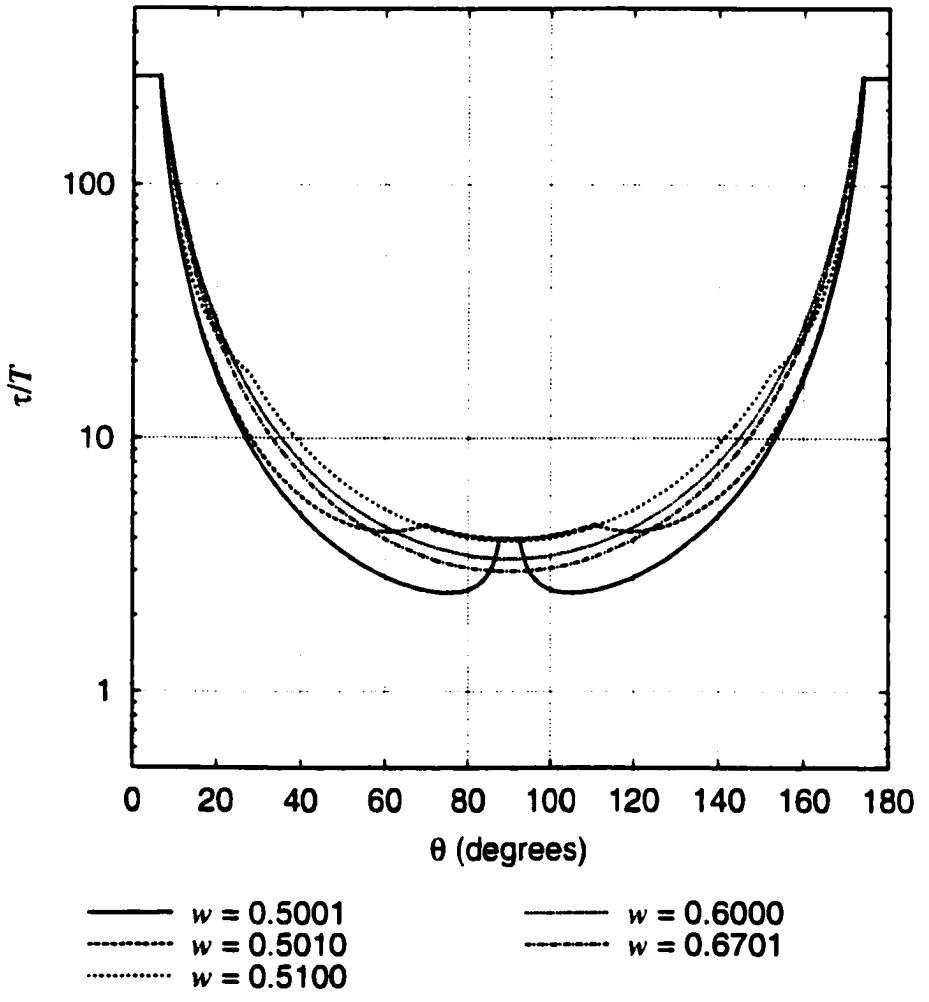


Figure 2.24: Like Figure 2.13, but with atmospheric boundaries, $w_{\mathrm{I}} = 0.5$ and $w_{\mathrm{O}} = 1.5$.

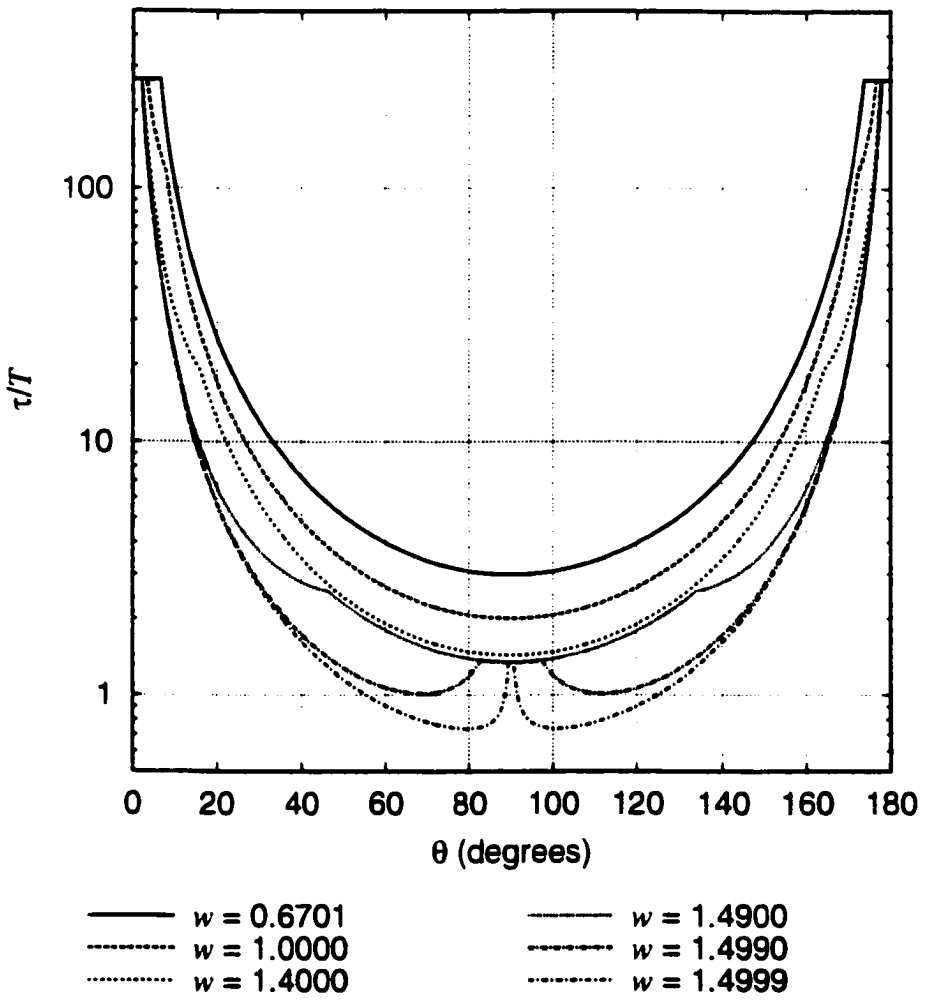


Figure 2.25: Like Figure 2.13, but with atmospheric boundaries, $w_I = 0.5$ and $w_O = 1.5$.

which the corresponding inbound beam becomes entirely within the line. For $w_I = 0.5$ and $w_O = 1.5$, $w_1 \approx 0.67$. The curve corresponding to $w = w_1$ has only two cusps; every other curve has four cusps. In Figure 2.24, each of the central cusps corresponds to an angle at which the impact-parameter side of the beam becomes entirely within the line; each of the lateral cusps corresponds to an angle at which the anti-impact-paramter side of the beam becomes entirely within the line. Each of the curves corresponding to one of the two resonance points closest to the inner boundary shows a decrease in optical depth just outside of the central cusp; although the impact-parameter side of the beam passes through a region of higher opacity as the beam becomes more radial, for a resonance point sufficiently close to the inner boundary the path length between the inner boundary and the resonance point at first decreases rapidly enough to dominate the behavior of the total optical depth.

In Figure 2.25, the meaning of the central cusps and the meaning of the lateral cusps are swapped with respect to the meanings in Figure 2.24. Figure 2.25 begins the sequence of w-curves where Figure 2.24 ends. In this case one clearly sees that the curve for $w = w_1$ only has two cusps. As in Figure 2.24, each curve corresponding to one of the two resonance points closest to the boundary—in this case, the outer boundary—shows a decrease in total optical depth just outside of the central cusp. Here, close proximity to the outer boundary leads to a rapid decrease in the beam path length between the outer boundary and the resonance point as the beam becomes radial, and again this decrease at first dominates the behavior of the total optical depth.

Like Figure 2.14, Figure 2.26 shows contours of constant optical depth $\tau/T = 0.1$ (for $w = 1$), but for the bounded case ($w_I = 0.5$ and $w_O = 1.5$). Unlike the open contours in Figure 2.14, each contour in the bounded case encloses a finite area on the graph. The shape and location of each of the inner and outer boundaries clearly influence the shape of each contour, which for $T = 1$ roughly marks the locus of points at which the beam

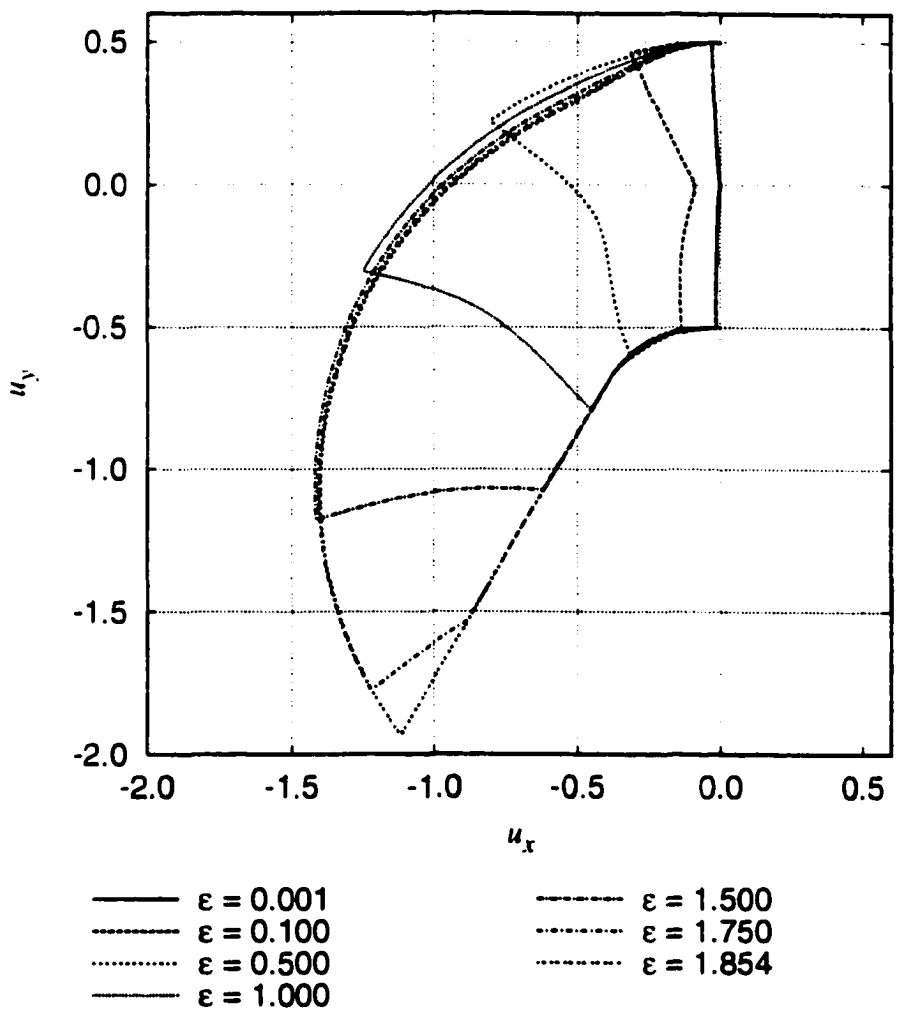


Figure 2.26: For the incoming beam in a bounded atmosphere, a contour of constant optical depth $\tau/T = 0.1$ (for $w = 1$) is plotted for each of several values of ε. The resonance point is at the origin, and the center of the velocity field is at $(0, -1)$. Compare with Figures 2.8 and 2.14.

begins to interact with the line. Each contour in the bounded case roughly resembles a truncated version of the corresponding contour in the unbounded case.

Like Figure 2.26, Figure 2.27 shows contours of constant optical depth but for $\tau/T = 1$. For $T = 1$, the contour outlines the core of the resonance region, for the beam is already strongly scattering as it approaches the resonance point. The presence of the boundaries accounts for significant distortion from the unbounded case.

Figure 2.28 shows contours of constant optical depth for a very narrow ($\varepsilon = 0.005$) line in a bounded atmosphere. For a narrow line, the various contours, as in the unbounded case, are roughly identical to each other, and the contour in the bounded case is basically a truncated version of the contour in the unbounded case.

Figure 2.29 magnifies the plot of Figure 2.28 in order to show behavior near the resonance point. Note that the contour corresponding to $t/T = 1.0$ just touches the origin because, apparently, $\varepsilon = 0.005$ is a good approximation of a delta function line, and T is the optical depth of a delta function line when the beam's closest approach to the center of the system is at the resonance point.

Figure 2.30 illustrates the narrow-line shape of the resonance region for different values of w. Again we see that the narrow-line contours for the bounded case have the appearance of those for the unbounded case except for truncation at the boundaries.

Although the imposition of an inner boundary and an outer boundary have significant effects on the nature of the resonance region for large line widths, the nature of the resonance region is hardly affected by the presence of shell boundaries for small line widths. That truncation of the resonance region by the outer boundary changes the region's extent from infinite to finite and that truncation by the inner boundary removes a singularity are small effects for a sufficiently narrow line because those effects only involve a tiny piece of direction-angle space. Indeed, as the width of the line goes to zero, the effect of atmospheric boundaries on the resonance region disappears

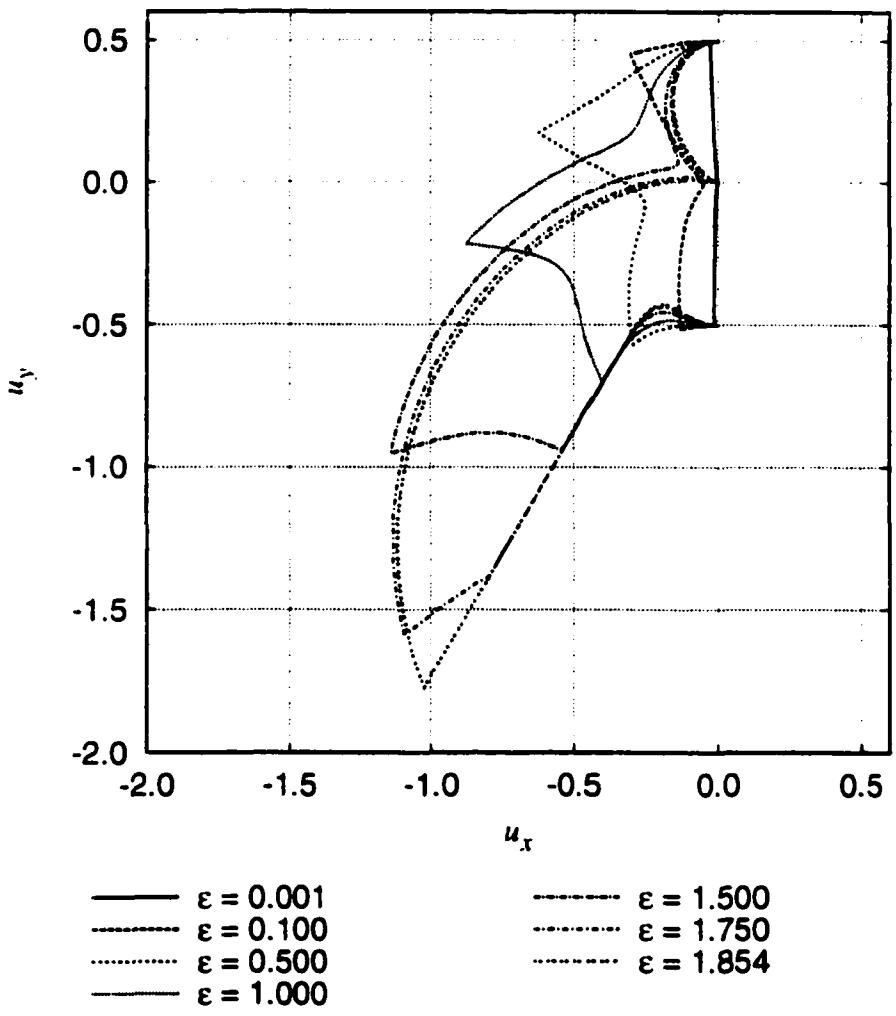


Figure 2.27: Like Figure 2.26, but for $\tau/T = 1$.

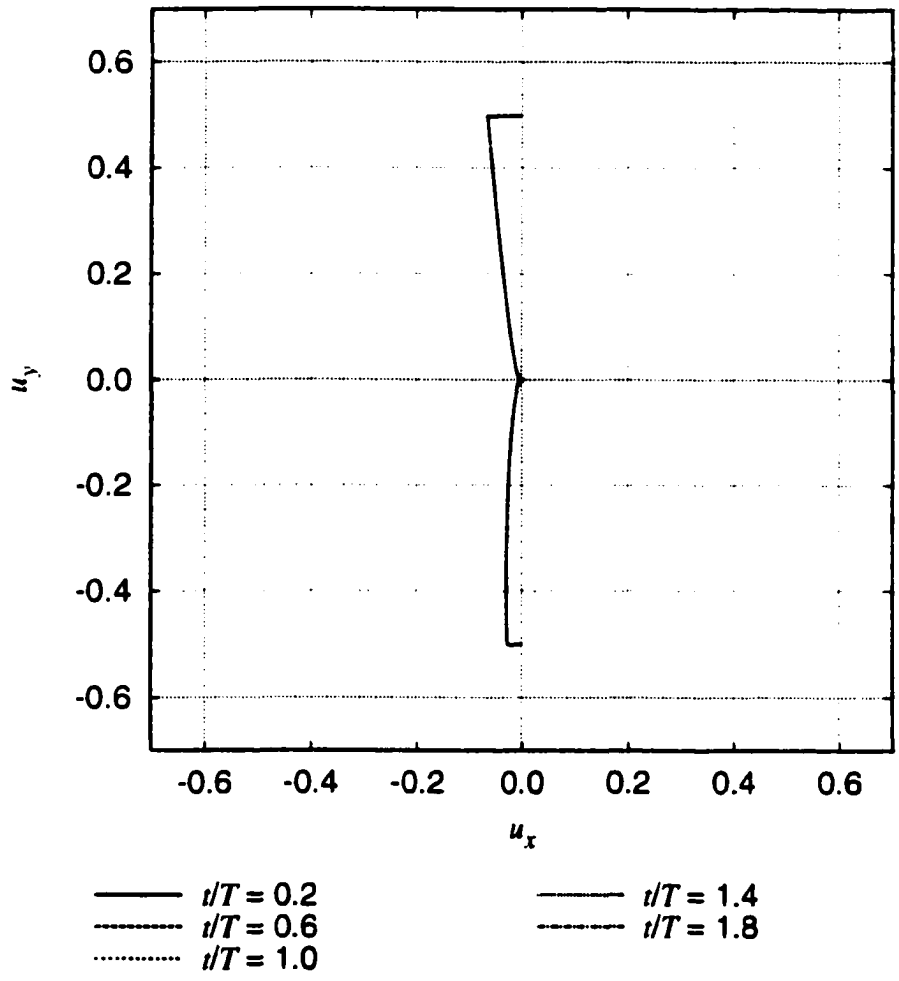


Figure 2.28: For the incoming beam in a bounded atmosphere, a contour of constant optical depth (for $\varepsilon = 0.005$ and $w = 1$) is plotted for each of several values of normalized optical depth t/T. The resonance point is at the origin, and the center of the velocity field is at $(0, -1)$. Although several contours are plotted, they overlap completely on this scale. Compare with Figure 2.16.

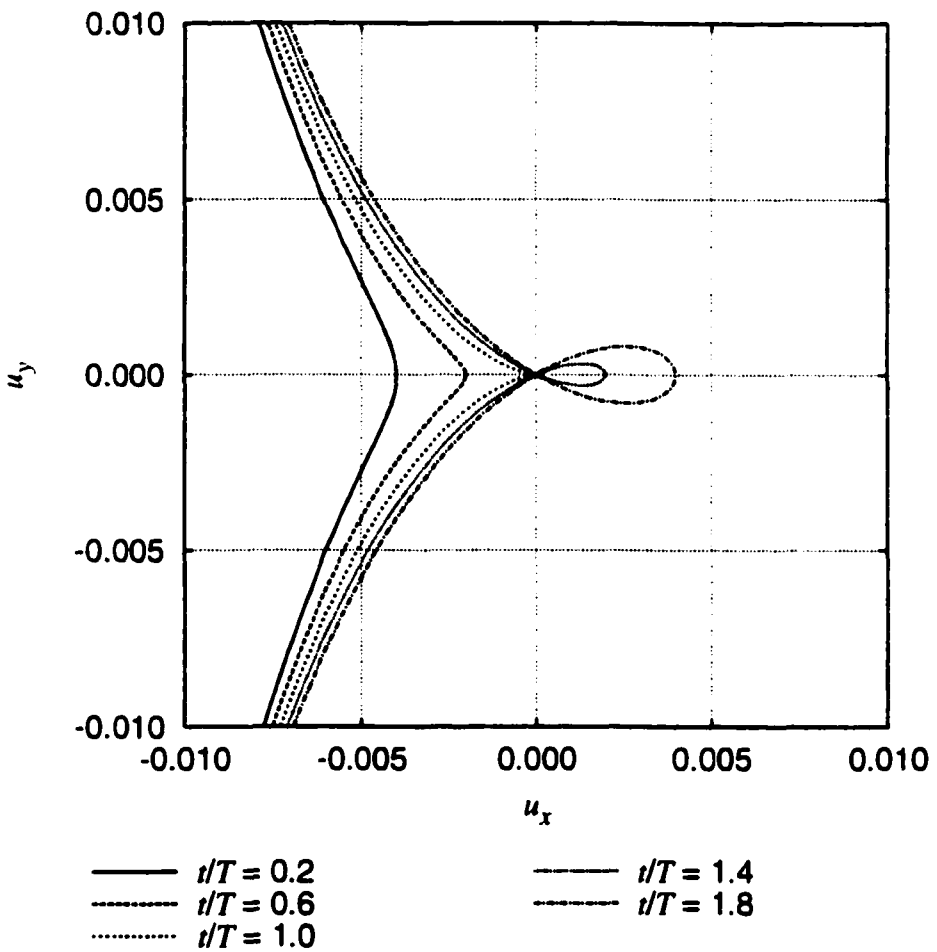


Figure 2.29: Like Figure 2.28, but a close-up view near the resonance point.

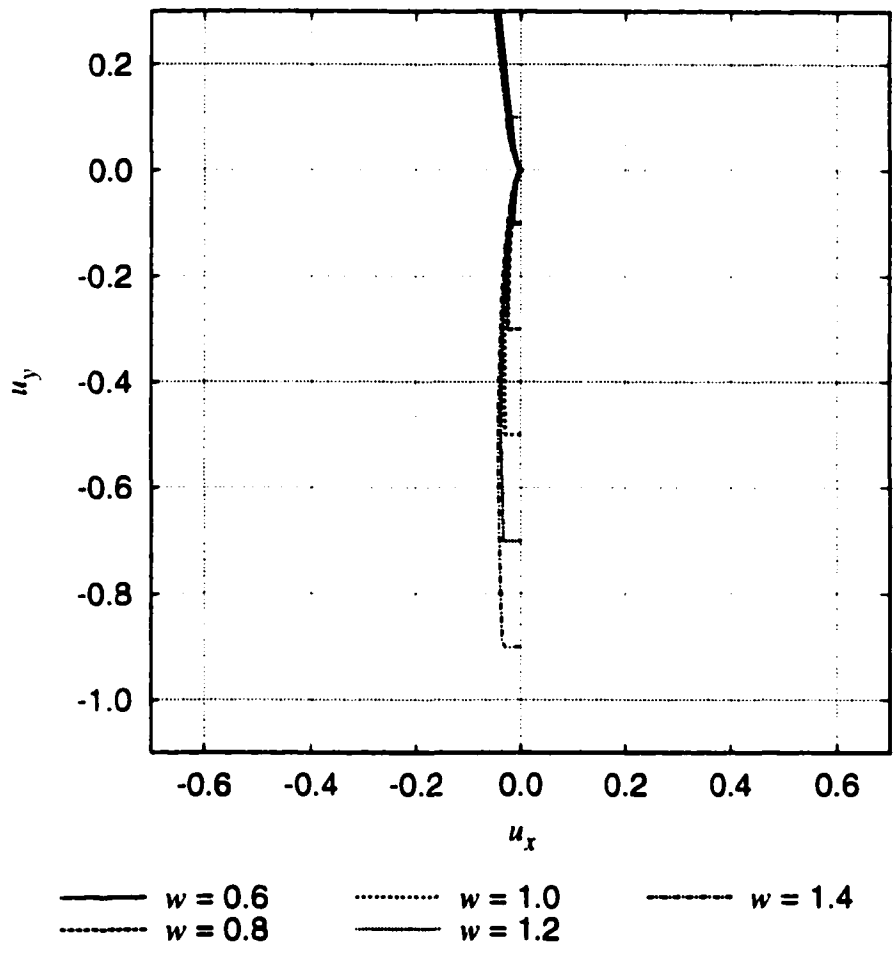


Figure 2.30: Like Figure 2.28, but with each contour corresponding to $t/T = 1$ for a particular value of w. The center of the velocity field is now located at $(0, -w)$.

completely (and there is even the reappearance of infinite optical depth along the radial direction). This allows for a relatively simple treatment of line radiative transfer in the next chapter.

Chapter 3

Sobolev Method for the Constant-Speed Radial Wind

In an atmosphere with power-law radial velocity, a delta-function line profile (together with the assumption of complete redistribution in wavelength) is sufficient but not always necessary for the application of the Sobolev method. In the Sobolev method, a beam directed toward the observer accumulates, at only one point in the atmosphere, all of the optical depth that a given line has to offer. Usually a sufficiently small but finite line width will guarantee that this is a good approximation. For the constant-speed wind, however, a delta-function line profile is necessary in order to evaluate certain integrals over all solid angle.

If the power-law index, a, is positive—as it is, for example, in a homologously expanding atmosphere—then the velocity gradient is positive for every beam direction away from a resonance point. A sufficiently narrow line allows for the application of the Sobolev method because for every beam direction through the resonance point the resonance region will, in correspondence with the sufficient narrowness of the line, be

sufficiently small so that physical conditions are approximately constant across the resonance region. Note that for a given integrated line opacity, the Sobolev method optical depth of the resonance region still depends on the width of the line. A delta-function line profile is associated with a particular limiting value of the resonance region's optical depth but is in no way required.

If $a < 0$, then there will always be beam directions for which the velocity gradient at the resonance point is (locally) zero, but a sufficiently narrow line will still be associated with a resonance region that is small in every direction. A single beam, however, in this case may have a pair of resonance points, and so there is extra complexity built into the analysis from the beginning.

Now the constant-speed wind, for which $a = 0$, has a radial velocity gradient that is globally (and not merely locally) zero. For the radial direction, the resonance point becomes a radial resonance ray, whose end point is the inner boundary of the line-forming region. A line with finite width—even an arbitrarily small width—produces a resonance region which for finite solid angle extends far enough to violate the assumption that physical properties are approximately constant across the region. Only a delta-function line profile reduces the violation's corresponding solid angle to zero (and still there is a singularity in the radial direction).

3.1 Integration of the Transfer Equation

Let the source function S be defined such that $S(w)$ is the source at radial distance $r_0 w$ from the center of the system. For a beam whose resonance point (with the line centered at λ') is located at a distance $r_0 w$ from the center of the system and whose direction angle at the resonance point is θ, let the intensity function I be defined such that $I(w, \theta, u)$ is the intensity at distance $r_0 u$ from the resonance point; $I(w, \theta, u)$ corresponds to light

traveling, for $u < 0$, toward the resonance point and, for $u > 0$, away from the resonance point. Then, in terms of symbols defined in the previous chapter, the radiative transfer equation becomes

$$\frac{1}{r_0}\frac{\partial I(w,\theta,u)}{\partial u} = K_{-2}(\,R(r_0u)\,)\,\phi(\,\Lambda(u)\,)\left[S\left(\frac{R(r_0u)}{r_0}\right) - I(w,\theta,u)\right]. \qquad (3.1)$$

For a line whose profile is the rectangular impulse of equation 2.23, the transfer equation becomes

$$\frac{\partial I(w,\theta,u)}{\partial u} = \begin{cases} \left[\frac{r_0}{R(r_0u)}\right]^2 \frac{r_0K_0}{2\varepsilon}\left[S\left(\frac{R(r_0u)}{r_0}\right) - I(w,\theta,u)\right] & \text{if } U_0(-\varepsilon,\theta) \le u \le U_0(\varepsilon,\theta) \\ 0 & \text{otherwise} \end{cases}. \qquad (3.2)$$

Here the function U_0 has been redefined so that the dependence on angle is now explicit. Let $\theta_C(w)$ be half of the angle subtended by an opaque central sphere, of radius w_C, that emits wavelength-independent continuum intensity, I_0. $\theta_C(w)$ is distinguished from $\theta_I(w)$, which is the angle subtended by the inner boundary of the line-forming shell. There is the explicit allowance for a vacuum gap between the emissive sphere and the shell, and so $w_C \le w_I$. If $\theta \le \theta_C(w)$, then $I(w,\theta,u) = I_0$ for $u < U_0(-\varepsilon,\theta)$; if, however, $\theta > \theta_C(w)$, then $I(w,\theta,u) = 0$ for $u < U_0(-\varepsilon,\theta)$.

There is no sufficiently small ε such that, for every non-radial direction, $R(r_0u)$ and $S(\,R(r_0u)/r_0\,)$ have small variation as u varies from $U_0(-\varepsilon,\theta)$ to $U_0(\varepsilon,\theta)$. Nevertheless, for each non-radial direction there is a sufficiently small ε (which might need to be smaller for one direction than for another) such that $R(r_0u)$ and $S(\,R(r_0u)/r_0\,)$ have small variation as u varies from $U_0(-\varepsilon,\theta)$ to $U_0(\varepsilon,\theta)$. So in order to apply the Sobolev method, one must consider the *limit* of small ε, which corresponds to a delta-function line profile. Because, *especially* for arbitrarily small ε, $I(w,\theta,u)$ changes rapidly for u near zero, the limit may not be evaluated explicitly until after the differential equation

has been integrated. For each non-radial direction, then, there is a sufficiently small ε such that, to a good approximation,

$$\frac{\partial I(w,\theta,u)}{\partial u} \simeq \frac{r_0 K_0}{2w^2\varepsilon}[S(w) - I(w,\theta,u)] \tag{3.3}$$

for

$$U_0(-\varepsilon,\theta) \simeq -\varepsilon w \csc^2\theta < u < \varepsilon w \csc^2\theta \simeq U_0(\varepsilon,\theta). \tag{3.4}$$

Defining

$$\tilde{I}(u) = S(w) - I(w,\theta,u) \tag{3.5}$$

simplifies the transfer equation so that

$$\tilde{I}'(u) \simeq -\frac{r_0 K_0}{2w^2\varepsilon}\tilde{I}(u). \tag{3.6}$$

Integrating and substituting, one finds

$$I(w,\theta,u) \simeq S(w)\left[1 - e^{-\frac{r_0 K_0\left[u+\varepsilon w\csc^2\theta\right]}{2w^2\varepsilon}}\right] + I\left(w,\theta,-\varepsilon w\csc^2\theta\right)\, e^{-\frac{r_0 K_0\left[u+\varepsilon w\csc^2\theta\right]}{2w^2\varepsilon}}, \tag{3.7}$$

and

$$I\left(w,\theta,\varepsilon w\csc^2\theta\right) \simeq S(w)\left[1 - e^{-\frac{r_0 K_0}{w\sin^2\theta}}\right] + I\left(w,\theta,-\varepsilon w\csc^2\theta\right)\, e^{-\frac{r_0 K_0}{w\sin^2\theta}}. \tag{3.8}$$

Now the limit of small ε may be taken, and the result is

$$I_+(w,\theta) = S(w)\left[1 - e^{-\tau(w,\theta)}\right] + I_-(w,\theta)\, e^{-\tau(w,\theta)}, \tag{3.9}$$

where

$$\tau(w,\theta) = \frac{r_0 K_0}{w \sin^2\theta} = \frac{2T}{w \sin^2\theta} \tag{3.10}$$

is, in agreement with equation 2.29 for a delta-function line profile, the optical depth of the resonance point at radial parameter w; $I_-(w,\theta)$ is the intensity incident on the resonance point; $I_+(w,\theta)$ is the intensity emergent from the resonance point.

3.2 Source Function for a Two-Level Atom

The line mean intensity function J is defined such that $J(w)$ is the mean intensity, averaged over solid angle and wavelength within the width of a vanishingly narrow line, at radial coordinate w. This mean intensity can be expressed as an integral over direction angle. For each beam trajectory, the integrand must add together the intensity contribution, weighted by the line absorption profile, from every distinct monochromatic beam that can interact with the line at the point corresponding to w. The integrand of the integral over direction angle must be evaluated in the limit of vanishing line width so that the integrand has the appropriate value for every non-radial direction.

$$J(w) = \frac{1}{2}\int_0^\pi \lim_{\varepsilon\to 0}\left[\frac{1}{2\varepsilon}\int_{-\varepsilon}^{\varepsilon} I\left(\frac{R(r_0 U_0(z,\theta))}{r_0}, \arcsin\frac{r_0 w \sin\theta}{R(r_0 U_0(z,\theta))}, U_0(z,\theta)\right) \mathrm{d}z\right] \sin\theta \,\mathrm{d}\theta \tag{3.11}$$

The integral over the line profile incorporates a contribution from distinct monochromatic beams that share the same trajectory. Each distinct beam has its resonance point with the center of the line profile at a distinct radial coordinate,

$$w' = \frac{R(r_0 U_0(z,\theta))}{r_0}, \tag{3.12}$$

and at that resonance point there is a distinct direction angle,

$$\theta' = \arcsin \frac{r_0 w \sin\theta}{R(r_0 U_0(z,\theta))}, \tag{3.13}$$

for the same beam trajectory.

As the normalized line width ε becomes very small, the intensity's variation due to the location of the beam's resonance point also becomes very small. Although the variation in the value of the intensity function always remains substantial as its third argument, $U_0(z,\theta)$, changes, the third argument may be expressed as $zw\csc^2\theta$ in the limit of small ε. The expression for the line mean intensity may thus be greatly simplified.

$$J(w) = \frac{1}{2}\int_0^\pi \lim_{\varepsilon\to 0}\left[\frac{1}{2\varepsilon}\int_{-\epsilon}^{\epsilon} I\left(w,\theta,\frac{zw}{\sin^2\theta}\right)\,\mathrm{d}z\right]\sin\theta\,\mathrm{d}\theta \tag{3.14}$$

The use of equation 3.7 leads to an innermost integrand whose variation is simply exponential with z. The exponentials are easily integrated, and the result is

$$J(w) = [1-\beta(w)]\,S(w) + \beta_0(w)\,I_0, \tag{3.15}$$

where

$$\beta(w) = \frac{1}{2}\int_0^\pi \frac{1-e^{-\tau(w,\theta)}}{\tau(w,\theta)}\sin\theta\,\mathrm{d}\theta, \tag{3.16}$$

and

$$\beta_0(w) = \frac{1}{2}\int_0^{\theta_c(w)} \frac{1-e^{-\tau(w,\theta)}}{\tau(w,\theta)}\sin\theta\,\mathrm{d}\theta. \tag{3.17}$$

If the source of opacity is modeled as a two level atom, then the rate equation for the transition can be written as

$$S(w) = [1-\varepsilon]\,J(w) + \varepsilon\,B(w), \tag{3.18}$$

where ε is the collisional fraction of de-excitations, and $B(w)$ is the thermal blackbody intensity in the comoving frame, at line center, and for radial parameter w. In combination with equation 3.15, this expression becomes

$$S(w) = \frac{[1-\varepsilon]\,\beta_0(w)\,I_0 + \varepsilon\,B(w)}{[1-\varepsilon]\,\beta(w) + \varepsilon}. \tag{3.19}$$

For each radial point in the model, this two-level-atom source function can be calculated from the model parameters, w_C, T, ε, and B, alone.

3.3 The Line Profile

A synthetic line profile is the calculation of what an observer would measure if the observer were to point a perfect telescope at a perfect realization of the model and to collect the light with a perfect spectrometer. Because the value of the source function is known at every radial point in the shell, the computation of the synthetic line profile becomes an exercise in geometry. The geometric exercise is simplified by the assumption that if there is more than one line in the shell, the minimum wavelength separation of a pair of lines is larger than the largest doppler shift obtainable in the shell. For the purpose of this discussion, such a simplification is equivalent to the assertion that there is only one line in the shell.

3.3.1 Geometry

Consider an opaque sphere that emits a continuum of light, and let this sphere be surrounded by a concentric spherical shell that is expanding with constant radial velocity v. In units of r_0 (the length scale for the radial diminution of the integrated line opacity), the radius of the emissive sphere is w_C; the radius of the inner boundary of the shell is w_I; and the radius of the outer boundary of the shell is w_O. In the shell, absorption,

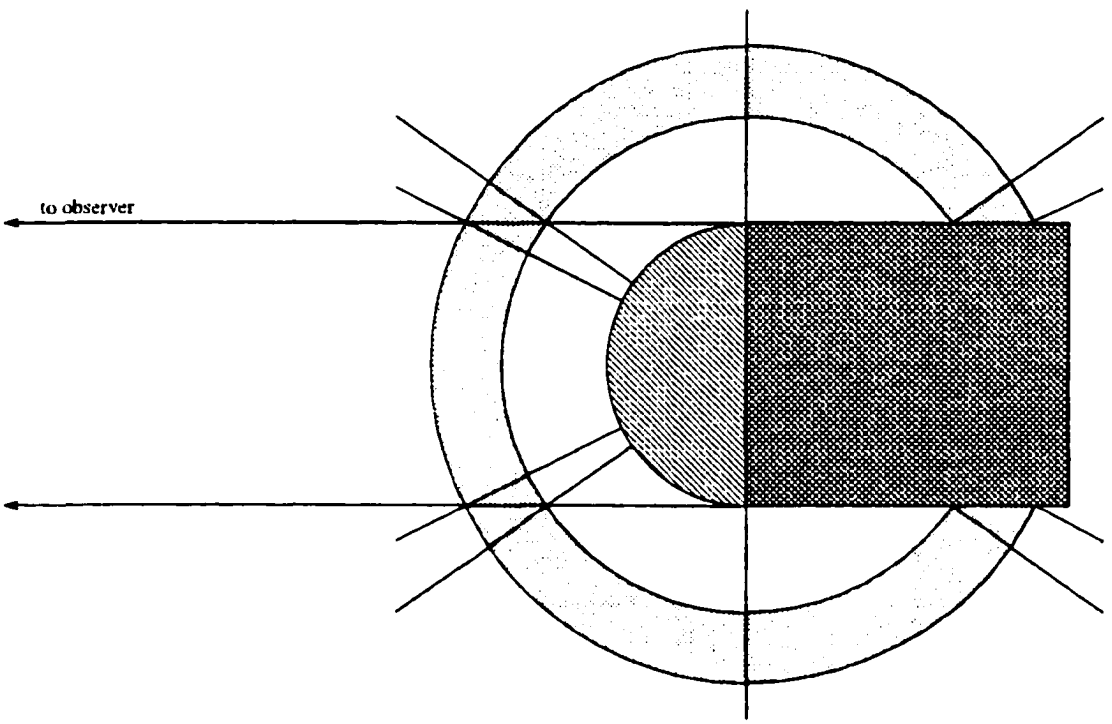


Figure 3.1: Three concentric spheres characterize the basic geometry of the scattering model. The innermost sphere (singly hatched) is opaque and emits light. Behind this sphere is an occulted region (doubly hatched) that is invisible to the observer. Between the two outer spheres is a shell (shaded) in which scattering takes place. A beam with a particular observer-frame wavelength will interact with material in the shell if it intersects within the shell a cone corresponding to that wavelength. Five cones (including the central, degenerate one that is a plane) are drawn.

scattering, and thermal emission occur at the line wavelength λ' in the comoving frame. The observer is at rest with respect to the center of spherical symmetry, and the observer is located far enough from the shell so that every line of sight through the shell shares essentially the same direction. Then, as indicated by the doubly hatched region in Figure 3.1, there is an occulted region from which light can not travel directly to the observer. Outside of this occulted region, light traveling from the emissive sphere and not toward the observer may be scattered toward the observer at some point in the shell. Also, light traveling from the emissive sphere and toward the observer may be scattered away from the line-of-sight direction as it passes through the shell.

Consider a beam that passes through the shell and that is parallel to the line of sight. This beam has at most one resonance point with the line in the shell. Consider an ensemble of such beams that that are distributed uniformly through space. If every beam shares the same observer-frame wavelength λ, and if λ is sufficiently near λ'—the wavelength of the line in the observer's frame of reference (and in a frame locally comoving with a mass element in the shell)—then the locus of resonance points is the intersection of the shell and a cone. The cone's apex is the center of the emissive sphere; the cone's axis is the line of sight; and the cone's half-angle is $\arccos([1 - \lambda/\lambda']c/v)$.

The line source function and optical depth at points on each cone, along with the intensity of the light emitted by the opaque sphere, completely determine the intensity that the observer measures. The source function, computed at the comoving line wavelength λ', can be used to calculate the amount of light emitted by the cone into the line of sight. Because every point on the cone has the same velocity component along the line of sight, light emitted at wavelength λ' from any point on the cone into the line of sight is collected by the observer at wavelength λ. Similarly, the optical depth, computed at the line wavelength λ', can be used to calculate the amount of light scattered out of the line of sight. For a beam emitted at observer-frame wavelength λ, from the

opaque sphere, and toward the observer, light is scattered out of the line of sight at the intersection of the beam and the cone, on which in the comoving frame the beam has wavelength λ'.

The general procedure for calculating a value proportional to the observed flux at wavelength λ is as follows:

- Calculate the angle of the cone corresponding to λ.
- At least some part of the surface of the emissive sphere is unobscured by the cone. Multiply I_0, the emissive sphere's intensity at λ by the area that results from the projection of the unobscured part the sphere onto the plane perpendicular to the line of sight. Store this product in the observed flux bin corresponding to λ.
- If any part of the surface of the emissive sphere is obscured by the cone, then integrate $S(w)[1 - e^{-\tau(w,\theta)}] + I_0 e^{-\tau(w,\theta)}$ (where w is the radial coordinate and θ is the direction angle of the observer's line of sight for a resonance point on the cone) over the projected area of the obscuring part of the truncated cone. Add this integral to the observed flux bin corresponding to λ.
- If any part of the truncated cone does not obscure the surface of the emissive sphere, then integrate $S(w)[1 - e^{-\tau(w,\theta)}]$ over the projected area of the non-obscuring part of the truncated cone. Add this integral to the observed flux bin corresponding to λ.

This procedure is carried out for every wavelength point on a linear grid in the observer frame. The result is a line profile.

3.3.2 Pure Scattering

If x, the collisional fraction of de-excitations, is zero, then none of the radiation emitted by the opaque sphere is absorbed in the shell, and so the line profile results from *pure scattering*. The source function becomes

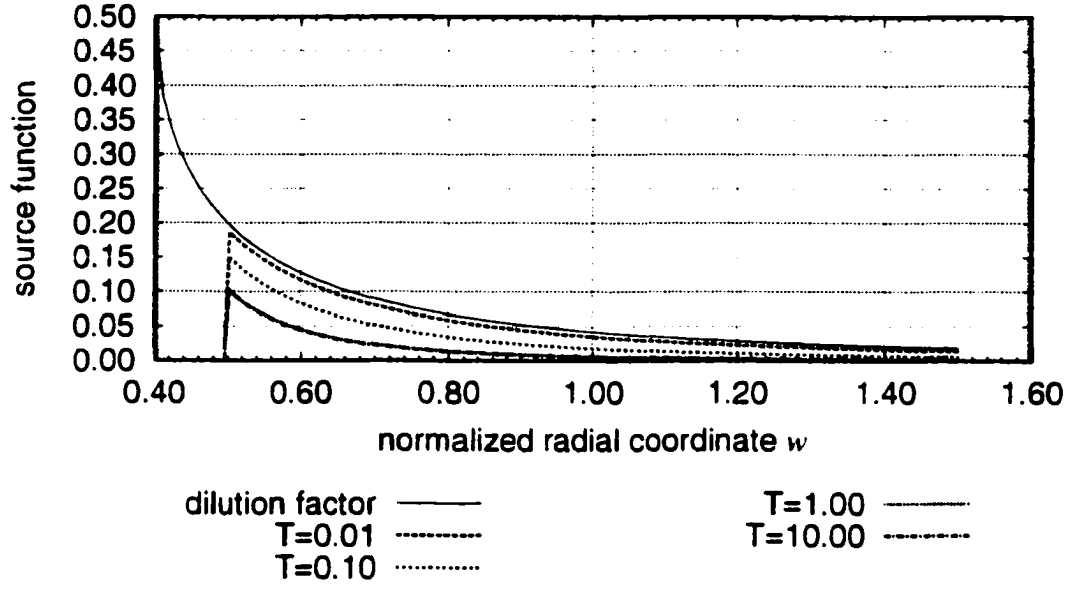
$$S(w) = \frac{\beta_0(w) I_0}{\beta(w)}. \tag{3.20}$$

Recall that each wavelength point in the synthetic line profile corresponds to a particular cone. Let a cone's half-angle be α, and let α near zero correspond to narrow cones opening toward the observer. The cone with a half-angle $\alpha = \pi/2$ is really a plane, and its intersection with the shell is an annulus that shines at λ' (line center) in the observer frame. In general, the intersection is a truncated cone that shines at $[1 - v\cos(\alpha)/c]\lambda'$. The *continuum* level of the pure-scattering synthetic line profile is the observed flux that corresponds to a completely unobscured view of the emissive sphere. For every half-angle α such that $\arcsin(w_C/w_I) < \alpha < \pi - \arcsin(w_C/w_O)$, the cone does not obscure the observer's view of the emissive sphere. Nevertheless, because the cone glows—that is, it scatters light into the line of sight along beams with impact parameter greater than w_C—the observer measures more than the continuum flux for the corresponding wavelength, and so the observed line profile is in emission at this wavelength. For every half-angle α such that $0 < \alpha < \arcsin(w_C/w_O)$, the truncated cone only obscures the observer's view of the emissive sphere. Because the cone scatters away from the line of sight some of the light originally headed directly toward the observer from the emissive sphere, the observer measures less than the continuum flux for the corresponding wavelength, and so the observed line profile is in absorption at this wavelength. At some wavelength corresponding to α between $\arcsin(w_C/w_O)$ and $\arcsin(w_C/w_I)$ the flux measured by the observer crosses the continuum level. For this

range of α values, part of the truncated cone diminishes the observed flux by obscuring the emissive sphere, but part of the truncated cone augments the observed flux by scattering light into the line of sight along beams with impact parameter greater than w_C. Finally, for half-angles greater than $\pi - \arcsin(w_C/w_O)$, each cone is occulted by the opaque sphere, and so the observer measures exactly the continuum flux at the corresponding wavelength.

For each of several line strengths T, Figure 3.2 shows source functions and line profiles. The geometry of the shell is the same as that used in the plots in the previous chapter: $w_I = 0.5$ and $w_O = 1.5$. The radius of the emissive sphere is $w_C = 0.4$, and so there is a small vacuum gap between the emissive sphere and the inner boundary of the shell. For a very weak line, the source function, expressed in units of the intensity emitted by the central sphere, is nearly the dilution factor, $W(w) = 0.5\left[1 - \sqrt{1 - [w_C/w]^2}\right]$, which represents the pure-scattering source function if the optical depth is independent of direction angle. Although the optical depth τ is still certainly dependent on the direction angle for small T, nevertheless τ can be small over a wide range of direction angles, and this results in the weak-line behavior of the source function. The angular dependence of τ causes the value of the source function to become smaller at any given radial coordinate as T grows larger. Beyond $T = 1$, however, the effect is not noticeable; the source function for $T = 10$ is indistinguishable from the source function for $T = 1$. Because the source function is the ratio of the local emissivity to the local opacity, the interpretation is that for $T < 1$ the emissivity climbs more slowly than the opacity, and that for $T > 1$ the emissivity is proportional to the opacity.

The behavior of the line profile reflects that of the source function for strong lines because for a strong line the only contribution to the observed flux from an obscured portion of the emissive sphere is just the source function. Note that the horizontal coordinate z of the line profile is such that -1 corresponds to the bluest wavelength



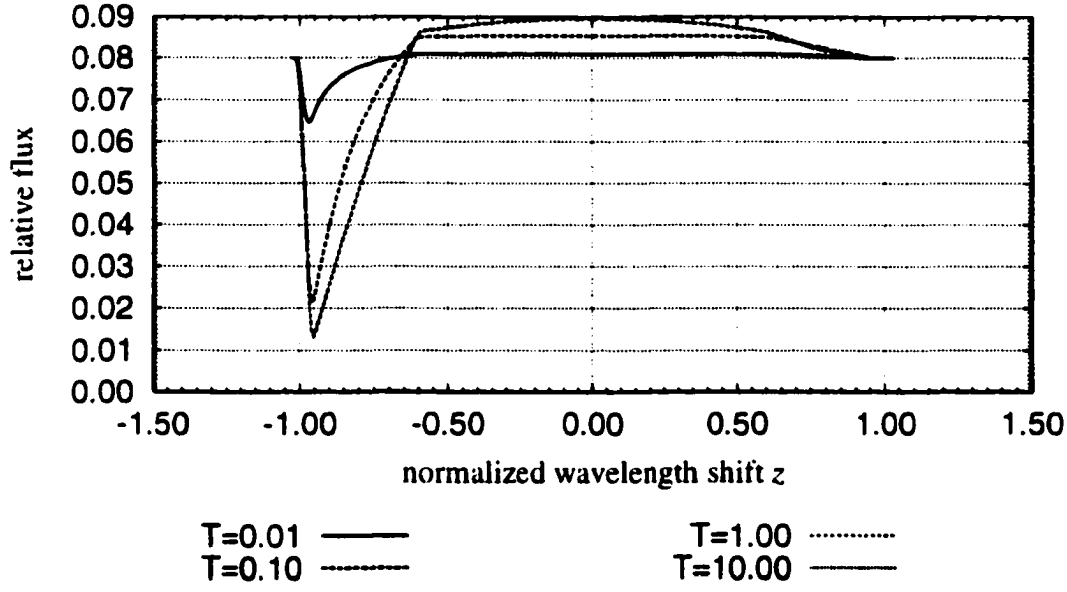
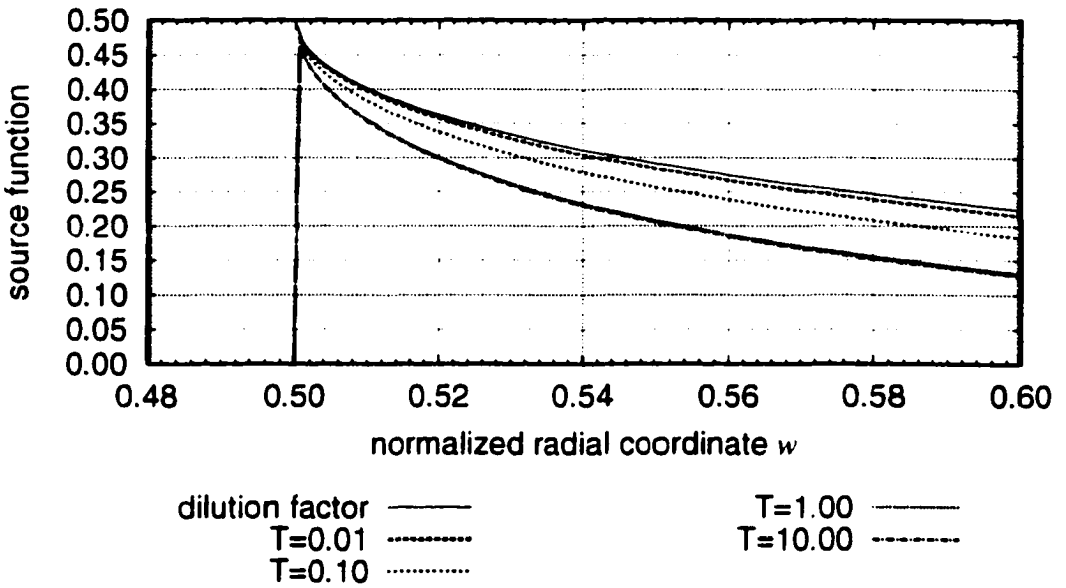


Figure 3.2: For $w_C = 0.4$ and for the standard shell geometry used in the previous chapter ($w_I = 0.5$ and $w_O = 1.5$), the source function and the line profile for each of several values of the line strength parameter T are shown. The dilution factor and the source functions are plotted in the top graph. The line profiles are plotted in the bottom graph.

that the line can produce, and +1 corresponds to the reddest wavelength that the line can produce. The rest-frame wavelength of the line corresponds to $z = 0$. The existence of the vacuum gap between the emissive sphere and the shell produces a cusp in the line profile because as the observed wavelength increases beyond a certain critical value, obscuration of the emissive sphere suddenly ceases. The shape of the blue absorption trough depends on the competition between two things. The decrease of the cone's half-angle α toward zero leads (1) to the increase in a cone's optical depth along the observer's line of sight but (2) to the decrease in the fraction of the emissive sphere obscured by the cone. Blueward of the minimum, the fractional coverage dominates the observed flux.

Figure 3.3 shows the same curves as Figure 3.2 but for different geometric parameters. The emissive sphere has been enlarged so that the vacuum gap has been eliminated, and the outer boundary of the shell has been reduced so that $w_O = 0.6$. The cusp has disappeared because, as the cone angle increases, (1) the fractional coverage of the emissive sphere by the cone decreases to zero more slowly than in the case of the gap, and (2) the optical depth along the line of sight decreases further than in the case of the gap as the coverage goes to zero. The minimum of the absorption trough has moved redward because the shell is thinner, and so, for larger values of α, the fractional coverage of the emissive sphere by the cone drops rapidly with α. The reduction of the thickness of the shell also renders the effect of the weakest line almost unobservable. Because the ratio of w_O to w_C is near unity, there is a large flat region of continuum flux at the red extreme of the line profile. This results from the complete occultation of cones over a relatively large range of α up to π.



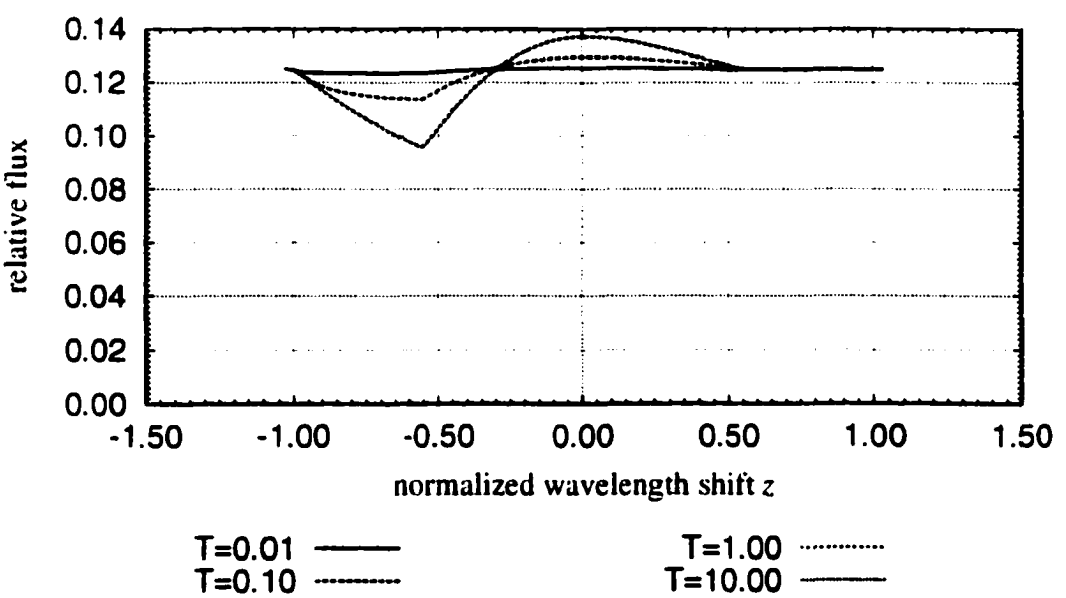
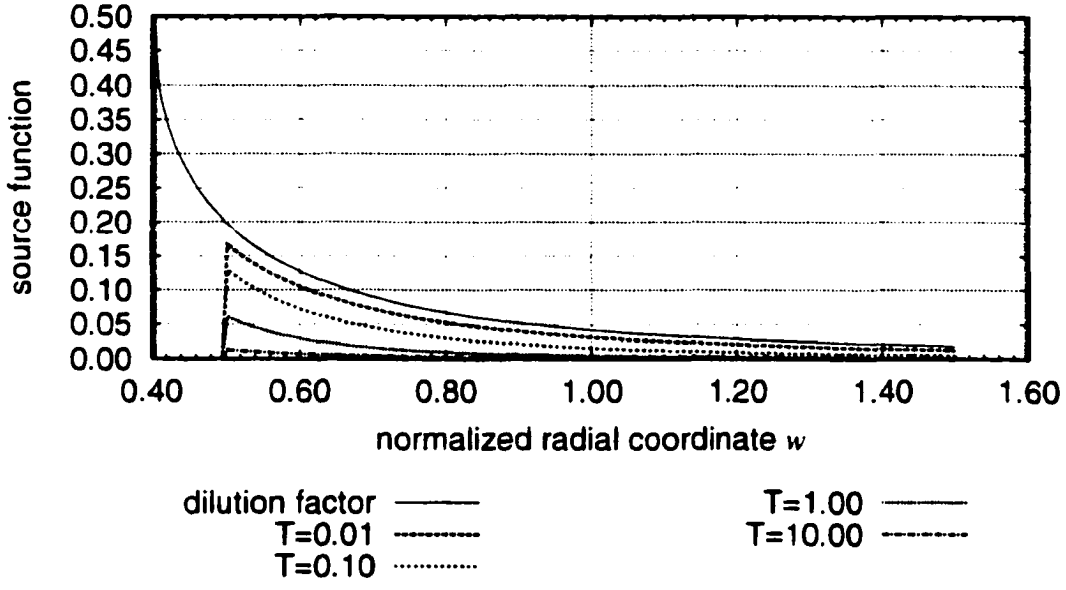


Figure 3.3: Like Figure 3.2 but for $w_C = 0.5$ and $w_O = 0.6$.



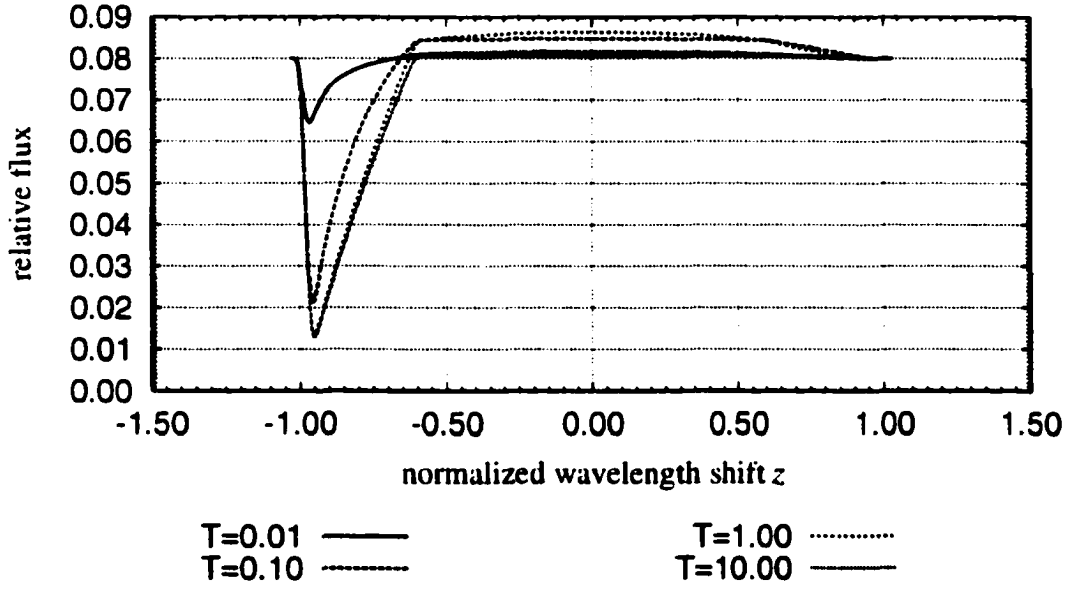
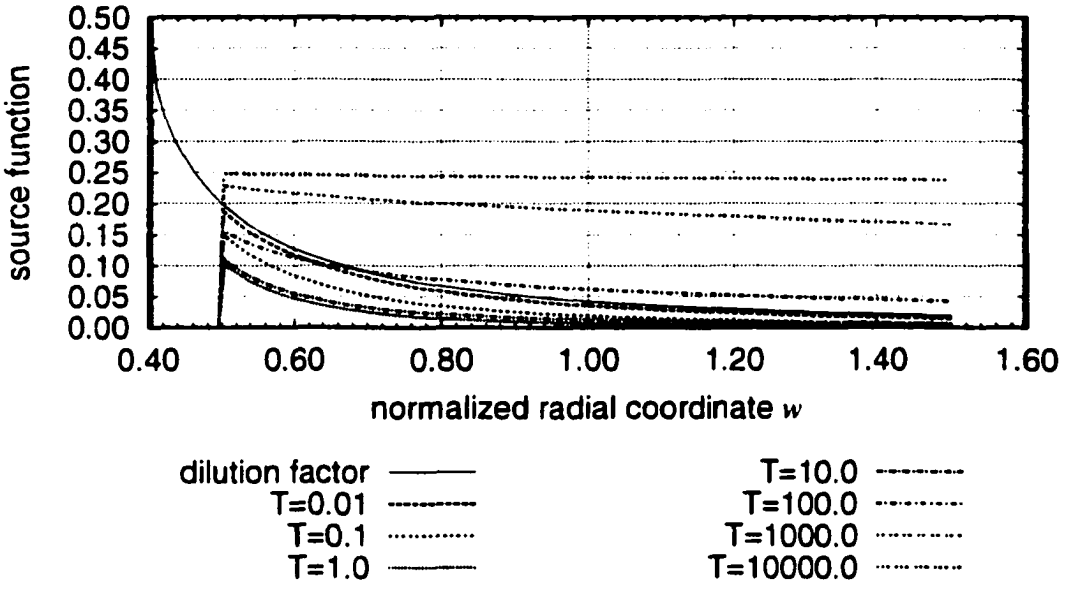


Figure 3.4: Like Figure 3.2 but for $x = 0.2$ and $B = 0$.



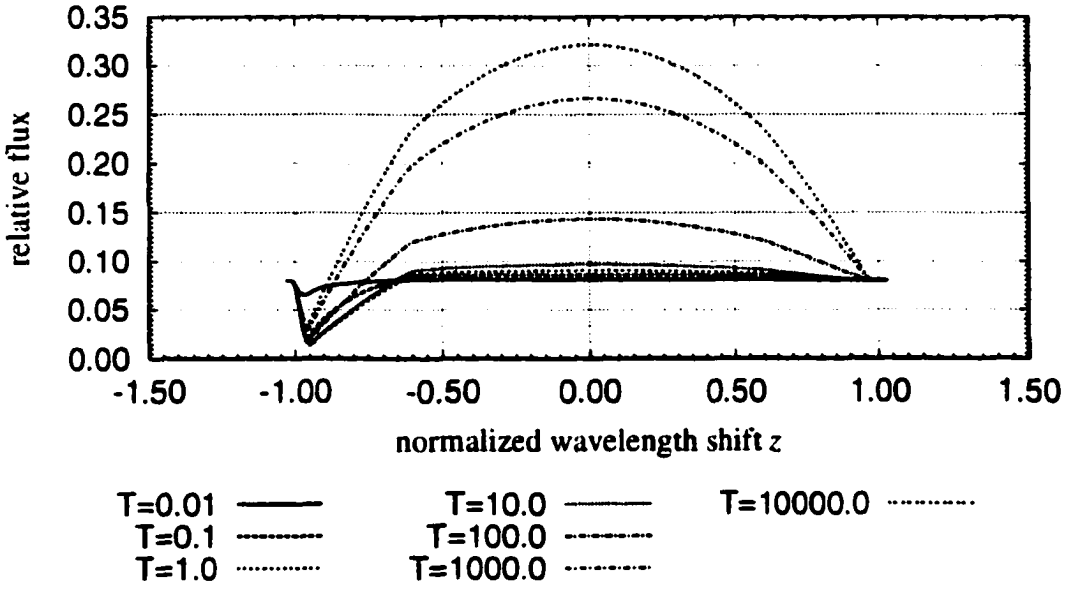


Figure 3.5: Like Figure 3.4 but for $x = 0.001$ and $B = 0.25$.

3.3.3 Isothermal Shell

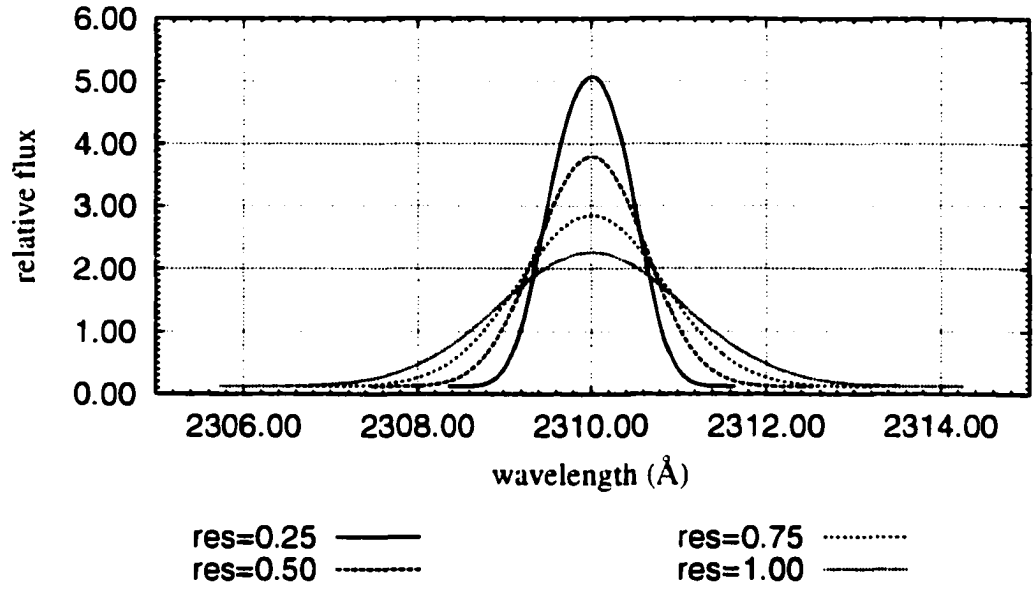
The admission of a non-zero probability ϵ of collisional de-excitation for the two-level atom requires the specification of a temperature. Specifying the ratio of the intensity B of blackbody radiation (at line center in the comoving frame) to the intensity I_0 emitted (at all wavelengths) by the opaque sphere is equivalent to specifying a temperature. The simplest way to do this is to set B/I_0 equal to some value that is fixed throughout the shell. Figure 3.4 illustrates the zero-temperature case for $\epsilon = 0.2$. One of the most noticeable differences between the case of pure scattering and the case in which the source function is coupled to a thermal pool has to do with the behavior for very strong lines. As T increases beyond unity, both $\beta_0(w)$ and $\beta(w)$ decrease at a given radial coordinate w. When both $\beta_0(w) \ll \epsilon B/[1-\epsilon]I_0$ and $\beta(w) \ll \epsilon/[1-\epsilon]$, the source function at w will have converged to B. For the case of pure scattering, every line with strength greater than about $T = 1$ has the same source function. However, when collisional de-excitation is turned on, the source function may continue to change even for very large values of T, especially if ϵ is small but positive. Because the pure-scattering source function is recovered as ϵ approaches zero, the source function will, for very small values of ϵ, vary only slowly with T as T becomes greater than unity until T becomes very large: then the source function will converge toward B as T increases still further. Figure 3.5 demonstrates this behavior for $\epsilon = 0.001$; note that the source function for $T = 1$ is almost identical to the source function for $T = 10$ but that the source function evolves toward $B = 0.25I_0$ for very large T.

Figure 3.4 shows that the zero-temperature shell, for a strong line, does little but absorb light from the emissive sphere. The line profile hardly ventures above the continuum for $T = 10$, even though it has a deep blueward absorption trough. Although light from this trough would be redistributed to the emission part of the profile in the case of

pure scattering, here the light is dissipated into the cold thermal pool from which it is not re-emitted. In comparison with the pure-scattering shell, the zero-temperature shell produces a very similar absorption trough because at the wavelengths in the trough, the observer cannot distinguish between the scattering of light out of the line of sight and the absorption of light by the material in the shell. Figure 3.5 shows that (for a sufficiently strong line) even a shell temperature for which B/I_0 is smaller than unity can produce a substantial emission component in the observed profile. At line center in the observer's frame, the flux contribution of $\pi w_C^2 I_0$ from the disk is augmented by a contribution of $\pi[w_O^2 - w_I^2]B$ from the shell. Nevertheless, so long as B/I_0 is smaller than unity, there will be a blueward absorption component because, for wavelengths corresponding to $\alpha < \arcsin(w_C/w_O)$, the only contribution from the shell obscures the brighter contribution that would otherwise come from the emissive sphere.

3.4 Instrumental Resolution

Because pure scattering merely redistributes light from one part of the observed profile to another, a pure-scattering observer-frame profile that is narrower than the instrumental resolution limit will be invisible. Figure 3.6 displays the change in the observer-frame line profile as the instrumental resolution varies. In the bottom graph, the pure-scattering profile virtually disappears as soon as the resolution of the detector is as large as the width of the profile. This example demonstrates that for a strong feature ($T = 10$) in a thick shell ($w_I = 0.5$ and $w_O = 5$) with an expansion speed $v = 100$ km s^{-1}, an instrumental resolution of 1Å is insufficient for the detection of the line in the case of pure scattering. The top graph, however, demonstrates that the same line coupled to a thermal pool in the shell can produce a detectable feature at a resolution of 1Å.



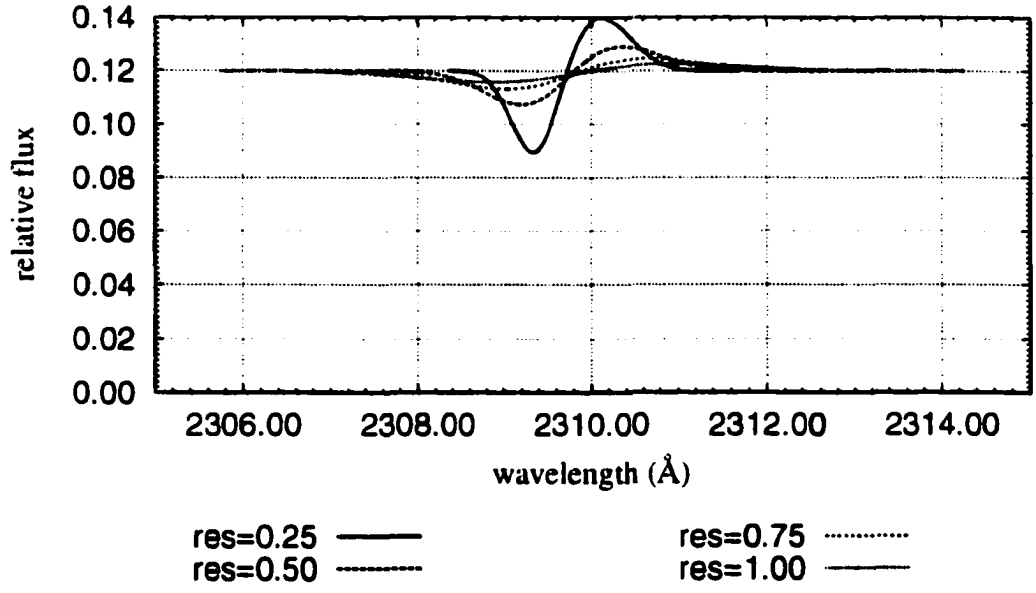
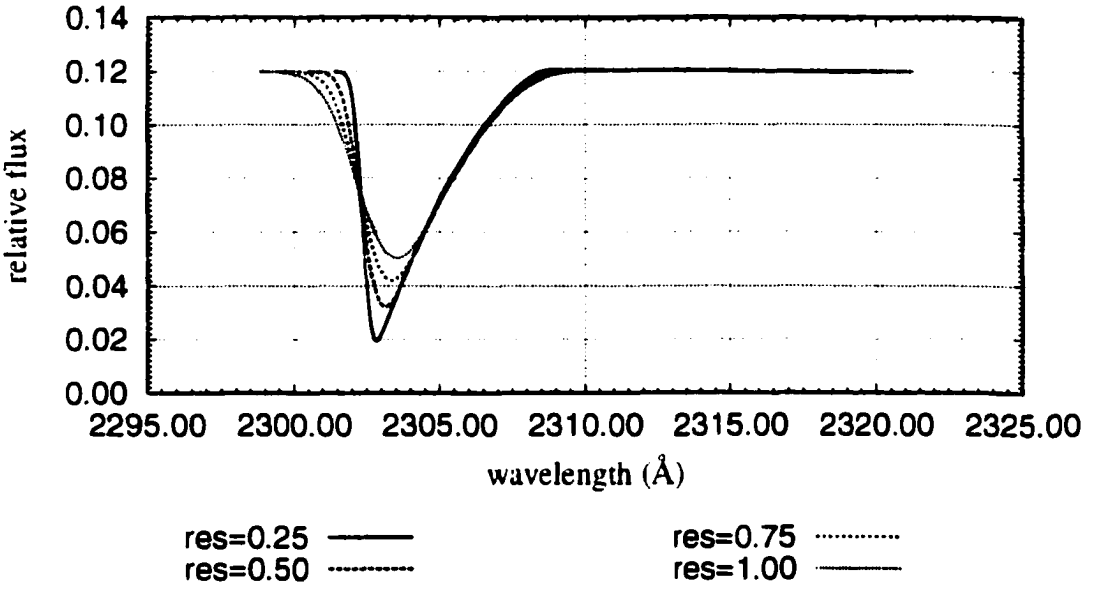


Figure 3.6: Pure-scattering profiles (bottom) correspond to $w_C = w_I = 0.5$ and $w_O = 5$. Line wavelength $\lambda' = 2310$Å; expansion speed $v = 100$ km s^{-1}; and line strength $T = 10$. Isothermal-shell profiles (top) add thermalization parameter $x = 0.5$ and blackbody intensity $B = 0.5I_0$. Each curve corresponds to a gaussian resolution with the indicated standard deviation (Å).



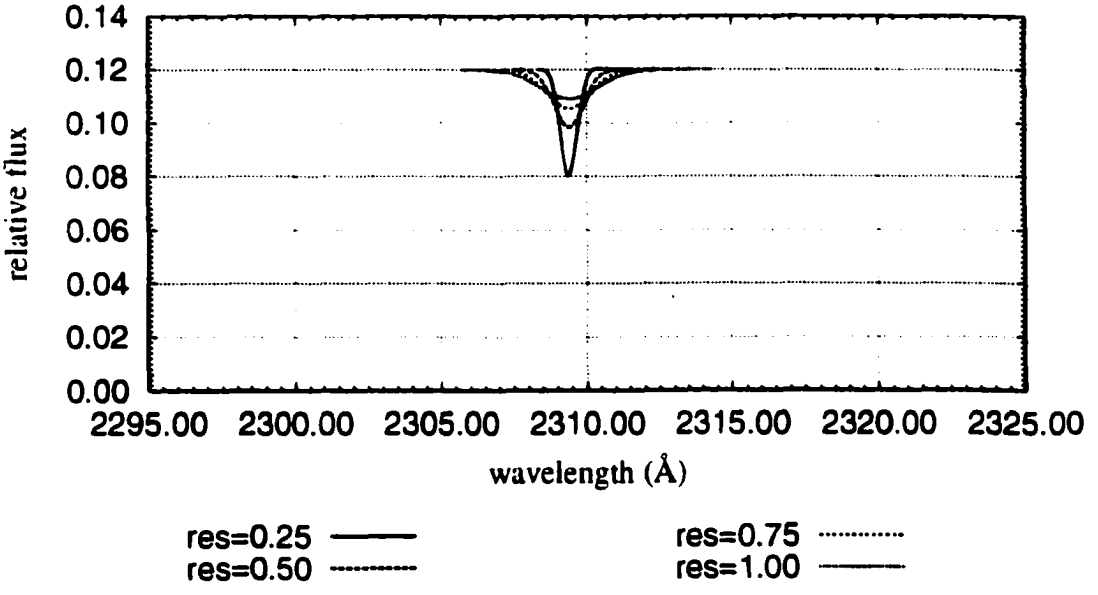


Figure 3.7: Like Figure 3.6 but for comparison of shell velocity at zero temperature. The bottom graph shows profiles for $v = 100$ km s^{-1}, and the top graph shows profiles for $v = 1000$ km s^{-1}.

Figure 3.7 is like Figure 3.6 but illustrates the effect of instrumental resolution on a strong line coupled to a zero-temperature thermal pool. The bottom graph, which shows profiles for a shell expansion speed $v = 100$ km s^{-1}, indicates that a weak feature could be detected with a resolution limit like that (about 0.75Å) of the detector that produced the data for Figure 1.3. The top graph, which corresponds to $v = 1000$ km s^{-1}, shows that the strength of the observed line profile really does increase with the velocity of the shell.

Chapter 4

Conclusion

The simple model presented here allows for the synthesis of line profiles. A wide variety of profile shapes may be produced in the exploration of the space formed by the model's input parameters. P-Cygni profiles in which the total flux is conserved but transfered from the absorption part to the emission part of the observer-frame profile correspond to the case of pure scattering ($x = 0$). A flattening and diminution of the emission part to a plateau can be accomplished by specifying a vacuum gap between the emissive sphere and the inner boundary of the shell. Alternatively, the emission part of the observer-frame profile can be reduced without sacrificing the depth of the absorption part by coupling the line to a low-temperature thermal pool. By coupling the line to higher-temperature pools, the emission part of the profile can be augmented, and, for $B \geq I_0$, a sufficiently strong line will not even show blueward absorption. Finally, the overall deviation of the profile from the continuum and the location of the absorption minimum (if it exists) can be controlled by adjusting the thickness of the shell. So, fitting a synthetic line profile produced by this model to an observed line profile would yield both a geometric interpretation of the feature's formation and an estimate of the strength of the transition.

For a spectral resolution of about an angstrom in the near-ultraviolet, a pure-scattering line profile described by this model will not be detected if the shell expansion speed v less than about 100 km s^{-1}. A strong line coupled to a zero-temperature shell would produce only a weak signal for $v = 100$ km s^{-1} and would likely be undetectable for $v = 10$ km s^{-1}. Even for the slowest expansion speeds, however, a strong line coupled with a thick shell whose blackbody emission intensity is at least a substantial fraction of the intensity emitted by the opaque sphere will produce a detectable unresolved emission feature. So this model would predict that for a very slowly expanding shell, the only detectable feature at 1-Å or broader resolution is an unresolved emission spike like those in Figure 1.2. The real test of the model will come in the analysis of line profiles measured with 0.1Å or better resolution.

There are, however, many potential problems. The simple shell wind model does not allow for variations in mass-loss rate and terminal velocity for the progenitor wind. Nor does it allow for the obvious types of asymmetry (prolate or oblate spheroidal) in the wind and in the SN ejecta; the model fails to distinguish between the identification of the emissive sphere with the SN photosphere and the identification of the emissive sphere with the interaction region between the ejecta and the wind. Nor does it take into account acceleration of the inner part of the wind by radiation pressure from the ejecta/wind interaction shock. Finally, the simple shell model assumes that the source of line opacity has no intrinsic width. Nevertheless, this model's simplicity allows for the complete comprehensibility of every aspect of every line profile that it produces. So, to the extent that it turns out to fit profiles consistently, the model provides direct physical insight into the nature of line formation in a circumstellar envelope around a SN.

The simplest extension of the model presented here is to include line blending. If two lines are separated in rest-frame wavelength by a wavelength shift smaller than the

largest shift obtainable by the doppler effect in the shell, then the bluer line may be treated exactly as above. In computing the source function for the redder line, however, intensity beams from the emissive sphere are augmented by intensity beams that come from a surface like those in Figure 2.8. Light in those extra beams is scattered from the bluer line into resonance with the redder line. The generalization to an arbitrary number of lines is equivalent to adding to the source function of each line a contribution from every bluer line.

Beyond line blending, an ambitious but interesting project would be to use internal data from a model that produces a synthetic fit to the broad features formed in the ejected material. Although the output of such a model is usually averaged over all of the (parallel) lines of sight, the full internal detail of such a model could provide a more realistic (than constant continuum intensity) lower boundary condition for the radiation field that illuminates the simple shell wind model.

Thank you for your attention.

Bibliography

M. S. Bessell. *UBVRI* passbands. *PASP*, 102:1181–1199, October 1990.

D. Branch, C. H. Lacy, M. L. McCall, P. G. Sutherland, A. Uomoto, J. C. Wheeler, and B. J. Wills. *ApJ*, 270:123, 1983.

A. Clocchiatti et al. A study of SN1992H in NGC 5377. *AJ*, 111:1286–1303, March 1996.

A. V. Filippenko. Optical spectra of supernovae. *Annu. Rev. Astron. Astrophys.*, 35, 1997.

A. Fisher, D. Branch, P. Nugent, and E. Baron. *MNRAS*, 481:L89, 1997.

K. Hatano, D. Branch, A. Fisher, J. Millard, and E. Baron. *ApJS*, 121:233, 1999.

P. Höflich and A. Khokhlov. *ApJ*, 457:500, 1996.

D. J. Jeffery, B. Leibundgut, R. P. Kirshner, S. Benetti, D. Branch, and G. Sonneborn. *ApJ*, 397:304, 1992.

E. J. Lentz, E. Baron, D. Branch, P. H. Hauschildt, and P. E. Nugent. *ApJ*, in press, 1999.

D. C. Leonard, A. V. Filippenko, A. J. Barth, and T. Matheson. *ApJ*, submitted, 1999.

D. Mihalas. *Stellar Atmospheres*. Freeman, 1978.

J. Millard et al. *ApJ*, in press, 1999.

R. Minkowski. *Publ. Astron. Soc. Pac.*, 53:224–225, 1941.

V. V. Sobolev. *Moving Envelopes of Stars*. Harvard University Press, 1960.

S. D. Van Dyk, R. A. Sramek, K. W. Weiler, and N. Panagia. *ApJ*, 409:162, 1993.

K. W. Weiler, S. D. Van Dyk, N. Panagia, and R. A. Sramek. *ApJ*, 398:248, 1992.

Appendix A

Source Code

A.1 Resonance Region Around Finite-Width Line

A.1.1 C++ Header (Interface-Description) Files

resonance-point.hh

```
#ifndef __RESONANCE_POINT_HH__
#define __RESONANCE_POINT_HH__

#define pi 3.14159265358

// For a monochromatic beam with direction angle, theta, and for a
// line with normalized width, epsilon, this class describes a
// resonance point in a bounded atmosphere.  The integrated line
// opacity is K_0 at r_0 from the center and is proportional to
// (r_0)^(-2).

class resonance_point
{
  double  w_I;      // inner boundary          in units of r_0
  double  w;        // resonance point location in units of r_0
  double  w_0;      // outer boundary          in units of r_0
  double  theta;    // direction angle of beam
  double  epsilon;  // normalized line width

  void    die( const char* fname, const char* msg ) const;

public :

  resonance_point();  // Read initialization from standard input.

  resonance_point( double i, double x, double o, double t, double e );
```

```
  double  get_w_I    () const { return w_I; }
  double  get_w      () const { return w; }
  double  get_w_0    () const { return w_0; }
  double  get_theta  () const { return theta; }
  double  get_epsilon() const { return epsilon; }

  void    set_w_I    ( double i );
  void    set_w      ( double x );
  void    set_w_0    ( double o );
  void    set_theta  ( double t );
  void    set_epsilon( double e );

  double  U( double z, int& flag ) const;
  double  tau()                    const;
  double  u( double t )            const;
};

#endif  // __RESONANCE_POINT_HH__
```

A.1.2 C++ Implementation Files for Library (Reusable) Components

resonance-point.cc

```
#include <math.h>
#include <string.h>
#include <iostream.h>

#include "resonance-point.hh"

void  resonance_point::die( const char* fname, const char* msg ) const
{
  cerr << "resonance_point::" << fname << ": " << msg << endl;
  exit( 1 );
}

resonance_point::resonance_point()
{
  for( int flag = 0; flag != 31; )
    {
      char    param_name[16];
      double  val;

      cin >> param_name;

      if( ! strcmp( param_name, "epsilon" ) )
{
```

```
  if( ! (cin >> val) )
    die( "resonance_point", "unable to read epsilon" );
  else
    set_epsilon( val );

  flag |= 1;
}
      else if( ! strcmp( param_name, "w_I" ) )
{
  if( ! (cin >> val) )
    die( "resonance_point", "unable to read w_I" );
  else
    set_w_I( val );

  flag |= 2;
}
      else if( ! strcmp( param_name, "w" ) )
{
  if( ! (cin >> val) )
    die( "resonance_point", "unable to read w" );
  else
    set_w( val );

  flag |= 4;
}
      else if( ! strcmp( param_name, "w_0" ) )
{
  if( ! (cin >> val) )
    die( "resonance_point", "unable to read w_0" );
  else
    set_w_0( val );

  flag |= 8;
}
      else if( ! strcmp( param_name, "theta" ) )
{
  if( ! (cin >> val) )
    die( "resonance_point", "unable to read theta" );
  else
    set_theta( val );

  flag |= 16;
}
      else
{
  cerr << "resonance_point::resonance_point: param_name='"
       << param_name << "'" << endl;
```

```
  die( "resonance_point", "unknown parameter" );
}
    }
}

resonance_point::resonance_point( double i,
  double x,
  double o,
  double t,
  double e  )
{
  set_w_I    ( i );
  set_w      ( x );
  set_w_0    ( o );
  set_theta  ( t );
  set_epsilon( e );
}

void  resonance_point::set_w_I( double i )
{
  if( i <= 0 )  die( "set_w_I", "w_I not positive" );
  else          w_I = i;
}

void  resonance_point::set_w( double x )
{
  if( x <= w_I )  die( "set_w", "w not larger than w_I" );
  else            w = x;
}

void  resonance_point::set_w_0( double o )
{
  if( o <= w )  die( "set_w_0", "w_0 not larger than w" );
  else          w_0 = o;
}

void  resonance_point::set_theta( double t )
{
  if( t <= 0 || t >= pi )
    die( "set_theta", "theta not between 0 and pi" );
  else
    theta = t;
}

void  resonance_point::set_epsilon( double e )
{
  if( e <=0 )  die( "set_epsilon", "epsilon not positive" );
  else         epsilon = e;
```

```
}

double resonance_point::U( double z, int& flag ) const
{
  double  ct  = cos( theta );
  double  zpc = z + ct;

  if( zpc <= -1 )
    {
      flag = -1;  // negative infinity
      return  0;  // return value irrelevant
    }
  else if( zpc >= 1 )
    {
      flag =  1;  // positive infinity
      return  0;  // return value irrelevant
    }
  else
    {
      flag =  0;  // finite value
      return  ( zpc*sin( theta )/sqrt( 1 - zpc*zpc ) - ct )*w;
    }
}

double  resonance_point::tau() const
{
  double  wc = w*cos( theta );
  double  ws = w*sin( theta );

  double  ti = asin( w_I/w );

  double  swow = sqrt( w_O*w_O - ws*ws );
  double  swiw = sqrt( w_I*w_I - ws*ws );

  double  u_n;
  double  u_p;

  if( theta <      ti )  u_n =  swiw - wc;
  else                   u_n = -swow - wc;

  if( theta < pi - ti )  u_p =  swow - wc;
  else                   u_p = -swiw - wc;

  int     flag;
  double  u;

  double  upsilon_p;
  double  upsilon_n;
```

```
  u = U( +epsilon, flag );

  if     ( flag == -1 )   upsilon_p = u_n;
  else if( flag ==  1 )   upsilon_p = u_p;
  else
    {
      if     ( u < u_n )  upsilon_p = u_n;
      else if( u > u_p )  upsilon_p = u_p;
      else                upsilon_p = u;
    }

  u = U( -epsilon, flag );

  if     ( flag == -1 )   upsilon_n = u_n;
  else if( flag ==  1 )   upsilon_n = u_p;
  else
    {
      if     ( u < u_n )  upsilon_n = u_n;
      else if( u > u_p )  upsilon_n = u_p;
      else                upsilon_n = u;
    }

  double  arg_p = ( upsilon_p + wc )/ws;
  double  arg_n = ( upsilon_n + wc )/ws;

  double  r = ( atan( arg_p ) - atan( arg_n ) )/epsilon/ws;

  if( r < 0.0 )  die( "tau", "tau less than zero" );

  return  r;
}

double  resonance_point::u( double t ) const
{
  double  wc = w*cos( theta );
  double  ws = w*sin( theta );

  double  ti = asin( w_I/w );

  double  swow = sqrt( w_0*w_0 - ws*ws );
  double  swiw = sqrt( w_I*w_I - ws*ws );

  double  u_n;
  double  u_p;

  if( theta <      ti )  u_n =  swiw - wc;
  else                   u_n = -swow - wc;
```

```
  if( theta < pi - ti )  u_p =  swow - wc;
  else                   u_p = -swiw - wc;

  int     flag;
  double  u;

  double  upsilon_n;

  u = U( -epsilon, flag );

  if     ( flag == -1 )   upsilon_n = u_n;
  else if( flag ==  1 )   upsilon_n = u_p;
  else
    {
      if     ( u < u_n )  upsilon_n = u_n;
      else if( u > u_p )  upsilon_n = u_p;
      else                upsilon_n = u;
    }

  double  arg = atan( (upsilon_n + wc)/ws + epsilon*ws*t );

  if     ( arg <= -pi/2 )  return  -100;
  else if( arg >=  pi/2 )  return   100;
  else                     return  ws*tan( arg ) - wc;
}
```

A.1.3 C++ Main-Program Files

t-contours-lim.0.cc

```
#include <stdlib.h>
#include <iostream.h>

#include "resonance-point.hh"

main( int argc, char** argv )
{
  if( argc != 2 )
    {
      cerr << "usage: " << argv[ 0 ] << " <optical_depth>" << endl;
      exit( 1 );
    }

  double           t = atof( argv[ 1 ] );
  resonance_point  rp;
```

```
  double  th_beg =   0.1;
  double  th_end = 179.9;
  int     th_num = 500;
  double  th_del = ( th_end - th_beg )/( th_num - 1 );

  for( int i=0; i<th_num; i++ )
    {
      double  th = (th_beg + th_del*i)*pi/180;
      rp.set_theta( th );
      double  u  = rp.u( t );
      cout << u*sin( th ) << " " << u*cos( th ) << endl;
    }
}
```

tau-angle-lim.0.cc

```
#include <stdlib.h>
#include <iostream.h>

#include "resonance-point.hh"

main( int argc, char** argv )
{
  if( argc != 2 )
    {
      cerr << "usage: " << argv[ 0 ] << " <direction_angle_deg>" << endl;
      exit( 1 );
    }

  double           epsilon = atof( argv[ 1 ] );
  resonance_point  rp;

  rp.set_epsilon( epsilon );

  double  theta_beg =    0.0001;
  double  theta_end =  179.9999;
  int     theta_num = 2000;
  double  theta_del = ( theta_end - theta_beg )/( theta_num - 1 );

  for( int j=0; j<theta_num; j++ )
    {
      rp.set_theta( (theta_beg + j*theta_del)*pi/180.0 );
      cout << rp.get_theta()*180/pi << " " << rp.tau() << endl;
    }
}
```

tau-angle2-lim.0.cc

```
#include <stdlib.h>
#include <iostream.h>

#include "resonance-point.hh"

main( int argc, char** argv )
{
  if( argc != 2 )
    {
      cerr << "usage: " << argv[ 0 ] << " <rad_coord_w>" << endl;
      exit( 1 );
    }

  double           w = atof( argv[ 1 ] );
  resonance_point  rp;

  rp.set_w( w );

  double  theta_beg =    0.0001;
  double  theta_end =  179.9999;
  int     theta_num = 2000;
  double  theta_del = ( theta_end - theta_beg )/( theta_num - 1 );

  for( int j=0; j<theta_num; j++ )
    {
      rp.set_theta( (theta_beg + j*theta_del)*pi/180.0 );
      cout << rp.get_theta()*180/pi << " " << rp.tau() << endl;
    }
}
```

tau-linewidth-lim.0.cc

```
#include <iostream.h>
#include <math.h>
#include <stdlib.h>

#include "resonance-point.hh"

main( int argc, char** argv )
{
  if( argc != 2 )
    {
      cerr << "usage: " << argv[ 0 ] << " <direction_angle_deg>" << endl;
```

```
      exit( 1 );
    }

  double           theta = atof( argv[ 1 ] );
  resonance_point  rp;

  rp.set_theta( theta*pi/180.0 );

  double  e_beg =   0.001;
  double  e_end =  10.0000;
  int     e_num = 999;

  double  e_rat = pow( e_end/e_beg, 1.0/(e_num - 1.0) );
  //double  e_del = ( e_end - e_beg )/( e_num - 1.0 );

  for( int j=0; j<e_num; j++ )
    {
      rp.set_epsilon( e_beg*pow( e_rat, j ) );
      //rp.set_epsilon( e_beg + j*e_del );

      cout << rp.get_epsilon() << " " << rp.tau() << endl;
    }
}
```

A.2 Observed Profile of Delta-Function Line

A.2.1 C++ Header (Interface-Description) Files

linterp.hh

```
// diss/sobolev/linterp.hh
//
// copyright 1999  Thomas E. Vaughan
//
// This is free software, redistributable only under the terms of the GNU
// General Public License (GPL).  See <http://www.gnu.org>.

// Provide a general linear interpolation function.

double linterp
(
     double xmin, double xmax, double dx, double x, double y[]
);
```

parameters.hh

```
// diss/sobolev/parameters.hh
//
// copyright 1999  Thomas E. Vaughan
//
// This is free software, redistributable only under the terms of the GNU
// General Public License (GPL).  See <http://www.gnu.org>.

#ifndef __PARAMETERS_HH__
#define __PARAMETERS_HH__

#include <map>
#include <string>

class parameters
{
     map<string,double> vals;

     public :

     parameters( const char* filename );
     double get( const string& key ) const;
};

extern parameters*  params;

#endif // __PARAMETERS_HH__
```

source_function.hh

```
// diss/sobolev/source_function.hh
//
// copyright 1999 Thomas E. Vaughan
//
// This is free software, redistributable only under the terms of the GNU
// General Public License (GPL).  See <http://www.gnu.org>.

// Calculate the Sobolev source function for a spherically-symmetric
// constant-speed radial wind whose only illumination comes from a central
// sphere.  The illumination is angle-independent (over the solid angle
// subtended by the central sphere) and wavelength-independent.  The line
// profile is a delta-function.

double source_function
```

```
(
double w_em,  // size of light-emitting sphere
double w,     // radial parameter of interest
double T,     // line-strength parameter
double x,     // collisional fraction of de-excitations
double B      // Blackbody at w in terms of intensity at w_em
);
```

trapezoid.hh

```
#ifndef __TRAPEZOID_HH__
#define __TRAPEZOID_HH__

// Provide an interface to a function that calculates an integral by
// incrementally doubling the number of partitions in the domain of the
// integrand and applying the trapezoid rule.

double trapezoid(
double (*f)( double x ),  // pointer to integrand
double a,                 // lower limit of integration
double b,                 // upper limit of integration
double e = 1.0E-05,       // desired accuracy
int    m = 20 );          // max number of bifurcations

#endif // __TRAPEZOID_HH__
```

A.2.2 C++ Implementation Files for Library (Reusable) Components

linterp.cc

```
// diss/sobolev/linterp.cc
//
// copyright 1999  Thomas E. Vaughan
//
// This is free software, redistributable only under the terms of the GNU
// General Public License (GPL).  See <http://www.gnu.org>.

// Provide a general linear interpolation function.

#include <iostream>

double linterp
(
     double xmin, double xmax, double dx, double x, double y[]
```

```
)
{
     if( x < xmin || x > xmax )
     {
          // Disallow extrapolation.

          cerr << "linterp: ERROR: x=" << x << " not in ["
               << xmin << "," << xmax << "]" << endl;

          exit( 1 );
     }

     if     ( x == xmax )  return y[ int( (x-xmin)/dx ) ];
     else if( x == xmin )  return y[ 0 ];

     // index of left edge of interpolation zone
     int i = int( (x - xmin)/dx );

     // coordinate of left edge of zone
     double xleft = xmin + i*dx;

     // slope of linear fit across zone
     double m = ( y[i+1] - y[i] )/dx;

     // linearly interpolated value
     double v = ( y[i] + m*(x - xleft) );

     return v;
}
```

parameters.cc

```
// diss/sobolev/parameters.cc
//
// copyright 1999 Thomas E. Vaughan
//
// This is free software, redistributable only under the terms of the GNU
// General Public License (GPL).  See <http://www.gnu.org>.

#include <fstream>
#include <string>

#include "parameters.hh"

parameters*  params;  // global pointer to parameters instance params
```

```
parameters::parameters( const char* filename )
{
     istream*  isp;  // base input stream pointer

     if( filename )
     {
          isp = new ifstream( filename );

          cerr << "parameters::parameters: reading from "
               << filename << endl;
     }
     else
     {
          isp = &cin;

          cerr << "parameters::parameters: reading from "
               << "standard input" << endl;
     }

     string k;  // current key

     // Read keys until the end of the input file is reached.  Assume
     // that the next word is a key.

     while( (*isp) >> k )
     {
          // k now holds candidate key.

          // Assume that the next word is the value corresponding to
          // the current key.

          (*isp) >> vals[ k ];
     }
}

// Allow for a public method to extract a parameter value by its key.

double parameters::get( const string& key ) const
{
     map<string,double>::const_iterator i = vals.find( key );

     if( i == vals.end() )
     {
          // key was not initialized in the constructor.

          cerr << "parameters::get: ERROR: no entry for " << key
               << endl;
```

```
        exit( 1 );
    }

    return i->second;
}
```

source_function.cc

```
// diss/sobolev/source_function.cc
//
// copyright 1999 Thomas E. Vaughan
//
// This is free software, redistributable only under the terms of the GNU
// General Public License (GPL).  See <http://www.gnu.org>.

// Calculate the Sobolev source function for a spherically-symmetric
// constant-speed radial wind whose only illumination comes from a central
// sphere.  The illumination is angle-independent (over the solid angle
// subtended by the central sphere) and wavelength-independent.  The line
// profile is a delta-function.

#include <iostream>
#include <math.h>
#include "trapezoid.hh"

// The following two variables are file-scope global because the integrand
// function below requires access to them but must have a signature with
// only one argument.

static double  T;  // half of minimum optical depth at w = 1
static double  w;  // radial coordinate in units of r_0

// The source function contains two integrals.  Each integral has the same
// integrand.  Defined as follows, f() is that integrand.  The limits of
// integration for one of the integrals are 0 and theta_I; for the other, 0
// and pi.

double f( double theta )
{
double st = sin( theta );
return 0.25*w*( 1 - exp( -2*T/w/st/st ) )*st*st*st/T;
}

double source_function
(
double w_em,  // size of light-emitting sphere
```

```
double ww,    // radial parameter of interest
double TT,    // half of minimum optical depth at w = 1

double x,     // collisional fraction of de-excitations
              // x=0 corresponds to pure resonance scattering.
      // x=1 corresponds to pure thermal emission

double B      // thermal blackbody intensity at ww in units of
              // intensity of light-emitting sphere
)
{
if( w_em > ww )
{
// Something weird is going on.

cerr << "source_function: ERROR: w_em=" << w_em
     << " > w=" << ww << endl;

exit( 1 );
}

T = TT;  // assignment to global variable for f()
w = ww;  // assignment to global variable for f()

double theta_I = asin( w_em/w );

double integ_1 = trapezoid( f, 1.0E-06, theta_I );
double integ_2 = trapezoid( f, theta_I, M_PI    );

return ( (1-x)*integ_1 + x*B )/( (1-x)*(integ_1+integ_2) + x );
}
```

trapezoid.cc

```
// diss/sobolev/trapezoid.cc
//
// copyright 1999  Thomas E. Vaughan
//
// This is free software, redistributable only under the terms of the GNU
// General Public License (GPL).  See <http://www.gnu.org>.

#include <iostream.h>
#include <math.h>

#include "trapezoid.hh"
```

```
// Provide a convenient way to dereference a function pointer.
#define F(x) ((*f)(x))

// Calculate the nth-order trapezoid rule sum.  This function assumes that
// it has already been called once, in turn, for each of the previous n-1
// sums.

double trap(
double (*f)(double),  // pointer to integrand function
double a,             // lower integration limit
double b,             // upper integration limit
int    n )            // number of bifurcations
{
static double s;  // sum to be returned

if( n == 1 )
{
return ( s = 0.5*(b - a)*( F(a) + F(b) ) );
}
else
{
int it;
int j;

for( it = 1, j = 1; j < n - 1; j++ ) it <<= 1;

double tnm = it;
double del = (b - a)/tnm;  // spacing of new points
double x   = a + 0.5*del;

double sum;
for( sum=0, j=1; j <= it; j++, x += del ) sum += F(x);

s = 0.5*( s + (b - a)*sum/tnm );  // refinement of s
return s;
}
}

// Provide the public interface, and handle the proper calling of trap()
// above.

double trapezoid(
double (*f)(double),  // pointer to integrand function
double a,             // lower integration limit
double b,             // upper integration limit
double e,             // error tolerance
int    m )            // maximum number of bifurcations
{
```

```
if( a == b ) return 0.0;  // quick escape for trivial case
double olds = -1.0E+30;   // unlikely to be function average

for( int j = 1; j <= m; j++ )
{
double s = trap( f, a, b, j );

// cerr << "trapezoid: j=" << j << " s=" << s << endl;

if( fabs(s - olds) < e*fabs(olds) ||
    s == 0.0 && olds == 0.0 && j > 6 )  return s;

olds = s;
}

cerr << "trapezoid: ERROR: too many steps" << endl;
exit( 1 );
}
```

A.2.3 C++ Main-Program Files

sf.cc

```
// dissertation/sobolev/sf.cc
//
// copyright 1999 Thomas E. Vaughan
//
// This is free software, redistributable only under the terms of the GNU
// General Public License (GPL).  See <http://www.gnu.org>.

#include <iostream>

#include "parameters.hh"
#include "source_function.hh"

// Provide a stand-alone program for use in producing a plot of the source
// function for various values of T.

main( int argc, char** argv )
{
if( argc == 1 )
{
// There are no command-line arguments.  Pass a null string
// pointer to the parameters constructor so that it reads
// from the standard input.
```

```
params = new parameters( 0 );
}
else if( argc == 2 )
{
// There is a command-line argument.  Assume that the
// argument is the name of a parameter file.

params = new parameters( argv[1] );
}
else
{
// There are two or more command-line arguments.  This is
// an error condition.

for( int i = 0; i < argc; i++ )
cerr << "argument " << i << " " << argv[i]
     << endl;

cerr << "usage: " << argv[0] << " <parameter file>"
     << endl;

exit( 1 );
}

// Get parameters.

double  T        = params->get( "T"        );
double  w_opaque = params->get( "w_opaque" );
double  w_min    = params->get( "w_min"    );
double  w_max    = params->get( "w_max"    );
double  x        = params->get( "x"        );
double  B_inner  = params->get( "B_inner"  );
double  B_outer  = params->get( "B_outer"  );

int     sgrid = int( params->get("sgrid") + 0.5 );
double  w_del = (w_max   - w_opaque)/(sgrid - 1.0);

for( int j = 0; j < sgrid; j++ )
{
double w = w_opaque + j*w_del;
double s;

if( w < w_min )
{
s = 0;
}
else
{
```

```
double r = (w - w_min)/(w_max - w_min);
double B = B_inner + r*(B_outer - B_inner);

s = source_function( w_opaque, w, T, x, B );
}

cout << w << " " << s << endl;
}
}
```

smooth.cc

```
// dissertation/sobolev/smooth.cc
//
// copyright 1999  Thomas E. Vaughan
//
// This is free software, redistributable only under the terms of the GNU
// General Public License (GPL).  See <http://www.gnu.org>.

// Transform two-column x-y data from standard input to standard output.
// Transformed data has been smoothed by a gaussian whose standard
// deviation is (optionally) specified on the command line.

#include <stdlib.h>      // exit() and strtod()
#include <iostream.h>    // usual C++ I/O
#include <map.h>         // STL map
#include <math.h>        // M_PI and sqrt()

#include "trapezoid.hh"  // trapezoid() trapezoid-rule integrator

// The following variables are file-scope global because they are needed by
// f1(), which must have only one parameter.  A pointer to f1() is passed
// to trapezoid() for integration.

// s2pi is a normalization factor for the gaussian.
static const double s2pi  = sqrt( 2.0*M_PI );

static map<double,double> f;  // map for standard input

static double sd;             // standard deviation of gaussian
static double x_center;       // central x value for the current gaussian

// f1() is an integrand, the product of (1) a gaussian and (2) an
// interpolant and extrapolant of the input data.  Interpolation is
// performed linearly, and the extrapolant is a constant equal to the edge
// value of the input data.
```

```
double f1( double x )
{
static double id;  // value of interpolant at x

if( x < ( f.begin() )->first )
{
// x is smaller than the smallest key, and so extrapolation
// is necessary.

id = ( f.begin() )->second;
}
else if( x > ( f.rbegin() )->first )
{
// x is larger than the largest key, and so extrapolation
// is necessary.

id = ( f.rbegin() )->second;
}
else
{
// Interpolate.

static map<double, double>::iterator ub;
static map<double, double>::iterator lb;

ub = f.upper_bound( x );  // upper bound
lb = ub;

--lb;                     // lower bound

static double x1;
static double x2;
static double y1;
static double y2;

// A map is a set of pairs, each of which has a 'first' and
// a 'second' member.

x1 = lb->first;
x2 = ub->first;
y1 = lb->second;
y2 = ub->second;

id = y1 + (x - x1)*(y2 - y1)/(x2 - x1);
}

static double z;
z = (x - x_center)/sd;
```

```
return id*exp( -0.5*z*z )/sd/s2pi;
}

// main program

int main( int argc, char* argv[] )
{
// Process command-line arguments, if any.

if( argc == 1 )
{
// There are no command-line arguments.  Indicate that sd
// should be chosen automatically.

sd = 0;
}
else if( argc == 2 )
{
// There is a command-line argument.  Assume that it
// represents the standard deviation.

char** endptr = NULL;
sd = strtod( argv[1], endptr );

if( sd < 0 )
{
cerr << argv[0] << ": ERROR: negative standard "
     << "deviation" << endl;

exit( 1 );
}
}
else
{
// There are too many command-line arguments.

cerr << "usage: " << endl;
cerr << argv[0] << " [<standard deviation>]" << endl;

exit( 1 );
}

// Read standard input.
{
double x;
double y;

while( cin >> x >> y )  f[ x ] = y;
```

```
}

     // total x size of input data
     double xsize = ( f.rbegin() )->first - ( f.begin() )->first;

// mean x step size in input data
double dx = xsize/( f.size() - 1 );

// If sd is unspecified or specified on command line as 0, then set
// sd equal to mean x step size in input data.

if( sd == 0 )  sd = dx;

// sdlim  is the number of standard deviations to consider for the
//        gaussian.
//
// i      is an iterator for (like a pointer to) an element of a
//        map<double,double>.  Each such element is a pair.

const double                  sdlim = 3.5;
map<double,double>::iterator  i;

// Extend the input map by at least sdlim standard deviations on
// each side.
{
i = f.begin();

double x1 = i->first;                // smallest      x
double y1 = i->second;               // corresponding y

double x2 = ( f.rbegin() )->first;   // largest       x
double y2 = ( f.rbegin() )->second;  // corresponding y

for( int j = 0; j < sdlim*sd/dx + 1; j++ )
{
f[ x1 - j*dx ] = y1;
f[ x2 + j*dx ] = y2;
}
}

// For each x in the input data, integrate the product of (1) a
// gaussian centered at that x and (2) the interpolated or
// extrapolated value from the input data.  The integral becomes
// the output value for that x.

for( i = f.begin(); i != f.end(); i++ )
{
x_center = i->first;  // global parameter for f1()
```

```
double x_lo = i->first - sdlim*sd;
double x_hi = i->first + sdlim*sd;

// Calculate and send results to standard output.

cout.precision( 11 );

cout << i->first << " " << trapezoid( f1, x_lo, x_hi )
     << endl;
}

// Indicate normal exit to operating system.
return( 0 );
}
```

test_syn.cc

```
// dissertation/sobolev/test_syn.cc
//
// copyright 1999  Thomas E. Vaughan
//
// This is free software, redistributable only under the terms of the GNU
// General Public License (GPL).  See <http://www.gnu.org>.

// Synthesize a line profile as measured by a distant observer of a
// radially expanding spherical shell.  The expansion speed is independent
// of radial coordinate.  In the comoving frame, the shell is transparent
// at every wavelength except for a single delta-function line profile.  An
// opaque sphere whose radius is smaller than that of the shell's inner
// boundary emits light with an intensity that is independent of angle and,
// for wavelengths in the vicinity of the line's rest-frame wavelength, is
// also independent of wavelength.  There is no source of opacity outside
// of the outer boundary; also, there is no source of opacity between the
// opaque sphere and the inner boundary.  The integrated opacity of the
// line varies as the inverse square of the radial coordinate.

// Allow access to standard input-output and math functions.

#include <iostream>
#include <math.h>

// Allow access to various functions that I have written.  These functions
// allow linear interpolation, reading of a parameter file, calculation of
// the source function, and calculation of an integral via an iterative
// trapezoid-rule scheme.
```

```
#include "linterp.hh"
#include "parameters.hh"
#include "source_function.hh"
#include "trapezoid.hh"

// The following variables are file-scope global because they are needed by
// f1() and f2(), each of which must have only one argument in its
// signature.  In this context, "static" means that the symbol cannot be
// accessed by a function in a different file scope.

static double* sf;       // pointer to start of source-function array
static double  s_del;    // step size for source-function grid

static double  z;        // wavelength shift (Think of redshift.)
static double  T;        // line strength (half of min opt depth at w=1)

static double  w_opaque; // opaque sphere radius in units of r_0
static double  w_min;    // shell inner   radius in units of r_0
static double  w_max;    // shell outer   radius in units of r_0

// Within the scope of each of the following functions, "static" means that
// storage for the symbol is allocated at compile time, and so no time will
// be required for automatic stack allocation at each function call.

// f1() maps impact parameter p to the product of p and the intensity at p
// for wavelength shift z.  f1() corresponds to the intensity emergent from
// a resonance point between the opaque sphere and the observer.

double f1( double p )
{
static double w;  // radial coordinate
static double t;  // optical depth

w = p/sqrt( 1.0 - z*z );
t = 2.0*T*w/p/p;

// w should lie within [w_min,w_max].  Correct for the inevitable
// slight numerical error.

if     ( w < w_min ) w = w_min;
else if( w > w_max ) w = w_max;

static double s;   // source function
static double wi;  // intensity weighted by impact parameter

s  = linterp( w_opaque, w_max, s_del, w, sf );
wi = p*( s*( 1.0 - exp(-t) ) + exp(-t) );
```

```
return wi;
}

// f2() maps impact parameter p to the product of p and the intensity at p
// for wavelength shift z.  f2() corresponds to the intensity emergent from
// a resonance point NOT between the opaque sphere and the observer.

double f2( double p )
{
static double w;  // radial coordinate
static double t;  // optical depth

w = p/sqrt( 1.0 - z*z );
t = 2.0*T*w/p/p;

// w should lie within [w_min,w_max].  Correct for the inevitable
// slight numerical error.

if     ( w < w_min ) w = w_min;
else if( w > w_max ) w = w_max;

static double s;   // source function
static double wi;  // intensity weighted by impact parameter

s  = linterp( w_opaque, w_max, s_del, w, sf );
wi = p*s*( 1.0 - exp(-t) );

return wi;
}

// Execution of the program begins with this function, main().

main( int argc, char** argv )
{
// First, process command-line arguments.

if( argc == 1 )
{
// There are no command-line arguments.  Pass a null string
// pointer to the parameters constructor so that it reads
// from the standard input.

params = new parameters( 0 );
}
else if( argc == 2 )
{
// There is a command-line argument.  Assume that the
```

```
// argument is the name of a parameter file.

params = new parameters( argv[1] );
}
else
{
// There are two or more command-line arguments.  This is
// an error condition.

cerr << "usage: " << argv[0] << " <parameter file>"
     << endl;

exit( 1 );
}

// Use parameters to initialize some variables.

T        = params->get( "T"        );  // line-strength parameter
w_opaque = params->get( "w_opaque" );  // opaque sphere radius
w_min    = params->get( "w_min"    );  // shell inner boundary
w_max    = params->get( "w_max"    );  // shell outer boundary

if( w_opaque > w_min )
{
cerr << argv[0] << ": ERROR: opaque sphere bigger than "
     << "inner boundary of shell: " << w_opaque << " > "
     << w_min << endl;

exit( 1 );
}

// sgrid is number of source-function  points
// zgrid is number of wavelength-shift points

int sgrid = int( params->get("sgrid") + 0.5 );
int zgrid = int( params->get("zgrid") + 0.5 );

sf = new double[ sgrid ]; // memory allocation for source function

double z_max =  1.0; // max wavelength shift in units of v_0/c
double z_min = -1.0; // min wavelength shift in units of v_0/c

// various step sizes

s_del        = (w_max - w_opaque)/(sgrid - 1);  // global variable
double z_del = (z_max - z_min   )/(zgrid - 1);  // local  variable

// Calculate source function for each point on a radial grid.
```

```
double x = params->get( "x" );  // coll. frac. of de-excitations

double B_inner = params->get( "B_inner" );  // at opaque sphere
double B_outer = params->get( "B_outer" );  // at outer boundary

for( int i = 0; i < sgrid; i++ )
{
double w = w_opaque + i*s_del;
double s;

if( w < w_min )
{
s = 0;
}
else
{
double r = (w - w_min)/(w_max - w_min);
double B = B_inner + r*(B_outer - B_inner);

s = source_function( w_opaque, w, T, x, B );
}

sf[i] = s;
}

// Consider each wavelength shift in turn.

cerr << argv[0] << ": calculating intensity for each wavelength"
     << flush;

int ip = -1;  // counter for progress graph

for( int i = 0; i < zgrid; i++ )
{
z = z_min + i*z_del;  // current wavelength shift

if( i*10/zgrid > ip )
{
// Add a tic to the progress graph.
cerr << "." << flush;
ip = i*10/zgrid;
}

// p0 is impact parameter for resonance at opaque surface
// p1 is impact parameter for resonance at inner boundary
// p2 is impact parameter for resonance at outer boundary
```

```
double p0 = w_opaque*sqrt( 1.0 - z*z );
double p1 = w_min   *sqrt( 1.0 - z*z );
double p2 = w_max   *sqrt( 1.0 - z*z );

// Initialize total emergent intensity for current
// wavelength shift.

double I = 0.0;

// First calculate the contribution due to each resonance
// point within the opaque sphere.  For the impact
// parameter at each such point, the observer sees directly
// the surface of the opaque sphere.

// line of sight to disk
I += p0*p0/2.0;

// Next calculate the contribution due to each resonance
// point between the opaque sphere and the point at which
// the impact parameter equals w_opaque.

if( z > 0 || p1 > w_opaque )
{
// For positive wavelength shift, the points in
// question are in the occulted region.  For p1 >
// w_opaque, every point now under consideration
// lies in the evacuated region between the opaque
// sphere and the inner boundary of the shell.  For
// the impact parameter at each such point, the
// observer sees directly the surface of the opaque
// sphere.

// line of sight to disk
I += (w_opaque*w_opaque - p0*p0)/2.0;
}
else
{
// Each point under consideration lies between the
// opaque sphere and the observer.  Moreover, there
// is, corresponding to some of these points, a
// range of impact parameters for which photons
// from the opaque surface may be scattered out of
// the line of sight.

// Photons from the opaque surface may not be
// scattered out of the line of sight until the
// impact parameter is large enough so that the
// resonance point lies within the shell.  So we
```

```
// integrate contributions from the smaller impact
// parameters for which there is a direct line of
// sight to the opaque surface.

// line of sight to disk
I += (p1*p1 - p0*p0)/2.0;

if( p2 < w_opaque )
{
// A resonance point at the outer boundary
// has impact parameter less than w_opaque.
// So the range of impact parameters for
// which photons from the opaque surface
// may be scattered out of the line of
// sight is [p1,p2].

// scattering away from line of sight
I += trapezoid( f1, p1, p2 );

// There remain some impact parameters for
// which the observer's line of sight ends
// on the opaque surface, but for which the
// resonance point lies outside the shell's
// outer boundary.

// line of sight to disk
I += (w_opaque*w_opaque - p2*p2)/2.0;
}
else
{
// A resonance point at the outer boundary
// has impact parameter greater than
// w_opaque.  So, resonance points within
// the shell extend at least as far as the
// limb of the opaque sphere.

// scattering away from line of sight
I += trapezoid( f1, p1, w_opaque );
}
}

// Finally add the contribution from each resonance point
// with impact parameter larger than w_opaque.

if( p1 > w_opaque )
{
// A resonance point at the inner boundary of the
// shell has impact parameter larger than w_opaque.
```

```
// Integrate all the way through the shell.

// scattering into the line of sight
I += trapezoid( f2, p1, p2 );
}
else if( p2 > w_opaque )
{
// The resonance point at the limb of the opaque
// sphere is somewhere within the shell.  Integrate
// from the limb to the outer boundary of the
// shell.

// scattering into the line of sight
I += trapezoid( f2, w_opaque, p2 );
}

cout.precision( 11 );
cout << z << " " << I << endl;
}

cerr << " done" << endl;
}
```

A.2.4 Perl Scripts

syn-T.pl

```
#!/usr/bin/perl

open( INFILE, "$ARGV[0]" );

$epsfile =  $ARGV[0];
$epsfile =~ s/\.(.*)/.eps/;

while( <INFILE> )
{
     my $k;               # key
     my @v;               # list of values

     ($k, @v)   = split;
     $param{$k} = \@v;    # reference to a list of values
}

@T        = @{ $param{'T'} };
$x        = $param{ 'x'        }->[0];
$B_inner  = $param{ 'B_inner'  }->[0];
$B_outer  = $param{ 'B_outer'  }->[0];
```

```
$w_min    = $param{ 'w_min'    }->[0];
$w_max    = $param{ 'w_max'    }->[0];
$w_opaque = $param{ 'w_opaque' }->[0];
$res      = $param{ 'res'      }->[0];

$T_OK = 1;
foreach $i ( @T ) { $T_OK = 0 if $i < 0; }

if(
     $T_OK                    &&
     $x        >=  0          &&
     $x        <=  1          &&
     $B_inner  >=  0          &&
     $B_outer  >=  0          &&
     $w_min    >   $w_opaque  &&
     $w_max    >   $w_min     &&
     $w_opaque >   0          &&
     $res      >=  0          &&
     $epsfile  ne  ""            )
{
     # Make a plot of the synthetic line profile.

     $plotline = " plot [:] [0:] ";

     foreach $T ( @T )
     {
          $params = "sgrid 151 zgrid 251 ".
          "T        $T ".
          "x        $x ".
          "B_inner  $B_inner ".
          "B_outer  $B_outer ".
          "w_min    $w_min ".
          "w_max    $w_max ".
          "w_opaque $w_opaque ";

          $dname = "syn-${T}.dat";
          $plotline  .= qq!${sep}"$dname" title "T=$T" with lines lw 3!;
          if( "$sep" eq "" ) { $sep = ", \\\n"; }

          `echo $params | ./test_syn > tmp.dat`;
          `./smooth $res < tmp.dat > $dname`;
     }

     $gpfile = "syn.gp";
     open( GPFILE, ">$gpfile" );
     print GPFILE <<"EOF";

set format "%.2f"
```

```
set grid
set key below
set mxtics
set mytics
set origin 0,0
set size 1.2,2.0
set term postscript eps enhanced 24
set xlabel "{/Times-Roman normalized wavelength shift {/Times-Italic z}}"
set ylabel "{/Times-Roman relative flux}"

set multiplot
set size 1.1,0.9

$plotline

EOF

     # Make a plot of the source function.

     $plotline = " plot (1-sqrt(1-($w_opaque/x)**2))/2 ".
     qq! title "dilution factor" with lines lw 1!;

     foreach $T ( @T )
     {
          $params = "sgrid 151 zgrid 251 ".
          "T         $T ".
          "x         $x ".
          "B_inner   $B_inner ".
          "B_outer   $B_outer ".
          "w_min     $w_min ".
          "w_max     $w_max ".
          "w_opaque $w_opaque ";

          $dname = "sf-${T}.dat";
          $plotline .= qq!${sep} "$dname" title "T=$T" with lines lw 3!;
          `echo $params | ./sf > $dname`;
     }

     print GPFILE <<"EOF";

set origin 0.0,1.1
set xlabel "normalized radial coordinate {/Times-Italic w}"
set ylabel "source function"

$plotline

EOF
     close( GPFILE );
```

```
     `gnuplot $gpfile > $epsfile`;
     `rm -v $gpfile`;
}
else
{
     # The form does not contain valid data.
     print "Choose appropriate parameters.\n";

     print "T @T\n";
     print "x $x\n";
     print "B_inner $B_inner\n";
     print "B_outer $B_outer\n";
     print "w_min $w_min\n";
     print "w_max $w_max\n";
     print "w_opaque $w_opaque\n";
     print "res $res\n";
     print "epsfile $epsfile\n";
}
```

syn.pl

```
#!/usr/bin/perl

open( INFILE, "$ARGV[0]" );

while( <INFILE> )
{
     my @v;
     ($k, @v) = split;
     $param{$k} = \@v;
}

$T        = $param{ 'T'        }->[0];
$x        = $param{ 'x'        }->[0];
$B_inner  = $param{ 'B_inner'  }->[0];
$B_outer  = $param{ 'B_outer'  }->[0];
$w_min    = $param{ 'w_min'    }->[0];
$w_max    = $param{ 'w_max'    }->[0];
$w_opaque = $param{ 'w_opaque' }->[0];
$res      = $param{ 'res'      }->[0];
$epsfile  = $param{ 'epsfile'  }->[0];

if(
     $T        >=  0         &&
     $x        >=  0         &&
     $x        <=  1         &&
     $B_inner  >=  0         &&
```

```
    $B_outer  >=  0         &&
    $w_min    >   $w_opaque &&
    $w_max    >   $w_min    &&
    $w_opaque >   0         &&
    $res      >=  0         &&
    $epsfile  ne  ""           )
{
    # Make a plot of the synthetic line profile.

    $params = "sgrid 151 zgrid 251 ".
        "T         $T ".
        "x         $x ".
        "B_inner   $B_inner ".
        "B_outer   $B_outer ".
        "w_min     $w_min ".
        "w_max     $w_max ".
        "w_opaque  $w_opaque ";

    $dname     = "syn.dat";
    $sname     = "smooth.dat";

    $plotline  = qq! plot "$dname" title "synthetic profile" !.
        qq! with lines lw 1, "$sname" title !.
        qq! "smoothed profile" with !.
        qq! lines lw 3 !;

    `echo $params | /usr/local/bin/test_syn > $dname`;
    `/usr/local/bin/smooth $res < $dname > $sname`;

    $gpfile = "syn.gp";

    open( GPFILE, ">$gpfile" );
    print GPFILE <<"EOF";

set format "%.2f"
set grid
set key below
set mxtics
set mytics
set origin 0,0
set size 1.2,2.0
set term postscript eps enhanced 24
set xlabel "{/Times-Roman normalized wavelength shift}"
set ylabel "{/Times-Roman relative flux}"

set multiplot
set size 1.1,0.9
```

```
$plotline

EOF

     # Make a plot of the source function.

     $dname    = "sf.dat";

     $plotline = qq! plot (1-sqrt(1-($w_opaque/x)**2))/2      !.
          qq! title "dilution factor" with lines lw 1, !.
          qq! "$dname" title "source function" with    !.
          qq! lines lw 3                               !;

     `echo $params | /usr/local/bin/sf > $dname`;

     print GPFILE <<"EOF";

set origin 0.0,1.1
set xlabel "normalized radial coordinate {/Times-Italic w}"
set ylabel "source function"

$plotline

EOF
     close( GPFILE );
     `gnuplot $gpfile > $epsfile`;
     `rm -v $gpfile`;
}
else
{
     # The form does not contain valid data.
     print "Choose appropriate parameters.\n";

     print "T $T\n";
     print "x $x\n";
     print "B_inner $B_inner\n";
     print "B_outer $B_outer\n";
     print "w_min $w_min\n";
     print "w_max $w_max\n";
     print "w_opaque $w_opaque\n";
     print "res $res\n";
     print "epsfile $epsfile\n";
}
```