A NON-LTE ANALYSIS OF A SAMPLE OF O STARS SELECTED FROM GALACTIC OB ASSOCIATIONS

Thesis by

Kent Grimmett Budge

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, California

1990

(Submitted February 16, 1990)

ACKNOWLEDGEMENTS

I wish to acknowledge the support of the National Science Foundation through a Graduate Research Fellowship. I am also grateful for the support of Wal Sargent, Judith Cohen, Ann Boesgaard, Jeremy Mould, and Ken Libbrecht, all of whom helped cover the balance of my tuition and stipend at one time or another through research assistantships and/or provided the computer facilities needed for research.

I am particularly grateful to Ann for arranging a very productive trip to the Institute for Astronomy of the University of Hawaii, whose hospitality I wish to acknowledge.

Many of the computations for this research were made at the San Diego Supercomputer Center, whose support I also wish to acknowledge.

ABSTRACT

The tables of non-LTE line profiles and equivalent widths published by Mihalas and his collaborators [33], [7], [35] have been revised and extended to four different values of the abundance ratio He/H. Bolometric corrections have been calculated for V magnitudes. The theoretical line profiles have been fit to echelle spectrograms of 22 galactic O stars by χ^2 minimization. It is found that the stars with the lowest surface gravities are fitted best by theoretical spectra with unexpectedly high helium abundances (He/H $\sim$ 0.50), while the stars with higher surface gravities are fitted best by theoretical spectra with He/H$\sim$ 0.10, the accepted cosmic ratio. This suggests a systematic failure of conventional non-LTE, plane-parallel models for the more luminous O stars, probably as a result of the neglect of geometrical dilution.

The formula, log He/H = 1.1234-0.4791 $\log g$, gives a good fit to the relation between the apparent helium abundance and $\log g$. Using this relationship, the apparent abundances have been reduced to what are probably true abundances relative to the normal cosmic abundance. It is found that there is no significant difference in the average helium abundances of the associations observed. However, the stars HD 12993, HD 242908, and HD 193595 may be blue stragglers with moderately enhanced helium abundance (He/H $\sim$ 0.19).

Relative carbon abundances have been determined empirically by comparison of the CIV 5812Å and HeII 4542Å equivalent widths. It is found that the association Cyg OB2 is overabundant in carbon by $\sim$ 50%. Likewise, the blue straggler HD 236894 is underabundant in carbon by a factor of two.

The estimated effective temperatures of the sample are compared to the previously accepted calibration of MK spectral types to the effective temperature. Estimates of the radii and masses of the stars in the sample have been calculated from their physical parameters and their absolute visual magnitudes.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The evolution of very young stellar populations is of great interest, since these stars have a decisive influence on the interstellar medium and the chemical evolution of the universe and are generally believed to be the precursors of Type II supernovae.

However, while young, massive stars are bright and easily observed, the observations are difficult to interpret. This is unfortunate, since recent developments in the theory of stellar evolution have led to detailed predictions of the evolution of the stellar parameters and surface chemical abundances of such stars [14], [30] that beg to be confirmed observationally.

This situation is changing. Advances in computational methods and capacity have made it possible to calculate efficiently model stellar atmospheres in which the approximation of local thermodynamic equilibrium (LTE) is not made [27]. Since departures from LTE give rise to many of the difficulties in analyzing the spectra of very hot stars, the new computational methods promise to revolutionize the interpretation of such spectra.

In addition, advances in instrumentation have made it possible to observe large numbers of moderately bright objects in a short time. Faint objects that were previously inaccessible may now be observed. For example, the McCarthy echelle spectrograph at the Palomar 2m telescope [29] can obtain spectrograms of 12th magnitude stars with a moderately high signal-to-noise ratio (~ 100) in the best orders in less than two hours. Thus, one can build up statistics by observing a large and varied sample.

1.1 MASS LOSS AND EVOLUTION OF YOUNG POPULATIONS

In a recent review [14], Chiosi and Maeder have summarized the results of recent theoretical calculations of the evolution of massive stars in which both convective overshoot and mass loss have been taken into account. They come to the following important conclusions:

1. Convective overshoot in the cores of evolving main sequence O stars results in the availability of significantly more hydrogen for core burning than previously thought. This results in the observed

extension of the main-sequence band to supergiants of type A0 [11], [32].

2. Mass loss explains the large numbers of red supergiants observed in the initial mass range from 20 to 50 $M_\odot$ and the absence of any red supergiants brighter than $M_{\rm bol} < -9.5$ [24] and may explain the compositional peculiarities of Wolf-Rayet stars that are not contact binaries. The statistics on red supergiants *vs.* blue supergiants are given in Table 1 .

They find that models calculated with a moderate degree of overshooting and average observed mass-loss rates have the following properties:

1. For stars initially more massive than 60 $M_\odot$, the mass-loss rates are so great that the outer layers of the stars are entirely peeled away, resulting in a bare helium core, quasi-homogeneous evolution, and no red supergiants. The stars pass through a luminous blue variable (LBV) phase and become Wolf-Rayet stars.

The abundance changes associated with this evolutionary scheme are very marked: The surface hydrogen abundance is reduced by a factor of 2 by the time the star reaches its lowest surface temperature of about 11000K, and hydrogen disappears completely as the star returns to the left side of the color-magnitude diagram. Later, triple-α products appear, and the surface helium mass fraction is reduced by a factor of two at the end of the carbon-abundant Wolf-Rayet (WC) phase.

2. For stars with initial masses in the range of 25 to 60 $M_\odot$, the evolution of the star is sensitive to the exact rate of mass loss at different evolutionary phases. Mass loss on the main sequence favors the formation of red supergiants by suppressing the intermediate convection zone and thus reducing the amount of core hydrogen available. Mass loss during the red supergiant phase, on the other hand, shortens the red supergiant phase. If mass loss during the red supergiant phase is sufficiently great, the star becomes a Wolf-Rayet star.

During the evolution from O star to red supergiant, CNO-processed material comes to the surface; the carbon-to-nitrogen ratio thus goes from the cosmic value of 4 prior to the red supergiant phase to the CNO equilibrium value of 0.025 after the star returns to the left side of the color-magnitude diagram. The surface hydrogen abundance drops abruptly to zero halfway through

Table 1. Theoretical Luminosities and Lifetimes (in units of 10^6 years)

Initial mass ($M/M_\odot$)	$\log L/L_\odot$	H-burning lifetime	He-burning lifetime	Fraction of He phase at $\log T_{eff} > 4.2$ (%)	Fraction of He phase at $\log T_{eff} < 3.8$ (%)
15	4.25	11.582	2.344	1	55
25	4.85	6.624	1.222	16	38
40	5.34	4.530	0.856	58	14
60	5.70	3.708	0.707	93	0
85	5.98	3.253	0.760	100	0
120	6.23	2.810	0.840	100	0

After Chiosi and Maeder [14]

the nitrogen-abundant Wolf-Rayet (WN) phase, accounting for the transition from WN stars with hydrogen (WNL) to WN stars without hydrogen (WNE). Finally, triple-α products appear very abruptly, the helium abundance goes to zero, and the star becomes a WC star.

3. Less massive stars always become red supergiants, but if the mass-loss rates are low, the star makes a blueward loop during the red supergiant phase and may cross the Cepheid instability strip repeatedly. Larger mass-loss rates tend to suppress this loop.

As with more massive stars, CNO-cycled material appears during the red supergiant phase and produces marked abundance changes. However, for stars of 15 $M_{\odot}$ or less, the C/N ratio does not drop below 1 during the first red supergiant phase and equilibrium CNO abundances are not yet established during the blueward loop.

1.1.1 Observational Consequences

The theoretical predictions described by Chiosi and Maeder may be checked observationally on several points. First, one may look for the surface abundance changes predicted for the more massive stars. For the most massive, one may expect to see a significant increase in the helium content while the star is still near the main sequence. At lower masses, an increase of the helium abundance will be evident when the star returns to the left side of the color-magnitude diagram. Unfortunately, the high excitation of the optical helium spectrum makes it impossible to trace the helium abundance during the cooler phases of the star's evolution, as we would otherwise wish to do.

The carbon-to-nitrogen ratio should change markedly at the same evolutionary phases at which the helium abundance changes. Unfortunately, although molecules of both elements are visible in red supergiants, the molecular data and model atmospheres are in a very primitive state at present and one cannot hope to trace accurately the abundances at cooler temperatures. One is restricted to an analysis of the atomic spectra at the same temperatures at which the helium abundance may be studied.

Finally, one may attempt to determine the stellar parameters T_{eff}, $\log g$, and $M_{\rm bol}$ with sufficient accuracy to make a reasonable estimate of the stellar radii and masses. One then detects the effects of mass loss in the same way that one detects mass exchange in binary systems: The stars that have experienced mass loss will be overluminous for their masses. However, since both the bolometric magnitude $M_{\rm bol}$ and $\log g$ are difficult to determine accurately, this procedure is of limited usefulness.

The determination of T_{eff} and $\log g$ for a sufficiently large sample of stars also permits an accurate calibration of the mapping of the (T_{eff}, $\log g$) plane to the two-dimensional MK spectral types. This is of value for studies in which the observed spectral types are used to construct theoretical temperature-luminosity diagrams for OB associations with known distance moduli [24], [25].

1.2 THE ANALYSIS OF O STARS USING SIMPLE NON-LTE MODELS

Rapid progress is being made in the computation of self-consistent models of radiation-driven stellar winds [2], [16] and in the application of the effects of wind blanketing to photosphere models [1], [12], [47]. Likewise, the new computational techniques are making possible an attack on the difficult problem of line blanketing in non-LTE model atmospheres [5], [48]. One may anticipate that it will soon be possible to calculate efficiently non-plane-parallel non-LTE models as well.

These refinements have the unfortunate effect of greatly increasing the computational cost of analyzing a particular star. Not only does each model computation take more computer time than one for a plane-parallel, unblanketed, non-LTE model, but the fact that such models require additional parameters prevents the computation of a master model grid for use in the analysis. One must perform a series of computations for each individual star analyzed. Since preliminary indications are that the effects of blanketing (both wind- and line-) and of departures from plane-parallel geometry are small for many O stars, it is reasonable to investigate what can be learned, using a grid of the simpler non-LTE models.

With these motivations, a sample of O stars stars selected from a number of galactic OB associations [24] has been observed with the McCarthy echelle at the Palomar 2m telescope. The observed line profiles have been compared with a grid of theoretical line profiles calculated at the San Diego Supercomputer Center, using non-LTE model atmospheres with a range of effective temperatures, surface gravities, and helium abundance fraction. The method of χ^2 minimization has been used to determine the choice of stellar parameters, giving the best fit of theoretical profiles to the observed profiles for each star.

2. OBSERVATIONS

2.1 OBJECT SELECTION

A sample of 36 stars from eight galactic associations was chosen for study. The particular OB associations sampled were selected to cover a range of ages (based on the relative numbers of O, B and M supergiants known to be members of each association) and galactocentric distances [24]. Table 2 lists the associations and stars selected for study. Figure 1 gives the theoretical H-R diagrams for the selected associations. In the diagram at upper left, circles represent members of Cyg OB2 and triangles, members of Aur OB2. In the upper right diagram, circles represent members of Cyg OB1 and triangles denote members of Cep OB2. In the lower left diagram, circles denote Gem OB1 and triangles denote Per OB1. The final diagram is a composite for all the associations observed. In all diagrams, filled symbols indicate stars for which parameters were eventually derived by the method of analysis described here. Parameters for other stars are taken from the paper by Humphreys [24]. The stars analyzed all fall in the region of moderate-luminosity O stars; cooler objects and Of stars were not observed or were dropped from the sample. The associations themselves appear to be of slightly different ages, with Aur OB2 and Cyg OB2 being the youngest and Per OB1 and Gem OB1 being the oldest.

The stars selected for analysis included no known binaries, and where estimates of $v \sin i$ were available, no stars with $v \sin i > 160$ km/sec. However, such data are scanty, and a number of the objects selected were found to have excessive $v \sin i$ and were not included in the final analysis. Likewise, stars showing strong Of characteristics were eventually dropped from the sample because the plane-parallel models described later were found to be completely inadequate to describe such stars. Those O stars observed but not analyzed are indicated in italics in Table 2.

Spectroscopic binarity is a more difficult problem, since for most companions bright enough to affect the measurements, the spectra of the two components would be nearly indistinguishable unless separated by large orbital Doppler shifts. Thus, it is possible that a number of spectroscopic

Table 2. O Stars Observed

Star	Spectrum	V	B-V	M_v	Remarks
Per OB1	R=11.73kpc				
236894	O8 V	9.37	0.19	-3.9	
12993	O6.5V	8.95	0.20	-4.4	
13022	O9.5 II-III	8.76	0.32	-4.9	
13268	O7:	8.18	0.13	-5.0	Large $v \sin i$
Aur OB2	R=13.14kpc				
242908	O4 V	9.04	0.28	-5.3	NGC 1893
242926	O7 V	9.35	0.34	-5.1	NGC 1893
242935	O7 V	9.43	0.20	-4.6	NGC 1893
35619	O7 V	8.55	0.24	-5.6	
Gem OB1	R=11.49kpc				
42088	O6.5	7.55	0.07	-4.5	
254755	O9 Vp	8.84	0.60	-4.8	
256035	O9 Vp	9.16	0.55	-4.3	Large $v \sin i$
Cyg OB1	R=9.71kpc				
193514	O7 Ibf	7.40	0.45	-6.2	Of
193595	O7	8.72	0.36	-4.6	
193682	O5	8.41	0.51	-5.4	Large $v \sin i$
194094	O9 III	9.02	0.59	-5.0	
194280	OC9.7 Iab	8.39	0.76	-6.1	
228841	O6.5 V	8.94	0.56	-5.0	Large $v \sin i$
229234	O9.5 III	8.92	0.77	-5.6	NGC 6913
+36 4063	O9.5 Ib	9.71	1.14	-5.9	Confused
Cyg OB2	R=9.85kpc				
# 1	O9 V	11.09	1.42	-5.4	Large $v \sin i$
# 3	O9:	10.22	1.61	-6.8	Large $v \sin i$
# 4	O7 III	10.22	1.17	-5.6	+40 4219
# 6	O8 V	10.67	1.22	-5.2	Large $v \sin i$
# 7	O3 If	10.50	1.44	-6.1	Of
# 8A	O6 Ib	8.98	1.29	-7.2	+40 4227 Large $v \sin i$
# 8B	O8:	10.31	1.35	-6.0	
# 8C	O5 III	10.08	1.35	-6.2	Large $v \sin i$
# 9	O5 If	10.80	1.93	-7.3	Of
# 10	O9.5 Ia	10.04	1.43	-6.5	+41 3804
# 11	O5 If	10.04	1.43	-6.5	+41 3807 Of
Cep OB2	R=10.21kpc				
204827	O9.5 V	7.95	0.81	-5.0	Tr 37
207198	O9 Ib-II	5.96	0.31	-5.5	
207538	O9.5 V	7.31	0.33	-4.2	
209975	O9.5 Ib	5.11	0.09	-5.7	19 Cep
Miscellaneous					
9 Sgr	O4V	5.97	0.03	-6.1	
λ Ori	O8	3.66	-0.19	-5.2	

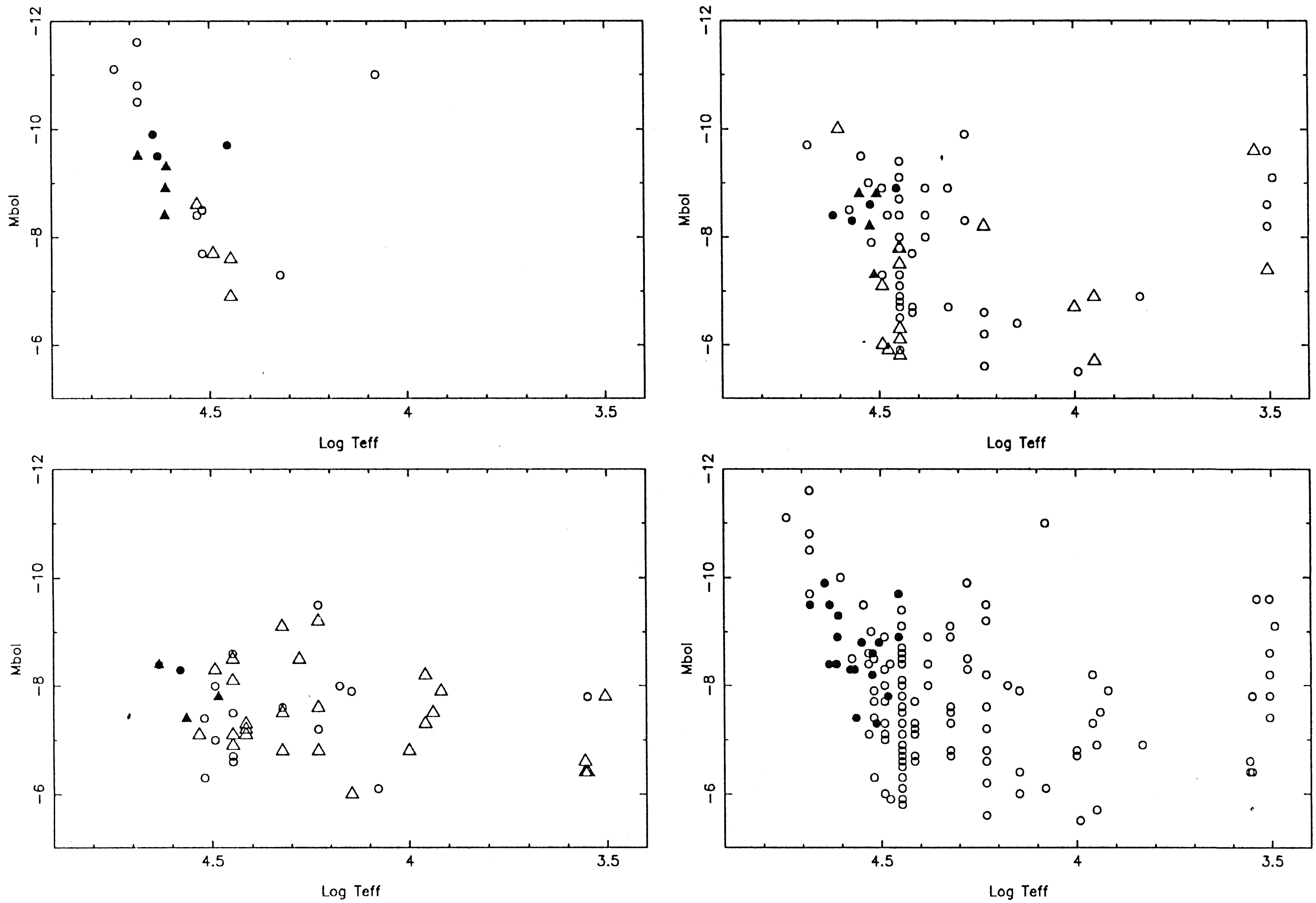

Figure 1. H-R diagram of observed associations. See text for explanation of symbols.

binaries have crept into the sample.

The selection of stars with a low $v \sin i$ will bias the sample towards slow rotators. Since the ultimate objective of this research is to test the models of Chiosi and Maeder [14], in which rotation is neglected, this bias may actually be advantageous.

2.2 DATA COLLECTION AND REDUCTIONS

The sample of stars was observed with the McCarthy echelle spectrograph on the Palomar 2m telescope [29]. This instrument covers the entire optical spectrum from about 3650Å to 8000Å, with partial coverage to 10000Å. Each order covers from 100Å (in the near infrared) to 50Å (in the violet). The detector is a Texas Instruments 800 × 800-pixel charge-coupled device (CCD) cooled with liquid nitrogen. It is routinely soaked in oxygen and flooded with ultraviolet light prior to use to improve the quantum efficiency at short wavelengths.

For each observation, the signal-to-noise ratio for the optimal orders, those covering the wavelength range from about 5850Å to 6550Å, was better than 100. The signal-to-noise ratio usually decreased rapidly farther towards the blue, both because of the decreased sensitivity of the CCD detector and because most of the objects observed show significant reddening from interstellar dust. The stars from OB2 Cyg were particularly difficult in this respect, with a typical reddening of $E(B-V) \sim 1.5$, or greater. In most cases it was impossible to obtain a high signal in the blue orders without saturating the red orders and ruining the exposure. As a result, the analysis was restricted to lines longward of the Hγ line at 4342Å.

Flat fields for calibration were prepared by summing frames obtained by exposing the echelle to a diffuse incandescent source through a series of broad-band filters. This made it possible to obtain a satisfactory exposure level in the blue orders of the echelle without overexposing the orders in the red. A thorium-argon lamp was used to obtain a comparison spectrum for the wavelength calibration.

The spectrograms were reduced using the echelle routines in the FIGARO data processing package [29]. The flat field, comparison, and object spectrograms were first remapped to remove curvature from the orders (using the FIGARO routine CDIST). A normalized flat field was then obtained by dividing the raw flat field by a copy of the flat field that had been smoothed in the direction parallel to the orders; this ensured that the flat field was close to unity everywhere (while retaining pixel-to-pixel variations and interference fringes), so that noisier, low-signal rows in each order were not artificially amplified when the flat field was used to calibrate the object spectra. After the observed spectrograms were divided by the normalized flat field, the rows making up each order were summed and an estimate of the crosstalk between orders was subtracted from each order (using the FIGARO routine ECHTRACT).

The profiles of the hydrogen and helium lines were obtained by fitting a polynomial to the two orders adjacent to each order containing a line of interest; the central order was then divided by the average of the two polynomials. This provided a way of estimating the continuum level even in the presence of very strong wings, as illustrated by the profile obtained this way for $H\beta$ in Vega (Figure 2). Unfortunately, the results were rarely this good for the spectrograms of O stars presented here; the Stark wings are much shallower in O stars, and confusion from stellar and interstellar lines and bands resulted in some uncertainty in the continuum level.

2.3 DATA QUALITY AND SYSTEMATIC ERRORS

The chief source of uncertainty in the observations is ambiguity in the continuum level, rather than Poisson noise. The technique of interpolating the continuum from adjacent orders is most effective for the Balmer lines, with their strong wings. The much narrower neutral and ionized helium lines are much more sensitive to small, local errors in the continuum-level determination.

Such errors arise primarily from two sources: imperfect flat-field calibration and confusion with stellar or interstellar lines and bands. The former effect is noticeable for the HeI 4471Å line and for the red wing of the $H\gamma$ Balmer line (which is located near the end of an echelle order). Here one sees that the local continuum level is not flat. The effect is reproducible from night to night despite

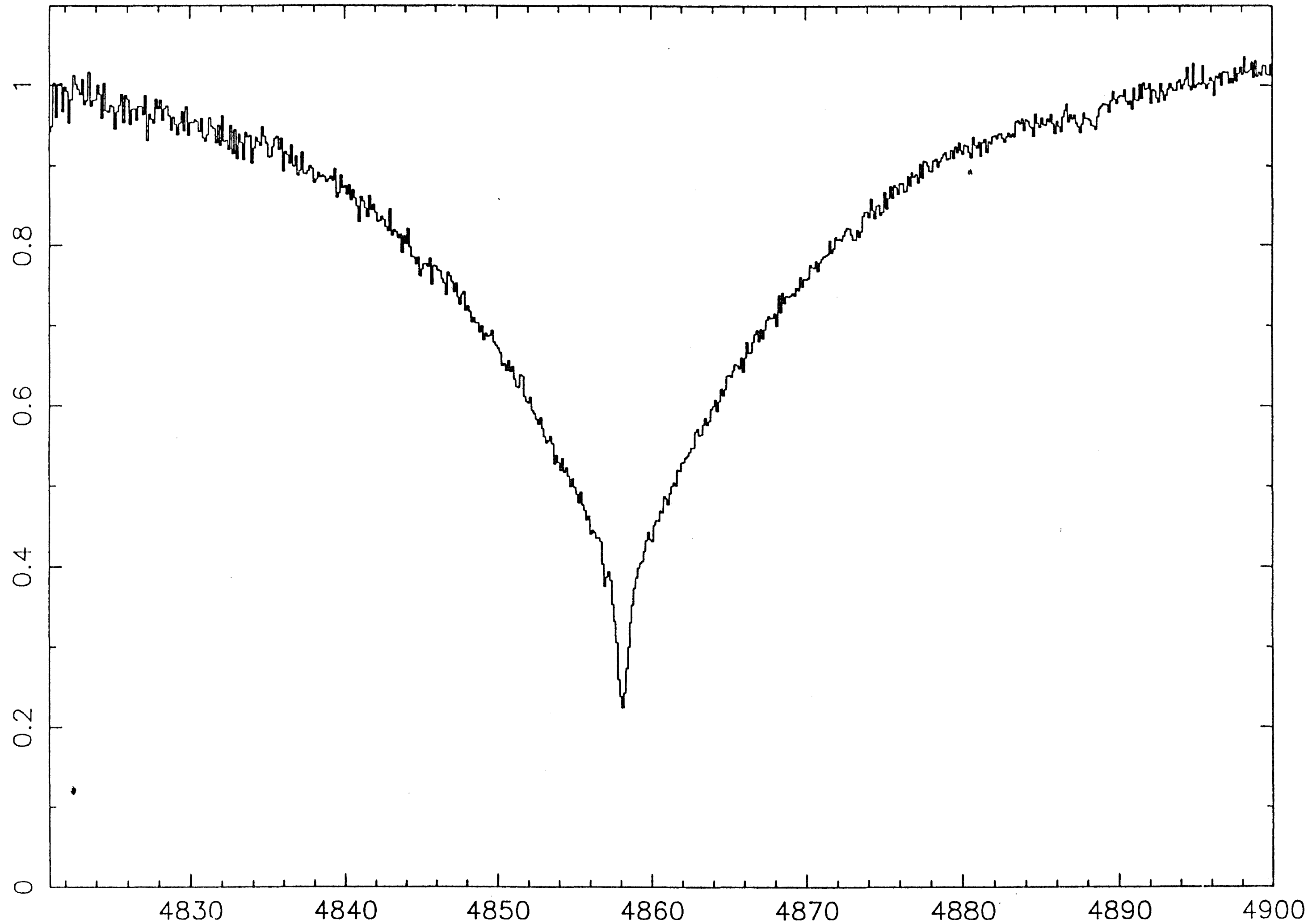

Figure 2. Continuum-flattened data for Hβ in Vega. The line profile has been normalized to the continuum level interpolated from the adjacent two echelle orders.

the fact that fresh flat fields were obtained and used each night.

The second effect is seen, e.g., with the HeI 6678Å line and the Balmer lines. In the case of HeI 6678Å, a diffuse feature (probably intrinsic to the star rather than an interstellar feature) badly confuses the red wing of the line. Likewise, stellar OII lines confuse the already uncertain calibration of the red wing of Hγ.

In general, the effects of poor flat-field calibration are more troublesome than the effects of interstellar bands. Future work on these objects would best be done on an instrument with superior photometric characteristics, even at the expense of a lower signal-to-noise ratio and/or resolution. However, the confusion from lines and bands sets an ultimate limit on the quality of observational line profiles.

Another uncertainty in the data is the absolute wavelength calibration. The FIGARO software for carrying out the calibration is computationally expensive; since little information about the physical parameters is contained in the wavelength of the line centers, it was decided to use a single wavelength calibration for all spectrograms. However, thorium-argon comparison spectra were obtained each night for possible future use in determining radial velocities.

Finally, a particular difficulty with echelle spectrograms is the problem of removing crosstalk between orders and accounting for scattered light in the instrument. The McCarthy software [29] for reducing echelle data does an excellent job of accounting for both effects for spectra that are well focused on nights of fair-to-excellent seeing. It is unlikely that such background light has significantly affected the line residual intensities.

2.4 OBSERVED EQUIVALENT WIDTHS

Although profiles were used in the fit of theory to data described in Chapter 4, the equivalent widths of helium lines contain useful information in a concise form. Table 3 gives equivalent widths (in mÅ) for lines of neutral and ionized helium, along with equivalent widths of the prominent CIII emission line at 5696Å and the CIV absorption lines at 5801Å and 5812Å. Some qualitative analysis

Table 3. Equivalent Widths of Helium and Carbon Lines

Object	HeI 4471	4713	4922	5016	5876	6678
Per OB1						
236894	770 ± 30	160 ± 50	598 ± 50	267 ± 10	893 ± 30	679 ± 50
12993	522	133	271	171	827	209
13022	695	230:	582	296	1022	733
Aur OB2						
242908	287	< 20	50	25:	514	< 22
242926	574	94	315	128	714	246:
242935	448	74	444	126	572	179
35619	532	115	242	126	800	299
Gem OB1						
42088	411	63	169	117	663	233
254755	577	175	464	226	911	527
Cyg OB1						
193595	606	197	294	238	916	509:
194094	567	225	402	311	952	658
194280	672	200	495	331	1035	787
229234	616	226	397	303	1027	660:
Cyg OB2						
# 4	425	106	200	197	892	544:
# 8B	347	< 60	< 26	116:	722	–
# 10	583	234	444	283	900	692
Cep OB2						
204827	801	186	540	292	780	634
207198	763	264	394	313	1101	721
207538	885	246	548	294	826	726
209975	822	245	405	312	1064	761
Miscellaneous						
9 Sgr	182	< 20	< 20	< 20	330	470:
λ Ori	589	149	254	238	974	490

Table 3. (Cont.)

Object	HeII 4542	4686	5412	CIII 5696	CIV 5801	5812
Per OB1						
236894	273 ± 70	466 ± 100	471 ± 40	$< 20 \pm 20$	72 ± 80	45 ± 50
12993	567	660	863	-123	285	225
13022	143	284	224	< 20	59	28:
Aur OB2						
242908	608	688	820	-92	278	240
242926	430	660	759	-111	214	170
242935	295	471:	509	-63	219	141
35619	382	643:	871	–	276	189
Gem OB1						
42088	479	683	715	-109	340	258
254755	318	259	433	-145	239	166
Cyg OB1						
193595	651	756	903	-170	306	230
194094	251	509	388	-61:	161	115
194280	99:	232	209	56	46:	47
229234	230	439	371	88	204:	206:
Cyg OB2						
# 4	446	439	741	-302	490:	422
# 8B	445	105	745	-401	638:	476
# 10	117	151	237	-166	–	266
Cep OB2						
204827	173	357:	252	< 80	64:	53:
207198	298	359:	511	-302	231	175
207538	125:	385	223	90	38	< 13
209975	224	230	372	-235	175	137
Miscellaneous						
9 Sgr	625	630	909	-32	161	132
λ Ori	389	484	624	-211	349	257

of the carbon features is presented in Chapter 5. The detectability limit for weak lines was ~ 80 mÅ for the stars with $v \sin i > 100$ km/sec and ~ 20 mÅ for stars with $v \sin i \sim 60$ km/sec.

The equivalent widths were measured using the FIGARO routine GAUSS. This routine determines the continuum level by fitting a polynomial to the continuum with iterative rejection of outlier points. Equivalent widths may then be obtained both by simple integration and by fitting of Gaussian profiles. The values quoted here were obtained by simple integration and compared with the results from fitting Gaussian profiles to estimate the uncertainties in the values. This allows for the effects of imperfect continuum normalization and blends and the uncertainty in the wings of the Stark-broadened ionized helium lines.

The Balmer lines, since they are dominated by the shallow $\Delta\nu^{-5.2}$ Stark wings, are difficult to measure equivalent widths for. It was quite common to measure values for the equivalent widths that were up to a factor of 2 smaller than those of the theoretical profiles that best matched the observed profiles. Hence, no observed equivalent widths for the Balmer lines are presented here. The profiles are presented later, along with the helium line profiles and the corresponding best theoretical fits.

3. THEORY

3.1 BACKGROUND

It is generally recognized that model atmospheres and synthetic spectra for hot stars cannot be approximated usefully under the assumption of local thermodynamic equilibrium [34]. The calculation of non-LTE model atmospheres is considerably more difficult than the calculation of LTE model atmospheres, primarily because of the detailed dependence of the source function at each frequency on the radiation field at all frequencies.

An LTE calculation may be thought of as a very restricted scattering problem in which the source function at each frequency is uniquely determined by a single weighted integral of the radiation field, the constraint of radiative equilibrium, which is characterized by a single quantity, the temperature. That is, at each depth point,

$$S(\nu) = B_\nu(T) \tag{1}$$

subject to the condition

$$\int_0^\infty (J_\nu - B_\nu(T))\kappa_\nu d\nu = 0. \tag{2}$$

(Convective energy transport is negligible in the atmospheres of O stars.) Thus, it is relatively easy to organize the transfer equations describing the model atmosphere by frequency in such a way that the resulting computational algorithm scales linearly with the number of frequencies [41]. One determines (to first order) the perturbation to the single integral of interest, the condition of radiative transfer, resulting from the perturbation of the radiation field at each frequency and depth: Since the condition of radiative equilibrium is only approximately satisfied by the starting model atmosphere (obtained by a gray atmosphere calculation or some other simple analytic approximation), one sets the perturbation in the condition of radiative equilibrium equal to the opposite of the current value of this integral (which should be, but is not yet, equal to zero):

$$-\int_0^\infty (J_\nu^0 - B_\nu^0(T^0))\kappa_\nu^0 d\nu = \Delta \int_0^\infty (J_\nu - B_\nu(T))\kappa_\nu d\nu = \int_0^\infty (\Delta J_\nu - \frac{\partial B_\nu}{\partial T}\Delta T)\kappa_\nu d\nu, \tag{3}$$

which, after the integral is replaced by a quadrature sum, has the form

$$A = \sum_N W_N \Delta J_\nu - \sum_N X_N \Delta T, \tag{4}$$

where A, W_N, and X_N are scalars. These coefficients are calculated for each depth point, requiring of order ND operations (where N is the number of frequency points in the quadrature sum and D the number of depth points). Here the opacity is assumed to be constant; this approximation may be dropped (as was done for the actual calculations) without changing the form of the last equation.

The transfer equation may then be used to determine the perturbations in the radiation field resulting from a perturbation to T; one gets equations of the form

$$\Delta J_\nu = \sum_D Y_D \Delta T_D. \tag{5}$$

The scalar coefficients Y_D are calculated for each frequency, and ΔJ_ν is eliminated from Equation 4; this takes of order ND^3 operations. One is left with the system of equations

$$\mathbf{A} = \mathbf{M \Delta T}, \tag{6}$$

where $\mathbf{A}$ and Δ $\mathbf{T}$ represent vectors over all depths and $\mathbf{M}$ is a $D \times D$ matrix [34]. This system is solved for the corrections to T (and thus the source function at each depth) in of order D^3 operations. The computational cost is dominated by the calculation of the coefficients to Equation 5 so that the calculation time scales as ND^3.

In the case of a non-LTE calculation, on the other hand, the source function is defined by a very large number of radiation field integrals, each with a different weighting function, in addition to the temperature; that is,

$$S(\nu) = S_\nu(T, N_L), \tag{7}$$

subject to the same constraint of radiative equilibrium (Equation 2), and *additional constraints* of the general form

$$N_L \left(\sum_{L' \neq L} A_{L,L'} \int_0^\infty J_\nu \kappa_{\nu,L \mapsto L'} d\nu + C_{L \mapsto L'}(T) \right) = \sum_{L' \neq L} N_{L'} \left(B_{L,L'} \int_0^\infty J_\nu \kappa_{\nu,L' \mapsto L} d\nu + C_{L' \mapsto L}(T) \right). \tag{8}$$

Here L represents the number of atomic levels and N_L represents the occupation numbers for these levels, which characterize the constraint equations. The quantity $C_{L' \mapsto L}(T)$ includes spontaneous radiative de-excitation or recombination.

If one attempted to proceed as with the LTE calculation, organizing by frequency, one would eventually end up with equations of the form

$$\mathbf{A} = \mathbf{M}\mathbf{\Delta}\mathbf{N}, \tag{9}$$

where $\mathbf{A}$ and Δ $\mathbf{N}$ represent vectors over all depths *and all constraints* and $\mathbf{M}$ is the corresponding square matrix. $\mathbf{M}$ will have $\sim$ 500-5000 rows and columns (since for a typical non-LTE calculation, there will be $\sim$ 50 depth points and $\sim 10 - 100$ constraints), and the computational cost becomes astronomical.

Thus, one cannot organize the problem by frequency, and one is forced to organize it by depth; the resulting equations have the form

$$\mathbf{Q} = \mathbf{A}\mathbf{\Delta}\psi_{d-1} + \mathbf{B}\mathbf{\Delta}\psi_d + \mathbf{C}\mathbf{\Delta}\psi_{d+1} \tag{10}$$

at each depth point d [33]; here ψ represents a vector over all frequencies and constraint equations at a single depth point. The only full matrices that must be inverted are $(N + L) \times (N + L)$ matrices. Thus, the computation time scales as $(N + L)^3 D$. Although this is much more practical than the organization by frequency, it puts a severe limit on the number of frequencies and atomic levels that may included in the calculation. The models calculated by Mihalas, Auer, and Heasley [36] (hereafter MAH) were restricted to 105 frequencies and 11 atomic levels and were expensive to compute even for such a simple description of the radiation field and atomic physics.

Recently Anderson [6] has pointed out that it should be possible to sum many transfer equations describing the same line or continuum into a single equation, approximately describing the total radiative energy density in that line or continuum. Thus, the number of equations linearized is greatly reduced, since many quantities ΔJ_ν are reduced to a few quantities ΔE_b representing the total energy density in a "block" of frequencies describing one line or continuum. In addition, the presence of both the occupation numbers (each characterizing a constraint equation) and the energy densities in the solution vector ψ is redundant, since the corrections to the one are given by the corrections in the other (to the same order of approximation as the overall computation); thus, the large solution vector ψ containing corrections to all frequencies and occupation numbers may be replaced by a much smaller vector ψ_{And} containing only corrections to the temperature and a few energy densities E_b. The temperature could, in principle, be eliminated as well, since, like the other variables of constraint, it is uniquely defined by the radiation field; but since its presence would introduce non-linearities into the subsequent calculation of the new values for the constraint variables (through collision terms), it is treated explicitly.

The actual numerical approximation made is that the detailed distribution of the radiation field in a particular block of frequencies dominated by a single line or continuum and formed at similar depths is fixed. One then solves (to first order) for the factor by which all the mean intensities in each frequency block should be multiplied in order to better satisfy the transfer and constraint equations. The resulting equations are similar in form to Equation 10. These are solved and the mean intensities in each block are multiplied by the correcting factor for that block. The level populations are then revised using the modified mean intensities and temperatures, the transfer equations for all frequencies are solved using the resulting source function, and the distribution of the radiation field in each frequency block is updated. The whole procedure is iterated to convergence.

The resulting method converges linearly rather than quadratically (as is characteristic of the complete linearization procedure), but the convergence rate is acceptable for moderate error tolerances. It typically requires less than fifteen iterations to reach a tolerance of 0.01% in the surface flux. The number of operations per iteration scales as DB^3, where B is the number of frequency blocks.

3.2 MODEL ATMOSPHERE AND SYNTHETIC SPECTRUM CALCULATIONS

The calculation of theoretical line profiles proceeds in several steps. A gray model atmosphere is first calculated, using an approximate T-τ relationship. This model is then used as the starting approximation for an LTE model atmosphere code. The resulting LTE model is then used as the starting approximation for a non-LTE code employing the Anderson algorithm.

At this point, the temperature and electron density at each depth in the model atmosphere is fixed. However, an effort is made to improve the accuracy of the hydrogen and helium level populations further by carrying out a calculation in which the level populations of one element are refined while holding the populations of the other element and the temperature and electron density fixed. This permits an approximate treatment of a larger number of transitions for each element, using more detailed line profiles.

Finally, profiles for selected lines are calculated, using the most accurate available theoretical line broadening data.

The whole set of computer codes is written in FORTRAN 77 for use on a CRAY or other compatible vector processor. The codes proved to be well suited for vector optimization, and it was usually possible to vectorize by frequency index rather than by depth index (the number of frequencies being much greater than the number of depths).

3.2.1 Approximations

The chief approximations made in the theoretical calculations are (1) the approximation of plane-parallel geometry; (2) the neglect of metallic line blanketing; and (3) the approximations in the atomic physics.

The use of plane-parallel models has a number of practical advantages. First, it is considerably more difficult and computationally expensive to calculate models with spherical symmetry than to calculate plane-parallel models, because one is obliged to use many more angle points in the moment sums. Second, the use of spherical models introduces an additional physical parameter, since T_{eff}

and $\log g$ are replaced by the total luminosity, mass, and radius at the base of the atmosphere. Since the plane-parallel models described here already have three physical parameters, the introduction of a fourth would make it impractical to calculate a grid of models, and one would be forced to make calculations for individual stars.

In any case, an analysis using spherical geometry would need starting estimates of the stellar parameters, and these would be best provided by the plane-parallel analysis described here.

It cannot be denied that atmospheric extension and stellar winds have important effects. The most obvious effect is that opaque transitions will arise from a larger radiating area, resulting in greater flux in the line. This is generally recognized as the mechanism whereby the prominent emission lines of supergiant stars, including the Of stars, originate. In particular, it will be seen later that the HeII 4686Å line is much weaker in most of the observed spectra than in the theoretical spectra calculated here.

The other important effect of atmospheric extension is the *geometric dilution* of the radiation field in the outer regions of the atmosphere. This has the effect of reducing the source function in strong lines. In particular, as was first pointed out by Ghobros [18], the population of the 2^3S level of neutral helium is most strongly overpopulated relative to a non-LTE plane-parallel model. Voels *et al.* [47] find that the the 6678Å, 5876Å, and 4471Å lines are all suspect. It will be seen later that the 5876Å line reproduces poorly the observational material presented here, but that the fits to the other two lines seem satisfactory (except for the 6678Å line at higher T_{eff}).

The effects of neglecting winds are difficult to estimate at present. Although rapid progress is being made in the theory of radiation-driven stellar winds, practical computations of stellar spectra (such as, e.g., Voels *et al.* [47]) use an assumed value for the mass-loss rate. This, in turn, must be estimated from observations of P Cygni profiles of ultraviolet resonance lines that are not available for the bulk of the stars observed here. Two of the five stars studied in detail by the Colorado group [12], [47] show significant effects from wind blanketing: ζ Pup, spectral type O4f, and α Cam, spectral type O9.5 Ia. For the other objects (ζ Ori A, spectral type O9.7 Ib; δ Ori, spectral type O9.5 II; and AE Aur, spectral type O9.5 V), the effects of wind blanketing were found to be negligible. Thus, for the most luminous supergiant and Of stars, the neglect of wind blanketing may be serious, while for less luminous objects, the effects are not important.

The second approximation, the neglect of metallic line blanketing, is made for the same reasons as the approximation of plane-parallel geometry. Non-LTE metallic line blanketing greatly increases the computational cost of a model and introduces at least one additional free parameter, the metallicity. The main physical consequences of metallic line blanketing in non-LTE models are a slight reduction in the effects of departures from LTE (because the mean thermalization depth in ionization continua is reduced) and heating of the atmosphere through backscattering. Since backscattering by stellar winds has a similar effect, the neglect of metallic line blanketing may be no worse an approximation than the neglect of spherical geometry.

Werner [48] has calculated plane-parallel non-LTE model atmospheres that treat detailed line blanketing by hydrogen, helium, and carbon. Although his models cover a somewhat different temperature range than those described here (60000K to 100000K), his general conclusions may be relevant. He finds that the optical lines of hydrogen and helium are only marginally affected, the changes in the line profiles being limited to the line cores. The models with line blanketing show shallower line cores for the combined Hβ/HeII and Hγ/HeII lines and the HeII 4686Å line and deeper line cores for all other lines. The effects, however, are not great.

The atomic data used are somewhat uncertain. Although the radiative rates are generally reliable (those for hydrogen and ionized helium being exact, and those for neutral helium being excellent approximations), the collision rates are uncertain even for the hydrogenic atoms because of the difficulty of the theoretical calculation. Those for the lowest levels ($n \leq 3$ for hydrogen and neutral helium and $n \leq 5$ for ionized helium) are from fairly sophisticated theoretical calculations and/or laboratory measurements. Those for higher levels are much more uncertain. However, since the collision rates are important mainly for determining the thermalization depth of line transitions, the uncertainty in these rates is unimportant when the thermalization of the corresponding lines is dominated by the overlapping continuum absorption (for which the cross sections are reliable). This is certainly the case for infrared transitions, since the continuum absorption increases rapidly towards lower frequencies.

3.2.2 The Gray Atmosphere Calculation

The gray atmosphere program, GRAY, is a highly modified version of the MAH code written for the same purpose. The main differences are in the formats of the input and output files and in the neutral helium bound-free cross sections used. The file format was developed with the intention that it be as general as possible, so that models calculated with different approximations, e.g., LTE *vs.* non-LTE, could be represented in the same format (Table 4).

The neutral helium bound-free cross sections for the lowest states are represented as Kramers' cross sections modified by Gaunt factors. The Gaunt factors were approximated by fitting rational functions to values calculated from the tables of Stewart [44], Jacobs [26], and Steward and Webb [45]. The resulting approximations are quite good close to the absorption edge, but are less accurate at higher frequencies where autoionization levels give a complex structure to the cross section. Since the lowest autoionizing state has a rather high excitation (55.4 eV), no autoionizing states have been included in the neutral helium model atom, and the structure these levels give to the photoionization cross sections are probably of little importance. The relative contribution of the radiation field at such high frequencies to the bound-free rates for neutral helium is small.

The program GRAY optionally includes negative hydrogen ion opacities [17], [19], although these were not used for the O star models described here (since they are negligible at temperatures characteristic of O star atmospheres).

3.2.3 The LTE Calculation

The gray model atmospheres are used as starting approximations for an LTE atmosphere code (LTE), which uses the highly efficient Rybicki method described above [34], [41]. This LTE calculation improved the convergence of the subsequent non-LTE calculation by providing a first

Table 4. Format of Machine-Readable Model Atmospheres

Line		FORTRAN Format	Explanation
1		(A80)	Model title–not used by programs
2		(2F9.0, F8.2, F8.8, F8.5, 4I5)	T_{eff}, T_{line}, Log g, He/H, Z, control switches
3-15		(8F10.6)	Log abundance (relative to H=12.0) for 92 elements.
16		(2I5)	Number of depth and frequency points
17+		(5E15.7)	Frequency grid
18+		(2I5)	Frequency quadrature rule and grid point: first value is 2 for trapezoid rule and 3 for Simpson's rule, second value gives the grid point for which the rule applies. The final line has zero for both values.
19		(2I5)	First value flags use of negative hydrogen ion opacities; second flags inclusions of lines
20+		(I5)	Atomic number of an element whose model and occupation numbers are given by the succeeding lines. The final line has zero as the value.
	21+	(16I5)	Gives number of levels and lines for each ion of the particular element being described, starting with the negative ion.
	22+	(5E15.7)	Ionization frequency of each level starting with those of the negative ion; a new line is started for each ion.
	23+	(2I5)	Lower and upper state of each line transition
	24+	(5E15.7)	Occupation numbers of all levels of all ions at each depth in the atmosphere, listed in that order without any breaks.

approximation with an accurate temperature and electron-density stratification at large optical depths. The continuum opacities used are identical with those in the gray model calculation, and line opacities have been added to the calculation. The line profiles used are depth-dependent Doppler profiles. It is assumed that microturbulence is negligible.

3.2.4 The Non-LTE Calculations

The MAH non-LTE model atmosphere code was modified to use the Anderson algorithm and to treat many more transitions and continua. Whereas the original calculations were made with model helium ions with only two levels and with six hydrogen line transitions (and none of helium), the modified code handles ten hydrogen levels, all neutral helium levels with a principal quantum number of five or less, and fifteen ionized helium levels. In addition to continuum radiative transition rates, the program handles ten hydrogen line transitions and fourteen neutral helium line transitions (all those connecting levels with principal quantum number $n \leq 5$) and ten ionized helium transitions (all those connecting levels with principal quantum number $2 \leq n \leq 6$). More transitions are used for neutral helium than hydrogen because of the non-degeneracy of the nl levels of the two spin systems. The code includes an automatic scheme for allocating frequency blocks; the criteria described by Anderson [6] are used to determine the frequency block assignment. Since the resonance lines of ionized helium are extremely opaque [22], they are assumed to be in detailed balance; the lines arising from the first excited level are treated as if they were resonance lines for purposes of determining block assignment.

As a test of the algorithm, a continuum-only model (non-LTE, no lines) was calculated, using complete linearization and compared with one calculated by the Anderson algorithm using an identical frequency grid and identical atomic data. The resulting level populations were identical to within the tolerance of the calculations.

The opacities used in the non-LTE calculations were identical with those used in the gray and LTE calculations. The collisional excitation rates for transitions between the lower states of hydrogen were taken from Aggarwal [3]. Collisional excitation rates for transitions between lower states of neutral helium were taken from Aggarwal *et al.* [4] and Berrington *et al.* [10]. Collision strengths for ionization from lower states of hydrogen, neutral helium, and ionized helium are from Lennon *et al.* [28]. All other collisional rates were identical with those of the original code. These are represented as the product of a Boltzmann term, $\exp\left(\frac{-\chi}{kT}\right)$, a power of T (usually $T^{-1/2}$), and a rational function of T determined by a least-squares fit.

Generally, model convergence was reasonably rapid. However, for a small number of models (roughly 10% of all those calculated), it was necessary to employ a radiative/collisional switching method [23]. The calculations using the switching method were carried out after normal calculations had been attempted for all points in the model grid, and it was later discovered that the switched calculations had inadvertently been made with a slightly modified set of ionization frequencies for the upper states of neutral helium. This did not appear to have had any significant effect on the calculations, which is not surprising, considering that the associated continua are very weak.

With a suitable grid of models, the next step was to refine the individual line spectra for hydrogen and helium by carrying out a more detailed calculation for each spectrum while holding the temperature, electron density, and level populations of the other element fixed. The hydrogen spectrum code replaces the Doppler profiles with more accurate line profiles that include the Stark effect, and allows for the effects of the overlapping ionized helium lines. Ten transitions are included explicitly, and the rates of all other transitions connecting the first ten levels of hydrogen are estimated at the start of the calculation using the equivalent two-level atom approach [34].

A similar calculation is used to refine the helium lines. The line profiles used are Voigt profiles with a damping width that is the sum of the natural damping width and the quadratic Stark damping width [20]. The effects of the overlapping hydrogen lines are taken into account for the ionized helium lines.

For the model at T_{eff} = 30000K, $\log g$ = 3.0, He/H=0.20, the equivalent widths of the HeII 4686Å and HeI 5016Å lines were vanishingly small. In the case of the HeII 4686Å line, this was due to a flux in the line wings that exceeded the continuum flux; the profile itself appeared reasonable. The tiny equivalent width for the HeI 5016Å line reflected the actual line profile. All other lines for this model were consistent with those of the adjacent models in the grid. The reasons for this inconsistency in the two lines is unclear, although it may have something to do with the low gravity of the model (close to the Eddington limit for this effective temperature). The equivalent widths and line profiles of these two lines were interpolated from adjacent models for use in the tables and fits presented here.

3.2.5 Line Synthesis

For the final line synthesis, the Balmer line profiles are calculated using theoretical absorption profiles from the tables of Vidal, Cooper, and Smith [46]. The ionized helium line profiles are taken from the very recent unified-theory calculations of Schöning and Butler [43]. For neutral helium, the Voigt profiles are replaced by unified-theory profiles for the lines with forbidden components (the 4471Å and 4922Å lines) at densities in excess of 10^{13} electrons/cm^2 [8], [9].

3.2.6 Computational Costs

The gray model computation is extremely fast and represents a negligible fraction of the total computation cost. The LTE calculation takes less than a minute of processor time on the CRAY to complete a model calculation. The non-LTE code is somewhat more expensive, taking slightly less than one minute per iteration for a total of up to 12 minutes per model. The code for refining the hydrogen populations takes less than a minute per model, but that for helium is quite expensive (more than 10 minutes per model) and is also expensive in memory usage. In light of the computational cost, the benefits of the refined population codes are probably insufficient to warrant their use for future calculations. They are included here primarily to rule out the possibility that discrepancies between theory and observation arise from too simplistic a model atom.

3.3 RESULTS

Figure 3 gives the profile of the Hβ line for a model with T_{eff} = 35000, $\log g$ = 4.0, and [He/H]=0.10 at each stage of the calculation. The profiles, in order of increasing depth, correspond to the gray model, the LTE model, the non-LTE model, and the non-LTE model with refined treatment of the hydrogen spectrum. It is evident that the effects of departures from LTE are sizable, as was noted two decades ago by Auer and Mihalas [7]. However, the more refined non-LTE calculation has little effect outside the line core.

The temperature structure for the same model is given in Figure 4 , along with the temperature structure of the Mihalas model with the same effective temperature, surface gravity, and helium abundance fraction [33]. It is evident that while the structures are identical at depth, the temperature is significantly higher in the outermost layers of the atmosphere in the new model. Since heating by photoionizations from the ground state of hydrogen largely determines the temperature structure in the upper atmosphere [34], it is likely that the higher temperature results from the additional transitions included in the model hydrogen atom. These permit electrons to cascade freely into the ground state. Although the change in the temperature structure is pronounced, it does not have a large effect on the calculated line profiles, the effects being restricted to the cores of the stronger lines.

3.3.1 Theoretical Equivalent Widths and Profiles

The equivalent widths of the synthesized optical lines are found in Table 5 . Since the equivalent widths of the hydrogen Balmer lines are of limited practical value, complete profiles have been given in Figure 5 for He/H=0.10, the normal cosmic abundance. (The hydrogen lines themselves are not very sensitive to composition, although the overlapping ionized helium lines are.) These are in general agreement with the results of the MAH calculations, the greatest differences being at the highest effective temperatures modeled and for the ionized helium lines.

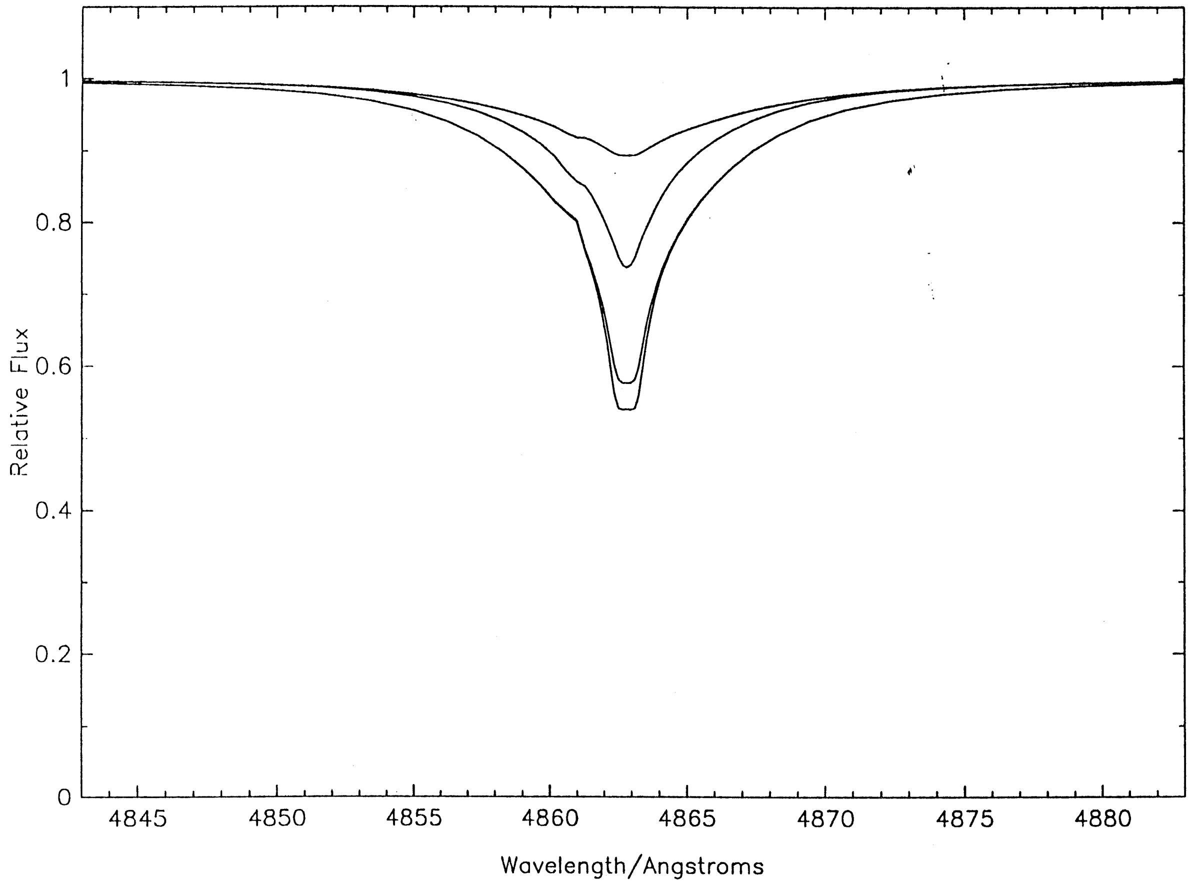

Figure 3. Theoretical profiles of Hβ at different points in the calculation. The profiles are, in order of increasing depth: for a gray model atmosphere; for an LTE model atmosphere; for a non-LTE model atmosphere; and for a non-LTE model atmosphere with improved profiles and approximate treatment of all transitions between states with $n \leq 10$.

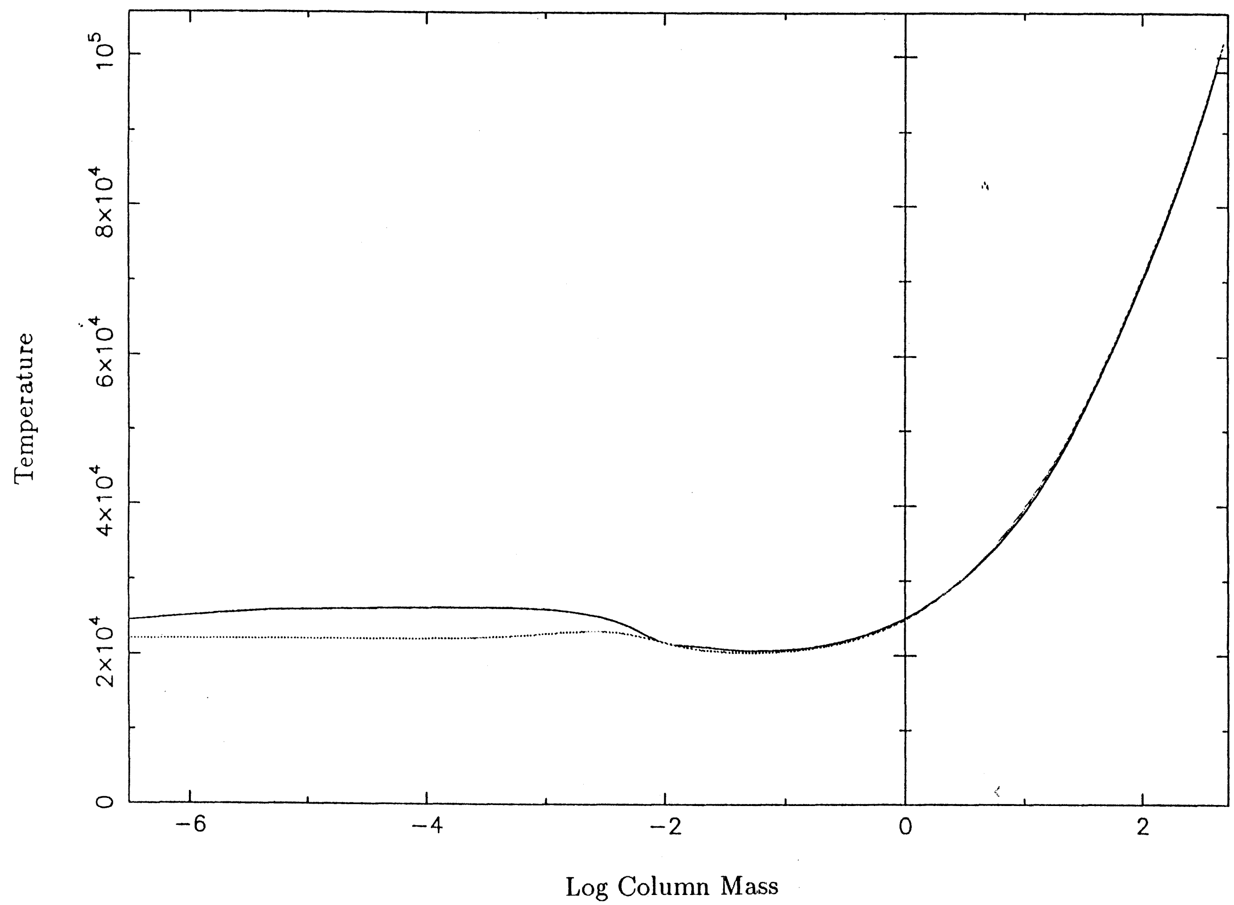

Figure 4. Temperature structure of a typical model atmosphere. The model chosen has T_{eff} = 35000K, $\log g = 4.5$, and He/H=0.10. The bottom curve is the temperature profile for the MAH model with these parameters; the top curve is the temperature profile for the author's model.

Table 5-1. Theoretical Equivalent Widths of Hβ

$\log g$	T_{eff}	He/H=0.05	He/H=0.10	He/H=0.20	He/H=0.50
4.5	30000	4502	4537	4531	4375
	32500	3991	4030	4036	3916
	35000	3604	3623	3700	3644
	37500	3360	3368	3356	3311
	40000	3202	3186	3128	3001
	45000	3022	2986	2909	2760
	50000	2926	2878	2792	2622
4.0	30000	3465	3510	3540	3473
	32500	3157	3203	3236	3187
	35000	2986	3007	3022	3005
	37500	2882	2889	2865	2781
	40000	2823	2811	2765	2643
	45000	2735	2705	2637	2492
	50000	2687	2643	2588	2403
3.5	30000	2561	2610	2667	2686
	32500	2467	2490	2522	2537
	35000	2426	2462	2482	2448
	37500	2382	2348	2417	2357
	40000	2308	2324	2325	2261
3.0	30000	1736	1801	1867	1932
	32500	1614	1730	1869	1970

Table 5-2. Theoretical Equivalent Widths of Hγ

$\log g$	T_{eff}	He/H=0.05	He/H=0.10	He/H=0.20	He/H=0.50
4.5	30000	4061	4083	4066	3923
	32500	3589	3625	3623	3517
	35000	3208	3247	3315	3280
	37500	2955	2961	2949	2915
	40000	2792	2772	2715	2600
	45000	2614	2579	2510	2378
	50000	2519	2476	2403	2263
4.0	30000	3223	3262	3284	3219
	32500	2885	2943	2984	2950
	35000	2664	2687	2713	2732
	37500	2527	2524	2500	2436
	40000	2438	2424	2387	2294
	45000	2325	2305	2263	2161
	50000	2247	2224	2184	2075
3.5	30000	2356	2420	2487	2520
	32500	2185	2222	2279	2340
	35000	2088	2118	2138	2144
	37500	2006	2005	2054	2033
	40000	1920	1949	1969	1946
3.0	30000	1470	1543	1635	1758
	32500	1331	1461	1589	1725

Table 5-3. Theoretical Equivalent Widths of HeII 5413Å

$\log g$	T_{eff}	He/H=0.05	He/H=0.10	He/H=0.20	He/H=0.50
4.5	30000	9	14	22	38
	32500	62	92	125	173
	35000	208	298	393	491
	37500	418	574	742	910
	40000	607	771	943	1128
	45000	780	939	1082	1239
	50000	813	973	1105	1222
4.0	30000	30	43	58	82
	32500	130	175	223	275
	35000	298	413	536	639
	37500	506	647	801	975
	40000	631	780	944	1081
	45000	732	900	1040	1189
	50000	739	925	816	1207
3.5	30000	83	108	131	156
	32500	216	280	351	413
	35000	385	512	665	797
	37500	531	568	831	984
	40000	597	737	906	1048
3.0	30000	164	218	255	278
	32500	332	444	528	684

Table 5-4. Theoretical Equivalent Widths of HeII 4543Å

$\log g$	T_{eff}	He/H=0.05	He/H=0.10	He/H=0.20	He/H=0.50
4.5	30000	10	16	26	44
	32500	60	96	139	201
	35000	187	302	434	593
	37500	365	556	794	1099
	40000	544	783	1052	1387
	45000	731	982	1230	1516
	50000	738	987	1230	1497
4.0	30000	32	48	68	99
	32500	129	191	260	341
	35000	279	411	569	757
	37500	440	623	835	1107
	40000	559	757	970	1231
	45000	627	831	1036	1276
	50000	605	803	1010	1241
3.5	30000	91	127	163	201
	32500	205	290	387	496
	35000	328	460	623	812
	37500	423	527	744	957
	40000	458	609	778	991
3.0	30000	139	197	257	324
	32500	228	308	418	571

Table 5-5. Theoretical Equivalent Widths of HeII 4686Å

$\log g$	T_{eff}	He/H=0.05	He/H=0.10	He/H=0.20	He/H=0.50
4.5	30000	50	73	101	149
	32500	163	220	283	381
	35000	404	477	605	720
	37500	585	700	834	990
	40000	702	830	9792	1150
	45000	815	947	1085	1208
	50000	846	976	1099	1228
4.0	30000	76	103	138	189
	32500	246	309	379	463
	35000	435	519	611	722
	37500	549	649	774	9267
	40000	616	728	859	1017
	45000	675	784	906	1045
	50000	698	804	1050	1035
3.5	30000	144	184	227	281
	32500	305	363	423	494
	35000	400	472	580	689
	37500	434	507	639	777
	40000	441	520	628	779
3.0	30000	177	228	272	321
	32500	121	189	289	444

Table 5-6. Theoretical Equivalent Widths of HeI 6678Å

$\log g$	T_{eff}	He/H=0.05	He/H=0.10	He/H=0.20	He/H=0.50
4.5	30000	507	569	696	909
	32500	519	576	705	910
	35000	499	570	753	946
	37500	417	493	647	874
	40000	313	362	490	675
	45000	49	18	110	220
	50000	-201	-328	-265	-188
4.0	30000	497	556	659	840
	32500	509	579	686	868
	35000	463	547	687	881
	37500	369	449	581	771
	40000	240	312	403	544
	45000	-87	-64	21	126
	50000	-207	-359	-342	-276
3.5	30000	504	568	652	795
	32500	471	564	682	832
	35000	382	479	643	793
	37500	235	364	400	576
	40000	228	131	252	353
3.0	30000	431	532	634	777
	32500	237	366	506	721

Table 5-7. Theoretical Equivalent Widths of HeI 5876Å

$\log g$	T_{eff}	He/H=0.05	He/H=0.10	He/H=0.20	He/H=0.50
4.5	30000	538	619	749	978
	32500	566	635	763	976
	35000	615	670	812	1000
	37500	642	719	849	1044
	40000	632	700	816	982
	45000	531	573	653	758
	50000	382	424	493	574
4.0	30000	500	568	663	844
	32500	559	618	709	872
	35000	618	688	787	931
	37500	636	705	807	962
	40000	611	665	748	863
	45000	479	525	590	665
	50000	282	353	427	504
3.5	30000	509	561	619	737
	32500	589	647	701	796
	35000	627	692	760	888
	37500	608	654	725	821
	40000	536	594	657	728
3.0	30000	566	597	635	665
	32500	561	634	687	776

Table 5-8. Theoretical Equivalent Widths of HeI 4922Å

$\log g$	T_{eff}	He/H=0.05	He/H=0.10	He/H=0.20	He/H=0.50
4.5	30000	863	783	1038	1439
	32500	888	724	964	1341
	35000	804	659	926	1085
	37500	474	555	762	926
	40000	324	427	588	814
	45000	108	152	251	382
	50000	-32	-43	5	84
4.0	30000	724	621	821	1154
	32500	709	588	783	1107
	35000	667	522	711	1017
	37500	503	415	567	801
	40000	212	279	396	557
	45000	-9	20	94	191
	50000	-40	-73	-79	-45
3.5	30000	390	497	650	913
	32500	352	455	610	865
	35000	281	367	511	716
	37500	172	224	353	481
	40000	26	97	181	290
3.0	30000	280	372	478	680
	32500	184	251	359	543

Table 5-9. Theoretical Equivalent Widths of HeI 4471Å

$\log g$	T_{eff}	He/H=0.05	He/H=0.10	He/H=0.20	He/H=0.50
4.5	30000	918	1190	1540	2088
	32500	844	1091	1419	1927
	35000	785	1012	1334	1813
	37500	719	925	1191	1601
	40000	626	798	1010	1308
	45000	394	497	635	822
	50000	193	261	352	477
4.0	30000	689	905	1181	1641
	32500	657	860	1121	1548
	35000	620	809	1051	1439
	37500	550	710	913	1220
	40000	450	570	723	946
	45000	244	312	399	524
	50000	81	131	195	279
3.5	30000	517	680	885	1247
	32500	500	653	845	1176
	35000	450	578	731	1036
	37500	369	420	560	765
	40000	270	339	418	541
3.0	30000	376	452	584	838
	32500	325	400	491	701

Table 5-10. Theoretical Equivalent Widths of HeI 5016Å

log g	T_{eff}	He/H=0.05	He/H=0.10	He/H=0.20	He/H=0.50
4.5	30000	213	251	317	433
	32500	204	236	296	401
	35000	197	228	297	389
	37500	182	214	266	342
	40000	160	189	232	289
	45000	82	107	145	187
	50000	13	21	49	88
4.0	30000	193	222	269	355
	32500	194	221	264	345
	35000	193	223	265	327
	37500	179	212	255	308
	40000	146	183	220	265
	45000	44	73	115	163
	50000	4	5	18	49
3.5	30000	184	207	238	296
	32500	193	220	249	292
	35000	187	224	271	303
	37500	151	185	233	280
	40000	81	131	182	231
3.0	30000	181	207	219	249
	32500	149	198	241	282

Table 5-11. Theoretical Equivalent Widths of HeI 4713Å

$\log g$	T_{eff}	He/H=0.05	He/H=0.10	He/H=0.20	He/H=0.50
4.5	30000	188	234	298	414
	32500	181	224	284	389
	35000	178	220	279	371
	37500	171	213	266	344
	40000	154	194	240	298
	45000	88	121	160	203
	50000	30	48	75	113
4.0	30000	167	205	251	338
	32500	172	209	254	330
	35000	177	218	262	324
	37500	169	212	257	312
	40000	140	182	224	271
	45000	53	83	122	169
	50000	11	20	37	67
3.5	30000	161	197	229	287
	32500	178	218	249	292
	35000	177	225	263	314
	37500	141	172	234	287
	40000	74	122	174	231
3.0	30000	175	211	242	261
	32500	136	208	264	316

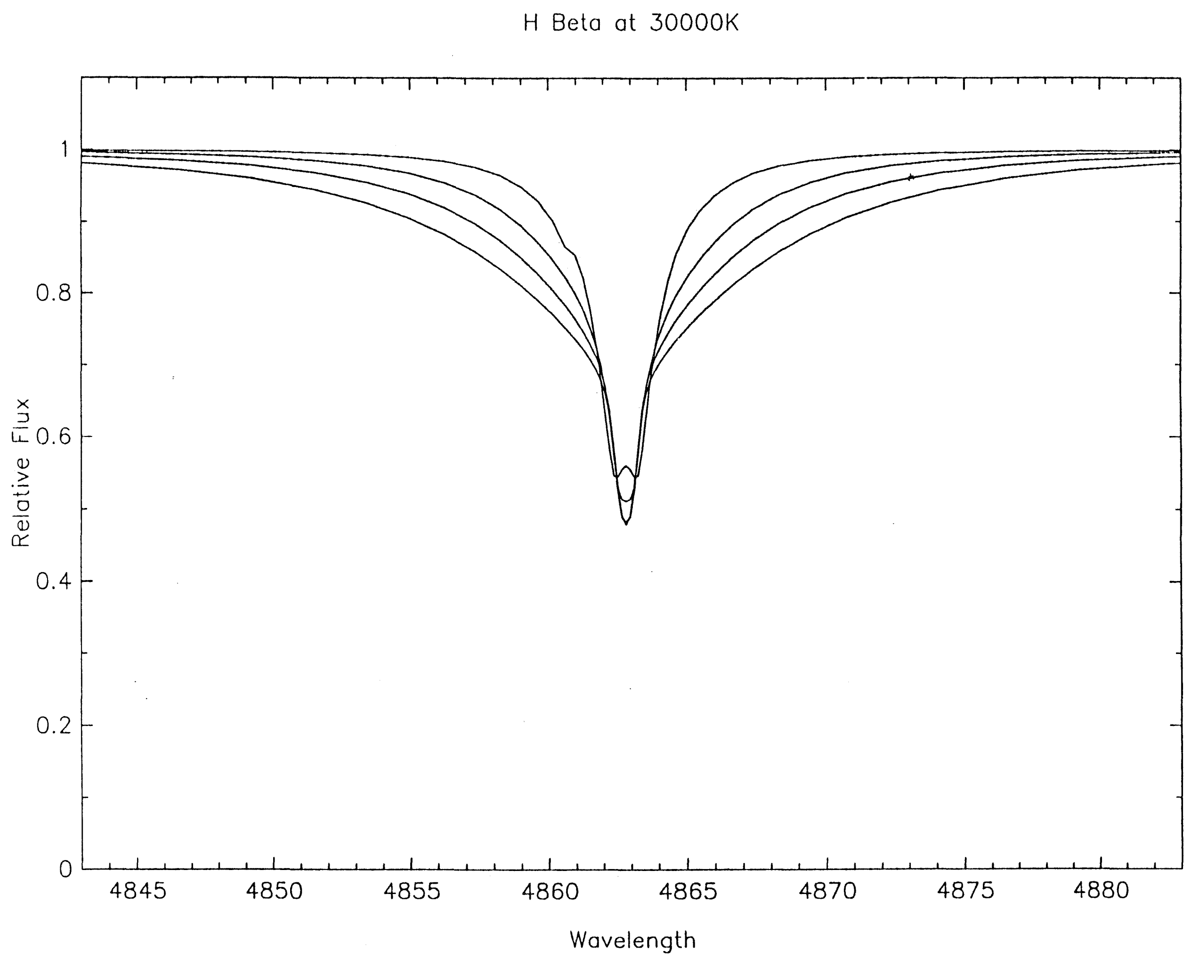

Figure 5. Theoretical hydrogen Balmer profiles. The profiles illustrated are for the composition He/H=0.10, the normal cosmic abundance.

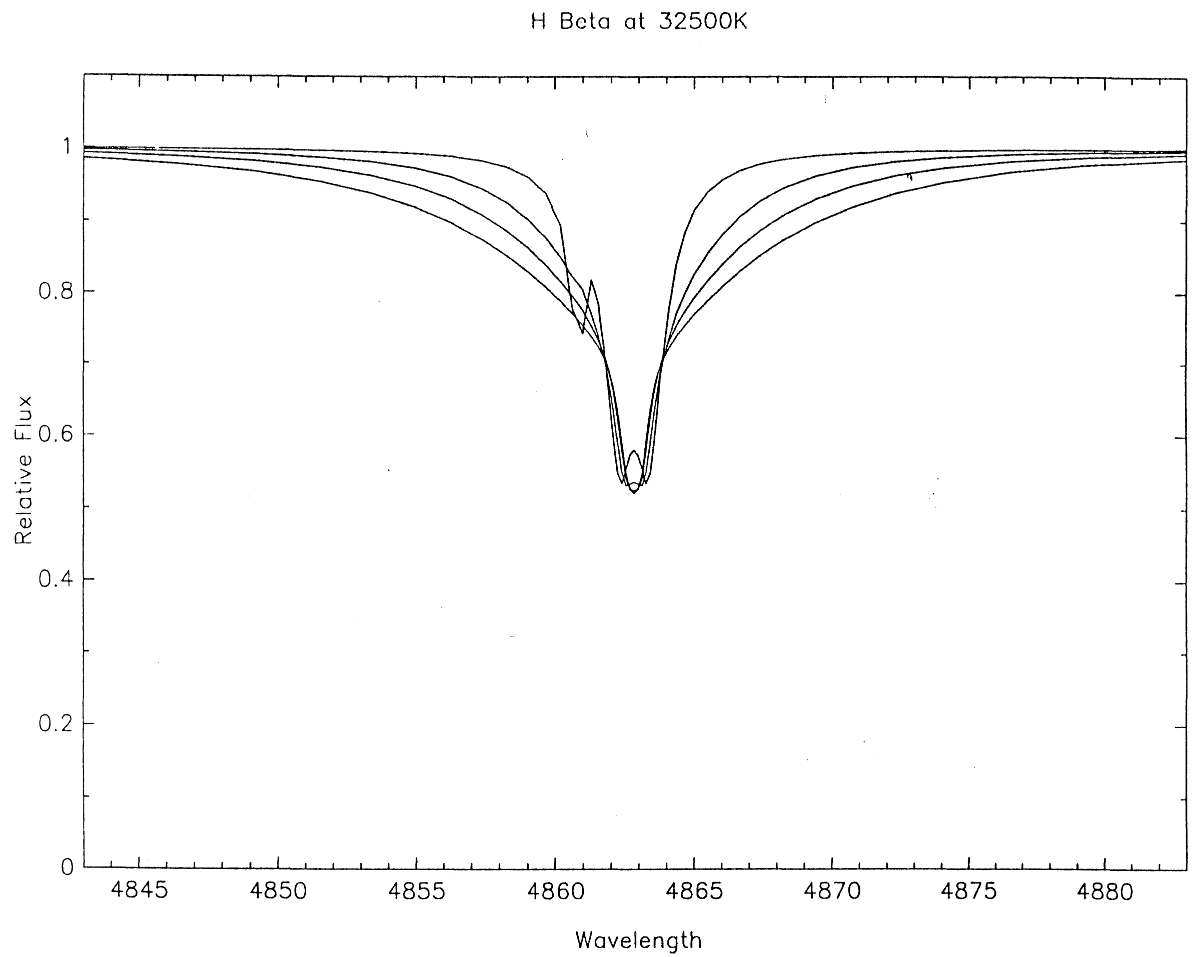

Figure 5. (Continued)

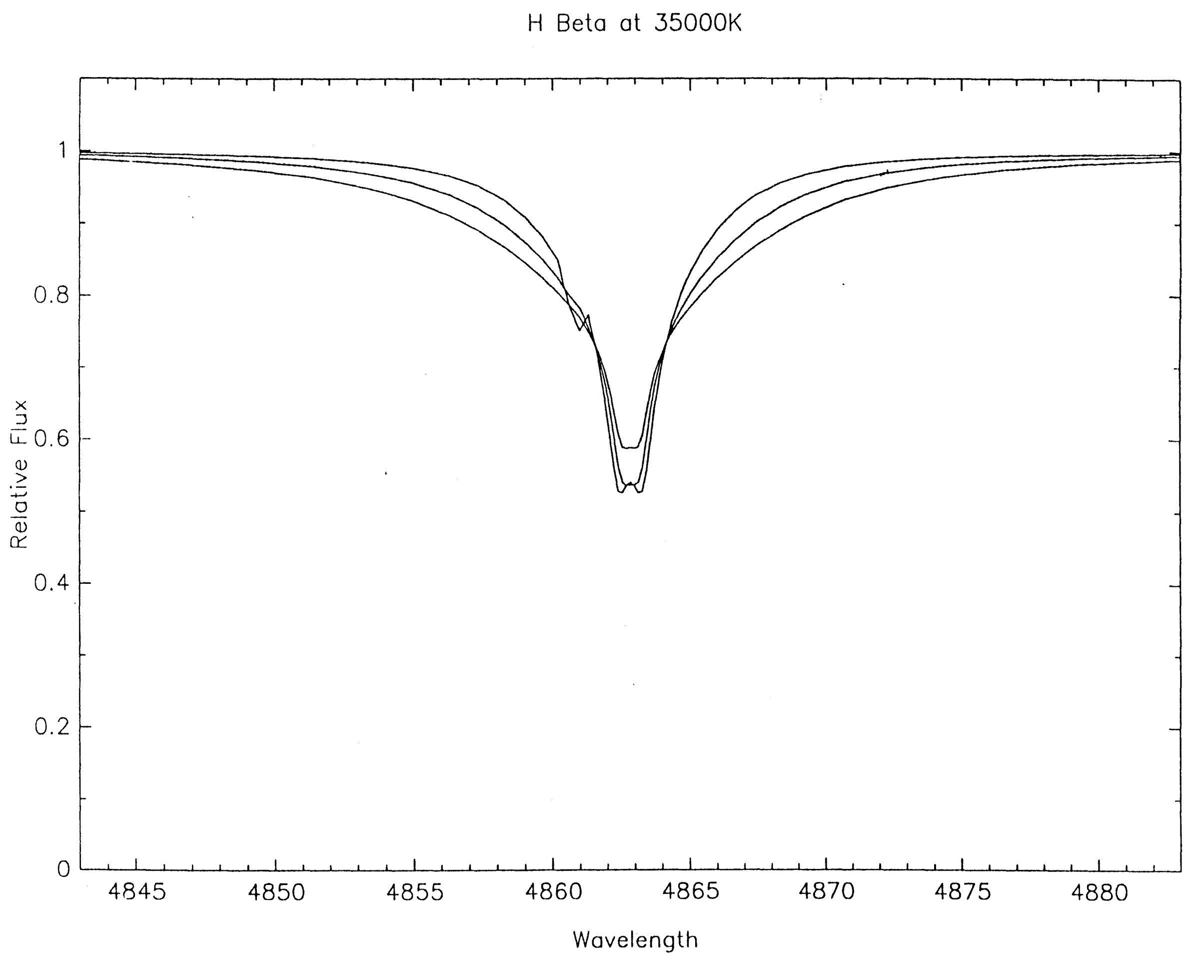

Figure 5. (Continued)

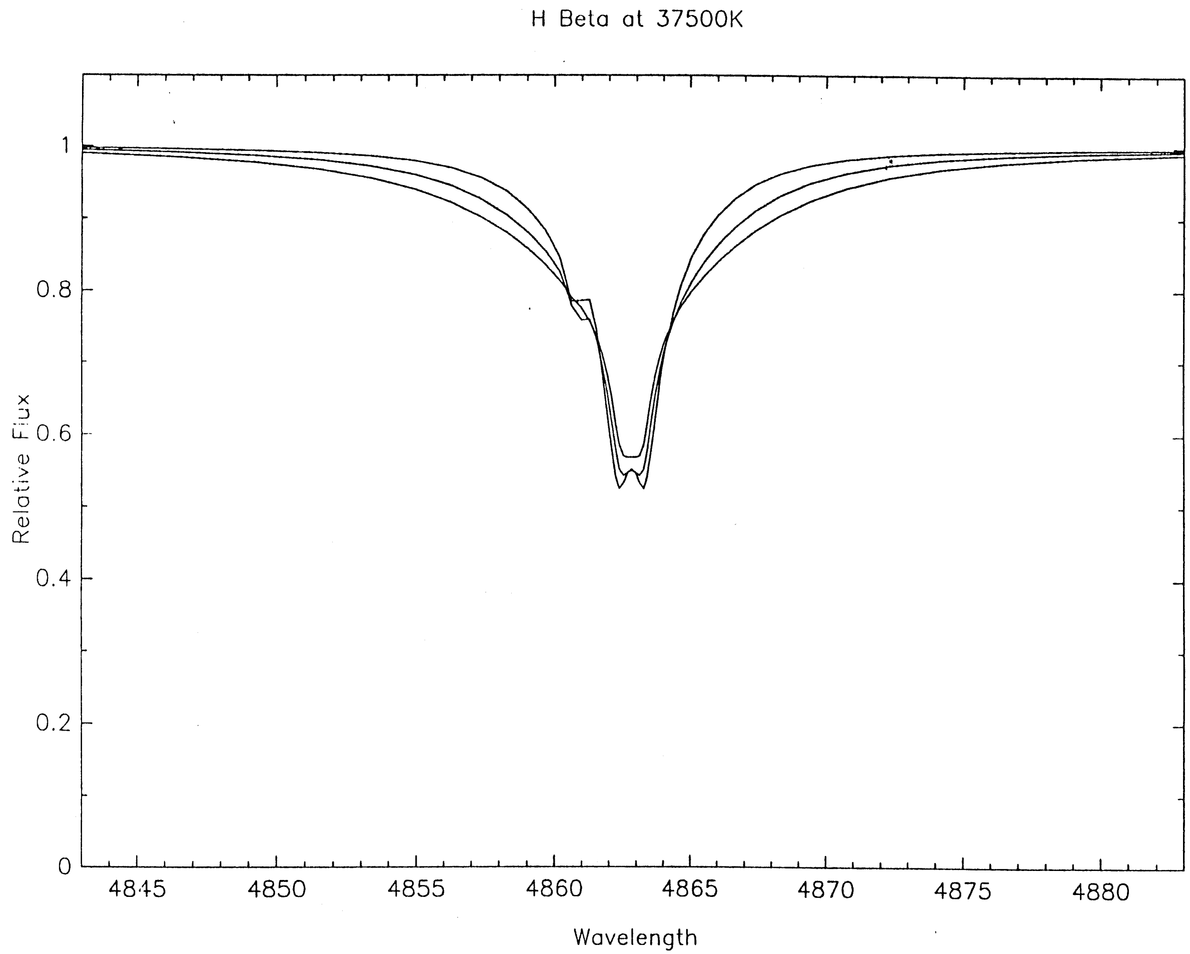

Figure 5. (Continued)

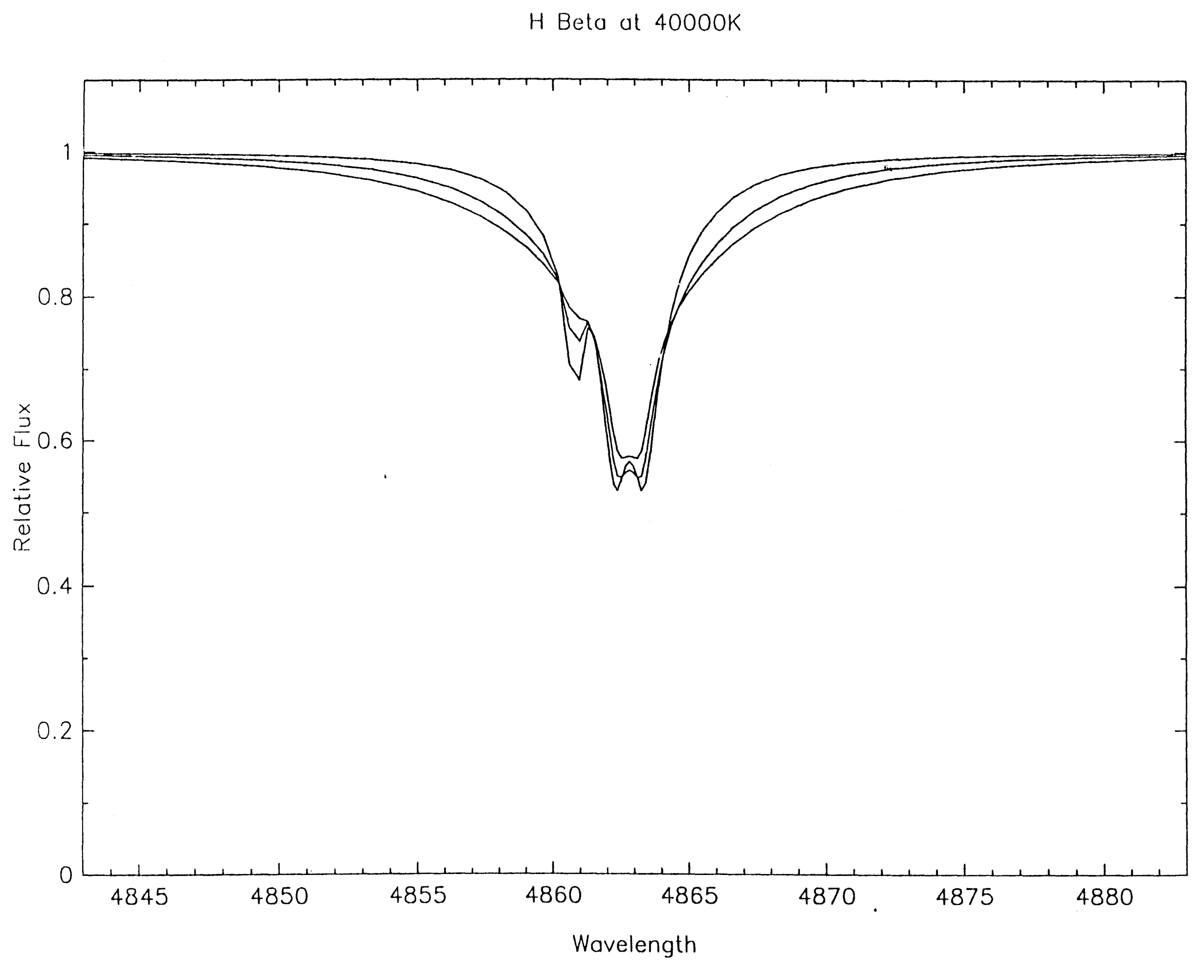

Figure 5. (Continued)

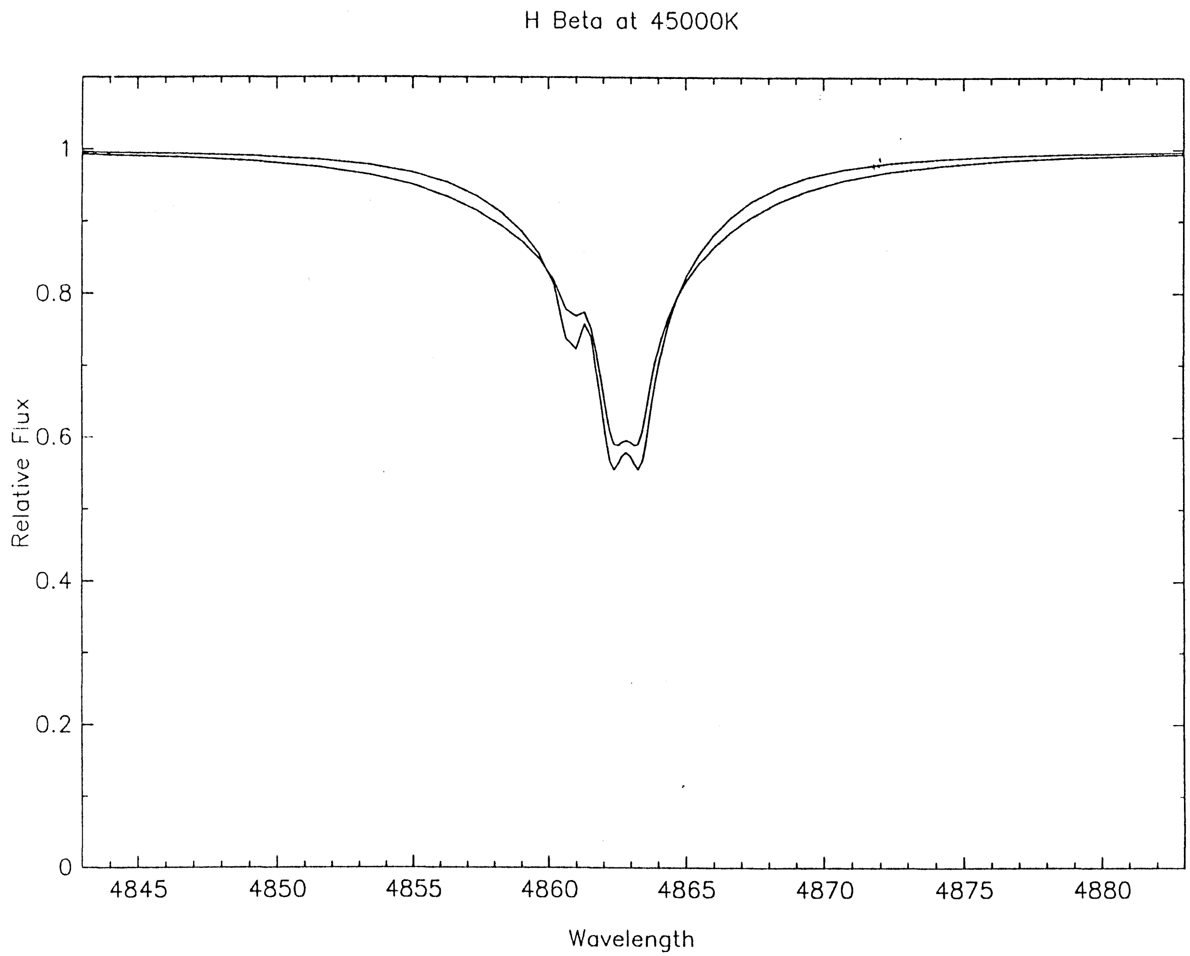

Figure 5. (Continued)

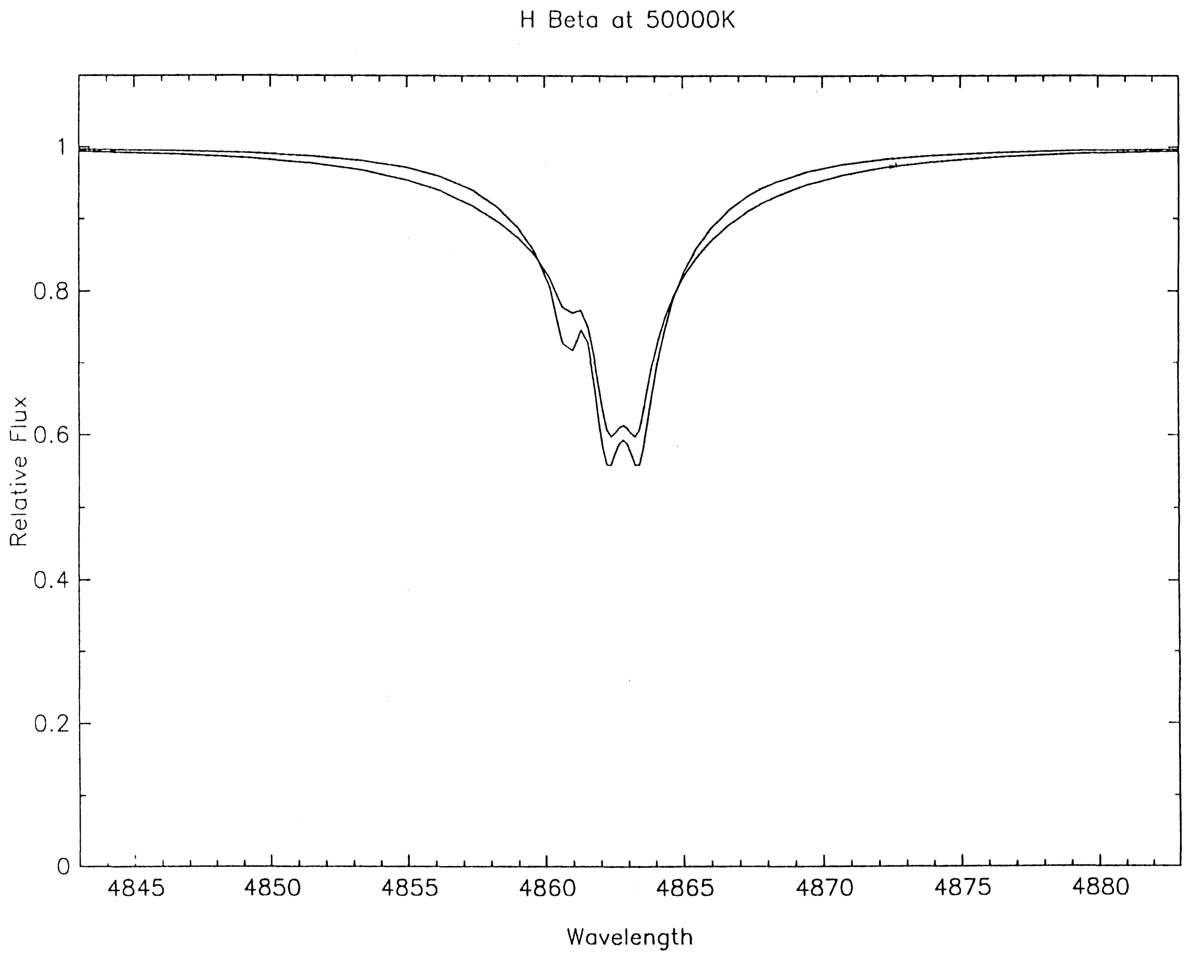

Figure 5. (Continued)

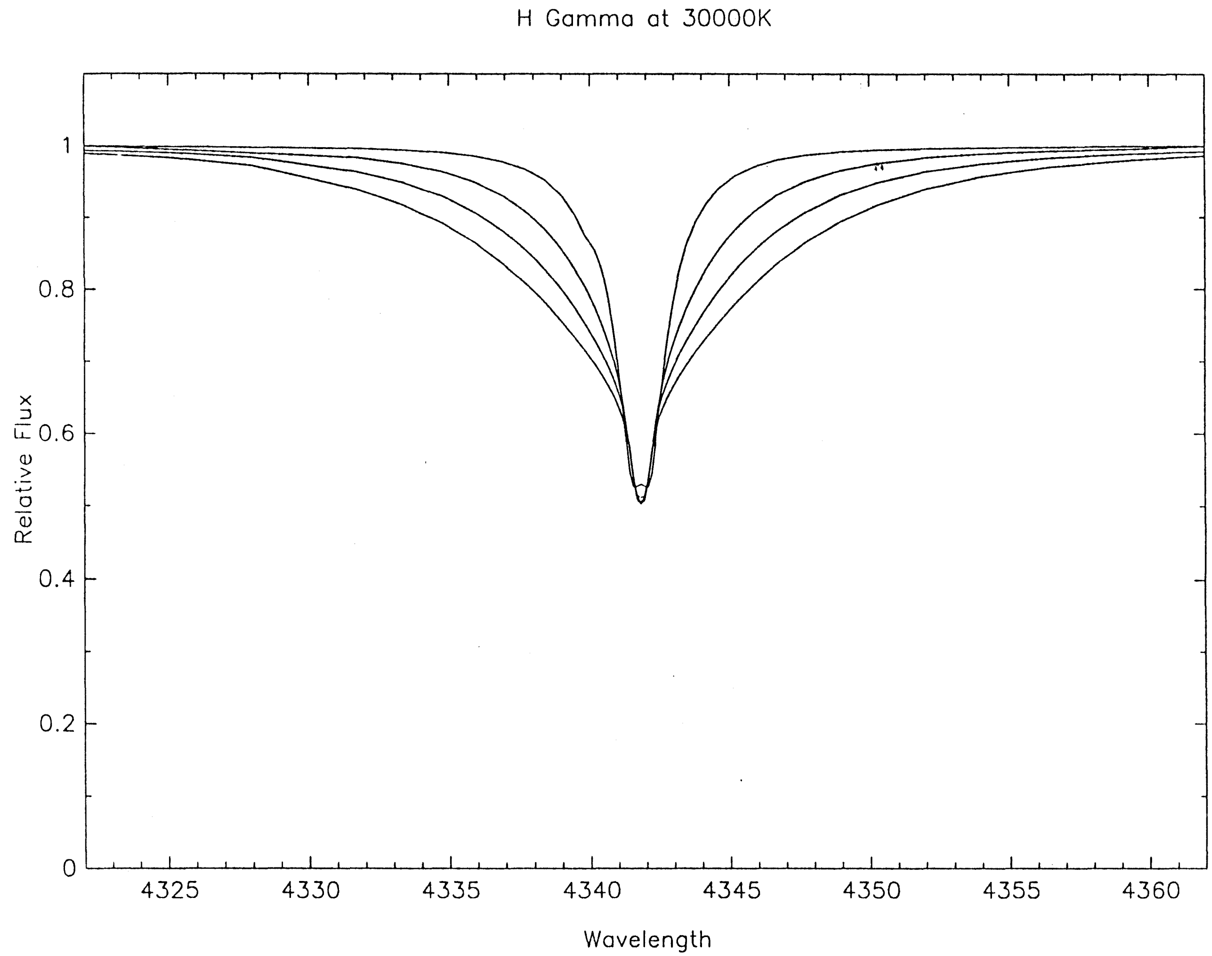

Figure 5. (Continued)

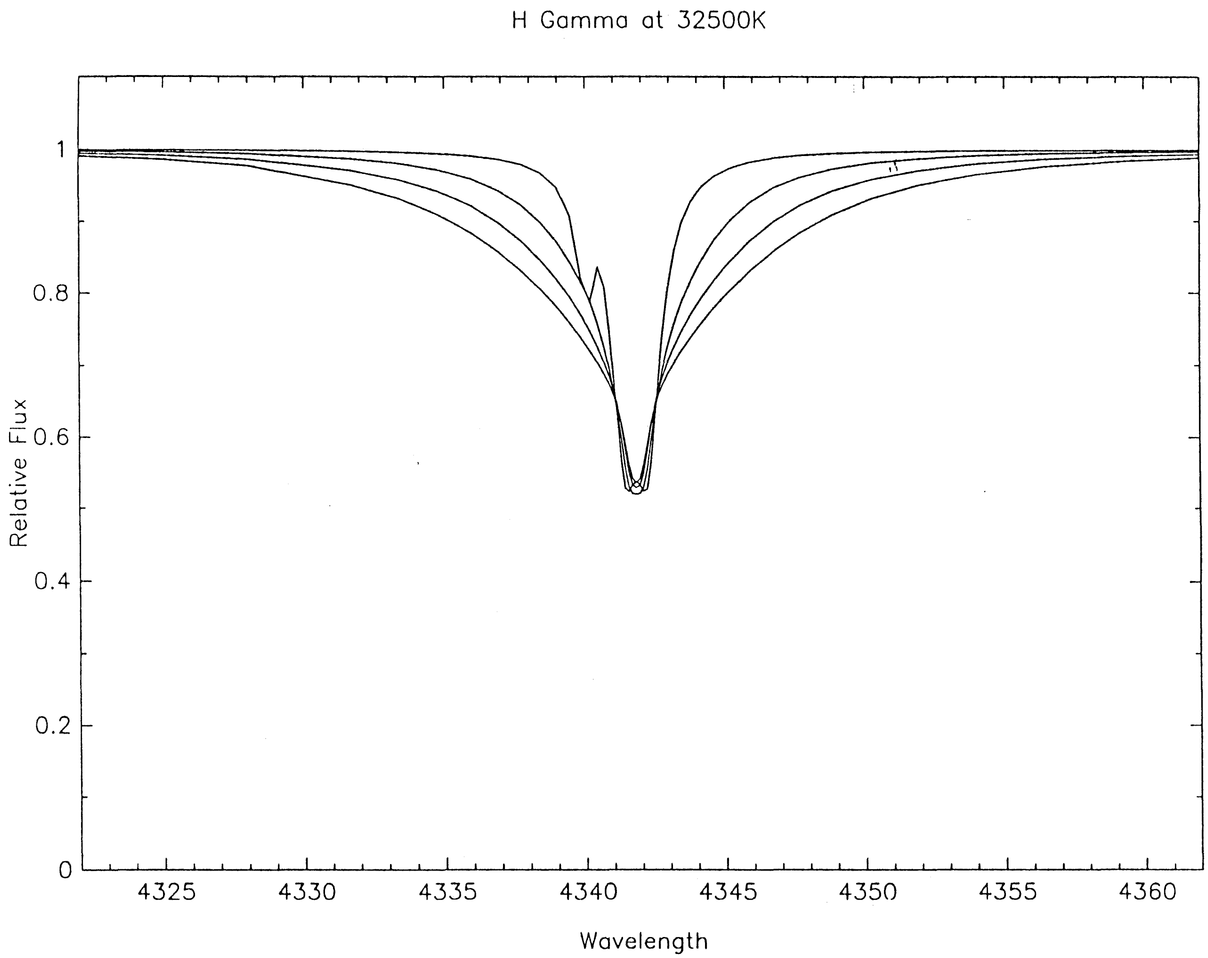

Figure 5. (Continued)

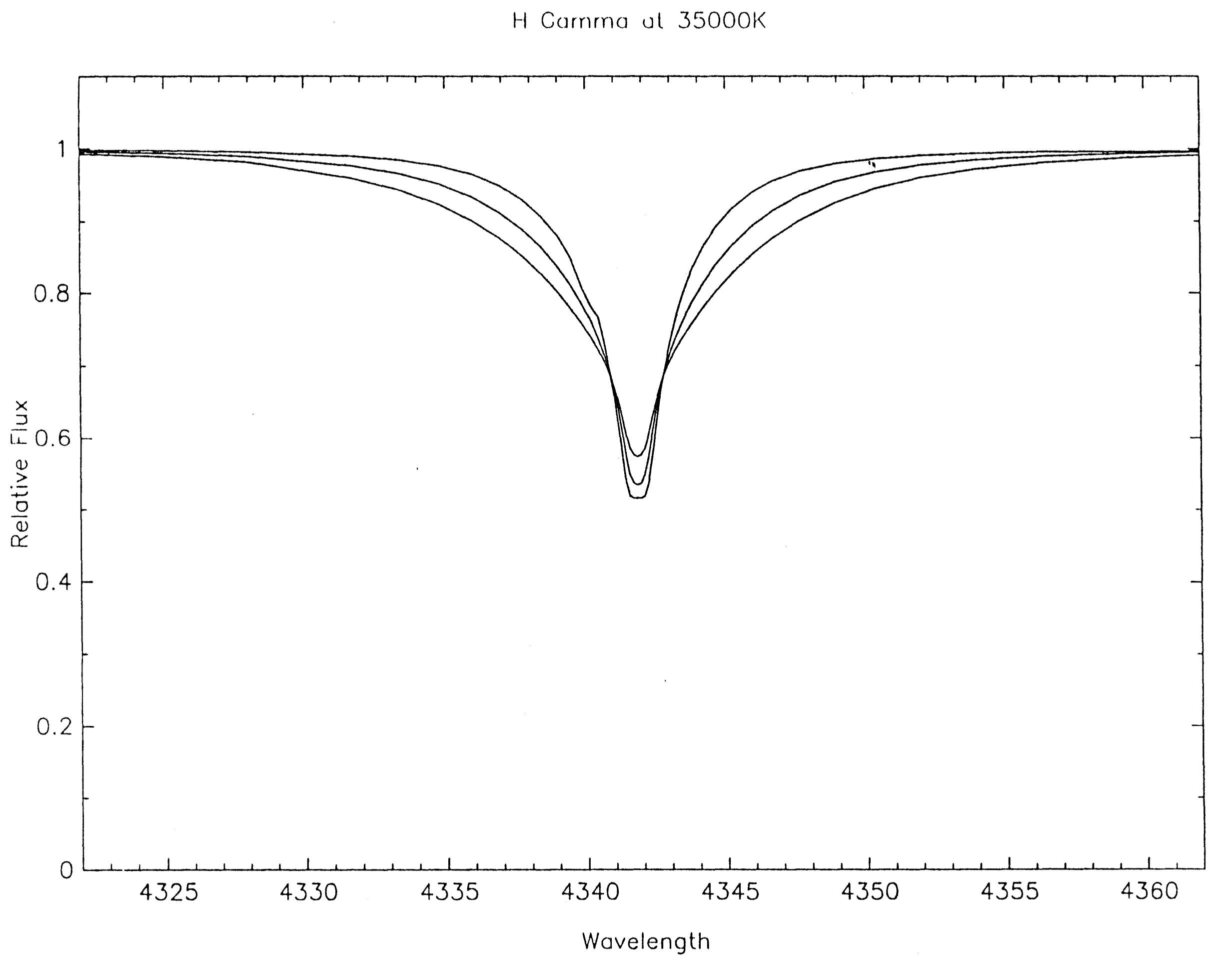

Figure 5. (Continued)

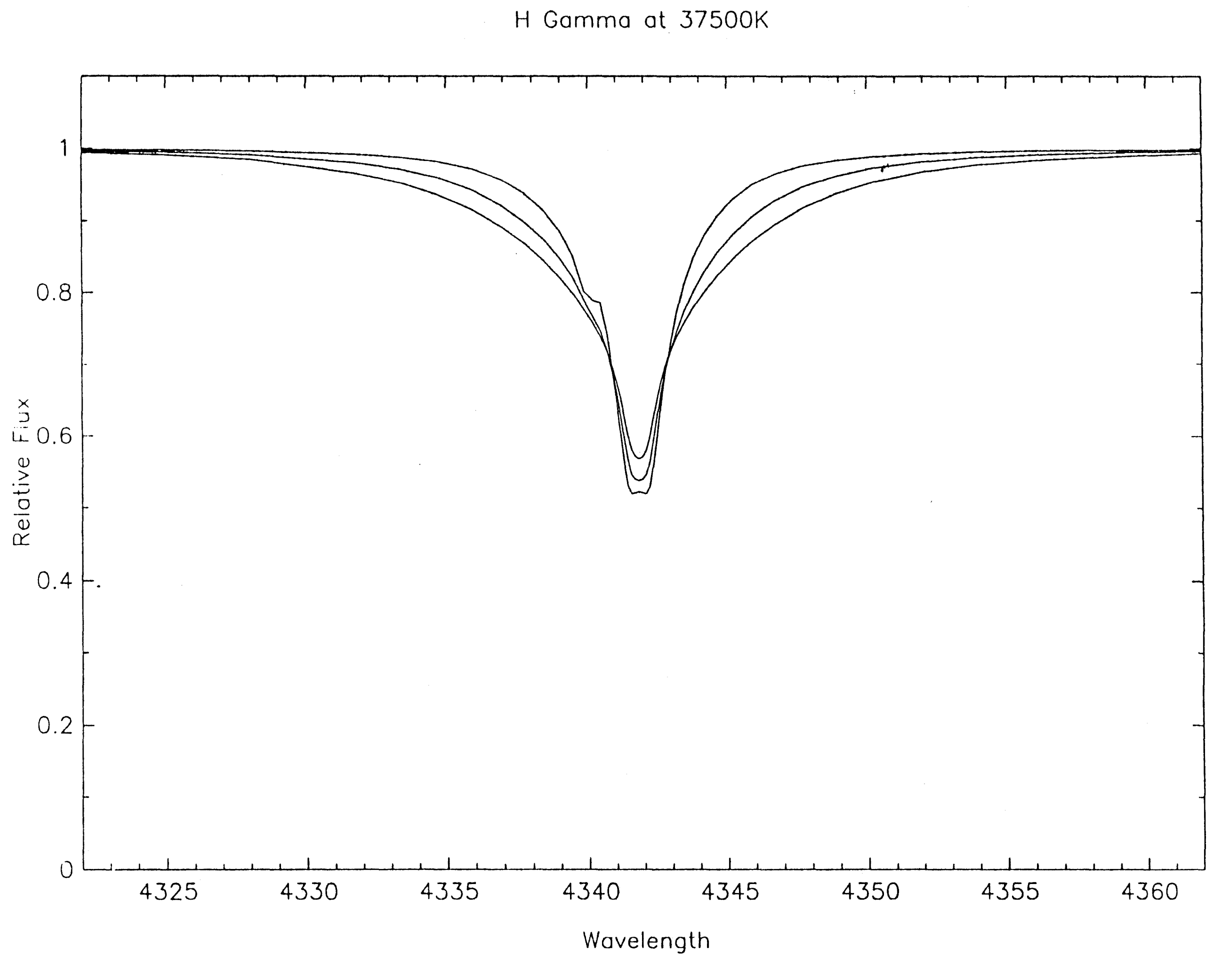

Figure 5. (Continued)

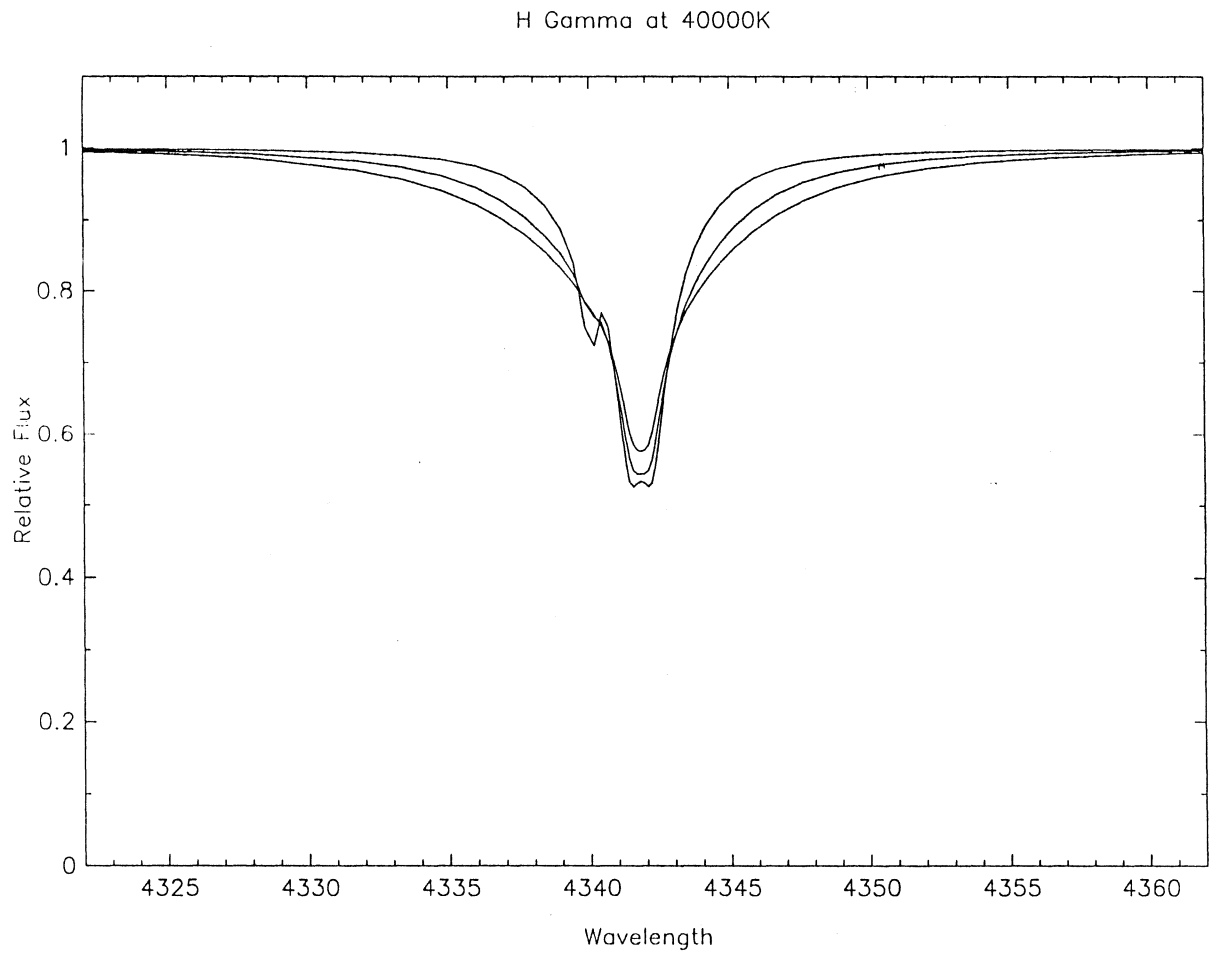

Figure 5. (Continued)

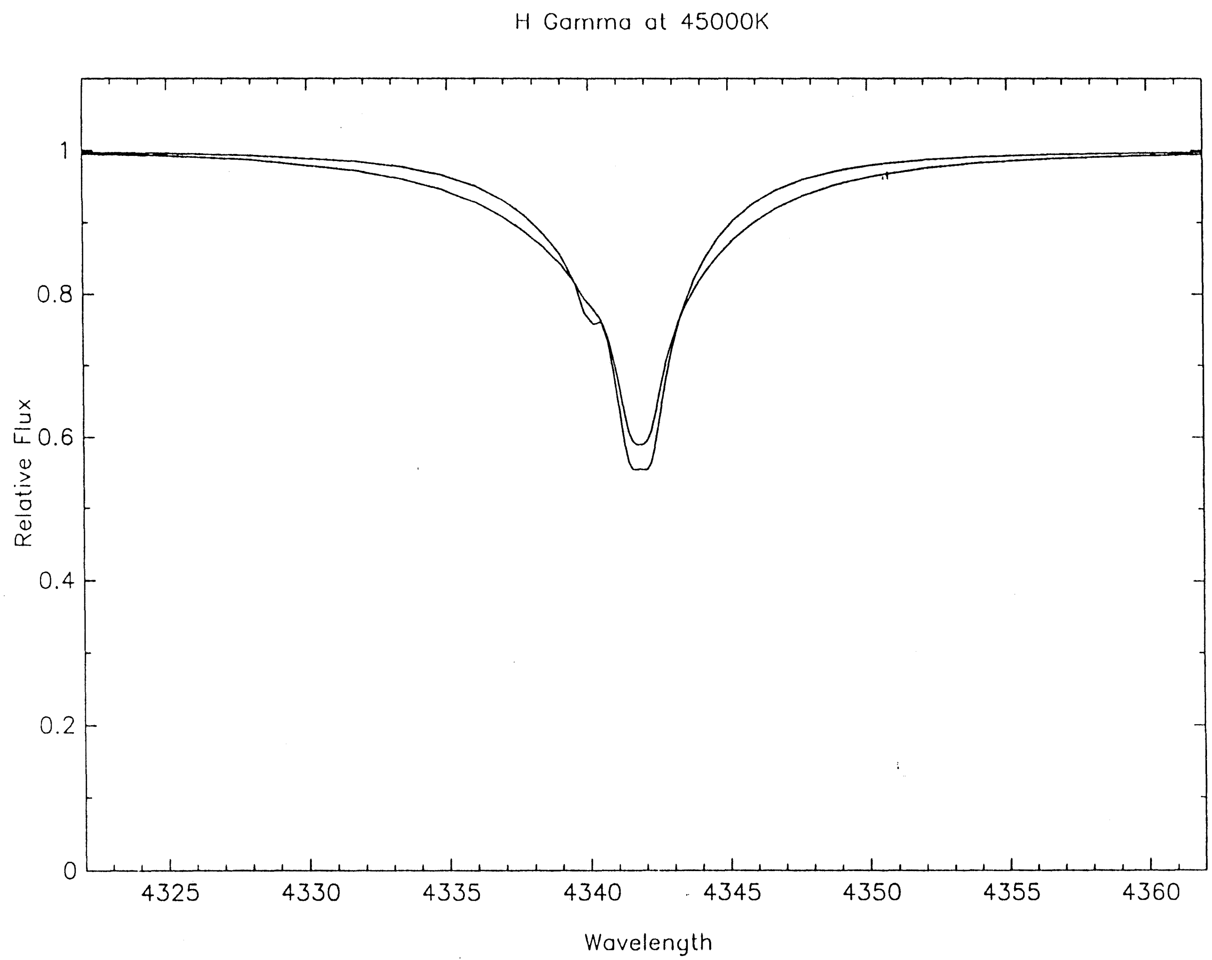

Figure 5. (Continued)

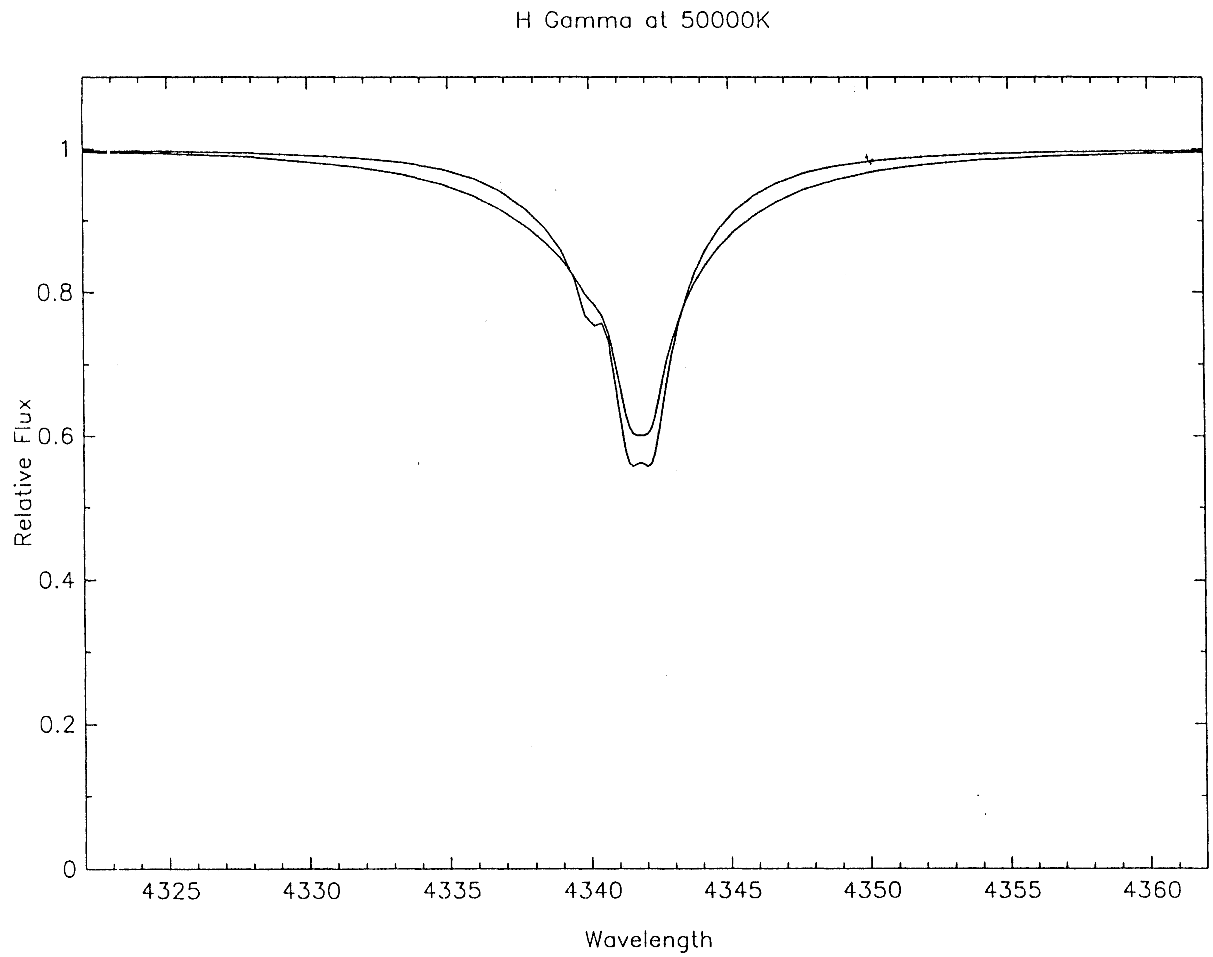

Figure 5. (Continued)

3.3.2 Bolometric Corrections

A very useful quantity is the bolometric correction, which, for stars of known T_{eff} and $\log g$, permits one to convert the visual magnitude M_v to the bolometric magnitude $M_{\rm bol}$.

Code *et al.* [15] give the following formula for calculating the bolometric correction:

$$\mathrm{B.C.} = 2.5 \log \left(\frac{\int_0^\infty f_\lambda S_V(\lambda) d\lambda}{\int_0^\infty f_\lambda d\lambda} \right) + 0.958, \tag{11}$$

where the zero point constant 0.958 has been determined empirically. The sensitivity function S_V is given by Matthews and Sandage [31].

Table 6 gives the bolometric corrections obtained using this formula for the models calculated here. One finds that the bolometric correction is rather insensitive to $\log g$ and quite insensitive to He/H; the values quoted here are for He/H=0.05, which differ by less than 0.06 magnitude from the values for He/H=0.50. These values show good agreement with the values obtained empirically by Code *et al.* [15] for stars with effective temperatures in the range of the theoretical calculations presented here.

Table 6. Bolometric Corrections

$\log g$	T_{eff}	B.C.
4.5	30000	-2.86
	32500	-3.00
	35000	-3.16
	37500	-3.34
	40000	-3.53
	45000	-3.90
	50000	-4.24
4.0	30000	-2.81
	32500	-2.97
	35000	-3.16
	37500	-3.35
	40000	-3.55
	45000	-3.92
	50000	-4.27
3.5	30000	-2.78
	32500	-2.98
	35000	-3.19
	37500	-3.42
	40000	-3.64
3.0	30000	-2.84
	32500	-3.16

4. DETERMINATION OF STELLAR PARAMETERS

4.1 THE FITTING OF LINE PROFILES BY χ^2 MINIMIZATION

The parameters of each star observed were determined by the method of χ^2 minimization. The χ^2 parameter is a weighted sum of the squares of the differences between the observed data $D(\lambda)$ and theoretical profiles $T(\lambda)$, which are characterized by the parameters T_{eff}, $\log g$, He/H, and the projected rotational velocity $v \sin i$. The values of the parameters for which the value of χ^2 is a minimum is found using the implementation of the Levenberg-Marquardt algorithm given by Press *et al.* [40].

The chief advantage of this method of fitting is that it eliminates most of the personal bias that unconsciously creeps into any fit obtained by simple inspection of the observational profile superimposed on a grid of theoretical profiles. Some personal bias remains, since the user specifies which wavelength regions are too contaminated by blends or continuum errors to be included in the fit; but it is felt that this is much less than the personal bias in a fit by inspection.

Another advantage of the χ^2 minimization is that it allows information from all parts of all the line profiles to be taken into account simultaneously. The human eye and brain cannot do this for eleven detailed line profiles, and so the human observer must pick a subset of lines sensitive to each parameter and estimate each parameter iteratively from a limited subset of the data.

As a result of the uncertainty of the continuum level and wavelength calibration, the actual fit of theoretical profiles to the observed profiles must allow a certain amount of flexibility in the assumed continuum level and wavelength zero point. This is introduced through three free parameters for each line that are allowed to vary with the physical parameters in the fit by χ^2 minimization. If $T_o(\lambda)$ is the theoretical profile of a given line, the profile used in the actual fit has the form

$$T(\lambda) = (a + b\lambda)T_o(\lambda + c),$$

where it is assumed that $a \sim 1.0$ and b and c are small.

Some justification for this use of a rather large number of "fudge parameters" is in order. The parameter c destroys the information contained in the central wavelength of each line. Since the central wavelengths are only very weakly dependent on the physical parameters, the information contained in the wavelength scale is swamped by noise resulting from the imperfect wavelength calibration. It is therefore reasonable to eliminate both the signal and noise by permitting the zero point of the wavelength scale for each line to be a free parameter in the χ^2 minimization. The continuum flux level (and the information in it) have already been eliminated by the normalization of the line profiles to the continuum level. This normalization is "fine-tuned" by the parameter a with no loss of information.

The hydrogen and ionized helium line profiles are symmetric through the properties of the linear Stark effect; the neutral helium line profiles are approximately symmetric for all but the lines with forbidden components (the 4471Å and 4922Å lines), and these are approximately symmetric after convolution with the rotational and instrumental profiles. Thus, an expansion of the line profiles about their centers is dominated by even terms, whereas local errors in the continuum normalization may be expected to be dominated by a linear term. The parameter b eliminates the noise from the lead term in the continuum level with little loss of information about the line profile.

The use of the line-fitting program, SPECTRUM, is as follows. SPECTRUM first prompts for a video-display device name, which is passed to the PGPLOT graphics library [39] initialization routines. The user then gives the names of the files containing the observed profiles. SPECTRUM plots the data and prompts the user for limits on the shoulders of the profile and the profile center; these are used to fit a Gaussian to the line profile and to obtain good starting values for a, b, and c. The user then supplies the program with starting estimates of the physical parameters T_{eff}, $\log g$, and He/H. For the stars analyzed here, the starting estimates of T_{eff} and $\log g$ were obtained from the calibration of Humphreys [24], and He/H was assumed to be equal to 0.11, close to the cosmic ratio. The program then determines which grid points enclose the starting estimates and reads the corresponding eight theoretical profiles from disk libraries. The user then gives the estimated

$v \sin i$ for the object, and SPECTRUM calculates the convolution of the theoretical profiles with a rotational profile. Mihalas and Auer [35] have shown that because of the reduced limb darkening in non-LTE atmospheres, this is a sufficient approximation of the effects of rotation.

The user then instructs SPECTRUM to carry out iterations of the χ^2 minimization. After each iteration, SPECTRUM returns the current value of χ^2 and asks the user whether to make another iteration. When the user is satisfied that the minimum has been found, SPECTRUM uses the covariance matrix from the minimization to give a conservative estimate of the uncertainty of the fit. The user may try a different set of initial values for the parameters (if, for example, the minimum that has been found lies outside the mesh in parameter space enclosing the initial guess) or may make a revised estimate of the value of $v \sin i$. Since the latter parameter is very insensitive to the estimates of the three physical parameters, it is not included in the χ^2 minimization.

When the user is satisfied with the fit, SPECTRUM prompts for a hardcopy- device type and produces plot files of the final fit.

4.2 RESULTS FOR THE SAMPLE OF O STARS

Estimated parameters for the objects observed are given in Table 7 . Figure 6 gives the actual fits obtained. It seems clear from these profiles and fits that the introduction of the three continuum variables was necessary. In most cases the resulting fits are quite good, particularly for the ionized helium lines, indicating that the Schöning-Butler ionized helium profiles are satisfactory.

The values of the projected rotational velocity $v \sin i$ include the effects of the instrumental profile; given the lowest values of $v \sin i$ measured, it appears that rotational velocities less than about 50 km/sec are not resolved.

It is important to determine the ambiguity of the fit (i.e., the range of parameters over which the fit remains good). SPECTRUM quotes errors calculated by determining the change in the parameters that would increase the value of χ^2 by 30%. These errors appear to be much too conservative; this is understandable, since most of the data points will be insensitive to one or more of the physical

Table 7. Estimated Parameters of Objects Observed

Star	T_{eff}	$\log g$	He/H	$v \sin i$	$(\mathrm{He/H})_\circ$
Per OB1					0.11 ± 0.03
236894	36600 ± 1600	3.97 ± 0.46	0.16 ± 0.07	132	0.10
12993	42900 ± 1100	3.78 ± 0.13	0.36 ± 0.05	101	0.17
13022	30400 ± 500	3.07 ± 0.09	0.35 ± 0.04	124	0.08
Aur OB2					0.09 ± 0.02
242908	47900 ± 1100	3.82 ± 0.11	0.34 ± 0.10	110	0.17
242926	40800 ± 1400	4.02 ± 0.24	0.12 ± 0.02	75	0.08
242935	41000 ± 800	4.22 ± 0.20	0.08 ± 0.03	60:	0.06
35619	40600 ± 610	4.07 ± 0.14	0.11 ± 0.03	66	-0.13
Gem OB1					0.11 ± 0.04
42088	42900 ± 700	3.91 ± 0.21	0.25 ± 0.09	80	0.14
254755	37900 ± 900	4.15 ± 0.25	0.11 ± 0.03	82	0.08
Cyg OB1					0.08 ± 0.02
193595	41300 ± 500	3.77 ± 0.20	0.39 ± 0.05	66	0.19
194094	36900 ± 1800	4.11 ± 0.54	0.16 ± 0.04	77:	0.11
194280	28500 ± 800	2.71 ± 0.10	0.77 ± 0.11	114	0.12
229234	33200 ± 1800	3.35 ± 0.29	0.30 ± 0.09	117	0.09
Cyg OB2					0.09 ± 0.02
# 4	42700 ± 1900	4.15 ± 0.36	0.18 ± 0.05	101	0.13
# 8B	43900 ± 1500	3.76 ± 0.35	0.13 ± 0.02	109:	0.06
# 10	28500 ± 1200	2.67 ± 0.11	0.53 ± 0.27	108	0.08
Cep OB2					0.10 ± 0.02
204827	33500 ± 1600	3.80 ± 0.32	0.14 ± 0.09	85	0.07
207198	35400 ± 1100	3.60 ± 0.29	0.24 ± 0.11	89	0.10
207538	32500 ± 600	3.70 ± 0.10	0.23 ± 0.03	52	0.10
209975	32000 ± 1000	3.25 ± 0.12	0.49 ± 0.15	97:	0.13
Miscellaneous					
9 Sgr	50500 ± 2300	4.44 ± 0.05	0.10 ± 0.06	92	0.10
λ Ori	40000 ± 1300	4.00 ± 0.17	0.18 ± 0.11	74	0.11

parameters and since the continuum parameters are held fixed. Another approach is to hold one parameter fixed at a value somewhat different from that for the best fit, and to recalculate the minimum with all other parameters allowed to vary. This method has been used to determine the errors quoted here, with $v \sin i$ being the parameter held fixed at a non-optimal value. The resulting change in χ^2 is used to estimate the change in $v \sin i$ that would significantly degrade the fit, and the corresponding change in the best fit for the physical parameters is quoted as the error.

Generally, the errors so obtained are close to the scatter one sees in the values of the physical parameters when the fit is repeated for different choices of the portion of the line spectra to fit. Occasionally, the errors seem not to be conservative enough; in these cases, the error is taken from the scatter in repeated fits, using different portions of the line profiles.

It must be emphasized that the quoted uncertainties are from the ambiguity of the fit alone; uncertainties arising from the theory are not included.

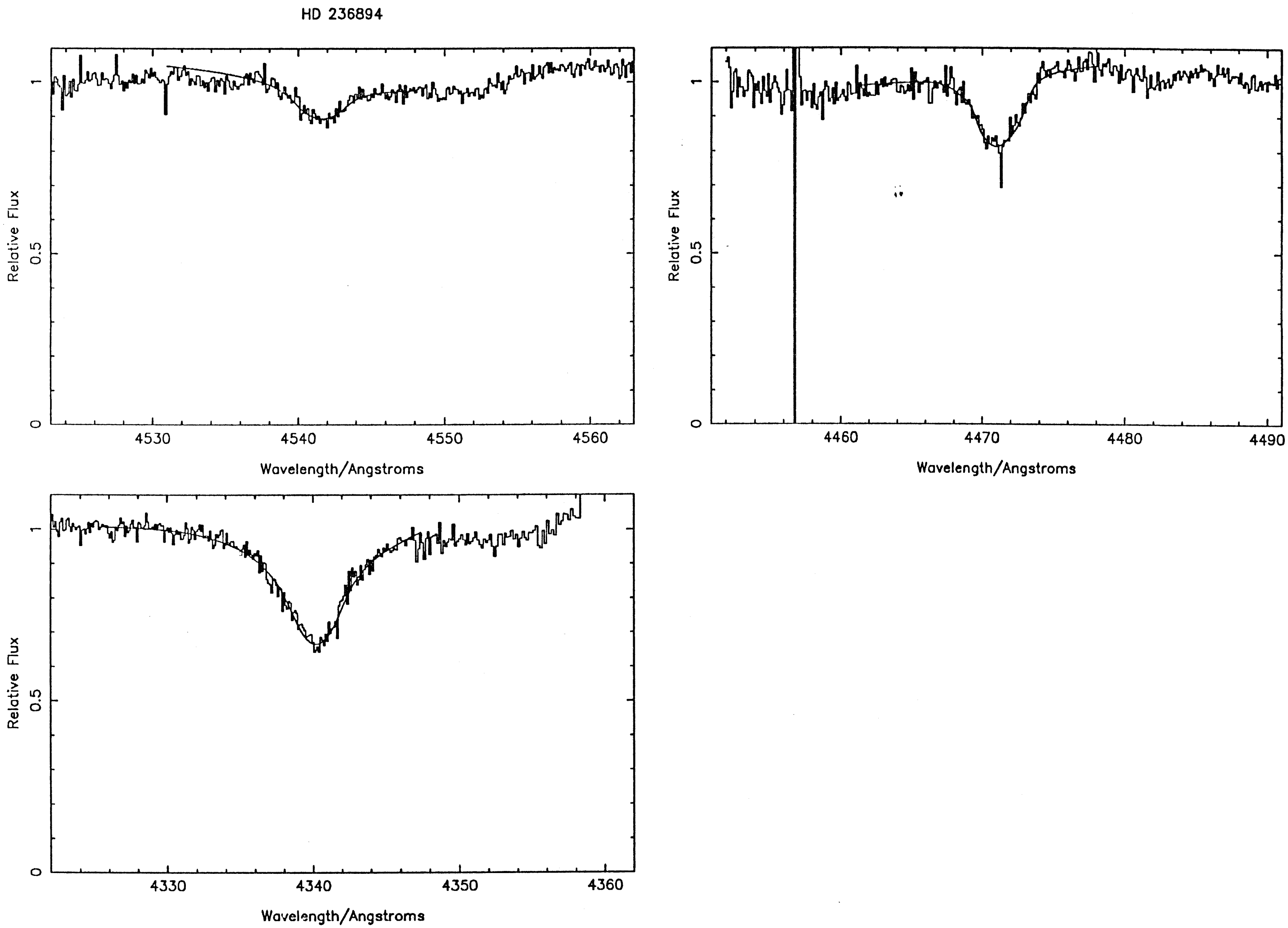

Figure 6. Observed line profiles with the best theoretical fit shown for HD 236894.

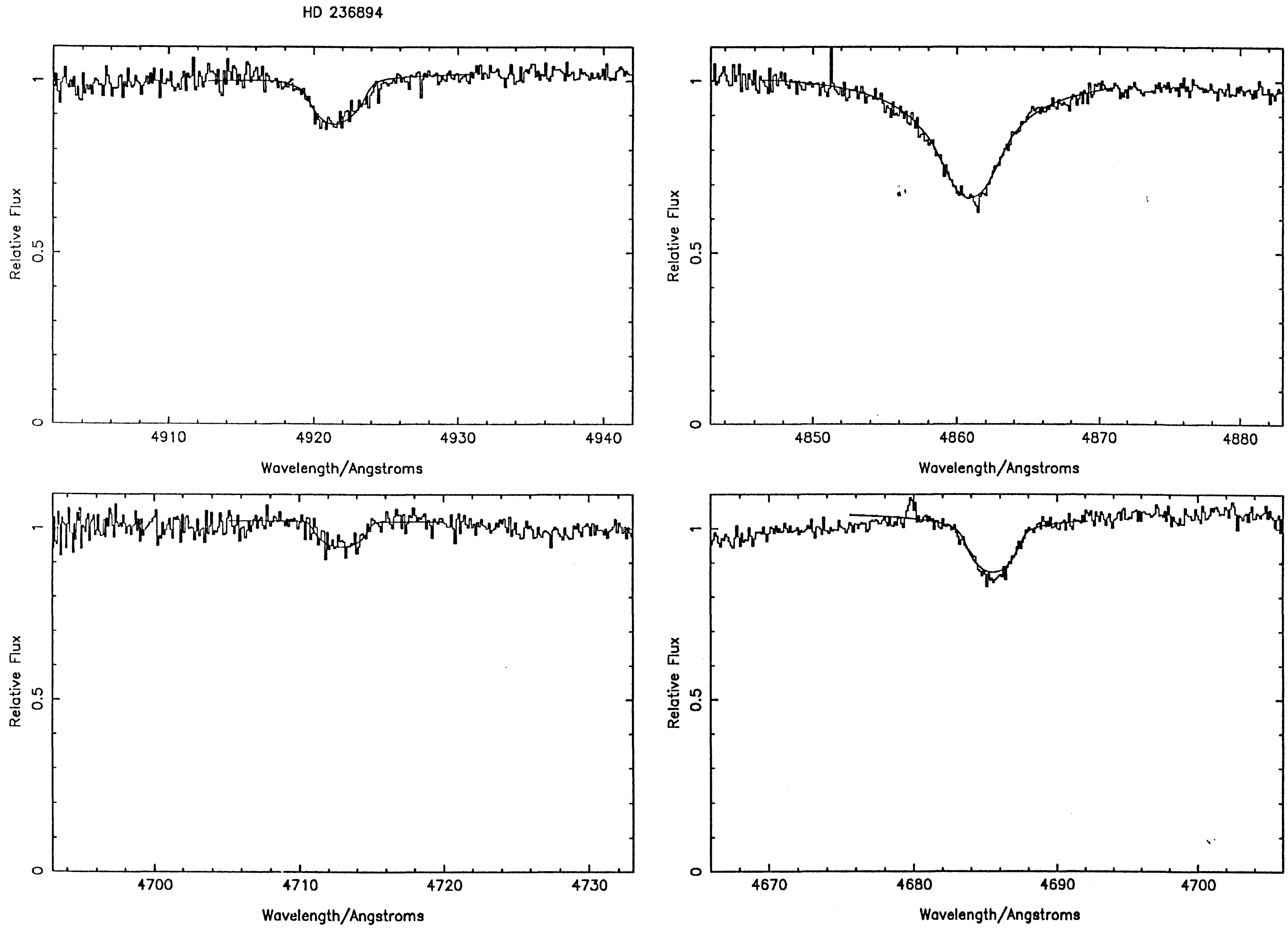

Figure 6. (Continued) HD 236894

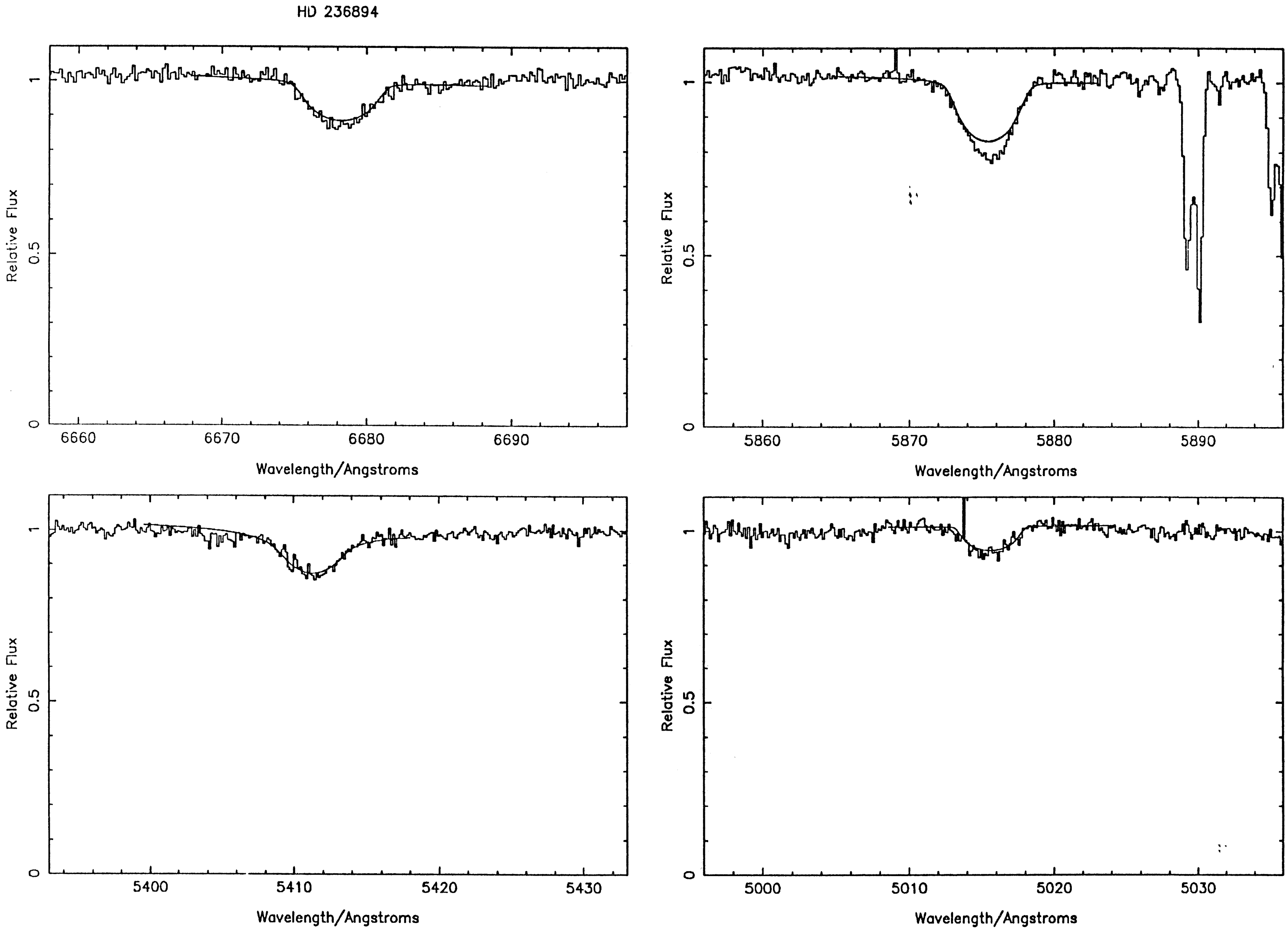

Figure 6. (Continued) HD 236894

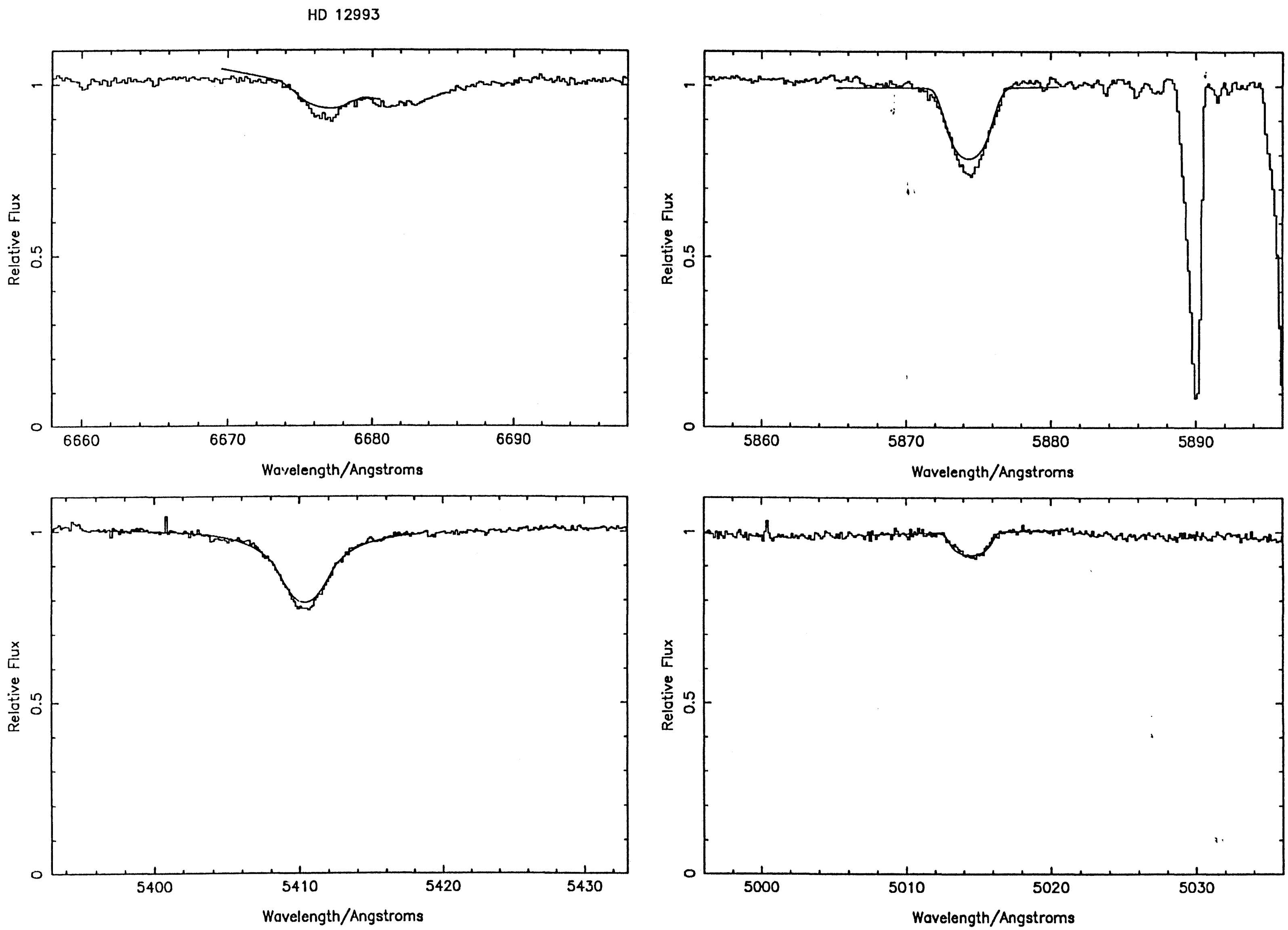

Figure 6. (Continued) HD 12993

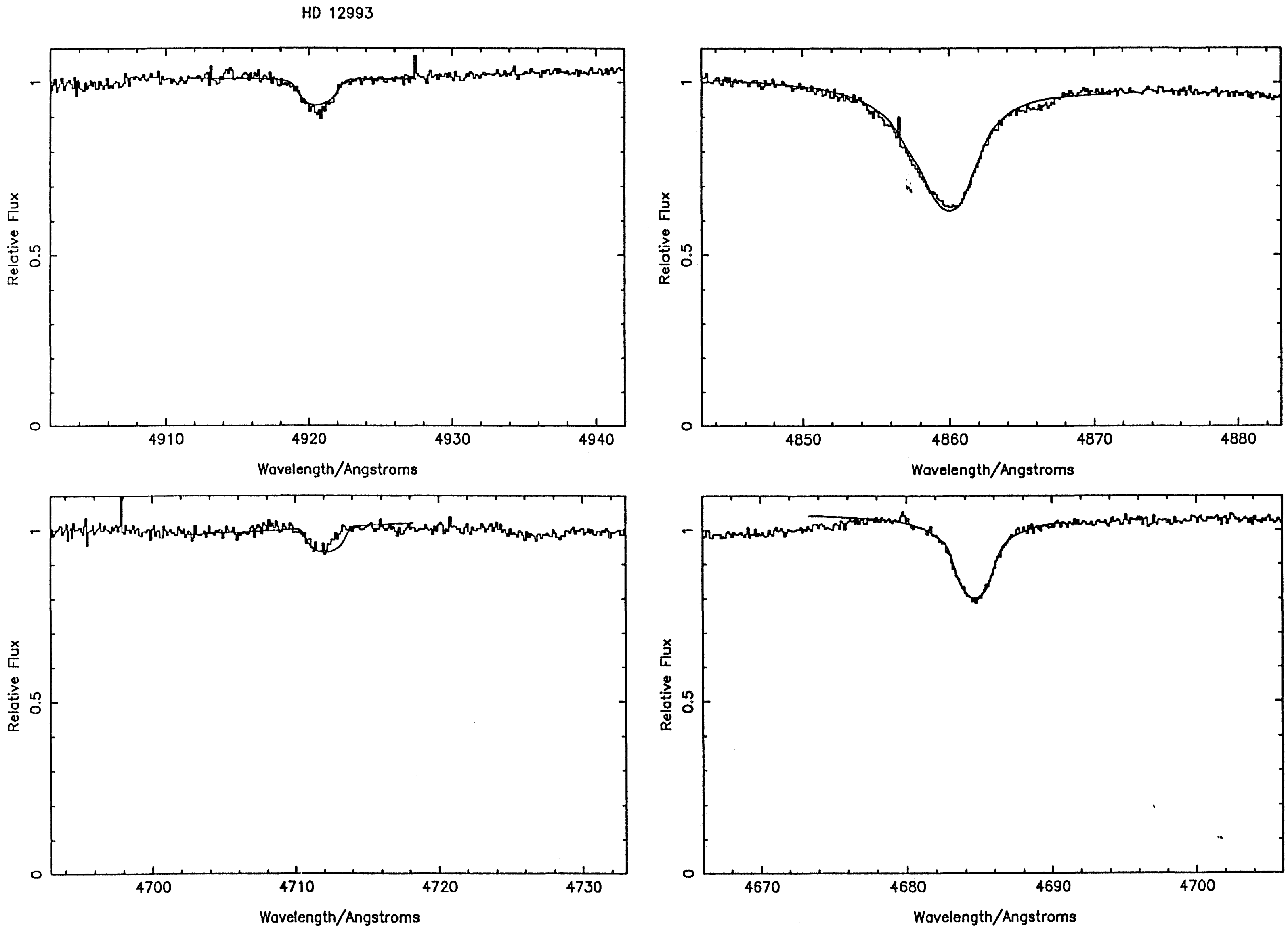

Figure 6. (Continued) HD 12993

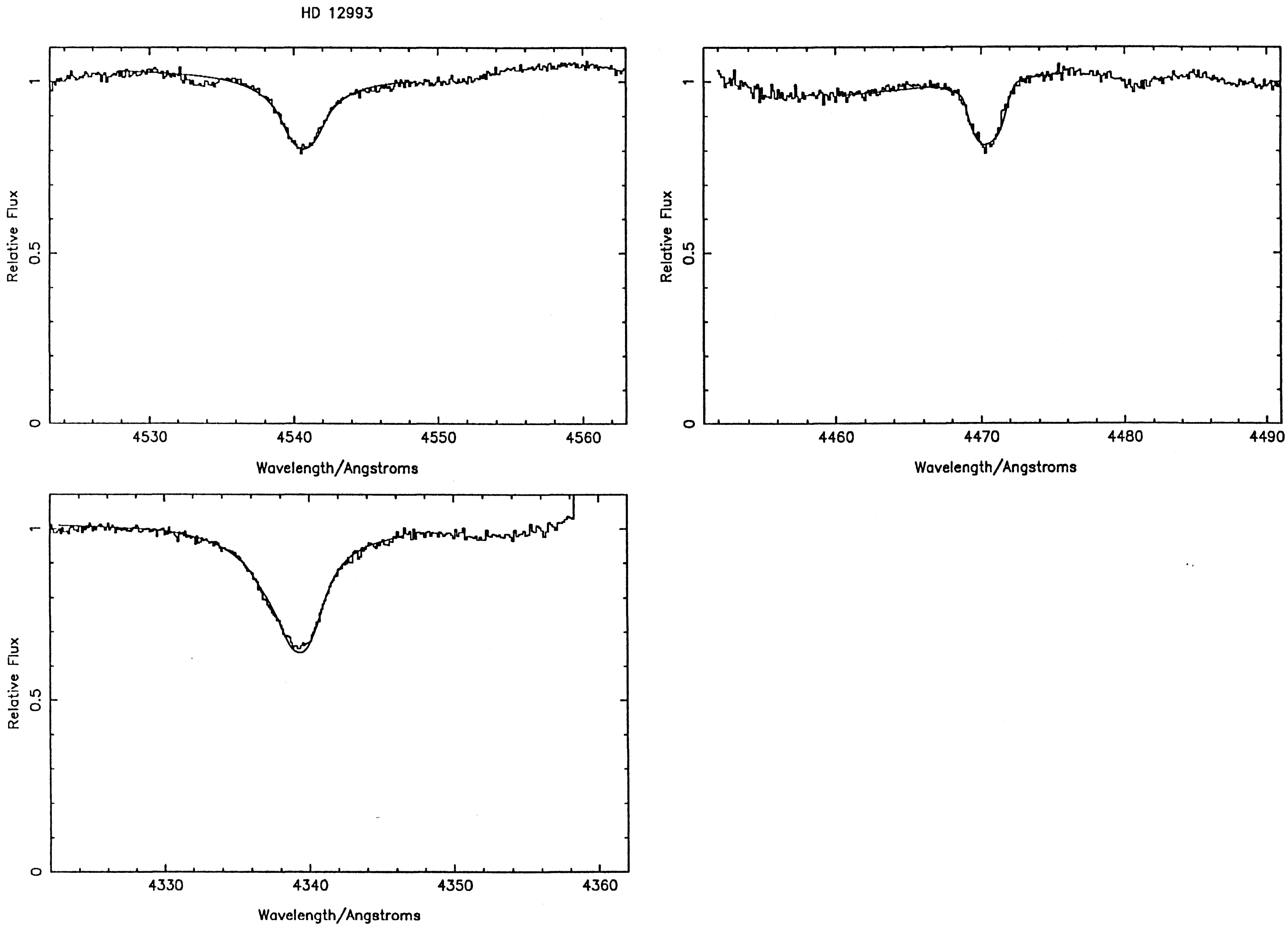

Figure 6. (Continued) HD 12993

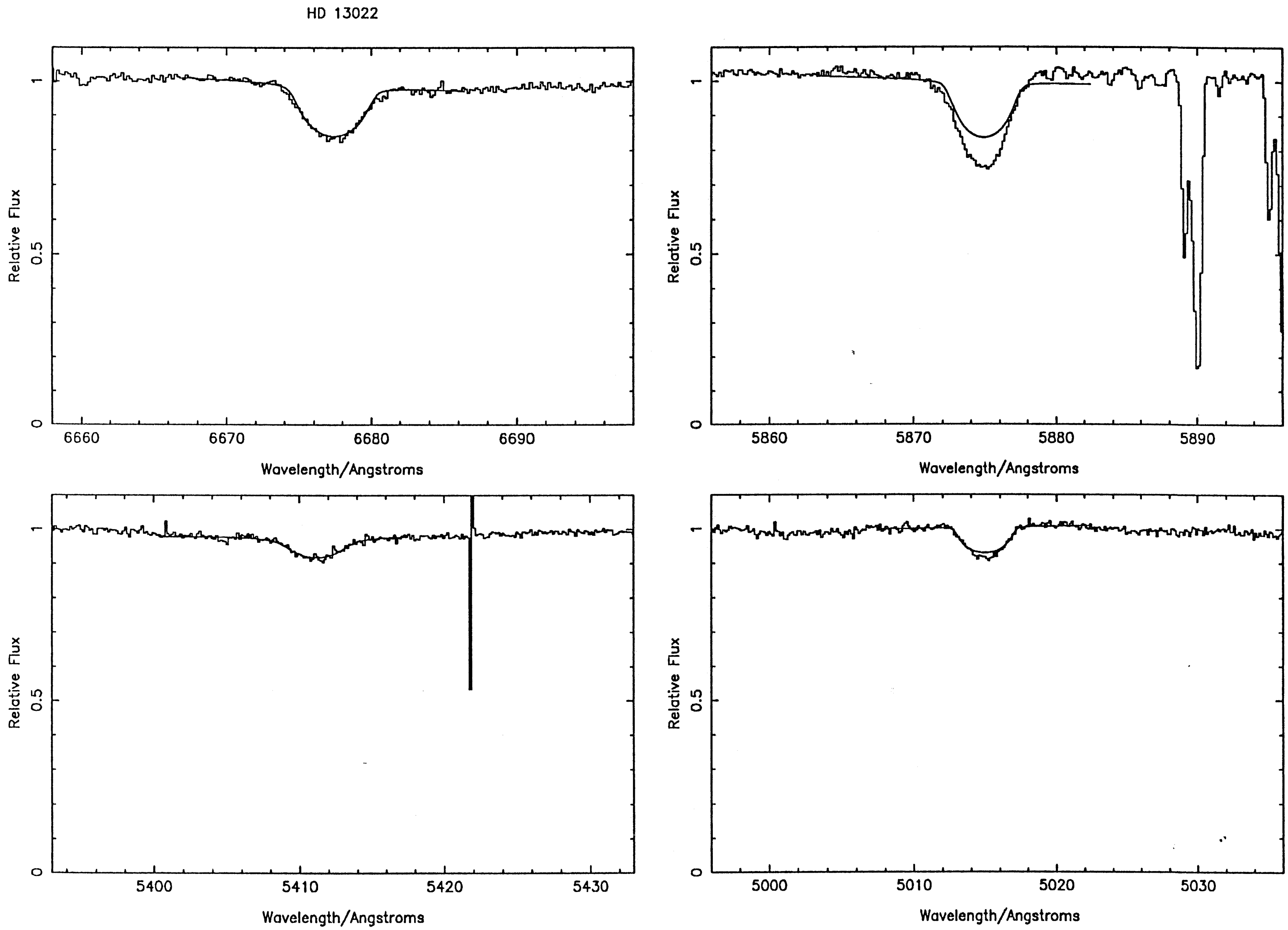

Figure 6. (Continued) HD 13022

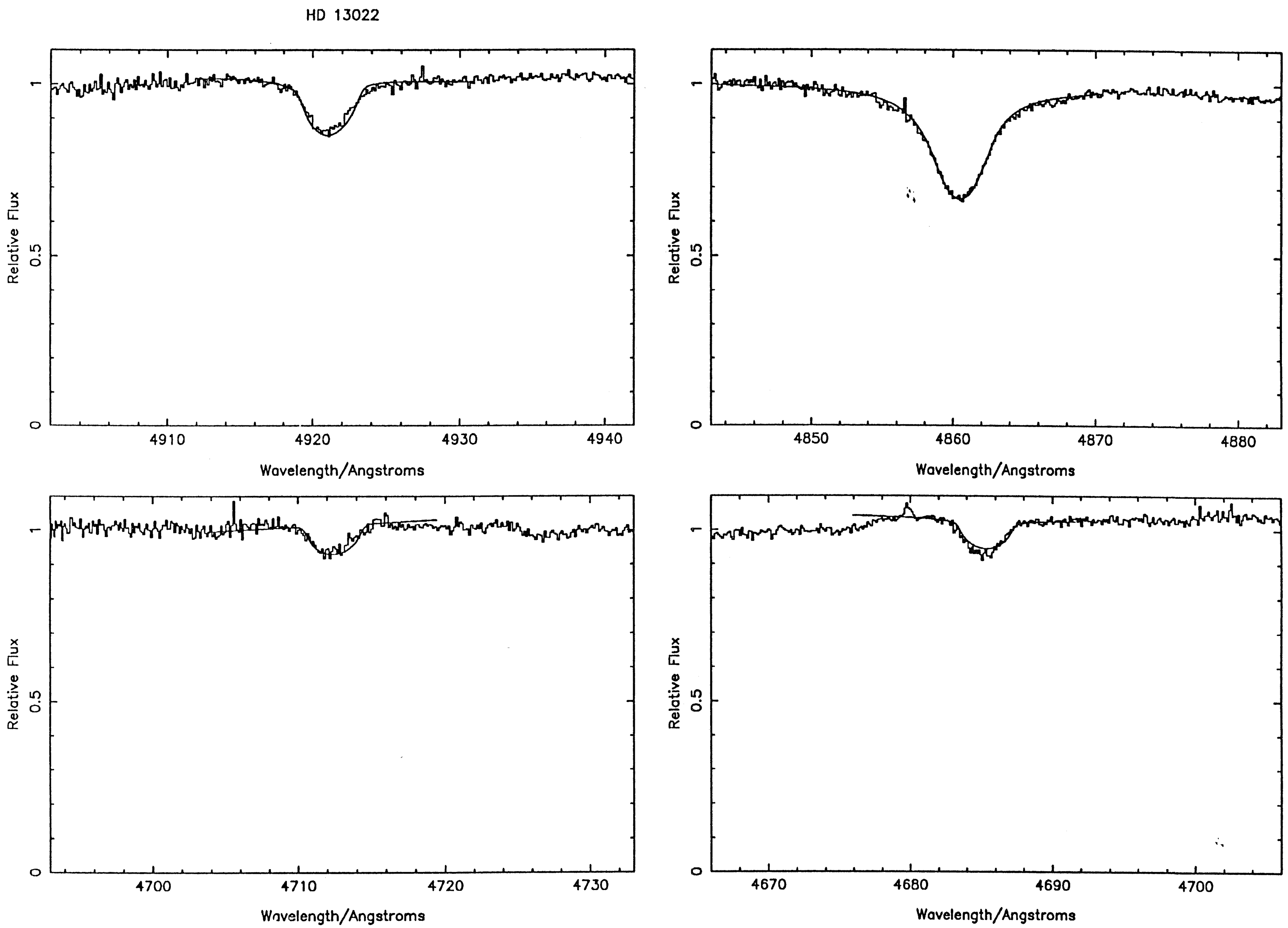

Figure 6. (Continued) HD 13022

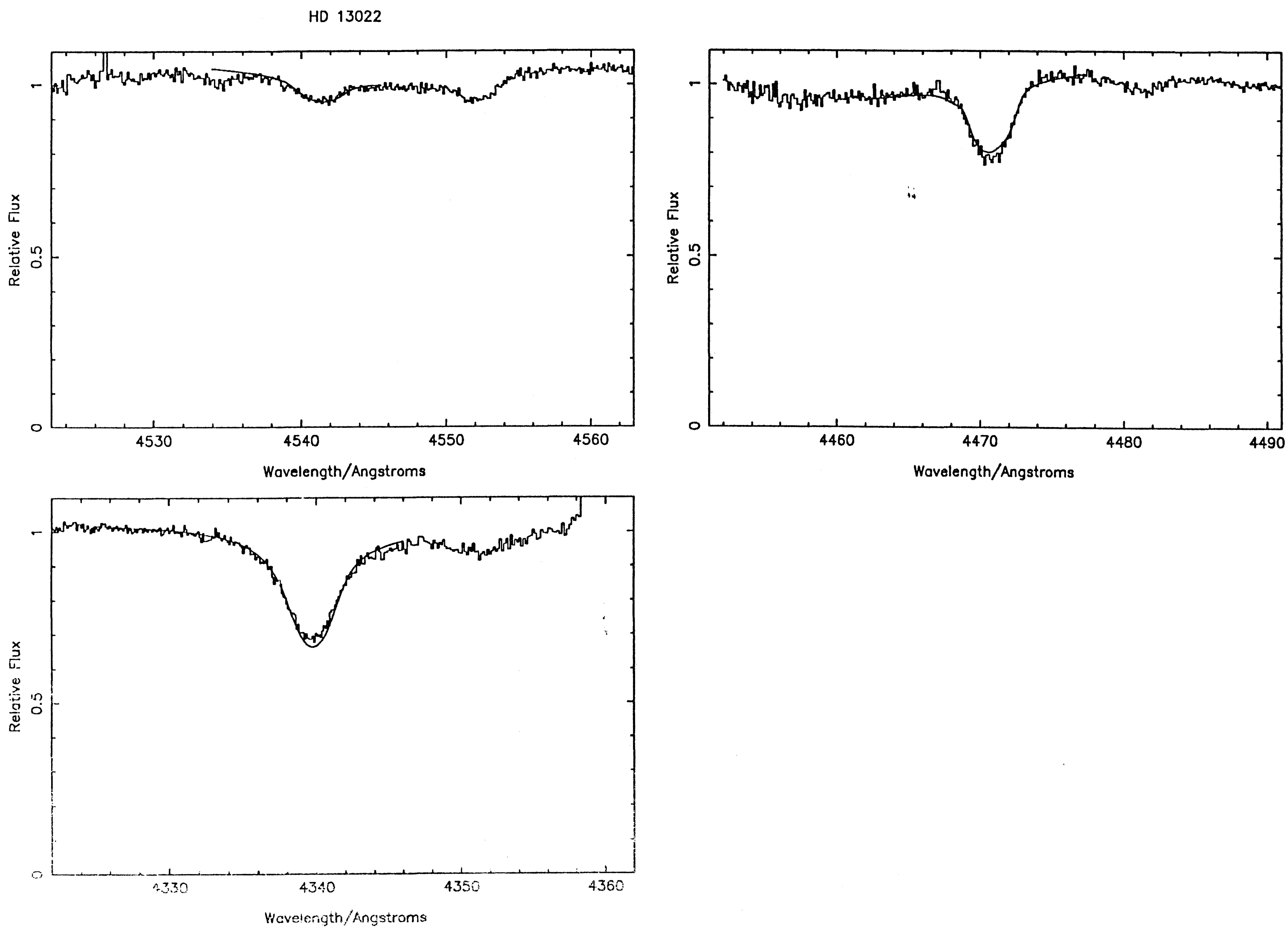

Figure 6. (Continued) HD 13022

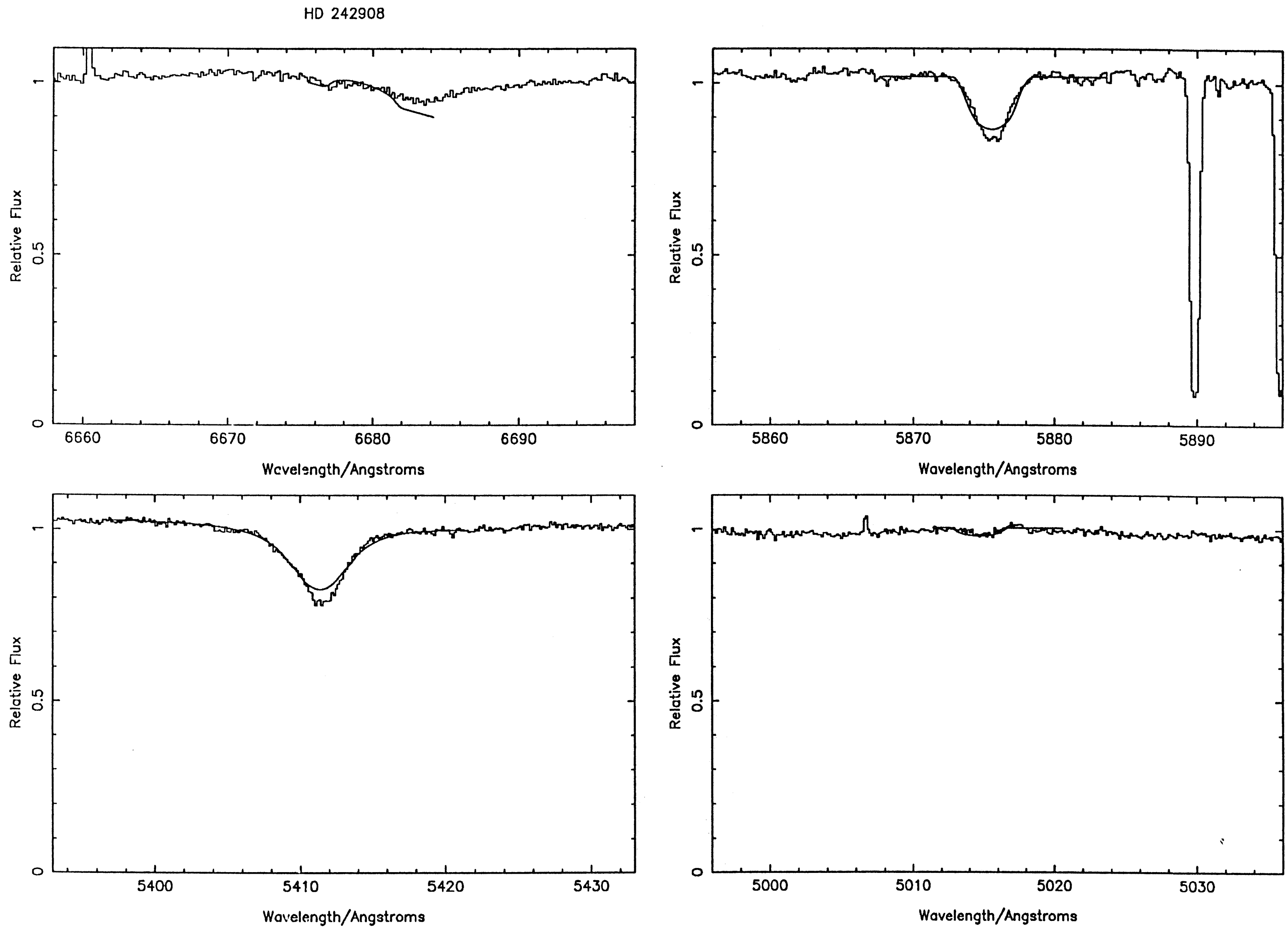

Figure 6. (Continued) HD 242908

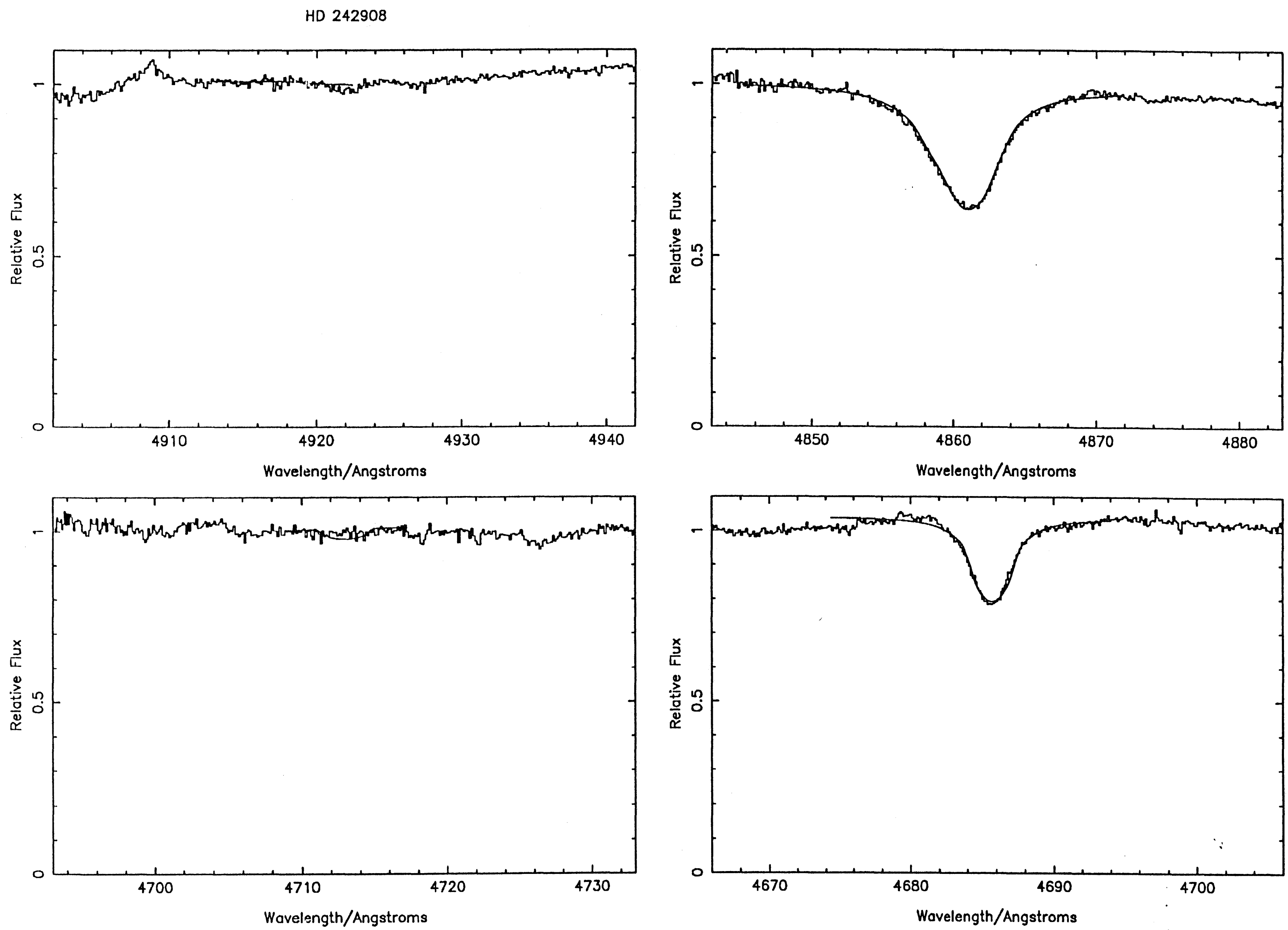

Figure 6. (Continued) HD 242908

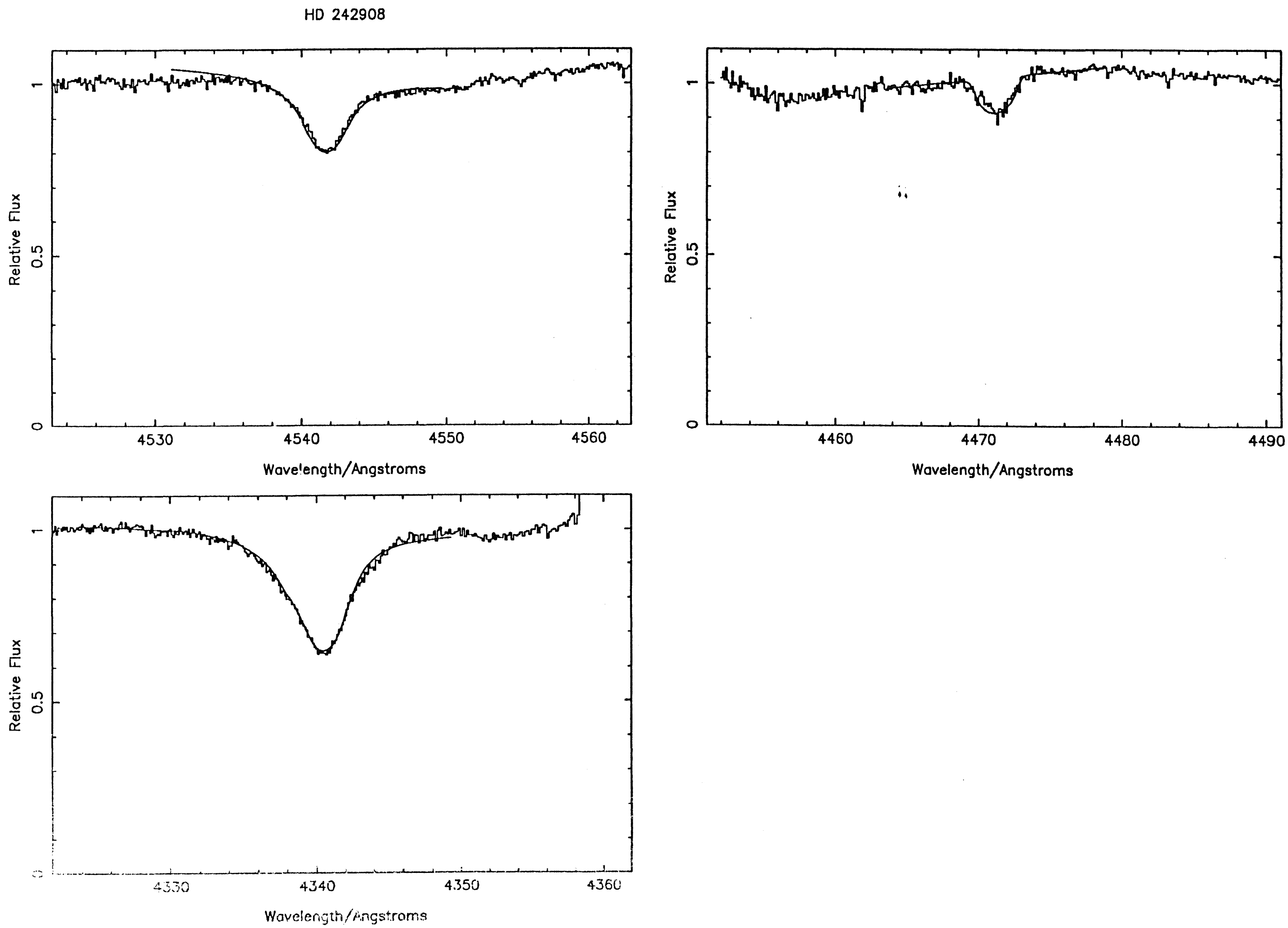

Figure 6. (Continued) HD 242908

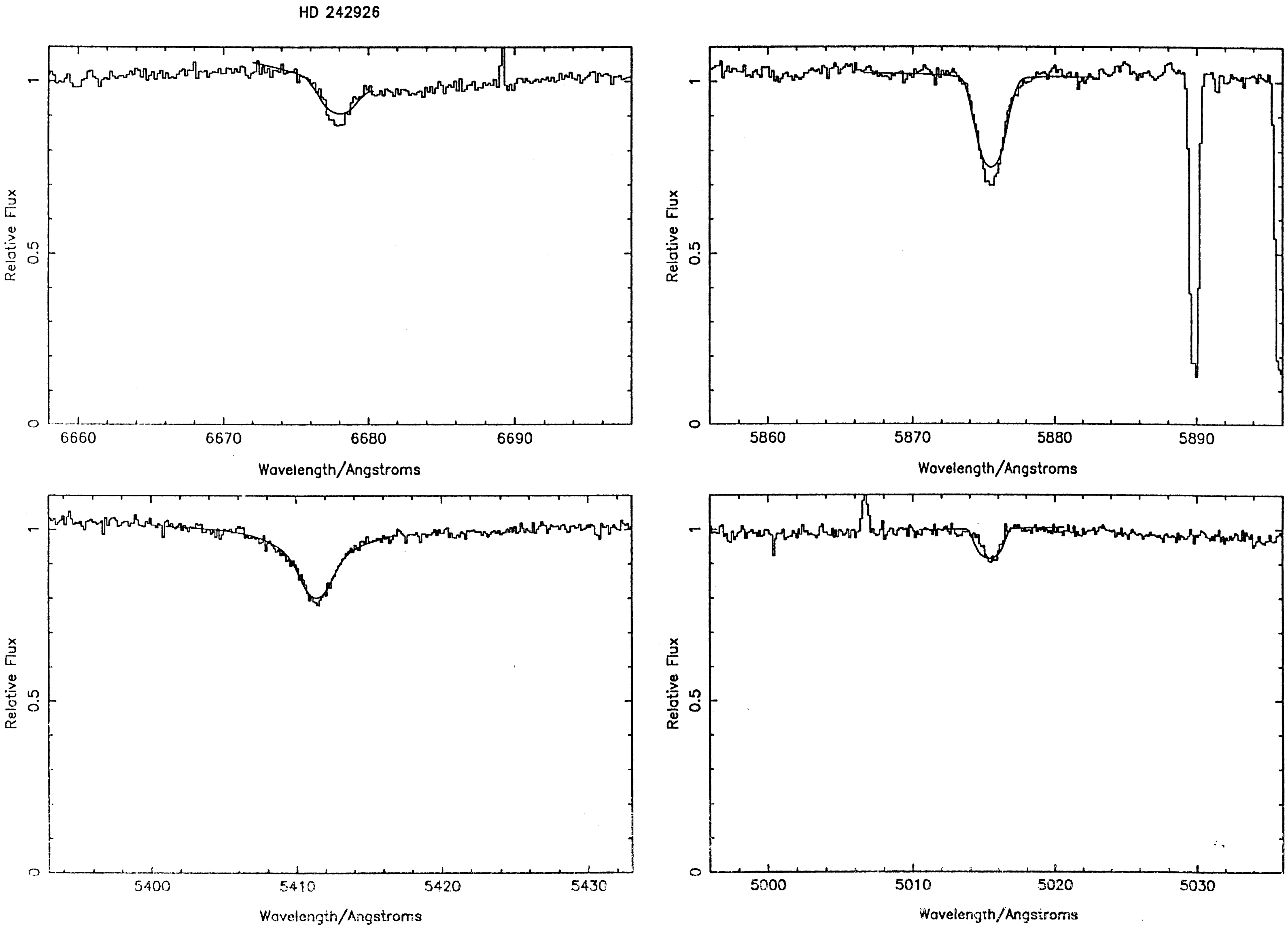

Figure 6. (Continued) HD 242926

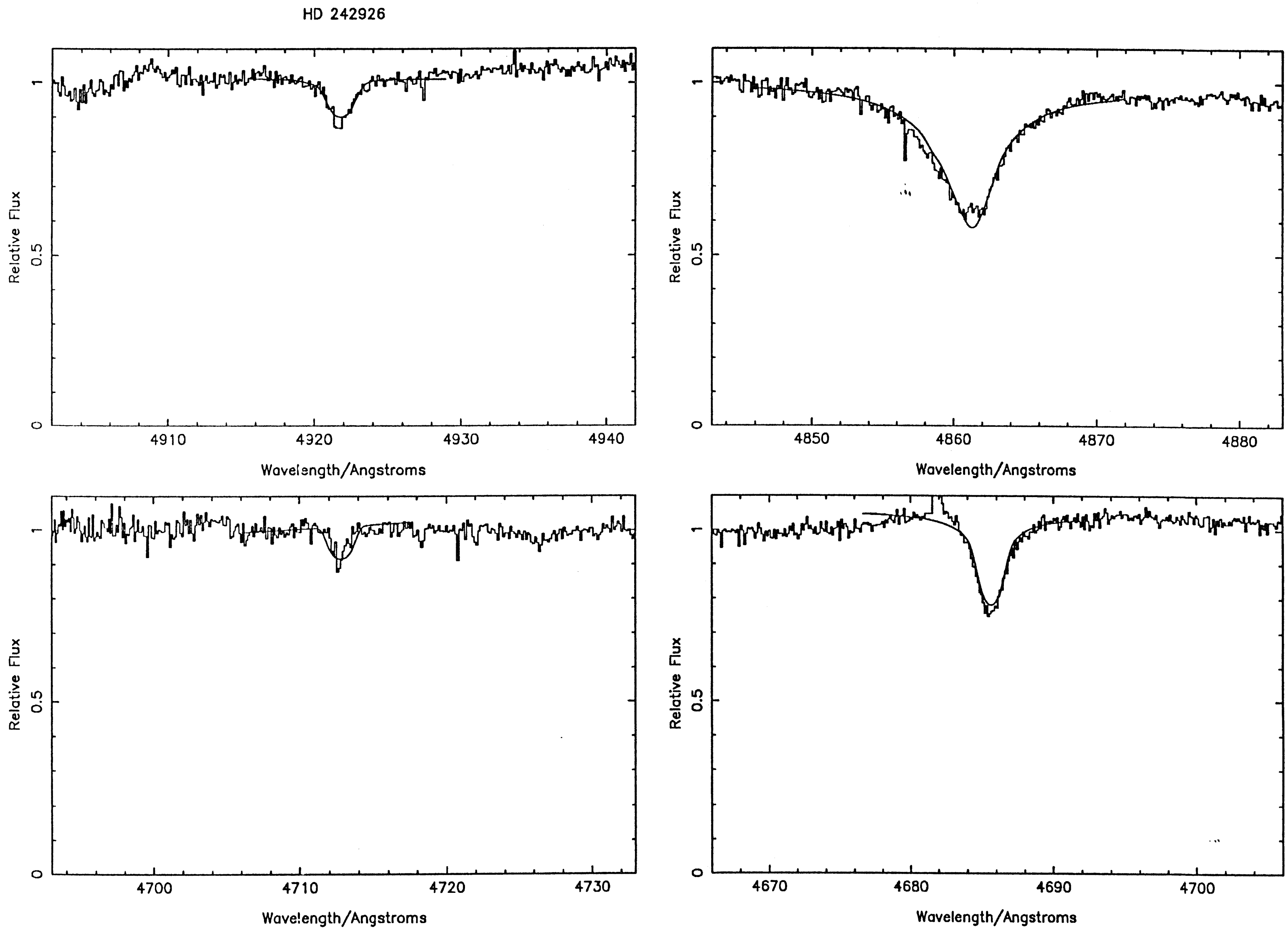

Figure 6. (Continued) HD 242926

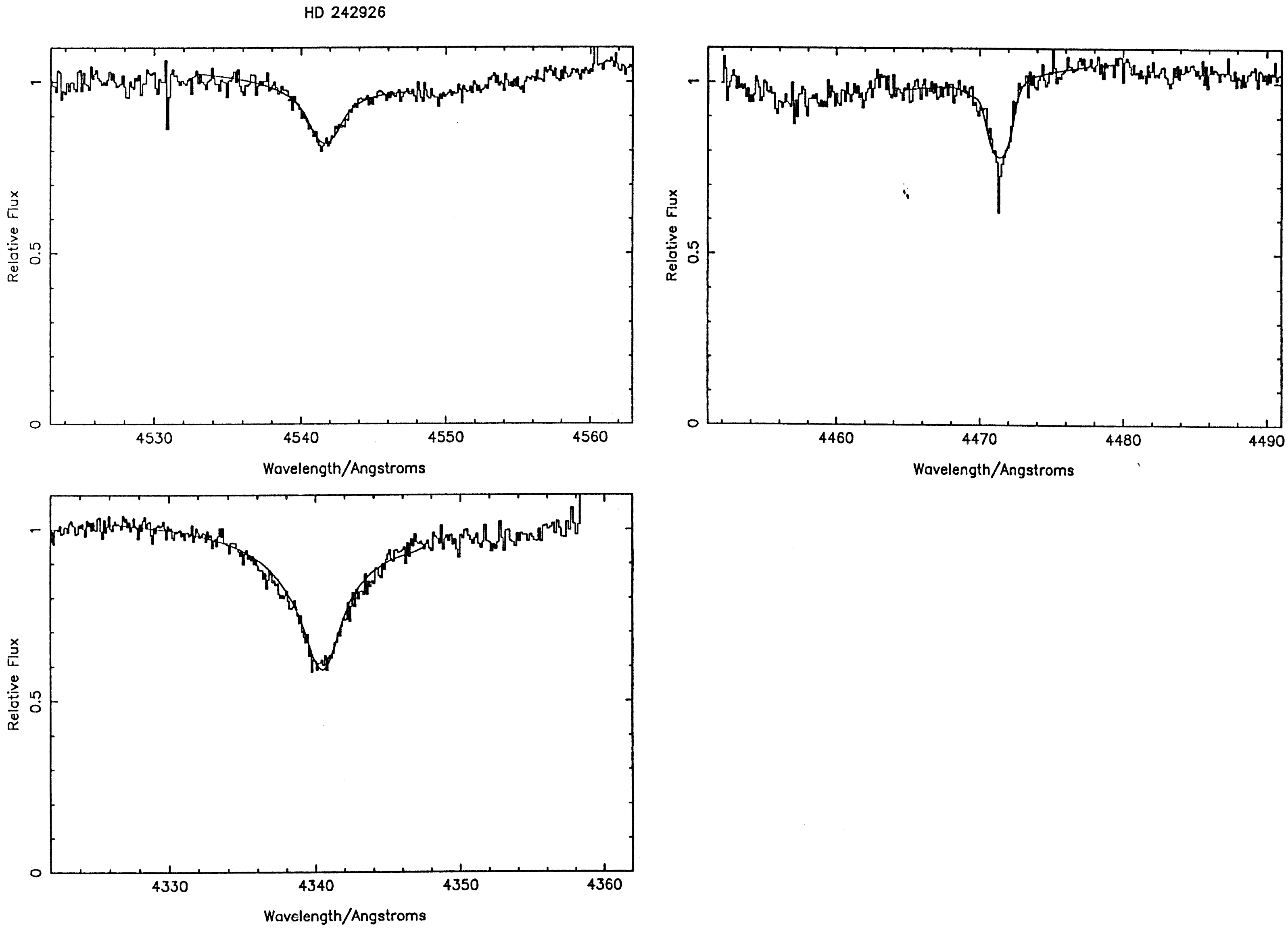

Figure 6. (Continued) HD 242926

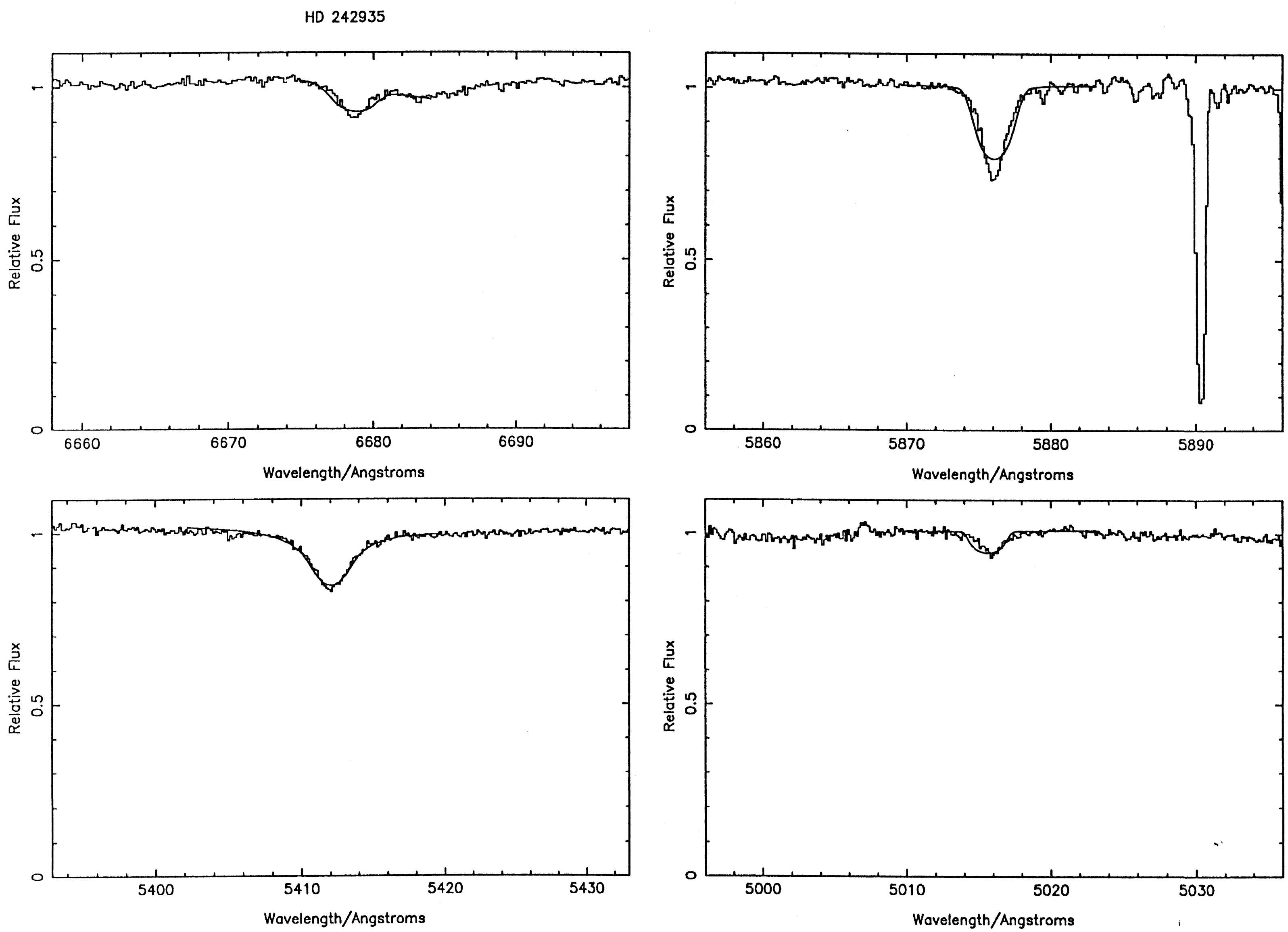

Figure 6. (Continued) HD 242935

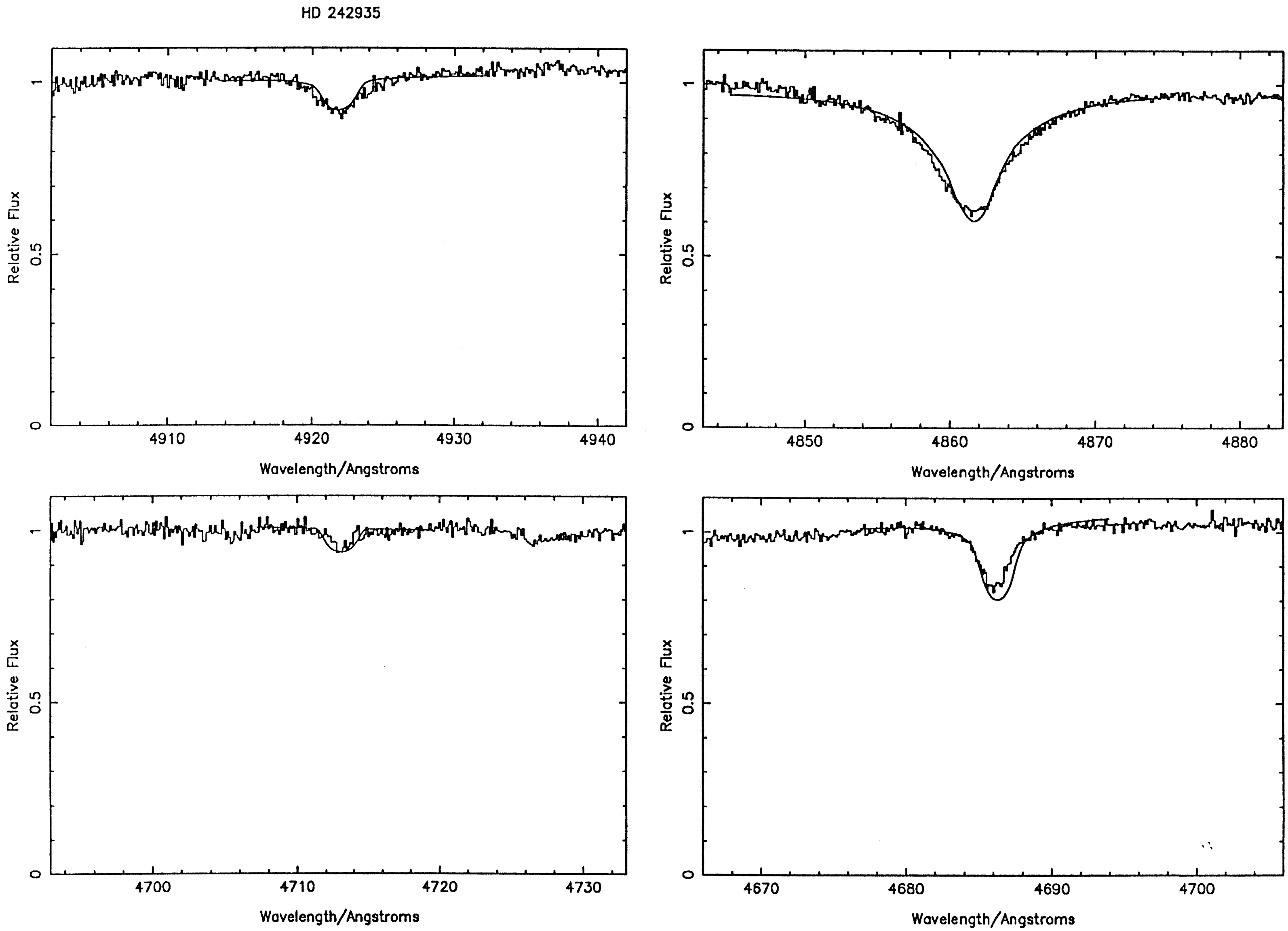

Figure 6. (Continued) HD 242935

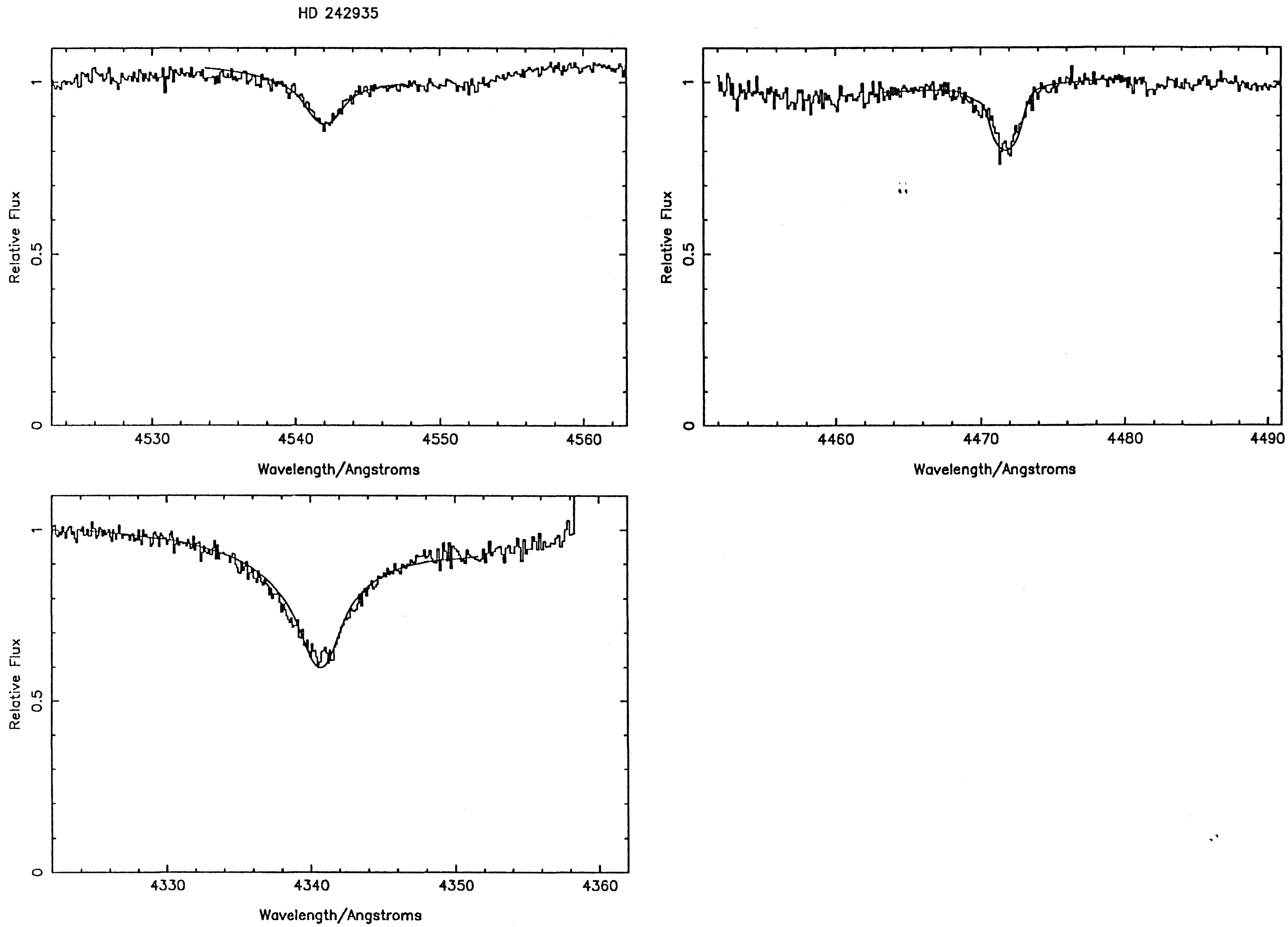

Figure 6. (Continued) HD 242935

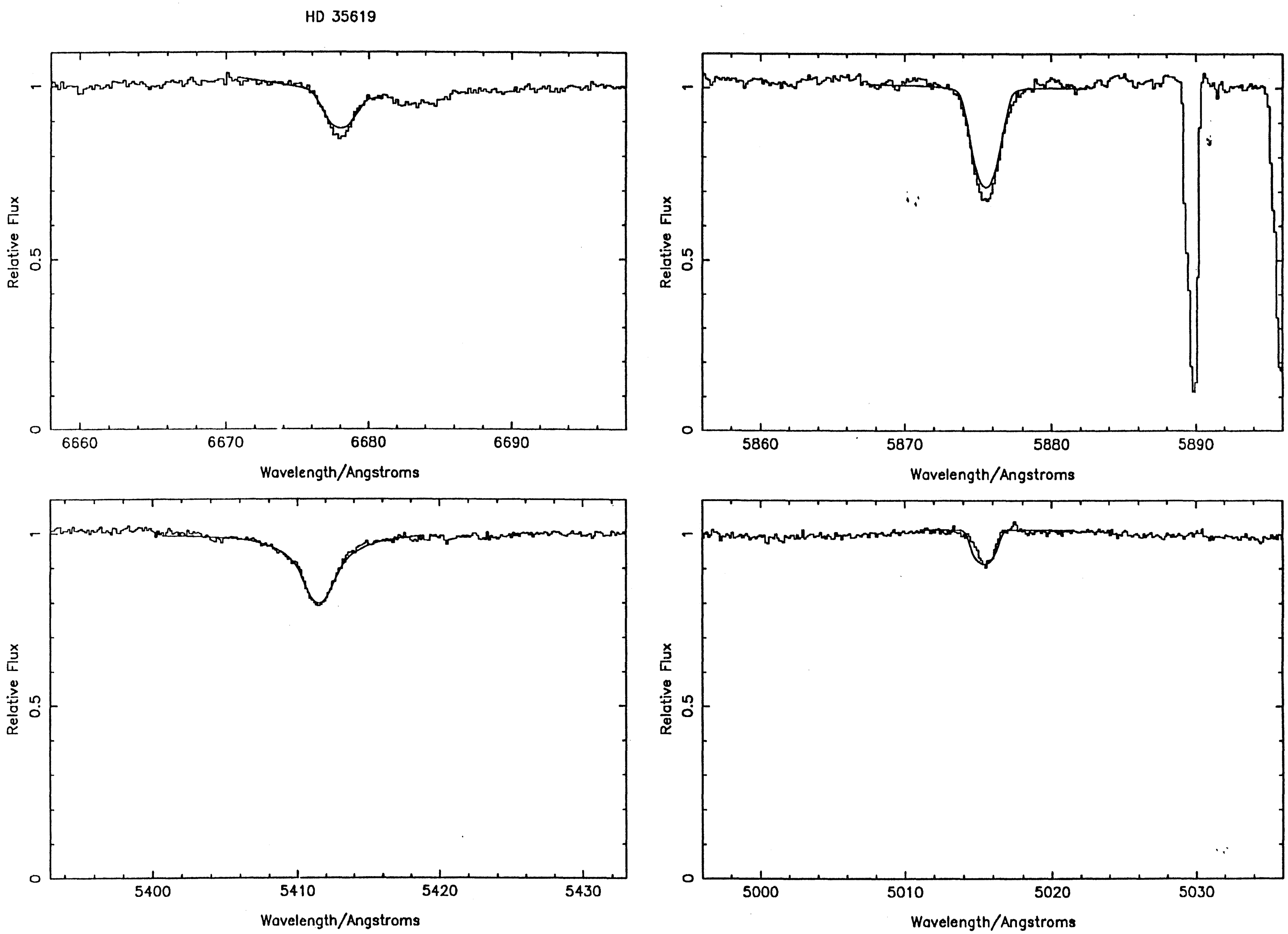

Figure 6. (Continued) HD 35619

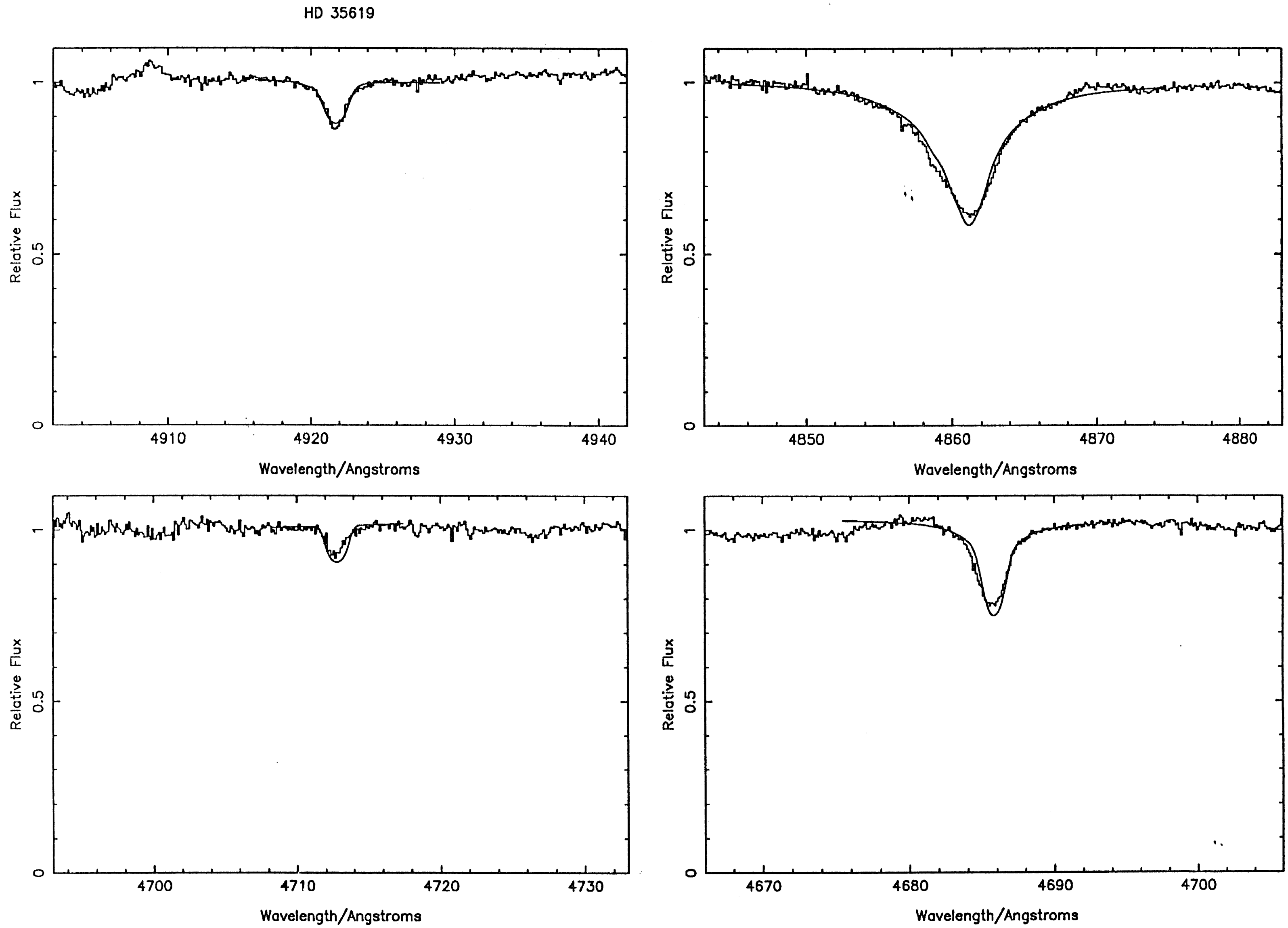

Figure 6. (Continued) HD 35619

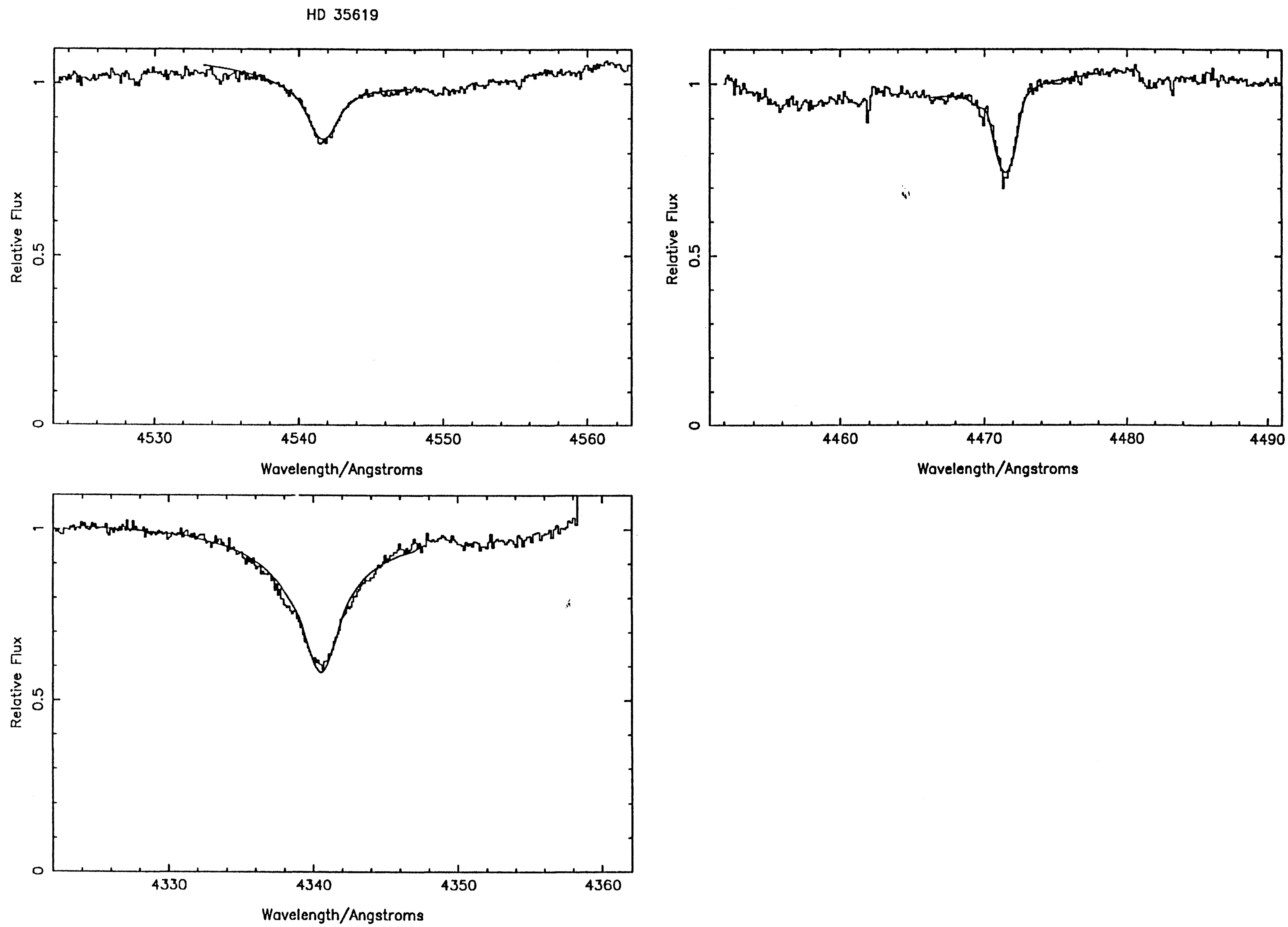

Figure 6. (Continued) HD 35619

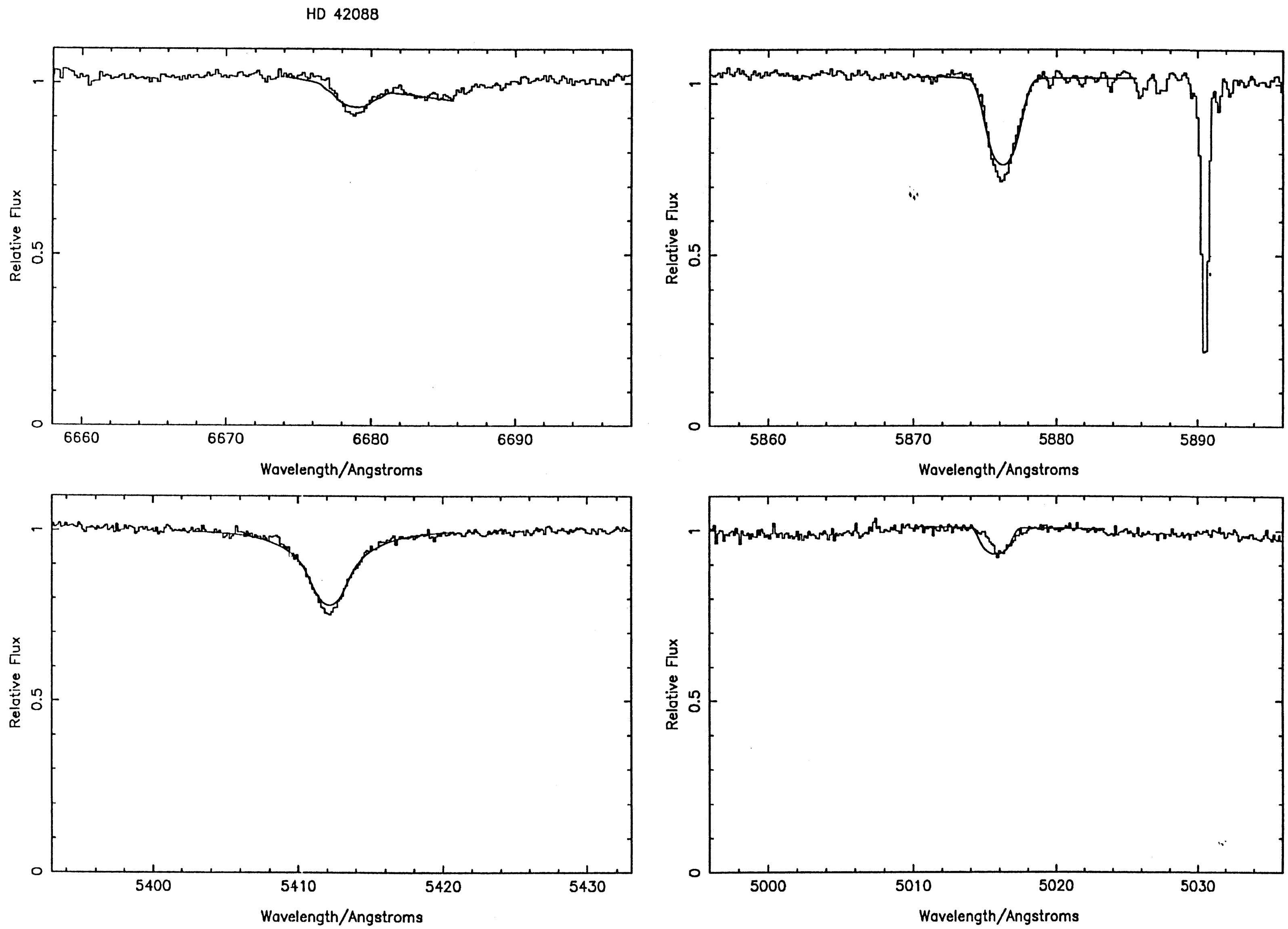

Figure 6. (Continued) HD 42088

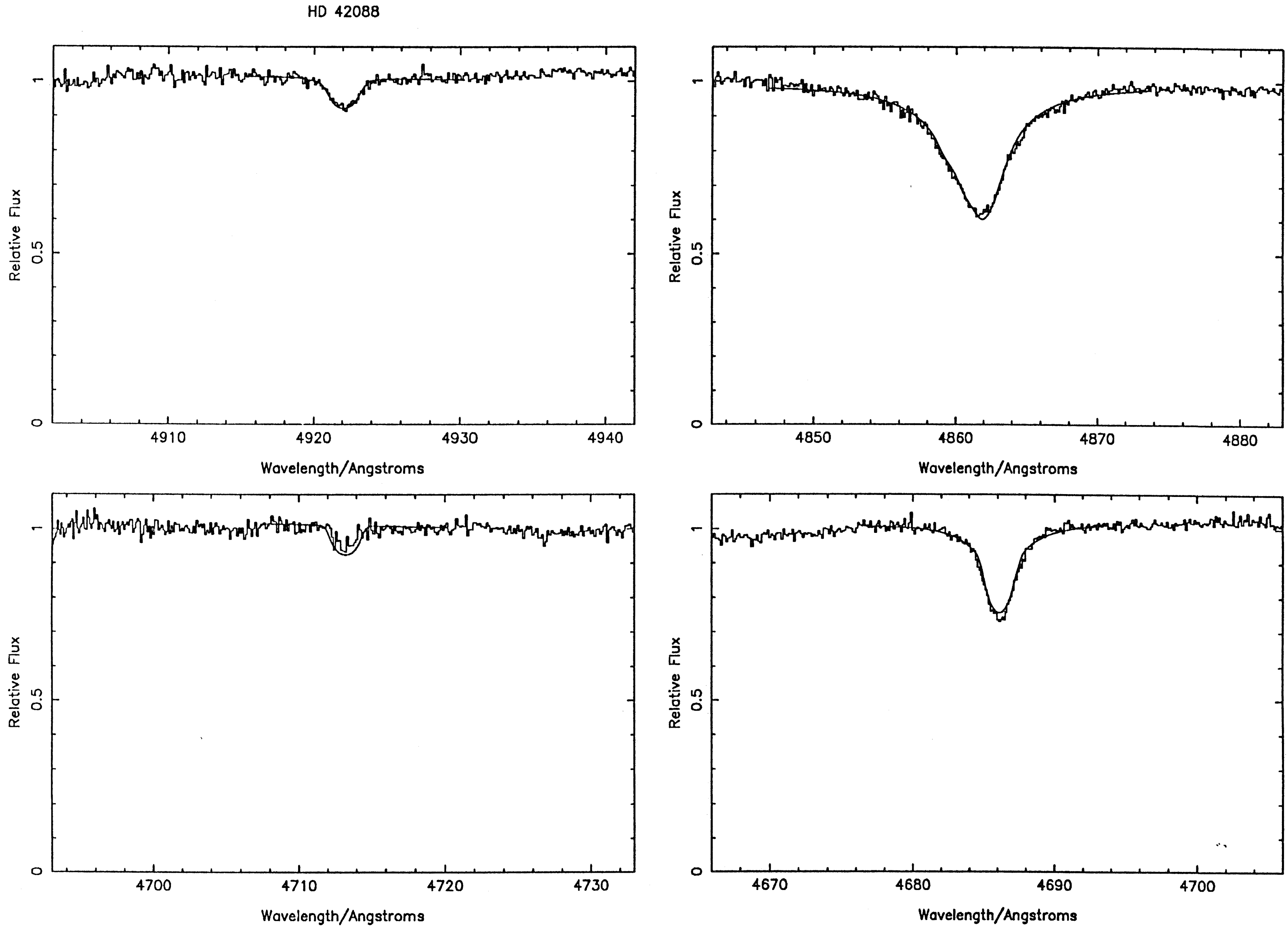

Figure 6. (Continued) HD 42088

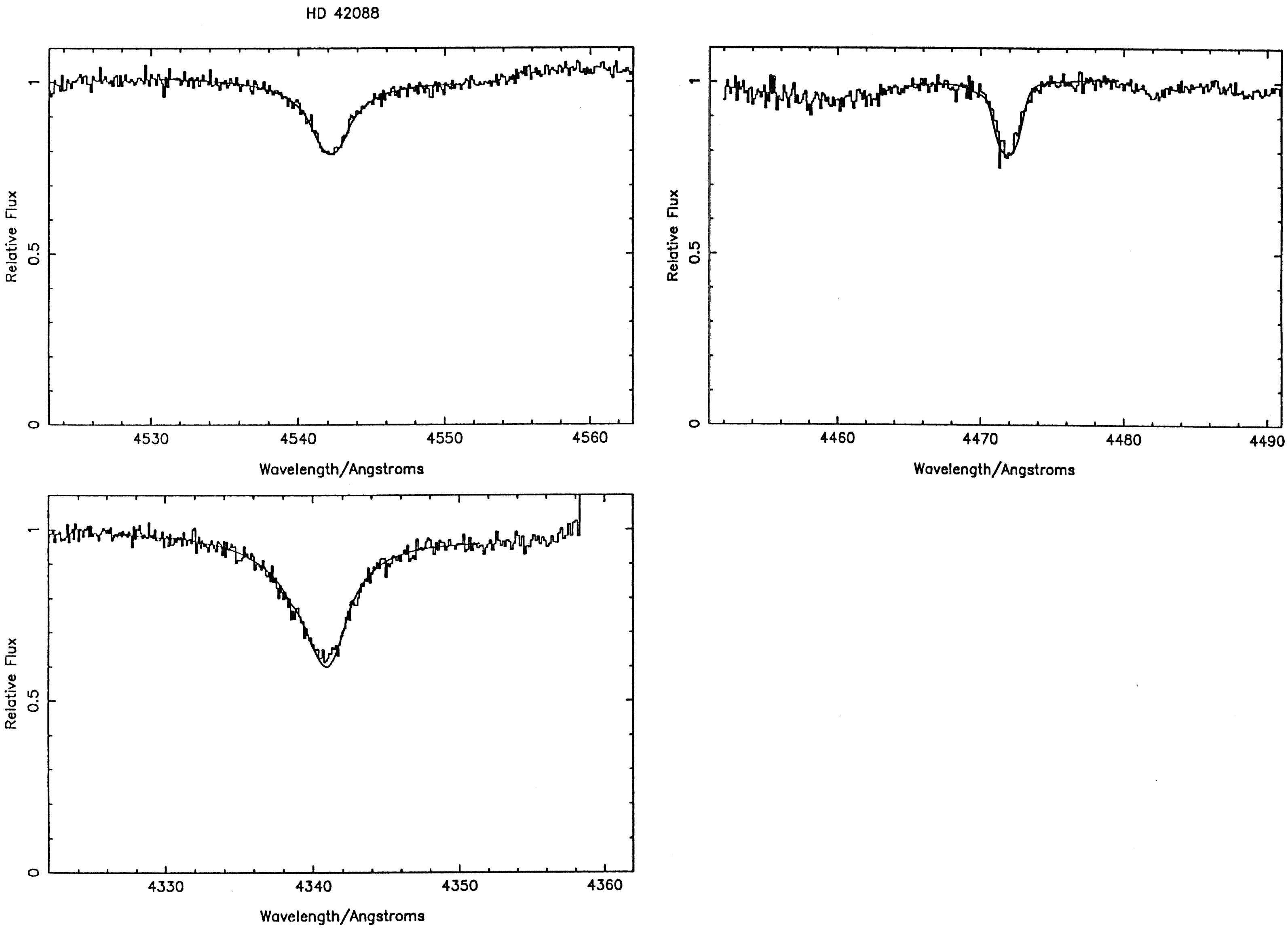

Figure 6. (Continued) HD 42088

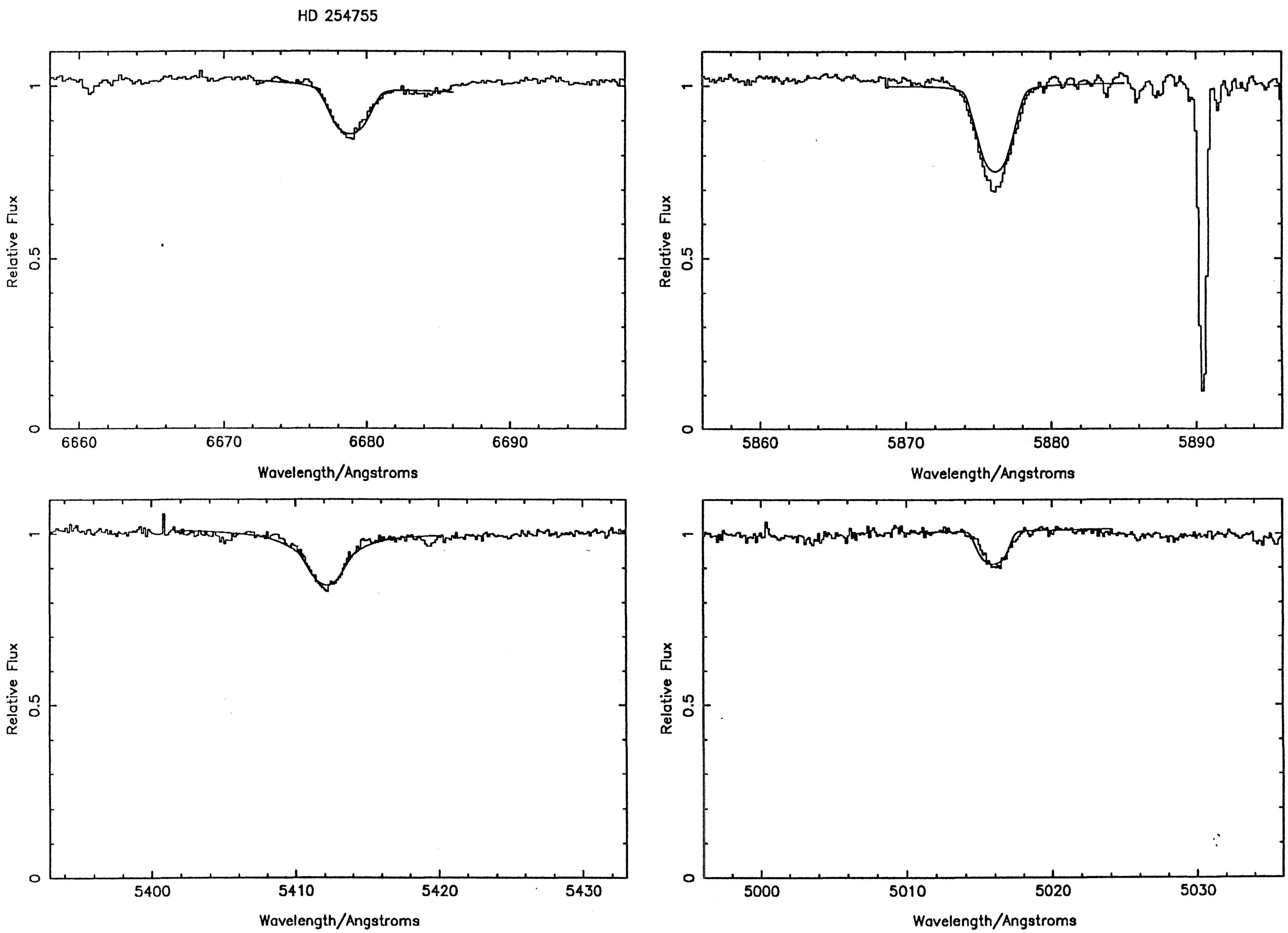

Figure 6. (Continued) HD 254755

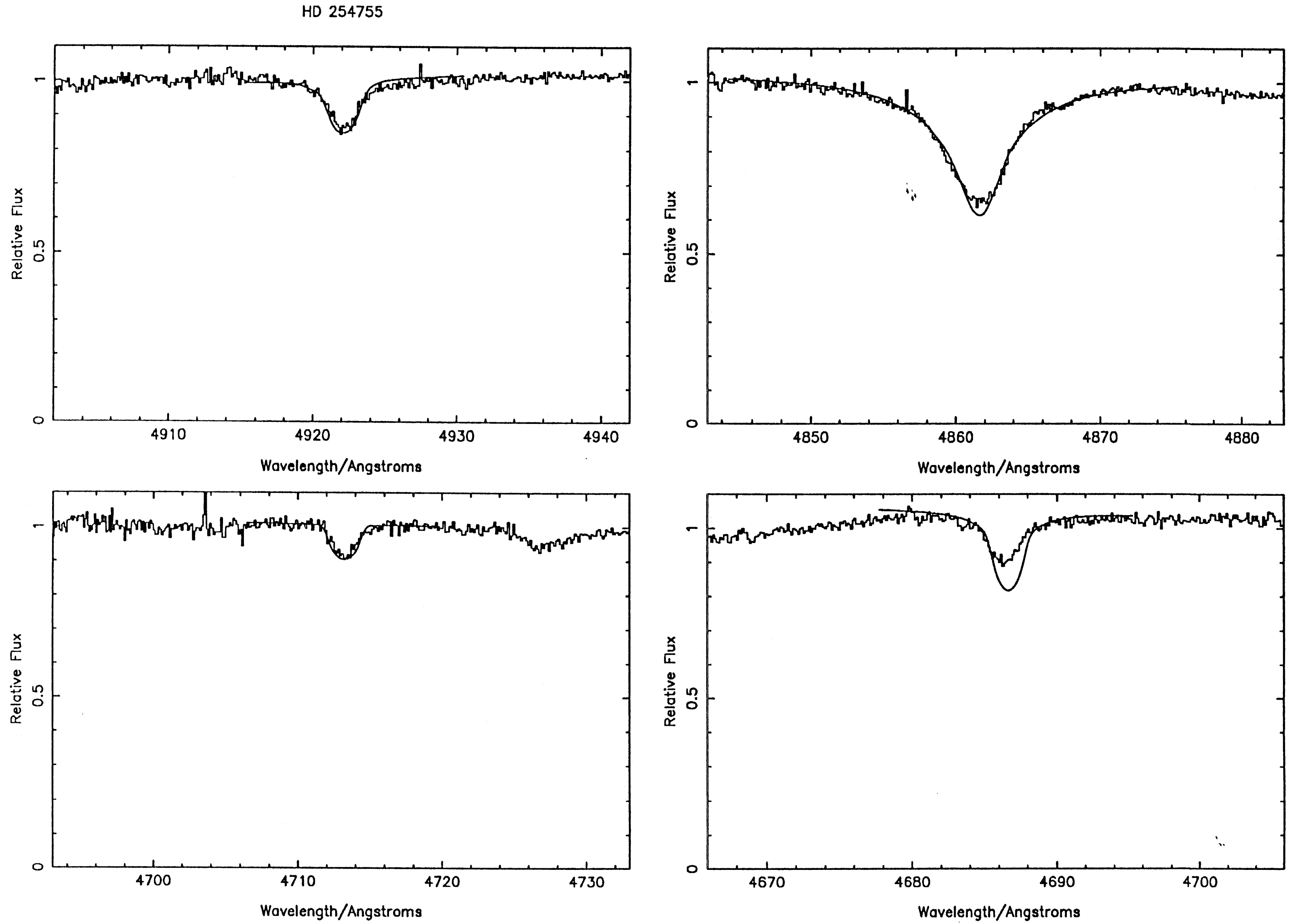

Figure 6. (Continued) HD 254755

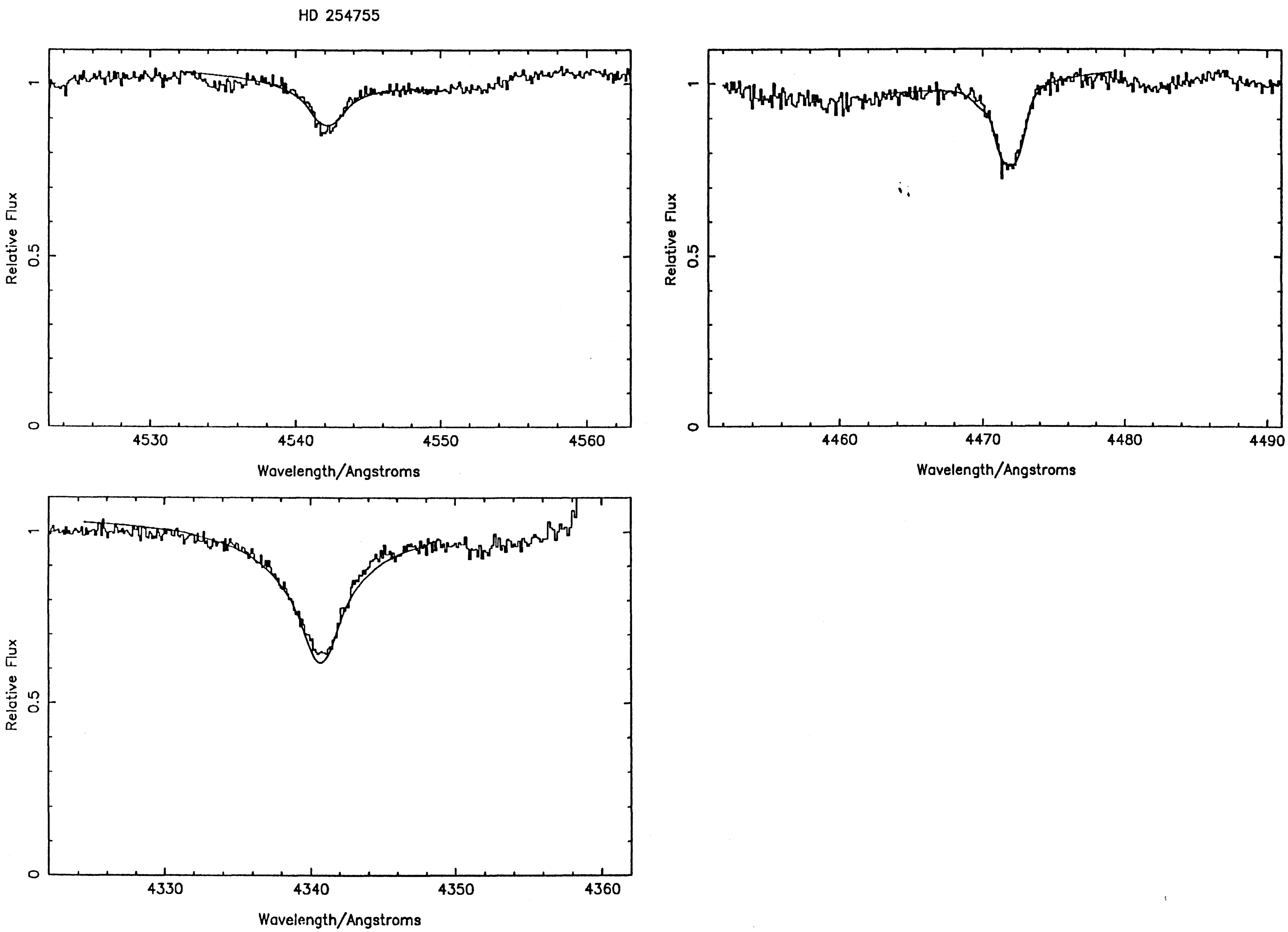

Figure 6. (Continued) HD 254755

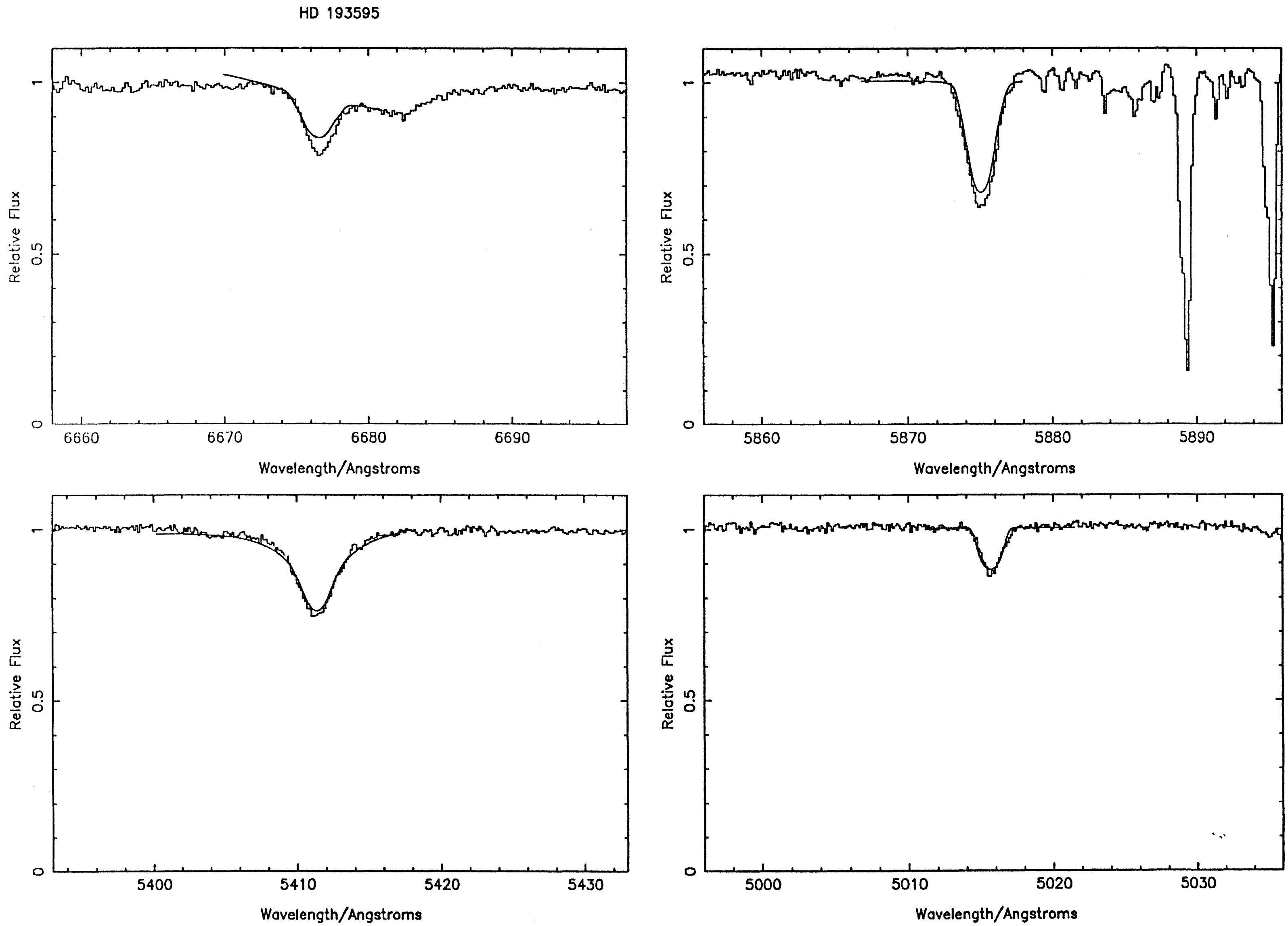

Figure 6. (Continued) HD 193595

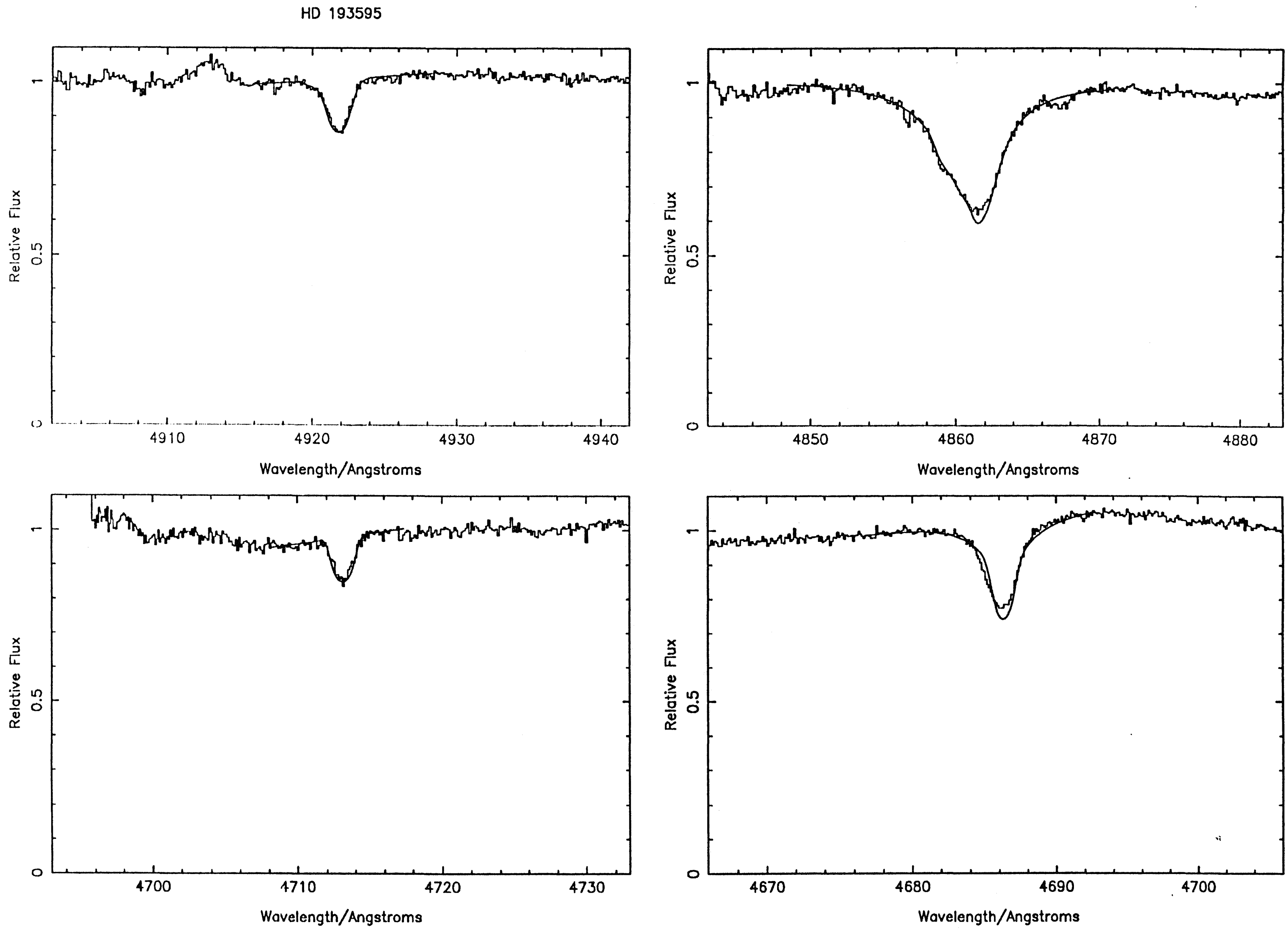

Figure 6. (Continued) HD 193595

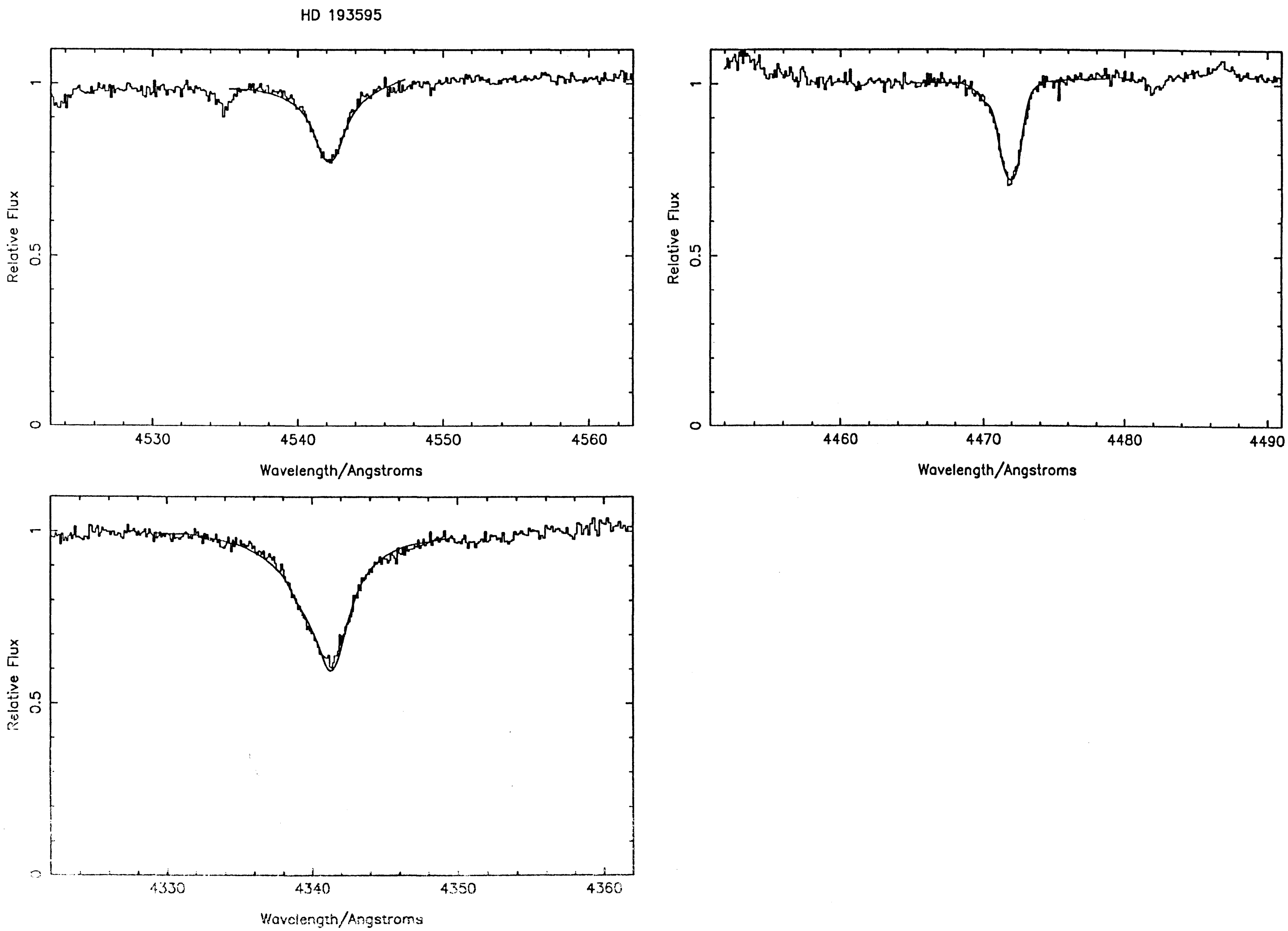

Figure 6. (Continued) HD 193595

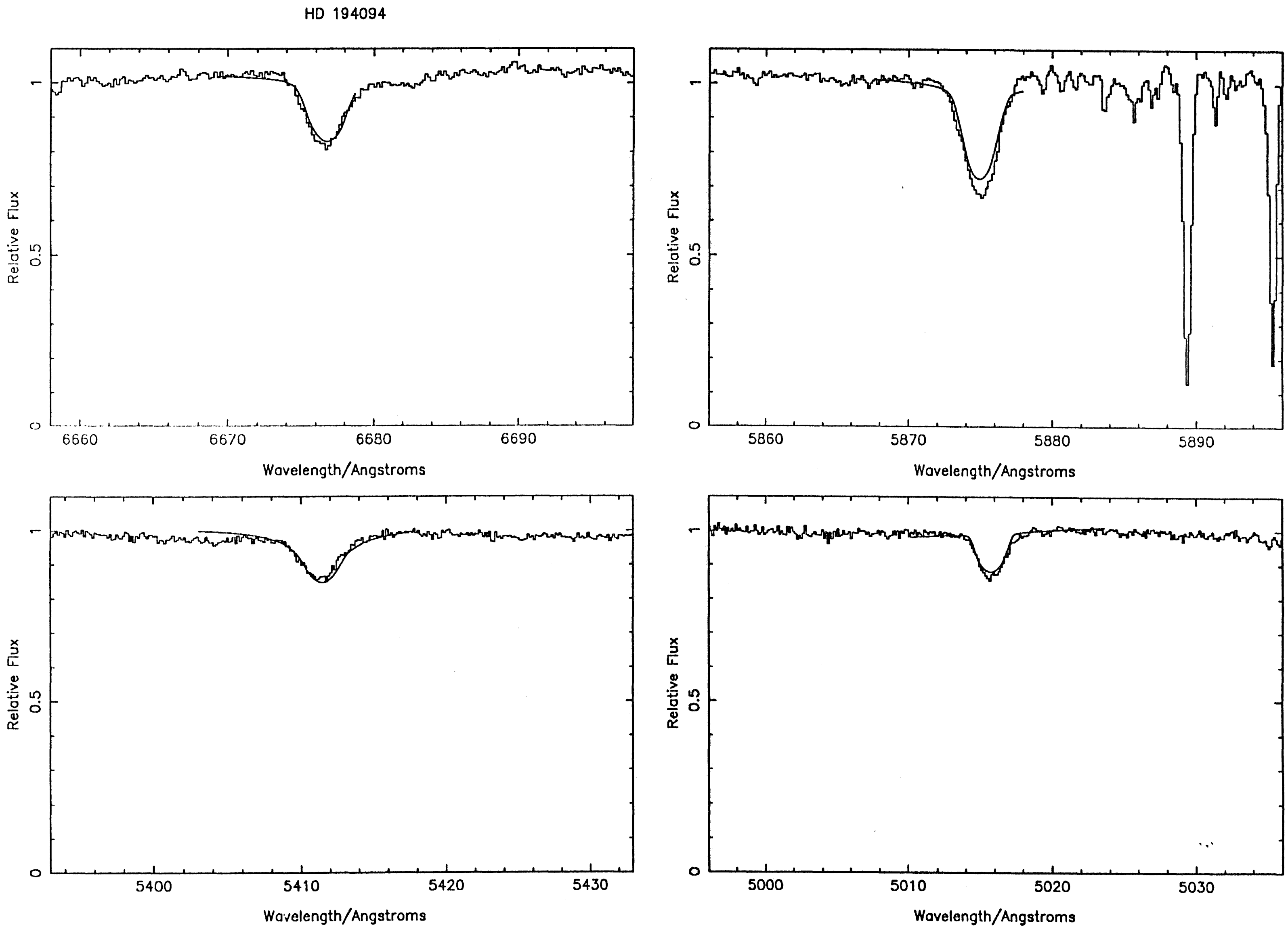

Figure 6. (Continued) HD 194094

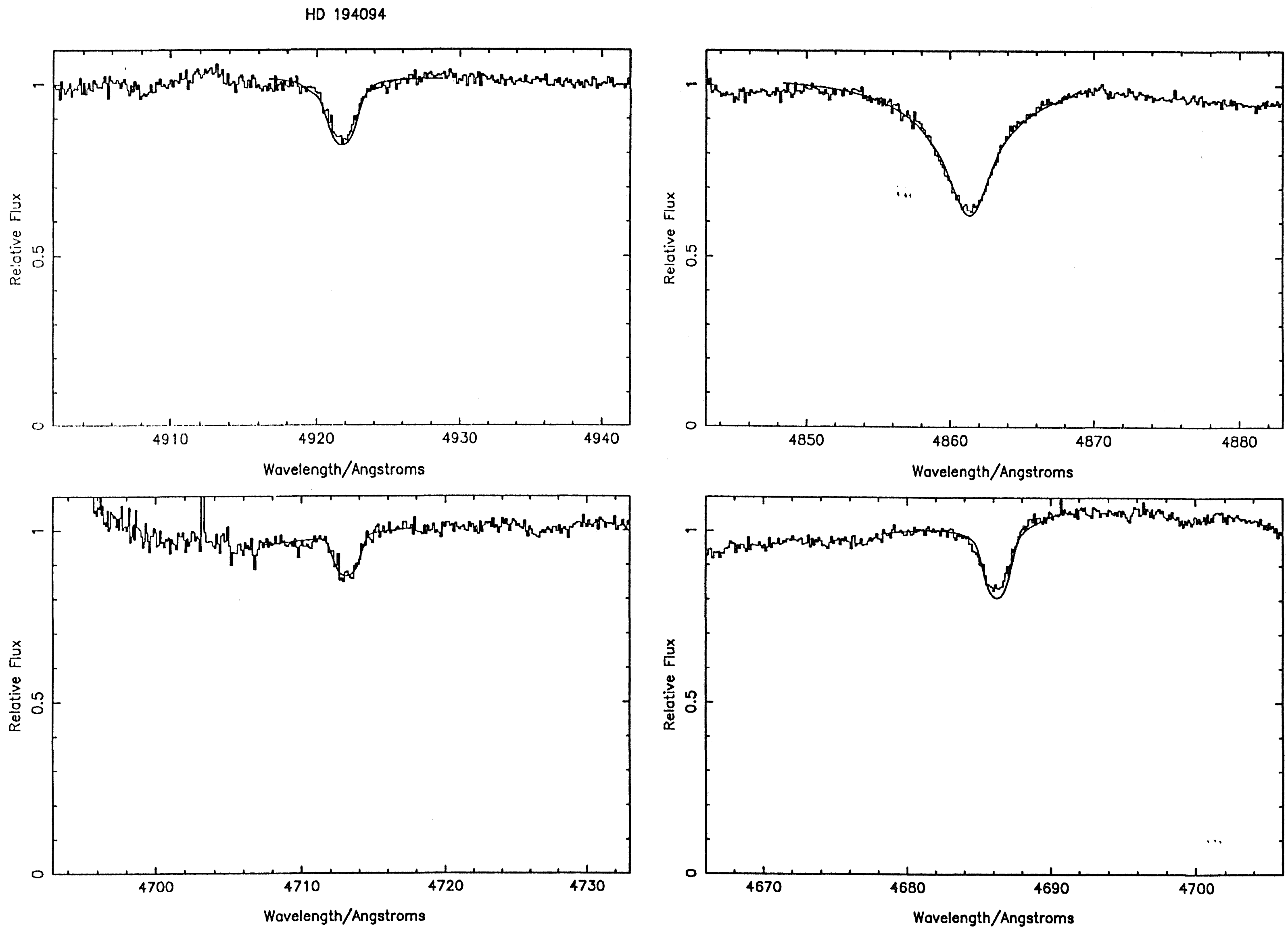

Figure 6. (Continued) HD 194094

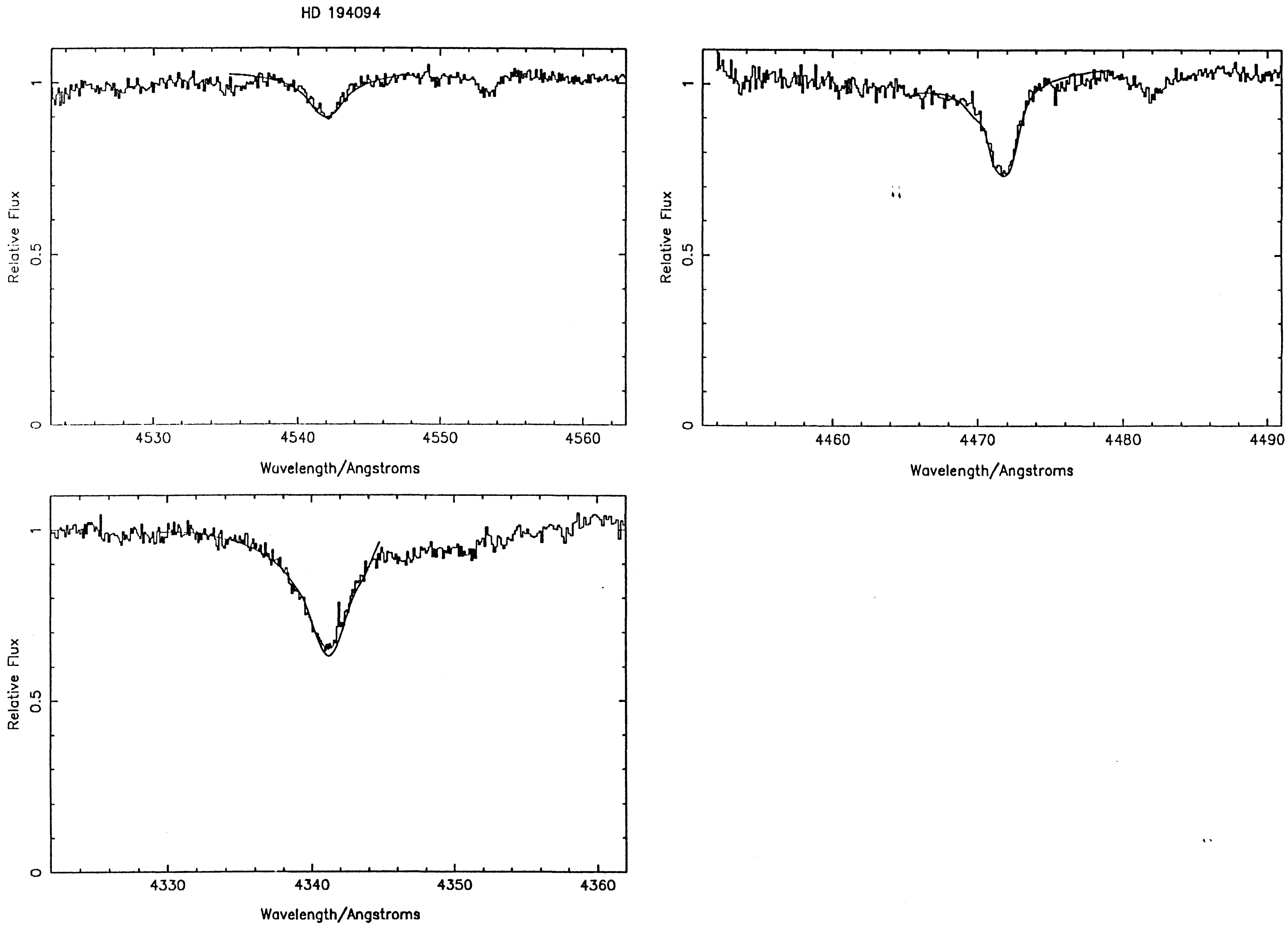

Figure 6. (Continued) HD 194094

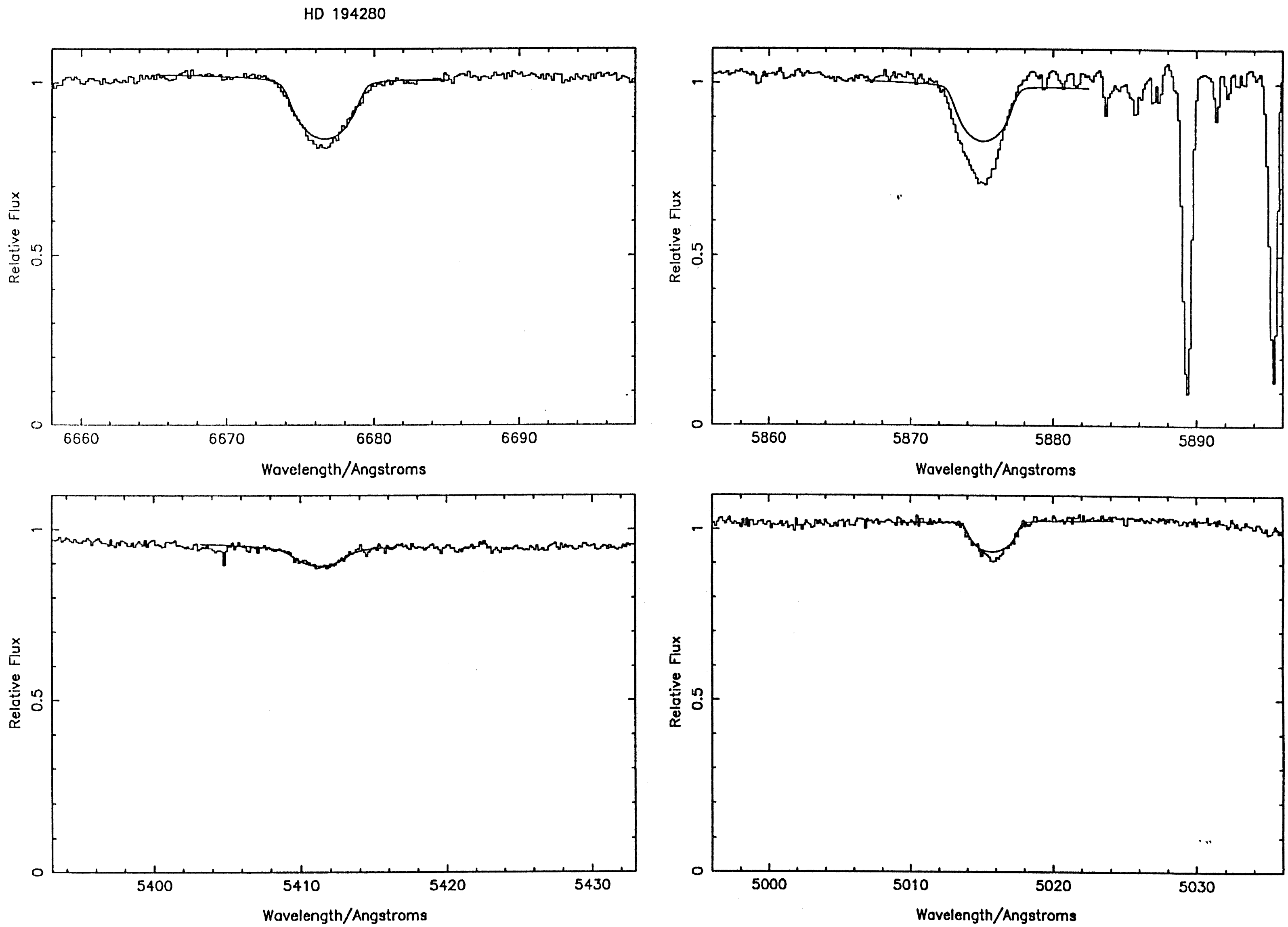

Figure 6. (Continued) HD 194280

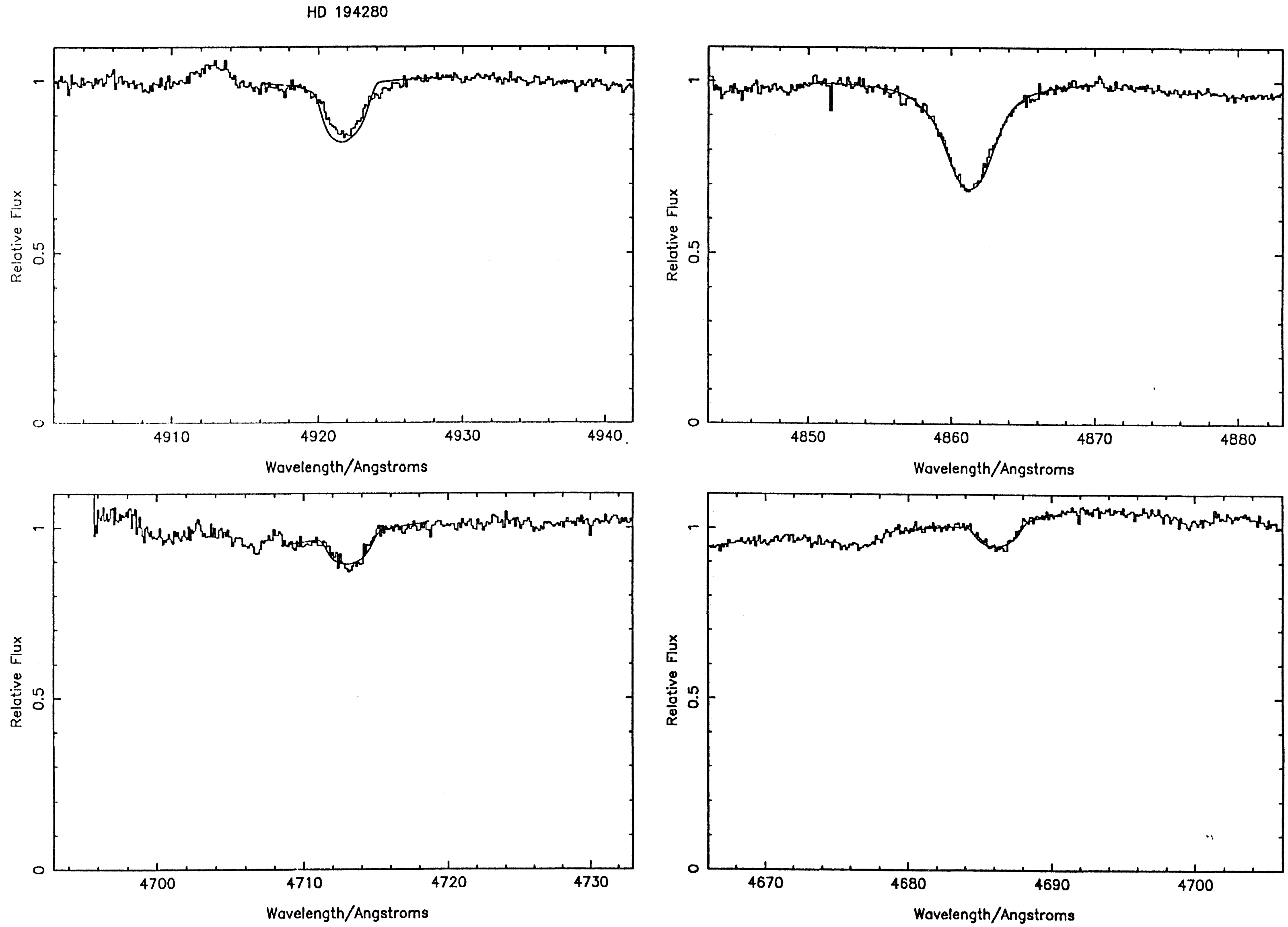

Figure 6. (Continued) HD 194280

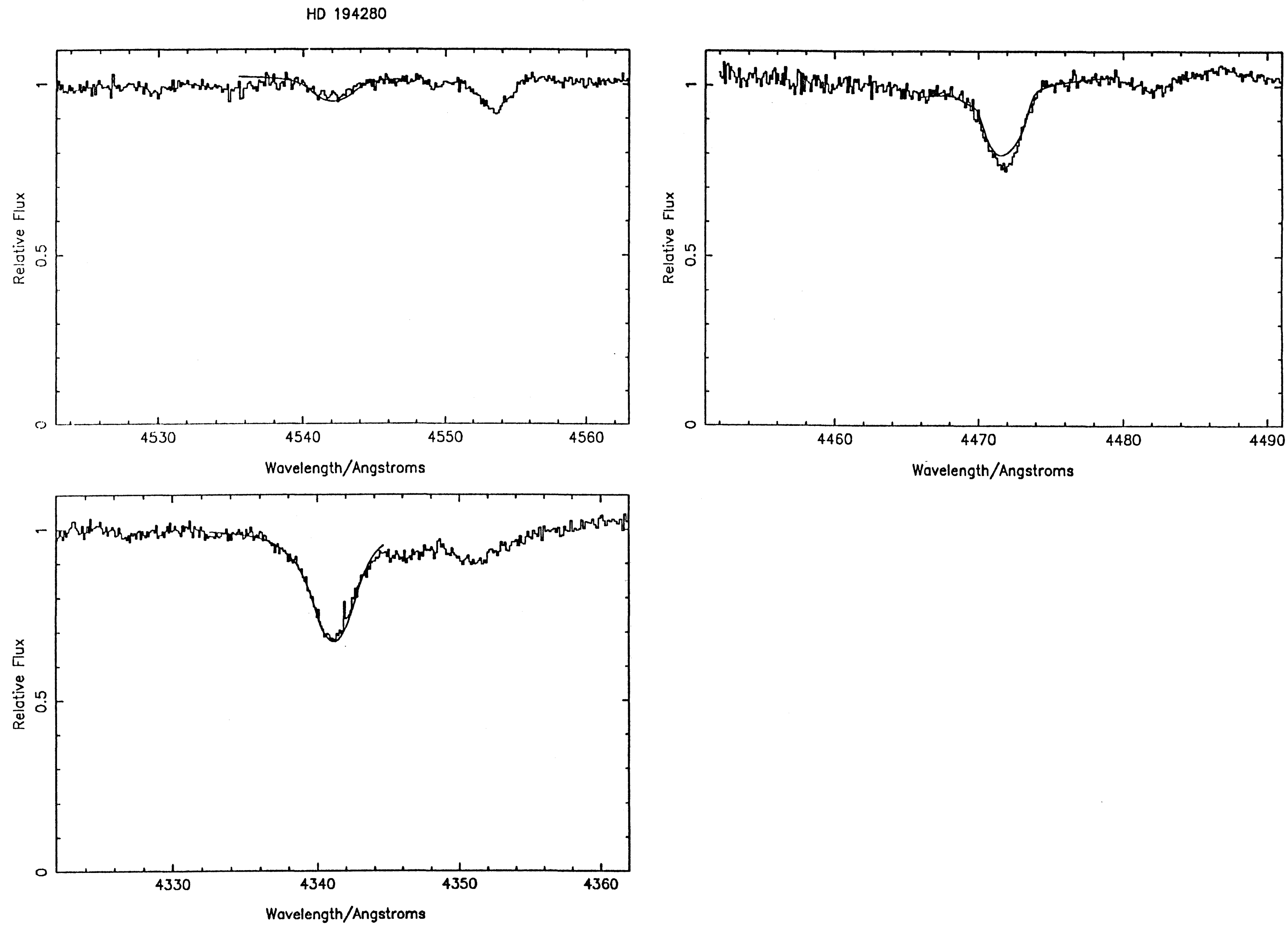

Figure 6. (Continued) HD 194280

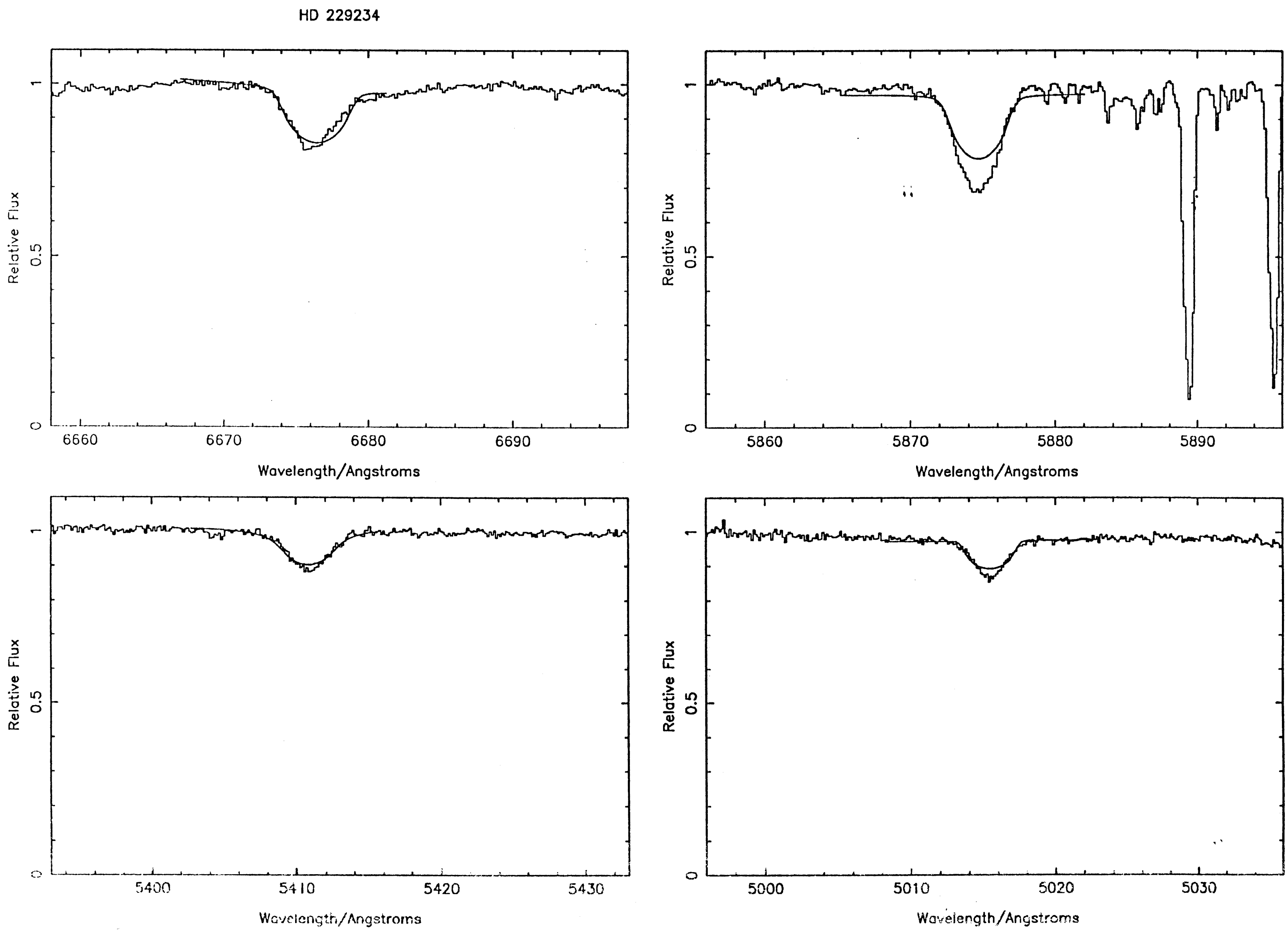

Figure 6. (Continued) HD 229234

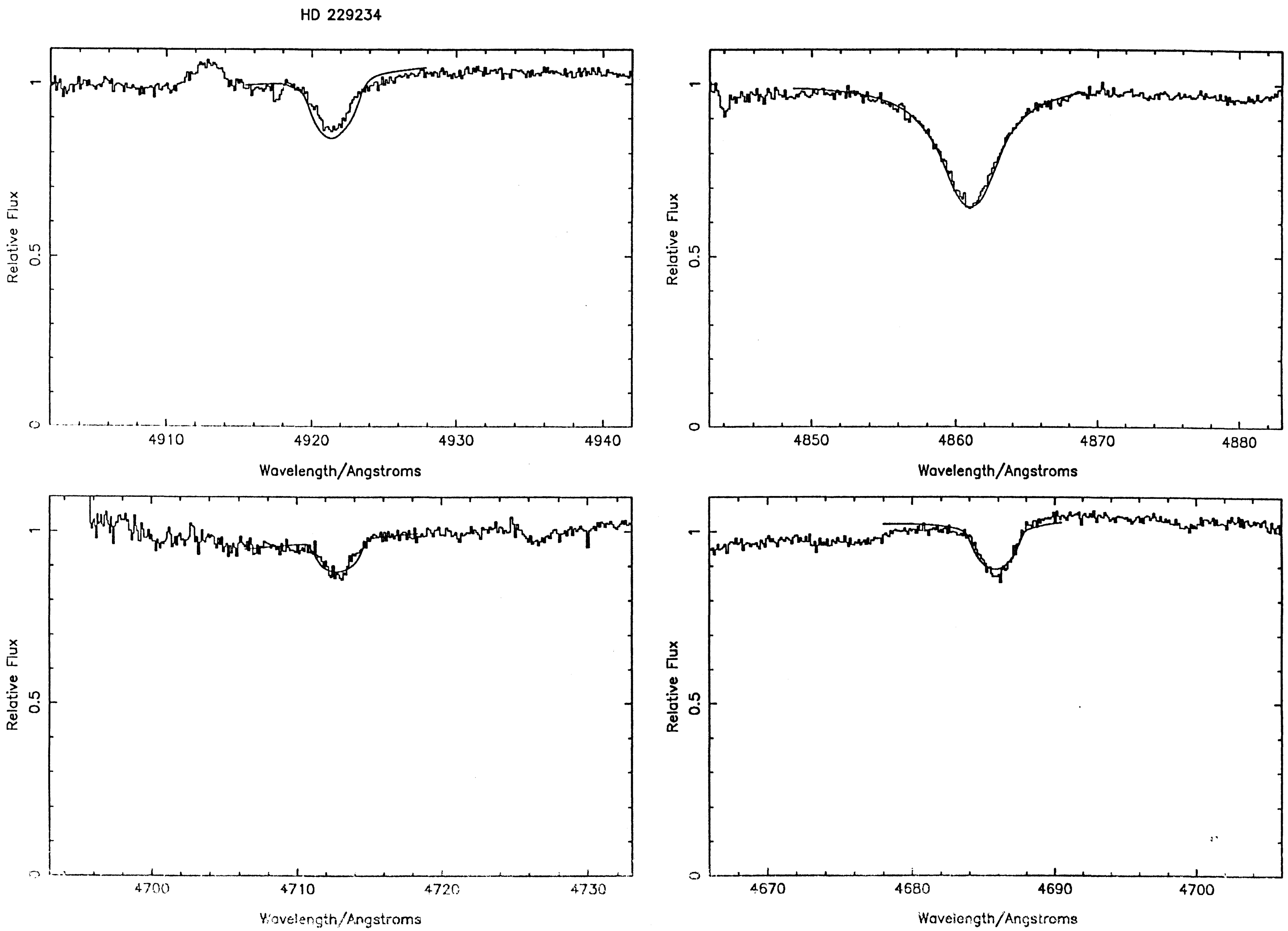

Figure 6. (Continued) HD 229234

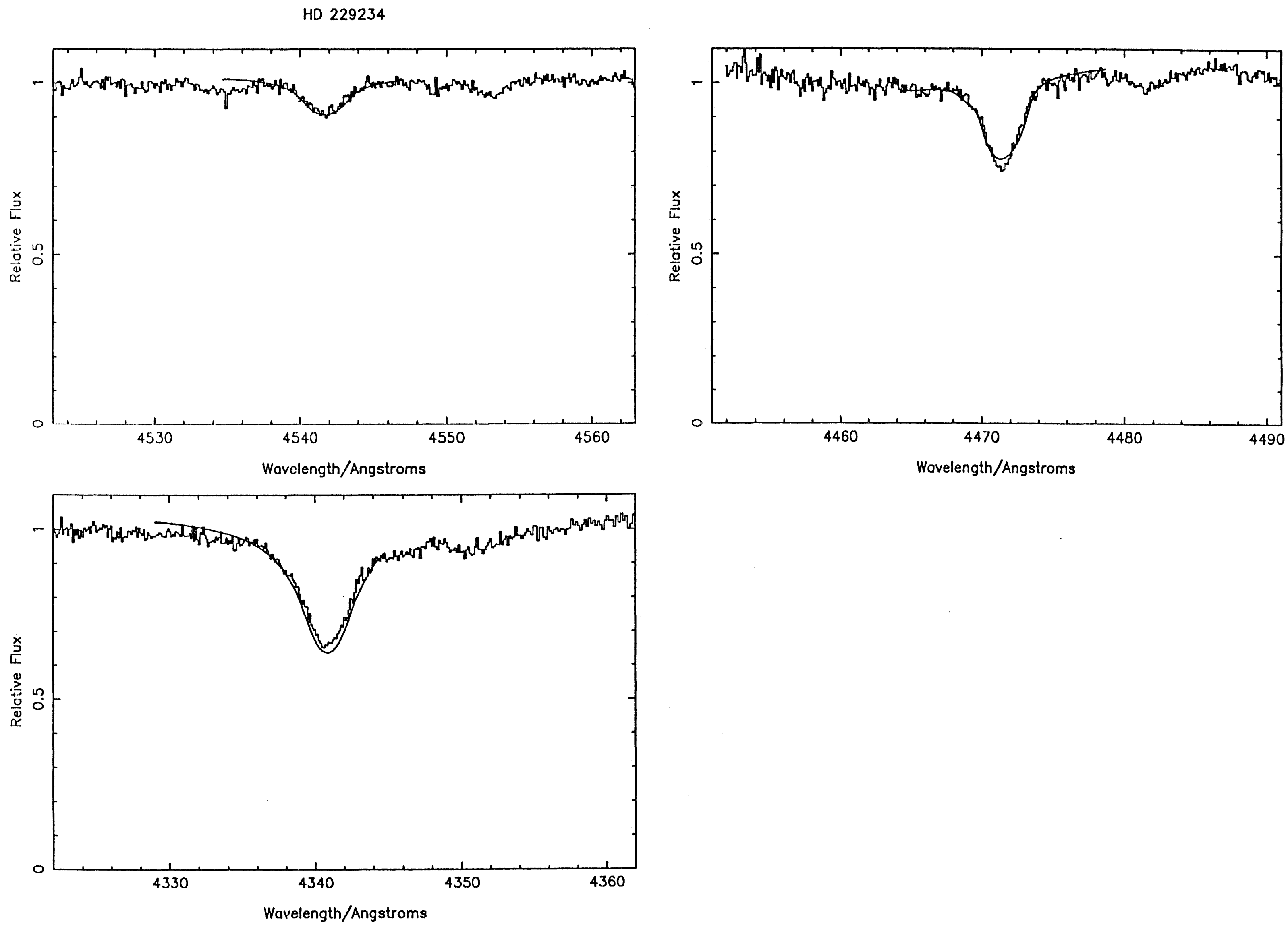

Figure 6. (Continued) HD 229234

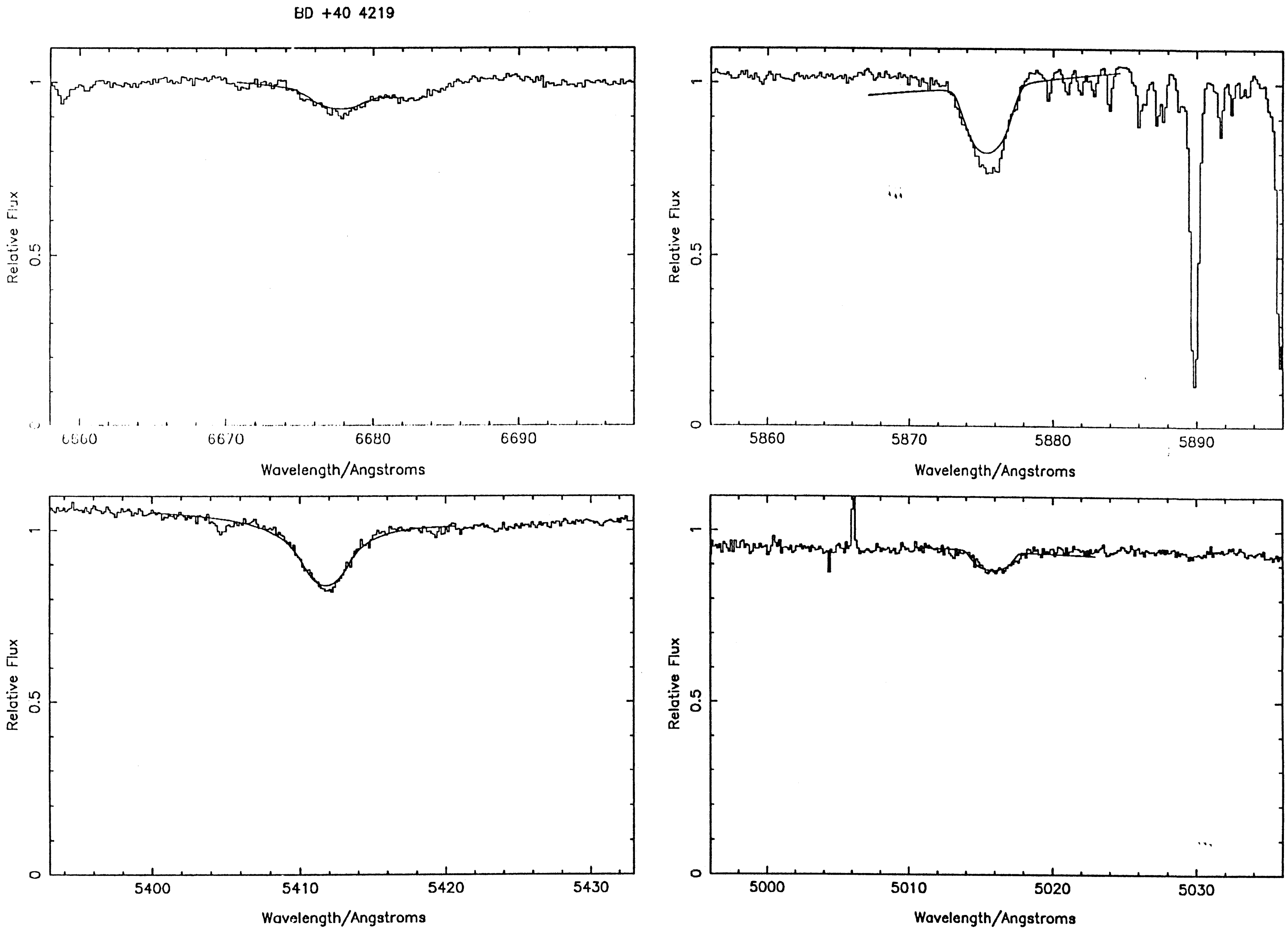

Figure 6. (Continued) #4 OB2 Cyg (BD +40 4219)

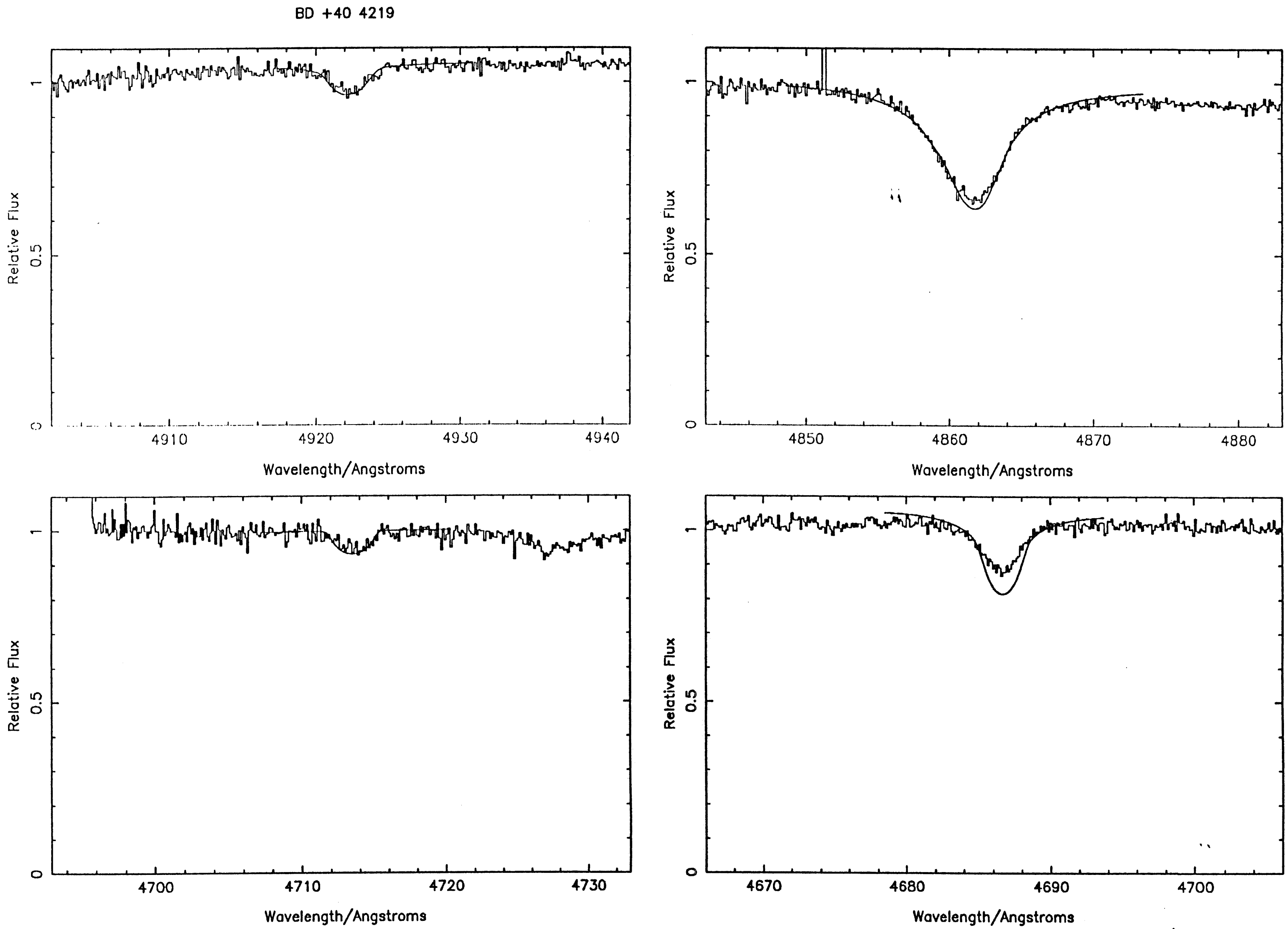

Figure 6. (Continued) #4 OB2 Cyg

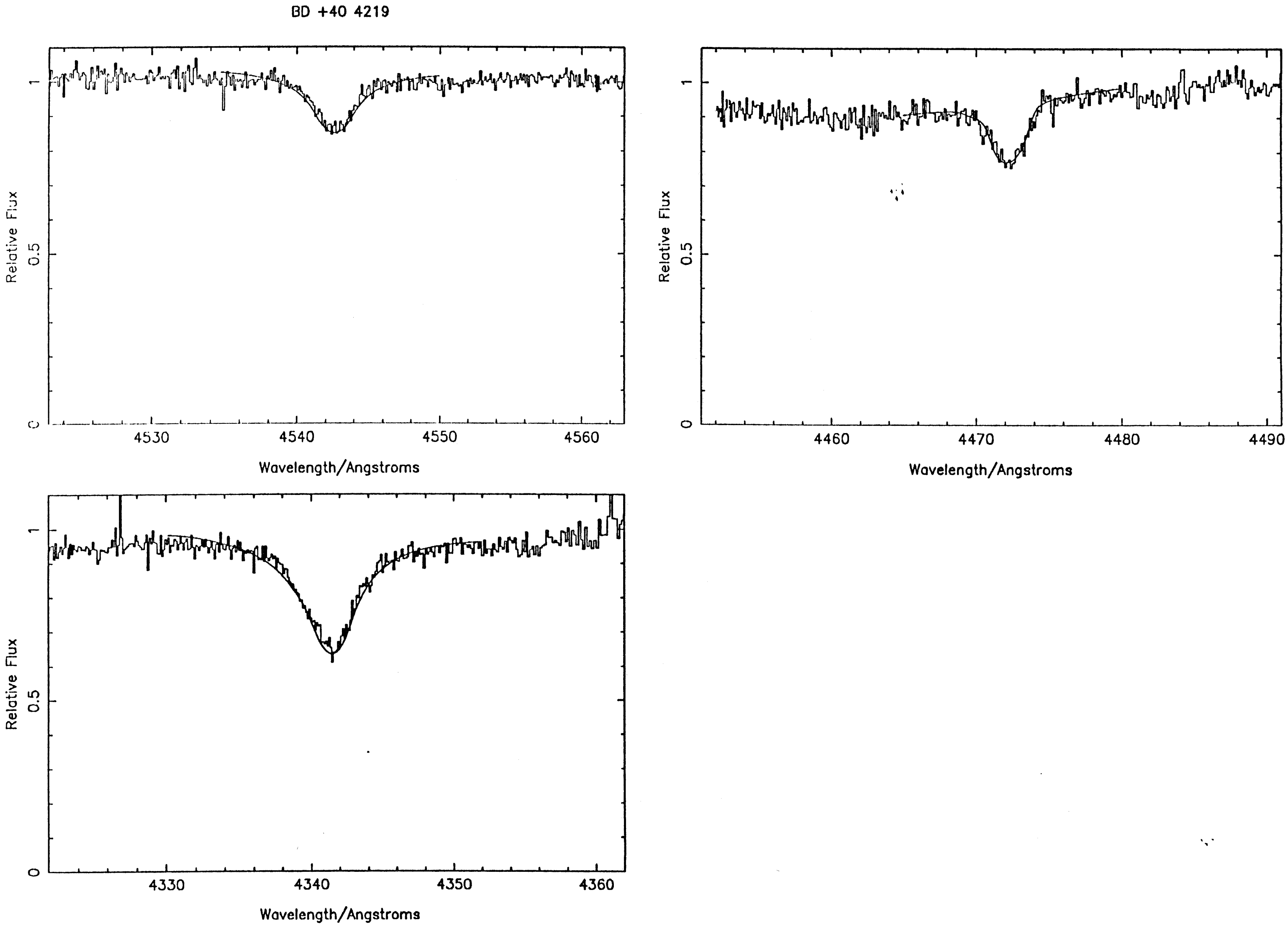

Figure 6. (Continued) #4 OB2 Cyg

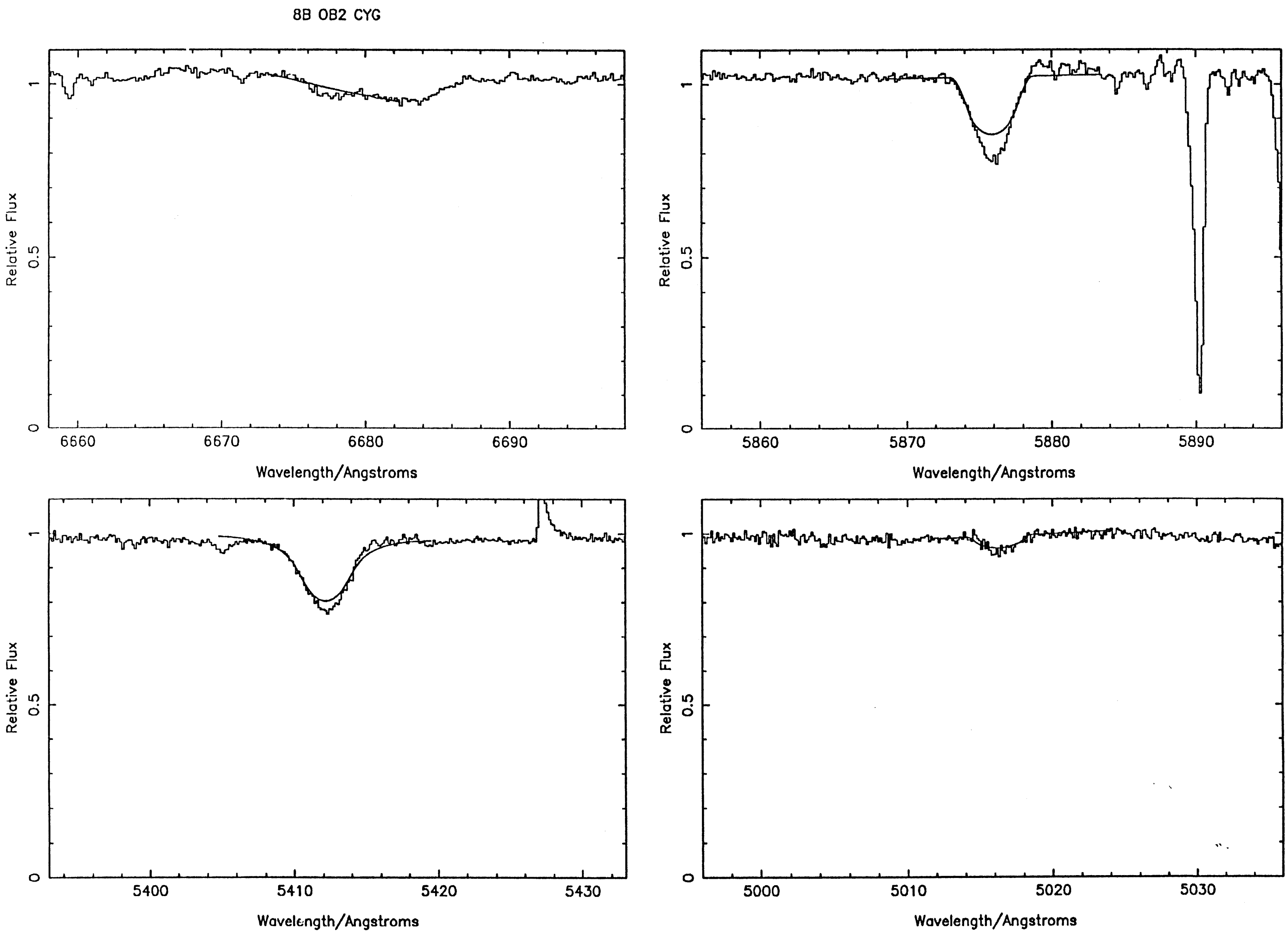

Figure 6. (Continued) #8B OB2 Cyg

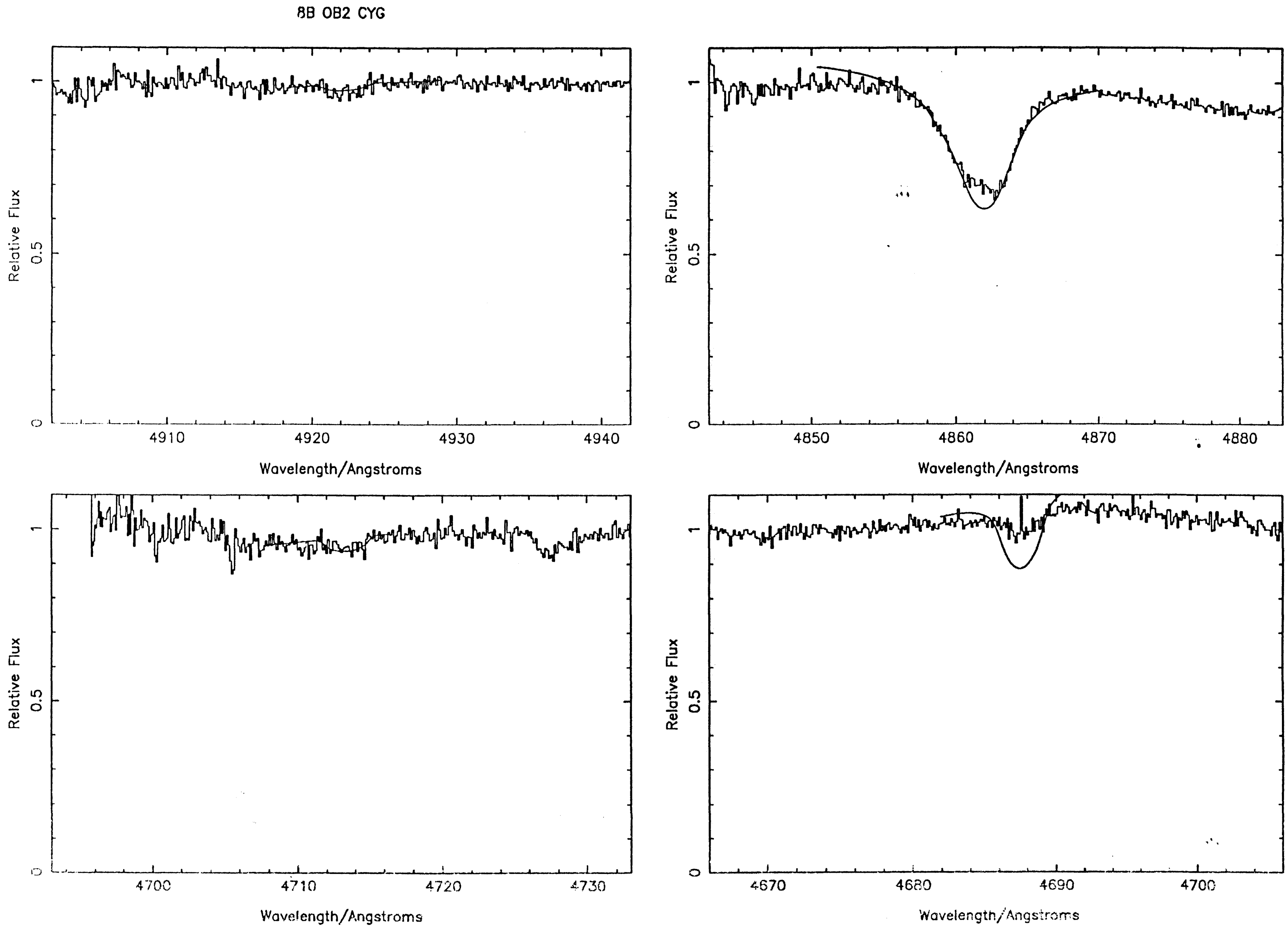

Figure 6. (Continued) #8B OB2 Cyg

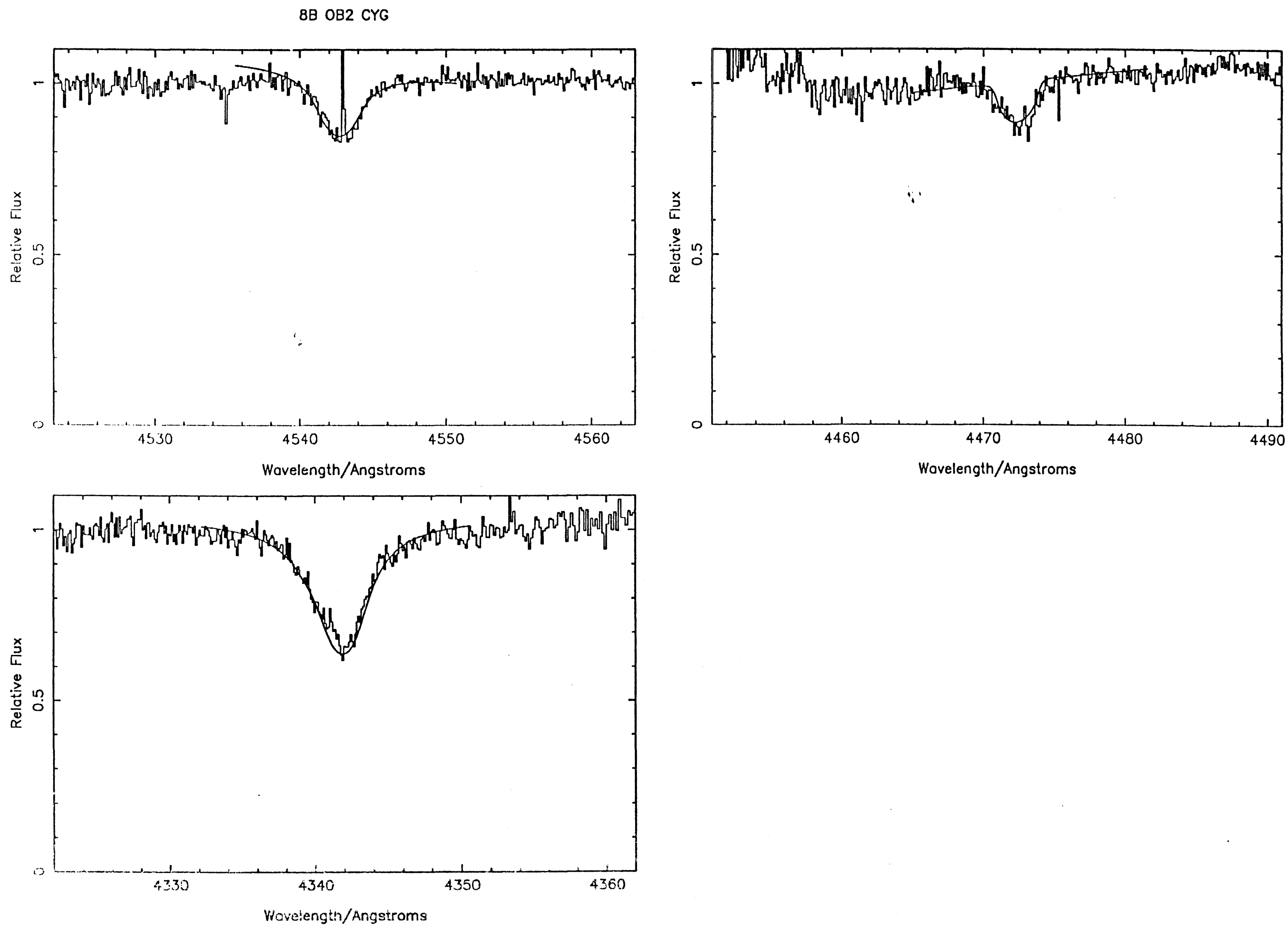

Figure 6. (Continued) #8B OB2 Cyg

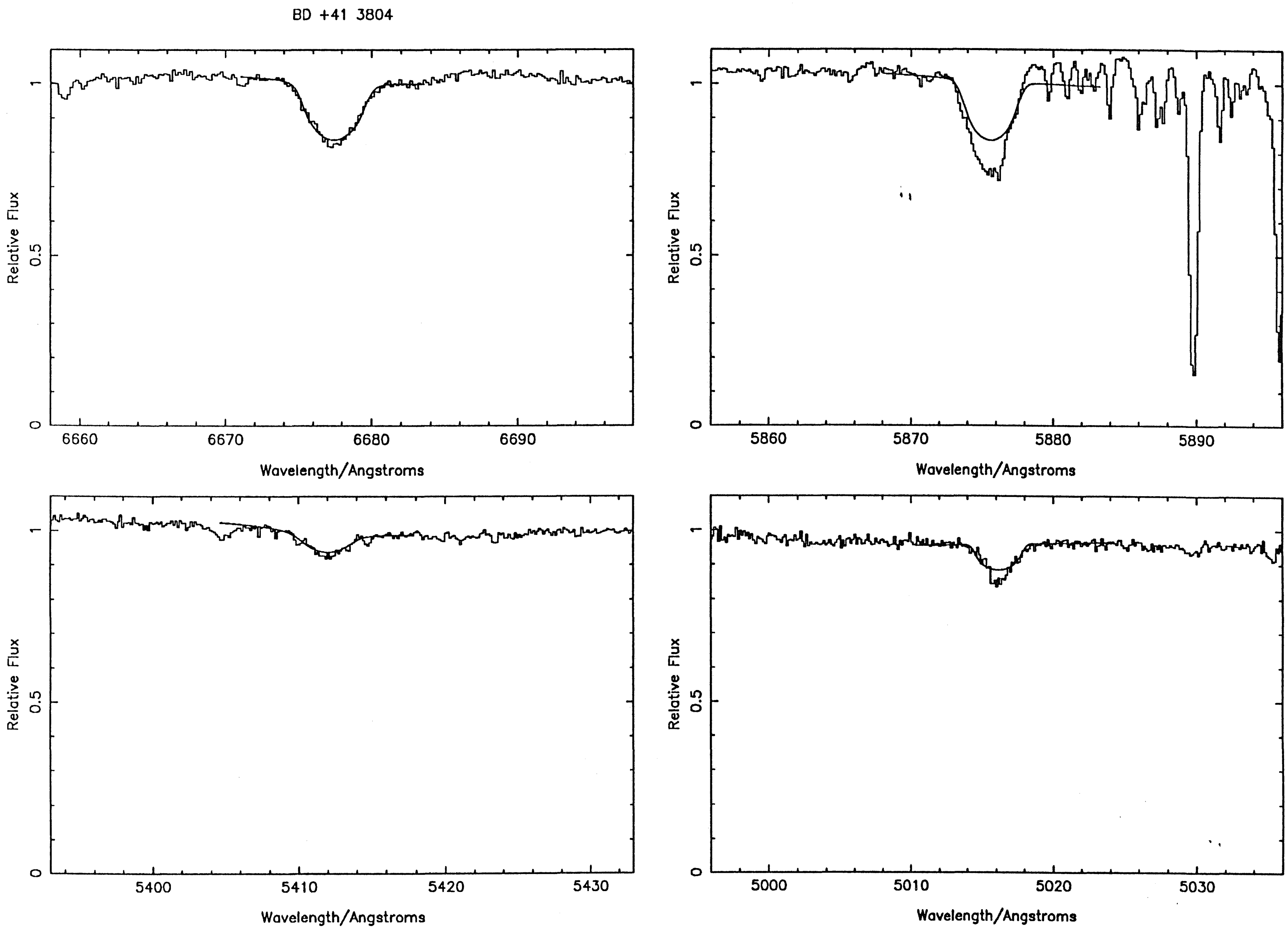

Figure 6. (Continued) #10 OB2 Cyg (BD +41 3804)

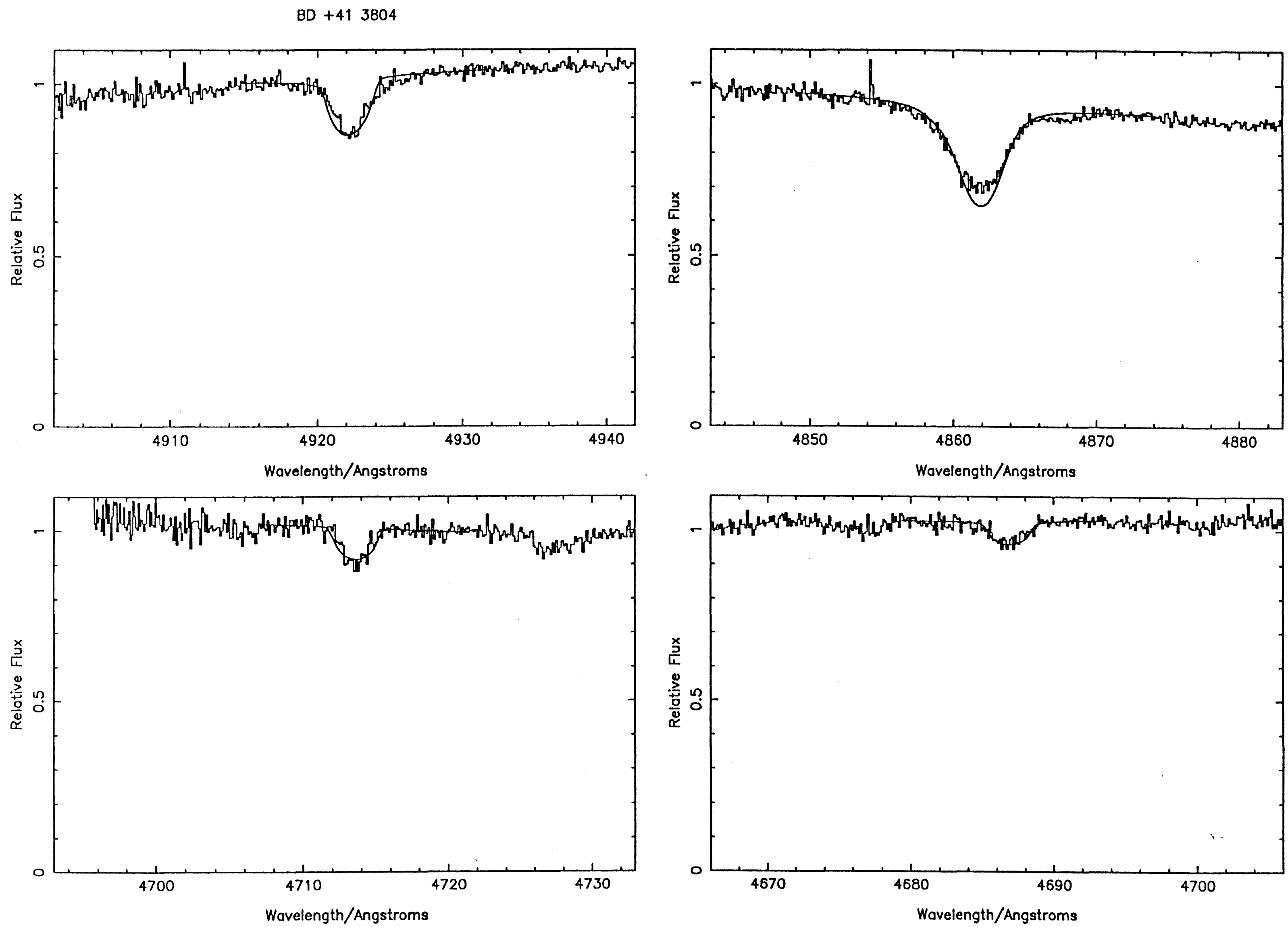

Figure 6. (Continued) #10 OB2 Cyg

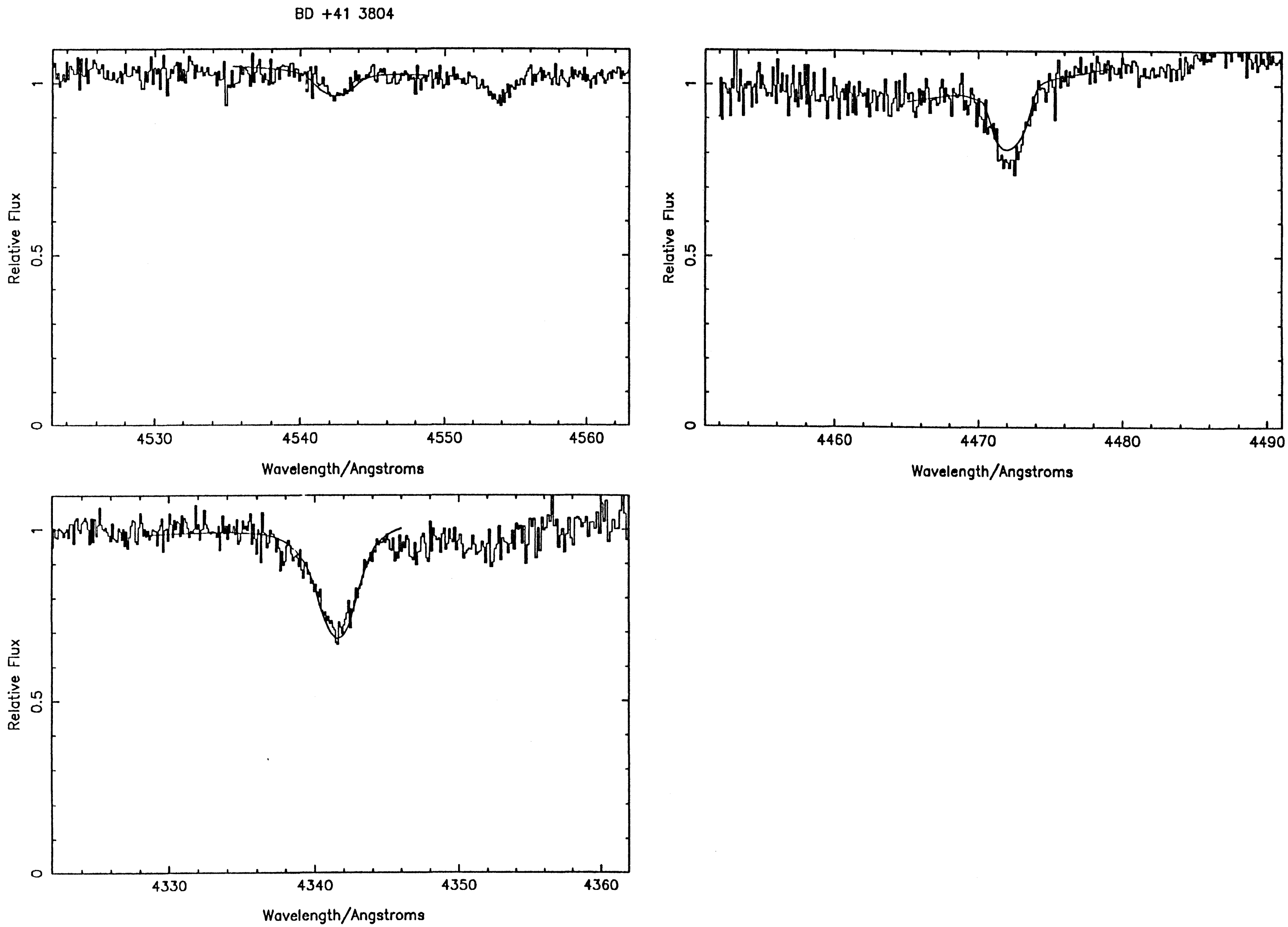

Figure 6. (Continued) #10 OB2 Cyg

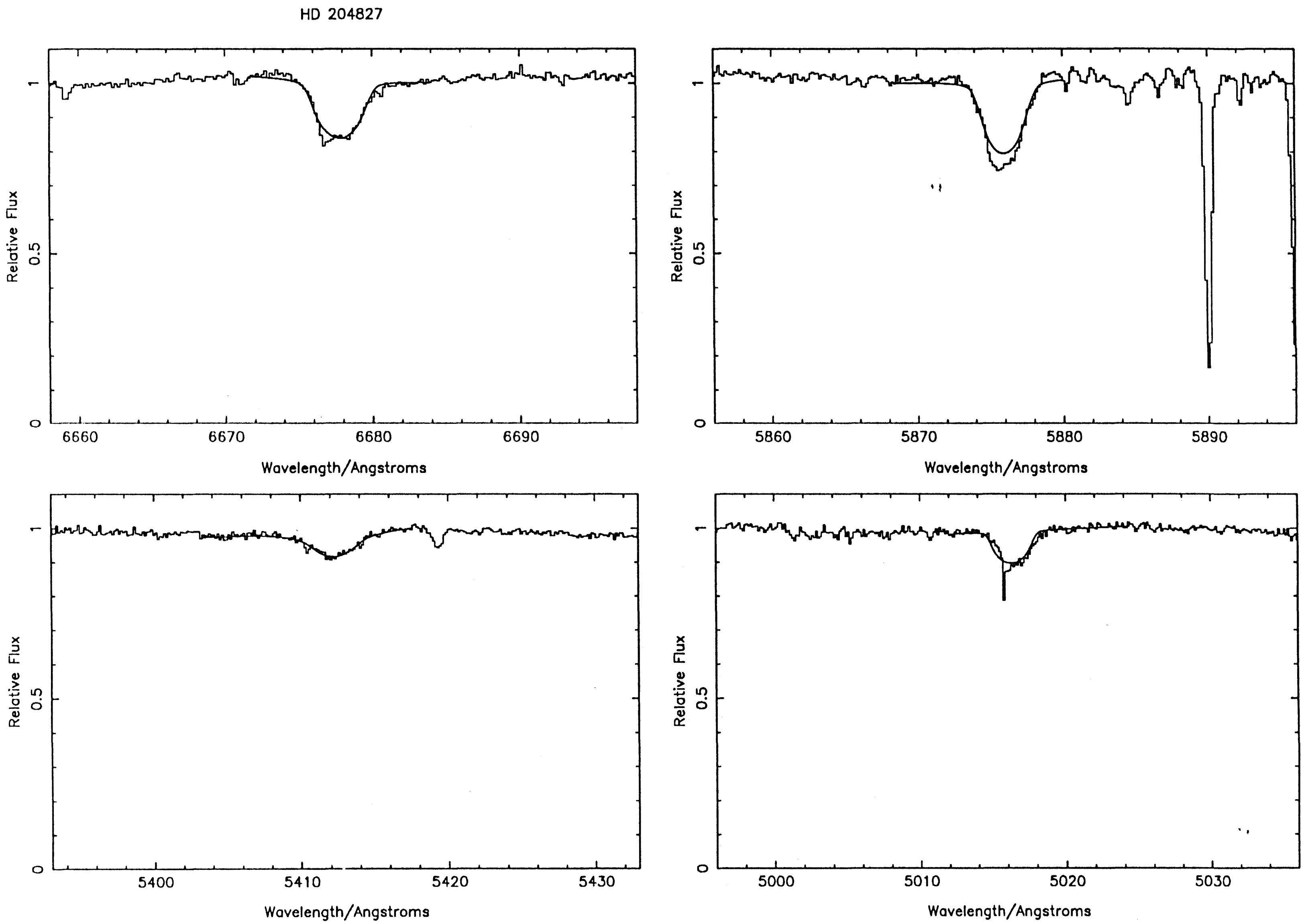

Figure 6. (Continued) HD 204827

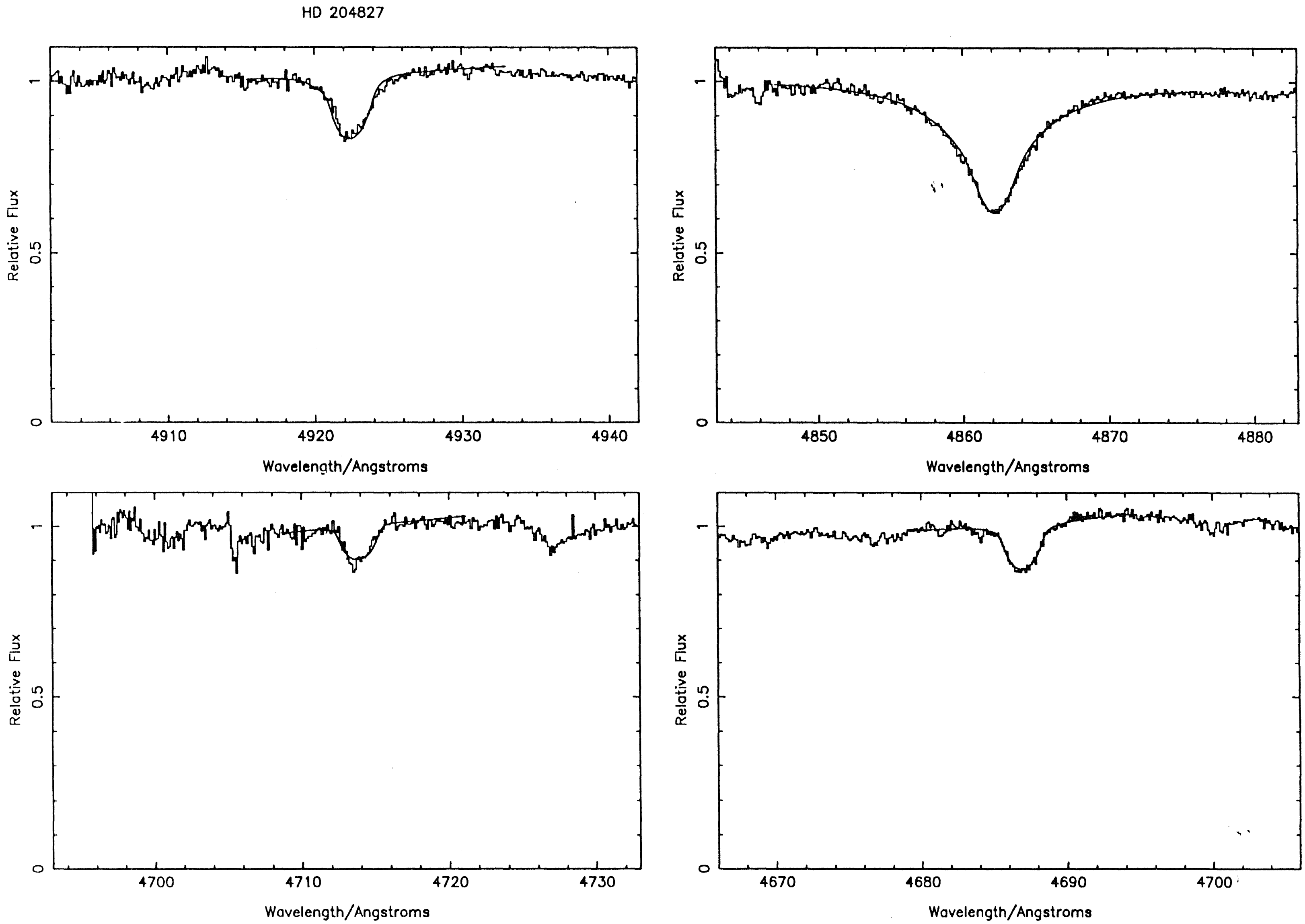

Figure 6. (Continued) HD 204827

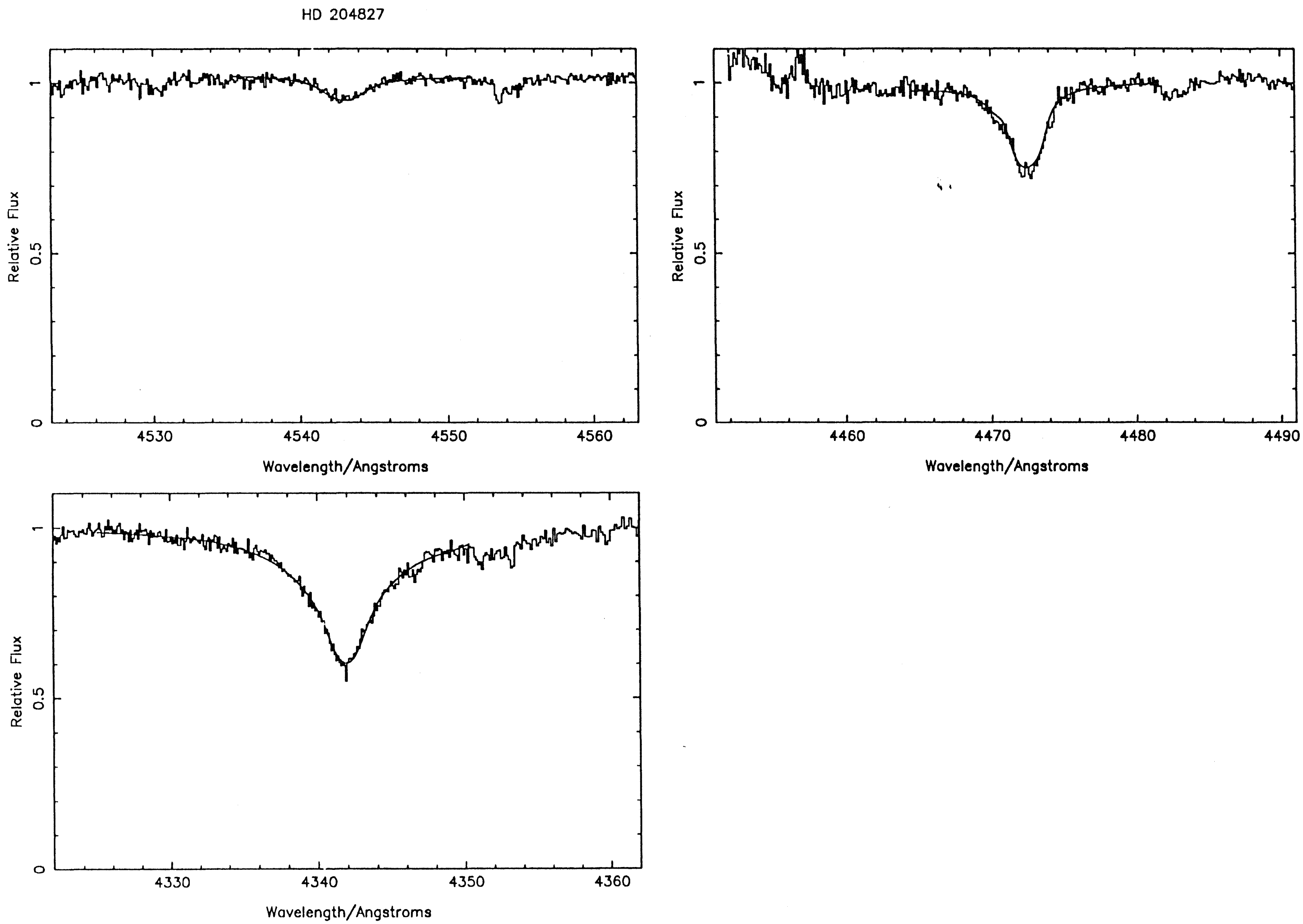

Figure 6. (Continued) HD 204827

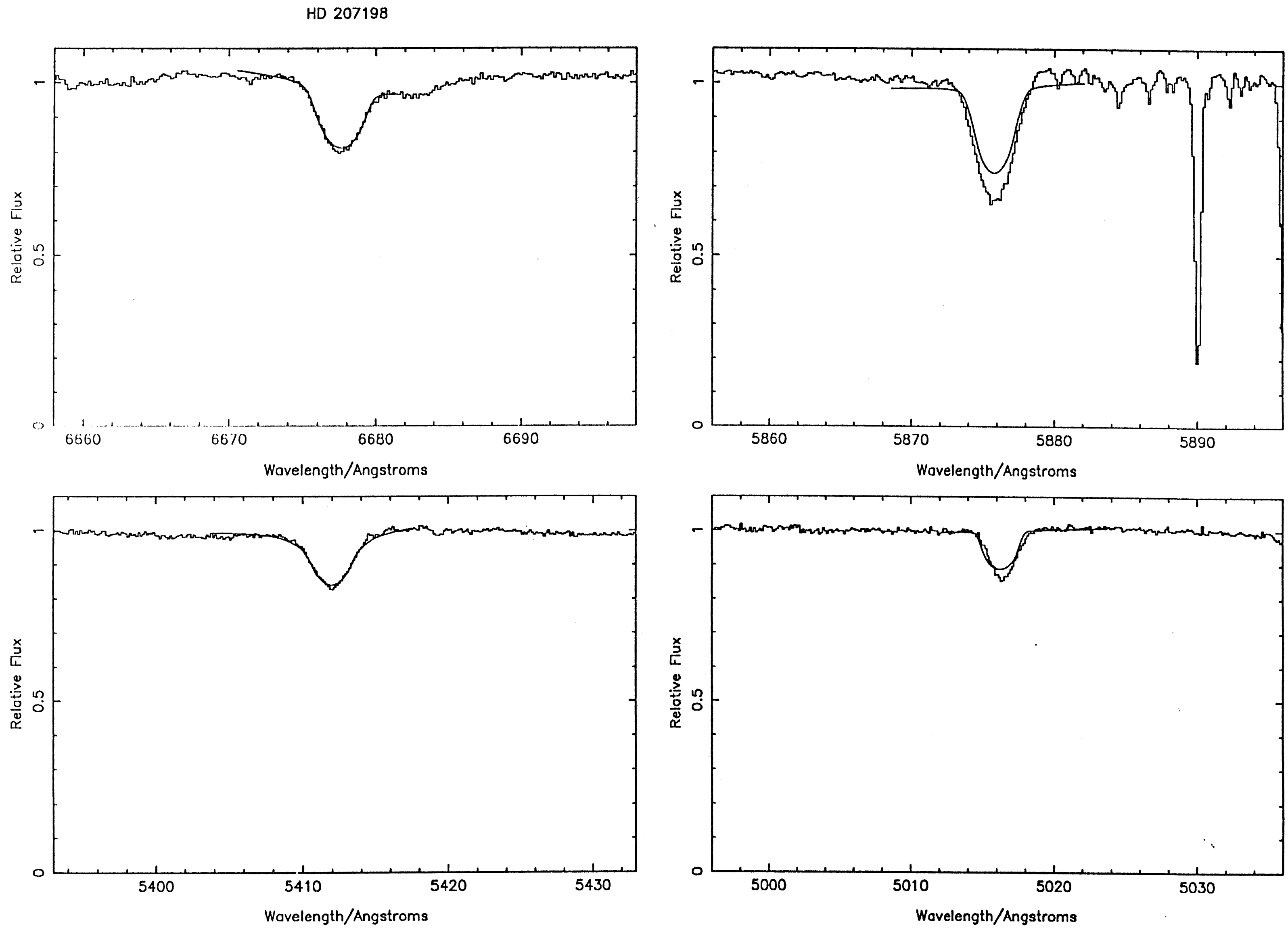

Figure 6. (Continued) HD 207198

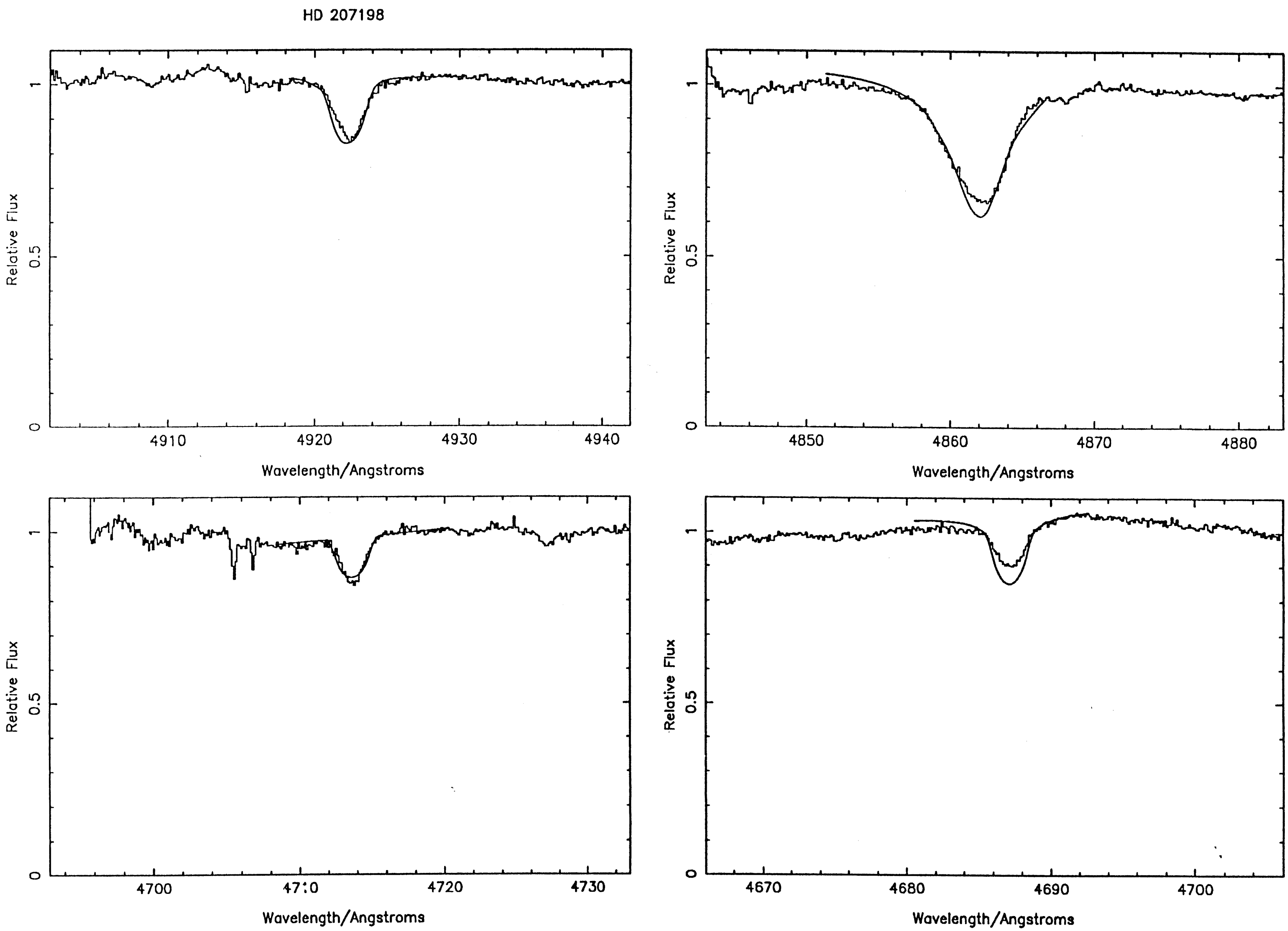

Figure 6. (Continued) HD 207198

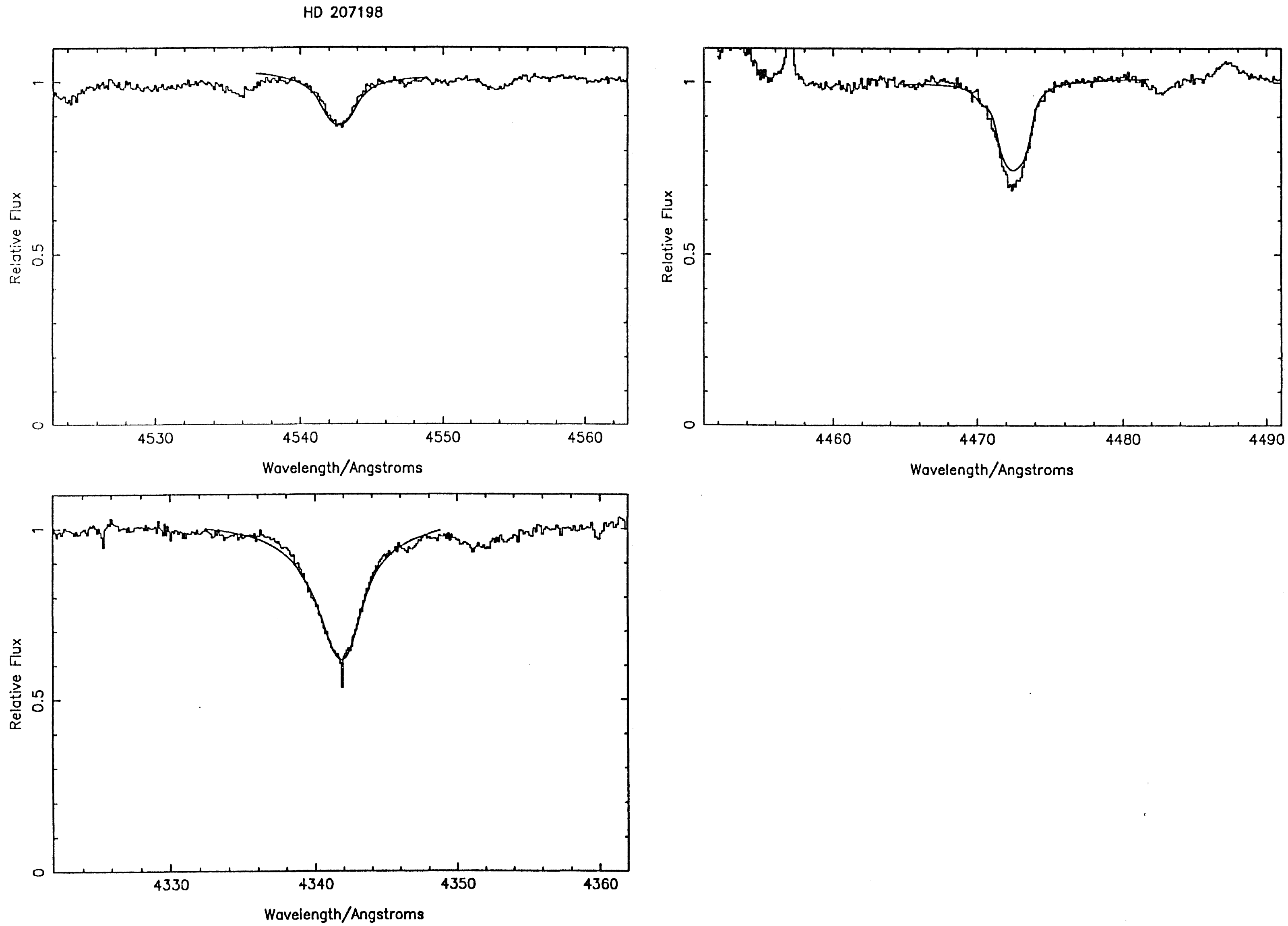

Figure 6. (Continued) HD 207198

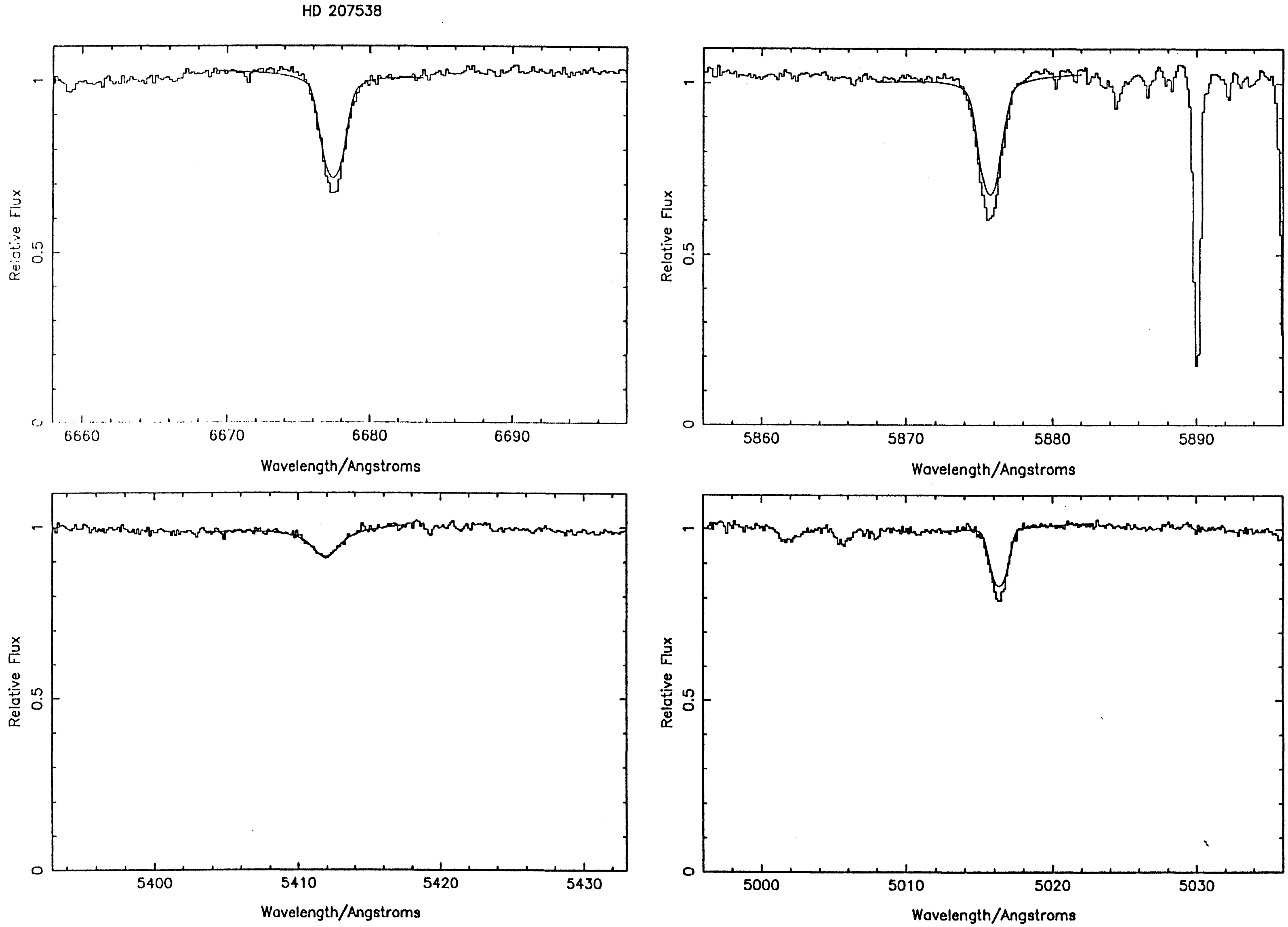

Figure 6. (Continued) HD 207538

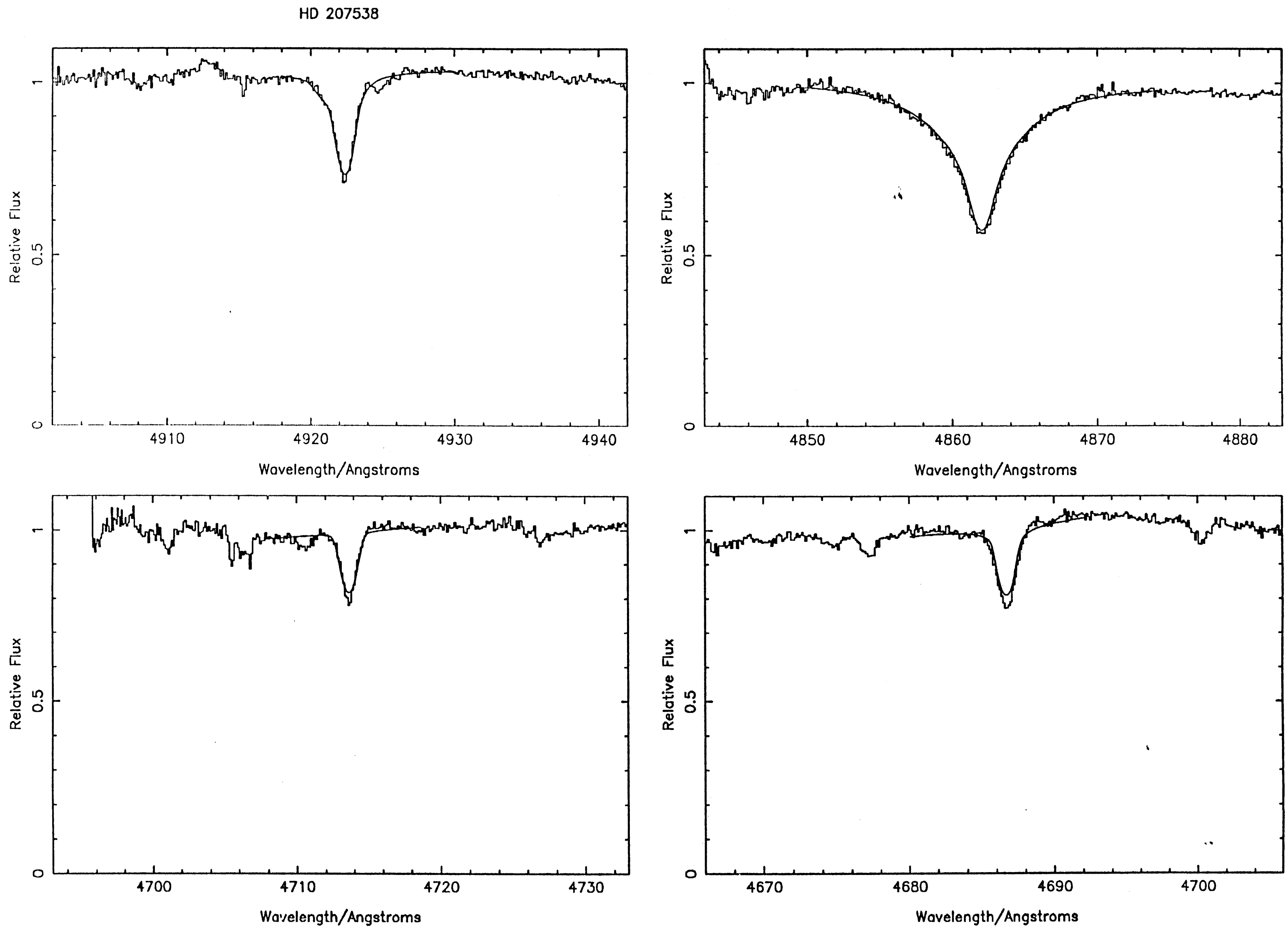

Figure 6. (Continued) HD 207538

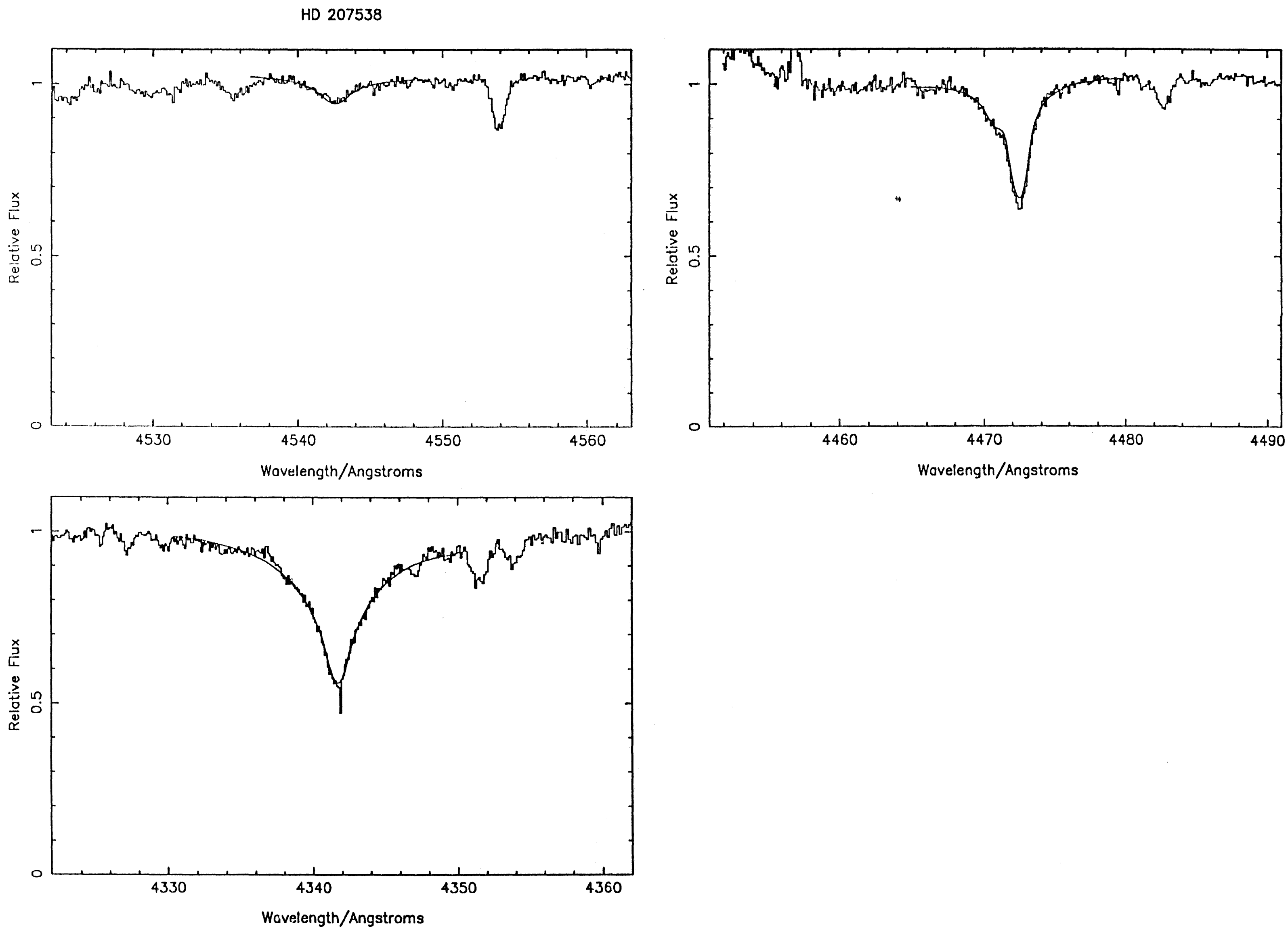

Figure 6. (Continued) HD 207538

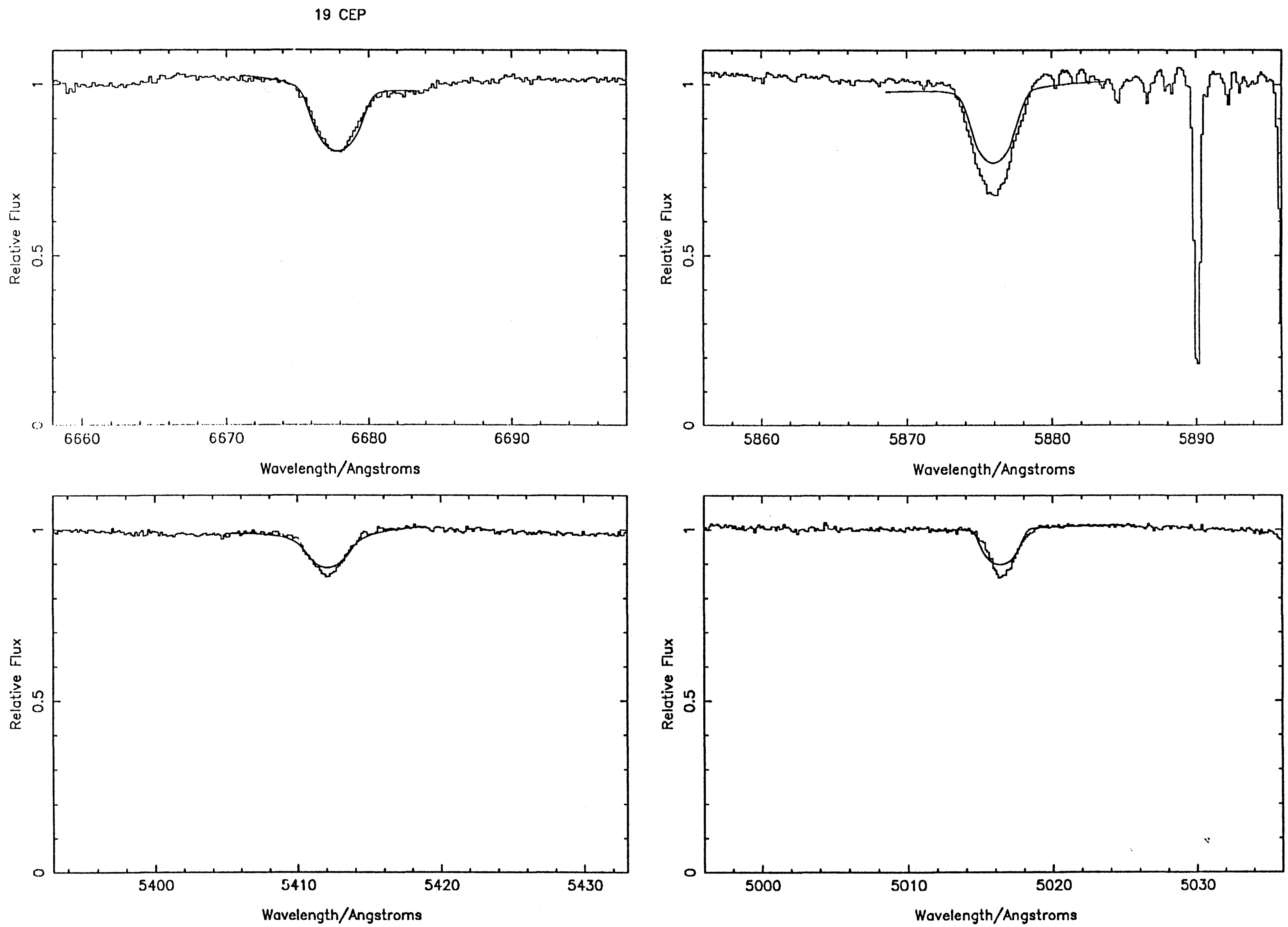

Figure 6. (Continued) HD 209975 (19 Cep)

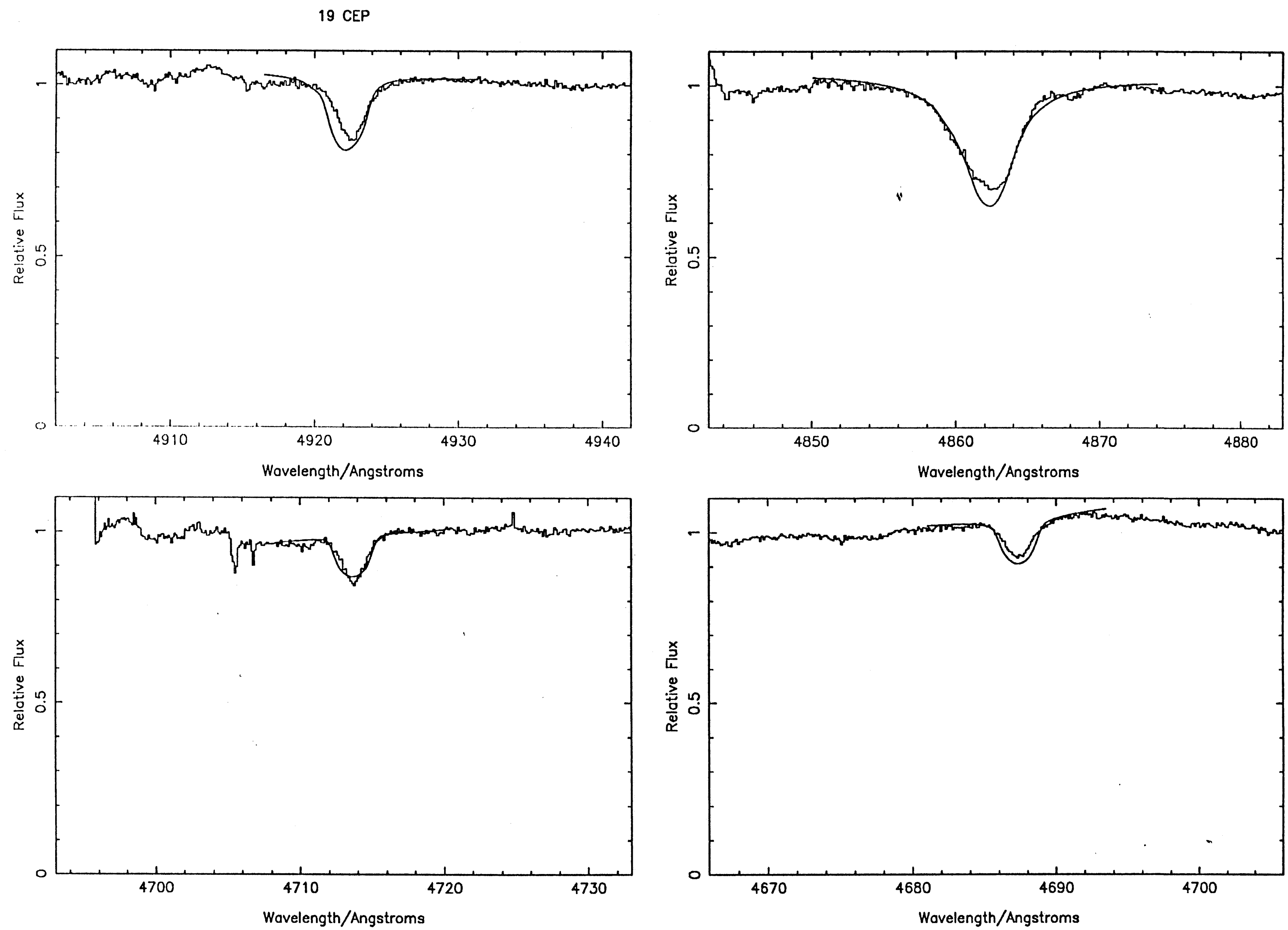

Figure 6. (Continued) HD 209975

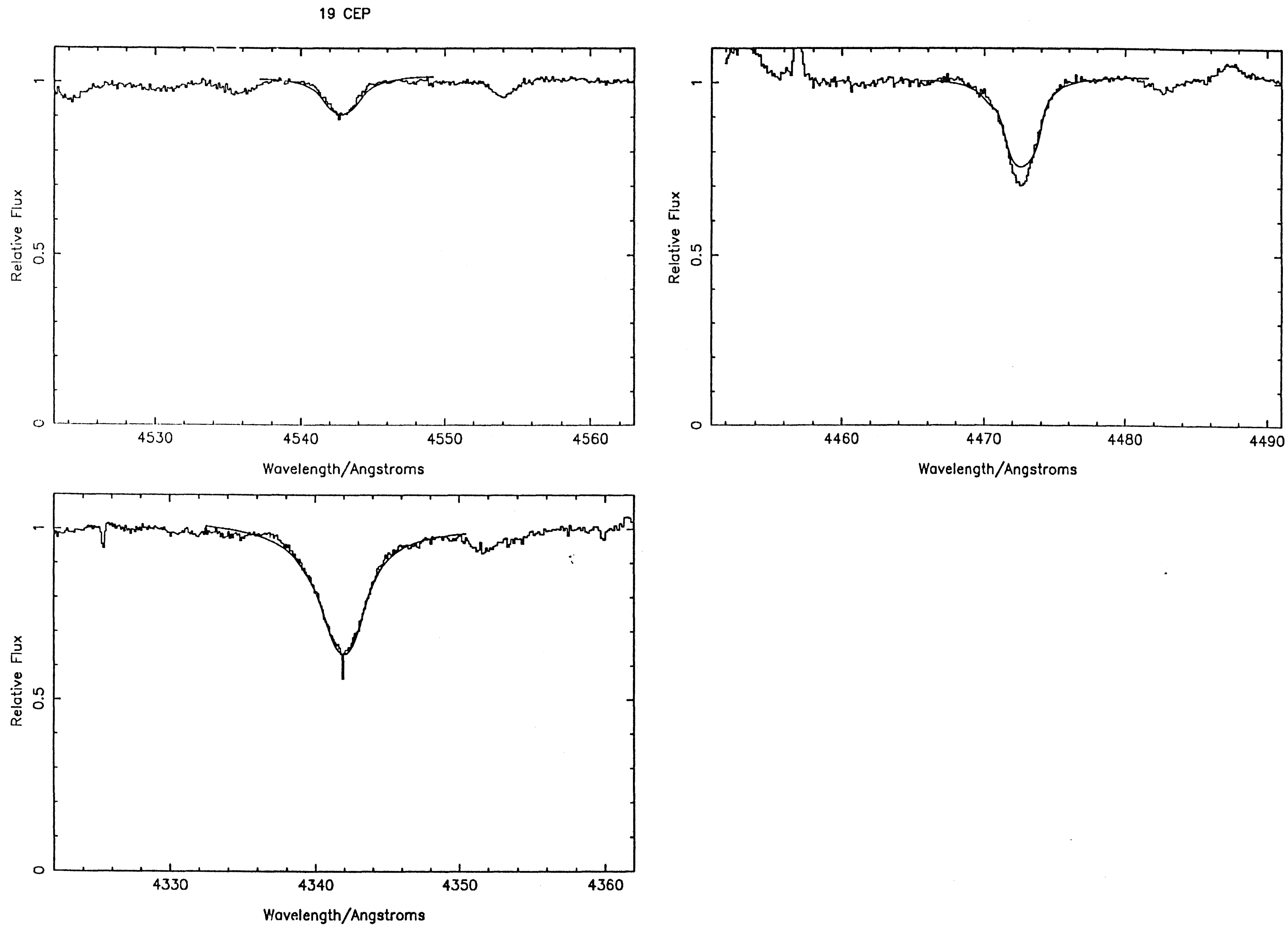

Figure 6. (Continued) HD 209975

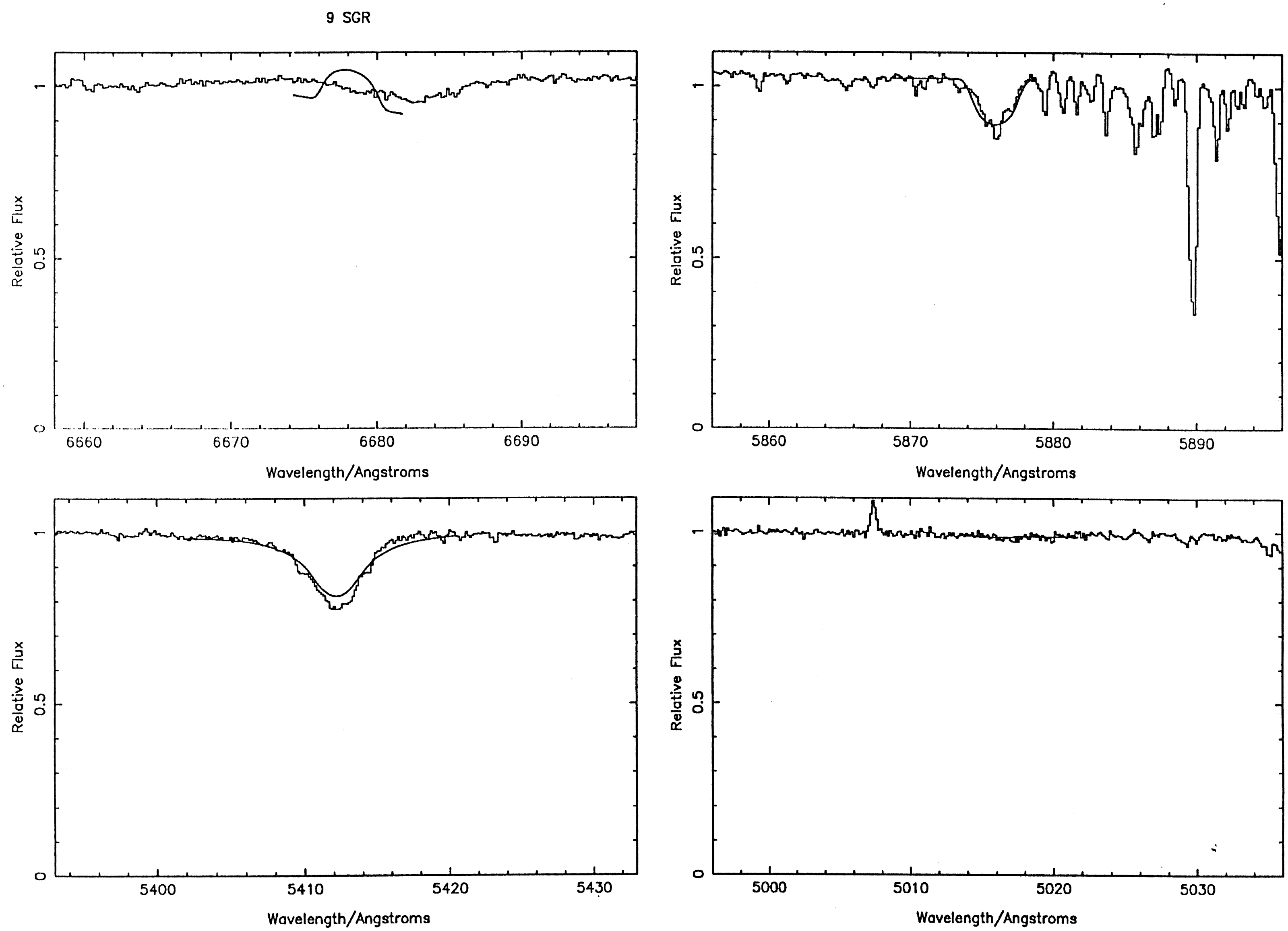

Figure 6. (Continued) 9 Sgr

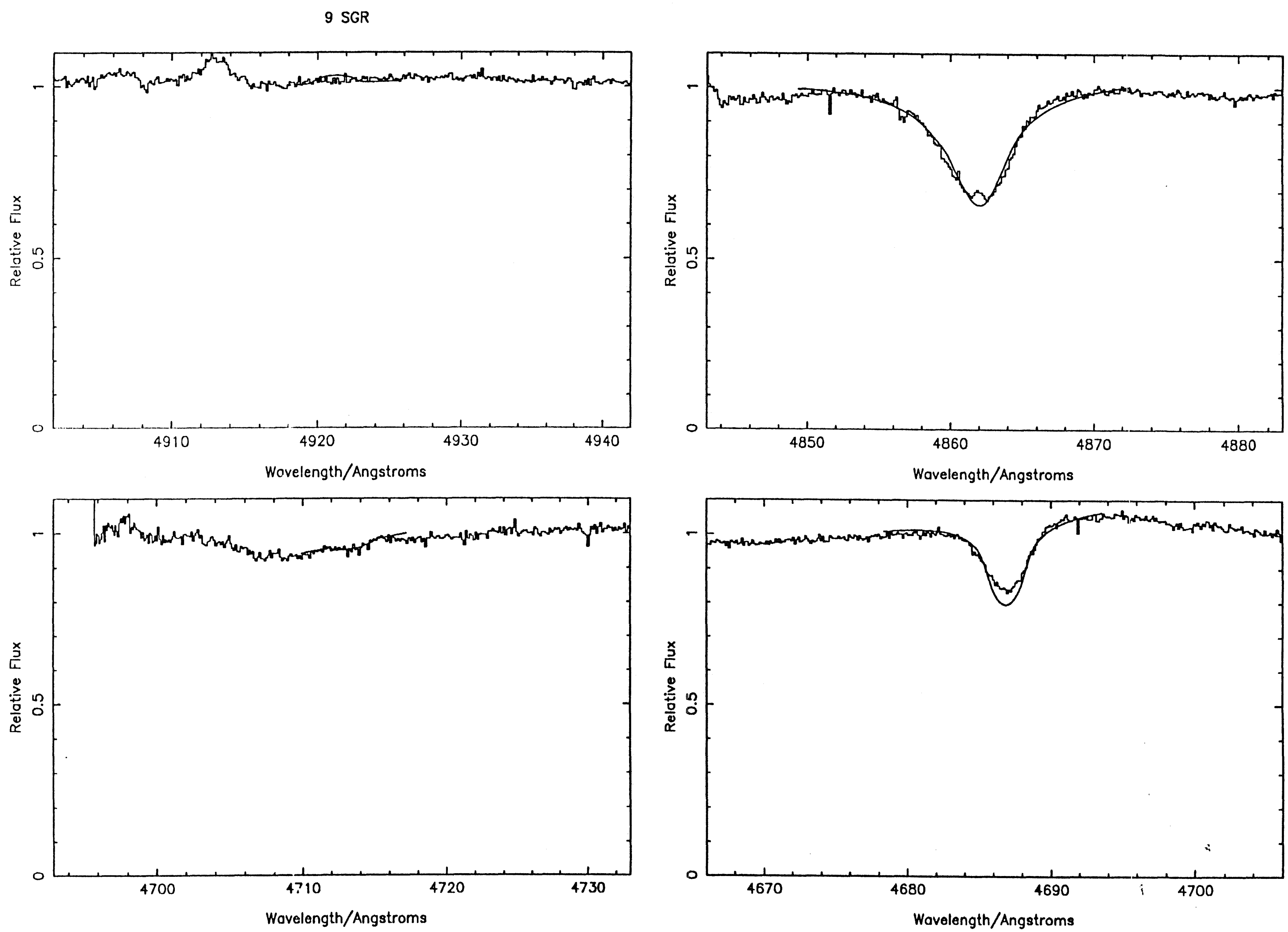

Figure 6. (Continued) 9 Sgr

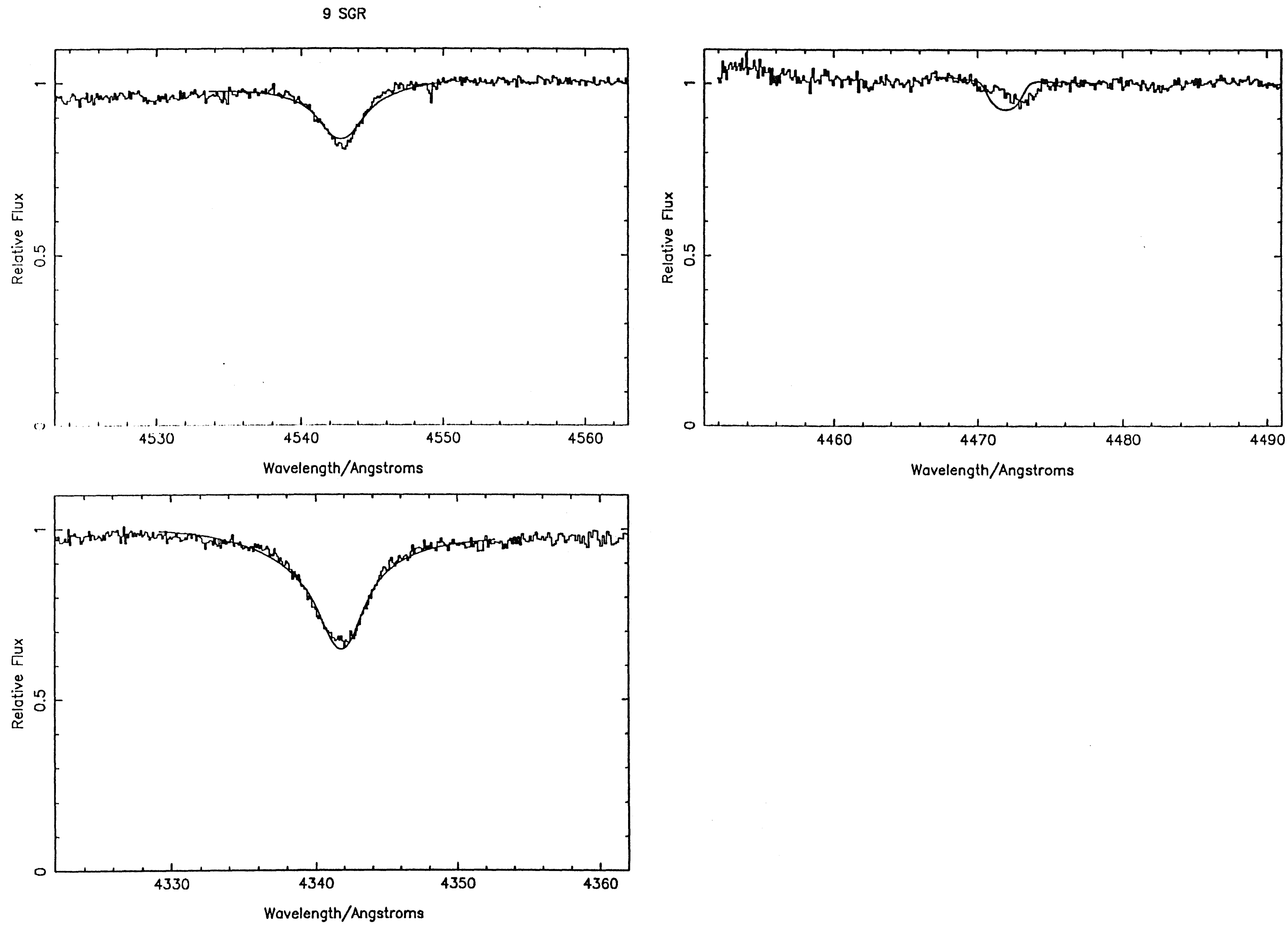

Figure 6. (Continued) 9 Sgr

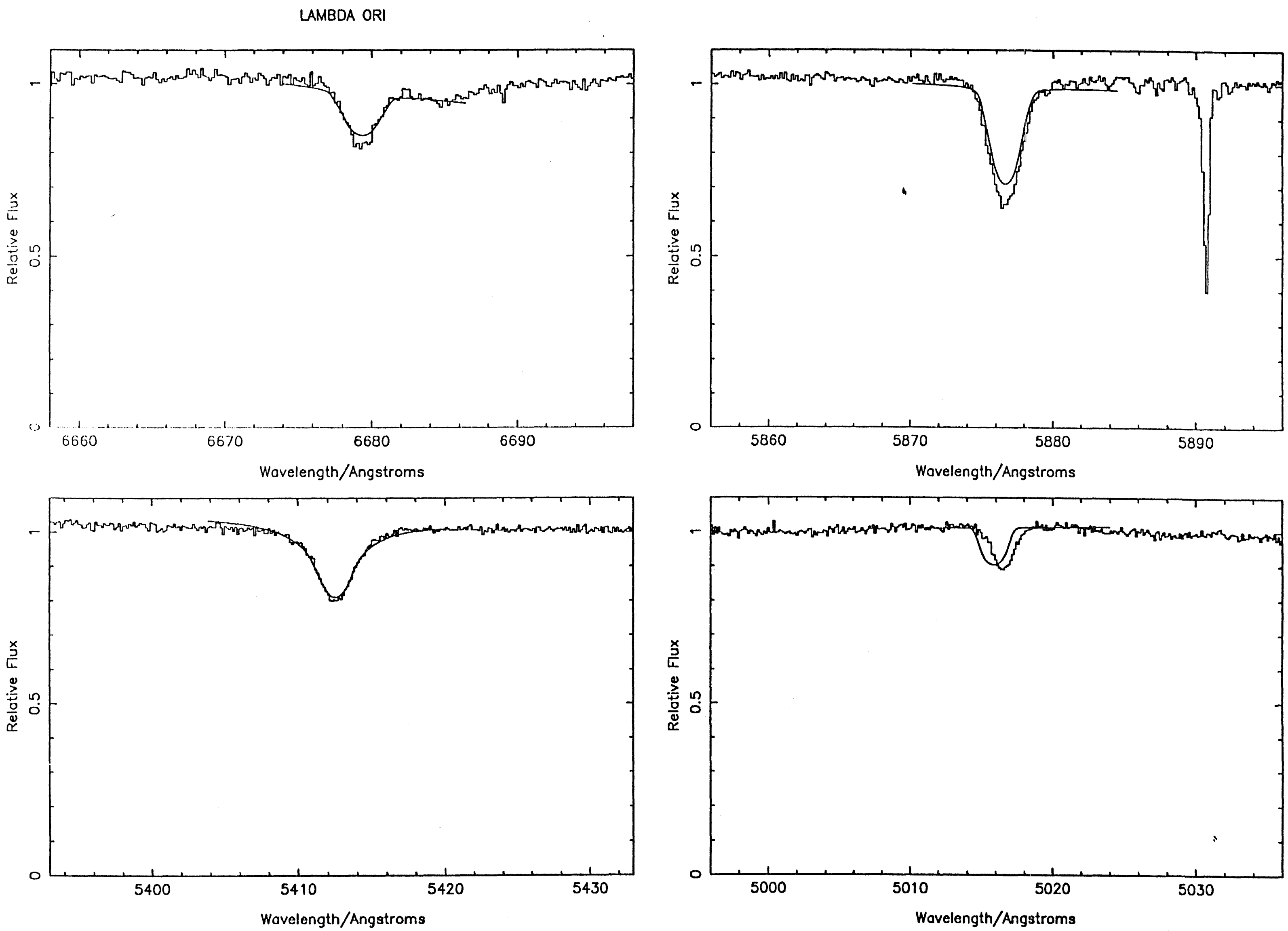

Figure 6. (Continued) λ Ori

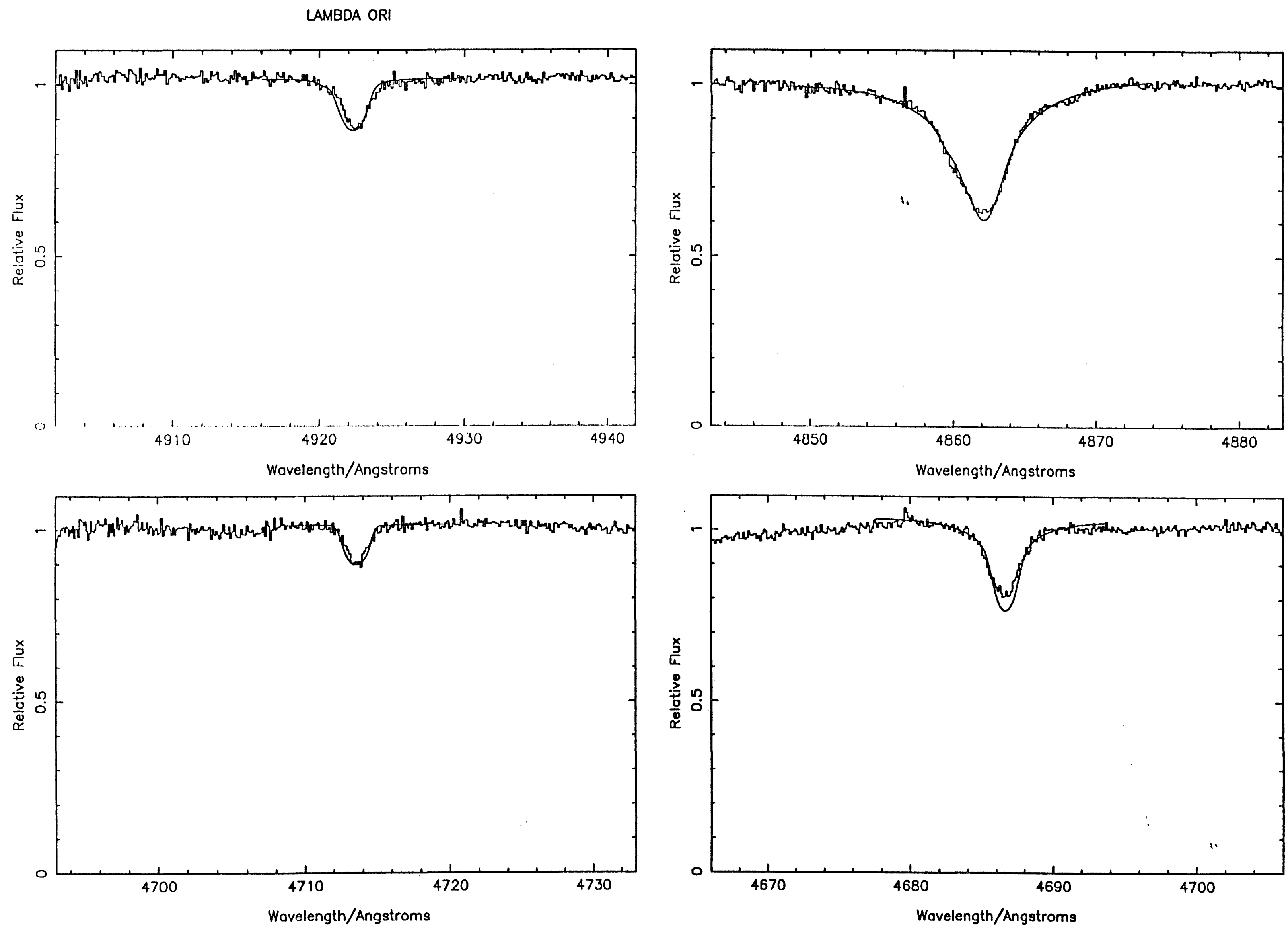

Figure 6. (Continued) λ Ori

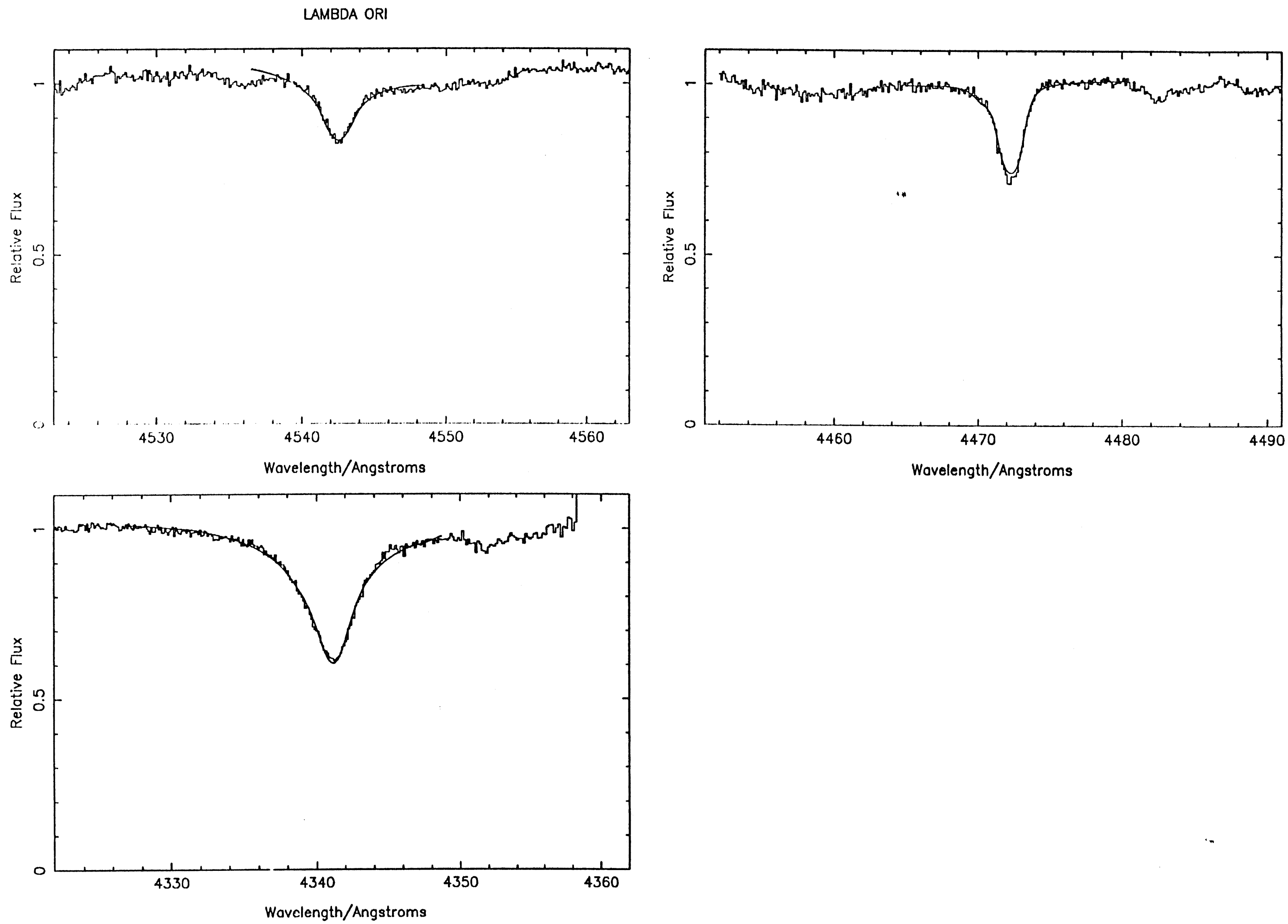

Figure 6. (Continued) λ Ori

5. CONCLUSIONS

5.1 COMPARISON OF THEORY TO OBSERVATION

In examining the fits for individual lines, one sees that two of the lines are not well reproduced by the theoretical calculations. The HeII 4686Å line is found in emission in the Of stars, and Auer and Mihalas [7] showed that this could not be explained by fluorescence arising from the overlap of the hydrogen Balmer and helium Pickering series. They concluded that the emission was likely a result of atmospheric extension. The analysis here shows that the theoretical calculations produce a line that is too strong even for normal O stars.

Auer and Mihalas also noted that the HeI 5876Å line was somewhat stronger in their small sample of stars than was predicted by their theoretical work, although the extent and severity of the discrepancy was unclear. The analysis here confirms the discrepancy. It is found that the effect is strongest in the cooler and more luminous stars in the sample; it is most pronounced in BD +41° 3804, HD 13022, HD 194280, HD 229234, HD 194094, HD 207198, and 19 Cephei, all of which have $T_{eff} < 36000K$ and $\log g < 3.7$. This discrepancy probably arises from the geometrical dilution effect described by Ghobros [18], as noted by Voels *et al.* [47]. However, the effect seems not to be significant for other neutral helium lines. As a check, a second fit to the observed profiles for BD +41° 3804 was made with the HeI 6678Å, 5876Å, and 4471Å lines omitted. The change in the fit parameters was not significant.

Because the theory poorly reproduces the HeII 4686Å and HeI 5876Å lines, these were given one-tenth the normal weight in the χ^2 fit.

Another discrepancy between theory and observations is that the theory predicts emission lines of HeI 6678Å and 4922Å in the hottest stars that are not observed. This is most evident for 9 Sagittarii, which is slightly hotter than 50000K. Since 9 Sagittarii appears to have a rather high surface gravity, this discrepancy probably does not arise from the assumption of plane-parallel geometry.

5.2 HELIUM ABUNDANCES

One disturbing tendency in the fitted parameters is the almost perfect correlation between low gravity (or high luminosity) and high He/H (Figure 7). Although the more luminous stars lose mass at a great rate, so that they are the most likely members of the sample to show a large He/H, it seems very unlikely that *all* the high-luminosity stars should show large He/H, since the associations chosen are not all of the same age. These fitted abundances most likely reflect a systematic failure of the theory rather than the actual compositions of the stars involved.

However, such a failure of the theory is not suggested by the quality of the fits, which are generally quite good. In this respect, the "normal" O stars analyzed differ from the Of stars that were dropped from the sample; attempts to analyze the Of stars by the method described yielded obviously poor fits, with, e.g., indications that the cores of the Balmer lines are much shallower in the observed profiles than in any of the theoretical profiles.

It is of note that Voels *et al.* [47] do not find a large value for He/H for the O supergiants they have analyzed (although two of their objects, ζ Pup and α Cam, show somewhat enhanced He). The only substantial difference between their models and the models used here is the inclusion of the effects of wind blanketing. It is surprising that neglect of wind blanketing should result in systematic overestimates of He/H, particularly since some of the O supergiants for which Voels *et al.* find cosmic He/H show negligible effects from the wind blanketing. Unfortunately, there is no overlap between their sample of objects and the sample of objects analyzed here, which makes a direct comparison difficult. Likewise, they do not present tables of profiles and their illustrations containing profiles do not have labels on the Y-axis. Thus no direct comparison is possible.

A final point that should be raised is that many of the objects with low values for $\log g$ are off the original model grid, and the values for the stellar parameters are extrapolated. In the case of the two lowest-gravity objects, HD 194280 and BD +41° 3804, the model grid was extended to T_{eff}=28000K and $\log g$=3.75 for He/H of 0.20 and 0.50, to reduce the amount of the extrapolation. The result was that the estimated value of He/H actually increased, which would seem to rule

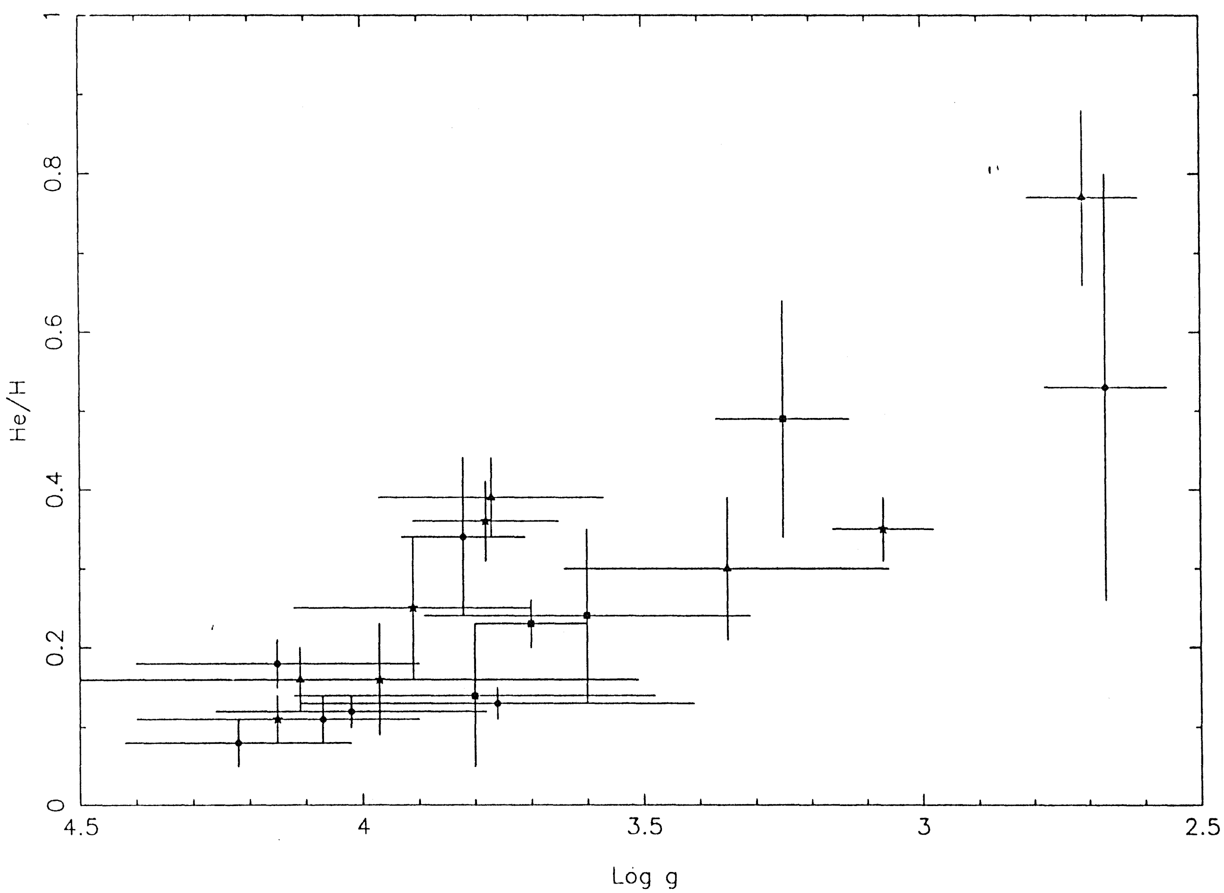

Figure 7. Correlation of Derived He/H with $\log g$. Error bars are indicated. See Figure 1 for explanation of symbols

out the extrapolation of the theoretical profiles as the cause for the overestimation of He/H. (The parameters given for these objects are those obtained with the extended model grid.)

5.2.1 Semi-Empirical Helium Abundances

If the correlation of He/H with $\log g$ is, in fact, the result of a failure of the theory, we may account for this failure by fitting a curve through the plotted points in the ($\log g$, He/H) plane and using it to reduce all values for He/H determined by our method of analysis. The data shown are well represented by the formula, $\log(\text{He/H}) = 1.1264 - 0.4791 \log g$. The reduced helium abundances calculated using this formula and assuming a cosmic He/H of 0.10 are given in the table of estimated parameters under the column labeled "$(\text{He/H})_\circ$," along with association averages. Figure 8 plots these corrected values against effective temperature.

When this reduction is made, we find that there is no object in the sample with an obviously high value for $(\text{He/H})_\circ$. The three stars near $\log g = 3.8$, He/H=0.4, HD 242908, HD 12993, HD 193595, are the likeliest candidates for a moderate helium overabundance ($(\text{He/H})_\circ \sim 0.19$). It may be significant that HD 12993 is certainly a blue straggler, while the other two stars are also possibly blue stragglers (being the hottest objects in each of their respective associations). Although this apparent helium overabundance may be a temperature effect, the normal abundances for HD 40800, HD 242935, HD 35619, HD 42088, #4 OB2 Cyg, and λ Ori argue against this.

One may also conclude from the plotted points and association averages and probable errors that there is no significant difference in the primordial helium abundance between the associations studied. This agrees with the results obtained by Wolf and Heasley [50]but does not agree with those of Nissin [38].

5.3 RADII AND MASSES

If one knows T_{eff}, $\log g$, and the bolometric magnitude M_{bol} for a star, one can derive the radii and masses. Although the analysis here has not determined these quantities with high precision, one can at least make rough estimates of the values of these parameters. These are given in Table 8 .

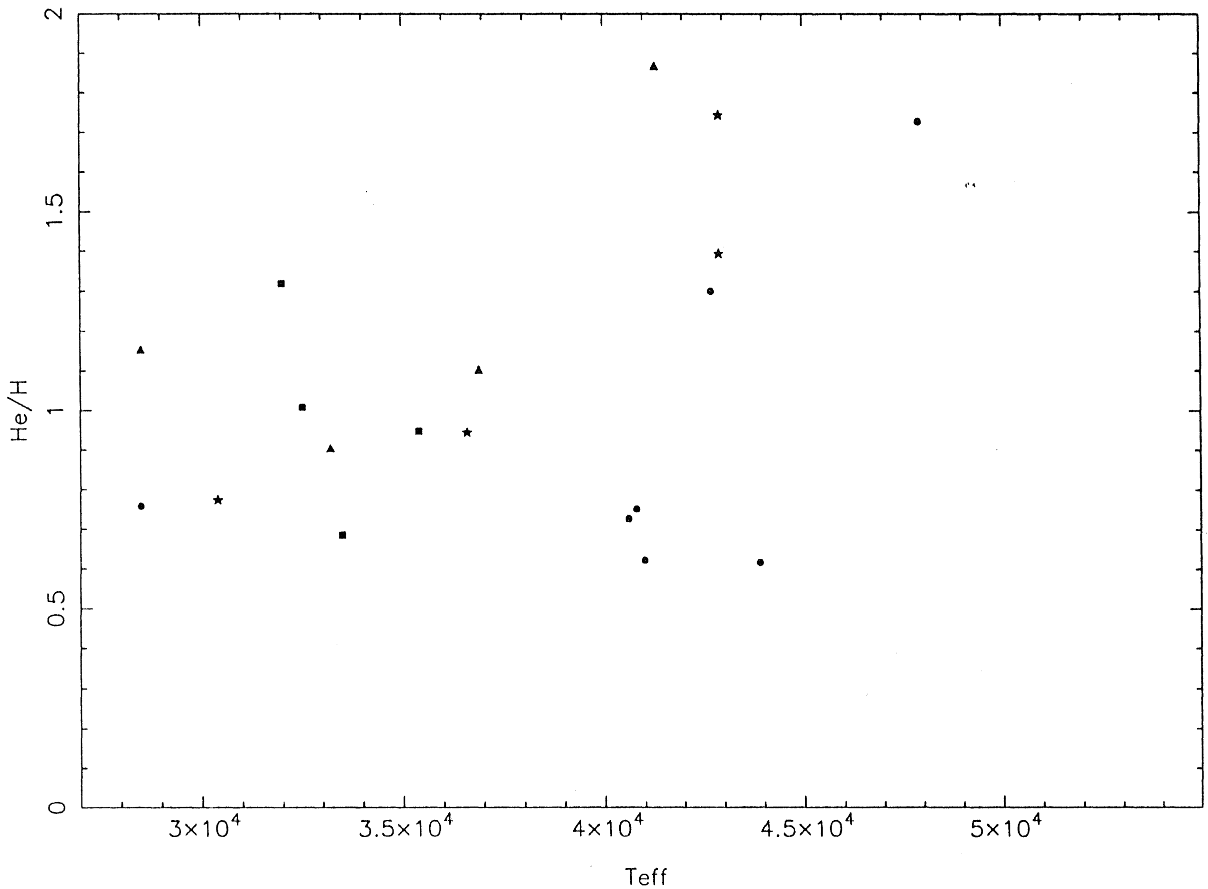

Figure 8. Corrected He/H *vs.* T_{eff}. See Figure 1 for explanation of symbols.

Table 8. Radii and Masses of the Stars in the Sample

Object	B.C.	$M_{\rm bol}$	Radius ($R_\odot$)	Mass ($M_\odot$)
Per OB1				
236894	-3.28	-7.2	6	20
12993	-3.76	-8.2	7	11
13022	-2.89	-7.8	12	6
Aur OB2				
242908	-4.12	-9.4	10	23
242926	-3.61	-8.7	10	37
242935	-3.60	-8.2	8	36
35619	-3.59	-9.2	13	67
Gem OB1				
42088	-3.76	-8.3	7	16
254755	-3.38	-8.2	9	42
Cyg OB1				
193595	-3.65	-8.2	8	12
194094	-3.30	-8.3	10	47
194280	-2.65	-8.8	21	8
Cyg OB2				
# 4	-3.75	-9.4	12	79
# 8B	-3.84	-9.8	14	42
# 10	-2.78	-9.7	32	17
Cep OB2				
204827	-3.08	-8.1	11	28
207198	-3.23	-8.7	13	25
207538	-2.98	-7.2	8	11
209975	-3.02	-8.7	16	17
Miscellaneous				
9 Sag	-4.27	-10.4	14	197
λ Ori	-3.55	-8.8	11	42

The visual magnitudes M_v and bolometric corrections used in this calculation are those reported by Humphreys [24]. Since T_{eff} is well determined, the uncertainty in the radii depends mainly on the uncertainty in the absolute bolometric magnitudes and is probably of order 30%. The uncertainty in the masses is greater, as it depends on the uncertainty of both the bolometric magnitude and the surface gravity, and is probably as large as a factor of two. Although the radii and masses seem generally to be of the right order of magnitude, some of the individual values are highly questionable. In particular, the mass for 9 Sgr is unacceptable. It is likely that $\log g$ for this object has been badly overestimated. The sensitivity of the hydrogen and helium lines to $\log g$ is not great at high T_{eff}.

Stars that have lost particularly large fractions of their mass should be overluminous. However, since an overestimate of the luminosity of a particular star leads to an overestimate of the mass as well, a much more accurate determination of the luminosities is needed than is available for the stars in this sample or, indeed, for any O star. A plot of $M_{\rm bol}$ *vs.* the logs of the masses would show far too much scatter to permit any conclusions about the presence or absence of overluminous stars.

Voels *et al.* [47] also estimate masses spectroscopically for their sample of O supergiants. They find values ranging from 19 to 36 $M_\odot$, in rough agreement with the values found here but with less scatter.

5.4 THE MAPPING OF SPECTRAL TYPES TO PHYSICAL PARAMETERS

Figure 9 shows the (T_{eff}, $\log g$) diagram of all the stars in the sample. The MK spectral class, or closest estimate, is given for each star. The estimated temperature calibration for MK spectral types is given in Table 9 .For comparison, the calibration employed by Humphreys [24] is also listed. There is considerable discrepancy, particularly near spectral types O7 and O8, with the new calibration giving a considerably higher effective temperature. Unfortunately, because of the scantiness of the data, our calibration is probably not much more reliable than the Humphreys calibration.

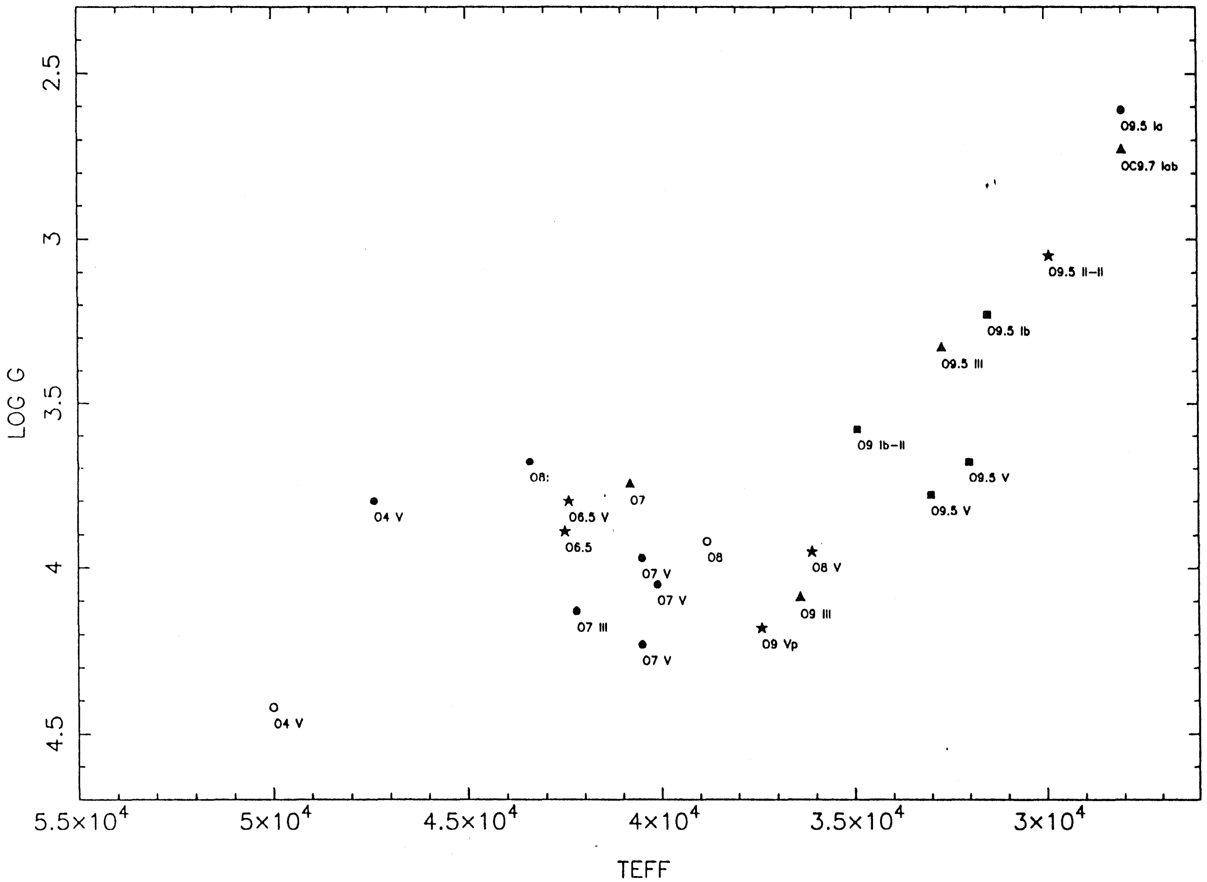

Figure 9. Spectral types *vs.* physical parameters.

Table 9. Calibration of T_{eff} to MK Spectral Type

Temperature Class	Luminosity Class I-II	Luminosity Class III-V
O9.7	28500	
	–	
O9.5	30300	32400
	31000	32000
O9	35400	37400
	32000	33000
O8		40200
		34000
O7		41280
		35400
O6.5		42900
		37500
O4		49200
		50000

Upper value is from the analysis here; lower value is the calibration given by Humphreys [24].

5.5 CARBON AND NITROGEN FEATURES

Although no detailed analysis of carbon is attempted here, it is worth noting that the CIV 5801Å and 5812Å features are stronger for the three members of the Cyg OB2 association than for stars of similar spectral type belonging to other associations. Figure 10 shows the correlation between the equivalent width of a prominent ionized helium line, HeII 4542Å, and the equivalent width of one member of the C IV doublet at 5812Å. (This member of the doublet was chosen because it is less confused with diffuse interstellar bands than the component at 5801Å.) The open points corresponding to the three stars analyzed from Cyg OB2 definitely lie above the average curve for this correlation, indicating a carbon overabundance for this association. The carbon line is roughly .35 dex stronger than normal for these stars. If we assume that the carbon line equivalent widths obeys the well-known square-root law for a strong line with a Voigt absorption profile, this corresponds to an approximate carbon overabundance of 50%.

Since the carbon overabundance is detected in three quite dissimilar O stars in Cyg OB2, it must be regarded as a primordial carbon enrichment. Charles *et al.* [13]find that the entire region around Cygnus OB2 is disturbed, and they attribute this to the explosion of 30-100 supernovae in the last few million years. These supernovae might account for the local carbon enrichment.

It should also be pointed out that HD 236894 is clearly carbon-deficient, the carbon abundance indicated by the line strength being about half normal. It would be very interesting to estimate the relative nitrogen abundance of this object, as this might confirm that the carbon underabundance is evolutionary. Such an estimate could not be made with the echelle spectrograms presented here, since the only detectable nitrogen lines are too close to the blue for accurate measurements. It would be interesting to examine the nitrogen abundances of the stars in OB2 Cyg as well.

It is probably worthwhile to mention the findings of Schild and Berthet [42]. Three of the objects for which they present nitrogen spectra are included in the analysis here. These are HD 42088 and HD 254755, for which they report enhanced nitrogen, and HD 12993, for which they report moderately enhanced nitrogen. HD 12993 is one of the three blue stragglers in the sample for

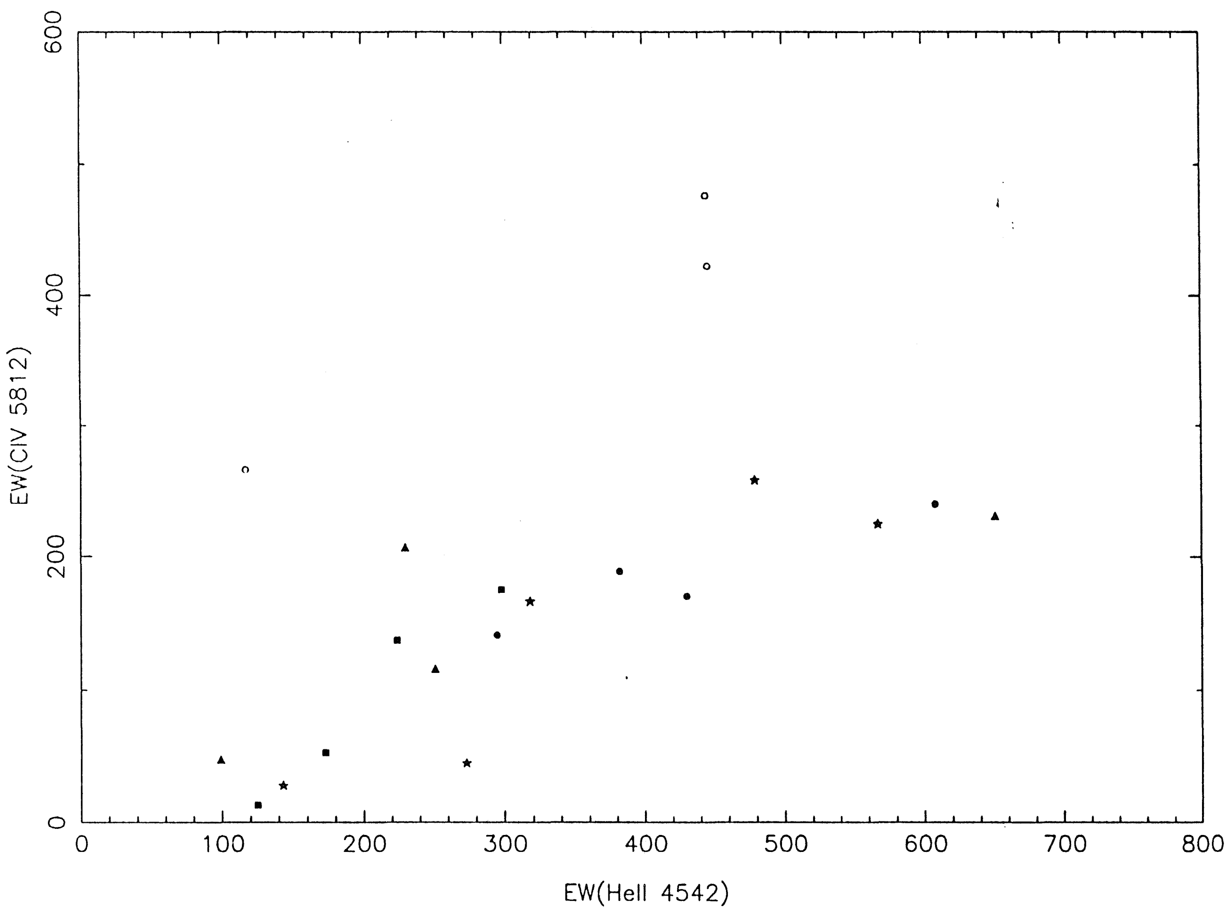

Figure 10. Equivalent Widths of C IV 5812Å *vs.* HeII 4542Å. See Figure 1 for explanation of symbols.

which a helium overabundance is suspected. No helium overabundance is suspected for the others, but this is not incompatible with the nitrogen overabundance; evolutionary changes in the nitrogen abundance may be expected to appear before changes in the helium abundance. It should be kept in mind that the helium overabundance of HD 12993 is only suspected and must be confirmed by a more careful analysis, using the best possible data and the most accurate available atmospheric models.

5.6 SUMMARY

Although reasonable values for the effective temperatures and gravities of O stars may be obtained by an analysis of their line spectra, using non-LTE, plane-parallel models, the helium abundances so obtained show a strong correlation with $\log g$; the stars with higher $\log g$ show an abundance ratio near the accepted cosmic value (He/H $\sim$ 0.10), but at lower gravities the ratio increases to unreasonable values (He/H $\sim$ 0.50). This failure of the theory probably arises from the approximation of plane-parallel geometry. The correlation is well fit by the relation log He/H = 1.1234-0.4791 $\log g$.

If this correlation is divided out of the fitted helium abundances, one obtains "corrected" abundances that probably correspond to the true helium-to-hydrogen ratio. It is found that there is no systematic difference in the helium abundance from association to association. However, the stars HD 12993, HD 242908, and HD 193595 may be blue stragglers with an enhanced helium abundance (He/H$\sim$0.19). The enhanced helium abundance would arise from the same mixing that explains their status as blue stragglers.

A strong correlation exists between the equivalent widths of the CIV 5812Å line and the HeII 4342Å line. Using this correlation, it is possible to estimate the approximate carbon abundance of different O stars relative to the average cosmic value. It is found that the association Cyg OB2 is carbon-rich by $\sim$ 50%. This carbon enrichment may be a result of heavy supernova activity in the Cygnus Superbubble [13]. The blue straggler HD 236894, on the other hand, is found to have roughly half the normal carbon abundance.

The calibration of physical parameters to the MK spectral classification is still very uncertain, with discrepancies between previous calibrations and the calibration given here being as great as 6200K. Many more MK-classified stars must be analyzed before an improved calibration can be given.

REFERENCES

1. Abbott,D., and Hummer,D. 1985. *Astrophys.J.* **294**:286.

2. Abbott,D., and Lucy,L. 1985. *Astrophys.J.* **288**:679.

3. Aggarwal,K. 1983. *M.N.R.A.S.* **202**:15p.

4. Aggarwal,K., Kingston,A., and McDowell,M. 1978. *Astrophys.J.* **278**:874.

5. Anderson,L. 1985. *Astrophys.J.* **298**:848.

6. Anderson,L. 1987. In *Numerical Radiative Transfer* (W.Kalkofen, ed.) Cambridge University Press, Cambridge

7. Auer,L.H., and Mihalas,D. 1971. *Astrophys.J.* **24**:193.

8. Barnard,A.J., Cooper,J., and Smith,E.W. 1974. *J.Quant.Spectrosc.Radiat.Transfer* **14**:1025.

9. — 1975. *J.Quant.Spectrosc.Radiat.Transfer* **15**:429.

10. Berrington,K.A., Burke,P.G., Freitas,L.C.G., and Kingston,A.E. 1985. *J.Phys.B* **18:4135**

11. Bertelli,G., Bressan,A., and Chiosi,C. 1984. *Astron.Astrophys.* **130**:279.

12. Bohannan,B., Abbott,D., Voels,S., and Hummer,D. 1986. *Astrophys.J.* **308**:728.

13. Cash,W., Charles,P., Bowyer,S., Walter,F., Garmire,G., and Riegler,G. 1980. *Astrophys.J.* **38**:309.

14. Chiosi,C., and Maeder,A. 1985. *Ann.Rev.Astron.Astrophys.* **24**:329.

15. Code,A., Davis,J., Bless,R., and Brown,R.H. 1976. *Astrophys.J.* **203**:417.

16. Drew,J. 1989. *Astrophys.J.Supp.* **71**:267.

17. Geltman,S. 1962. *Astrophys.J.* **136**:935.

18. Ghobros,R.A. 1962. *Zeits.f.Astrophysik* **56**:113.

19. Gingerich,O. 1969. *Theory and Observation of Normal Stellar Atmospheres*, (Gin-

gerich,O., ed.), MIT Press, Cambridge, MA, p.17.

20. Griem,H.R., Baranger,M., Kolb,A.C., and Oertel,G. 1962. *Phys.Rev.* **125**:177.

21. Hata,J., Morgan,L.A., and McDowell,M.R.C. 1980. *J.Phys.B* **13**:L347.

22. Herrerro,A. 1987. *Astron.Astrophys.* **186**:231.

23. Hummer,D., and Voels,S. 1988. *Astron.Astrophys.* **192**:279.

24. Humphreys,R. 1978. *Astrophys.J.* **38**:309.

25. Humphreys,R., and McElroy,D. 1984. *Astrophys.J.* **284**:565.

26. Jacobs,V.L. 1974. *Phys.Rev.* **A9**:1938.

27. Kalkofen,W. (ed.) 1987. *Numerical Radiative Transfer.* Cambridge University Press, Cambridge.

28. Lennon,D.J., Lynas-Gray,A.E., Brown,P.J.F., and Dufton,P.L. 1986. *M.N.R.A.S.* **222**:719.

29. McCarthy,J. 1988. Ph.D. dissertation. Pasadena: California Institute of Technology.

30. Maeder,A. 1983. *Astron.Astrophys.* **120**:113.

31. Matthews,T., and Sandage,A. 1963. *Astrophys.J.* **138**:30.

32. Meylan,G., and Maeder,A. 1982. *Astron.Astrophys.* **108**:148.

33. Mihalas,D. 1972. NCAR-TN/STR-76. Boulder: National Center for Atmospheric Research.

34. Mihalas,D. 1978. *Stellar Atmospheres.* San Francisco: W.H. Freeman and Co

35. Mihalas,D., and Auer,L. 1970. *Astrophys.J.* **161**:1129.

36. Mihalas,D., Auer,L., and Heasley,J. 1975. NCAR-TN/STR-104. Boulder: National Center for Atmospheric Research.

37. Mihalas.D., and Stone,M.E. 1968. *Astrophys.J.* **151**:293.

38. Nissen, P. 1983. *Primordial Helium.* (Shaver,P., Kunth,D., and Kjar,K., ed.), European Southern Observatory, Munich, pp. 163-177.

39. Pearson,T. 1988. *PGPLOT Graphics Subroutine Library.* Pasadena: California

Institute of Technology.

40. Press,W., Flannery,B., Teukolsky,S., and Vetterling,W. 1986. *Numerical Recipes.* Cambridge University Press, Cambridge.

41. Rybicki,G. 1971. *J.Q.S.R.T.* **11**:589.

42. Schild,H., and Berthet,S. 1986. *Astron.Astrophys.* **162**:369.

43. Schöning,T., and Butler,K. 1989. *Astron.Astrophys.Suppl.* **78**:51.

44. Stewart,A.L. 1978. *J.Phys.B* **11**:L431.

45. Stewart,A.L., and Webb,T.G. 1963. *Proc.Phys.Soc.* **82**:532.

46. Vidal,C.R., Cooper,J., and Smith,E.W. 1973. *Astrophys.J.Suppl.* **25**:37.

47. Voels,S., Bohannan,B., Abbott,D., and Hummer,D. 1989. *Astrophys.J.* **340**:1073.

48. Werner,K. 1988.. *Astron.Astrophys.* **204**:159.

49. Wiese,L., Smith,M., and Glennon,B. 1966. *Atomic Transition Probabilities.* NSRDS-NBS 4.

50. Wolf,S., and Heasley,J. 1985. *Astrophys.J.* **292**:589.

Appendix. Selected Listings of Computer Programs

The following pages list portions of the FORTRAN source codes used to carry out the calculations described in this dissertation. Wherever possible, this code has adhered strictly to the ANSI 1977 standard. The listings are not complete; omissions are noted in the introduction to each code and in certain places in the code listings themselves. Readers interested in obtaining the full listings in machine-readable format are encouraged to contact the author at the following address:

Kent G. Budge

Organization 1531

Sandia National Laboratory

Albuquerque, NM 87185

A. Program GRAY

GRAY is listed in its entirety.

The program takes an input file in standard atmosphere format, but no lines after the opacity specification line are used. The output file is also in standard atmosphere format, and includes the LTE populations of hydrogen and helium for standard model ions. It may be read directly into the program LTE if the default number of iterations (15) is acceptable, as is usually the case.

The subroutines LINK and EXIT are CRAY FORTLIB routines to create a dropfile and initial I/O links and to exit the program. EXIT(n) is equivalent to a Fortran STOP n statement, while LINK has no normal FORTRAN equivalent.

```
      PROGRAM GRAY
C
C     COMMON BLOCK MACROS FOR ALL GLOBAL DATA IN PROGRAM
C     (CRAY PRECOMPILER)
C
C     COMMON PARAMETERS:
C
C     MNDEPTH    MAXIMUM NUMBER OF PHYSICAL DEPTHS IN MODEL
C     MNJ        MAXIMUM NUMBER OF FREQUENCIES
C     MQH        MAXIMUM HYDROGEN LEVELS FOR PARTITION FUNCTION
C     MQHE1      MAXIMUM HELIUM LEVELS FOR PARTITION FUNCTION
C                HELIUM-2 USES MQH*2 FOR PARTITION FUNCTION
C     NLH        NUMBER OF HYDROGEN LEVELS TREATED EXPLICITLY
C     NLHE1      NUMBER OF NEUTRAL HELIUM LEVELS TREATED EXPLICITLY
C     NLHE2      NUMBER OF SINGLY-IONIZED HELIUM LEVELS TREATED EXPLICITLY
C
C     HK         PLANK'S CONSTANT OVER BOLTZMANN'S CONSTANT
C     MHYD       1./AVOGADRO'S NUMBER = MASS OF HYDROGEN NUCLEUS
C     SIGE       THOMPSON CROSS-SECTION FOR FREE ELECTRONS
C
C     COMMON VARIABLES:
C
C     NDEPTH     NUMBER OF DEPTH POINTS
C     NJ         NUMBER OF FREQUENCY POINTS
C
C     CHI        EXTINCTION COEFFICIENT
C     FF         FREE-FREE CUTOFF FREQUENCY
C                        THIS ALLOWS FOR NEARLY-FREE BOUND STATES
C     FREQ       FREQUENCY GRID
C     FRQH       HYDROGEN IONIZATION FREQUENCIES
C     FRQHE1     NEUTRAL HELIUM IONIZATION FREQUENCIES
C     FRQHE2     IONIZED HELIUM IONIZATION FREQUENCIES
C     GH         HYDROGEN LEVEL STATISTICAL WEIGHTS
C     GHE1       NEUTRAL HELIUM LEVEL STATISTICAL WEIGHTS
C     GHE2       IONIZED HELIUM LEVEL STATISTICAL WEIGHTS
C     GRAV       SURFACE GRAVITY
C     M          MASS GRID (DIFFERENCES)
C     MU         MEAN MOLECULAR WEIGHT
C     MU1        NUCLEI PER PROTON
C     N          HYDROGEN NUMBER DENSITY
C     NE         ELECTRON NUMBER DENSITY
C     NHE1       NEUTRAL HELIUM NUMBER DENSITY
C     NHE2       IONIZED HELIUM NUMBER DENSITY
C     NHE3       DOUBLY-IONIZED HELIUM NUMBER DENSITY
C     NPROT      IONIZED HYDROGEN NUMBER DENSITY
C     NTOT       TOTAL PARTICLE NUMBER DENSITY
C     SIG        HYDROGEN PHOTOIONIZATION CROSS-SECTIONS
C     SIGHE1     NEUTRAL HELIUM PHOTOIONIZATION CROSS-SECTIONS
C     SIGHE2     IONIZED HELIUM PHOTOIONIZATION CROSS-SECTIONS
C     SUMH       PARTITION FUNCTION OF HYDROGEN
C     SUMHE1     PARTITION FUNCTION OF NEUTRAL HELIUM
C     SUMHE2     PARTITION FUNCTION OF IONIZED HELIUM
C     TEMP       ELECTRON TEMPERATURE
C     WT         QUADRATURE WEIGHTS FOR FREQUENCY GRID
C     Y          NUMBER RATIO OF HELIUM TO HYDROGEN
C
      CLICHE PLIST
C
      INTEGER MNDEPTH, MNJ, MQH, MQHE1, NLH, NLHE1, NLHE2
      REAL DELQUAD, HK, MHYD, SIGE
      PARAMETER (MNDEPTH=100, MNJ=105, MQH=16, MQHE1=31, NLH=10)
      PARAMETER (NLHE1=25, NLHE2=15, DELQUAD=0.6, HK=4.79864E-11)
      PARAMETER (MHYD=1.66058E-24, SIGE=0.664E-24)
C
C     POINTERS FOR STATE MATRIX
C
      INTEGER INHE2, INHE3, INPROT, INTOT, MNNN
C
      PARAMETER (INHE2=1, INHE3=2, INPROT=3, INTOT=4, MNNN=4)

      ENDCLICHE

      CLICHE BLANK
C
      REAL CHI(MNJ)
C
      COMMON //CHI
C
      ENDCLICHE
```

```
      CLICHE ROOT
C
      INTEGER NDEPTH, NJ
C
      REAL FF(MNJ,3), FREQ(MNJ), GRAV, M(MNDEPTH), MU, MU1
      REAL NE(MNDEPTH), N(NLH,MNDEPTH), NHE1(NLHE1,MNDEPTH)
      REAL NHE2(NLHE2,MNDEPTH), NHE3(MNDEPTH)
      REAL NPROT(MNDEPTH), NTOT(MNDEPTH), SIGHM(MNJ)
      REAL SIG(NLH+1,MNJ), SIGHE1(NLHE1,MNJ), SIGHE2(NLHE2,MNJ)
      REAL SUMH(MNDEPTH), SUMHE1(MNDEPTH), SUMHE2(MNDEPTH)
      REAL TEFF, TEMP(MNDEPTH), WT(MNJ), Y
      LOGICAL RULE(2)
C
      COMMON /ROOT/NDEPTH, NJ, FF, FREQ, GRAV, M, MU, MU1, N, NE,
     : NHE1, NHE2, NHE3, NPROT, NTOT, SIG, SIGHE1, SIGHM,
     : SIGHE2, SUMH, SUMHE1, SUMHE2, TEFF, TEMP, WT, Y, RULE
C
      ENDCLICHE

      CLICHE ROOTI
C
      REAL FRQHM, FRQH(MQH), FRQHE1(MQHE1), FRQHE2(2*MQH)
      REAL GH(MQH), GHE1(MQHE1), GHE2(2*MQH)
C
      COMMON /ROOTI/FRQHM, FRQH, FRQHE1, FRQHE2, GH, GHE1, GHE2
C
      ENDCLICHE

      PROGRAM GRAY
C
C     CREATE A GRAY MODEL ATMOSPHERE FOR USE AS FIRST APPROXIMATION
C     IN MORE SOPHISTICATED CODES
C
      IMPLICIT NONE
C
      PLIST
      ROOT
C
C     LOCAL VARIABLES
C
      REAL ERR, FLUX, KAP1, KAP2, M2(70), MASS(99), MEND, MSTART, MSTEP
      REAL TAU(99), TEND, TSTART, TSTD(70), TSTEP
      LOGICAL SECOND, STDGRID
      INTEGER I, IO, J, NO
C
C     STANDARD GRID
C
      DATA TSTD/1.000E-07, 3.162E-07, 1.000E-06, 1.334E-06, 1.778E-06,
     : 2.371E-06, 3.162E-06, 4.217E-06, 5.623E-06, 7.499E-06, 1.000E-05,
     : 1.334E-05, 1.778E-05, 2.371E-05, 3.162E-05, 4.217E-05, 5.623E-05,
     : 7.499E-05, 1.000E-04, 1.334E-04, 1.778E-04, 2.371E-04, 3.162E-04,
     : 4.217E-04, 5.623E-04, 7.499E-04, 1.000E-03, 1.334E-03, 1.778E-03,
     : 2.371E-03, 3.162E-03, 4.217E-03, 5.623E-03, 7.499E-03, 1.000E-02,
     : 1.334E-02, 1.778E-02, 2.371E-02, 3.162E-02, 4.217E-02, 5.623E-02,
     : 7.499E-02, 1.000E-01, 1.334E-01, 1.778E-01, 2.371E-01, 3.162E-01,
     : 4.217E-01, 5.623E-01, 7.499E-01, 1.000E-00, 1.334E-00, 1.778E-00,
     : 2.371E-00, 3.162E-00, 4.217E-00, 5.623E-00, 7.499E-00, 1.000E+01,
     : 1.334E+01, 1.778E+01, 2.371E+01, 3.162E+01, 4.217E+01, 5.623E+01,
     : 7.499E+01, 1.000E+02, 1.334E+02, 1.778E+02, 1.780E+02/
C
      DATA MASS/10.000E-08, 1.259E-07, 1.585E-07, 1.995E-07, 2.512E-07,
     : 3.162E-07, 3.981E-07, 5.012E-07, 6.310E-07, 7.943E-07,10.000E-07,
     : 1.259E-06, 1.585E-06, 1.995E-06, 2.512E-06, 3.162E-06, 3.981E-06,
     : 5.012E-06, 6.310E-06, 7.943E-06,10.000E-06, 1.259E-05, 1.585E-05,
     : 1.995E-05, 2.512E-05, 3.162E-05, 3.981E-05, 5.012E-05, 6.310E-05,
     : 7.943E-05,10.000E-05, 1.259E-04, 1.585E-04, 1.995E-04, 2.512E-04,
     : 3.162E-04, 3.981E-04, 5.012E-04, 6.310E-04, 7.943E-04,10.000E-04,
     : 1.259E-03, 1.585E-03, 1.995E-03, 2.512E-03, 3.162E-03, 3.981E-03,
     : 5.012E-03, 6.310E-03, 7.943E-03,10.000E-03, 1.259E-02, 1.585E-02,
     : 1.995E-02, 2.512E-02, 3.162E-02, 3.981E-02, 5.012E-02, 6.310E-02,
     : 7.943E-02,10.000E-02, 1.259E-01, 1.585E-01, 1.995E-01, 2.512E-01,
     : 3.162E-01, 3.981E-01, 5.012E-01, 6.310E-01, 7.943E-01,10.000E-01,
     : 1.259E-00, 1.585E-00, 1.995E-00, 2.512E-00, 3.162E-00, 3.981E-00,
     : 5.012E-00, 6.310E-00, 7.943E-00,10.000E-00, 1.259E+01, 1.585E+01,
     : 1.995E+01, 2.512E+01, 3.162E+01, 3.981E+01, 5.012E+01, 6.310E+01,
     : 7.943E+01,10.000E+01, 1.259E+02, 1.585E+02, 1.995E+02, 2.512E+02,
     : 3.162E+02, 3.981E+02, 5.012E+02, 6.310E+02/
C
C     EXTERNALS
```

```
C
      REAL EXIT, FNT, LINK, MASINT, PUTOUT, SETUP, STATE
      EXTERNAL EXIT, FNT, LINK, MASINT, PUTOUT, SETUP, STATE
C
C     CRAY DROPFILE LINK FUNCTION
C
      CALL LINK("UNIT5=(input,OPEN,TEXT),UNIT12=
     :(output,CREATE,TEXT),UNIT6=TERMINAL//")
C
C     PREPARE MASS DIFFERENCE GRID
C
      M(1)=MASS(1)
      DO 10 I=2,99
         M(I)=MASS(I)-MASS(I-1)
10    CONTINUE
C
C     READ IN PARAMETER/WAVELENGTH FILE AND PREPARE FOR COMPUTATION
C
      CALL SETUP
C
C     CALCULATE SURFACE BOUNDARY CONDITIONS
C
      FLUX=5.6697E-5*TEFF**4/2.99792E10
      TEMP(1)=FNT(0.)
      SECOND=.FALSE.
      NO=99
20    NTOT(1)=GRAV*M(1)/(1.38E-16*TEMP(1))
30    CALL STATE(1,KAP1)
      TAU(1)=KAP1*M(1)
      TEMP(1)=FNT(TAU(1))
      ERR=NTOT(1)
      NTOT(1)=MAX(NTOT(1)*.1,M(1)*(GRAV-KAP1*FLUX)/TEMP(1)/1.3806E-16)
      ERR=ABS((ERR-NTOT(1))/NTOT(1))
      IF (ERR.GT..0005)GO TO 30
C
C     INTEGRATE OVER DEPTH GRID
C
      KAP2=KAP1
      NDEPTH=70
      DO 50 I=2,NO
         TEMP(I)=TEMP(I-1)
         NTOT(I)=NTOT(I-1)
40       TAU(I)=TAU(I-1)+.5*(KAP1+KAP2)*M(I)
         TEMP(I)=FNT(TAU(I))
         CALL STATE(I,KAP2)
         ERR=NTOT(I)
         NTOT(I)=NTOT(I-1)+M(I)*(GRAV-.5*(KAP1+KAP2)*FLUX)/TEMP(I)/
     :            1.3806E-16
         ERR=ABS((ERR-NTOT(I))/NTOT(I))
         IF (ERR.GT..0005)GO TO 40
         KAP1=KAP2
50    CONTINUE
      IF (SECOND)GO TO 80
C
C     FIRST CYCLE--CALCULATE MASS GRID CORRESPONDING TO DEPTH GRID
C
      CALL MASINT(TAU,MASS,99,TSTD,M2,NDEPTH)
      DO 60 I=1,NDEPTH
         MASS(I)=M2(I)
60    CONTINUE
      M(1)=MASS(1)
      DO 70 I=2,NDEPTH
         M(I)=MASS(I)-MASS(I-1)
70    CONTINUE
      SECOND=.TRUE.
      NO=NDEPTH
      GO TO 20
C
C     SECOND CYCLE COMPLETE--WRITE OUT GRAY MODEL
C
80    CALL PUTOUT
      CALL EXIT(0)
      END

      SUBROUTINE MASINT (TO,MO,NO,T1,M1,N1)
C
C     INTERPOLATE TO DEFINED AT POINTS MO TO T1 DEFINED AT POINTS M1
C     ADAPTED FROM AUER/MIHALAS CODE
C
      REAL TO(NO), MO(NO), T1(N1), M1(N1)
```

```
      IF (TO(1).GT.T1(1))GO TO 40
      IF (TO(NO).LT.T1(N1))GO TO 40
      IID=2
      DO 30 I1=1,N1
         DO 10 ID=IID,NO
            IF (TO(ID).GT.T1(I1))GO TO 20
10       CONTINUE
20       IID=ID
         ID=IID-1
         M1(I1)=MO(ID)+
     :            (MO(IID)-MO(ID))*(T1(I1)-TO(ID))/(TO(IID)-TO(ID))
30    CONTINUE
      RETURN
40    WRITE (6,*) 'INTERPOLATION TABLE TOO SMALL: TO(1)=',TO(1),
     :                   ', TO(MAX)=',TO(NO)
      STOP
      END


      SUBROUTINE FRE(SKK,ID)
C
C     CALCULATE FREE-FREE CROSS SECTIONS
C     ADAPTED FROM AUER/MIHALAS CODE
C
      IMPLICIT NONE
C
      PLIST
      ROOT
C
C     LOCAL VARIABLES
C
      INTEGER ID, IJ
      REAL SKK(4,MNJ)
      REAL EX, EXT, EXGF, EXGFT, GF, GFT, HKT, HKTF, HKTFT, SRT
C
C     EXTERNAL PROCEDURE
C
      REAL GFREE, HMFREE
      EXTERNAL GFREE, HMFREE
C
      SRT=1.0/SQRT(TEMP(ID))
      HKT=HK/TEMP(ID)
      DO 10 IJ=1,NJ
C
C     HYDROGEN CONTRIBUTION
C
C     THE EXPONENTIAL TERM REFLECTS THE UPPER-STATE BOUND-FREE
C     CONTRIBUTIONS (BOTH HERE AND BELOW, FOR HELIUM).
C
         HKTF=HKT*FF(IJ,1)
         EX=EXP(HKTF)
C
C     GAUNT FACTOR
C
         GF=GFREE(FREQ(IJ),TEMP(ID))
         EXGF=EX+GF-1.0
         SKK(1,IJ)=SIG(NLH+1,IJ)*SRT*EXGF
C
C     SINGLY IONIZED HELIUM CONTRIBUTION
C     (NO GAUNT FACTOR USED.)
C
         HKTF=HKT*FF(IJ,2)
         EX=EXP(HKTF)
         SKK(2,IJ)=SIG(NLH+1,IJ)*SRT*EX
C
C     IONIZED HELIUM CONTRIBUTION.
C     (NO GAUNT FACTOR USED.)
C
         HKTF=HKT*FF(IJ,3)
         EX=EXP(HKTF)
         SKK(3,IJ)=4.0*SIG(NLH+1,IJ)*SRT*EX
C
C     NEGATIVE HYDROGEN ION FREE-FREE
C
         SKK(4,IJ)=HMFREE(TEMP(ID),FREQ(IJ))
10    CONTINUE
      RETURN
      END
```

```
      REAL FUNCTION GAUNT(NN,QF)
C
C     RETURNS GAUNT FUNCTIONS OF HYDROGENIC ATOMS
C     ADAPTED FROM AUER/MIHALAS CODE
C
      IMPLICIT NONE
C
      REAL C0(11)
      DATA C0/1.2302628,1.1595421,1.1450949,1.1306695,1.1190904,
     :1.1168376,1.1128632,1.1093137,1.1078717,1.1052734,1./
      REAL C1(11)
      DATA C1/-2.9094219E-3,-2.0735860E-3,-1.9366592E-3,-1.3482273E-3,
     :-1.0401085E-3,-8.9466573E-4,-7.4833260E-4,-6.2619148E-4,
     :-5.4837392E-4,-4.434157E-4,0./
      REAL C2(11)
      DATA C2/7.3993579E-6,2.7033384E-6,2.3572356E-6,-4.6949424E-6,
     :-6.9943488E-6,-8.8393133E-6,-1.0244504E-5,-1.1342068E-5,
     :-1.2157943E-5,-1.3235905E-5,0./
      REAL C3(11)
      DATA C3/-8.7356966E-9,0.,0.,2.3548636E-8,
     :2.8496742E-8,3.4696768E-8,
     :3.8595771E-8,4.1477731E-8,4.3796716E-8,4.7003140E-8,0./
      REAL CN1(11)
      DATA CN1/-5.5759885,-1.2709045,-.55936432,-.31190730,-.16051018,
     :-.13075417,-9.5441161E-2,-7.1010560E-2,-5.6046560E-2,
     :-4.7326370E-2,0./
      REAL CN2(11)
      DATA CN2/12.803223,2.1325684,.52471924,.19683564,5.5545091E-2,
     :4.1921183E-2,2.3350812E-2,1.3298411E-2,8.5139736E-3,6.1516856E-3,
     :0./
      REAL CN3(11)
      DATA CN3/0.,-2.0244141,-.23387146,-5.4418565E-2,-8.9182854E-3,
     :-5.5303574E-3,2.2752881E-3,-9.7200274E-4,-4.9576163E-4,
     :-2.9467046E-4,0./
C
      INTEGER NN, N
      REAL QF,X
C
      X=QF/0.299793E15
      N=MIN(NN,11)
      GAUNT=C0(N)+X*(C1(N)+X*(C2(N)+X*C3(N)))+(CN1(N)+(CN2(N)+
     : CN3(N)/X)/X)/X
      RETURN
      END


      REAL FUNCTION GFREE(FRQ,T)
C
C     CALCULATES HYDROGENIC FREE-FREE GAUNT FACTORS
C     IDENTICAL WITH AUER/MIHALAS CODE
C
      IMPLICIT NONE
      REAL FRQ, T, THET, X, C1, C2, C3, C4
      THET=5.040E3/T
      IF (THET.LT.4.0E-2)THET=4.0E-2
      X=FRQ/0.299793E15
      IF (X.GT.1.0)GO TO 10
      IF (X.LT.0.2)X=0.2
      GFREE=(1.0823+2.98E-2/THET)+(6.7E-3+1.12E-2/THET)/X
      RETURN
10    C1=(3.9999187E-3-7.78622889E-5/THET)/THET+1.070192
      C2=(6.64628601E-2-6.1953813E-4/THET)/THET+2.6061249E-1
      C3=(1.3983474E-5/THET+3.7542343E-2)/THET+5.7917786E-1
      C4=3.4169006E-1+1.1852264E-2/THET
      GFREE=((C4/X-C3)/X+C2)/X+C1
      RETURN
      END


      REAL FUNCTION HEGAUNT(IL,A)
C
C     NEUTRAL HELIUM BOUND-FREE GAUNT FACTORS APPROXIMATED FROM
C     STEWART (1978), JACOBS (1974), AND STEWART AND WEBB (1963)
C
      IMPLICIT NONE
C
      INTEGER IL
      REAL A,X
      REAL RYD, CUT(25), CM2(5), CM1(5), C0(5), CP1(5), CP2(5)
C
```

```
      PARAMETER (RYD=3.2880E15)
C
      DATA CUT/32.0, 4*2.40, 20*1.E38/
      DATA CM2/51.15909,3.077759,-.022590913,-.2224690,.071927741/
      DATA CM1/-66.45061,-15.39484,.1931397,.6149698,-.4675104/
      DATA C0/26.40076,25.32980,-1.213830,1.033202,2.053274/
      DATA CP1/.1051292,-12.04806,6.418941,-.1599973,-1.610076/
      DATA CP2/-.0058184229,4.235189,-1.472744,.060324013,.3983392/
C
      X=MIN(CUT(IL),A/RYD)
      IF (IL.LE.5)THEN
         HEGAUNT=C0(IL)+(CM1(IL)+CM2(IL)/X)/X+X*(CP1(IL)+X*CP2(IL))
      ELSE
         HEGAUNT=1.00
      ENDIF
      RETURN
      END


      REAL FUNCTION HMFREE(T,FREQ)
C
C     NEGATIVE HYDROGEN FREE-FREE OPACITY FORMULA FROM GINGERICH (1969)
C
      REAL LAMBDA
C
      THETA=5040./T
      LAMBDA=2.99792E16/FREQ
      C0=0.005366+THETA*(-0.011493+THETA*0.027039)
      C1=1.E-6*(-3.2062+THETA*(11.924-THETA*5.9390))
      C2=1.E-9*(-0.40192+THETA*(7.0355-THETA*0.34592))
      HMFREE=1.E-26*(C0+LAMBDA*(C1+LAMBDA*C2))*T*1.3806E-16
      RETURN
      END


      REAL FUNCTION HMINUS(FREQ)
C
C     NEGATIVE HYDROGEN BOUND-FREE OPACITY APPROXIMATED FROM TABLE
C     OF GELTMAN (1962)
C
      REAL LAMBDA
C
      LAMBDA=2.99792E16/FREQ
      HMINUS=1.E-17*LAMBDA*(1.1038237109E-3+LAMBDA*(2.7762168339E-2+
     : LAMBDA*(1.68296722E-4+LAMBDA*(5.5730765568E-5+LAMBDA*
     : (-1.7683406278E-6+LAMBDA*(2.3013577770E-8+LAMBDA*
     : (-1.5666829428E-10+LAMBDA*(5.4465497521E-13-LAMBDA*
     : 7.5598187662E-16))))))))
      RETURN
      END


      SUBROUTINE NURATE(LHS,RHS,ID)
C
C     CALCULATE THE RATE MATRIX FOR LTE GIVEN TEMPERATURE
C     AND ELECTRON DENSITY.
C
      IMPLICIT NONE
C
      PLIST
      ROOT
      ROOTI
C
      INTEGER I, ID, IL, J, L, U
      REAL LHS(MNNN,MNNN), RHS(MNNN)
      REAL SUM, T, VX, VXT
C
      REAL SB, SBHE1, SBHE2
C
      SB(I,T)=2.0706E-16*GH(I)*EXP(HK*FRQH(I)/T)/T/SQRT(T)
      SBHE1(I,T)=2.0706E-16*GHE1(I)*EXP(HK*FRQHE1(I)/T)/T/SQRT(T)/2.
      SBHE2(I,T)=2.0706E-16*GHE2(I)*EXP(HK*FRQHE2(I)/T)/T/SQRT(T)
C
C     START OF EXECUTABLE STATEMENTS
C
C     PARTITION FUNCTIONS
C
      SUMH(ID)=0.0
      DO 10 IL=1,MQH
         VX=SB(IL,TEMP(ID))
```

```
         SUMH(ID)=SUMH(ID)+VX
10    CONTINUE
      SUMHE1(ID)=0.0
      DO 20 IL=1,MQHE1
         VX=SBHE1(IL,TEMP(ID))
         SUMHE1(ID)=SUMHE1(ID)+VX
20    CONTINUE
      SUMHE2(ID)=0.0
      DO 30 IL=1,MQH*2
         VX=SBHE2(IL,TEMP(ID))
         SUMHE2(ID)=SUMHE2(ID)+VX
30    CONTINUE
      DO 50 I=1,MNNN
         RHS(I)=0.0
         DO 40 J=1,MNNN
            LHS(I,J)=0.0
40       CONTINUE
50    CONTINUE
C
C     PARTICLE CONSERVATION
C
      RHS(4)=-NE(ID)
      LHS(4,INTOT)=-1.
      LHS(4,INPROT)=MU1*(1.+NE(ID)*SUMH(ID))
C
C     CHARGE CONSERVATION EQUATION
C
      RHS(3)=NE(ID)
      LHS(3,INPROT)=1.0
      LHS(3,INHE3)=2.0+NE(ID)*SUMHE2(ID)
C
C     STATISTICAL EQUILIBRIUM
C     SINGLY IONIZED HELIUM
C
      LHS(1,INHE3)=-SBHE2(1,TEMP(ID))*NE(ID)
      LHS(1,INHE2)=1.0
C
C     HELIUM ABUNDANCE EQUATION
C
      LHS(2,INHE2)=NE(ID)*SUMHE1(ID)+1.0
      LHS(2,INHE3)=NE(ID)*SUMHE2(ID)+1.0
      LHS(2,INPROT)=-Y*(1.0+NE(ID)*SUMH(ID))
      RETURN
      END


      SUBROUTINE GENER(ID)
C
C     CALCULATES LTE OPACITIES
C
      IMPLICIT NONE
C
      PLIST
      ROOT
      BLANK
      ROOTI
C
      INTEGER I, ID, IJ, IL, IT, J, L, U
      REAL C, E, SIGMA, T, VX, X0
      REAL EX(MNJ), HKT
      REAL SKK(4,MNJ), SRT, X1
C
      REAL FRE
      EXTERNAL FRE
C
      REAL SB, SBHE1, SBHE2, SBHM
      SB(I,T)=2.0706E-16*GH(I)*EXP(HK*FRQH(I)/T)/T/SQRT(T)
      SBHE1(I,T)=2.0706E-16*GHE1(I)*EXP(HK*FRQHE1(I)/T)/T/SQRT(T)/2.
      SBHE2(I,T)=2.0706E-16*GHE2(I)*EXP(HK*FRQHE2(I)/T)/T/SQRT(T)
      SBHM(T)=2.0706E-16*0.5*EXP(HK*FRQHM/T)/T/SQRT(T)
C
C     START OF EXECUTABLE STATEMENTS
C     FREQUENCY-INDEPENDENT CALLS AND STATEMENTS
C
      HKT=HK/TEMP(ID)
      CALL FRE(SKK,ID)
      DO 10 IJ=1,NJ
         CHI(IJ)=0.0
         EX(IJ)=EXP(-HKT*FREQ(IJ))
10    CONTINUE
      DO 50 IL=1,NLHE1
```

```
         DO 40 IJ=1,NJ
            C=SIGHE1(IL,IJ)*NHE1(IL,ID)
            CHI(IJ)=CHI(IJ)+C
40       CONTINUE
50    CONTINUE
      DO 90 IL=1,NLHE2
         DO 80 IJ=1,NJ
            C=SIGHE2(IL,IJ)*NHE2(IL,ID)
            CHI(IJ)=CHI(IJ)+C
80       CONTINUE
90    CONTINUE
      DO 130 IJ=1,NJ
         C=SKK(2,IJ)*NE(ID)*NE(ID)*NHE3(ID)*SUMHE2(ID)
         CHI(IJ)=CHI(IJ)+C
         C=NHE3(ID)*NE(ID)*SKK(3,IJ)
         CHI(IJ)=CHI(IJ)+C
         C=NPROT(ID)*NE(ID)*SKK(1,IJ)
         CHI(IJ)=CHI(IJ)+C
130   CONTINUE
C
C     NEGATIVE HYDROGEN ION
C
      IF (RULE(1))THEN
         DO 140 IJ=1,NJ
            CHI(IJ)=CHI(IJ)+SKK(4,IJ)*N(1,ID)*NE(ID)
            CHI(IJ)=CHI(IJ)+N(1,ID)*NE(ID)*SBHM(TEMP(ID))*SIGHM(IJ)
 140     CONTINUE
      ENDIF
      DO 200 IL=1,NLH
         DO 190 IJ=1,NJ
            C=SIG(IL,IJ)*N(IL,ID)
            CHI(IJ)=CHI(IJ)+C
190      CONTINUE
200   CONTINUE
      DO 240 IJ=1,NJ
         CHI(IJ)=CHI(IJ)*(1.-EX(IJ))+NE(ID)*SIGE
240   CONTINUE
      RETURN
      END


      SUBROUTINE PUTOUT
C
C     WRITE OUT A COMPLETED MODEL TO DISK
C
      IMPLICIT NONE
C
      PLIST
      ROOT
      ROOTI
C
C     LOCAL VARIABLES
C
      LOGICAL LINES
      INTEGER I, ID, IL
      REAL ABUND(92)
      REAL T
      INTEGER LOWERH(10),UPPERH(10),LOWERHE1(14),UPPERHE1(14)
      INTEGER LOWERHE2(10),UPPERHE2(10)
C
      DATA LOWERH/1,1,1,1,2,2,2,3,3,4/
      DATA UPPERH/2,3,4,5,3,4,5,4,5,5/
      DATA LOWERHE1/1,1,2,2,3,3,4,4,4,4,5,5,5,5/
      DATA UPPERHE1/5,11,4,8,5,11,6,10,12,16,7,9,13,15/
      DATA LOWERHE2/2,2,2,2,3,3,3,4,4,5/
      DATA UPPERHE2/3,4,5,6,4,5,6,5,6,6/
C
C     START OF EXECUTABLE STATEMENTS
C
C     PREVIOUS OUTPUT LINES WERE ALREADY TAKEN CARE OF IN SETUP
C
      WRITE (12,1007)(M(ID),TEMP(ID),NTOT(ID),NE(ID),ID=1,NDEPTH)
C
C     HYDROGEN OCCUPATION NUMBERS
C
      WRITE (12,1005)1
      WRITE (12,1005)0,0,5,10
      WRITE (12,1006)(FRQH(IL),IL=1,5)
      DO 10 IL=1,10
         WRITE (12,1005)LOWERH(IL),UPPERH(IL)
10    CONTINUE
```

```
      WRITE (12,1006)((N(IL,ID),IL=1,5),NPROT(ID),ID=1,NDEPTH)
C
C     HELIUM OCCUPATION NUMBERS
C
      WRITE (12,1005)2
      WRITE (12,1005)0,0,19,14,10,10
      WRITE (12,1006)(FRQHE1(IL),IL=1,19)
      DO 20 IL=1,14
         WRITE (12,1005)LOWERHE1(IL),UPPERHE1(IL)
20    CONTINUE
      WRITE (12,1006)(FRQHE2(IL),IL=1,10)
      DO 30 IL=1,10
         WRITE (12,1005)LOWERHE2(IL),UPPERHE2(IL)
30    CONTINUE
      WRITE (12,1006)((NHE1(IL,ID),IL=1,19),(NHE2(IL,ID),
     :                IL=1,10),NHE3(ID),ID=1,NDEPTH)
      WRITE (12,1005)0
      WRITE (12,1005)0
      RETURN
C
C     FORMAT STATEMENTS
C
1005  FORMAT (16I5)
1006  FORMAT (5E15.7)
1007  FORMAT (4E15.7)
C
      END


      SUBROUTINE SETUP
C
C     READ IN THE APPROXIMATE MODEL MAKE ALL INITIAL CALCULATIONS
C
      IMPLICIT NONE
C
      PLIST
      ROOT
      ROOTI
C
C     LOCAL VARIABLES
C
      CHARACTER*80 HEADER
      INTEGER I, ID, II, IJ, IL, IT, J, NO, N1, N2, N3, N4, N5, NITER
      LOGICAL LINES
      REAL ABUND(92)
      REAL ALPHA, BETA, DUMMY, FCON, GLOG, T, TLINE, VX, Z
C
C     TABLE OF ATOMIC WEIGHTS
C
      REAL WEIGHT(92)
      DATA WEIGHT/1.0,4.0,6.9,9.0,10.8,12.0,14.0,16.0,19.0,20.2,23.0,
     : 24.3,27.0,28.1,31.0,32.1,35.5,39.9,39.1,40.1,45.0,47.9,50.9,52.0,
     : 54.9,55.8,58.9,58.7,63.5,65.4,69.7,72.6,74.9,79.0,79.9,83.8,85.5,
     : 87.6,88.9,91.2,92.9,95.9,98.9,101.1,102.9,106.4,107.9,112.4,
     : 114.8,118.7,121.8,127.6,126.9,131.3,132.9,137.3,138.9,140.1,
     : 140.9,144.2,145.0,150.4,152.0,157.3,158.9,162.5,164.9,167.3,
     : 168.9,173.0,175.0,178.5,180.9,183.9,186.2,190.2,192.2,195.1,
     : 197.0,200.6,204.4,207.2,209.0,209.0,210.0,222.0,223.0,226.0,
     : 227.0,232.0,231.0,238.0/
C
C     TABLE GIVING QUANTUM NUMBER OF ACTIVE ELECTRON OF EACH NEUTRAL
C     HELIUM STATE TREATED BY THE PROGRAM
C
      INTEGER QN(25)
      DATA QN/1,4*2,6*3,8*4,5,6,7,8,9,10/
C
C     EXTERNAL PROCEDURES
C
      REAL EXIT, HEGAUNT, GAUNT, HMINUS
      EXTERNAL EXIT, HEGAUNT, GAUNT, HMINUS
C
C     START OF EXECUTABLE STATEMENTS
C
C     READ COMMENT LINE OF INPUT MODEL
C
      READ (5,1001)HEADER
      WRITE (6,1001)HEADER
      WRITE (12,1001)'GRAY MODEL'
C
C     READ BASIC MODEL PARAMETERS
C
```

```
      READ (5,1002)TEFF,TLINE,GLOG,Y,Z,NITER
      WRITE (12,1002)TEFF,TLINE,GLOG,Y,Z,NITER,5.0,20.0
C
C     CALCULATE EDDINGTON FLUX AND SURFACE GRAVITY FROM PARAMETERS
C
      GRAV=EXP(2.302585093*GLOG)
      WRITE (6,1003)TEFF,TLINE,GLOG,Y,Z,NITER
C
C     READ COMPOSITION, WHICH TAKES THE FORM OF LOG NUMBER RELATIVE
C     TO HYDROGEN=12.0
C
      READ (5,1004)ABUND
      WRITE (12,1004)ABUND
C
C     CALCULATE MEAN MOLECULAR WEIGHT (MU) AND NUCLEI PER PROTON (MU1)
C
      MU=1.0+4.0*Y
      MU1=1.0+Y
      DO 10 I=3,92
         VX=1.E-12*EXP(2.302585093*ABUND(I))*Z
         MU=MU+VX*WEIGHT(I)
         MU1=MU1+VX
10    CONTINUE
      MU=MU/MU1
C
C     READ NUMBER OF DEPTH POINTS AND FREQUENCIES
C     AND THEN READ THE FREQUENCIES
C
      READ (5,1005)NDEPTH,NJ
      READ (5,1006)(FREQ(IJ),IJ=1,NJ)
      WRITE (12,1005)NDEPTH,NJ
      WRITE (12,1006)(FREQ(IJ),IJ=1,NJ)
C
C     CALCULATE FREQUENCY QUADRATURE WEIGHTS
C     NO CORRECTION IS MADE FOR LINES; I.E. IT IS
C     ASSUMED THAT THE LINES ARE NARROW.
C
      DO 20 IJ=1,MNJ
         WT(IJ)=0.0
20    CONTINUE
C
C     READ IN FREQUENCY INTERVAL NUMBER AND RULE
C
30       READ (5,1005)NO,N1
         WRITE (12,1005)NO,N1
         IF (NO.EQ.2)THEN
C
C     TRAPEZOIDAL RULE:  N1 IS NUMBER OF FIRST FREQUENCY
C
            VX=0.5*ABS(FREQ(N1)-FREQ(N1+1))
            WT(N1)=WT(N1)+VX
            WT(N1+1)=WT(N1+1)+VX
            GO TO 30
         ELSE IF (NO.EQ.3)THEN
C
C     SIMPSON'S RULE; N1 IS NUMBER OF CENTRAL FREQUENCY
C     **NOTE THAT NO CHECK IS MADE TO BE SURE THAT THE
C     TWO FREQUENCY SUBINTERVALS ARE EQUAL, AS THEY NEED
C     TO BE.**
C
            VX=ABS(FREQ(N1+1)-FREQ(N1-1))/6.0
            WT(N1+1)=WT(N1+1)+VX
            WT(N1-1)=WT(N1-1)+VX
            WT(N1)=WT(N1)+4.0*VX
            GO TO 30
         ELSE IF (NO.NE.0)THEN
C
C     PROGRAM DOESN'T RECOGNIZE THE RULE
C
            WRITE (6,*) 'ERROR IN FREQUENCY QUADRATURE RULE'
            CALL EXIT(1)
         ENDIF
C
C     READ OPACITY RULE
C
      READ (5,1005)RULE
      WRITE (12,1005)RULE
C
C     REMAINDER OF DECK IS IGNORED
C
```

```
C     CALCULATE CONTINUUM ABSORPTION COEFFICIENTS
C
      DO 70 IJ=1,NJ
C
C     BOUND-FREE ABSORPTION
C
         FCON=FREQ(IJ)
         DO 40 IL=1,NLHE2
            SIGHE2(IL,IJ)=0.0
            IF (FCON.GT.FRQHE2(IL))SIGHE2(IL,IJ)=45.04E29*GAUNT(IL,
     :                             FCON/4.0)/FLOAT(IL)**5/FCON**3
40       CONTINUE
         DO 50 IL=1,NLHE1
            SIGHE1(IL,IJ)=0.0
            IF (FCON.GT.FRQHE1(IL))SIGHE1(IL,IJ)=2.815E29*
     :                  HEGAUNT(IL,FCON)/FLOAT(QN(IL))**5/FCON**3
50       CONTINUE
C
C     SINGLY-IONIZED HELIUM IS HYDROGENIC AND THUS USES SCALED
C     HYDROGEN EXPRESSIONS.
C
         DO 60 IL=1,NLH
            SIG(IL,IJ)=0.0
            IF (FCON.GT.FRQH(IL))SIG(IL,IJ)=2.815E29*GAUNT(IL,FCON)/
     :                  FLOAT(IL)**5/FCON**3
60       CONTINUE
C
C     FREE-FREE ABSORPTION
C
         SIG(NLH+1,IJ)=3.69E8/FCON**3
70    CONTINUE
C
C     NEGATIVE HYDROGEN ION
C
      IF (RULE(1))THEN
         DO 80 IJ=1,NJ
            SIGHM(IJ)=0.
            IF (FREQ(IJ).GT.FRQHM)SIGHM(IJ)=MAX(0.,HMINUS(FREQ(IJ)))
80       CONTINUE
      ENDIF
C
C     DETERMINE CUTOFF FREQUENCIES FOR FREE-FREE OPACITY.
C     THESE ALLOW THE INCLUSION OF BOUND LEVELS ABOVE THOSE
C     ACCOUNTED FOR EXPLICITLY AS PART OF THE FREE-FREE OPACITY.
C
      DO 90 IJ=1,NJ
         FF(IJ,1)=MIN(FRQH(NLH)*(MQH/(MQH+1))**2,FREQ(IJ))
         FF(IJ,2)=MIN(FRQHE1(NLHE1)*(MQHE1/(MQHE1+1))**2,FREQ(IJ))
         FF(IJ,3)=MIN(FRQHE2(NLHE2)*(2*MQH/(2*MQH+1))**2,FREQ(IJ))
90    CONTINUE
      RETURN
C
1001  FORMAT (A80)
1002  FORMAT (2F9.0,F8.2,F8.3,F8.5,I5,F8.5,F8.3)
1003  FORMAT (' EFFECTIVE TEMPERATURE = ',F9.0,/,
     :        ' LINE TEMPERATURE      = ',F9.0,/,
     :        ' LOG SURFACE GRAVITY   = ',F8.2,/,
     :        ' HELIUM/HYDROGEN       = ',F8.3,/,
     :        ' Z SCALE FACTOR        = ',F8.5,/,
     :        ' ITERATION LIMIT       = ',I5,/)
1004  FORMAT (8F10.6)
1005  FORMAT (16I5)
1006  FORMAT (5E15.7)
C
      END


      BLOCK DATA TABLES
C
C     CONTAINS ALL DATA STATEMENTS FOR COMMON BLOCKS, IN ACCORDANCE
C     WITH THE ANSI STANDARD
C
      PLIST
      ROOTI
C
C     HYDROGEN STATISTICAL WEIGHTS
C
      DATA GH/2.,8.,18.,32.,50.,72.,98.,128.,162.,200.,242.,288.,338.,
     : 392.,450.,512./
C
C     NEUTRAL HELIUM STATISTICAL WEIGHTS
```

```
C
      DATA GHE1/1.,3.,1.,9.,3.,3.,1.,9.,5.,15.,3.,3.,1.,9.,5.,15.,3.,
     : 21.,7.,100.,144.,196.,256.,324.,400.,484.,576.,676.,784.,900.,
     : 1024./
C
C     IONIZED HELIUM STATISTICAL WEIGHTS
C
      DATA GHE2/2.,8.,18.,32.,50.,72.,98.,128.,162.,200.,242.,288.,338.,
     : 392.,450.,512.,578.,648.,722.,800.,882.,968.,1058.,1152.,1250.,
     : 1352.,1458.,1568.,1682.,1800.,1922.,2048./
C
C     NEGATIVE HYDROGEN ION IONIZATION FREQUENCY
C
      DATA FRQHM/1.874E14/
C
C     HYDROGEN IONIZATION FREQUENCIES
C
      DATA FRQH/3.28799E15,0.821997E15,0.365332E15,0.205499E15,
     :0.131519E15,0.0913329E15,0.0671018E15,0.0513748E15,
     :0.0405924E15,0.0328799E15,0.0271735E15,0.0228333E15,
     :0.0194556E15,0.0167755E15,0.0146133E15,0.0128437E15/
C
C     NEUTRAL HELIUM IONIZATION FREQUENCIES
C
      DATA FRQHE1/5.94520E15,1.15305E15,0.957439E15,0.876230E15,
     :0.811774E15,0.451896E15,0.400142E15,0.381976E15,0.362850E15,
     :0.366032E15,0.362480E15,0.240134E15,0.217774E15,0.212670E15,
     :0.202689E15,0.205704E15,0.202057E15,0.202703E15,0.199689E15,
     :0.131520E15,0.0913331E15,0.0671018E15,0.0513748E15,
     :0.0405924E15,0.0328799E15,0.0271735E15,0.0228333E15,
     :0.0194556E15,0.0167755E15,0.0146133E15,0.0128437E15/
C
C     IONIZED HELIUM IONIZATION FREQUENCIES
C
      DATA FRQHE2/13.1520E15,3.28799E15,1.46133E15,0.821997E15,
     :0.526078E15,0.365332E15,0.268407E15,0.205499E15,0.162370E15,
     :0.131519E15,0.108694E15,0.0913329E15,0.0778222E15,
     :0.0671018E15,0.0584532E15,0.0513748E15,0.0455085E15,
     :0.0405924E15,0.0364320E15,0.0328799E15,0.0298230E15,
     :0.0271735E15,0.0248619E15,0.0228333E15,0.0210431E15,
     :0.0194556E15,0.0180411E15,0.0167755E15,0.0156385E15,
     :0.0146133E15,0.0136857E15,0.0128437E15/
C
      END


      REAL FUNCTION FNT(DUM)
C
C     APPROXIMATE T-TAU RELATIONSHIP
C
      PLIST
      ROOT
C
      REAL DUM
C
      FNT=TEFF*SQRT(SQRT(.75*(.710+DUM-.1331*EXP(-3.4488*DUM))))
      RETURN
      END


      SUBROUTINE STATE(ID,KAP)
C
C     MAIN ROUTINE FOR CALCULATING STATE VARIABLES
C
      IMPLICIT NONE
C
C     COMMON INCLUSION
C
      PLIST
      ROOT
      ROOTI
      BLANK
C
C     LOCAL VARIABLES
C
      INTEGER I, II, IJ, IL, J, K, ID
      REAL ERR, DEL, CLHS(MNNN,MNNN), RHO
      REAL RHS(MNNN), LHS(MNNN,MNNN), T, VX
      REAL ANS(MNNN), A(MNNN), KAP, BSUM
C
```

```
      REAL LINSLV, MATINV, NURATE, GENER
      EXTERNAL LINSLV, MATINV, NURATE, GENER
C
      REAL SB, SBHE1, SBHE2
      SB(I,T)=2.0706E-16*GH(I)*EXP(HK*FRQH(I)/T)/T/SQRT(T)
      SBHE1(I,T)=2.0706E-16*GHE1(I)*EXP(HK*FRQHE1(I)/T)/T/SQRT(T)/2.
      SBHE2(I,T)=2.0706E-16*GHE2(I)*EXP(HK*FRQHE2(I)/T)/T/SQRT(T)
C
C
C     START OF EXECUTABLE STATEMENTS
C
C     INITIAL GUESS OF ELECTRON DENSITY
C
      IF (ID.EQ.1)THEN
         NE(ID)=1.
      ELSE
         NE(ID)=NE(ID-1)
      ENDIF
      DO 40 II=1,30
         CALL NURATE(LHS,RHS,ID)
         DO 20 I=1,MNNN
            DO 10 J=1,MNNN
               CLHS(I,J)=LHS(I,J)
10          CONTINUE
20       CONTINUE
         CALL LINSLV(CLHS,RHS,ANS,MNNN,MNNN)
         A(4)=-1.-ANS(INPROT)*MU1*SUMH(ID)
         A(3)=1.-ANS(INHE3)*SUMHE2(ID)
         A(1)=ANS(INHE3)*SBHE2(1,TEMP(ID))
         A(2)=-ANS(INHE2)*SUMHE1(ID)-ANS(INHE3)*SUMHE2(ID)-
     :         Y*ANS(INPROT)*SUMH(ID)
         CALL MATINV(LHS,MNNN,MNNN)
         DEL=0.
         DO 30 I=1,4
            DEL=DEL+LHS(4,I)*A(I)
30       CONTINUE
         ERR=NTOT(ID)-ANS(4)
         NE(ID)=NE(ID)+ERR/DEL
         IF (ABS(ERR)/NTOT(ID).LT.1.E-5)GO TO 50
40    CONTINUE
      WRITE (6,*)' WARNING - STATE ITERATIONS DO NOT CONVERGE AT ',ID
50    NPROT(ID)=ANS(INPROT)
      DO 60 IL=1,NLH
         N(IL,ID)=NPROT(ID)*NE(ID)*SB(IL,TEMP(ID))
60    CONTINUE
      NHE3(ID)=ANS(INHE3)
      DO 70 IL=1,NLHE2
         NHE2(IL,ID)=NHE3(ID)*NE(ID)*SBHE2(IL,TEMP(ID))
70    CONTINUE
      DO 80 IL=1,NLHE1
         NHE1(IL,ID)=NHE2(1,ID)*NE(ID)*SBHE1(IL,TEMP(ID))
80    CONTINUE
C
C     CALCULATE OPACITIES AND SOURCE FUNCTION
C
      CALL GENER(ID)
C
C     FORM ROSSELAND MEAN
C
      KAP=0.
      BSUM=0.
      DO 90 IJ=1,NJ
         VX=WT(IJ)*FREQ(IJ)**3/(EXP(HK*FREQ(IJ)/TEMP(ID))-1.)
         BSUM=BSUM+VX
         KAP=KAP+VX/CHI(IJ)
90    CONTINUE
      RHO=MU*(NTOT(ID)-NE(ID))/6.022E23
      KAP=BSUM/KAP/RHO
      RETURN
      END


      SUBROUTINE LINSLV(A,B,X,N,NR)
C
C     SOLVE SET OF LINEAR EQUATIONS
C     FROM AUER/MIHALAS CODE
C
      IMPLICIT NONE
C
      INTEGER NEQN
      PARAMETER (NEQN=4)
```

```
C
      INTEGER J, ITEMP, JP1, N, I, NR, K, IP1, MR1, KP1, R, P(NEQN)
      REAL M, SUM, D(NEQN)
      REAL A(NR,NR), B(NR), X(NR)
C
      REAL EXIT
      EXTERNAL EXIT
C
      IF (N.GT.NEQN)THEN
         WRITE (6,*)'N TOO BIG IN LINSLV; = ',N
         CALL EXIT(1)
      ENDIF
      DO 70 R=1,N
         DO 10 K=1,N
            D(K)=A(K,R)
10       CONTINUE
         MR1=R-1
         IF (MR1.LT.1)GO TO 40
         DO 30 J=1,MR1
            ITEMP=P(J)
            A(J,R)=D(ITEMP)
            D(ITEMP)=D(J)
            JP1=J+1
            DO 20 I=JP1,N
               D(I)=D(I)-A(I,J)*A(J,R)
20          CONTINUE
30       CONTINUE
40       M=ABS(D(R))
         P(R)=R
         DO 50 I=R,N
            IF (M.GE.ABS(D(I)))GO TO 50
            P(R)=I
            M=ABS(D(I))
50       CONTINUE
         ITEMP=P(R)
         A(R,R)=D(ITEMP)
         D(ITEMP)=D(R)
         MR1=R+1
         IF (MR1.GT.N)GO TO 80
         DO 60 I=MR1,N
            A(I,R)=D(I)/A(R,R)
60       CONTINUE
70    CONTINUE
80    DO 100 I=1,N
         ITEMP=P(I)
         X(I)=B(ITEMP)
         B(ITEMP)=B(I)
         IP1=I+1
         IF (IP1.GT.N)GO TO 110
         DO 90 J=IP1,N
            B(J)=B(J)-A(J,I)*X(I)
90       CONTINUE
100   CONTINUE
110   DO 140 I=1,N
         K=N-I+1
         SUM=0.0
         KP1=K+1
         IF (KP1.GT.N)GO TO 130
         DO 120 J=KP1,N
            SUM=SUM+A(K,J)*X(J)
120      CONTINUE
130      X(K)=(X(K)-SUM)/A(K,K)
140   CONTINUE
      RETURN
      END


      SUBROUTINE MATINV(A,N,NR)
C
C     INVERT MATRIX IN PLACE
C     FROM AUER/MIHALAS CODE
C
      IMPLICIT NONE
C
      INTEGER N, K, NR, I, L, KO, II, J, JJ
      REAL A(NR,NR), DIV, SUM
C
      DO 60 I=2,N
         DIV=A(1,1)
         A(I,1)=A(I,1)/DIV
         DO 30 J=2,I-1
```

```
            DIV=A(J,J)
            SUM=0.
            DO 10 L=1,J-1
               SUM=SUM+A(I,L)*A(L,J)
10          CONTINUE
            A(I,J)=(A(I,J)-SUM)/DIV
30       CONTINUE
         DO 50 J=I,N
            SUM=0.
            DO 40 L=1,I-1
               SUM=SUM+A(I,L)*A(L,J)
40          CONTINUE
            A(I,J)=A(I,J)-SUM
50       CONTINUE
60    CONTINUE
      DO 100 I=N,2,-1
         DO 90 J=I-1,1,-1
            SUM=0.
            DO 70 K=J+1,I-1
               SUM=SUM+A(I,K)*A(K,J)
70          CONTINUE
            A(I,J)=-A(I,J)-SUM
90       CONTINUE
100   CONTINUE
      A(N,N)=1.0/A(N,N)
      DO 140 I=N-1,1,-1
         DIV=A(I,I)
         DO 120 J=N,I+1,-1
            SUM=0.
            DO 110 K=I+1,J
               SUM=SUM+A(I,K)*A(K,J)
110         CONTINUE
            A(I,J)=-SUM/DIV
120      CONTINUE
         A(I,I)=1.0/A(I,I)
140   CONTINUE
      DO 190 I=1,N
         DO 160 J=1,I-1
            SUM=0.0
            DO 150 K=I,N
               SUM=SUM+A(I,K)*A(K,J)
150         CONTINUE
            A(I,J)=SUM
160      CONTINUE
         DO 180 J=I,N
            SUM=A(I,J)
            DO 170 K=J+1,N
               SUM=SUM+A(I,K)*A(K,J)
170         CONTINUE
            A(I,J)=SUM
180      CONTINUE
190   CONTINUE
      RETURN
      END
```

B. Program LTE

LTE is listed in its entirety, except for those program units that are essentially identical to the ones in GRAY. These are:

FRE

GAUNT

GENER

HEGAUNT

HMINUS

LINSLV

MATINV

```
C      PROGRAM LTE
C
C      COMMON BLOCK MACROS FOR ALL GLOBAL DATA IN PROGRAM
C      (CRAY PRECOMPILER)
C
C      BASIC PARAMETERS
C
       CLICHE PLIST
       INTEGER MNDEPTH, MNJC, MNTRH, MNTRHE1, MNTRHE2, MQH, MQHE1, NLH
       INTEGER NLHE1, NLHE2, NQUAD
       PARAMETER (MNDEPTH=70, MNJC=105, MNTRH=10, MNTRHE1=14)
       PARAMETER (MNTRHE2=10, MQH=16, MQHE1=31, NLH=10)
       PARAMETER (NLHE1=25, NLHE2=15, NQUAD=7)
C
       INTEGER MNJ
       PARAMETER (MNJ=MNJC+NQUAD*(MNTRH+MNTRHE1+MNTRHE2))
C
       REAL CC, DELQUAD, EMASS, ESU, HP, KB, MHYD, PI
       PARAMETER (CC=2.99792E10, DELQUAD=0.6, EMASS=9.10953E-28)
       PARAMETER (ESU=4.80325E-10, HP=6.62618E-27, KB=1.38066E-16)
       PARAMETER (MHYD=1.66058E-24, PI=3.141592654)
C
       REAL ACCOF, BBCOF, DOPCOF, HK, HYDCOF, PIE2MC, SCOF, SIGE
       PARAMETER (ACCOF=2.074E-16, BBCOF=2.*HP/(CC*CC), DOPCOF=4.286E-7)
       PARAMETER (HK=HP/KB, HYDCOF=4.*PI/CC)
       PARAMETER (PIE2MC=PI*ESU*ESU/(EMASS*CC), SCOF=4.*PI/HP)
       PARAMETER (SIGE=8.*PI*ESU*ESU*ESU*ESU/
      :                (3.*EMASS*EMASS*CC*CC*CC*CC))
C
       ENDCLICHE

       CLICHE BLANK
C
       REAL CHI(MNJ), ETA(MNJ), FK(MNDEPTH), RAD(MNDEPTH)
C
       COMMON //CHI, ETA, FK, RAD
C
       ENDCLICHE

       CLICHE ROOT
C
       INTEGER ITPTRH, ITPTRHE1, ITPTRHE2, NDEPTH, NJ, NTRH, NTRHE1
       INTEGER NTRHE2
C
       LOGICAL RULE(2)
C
       REAL DSHDT(MNDEPTH), DSHE1(MNDEPTH), DSHE2(MNDEPTH), FF(MNJ,3)
       REAL FH, FREQ(MNJ), GRAV, HO, M(MNDEPTH), MU, MU1
       REAL NE(MNDEPTH), N(NLH,MNDEPTH), NHE1(NLHE1,MNDEPTH)
       REAL NHE2(NLHE2,MNDEPTH), NHE3(MNDEPTH)
       REAL NM(MNDEPTH), NPROT(MNDEPTH), NTOT(MNDEPTH), SIGHM(MNJ)
       REAL SIG(NLH+1,MNJ), SIGHE1(NLHE1,MNJ), SIGHE2(NLHE2,MNJ)
       REAL SUMH(MNDEPTH), SUMHE1(MNDEPTH), SUMHE2(MNDEPTH)
       REAL TEMP(MNDEPTH), TLINE, WT(MNJ), Y
C
       COMMON /ROOT/ITPTRH, ITPTRHE1, ITPTRHE2, NDEPTH, NJ, NTRH,
      : NTRHE1, NTRHE2, RULE, DSHDT, DSHE1, DSHE2, FF, FH, FREQ, GRAV,
      : HO, M, MU, MU1, N, NE, NHE1, NHE2, NHE3, NM, NPROT, NTOT, SIG,
      : SIGHE1, SIGHE2, SIGHM, SUMH, SUMHE1, SUMHE2, TEMP, TLINE, WT, Y
C
       ENDCLICHE
       CLICHE ROOTI
C
       INTEGER ITRH(5,5), ITRHE1(19,19), ITRHE2(10,10), LOWERH(MNTRH)
       INTEGER LOWERHE1(MNTRHE1), LOWERHE2(MNTRHE2), LOWH(MNJ)
       INTEGER LOWHE1(MNJ), LOWHE2(MNJ), UPH(MNJ), UPHE1(MNJ), UPHE2(MNJ)
       INTEGER UPPERH(MNTRH), UPPERHE1(MNTRHE1), UPPERHE2(MNTRHE2)
       REAL FRQHM, FRQH(MQH), FRQHE1(MQHE1), FRQHE2(2*MQH), GH(2*MQH)
       REAL GHE1(MQHE1), OSCH(10), OSCHE1(34), OSCHE2(45)
C
       COMMON /ROOTI/ITRH, ITRHE1, ITRHE2, LOWERH, LOWERHE1, LOWERHE2,
      : LOWH, LOWHE1, LOWHE2, UPH, UPHE1, UPHE2, UPPERH, UPPERHE1,
      : UPPERHE2, FRQHM, FRQH, FRQHE1, FRQHE2, GH, GHE1, OSCH, OSCHE1,
      : OSCHE2
C
       ENDCLICHE

       CLICHE SEGH
C
```

```
C     POINTERS FOR STATE MATRIX
C
      INTEGER INHE2, INHE3, INPROT, INTOT, MNNN
C
      PARAMETER (INHE2=1, INHE3=2, INPROT=3, INTOT=4, MNNN=4)

      REAL DN(MNNN,MNDEPTH), DT(MNNN,MNDEPTH)
      COMMON /SEGH/DN, DT

      ENDCLICHE

      PROGRAM LTE
C
C     CALCULATE AN LTE ATMOSPHERE OF HYDROGEN AND HELIUM.
C
C     THE PROGRAM UTILIZES THE OUTPUT MODEL OF PROGRAM GRAY
C     AS A FIRST APPROXIMATION.
C
      IMPLICIT NONE
C
C     LOCAL VARIABLES
C
      INTEGER I, NITER
      REAL ERR, TEFF, Z
C
C     EXTERNAL PROCEDURES
C
      REAL EXIT, LINK, LTESTEP, PUTOUT, SETUP
      EXTERNAL EXIT, LINK, LTESTEP, PUTOUT, SETUP
C
C     START OF EXECUTABLE STATEMENTS
C
C     THE NEXT LINE MAY BE DELETED ON NON-CRAY IMPLEMENTATIONS
C
C     THE FILE input HOLDS THE FIRST APPROXIMATION MODEL.
C     THE FILE output RECEIVES THE CALCULATED MODEL.
C     THE FILE monitor RECEIVES VARIOUS PROGRAM-GENERATED REMARKS.
C
      CALL LINK("UNIT5=(input,OPEN,TEXT),UNIT12=(output,CREATE,TEXT),
     : UNIT6=(monitor,CREATE,TEXT)//")
C
C     READ THE FIRST APPROXIMATION MODEL AND CALCULATE QUANTITIES
C     DEPENDENT ONLY ON THE FREQUENCY MESH AND IONIZATION ENERGIES.
C
      CALL SETUP(NITER,TEFF,Z)
C
C     MAIN LOOP:
C
      DO 10 I=1,NITER
C
C        PERFORM AN LTE COMPLETE LINEARIZATION ITERATION
C
         CALL LTESTEP(ERR)
C
C     TELL THE USER HOW WE'RE COMING ALONG.
C
         WRITE (6,1008)I,ERR
C
C     SUCCESS!  EXIT THE LOOP AND SAVE THE POLISHED MODEL.
C
         IF (ERR.LT.1.E-4)GO TO 20
10    CONTINUE
C
C     SAVE THE COMPLETED MODEL
C
20    CALL PUTOUT(TEFF,Z)
      CALL EXIT(0)
C
C     FORMAT STATEMENTS
C
1008  FORMAT (' ITERATION',I3,' COMPLETE WITH ERR = ',F9.4)
C
      END

      SUBROUTINE EDDFAC(IJ)
C
C     SOLVE EQUATION OF TRANSFER FOR A GIVEN SOURCE FUNCTION AND
C     CALCULATE VARIABLE EDDINGTON FACTORS
C
      IMPLICIT NONE
```

```
      PLIST
      BLANK
      ROOT
C
      INTEGER I, ID, IJ, IMU, J
      REAL A(3), AA, B(3,3), BB, C(3), CCC, D(3,3,MNDEPTH), DD(MNDEPTH)
      REAL DIV, DP, DT(MNDEPTH), DT0, DP, EH, EJ, EK, L, NU(MNDEPTH), P
      REAL PSI(3,MNDEPTH), Q(3), QQ, QS, RO
C
C     GAUSSIAN INTEGRATION ORDINATES AND WEIGHTS
C
      REAL AMU(3),WTMU(3)
      DATA WTMU/.2777777777777778,.4444444444444444,
     : .2777777777777778/,AMU/.887298334620742,.5,.112701665379258/
C
C     START OF EXECUTABLE STATEMENTS
C
C
C     CALCULATE OPTICAL DEPTH SCALE
C
         DO 20 ID=1,NDEPTH-1
            DT(ID)=0.5*(CHI(ID+1)/NM(ID+1)+CHI(ID)/NM(ID))*
     :                 M(ID+1)/MHYD
20       CONTINUE
C
C        UPPER BOUNDARY CONDITIONS
C
         QQ=ETA(1)/CHI(1)
         QS=NE(1)*SIGE/CHI(1)
         DO 40 I=1,3
            PSI(I,1)=0.
            DIV=0.5*DT(1)/AMU(I)
            Q(I)=DIV*QQ
            C(I)=AMU(I)/DT(1)
            DO 30 J=1,3
               B(I,J)=-WTMU(J)*QS*DIV
30          CONTINUE
            B(I,I)=B(I,I)+1.0+C(I)+DIV
40       CONTINUE
C
C     INVERT MATRIX (DONE EXPLICITLY SINCE SMALL MATRIX)
C
         B(2,1)=B(2,1)/B(1,1)
         B(2,2)=B(2,2)-B(2,1)*B(1,2)
         B(2,3)=B(2,3)-B(2,1)*B(1,3)
         B(3,1)=B(3,1)/B(1,1)
         B(3,2)=(B(3,2)-B(3,1)*B(1,2))/B(2,2)
         B(3,3)=B(3,3)-B(3,1)*B(1,3)-B(3,2)*B(2,3)
         B(3,2)=-B(3,2)
         B(3,1)=-B(3,1)-B(3,2)*B(2,1)
         B(2,1)=-B(2,1)
         B(3,3)=1.0/B(3,3)
         B(2,3)=-B(2,3)*B(3,3)/B(2,2)
         B(2,2)=1.0/B(2,2)
         B(1,3)=-(B(1,2)*B(2,3)+B(1,3)*B(3,3))/B(1,1)
         B(1,2)=-B(1,2)*B(2,2)/B(1,1)
         B(1,1)=1.0/B(1,1)
         B(1,1)=B(1,1)+B(1,2)*B(2,1)+B(1,3)*B(3,1)
         B(1,2)=B(1,2)+B(1,3)*B(3,2)
         B(2,1)=B(2,2)*B(2,1)+B(2,3)*B(3,1)
         B(2,2)=B(2,2)+B(2,3)*B(3,2)
         B(3,1)=B(3,3)*B(3,1)
         B(3,2)=B(3,3)*B(3,2)
         DO 60 I=1,3
            DO 50 J=1,3
               D(I,J,1)=B(I,J)*C(J)
               PSI(I,1)=PSI(I,1)+B(I,J)*Q(J)
50          CONTINUE
60       CONTINUE
C
C        NORMAL DEPTH POINTS
C
         DO 130 ID=2,NDEPTH-1
            DT0=0.5*(DT(ID-1)+DT(ID))
            QQ=ETA(ID)/CHI(ID)
            QS=NE(ID)*SIGE/CHI(ID)
            DO 80 I=1,3
               A(I)=AMU(I)**2/DT(ID-1)/DT0
               C(I)=AMU(I)**2/DT(ID)/DT0
               Q(I)=QQ+A(I)*PSI(I,ID-1)
```

```
            DO 70 J=1,3
               B(I,J)=-WTMU(J)*QS
70          CONTINUE
            B(I,I)=B(I,I)+1.+A(I)+C(I)
80       CONTINUE
         DO 100 I=1,3
            DO 90 J=1,3
               B(I,J)=B(I,J)-A(I)*D(I,J,ID-1)
90          CONTINUE
100      CONTINUE
         B(2,1)=B(2,1)/B(1,1)
         B(2,2)=B(2,2)-B(2,1)*B(1,2)
         B(2,3)=B(2,3)-B(2,1)*B(1,3)
         B(3,1)=B(3,1)/B(1,1)
         B(3,2)=(B(3,2)-B(3,1)*B(1,2))/B(2,2)
         B(3,3)=B(3,3)-B(3,1)*B(1,3)-B(3,2)*B(2,3)
         B(3,2)=-B(3,2)
         B(3,1)=-B(3,1)-B(3,2)*B(2,1)
         B(2,1)=-B(2,1)
         B(3,3)=1.0/B(3,3)
         B(2,3)=-B(2,3)*B(3,3)/B(2,2)
         B(2,2)=1.0/B(2,2)
         B(1,3)=-(B(1,2)*B(2,3)+B(1,3)*B(3,3))/B(1,1)
         B(1,2)=-B(1,2)*B(2,2)/B(1,1)
         B(1,1)=1.0/B(1,1)
         B(1,1)=B(1,1)+B(1,2)*B(2,1)+B(1,3)*B(3,1)
         B(1,2)=B(1,2)+B(1,3)*B(3,2)
         B(2,1)=B(2,2)*B(2,1)+B(2,3)*B(3,1)
         B(2,2)=B(2,2)+B(2,3)*B(3,2)
         B(3,1)=B(3,3)*B(3,1)
         B(3,2)=B(3,3)*B(3,2)
         DO 120 I=1,3
            PSI(I,ID)=0.
            DO 110 J=1,3
               PSI(I,ID)=PSI(I,ID)+B(I,J)*Q(J)
               D(I,J,ID)=B(I,J)*C(J)
110         CONTINUE
120      CONTINUE
130   CONTINUE
C
C     LOWER BOUNDARY CONDITIONS
C
      RO=BBCOF*FREQ(IJ)**3
      P=RO/(EXP(HK*FREQ(IJ)/TEMP(NDEPTH))-1.)
      DP=RO/(EXP(HK*FREQ(IJ)/TEMP(NDEPTH-1))-1.)
      DP=(P-DP)/DT(NDEPTH-1)
      DO 150 I=1,3
         A(I)=AMU(I)/DT(NDEPTH-1)
         Q(I)=P+AMU(I)*DP+A(I)*PSI(I,NDEPTH-1)
         DO 140 J=1,3
            B(I,J)=0.
140      CONTINUE
         B(I,I)=1.+A(I)
150   CONTINUE
      DO 170 I=1,3
         DO 160 J=1,3
            B(I,J)=B(I,J)-A(I)*D(I,J,NDEPTH-1)
160      CONTINUE
170   CONTINUE
      B(2,1)=B(2,1)/B(1,1)
      B(2,2)=B(2,2)-B(2,1)*B(1,2)
      B(2,3)=B(2,3)-B(2,1)*B(1,3)
      B(3,1)=B(3,1)/B(1,1)
      B(3,2)=(B(3,2)-B(3,1)*B(1,2))/B(2,2)
      B(3,3)=B(3,3)-B(3,1)*B(1,3)-B(3,2)*B(2,3)
      B(3,2)=-B(3,2)
      B(3,1)=-B(3,1)-B(3,2)*B(2,1)
      B(2,1)=-B(2,1)
      B(3,3)=1.0/B(3,3)
      B(2,3)=-B(2,3)*B(3,3)/B(2,2)
      B(2,2)=1.0/B(2,2)
      B(1,3)=-(B(1,2)*B(2,3)+B(1,3)*B(3,3))/B(1,1)
      B(1,2)=-B(1,2)*B(2,2)/B(1,1)
      B(1,1)=1.0/B(1,1)
      B(1,1)=B(1,1)+B(1,2)*B(2,1)+B(1,3)*B(3,1)
      B(1,2)=B(1,2)+B(1,3)*B(3,2)
      B(2,1)=B(2,2)*B(2,1)+B(2,3)*B(3,1)
      B(2,2)=B(2,2)+B(2,3)*B(3,2)
      B(3,1)=B(3,3)*B(3,1)
      B(3,2)=B(3,3)*B(3,2)
```

```
C
C       BACKSUBSTITUTION TO DETERMINE SPECIFIC INTENSITIES
C
        DO 190 I=1,3
           PSI(I,NDEPTH)=0.
           DO 180 J=1,3
              PSI(I,NDEPTH)=PSI(I,NDEPTH)+B(I,J)*Q(J)
180        CONTINUE
190     CONTINUE
        DO 230 ID=NDEPTH,1,-1
           EJ=0.
           EK=0.
           DO 220 I=1,3
              IF (ID.EQ.NDEPTH) GO TO 210
              DO 200 J=1,3
                 PSI(I,ID)=PSI(I,ID)+D(I,J,ID)*PSI(J,ID+1)
200           CONTINUE
C
C             CALCULATE MEAN INTENSITY AND SECOND MOMENT
C
210           EJ=EJ+WTMU(I)*PSI(I,ID)
              EK=EK+WTMU(I)*AMU(I)**2*PSI(I,ID)
220        CONTINUE
C
C          CALCULATE VARIABLE EDDINGTON FACTOR
C
           FK(ID)=EK/EJ
230     CONTINUE
        FK(NDEPTH)=1./3.
        EH=0.0
        DO 240 I=1,3
           EH=EH+WTMU(I)*AMU(I)*PSI(I,1)
240     CONTINUE
        FH=EH/EJ
C
C     NOW RECALCULATE MEAN INTENSITY USING EDDINGTON FACTORS
C
        DIV=0.5*DT(1)
        BB=FK(1)/DT(1)+FH+DIV*(1.-NE(1)*SIGE/CHI(1))
        CCC=FK(2)/DT(1)
        L=DIV*ETA(1)/CHI(1)
        NU(1)=L/BB
        DD(1)=CCC/BB
        DO 250 ID=2,NDEPTH-1
           DTO=0.5*(DT(ID-1)+DT(ID))
           AA=FK(ID-1)/(DT(ID-1)*DTO)
           CCC=FK(ID+1)/(DT(ID)*DTO)
           BB=FK(ID)*(1./DT(ID-1)+1./DT(ID))/DTO+1.0-
     :                NE(ID)*SIGE/CHI(ID)
           L=ETA(ID)/CHI(ID)
           BB=BB-AA*DD(ID-1)
           DD(ID)=CCC/BB
           NU(ID)=(L+AA*NU(ID-1))/BB
250     CONTINUE
        BB=FK(NDEPTH)/DT(NDEPTH-1)+0.5
        AA=FK(NDEPTH-1)/DT(NDEPTH-1)
        BB=BB-AA*DD(NDEPTH-1)
        L=0.5*P+DP/3.0
        RAD(ID)=(L+AA*NU(NDEPTH-1))/BB
        DO 260 ID=NDEPTH-1,1,-1
           RAD(ID)=NU(ID)+DD(ID)*RAD(ID+1)
260     CONTINUE
      RETURN
      END

      SUBROUTINE LSTATE(NTN,NTT,BSUM,BSUMN,BSUMT)
C
C     CALCULATES NTOT DERIVATIVES AND LOWER BOUNDARY SUMS FOR LTESTEP
C
      IMPLICIT NONE
C
      PLIST
      BLANK
      ROOT
      ROOTI
      SEGH
C
      INTEGER I, ID, IJ, IL, IT, J, L, U
      REAL BSUM, BSUMN, BSUMT, C, DOP, E, FRQO, HKT
      REAL HKTT, SIGMA, SIGMAT, SRT, SRT2, T, XO, X1, X1N, X1T, VX, VXT
```

```
      REAL CHIN(MNJ), CHIT(MNJ), CLHS(MNNN,MNNN), DELTA(MNNN)
      REAL DSKK(4,MNJ), ETAN(MNJ), ETAT(MNJ)
      REAL EX(MNJ), LHS(MNNN,MNNN), LN(MNNN,MNNN)
      REAL LT(MNNN,MNNN), NTN(MNDEPTH), NTT(MNDEPTH), RHS(MNNN)
      REAL RN(MNNN), RT(MNNN), SKK(4,MNJ)
C
C     EXTERNALS
C
      REAL FRE, MATINV, NURATE
      EXTERNAL FRE, MATINV, NURATE
C
C     STATEMENT FUNCTIONS
C
      REAL BB, BBT, BBTT, FQ, TQ
      REAL SB, SBHE1, SBHE2, SBT, SBHE1T, SBHE2T
      REAL SBHM, SBHMT
      SBHM(T)=ACCOF*0.5*EXP(HK*FRQHM/T)/T/SQRT(T)
      SBHMT(T)=(-1.5-HK*FRQHM/T)/T
      SB(I,T)=ACCOF*GH(I)*EXP(HK*FRQH(I)/T)/T/SQRT(T)
      SBT(I,T)=(-1.5-HK*FRQH(I)/T)/T
      SBHE1(I,T)=ACCOF*GHE1(I)*EXP(HK*FRQHE1(I)/T)/T/SQRT(T)/2.
      SBHE1T(I,T)=(-1.5-HK*FRQHE1(I)/T)/T
      SBHE2(I,T)=ACCOF*GH(I)*EXP(HK*FRQHE2(I)/T)/T/SQRT(T)
      SBHE2T(I,T)=(-1.5-HK*FRQHE2(I)/T)/T
      BB(FQ,TQ)=BBCOF*FQ**3/(EXP(HK*FQ/TQ)-1.0)
      BBT(FQ,TQ)=BB(FQ,TQ)*HK*FQ/TQ/TQ/(1.0-EXP(-HK*FQ/TQ))
      BBTT(FQ,TQ)=BBT(FQ,TQ)*(HK*FQ*(1.0+EXP(-HK*FQ/TQ))/
     :            (1.0-EXP(-HK*FQ/TQ))/TQ-2.0)/TQ
C
C     START OF EXECUTABLE STATEMENTS
C
C     ESTABLISH LTE POPULATIONS AND DERIVATIVES
C
      DO 120 ID=1,NDEPTH
         CALL NURATE(LHS,LN,LT,RHS,RN,RT,ID)
         CALL MATINV(LHS,MNNN,MNNN)
         DO 20 I=1,MNNN
            DN(I,ID)=0.0
            DT(I,ID)=0.0
            DO 10 J=1,MNNN
               CLHS(I,J)=LHS(I,J)
10          CONTINUE
20       CONTINUE
         DO 40 I=1,MNNN
            DELTA(I)=0.0
40       CONTINUE
         DO 30 J=1,MNNN
            DO 35 I=1,MNNN
               DELTA(I)=DELTA(I)+LHS(I,J)*RHS(J)
35          CONTINUE
30       CONTINUE
         NTOT(ID)=DELTA(INTOT)
         NPROT(ID)=DELTA(INPROT)
         NHE2(1,ID)=DELTA(INHE2)
         NHE3(ID)=DELTA(INHE3)
         DO 50 IL=1,NLH
            N(IL,ID)=SB(IL,TEMP(ID))*NE(ID)*NPROT(ID)
50       CONTINUE
         DO 60 IL=1,NLHE1
            NHE1(IL,ID)=SBHE1(IL,TEMP(ID))*NE(ID)*NHE2(1,ID)
60       CONTINUE
         DO 70 IL=2,NLHE2
            NHE2(IL,ID)=SBHE2(IL,TEMP(ID))*NE(ID)*NHE3(ID)
70       CONTINUE
         DO 90 I=1,MNNN
            DO 80 J=1,MNNN
               RN(I)=RN(I)-LN(I,J)*DELTA(J)
               RT(I)=RT(I)-LT(I,J)*DELTA(J)
80          CONTINUE
90       CONTINUE
         DO 110 I=1,MNNN
            DO 100 J=1,MNNN
               DN(I,ID)=DN(I,ID)+CLHS(I,J)*RN(J)
               DT(I,ID)=DT(I,ID)+CLHS(I,J)*RT(J)
100         CONTINUE
110      CONTINUE
         NM(ID)=MU*(NTOT(ID)-NE(ID))
         NTN(ID)=DN(INTOT,ID)
         NTT(ID)=DT(INTOT,ID)
120   CONTINUE
```

```
C
C     CALCULATE LOWER BOUNDARY OPACITIES AND DERIVATIVES
C
      SRT=SQRT(TLINE/TEMP(NDEPTH))
      SRT2=SQRT(TEMP(NDEPTH))
      CALL FRE(SKK,DSKK,NDEPTH)
      HKT=HK/TEMP(NDEPTH)
      HKTT=-HKT/TEMP(NDEPTH)
C
C     ELECTRON SCATTERING
C
      DO 130 IJ=1,NJ
         CHI(IJ)=NE(NDEPTH)*SIGE
         CHIN(IJ)=SIGE
         EX(IJ)=EXP(-HKT*FREQ(IJ))
         ETA(IJ)=0.0
         ETAT(IJ)=0.0
         ETAN(IJ)=0.0
         CHIT(IJ)=0.0
130   CONTINUE
C
C     BOUND-FREE OPACITIES
C
      DO 150 IL=1,NLH
         DO 140 IJ=1,NJ
            C=SIG(IL,IJ)*N(IL,NDEPTH)
            E=C*EX(IJ)
            ETA(IJ)=ETA(IJ)+E
            ETAT(IJ)=ETAT(IJ)+E*(SBT(IL,TEMP(NDEPTH))-
     :                    HKTT*FREQ(IJ))
            ETAN(IJ)=ETAN(IJ)+E/NE(NDEPTH)
            ETAT(IJ)=ETAT(IJ)+E*DT(INPROT,NDEPTH)/NPROT(NDEPTH)
            ETAN(IJ)=ETAN(IJ)+E*DN(INPROT,NDEPTH)/NPROT(NDEPTH)
            CHI(IJ)=CHI(IJ)+C
            CHIT(IJ)=CHIT(IJ)+C*SBT(IL,TEMP(NDEPTH))
            CHIN(IJ)=CHIN(IJ)+C/NE(NDEPTH)
            CHIT(IJ)=CHIT(IJ)+C*DT(INPROT,NDEPTH)/NPROT(NDEPTH)
            CHIN(IJ)=CHIN(IJ)+C*DN(INPROT,NDEPTH)/NPROT(NDEPTH)
140      CONTINUE
150   CONTINUE
      DO 170 IL=1,NLHE1
         DO 180 IJ=1,NJ
            C=SIGHE1(IL,IJ)*NHE1(IL,NDEPTH)
            E=C*EX(IJ)
            ETA(IJ)=ETA(IJ)+E
            ETAT(IJ)=ETAT(IJ)+E*(SBHE1T(IL,TEMP(NDEPTH))-
     :                    HKTT*FREQ(IJ))
            ETAN(IJ)=ETAN(IJ)+E/NE(NDEPTH)
            ETAT(IJ)=ETAT(IJ)+E*DT(INHE2,NDEPTH)/NHE2(1,NDEPTH)
            ETAN(IJ)=ETAN(IJ)+E*DN(INHE2,NDEPTH)/NHE2(1,NDEPTH)
            CHI(IJ)=CHI(IJ)+C
            CHIT(IJ)=CHIT(IJ)+C*SBHE1T(IL,TEMP(NDEPTH))
            CHIN(IJ)=CHIN(IJ)+C/NE(NDEPTH)
            CHIT(IJ)=CHIT(IJ)+C*DT(INHE2,NDEPTH)/NHE2(1,NDEPTH)
            CHIN(IJ)=CHIN(IJ)+C*DN(INHE2,NDEPTH)/NHE2(1,NDEPTH)
170      CONTINUE
180   CONTINUE
      DO 200 IL=1,NLHE2
         DO 190 IJ=1,NJ
            C=SIGHE2(IL,IJ)*NHE2(IL,NDEPTH)
            E=C*EX(IJ)
            ETA(IJ)=ETA(IJ)+E
            ETAT(IJ)=ETAT(IJ)+E*(SBHE2T(IL,TEMP(NDEPTH))-
     :                    HKTT*FREQ(IJ))
            ETAN(IJ)=ETAN(IJ)+E/NE(NDEPTH)
            ETAT(IJ)=ETAT(IJ)+E*DT(INHE3,NDEPTH)/NHE3(NDEPTH)
            ETAN(IJ)=ETAN(IJ)+E*DN(INHE3,NDEPTH)/NHE3(NDEPTH)
            CHI(IJ)=CHI(IJ)+C
            CHIT(IJ)=CHIT(IJ)+C*SBHE2T(IL,TEMP(NDEPTH))
            CHIN(IJ)=CHIN(IJ)+C/NE(NDEPTH)
            CHIT(IJ)=CHIT(IJ)+C*DT(INHE3,NDEPTH)/NHE3(NDEPTH)
            CHIN(IJ)=CHIN(IJ)+C*DN(INHE3,NDEPTH)/NHE3(NDEPTH)
190      CONTINUE
200   CONTINUE
C
C     FREE-FREE OPACITIES
C
      X0=NHE3(NDEPTH)*NE(NDEPTH)**2
      X1=X0*SUMHE2(NDEPTH)
      X1T=X0*DSHE2(NDEPTH)
```

```
      X1N=2.0*X1/NE(NDEPTH)
      DO 210 IJ=1,NJ
         C=X1*SKK(2,IJ)
         E=C*EX(IJ)
         ETA(IJ)=ETA(IJ)+E
         ETAT(IJ)=ETAT(IJ)+E*(X1T/X1+DSKK(2,IJ)-
     :                     HKTT*FREQ(IJ))
         ETAN(IJ)=ETAN(IJ)+E*X1N/X1
         ETAT(IJ)=ETAT(IJ)+NE(NDEPTH)**2*SUMHE2(NDEPTH)*
     :              SKK(2,IJ)*EX(IJ)*DT(INHE3,NDEPTH)
         ETAN(IJ)=ETAN(IJ)+NE(NDEPTH)**2*SUMHE2(NDEPTH)*
     :              SKK(2,IJ)*EX(IJ)*DN(INHE3,NDEPTH)
         CHI(IJ)=CHI(IJ)+C
         CHIT(IJ)=CHIT(IJ)+C*(X1T/X1+DSKK(2,IJ))
         CHIN(IJ)=CHIN(IJ)+C*X1N/X1
         CHIT(IJ)=CHIT(IJ)+NE(NDEPTH)**2*SUMHE2(NDEPTH)*
     :              SKK(2,IJ)*DT(INHE3,NDEPTH)
         CHIN(IJ)=CHIN(IJ)+NE(NDEPTH)**2*SUMHE2(NDEPTH)*
     :              SKK(2,IJ)*DN(INHE3,NDEPTH)
210   CONTINUE
      DO 220 IJ=1,NJ
         C=NHE3(NDEPTH)*NE(NDEPTH)*SKK(3,IJ)
         E=C*EX(IJ)
         ETA(IJ)=ETA(IJ)+E
         ETAT(IJ)=ETAT(IJ)+E*(DSKK(3,IJ)-HKTT*FREQ(IJ))
         ETAN(IJ)=ETAN(IJ)+E/NE(NDEPTH)
         ETAT(IJ)=ETAT(IJ)+E*DT(INHE3,NDEPTH)/NHE3(NDEPTH)
         ETAN(IJ)=ETAN(IJ)+E*DN(INHE3,NDEPTH)/NHE3(NDEPTH)
         CHI(IJ)=CHI(IJ)+C
         CHIT(IJ)=CHIT(IJ)+C*DSKK(3,IJ)
         CHIN(IJ)=CHIN(IJ)+C/NE(NDEPTH)
         CHIT(IJ)=CHIT(IJ)+C*DT(INHE3,NDEPTH)/NHE3(NDEPTH)
         CHIN(IJ)=CHIN(IJ)+C*DN(INHE3,NDEPTH)/NHE3(NDEPTH)
         C=NPROT(NDEPTH)*NE(NDEPTH)*SKK(1,IJ)
         E=C*EX(IJ)
         ETA(IJ)=ETA(IJ)+E
         ETAT(IJ)=ETAT(IJ)+E*(DSKK(1,IJ)-HKTT*FREQ(IJ))
         ETAN(IJ)=ETAN(IJ)+E/NE(NDEPTH)
         ETAT(IJ)=ETAT(IJ)+E*DT(INPROT,NDEPTH)/NPROT(NDEPTH)
         ETAN(IJ)=ETAN(IJ)+E*DN(INPROT,NDEPTH)/NPROT(NDEPTH)
         CHI(IJ)=CHI(IJ)+C
         CHIT(IJ)=CHIT(IJ)+C*DSKK(1,IJ)
         CHIN(IJ)=CHIN(IJ)+C/NE(NDEPTH)
         CHIT(IJ)=CHIT(IJ)+C*DT(INPROT,NDEPTH)/NPROT(NDEPTH)
         CHIN(IJ)=CHIN(IJ)+C*DN(INPROT,NDEPTH)/NPROT(NDEPTH)
220   CONTINUE
C
C     LINE OPACITIES
C
      DO 230 IJ=ITPTRH,ITPTRHE1-1
         L=LOWH(IJ)
         U=UPH(IJ)
         IT=ITRH(L,U)
         FRQ0=FRQH(L)-FRQH(U)
         DOP=SRT2*FRQ0*DOPCOF
         VX=DELQUAD*MOD(IJ-ITPTRH,NQUAD)*SRT
         SIGMA=PIE2MC*OSCH(IT)*EXP(-VX*VX)/DOP/1.7724539
         SIGMAT=(VX*VX-0.5)/TEMP(NDEPTH)
         E=GH(L)*SIGMA*N(U,NDEPTH)/GH(U)
         C=SIGMA*N(L,NDEPTH)
         ETA(IJ)=ETA(IJ)+E
         CHI(IJ)=CHI(IJ)+C
         ETAT(IJ)=ETAT(IJ)+E*(SIGMAT+SBT(U,TEMP(NDEPTH)))
         CHIT(IJ)=CHIT(IJ)+C*(SIGMAT+SBT(L,TEMP(NDEPTH)))
         ETAN(IJ)=ETAN(IJ)+E/NE(NDEPTH)
         ETAT(IJ)=ETAT(IJ)+E*DT(INPROT,NDEPTH)/NPROT(NDEPTH)
         ETAN(IJ)=ETAN(IJ)+E*DN(INPROT,NDEPTH)/NPROT(NDEPTH)
         CHIN(IJ)=CHIN(IJ)+C/NE(NDEPTH)
         CHIT(IJ)=CHIT(IJ)+C*DT(INPROT,NDEPTH)/NPROT(NDEPTH)
         CHIN(IJ)=CHIN(IJ)+C*DN(INPROT,NDEPTH)/NPROT(NDEPTH)
230   CONTINUE
      DO 240 IJ=ITPTRHE1,ITPTRHE2-1
         L=LOWHE1(IJ)
         U=UPHE1(IJ)
         IT=ITRHE1(L,U)
         FRQ0=FRQHE1(L)-FRQHE1(U)
         DOP=SRT2*FRQ0*DOPCOF*0.5
         VX=DELQUAD*MOD(IJ-ITPTRHE1,NQUAD)*SRT
         SIGMA=PIE2MC*OSCHE1(IT)*EXP(-VX*VX)/DOP/1.7724539
         SIGMAT=(VX*VX-0.5)/TEMP(NDEPTH)
```

```
         E=GHE1(L)*SIGMA*NHE1(U,NDEPTH)/GHE1(U)
         C=SIGMA*NHE1(L,NDEPTH)
         ETA(IJ)=ETA(IJ)+E
         CHI(IJ)=CHI(IJ)+C
         ETAT(IJ)=ETAT(IJ)+E*(SIGMAT+SBHE1T(U,TEMP(NDEPTH)))
         CHIT(IJ)=CHIT(IJ)+C*(SIGMAT+SBHE1T(L,TEMP(NDEPTH)))
         ETAN(IJ)=ETAN(IJ)+E/NE(NDEPTH)
         ETAT(IJ)=ETAT(IJ)+E*DT(INHE2,NDEPTH)/NHE2(1,NDEPTH)
         ETAN(IJ)=ETAN(IJ)+E*DN(INHE2,NDEPTH)/NHE2(1,NDEPTH)
         CHIN(IJ)=CHIN(IJ)+C/NE(NDEPTH)
         CHIT(IJ)=CHIT(IJ)+C*DT(INHE2,NDEPTH)/NHE2(1,NDEPTH)
         CHIN(IJ)=CHIN(IJ)+C*DN(INHE2,NDEPTH)/NHE2(1,NDEPTH)
240   CONTINUE
      DO 250 IJ=ITPTRHE2,NJ
         L=LOWHE2(IJ)
         U=UPHE2(IJ)
         IT=ITRHE2(L,U)
         FRQ0=FRQHE2(L)-FRQHE2(U)
         DOP=SRT2*FRQ0*DOPCOF*0.5
         VX=DELQUAD*MOD(IJ-ITPTRHE2,NQUAD)*SRT
         SIGMA=PIE2MC*OSCHE2(IT)*EXP(-VX*VX)/DOP/1.7724539
         SIGMAT=(VX*VX-0.5)/TEMP(NDEPTH)
         E=GH(L)*SIGMA*NHE2(U,NDEPTH)/GH(U)
         C=SIGMA*NHE2(L,NDEPTH)
         ETA(IJ)=ETA(IJ)+E
         CHI(IJ)=CHI(IJ)+C
         ETAT(IJ)=ETAT(IJ)+E*(SIGMAT+SBHE2T(U,TEMP(NDEPTH)))
         CHIT(IJ)=CHIT(IJ)+C*(SIGMAT+SBHE2T(L,TEMP(NDEPTH)))
         ETAN(IJ)=ETAN(IJ)+E/NE(NDEPTH)
         ETAT(IJ)=ETAT(IJ)+E*DT(INHE3,NDEPTH)/NHE3(NDEPTH)
         ETAN(IJ)=ETAN(IJ)+E*DN(INHE3,NDEPTH)/NHE3(NDEPTH)
         CHIN(IJ)=CHIN(IJ)+C/NE(NDEPTH)
         CHIT(IJ)=CHIT(IJ)+C*DT(INHE3,NDEPTH)/NHE3(NDEPTH)
         CHIN(IJ)=CHIN(IJ)+C*DN(INHE3,NDEPTH)/NHE3(NDEPTH)
250   CONTINUE
C
C     H MINUS
C
      IF (RULE(1))THEN
         DO 260 IJ=1,NJ
            C=SKK(4,IJ)*N(1,NDEPTH)*NE(NDEPTH)
            E=C*EX(IJ)
            CHI(IJ)=CHI(IJ)+C
            ETA(IJ)=ETA(IJ)+E
            CHIN(IJ)=CHIN(IJ)+C*(2./NE(NDEPTH)+
     :               DN(INPROT,NDEPTH)/NPROT(NDEPTH))
            CHIT(IJ)=CHIT(IJ)+C*(DSKK(4,IJ)/SKK(4,IJ)+
     :               SBT(1,TEMP(NDEPTH))+
     :               DT(INPROT,NDEPTH)/NPROT(NDEPTH))
            ETAN(IJ)=ETAN(IJ)+E*(2./NE(NDEPTH)+
     :               DN(INPROT,NDEPTH)/NPROT(NDEPTH))
            ETAT(IJ)=ETAT(IJ)+E*(DSKK(4,IJ)/SKK(4,IJ)+
     :               SBT(1,TEMP(NDEPTH))-
     :               HKTT*FREQ(IJ)+DT(INPROT,NDEPTH)/NPROT(NDEPTH))
            C=N(1,NDEPTH)*NE(NDEPTH)*SBHM(TEMP(NDEPTH))*SIGHM(IJ)
            E=C*EX(IJ)
            CHI(IJ)=CHI(IJ)+C
            ETA(IJ)=ETA(IJ)+E
            CHIN(IJ)=CHIN(IJ)+C*(2./NE(NDEPTH)+
     :               DN(INPROT,NDEPTH)/NPROT(NDEPTH))
            CHIT(IJ)=CHIT(IJ)+C*(SBHMT(TEMP(NDEPTH))+
     :               SBT(1,TEMP(NDEPTH))+
     :               DT(INPROT,NDEPTH)/NPROT(NDEPTH))
            ETAN(IJ)=ETAN(IJ)+E*(2./NE(NDEPTH)+
     :               DN(INPROT,NDEPTH)/NPROT(NDEPTH))
            ETAT(IJ)=ETAT(IJ)+E*(SBHMT(TEMP(NDEPTH))+
     :               SBT(1,TEMP(NDEPTH))+
     :               DT(INPROT,NDEPTH)/NPROT(NDEPTH)-HKTT*FREQ(IJ))
260      CONTINUE
      ENDIF
C
C     SUBTRACT STIMULATED EMISSION FROM OPACITY AND CALCULATE EMISSIVITY
C
      DO 270 IJ=1,NJ
         CHI(IJ)=CHI(IJ)-ETA(IJ)
         CHIT(IJ)=CHIT(IJ)-ETAT(IJ)
         CHIN(IJ)=CHIN(IJ)-ETAN(IJ)
         VX=BBCOF*FREQ(IJ)**3
         ETA(IJ)=ETA(IJ)*VX
         ETAT(IJ)=ETAT(IJ)*VX
```

```
         ETAN(IJ)=ETAN(IJ)*VX
 270  CONTINUE
 C
 C    LOWER BOUNDARY SUMS
 C
      BSUM=0.0
      BSUMN=0.0
      BSUMT=0.0
      DO 280 IJ=1,NJ
         VX=WT(IJ)*BBT(FREQ(IJ),TEMP(NDEPTH))/CHI(IJ)
         BSUM=BSUM+VX
         VX=-VX/CHI(IJ)
         BSUMN=BSUMN+VX*CHIN(IJ)
         BSUMT=BSUMT+VX*CHIT(IJ)+WT(IJ)*
     :             BBTT(FREQ(IJ),TEMP(NDEPTH))/CHI(IJ)
 280  CONTINUE
      RETURN
      END

      SUBROUTINE LTESTEP(ERR)
C
C     SUBROUTINE TO PERFORM LTE RYBICKI STEP.
C
      IMPLICIT NONE
C
      PLIST
      BLANK
      ROOT
C
      INTEGER INDX(2*MNDEPTH)
      INTEGER I, ID, IJ, J, K
      REAL CHIN(MNDEPTH), CHIT(MNDEPTH)
      REAL DNTOTN(MNDEPTH), DNTOTT(MNDEPTH)
      REAL ETAN(MNDEPTH), ETAT(MNDEPTH)
      REAL MB(2*MNDEPTH,2*MNDEPTH), ML(2*MNDEPTH)
      REAL MT(MNDEPTH,3)
      REAL MUE(MNDEPTH,MNDEPTH), MV(MNDEPTH,MNDEPTH)
      REAL MK(2*MNDEPTH)
      REAL BSUM, BSUMN, BSUMT, DET, DTC, DTM, DTP
      REAL ERR, FLUX, MX, MW
      REAL OMEGA1, OMEGA1N, OMEGA1T, OMEGA2, OMEGA2N, OMEGA2T, OMEGA3
      REAL OMEGA3N, OMEGA3T, RHO1, RHO1N, RHO1T, RHO2, RHO2N, RHO2T
      REAL RHO3, RHO3N, RHO3T, S, SN, ST, VA, VB, VC, VALPHA, VBETA
      REAL VGAMMA, VX, VXT, VXN
C
C     EXTERNALS
C
      REAL EDDFAC, LINSLV, LSTATE, GENER
      EXTERNAL EDDFAC, LINSLV, LSTATE, GENER
C
C     STATEMENT FUNCTIONS
C
      REAL BB, BBT, BBTT, FQ, TQ
      BB(FQ,TQ)=BBCOF*FQ**3/(EXP(HK*FQ/TQ)-1.0)
      BBT(FQ,TQ)=BB(FQ,TQ)*HK*FQ/TQ/TQ/(1.0-EXP(-HK*FQ/TQ))
      BBTT(FQ,TQ)=BBT(FQ,TQ)*(HK*FQ*(1.0+EXP(-HK*FQ/TQ))/
     :             (1.0-EXP(-HK*FQ/TQ))/TQ-2.0)/TQ
C
C     START OF EXECUTABLE STATEMENTS
C
      CALL LSTATE(DNTOTN,DNTOTT,BSUM,BSUMN,BSUMT)
      FLUX=0.
      DO 20 I=1,NDEPTH*2
         ML(I)=0.0
         DO 10 J=1,NDEPTH*2
            MB(I,J)=0.0
10       CONTINUE
20    CONTINUE
C
C     SOLVE TRANSFER EQUATION AND CALCULATE DERIVATIVES AT EACH FREQUENCY
C
      DO 200 IJ=1,NJ
         CALL GENER(IJ,ETAN,ETAT,CHIN,CHIT)
         CALL EDDFAC(IJ)
         DO 40 I=1,NDEPTH
            MK(I)=0.0
            DO 30 J=1,NDEPTH
               MUE(I,J)=0.0
               MV(I,J)=0.0
30          CONTINUE
```

```
40        CONTINUE
          RHO1=NM(1)*MHYD
          RHO1N=(DNTOTN(1)-1.0)*MHYD*MU
          RHO1T=DNTOTT(1)*MHYD*MU
          OMEGA1=CHI(1)/RHO1
          OMEGA1T=CHIT(1)/RHO1-OMEGA1*RHO1T/RHO1
          OMEGA1N=CHIN(1)/RHO1-OMEGA1*RHO1N/RHO1
          RHO2=NM(2)*MHYD
          RHO2N=(DNTOTN(2)-1.0)*MHYD*MU
          RHO2T=DNTOTT(2)*MHYD*MU
          OMEGA2=CHI(2)/RHO2
          OMEGA2T=CHIT(2)/RHO2-OMEGA2*RHO2T/RHO2
          OMEGA2N=CHIN(2)/RHO2-OMEGA2*RHO2N/RHO2
          DTP=0.5*(OMEGA1+OMEGA2)*M(2)
          VALPHA=(FK(2)*RAD(2)-FK(1)*RAD(1))/DTP
          S=(ETA(1)+SIGE*NE(1)*RAD(1))/CHI(1)
          SN=(ETAN(1)+SIGE*RAD(1)-S*CHIN(1))/CHI(1)
          ST=(ETAT(1)-S*CHIT(1))/CHI(1)
          VB=(VALPHA+0.5*DTP*(RAD(1)-S))/(OMEGA1+OMEGA2)
          MK(1)=FH*RAD(1)+0.5*(RAD(1)-S)*DTP-VALPHA
          MT(1,2)=-FK(1)/DTP-FH-0.5*(1.0-SIGE*NE(1)/
     :           CHI(1))*DTP
          MT(1,3)=FK(2)/DTP
          MUE(1,1)=-VB*OMEGA1N+0.5*DTP*SN
          MUE(1,2)=-VB*OMEGA2N
          MV(1,1)=-VB*OMEGA1T+0.5*DTP*ST
          MV(1,2)=-VB*OMEGA2T
          DO 50 ID=2,NDEPTH-1
             DTM=DTP
             RHO3=NM(ID+1)*MHYD
             RHO3N=(DNTOTN(ID+1)-1.0)*MHYD*MU
             RHO3T=DNTOTT(ID+1)*MHYD*MU
             OMEGA3=CHI(ID+1)/RHO3
             OMEGA3T=CHIT(ID+1)/RHO3-OMEGA3*RHO3T/RHO3
             OMEGA3N=CHIN(ID+1)/RHO3-OMEGA3*RHO3N/RHO3
             DTP=0.5*(OMEGA3+OMEGA2)*M(ID+1)
             DTC=0.5*(DTM+DTP)
             VALPHA=(FK(ID)*RAD(ID)-FK(ID-1)*RAD(ID-1))/
     :              (DTM*DTC)
             VGAMMA=(FK(ID)*RAD(ID)-FK(ID+1)*RAD(ID+1))/
     :              (DTP*DTC)
             VBETA=VALPHA+VGAMMA
             VA=(VALPHA+0.5*VBETA*DTM/DTC)/(OMEGA1+OMEGA2)
             VC=(VGAMMA+0.5*VBETA*DTP/DTC)/(OMEGA2+OMEGA3)
             VB=VA+VC
             MK(ID)=VBETA+RAD(ID)-(NE(ID)*SIGE*RAD(ID)+
     :             ETA(ID))/CHI(ID)
             MT(ID,1)=FK(ID-1)/DTM/DTC
             MT(ID,2)=-(FK(ID)*(1./DTM+1./DTP)/DTC+(1.0-NE(ID)*
     :                SIGE/CHI(ID)))
             MT(ID,3)=FK(ID+1)/DTP/DTC
             MUE(ID,ID-1)=VA*OMEGA1N
             MUE(ID,ID)=VB*OMEGA2N+(-(ETA(ID)+NE(ID)*SIGE*
     :                 RAD(ID))*CHIN(ID)/CHI(ID)+
     :                 ETAN(ID)+SIGE*RAD(ID))/CHI(ID)
             MUE(ID,ID+1)=VC*OMEGA3N
             MV(ID,ID-1)=VA*OMEGA1T
             MV(ID,ID)=VB*OMEGA2T+(-(ETA(ID)+NE(ID)*SIGE*
     :                  RAD(ID))*CHIT(ID)/CHI(ID)+
     :                  ETAT(ID))/CHI(ID)
             MV(ID,ID+1)=VC*OMEGA3T
             OMEGA1=OMEGA2
             OMEGA1N=OMEGA2N
             OMEGA1T=OMEGA2T
             OMEGA2=OMEGA3
             OMEGA2T=OMEGA3T
             OMEGA2N=OMEGA3N
             RHO1=RHO2
             RHO1T=RHO2T
             RHO1N=RHO2N
             RHO2=RHO3
             RHO2N=RHO3N
             RHO2T=RHO3T
50        CONTINUE
          DTM=DTP
          ID=NDEPTH
          VALPHA=(FK(ID)*RAD(ID)-FK(ID-1)*RAD(ID-1))/DTM
          VB=-VALPHA/(OMEGA1+OMEGA2)
          VC=HO*BBT(FREQ(IJ),TEMP(ID))/CHI(ID)/BSUM
          MK(ID)=VC-VALPHA
```

```
      MT(ID,1)=-FK(ID-1)/DTM
      MT(ID,2)=FK(ID)/DTM
      MUE(ID,ID-1)=VB*OMEGA1N
      MUE(ID,ID)=VB*OMEGA2N+VC*(CHIN(ID)/CHI(ID)+BSUMN/BSUM)
      MV(ID,ID-1)=VB*OMEGA1T
      MV(ID,ID)=VB*OMEGA2T-VC*(BBTT(FREQ(IJ),TEMP(ID))/
     :            BBT(FREQ(IJ),TEMP(ID))-CHIT(ID)/CHI(ID)-
     :            BSUMT/BSUM)
      MT(1,3)=MT(1,3)/MT(1,2)
      MV(1,1)=MV(1,1)/MT(1,2)
      MV(1,2)=MV(1,2)/MT(1,2)
      MUE(1,1)=MUE(1,1)/MT(1,2)
      MUE(1,2)=MUE(1,2)/MT(1,2)
      MK(1)=MK(1)/MT(1,2)
      MT(2,2)=MT(2,2)-MT(2,1)*MT(1,3)
      MV(2,1)=MV(2,1)-MT(2,1)*MV(1,1)
      MV(2,2)=MV(2,2)-MT(2,1)*MV(1,2)
      MUE(2,1)=MUE(2,1)-MT(2,1)*MUE(1,1)
      MUE(2,2)=MUE(2,2)-MT(2,1)*MUE(1,2)
      MK(2)=MK(2)-MK(1)*MT(2,1)
      DO 80 J=2,NDEPTH-1
         MT(J,3)=MT(J,3)/MT(J,2)
         DO 60 K=1,J+1
            MV(J,K)=MV(J,K)/MT(J,2)
            MUE(J,K)=MUE(J,K)/MT(J,2)
60       CONTINUE
         MK(J)=MK(J)/MT(J,2)
         MT(J+1,2)=MT(J+1,2)-MT(J+1,1)*MT(J,3)
         DO 70 K=1,J+1
            MV(J+1,K)=MV(J+1,K)-MT(J+1,1)*MV(J,K)
            MUE(J+1,K)=MUE(J+1,K)-MT(J+1,1)*MUE(J,K)
70       CONTINUE
         MK(J+1)=MK(J+1)-MT(J+1,1)*MK(J)
80    CONTINUE
      DO 90 J=1,NDEPTH
         MV(NDEPTH,J)=MV(NDEPTH,J)/MT(NDEPTH,2)
         MUE(NDEPTH,J)=MUE(NDEPTH,J)/MT(NDEPTH,2)
90    CONTINUE
      MK(NDEPTH)=MK(NDEPTH)/MT(NDEPTH,2)
      DO 110 J=NDEPTH-1,1,-1
         DO 100 K=1,NDEPTH
            MV(J,K)=MV(J,K)-MT(J,3)*MV(J+1,K)
            MUE(J,K)=MUE(J,K)-MT(J,3)*MUE(J+1,K)
100      CONTINUE
         MK(J)=MK(J)-MT(J,3)*MK(J+1)
110   CONTINUE
C
C     RADIATION FIELD DERIVATIVES OF HYDROSTATIC EQUATION
C
      MX=M(1)*(4.*PI/CC)*WT(IJ)*CHI(1)*FH/NM(1)/MHYD
      DO 120 K=1,NDEPTH
         MB(1,K+NDEPTH)=MB(1,K+NDEPTH)-MX*MV(1,K)
         MB(1,K)=MB(1,K)-MX*MUE(1,K)
120   CONTINUE
      ML(1)=ML(1)-MX*MK(1)
      MX=FK(1)*WT(IJ)*(4.*PI/CC)
      DO 150 J=2,NDEPTH
         DO 130 K=1,NDEPTH
            MB(J,K)=MB(J,K)+MX*MUE(J-1,K)
            MB(J,K+NDEPTH)=MB(J,K+NDEPTH)+MX*MV(J-1,K)
130      CONTINUE
         ML(J)=ML(J)+MX*MK(J-1)
         MX=FK(J)*WT(IJ)*(4.*PI/CC)
         DO 140 K=1,NDEPTH
            MB(J,K)=MB(J,K)-MX*MUE(J,K)
            MB(J,K+NDEPTH)=MB(J,K+NDEPTH)-MX*MV(J,K)
140      CONTINUE
         ML(J)=ML(J)-MX*MK(J)
150   CONTINUE
C
C     RADIATION FIELD DERIVATIVES OF CONSTRAINT OF RADIATIVE EQUILIBRIUM
C
      DO 170 J=1,NDEPTH
         MW=WT(IJ)*(CHI(J)-NE(J)*SIGE)
         DO 160 K=1,NDEPTH
            MB(J+NDEPTH,K)=MB(J+NDEPTH,K)-MW*MUE(J,K)
            MB(J+NDEPTH,K+NDEPTH)=MB(J+NDEPTH,K+NDEPTH)-MW*MV(J,K)
160      CONTINUE
         ML(J+NDEPTH)=ML(J+NDEPTH)-MW*MK(J)
170   CONTINUE
```

```
C
C     RADIATION FIELD CONTRIBUTIONS TO CONSTRAINT EQUATIONS
C
C     RADIATIVE EQUILIBRIUM
C
         DO 180 ID=1,NDEPTH
            ML(ID+NDEPTH)=ML(ID+NDEPTH)+WT(IJ)*(ETA(ID)-
     :           (CHI(ID)-NE(ID)*SIGE)*RAD(ID))
            MB(ID+NDEPTH,ID)=MB(ID+NDEPTH,ID)-WT(IJ)*
     :           (ETAN(ID)-(CHIN(ID)-SIGE)*RAD(ID))
            MB(ID+NDEPTH,ID+NDEPTH)=MB(ID+NDEPTH,ID+NDEPTH)-
     :           WT(IJ)*(ETAT(ID)-CHIT(ID)*RAD(ID))
180      CONTINUE
C
C     HYDROSTATIC EQUILIBRIUM
C
         RHO1=NM(1)*MHYD
         RHO1N=(DNTOTN(1)-1.0)*MHYD*MU
         RHO1T=DNTOTT(1)*MHYD*MU
         VA=(4.*PI/CC)*M(1)*WT(IJ)*FH*RAD(1)*CHI(1)/RHO1
         ML(1)=ML(1)-VA
         MB(1,1)=MB(1,1)+VA*(CHIN(1)/CHI(1)-RHO1N/RHO1)
         MB(1,1+NDEPTH)=MB(1,1+NDEPTH)+VA*(CHIT(1)/CHI(1)-
     :                  RHO1T/RHO1)
         DO 190 ID=2,NDEPTH
            ML(ID)=ML(ID)-(4.*PI/CC)*WT(IJ)*(FK(ID)*RAD(ID)-
     :               FK(ID-1)*RAD(ID-1))
190      CONTINUE
         FLUX=FLUX+WT(IJ)*FH*RAD(1)
200   CONTINUE
C
C     HYDROSTATIC EQUILIBRIUM (NON-RADIATIVE TERMS)
C
      MB(1,1)=MB(1,1)+DNTOTN(1)*TEMP(1)*KB
      MB(1,1+NDEPTH)=MB(1,1+NDEPTH)+KB*(DNTOTT(1)*TEMP(1)+NTOT(1))
      ML(1)=ML(1)+M(1)*GRAV-KB*NTOT(1)*TEMP(1)
      DO 210 ID=2,NDEPTH
         ML(ID)=ML(ID)+M(ID)*GRAV-KB*(NTOT(ID)*TEMP(ID)-NTOT(ID-1)*
     :                         TEMP(ID-1))
         MB(ID,ID-1)=MB(ID,ID-1)-KB*DNTOTN(ID-1)*TEMP(ID-1)
         MB(ID,ID)=MB(ID,ID)+KB*DNTOTN(ID)*TEMP(ID)
         MB(ID,ID+NDEPTH-1)=MB(ID,ID+NDEPTH-1)-KB*(DNTOTT(ID-1)*
     :                         TEMP(ID-1)+NTOT(ID-1))
         MB(ID,ID+NDEPTH)=MB(ID,ID+NDEPTH)+KB*(DNTOTT(ID)*TEMP(ID)+
     :                         NTOT(ID))
210   CONTINUE
C
C     SURFACE FLUX CHECK
C
      WRITE (6,1000)FLUX,HO
C
C     SOLVE THE SYSTEM OF EQUATIONS
C
      CALL LINSLV(MB,ML,MK,2*NDEPTH,2*MNDEPTH)
      ERR=0.
      DO 220 ID=1,NDEPTH
         ERR=MAX(ERR,ABS(MK(ID+NDEPTH)/TEMP(ID)),ABS(MK(ID)/NE(ID)))
220   CONTINUE
C
C     SCALE DOWN CORRECTIONS IF VERY LARGE TO IMPROVE HYPERCIRCLE
C     OF CONVERGENCE
C
      VA=MIN(.95,0.1/ERR)
      WRITE (6,*)
      WRITE (6,1001)
      WRITE (6,*)
      DO 230 ID=1,NDEPTH
         WRITE (6,*)ID,TEMP(ID),MK(ID+NDEPTH)/TEMP(ID)
         WRITE (6,*)'        ',NE(ID),MK(ID)/NE(ID)
         WRITE (6,*)
         TEMP(ID)=TEMP(ID)+MK(ID+NDEPTH)*VA
         NE(ID)=MAX(.1*NE(ID),NE(ID)+MK(ID)*VA)
230   CONTINUE
      RETURN
C
1000  FORMAT (' CALCULATED FLUX IS ',E12.5,/,
     :        ' EXPECTED FLUX IS   ',E12.5)
1001  FORMAT ('TEMPERATURE AND ELECTRON DENSITY STRATIFICATION AND
     :ERRORS:')
      END
```

```
      SUBROUTINE NURATE(LHS,LN,LT,RHS,RN,RT,ID)
C
C     CALCULATE THE RATE MATRIX FOR PSEUDO-LTE GIVEN DEPARTURE
C     COEFFICIENTS, TEMPERATURE, AND ELECTRON DENSITY
C
      IMPLICIT NONE
C
      PLIST
      ROOT
      ROOTI
      SEGH
C
      INTEGER I, ID, IL, J, L, U
      REAL LHS(MNNN,MNNN), LN(MNNN,MNNN), LT(MNNN,MNNN)
      REAL RHS(MNNN), RN(MNNN), RT(MNNN)
      REAL SUM, T, VX, VXT
C
      REAL SB, SBHE1, SBHE2, SBT, SBHE1T, SBHE2T
C
      SB(I,T)=ACCOF*GH(I)*EXP(HK*FRQH(I)/T)/T/SQRT(T)
      SBT(I,T)=-(HK*FRQH(I)/T+1.5)/T
      SBHE1(I,T)=ACCOF*GHE1(I)*EXP(HK*FRQHE1(I)/T)/T/SQRT(T)/2.
      SBHE1T(I,T)=-(HK*FRQHE1(I)/T+1.5)/T
      SBHE2(I,T)=ACCOF*GH(I)*EXP(HK*FRQHE2(I)/T)/T/SQRT(T)
      SBHE2T(I,T)=-(HK*FRQHE2(I)/T+1.5)/T
C
C     START OF EXECUTABLE STATEMENTS
C
C     PARTITION FUNCTIONS
C
      SUMH(ID)=0.0
      DSHDT(ID)=0.0
      DO 10 IL=1,MQH
         VX=SB(IL,TEMP(ID))
         VXT=VX*SBT(IL,TEMP(ID))
         SUMH(ID)=SUMH(ID)+VX
         DSHDT(ID)=DSHDT(ID)+VXT
10    CONTINUE
      SUMHE1(ID)=0.0
      DSHE1(ID)=0.0
      DO 20 IL=1,MQHE1
         VX=SBHE1(IL,TEMP(ID))
         VXT=VX*SBHE1T(IL,TEMP(ID))
         SUMHE1(ID)=SUMHE1(ID)+VX
         DSHE1(ID)=DSHE1(ID)+VXT
20    CONTINUE
      SUMHE2(ID)=0.0
      DSHE2(ID)=0.0
      DO 30 IL=1,MQH*2
         VX=SBHE2(IL,TEMP(ID))
         VXT=VX*SBHE2T(IL,TEMP(ID))
         SUMHE2(ID)=SUMHE2(ID)+VX
         DSHE2(ID)=DSHE2(ID)+VXT
30    CONTINUE
      DO 50 I=1,MNNN
         RHS(I)=0.0
         RN(I)=0.0
         RT(I)=0.0
         DO 40 J=1,MNNN
            LHS(I,J)=0.0
            LN(I,J)=0.0
            LT(I,J)=0.0
40       CONTINUE
50    CONTINUE
C
C     PARTICLE CONSERVATION
C
      RHS(1)=-NE(ID)
      RN(1)=-1.0
      LHS(1,INTOT)=-1.
      LHS(1,INPROT)=MU1*(1.+NE(ID)*SUMH(ID))
      LN(1,INPROT)=MU1*SUMH(ID)
      LT(1,INPROT)=MU1*NE(ID)*DSHDT(ID)
C
C     CHARGE CONSERVATION EQUATION
C
      RHS(2)=NE(ID)
      RN(2)=1.0
      LHS(2,INPROT)=1.0
```

```
      LHS(2,INHE3)=2.0+NE(ID)*SUMHE2(ID)
      LN(2,INHE3)=SUMHE2(ID)
      LT(2,INHE3)=NE(ID)*DSHE2(ID)
C
C     STATISTICAL EQUILIBRIUM
C     SINGLY-IONIZED HELIUM
C
      LN(3,INHE3)=-SBHE2(1,TEMP(ID))
      LHS(3,INHE3)=LN(3,INHE3)*NE(ID)
      LT(3,INHE3)=LHS(3,INHE3)*SBHE2T(1,TEMP(ID))
      LHS(3,INHE2)=1.0
C
C     HELIUM ABUNDANCE EQUATION
C
      LHS(4,INHE2)=NE(ID)*SUMHE1(ID)+1.0
      LN(4,INHE2)=SUMHE1(ID)
      LT(4,INHE2)=NE(ID)*DSHE1(ID)
      LHS(4,INHE3)=NE(ID)*SUMHE2(ID)+1.0
      LN(4,INHE3)=SUMHE2(ID)
      LT(4,INHE3)=NE(ID)*DSHE2(ID)
      LHS(4,INPROT)=-Y*(1.0+NE(ID)*SUMH(ID))
      LN(4,INPROT)=-Y*SUMH(ID)
      LT(4,INPROT)=-Y*NE(ID)*DSHDT(ID)
      RETURN
      END


      SUBROUTINE PUTOUT(TEFF,Z)
C
C     WRITE OUT A COMPLETED MODEL TO DISK
C
      IMPLICIT NONE
C
      PLIST
      ROOT
      ROOTI
C
C     LOCAL VARIABLES
C
      LOGICAL LINES
      INTEGER I, ID, IL
      REAL ABUND(92)
      REAL T, TEFF, Z
C
C     START OF EXECUTABLE STATEMENTS
C
C     PREVIOUS OUTPUT LINES WERE ALREADY TAKEN CARE OF IN SETUP
C
      WRITE (12,1007)(M(ID),TEMP(ID),NTOT(ID),NE(ID),ID=1,NDEPTH)
C
C     HYDROGEN OCCUPATION NUMBERS
C
      WRITE (12,1005)1
      WRITE (12,1005)0,0,5,NTRH
      WRITE (12,1006)(FRQH(IL),IL=1,5)
      DO 10 IL=1,NTRH
         WRITE (12,1005)LOWERH(IL),UPPERH(IL)
10    CONTINUE
      WRITE (12,1006)((N(IL,ID),IL=1,5),NPROT(ID),ID=1,NDEPTH)
C
C     HELIUM OCCUPATION NUMBERS
C
      WRITE (12,1005)2
      WRITE (12,1005)0,0,19,NTRHE1,10,NTRHE2
      WRITE (12,1006)(FRQHE1(IL),IL=1,19)
      DO 20 IL=1,NTRHE1
         WRITE (12,1005)LOWERHE1(IL),UPPERHE1(IL)
20    CONTINUE
      WRITE (12,1006)(FRQHE2(IL),IL=1,10)
      DO 30 IL=1,NTRHE2
         WRITE (12,1005)LOWERHE2(IL),UPPERHE2(IL)
30    CONTINUE
      WRITE (12,1006)((NHE1(IL,ID),IL=1,19),(NHE2(IL,ID),
     :                IL=1,10),NHE3(ID),ID=1,NDEPTH)
      WRITE (12,1005)0
      WRITE (12,1005)0
      RETURN
C
C     FORMAT STATEMENTS
C
1005  FORMAT (16I5)
```

```
1006  FORMAT (5E15.7)
1007  FORMAT (4E15.7)
C
      END

      SUBROUTINE SETUP(NITER,TEFF,Z)
C
C     READ IN THE APPROXIMATE MODEL MAKE ALL INITIAL CALCULATIONS
C
      IMPLICIT NONE
C
      PLIST
      ROOT
      ROOTI
C
C     LOCAL VARIABLES
C
      CHARACTER*80 HEADER
      INTEGER I, ID, II, IJ, IL, IT, J, NO, N1, N2, N3, N4, N5, NITER
      REAL ABUND(92)
      REAL ALPHA, BETA, CON, CON1, DOP, DUMMY, FCON, FRQ3, GLOG, T
      REAL TEFF, VX, Z, SRT
C
C     TABLE OF ATOMIC WEIGHTS
C
      REAL WEIGHT(92)
      DATA WEIGHT/1.0,4.0,6.9,9.0,10.8,12.0,14.0,16.0,19.0,20.2,23.0,
     : 24.3,27.0,28.1,31.0,32.1,35.5,39.9,39.1,40.1,45.0,47.9,50.9,52.0,
     : 54.9,55.8,58.9,58.7,63.5,65.4,69.7,72.6,74.9,79.0,79.9,83.8,85.5,
     : 87.6,88.9,91.2,92.9,95.9,98.9,101.1,102.9,106.4,107.9,112.4,
     : 114.8,118.7,121.8,127.6,126.9,131.3,132.9,137.3,138.9,140.1,
     : 140.9,144.2,145.0,150.4,152.0,157.3,158.9,162.5,164.9,167.3,
     : 168.9,173.0,175.0,178.5,180.9,183.9,186.2,190.2,192.2,195.1,
     : 197.0,200.6,204.4,207.2,209.0,209.0,210.0,222.0,223.0,226.0,
     : 227.0,232.0,231.0,238.0/
C
C     TABLE GIVING QUANTUM NUMBER OF ACTIVE ELECTRON OF EACH NEUTRAL
C     HELIUM STATE TREATED BY THE PROGRAM
C
      INTEGER QN(25)
      DATA QN/1,4*2,6*3,8*4,5,6,7,8,9,10/
C
C     EXTERNAL PROCEDURES
C
      REAL EXIT, HEGAUNT, GAUNT, SB, SBHE1, SBHE2, HMINUS
      EXTERNAL EXIT, HEGAUNT, GAUNT, HMINUS
C
C     LOCAL STATEMENT FUNCTIONS
C
C     THESE FUNCTIONS GIVE ACTIVITIES OF THE VARIOUS STATES
C     OF HYDROGEN (SB), NEUTRAL HELIUM (SBHE1), AND IONIZED
C     HELIUM (SBHE2).
C
      SB(I,T)=ACCOF*GH(I)*EXP(HK*FRQH(I)/T)/T/SQRT(T)
      SBHE1(I,T)=ACCOF*GHE1(I)*EXP(HK*FRQHE1(I)/T)/T/SQRT(T)/2.
      SBHE2(I,T)=ACCOF*GH(I)*EXP(HK*FRQHE2(I)/T)/T/SQRT(T)
C
C     START OF EXECUTABLE STATEMENTS
C
      WRITE (6,*)'LTE MODEL CALCULATION PROGRAM'
C
C     READ COMMENT LINE OF INPUT MODEL
C
      READ (5,1001)HEADER
      WRITE (6,1001)HEADER
      WRITE (12,1001)'LTE MODEL WITHOUT LINES'
C
C     READ BASIC MODEL PARAMETERS
C
      READ (5,1002)TEFF,TLINE,GLOG,Y,Z,NITER
      WRITE (12,1002)TEFF,TLINE,GLOG,Y,Z,NITER,-1,0,0
C
C     CALCULATE EDDINGTON FLUX AND SURFACE GRAVITY FROM PARAMETERS
C
      H0=5.6692E-5*TEFF**4/12.5663708
      GRAV=EXP(2.302585093*GLOG)
      WRITE (6,1003)TEFF,TLINE,GLOG,Y,Z,NITER
C
C     READ COMPOSITION, WHICH TAKES THE FORM OF LOG NUMBER RELATIVE
C     TO HYDROGEN=12.0
```

```
C
      READ (5,1004)ABUND
      WRITE (12,1004)ABUND
C
C     CALCULATE MEAN MOLECULAR WEIGHT (MU) AND NUCLEI PER PROTON (MU1)
C
      MU=1.0+4.0*Y
      MU1=1.0+Y
      DO 10 I=3,92
         VX=1.E-12*EXP(2.302585093*ABUND(I))*Z
         MU=MU+VX*WEIGHT(I)
         MU1=MU1+VX
10    CONTINUE
      MU=MU/MU1
C
C     READ NUMBER OF DEPTH POINTS AND FREQUENCIES
C     AND THEN READ THE FREQUENCIES
C
      READ (5,1005)NDEPTH,NJ
      READ (5,1006)(FREQ(IJ),IJ=1,NJ)
      WRITE (12,1005)NDEPTH,NJ
      WRITE (12,1006)(FREQ(IJ),IJ=1,NJ)
C
C     CALCULATE FREQUENCY QUADRATURE WEIGHTS
C     NO CORRECTION IS MADE FOR LINES; I.E. IT IS
C     ASSUMED THAT THE LINES ARE NARROW.
C
      DO 20 IJ=1,MNJ
         WT(IJ)=0.0
20    CONTINUE
C
C     READ IN FREQUENCY INTERVAL NUMBER AND RULE
C
30       READ (5,1005)NO,N1
         WRITE (12,1005)NO,N1
         IF (NO.EQ.2)THEN
C
C     TRAPEZOIDAL RULE:  N1 IS NUMBER OF FIRST FREQUENCY
C
            VX=0.5*ABS(FREQ(N1)-FREQ(N1+1))
            WT(N1)=WT(N1)+VX
            WT(N1+1)=WT(N1+1)+VX
            GO TO 30
         ELSE IF (NO.EQ.3)THEN
C
C     SIMPSON'S RULE; N1 IS NUMBER OF CENTRAL FREQUENCY
C     **NOTE THAT NO CHECK IS MADE TO BE SURE THAT THE
C     TWO FREQUENCY SUBINTERVALS ARE EQUAL, AS THEY NEED
C     TO BE.**
C
            VX=ABS(FREQ(N1+1)-FREQ(N1-1))/6.0
            WT(N1+1)=WT(N1+1)+VX
            WT(N1-1)=WT(N1-1)+VX
            WT(N1)=WT(N1)+4.0*VX
            GO TO 30
         ELSE IF (NO.NE.0)THEN
C
C     PROGRAM DOESN'T RECOGNIZE THE RULE
C
            WRITE (6,*) 'ERROR IN FREQUENCY QUADRATURE RULE'
            CALL EXIT(1)
         ENDIF
C
C     READ OPACITY RULE AND DISCARD IT
C
      READ (5,1005)RULE
      WRITE (12,1005)RULE
      IF (RULE(2))WRITE (6,*)' LINES INCLUDED IN MODEL'
C
C     READ MODEL
C
      READ (5,1007)(M(ID),TEMP(ID),NTOT(ID),NE(ID),ID=1,NDEPTH)
C
C     BE SURE MASS VARIABLE IS MASS DIFFERENCE, NOT COLUMN MASS.
C     THIS IS INDICATED BY LOOKING AT LAST TWO ENTRIES; THE LAST
C     MASS DIFFERENCE SHOULD ALWAYS BE VERY SMALL.
C
      IF (M(NDEPTH).GT.M(NDEPTH-1))THEN
         DO 40 ID=NDEPTH,2,-1
            M(ID)=M(ID)-M(ID-1)
```

```
40        CONTINUE
       ENDIF
C
C      READ ATOMIC OCCUPATION NUMBERS
C
50        READ (5,1005)I
          IF (I.EQ.1)THEN
C
C      READ NUMBER OF HYDROGEN LEVELS TO USE FOR EACH IONIZATION.
C      ALSO READ LINES TO USE.
C
             READ (5,1005)NO,N1,N2,NTRH
             IF (N1.NE.0)THEN
                WRITE (6,*)'PLEASE DO NOT SPECIFY ANY H- LINES.'
                CALL EXIT(1)
             ENDIF
C
C      THROW AWAY NEGATIVE HYDROGEN IONIZATION FREQUENCY.
C
             IF (NO.NE.0)READ (5,1006)CON
C
C      READ NEUTRAL HYDROGEN IONIZATION FREQUENCIES.
C
             READ (5,1006)(FRQH(IL),IL=1,N2)
C
C      READ LIST OF LINES TO USE FOR HYDROGEN
C
             IF (NTRH.GT.0)
     :           READ (5,1011)(LOWERH(IL),UPPERH(IL),IL=1,NTRH)
C
C      READ ALL HYDROGEN OCCUPATION NUMBERS.
C      NEGATIVE HYDROGEN ION OCCUPATION IS THROWN AWAY.
C
             READ (5,1006)((CON,IL=1,NO),(N(IL,ID),IL=1,N2),
     :                    NPROT(ID),ID=1,NDEPTH)
C
C      CALCULATE LTE POPULATIONS FOR LEVELS NOT INCLUDED IN APPROXIMATE
C      INPUT MODEL.
C
             DO 70 ID=1,NDEPTH
                DO 60 IL=1,NLH
                   N(IL,ID)=NPROT(ID)*NE(ID)*SB(IL,TEMP(ID))
60              CONTINUE
70           CONTINUE
             GO TO 50
          ELSE IF (I.EQ.2)THEN
C
C      TREAT HELIUM THE SAME WAY AS HYDROGEN.
C
             READ (5,1005)NO,N1,N2,NTRHE1,N4,NTRHE2
             IF (N1.NE.0)THEN
                WRITE (6,*)'PLEASE DO NOT SPECIFY ANY HE- LINES.'
                CALL EXIT(1)
             ENDIF
             IF (NO.NE.0)READ (5,1006)CON
             READ (5,1006)(FRQHE1(IL),IL=1,N2)
             IF (NTRHE1.GT.0)
     :          READ (5,1011)(LOWERHE1(IL),UPPERHE1(IL),IL=1,NTRHE1)
             READ (5,1006)(FRQHE2(IL),IL=1,N4)
             IF (NTRHE2.GT.0)
     :          READ (5,1011)(LOWERHE2(IL),UPPERHE2(IL),IL=1,NTRHE2)
             READ (5,1006)((CON,IL=1,NO),(NHE1(IL,ID),IL=1,N2),
     :                   (NHE2(IL,ID),IL=1,N4),NHE3(ID),ID=1,NDEPTH)
             DO 100 ID=1,NDEPTH
                DO 80 IL=1,NLHE2
                   NHE2(IL,ID)=NHE3(ID)*NE(ID)*SBHE2(IL,TEMP(ID))
80              CONTINUE
                DO 90 IL=1,NLHE1
                   NHE1(IL,ID)=NHE2(1,ID)*NE(ID)*SBHE1(IL,TEMP(ID))
90              CONTINUE
100          CONTINUE
             GO TO 50
          ELSE IF (I.NE.0)THEN
C
C      OTHER ELEMENTS INCLUDED BUT NOT WANTED.
C      PARDON THE ANACHRONISTIC USE OF THE TERMS "CARDS" AND "DECK."
C      THESE ARE PROBABLY REALLY LINES IN AN EDITOR-CREATED DISK FILE.
C
             WRITE (6,*) 'PLEASE REMOVE HEAVY ION CARDS FROM DECK'
             CALL EXIT(1)
```

```
         ENDIF
C
C     REMAINDER OF DECK IS IGNORED.
C     PREPARE PRECALCULATED QUANTITIES.
C
      ITPTRH=NJ+1
      ITPTRHE1=ITPTRH
      ITPTRHE2=ITPTRH
      IF (RULE(2))THEN
         SRT=SQRT(TLINE)
C
C     HYDROGEN LINES
C
         N1=NTRH
         DO 130 IL=1,N1
C
C     GET UPPER AND LOWER LEVEL NUMBERS.
C
            J=UPPERH(IL)
            I=LOWERH(IL)
            IT=ITRH(I,J)
C
C     REJECT LINE IF FORBIDDEN
C
            IF (IT.EQ.0)THEN
               WRITE (6,*)'H LINE ',I,' TO ',J,' IS FORBIDDEN.'
               GO TO 130
            ENDIF
C
C     CALCULATE LINE CENTRAL FREQUENCY AND DOPPLER WIDTH.
C     NOTE THAT THIS PROGRAM ASSUMES LINE BROADENING IS DOMINATED
C     BY DOPPLER BROADENING.
C
            FCON=FRQH(I)-FRQH(J)
            DOP=SRT*FCON*DOPCOF
            IF (FCON.LE.0)THEN
               WRITE (6,*) 'H IONIZATION FREQUENCY ERROR FOR LINE ',
     :                     I,',',J
               FCON=-FCON
            ENDIF
C
C     ADD FREQUENCIES TO FREQUENCY LIST FOR THE LINE.
C
            DO 110 II=1,NQUAD
               FREQ(II+NJ)=FCON
               UPH(II+NJ)=J
               LOWH(II+NJ)=I
110         CONTINUE
C
C     CALCULATE WEIGHTS FOR THE LINE FREQUENCIES.
C     NOTE THAT SIMPSON'S RULE IS USED; THUS NQUAD SHOULD BE ODD.
C
            DO 120 I=1,NQUAD-2,2
               WT(I+NJ)=WT(I+NJ)+2.0*DELQUAD*DOP/3.0
               WT(I+NJ+1)=WT(I+NJ+1)+8.*DELQUAD*DOP/3.0
               WT(I+NJ+2)=WT(I+NJ+2)+2.0*DELQUAD*DOP/3.0
120         CONTINUE
            NJ=NJ+NQUAD
130      CONTINUE
C
C     NOW DO ALL THE SAME FOR NEUTRAL AND SINGLY IONIZED HELIUM.
C
         ITPTRHE1=NJ+1
         N1=NTRHE1
         DO 160 IL=1,N1
            J=UPPERHE1(IL)
            I=LOWERHE1(IL)
            IT=ITRHE1(I,J)
            IF (IT.EQ.0)THEN
               WRITE (6,*)'HE1 LINE ',I,' TO ',J,' IS FORBIDDEN.'
               GO TO 160
            ENDIF
            FCON=FRQHE1(I)-FRQHE1(J)
            DOP=SRT*FCON*DOPCOF*0.5
            IF (FCON.LE.0)THEN
               WRITE (6,*) 'HE1 IONIZATION FREQUENCY ERROR FOR LINE ',
     :                     I,',',J
               FCON=-FCON
            ENDIF
            DO 140 II=1,NQUAD
```

```
                FREQ(II+NJ)=FCON
                UPHE1(II+NJ)=J
                LOWHE1(II+NJ)=I
140          CONTINUE
             DO 150 I=1,NQUAD-2,2
                WT(I+NJ)=WT(I+NJ)+2.0*DELQUAD*DOP/3.0
                WT(I+NJ+1)=WT(I+NJ+1)+8.0*DELQUAD*DOP/3.0
                WT(I+NJ+2)=WT(I+NJ+2)+2.0*DELQUAD*DOP/3.0
150          CONTINUE
             NJ=NJ+NQUAD
160       CONTINUE
          ITPTRHE2=NJ+1
          N1=NTRHE2
          DO 190 IL=1,N1
             J=UPPERHE2(IL)
             I=LOWERHE2(IL)
             IT=ITRHE2(I,J)
             IF (IT.EQ.0)THEN
                WRITE (6,*)'HE2 LINE ',I,' TO ',J,' IS FORBIDDEN.'
                GO TO 190
             ENDIF
             FCON=FRQHE2(I)-FRQHE2(J)
             DOP=SRT*FCON*DOPCOF*0.5
             IF (FCON.LE.0)THEN
                WRITE (6,*) 'HE2 IONIZATION FREQUENCY ERROR FOR LINE ',
     :                  I,',',J
                FCON=-FCON
             ENDIF
             DO 170 II=1,NQUAD
                FREQ(II+NJ)=FCON
                UPHE2(II+NJ)=J
                LOWHE2(II+NJ)=I
170          CONTINUE
             DO 180 I=1,NQUAD-2,2
                WT(I+NJ)=WT(I+NJ)+2.0*DELQUAD*DOP/3.0
                WT(I+NJ+1)=WT(I+NJ+1)+8.0*DELQUAD*DOP/3.0
                WT(I+NJ+2)=WT(I+NJ+2)+2.0*DELQUAD*DOP/3.0
180          CONTINUE
             NJ=NJ+NQUAD
190       CONTINUE
       ENDIF
       IF (NJ.GT.MNJ)THEN
          WRITE (6,*) 'TOO MANY FREQUENCY POINTS'
          CALL EXIT(1)
       ENDIF
       WRITE (6,*)' FREQUENCY POINTS = ',NJ
       IF (RULE(1))THEN
          DO 200 IJ=1,NJ
             IF (FCON.GT.FRQHM)SIGHM(IJ)=HMINUS(IJ)
200       CONTINUE
       ENDIF
C
C      BOUND-FREE CROSS SECTIONS
C
       DO 240 IJ=1,NJ
          FCON=FREQ(IJ)
          FRQ3=FCON**3
          CON=2.815E29/FRQ3
          CON1=FCON/4.0
          DO 210 I=1,NLHE2
             SIGHE2(I,IJ)=0.
             IF (FCON.GT.FRQHE2(I))
     :            SIGHE2(I,IJ)=16.*CON*GAUNT(I,CON1)/FLOAT(I)**5
210       CONTINUE
          DO 220 I=1,NLHE1
             SIGHE1(I,IJ)=0.
             IF (FCON.GT.FRQHE1(I))
     :            SIGHE1(I,IJ)=CON*HEGAUNT(I,FCON)/FLOAT(QN(I))**5
220       CONTINUE
          DO 230 I=1,NLH
             SIG(I,IJ)=0.
             IF (FCON.GT.FRQH(I))SIG(I,IJ)=CON*GAUNT(I,FCON)/FLOAT(I)**5
230       CONTINUE
          SIG(NLH+1,IJ)=3.69E8/FRQ3
240    CONTINUE
C
C      DETERMINE CUTOFF FREQUENCIES FOR FREE-FREE OPACITY.
C      THESE ALLOW THE INCLUSION OF BOUND LEVELS ABOVE THOSE
C      ACCOUNTED FOR EXPLICITLY AS PART OF THE FREE-FREE OPACITY.
C
```

```
      DO 250 IJ=1,NJ
         FF(IJ,1)=MIN(FRQH(NLH)*(MQH/(MQH+1))**2,FREQ(IJ))
         FF(IJ,2)=MIN(FRQHE1(NLHE1)*(MQHE1/(MQHE1+1))**2,FREQ(IJ))
         FF(IJ,3)=MIN(FRQHE2(NLHE2)*(2*MQH/(2*MQH+1))**2,FREQ(IJ))
250   CONTINUE
C
C     CALCULATE AUER/MIHALAS PSEUDO HEAVY PARTICLE DENSITY
C
      DO 260 ID=1,NDEPTH
         NM(ID)=MU*(NTOT(ID)-NE(ID))
260   CONTINUE
      RETURN
C
1001  FORMAT (A80)
1002  FORMAT (2F9.0,F8.2,F8.3,F8.5,4I5)
1003  FORMAT (' EFFECTIVE TEMPERATURE = ',F9.0,/,
     :        ' LINE TEMPERATURE      = ',F9.0,/,
     :        ' LOG SURFACE GRAVITY   = ',F8.2,/,
     :        ' HELIUM/HYDROGEN       = ',F8.3,/,
     :        ' Z SCALE FACTOR        = ',F8.5,/,
     :        ' ITERATION LIMIT       = ',I5,/)
1004  FORMAT (8F10.6)
1005  FORMAT (16I5)
1006  FORMAT (5E15.7)
1007  FORMAT (4E15.7)
1011  FORMAT (2I5)
C
      END


      BLOCK DATA TABLES
C
C     CONTAINS ALL DATA STATEMENTS FOR COMMON BLOCKS, IN ACCORDANCE
C     WITH THE ANSI STANDARD
C
      PLIST
      ROOTI
C
C     NEUTRAL HELIUM STATISTICAL WEIGHTS
C
      DATA GHE1/1.,3.,1.,9.,3.,3.,1.,9.,5.,15.,3.,3.,1.,9.,5.,15.,3.,
     : 21.,7.,100.,144.,196.,256.,324.,400.,484.,576.,676.,784.,900.,
     : 1024./
C
C     HYDROGEN/IONIZED HELIUM STATISTICAL WEIGHTS
C
      DATA GH/2.,8.,18.,32.,50.,72.,98.,128.,162.,200.,242.,288.,338.,
     : 392.,450.,512.,578.,648.,722.,800.,882.,968.,1058.,1152.,1250.,
     : 1352.,1458.,1568.,1682.,1800.,1922.,2048./
C
C     HYDROGENIC OSCILLATOR STRENGTHS
C
      DATA OSCH/4.162E-1,7.910E-2,2.899E-2,1.394E-2, 6.408E-1,1.193E-1,
     :4.467E-2, 8.420E-1,1.506E-1, 1.038/
C
C     HYDROGEN TRANSITION MATRIX
C
      DATA (ITRH(1,I),I=1,5)/0,1,2,3,4/
      DATA (ITRH(2,I),I=1,5)/1,0,5,6,7/
      DATA (ITRH(3,I),I=1,5)/2,5,0,8,9/
      DATA (ITRH(4,I),I=1,5)/3,6,8,0,10/
      DATA (ITRH(5,I),I=1,5)/4,7,9,10,0/
C
C     NEUTRAL HELIUM OSCILLATOR STRENGTHS
C
      DATA OSCHE1/.2762,.0734,.0302, .5391,.06446,.0231, .3764,.1514,
     :.0507, .0693,.6090,.0118,.1250, .0480,.7110,.00834,.1220, .8960,
     :.0429, .6290,.1400, .1110,.1450,.4820, .0139,.00858,1.0100,
     : .0205,1.0200, .1030,.6470, 1.2100, .8530, .2000/
C
C     NEUTRAL HELIUM TRANSITION MATRIX
C
      DATA (ITRHE1(1,I),I=1,19)/0,0,0,0,1,0,0,0,0,0,2,0,0,0,0,0,3,0,0/
      DATA (ITRHE1(2,I),I=1,19)/0,0,0,4,0,0,0,5,0,0,0,0,0,6,0,0,0,0,0/
      DATA (ITRHE1(3,I),I=1,19)/0,0,0,0,7,0,0,0,0,0,8,0,0,0,0,0,9,0,0/
      DATA (ITRHE1(4,I),I=1,19)/0,4,0,0,0,10,0,0,0,11,0,12,0,0,0,13,
     : 3*0/
      DATA (ITRHE1(5,I),I=1,19)/1,0,7,0,0,0,14,0,15,0,0,0,16,0,17,
     : 4*0/
      DATA (ITRHE1(6,I),I=1,19)/0,0,0,10,0,0,0,18,0,0,0,0,0,19,5*0/
      DATA (ITRHE1(7,I),I=1,19)/0,0,0,0,14,0,0,0,0,0,20,0,0,0,0,0,21,
```

```
      : 2*0/
       DATA (ITRHE1(8,I),I=1,19)/0,2,0,0,0,18,0,0,0,22,0,23,0,0,0,24,
      : 3*0/
       DATA (ITRHE1(9,I),I=1,19)/0,0,0,0,15,0,0,0,0,0,25,0,0,0,0,0,26,
      : 0,27/
       DATA (ITRHE1(10,I),I=1,19)/0,0,0,11,0,0,0,22,0,0,0,0,0,28,0,0,
      : 0,29,0/
       DATA (ITRHE1(11,I),I=1,19)/2,0,8,0,0,0,20,0,25,0,0,0,30,0,31,
      : 4*0/
       DATA (ITRHE1(12,I),I=1,19)/0,0,0,12,0,0,0,23,0,0,0,0,0,32,5*0/
       DATA (ITRHE1(13,I),I=1,19)/0,0,0,0,16,0,0,0,0,0,30,0,0,0,0,0,
      : 33,0,0/
       DATA (ITRHE1(14,I),I=1,19)/0,6,0,0,0,19,0,0,0,28,0,32,0,0,0,34,
      : 0,0,0/
       DATA (ITRHE1(15,I),I=1,19)/0,0,0,0,17,0,0,0,0,0,31,0,0,0,0,0,0,
      : 0,0/
       DATA (ITRHE1(16,I),I=1,19)/0,0,0,13,0,0,0,24,0,0,0,0,0,34,0,0,
      : 0,0,0/
       DATA (ITRHE1(17,I),I=1,19)/0,0,0,0,0,0,21,0,26,0,0,0,33,0,0,
      : 0,0,0,0/
       DATA (ITRHE1(18,I),I=1,19)/0,0,0,0,0,0,0,0,0,0,29,0,0,0,0,0,0,
      : 0,0,0/
       DATA (ITRHE1(19,I),I=1,19)/0,0,0,0,0,0,0,0,0,27,0,0,0,0,0,0,0,
      : 0,0,0/
C
C      IONIZED HELIUM OSCILLATOR STRENGTHS
C
       DATA OSCHE2/4.162E-1,7.910E-2,2.899E-2,1.394E-2,7.800E-3,
      :4.814E-3,3.184E-3,2.216E-3,1.605E-3, 6.408E-1,1.193E-1,
      :4.467E-2,2.209E-2,1.271E-2,8.037E-3,5.429E-3,3.851E-3,
      : 8.420E-1,1.506E-1,5.585E-2,2.768E-2,1.604E-2,1.023E-2,
      :6.981E-3, 1.038,.1794,6.551E-2,3.229E-2,1.872E-2,1.195E-2,
      : 1.231,.2070,7.455E-2,3.644E-2,2.102E-2, 1.424,.234,.08315,
      :.04038, 1.616,.2609,.09163, 1.807,.2876, 1.999/
C
C      IONIZED HELIUM TRANSITION MATRIX
C
       DATA (ITRHE2(1,I),I=1,10)/ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9/
       DATA (ITRHE2(2,I),I=1,10)/ 1, 0,10,11,12,13,14,15,16,17/
       DATA (ITRHE2(3,I),I=1,10)/ 2,10, 0,18,19,20,21,22,23,24/
       DATA (ITRHE2(4,I),I=1,10)/ 3,11,18, 0,25,26,27,28,29,30/
       DATA (ITRHE2(5,I),I=1,10)/ 4,12,19,25, 0,31,32,33,34,35/
       DATA (ITRHE2(6,I),I=1,10)/ 5,13,20,26,31, 0,36,37,38,39/
       DATA (ITRHE2(7,I),I=1,10)/ 6,14,21,27,32,36, 0,40,41,42/
       DATA (ITRHE2(8,I),I=1,10)/ 7,15,22,28,33,37,40, 0,43,44/
       DATA (ITRHE2(9,I),I=1,10)/ 8,16,23,29,34,38,41,43, 0,45/
       DATA (ITRHE2(10,I),I=1,10)/9,17,24,30,35,39,42,44,45, 0/
C
C      NEGATIVE HYDROGEN ION IONIZATION FREQUENCY
C
       DATA FRQHM/1.874E14/
C
C      HYDROGEN IONIZATION FREQUENCIES
C
       DATA FRQH/3.28799E15,0.821997E15,0.365332E15,0.205499E15,
      :0.131519E15,0.0913329E15,0.0671018E15,0.0513748E15,
      :0.0405924E15,0.0328799E15,0.0271735E15,0.0228333E15,
      :0.0194556E15,0.0167755E15,0.0146133E15,0.0128437E15/
C
C      NEUTRAL HELIUM IONIZATION FREQUENCIES
C
       DATA FRQHE1/5.94520E15,1.15305E15,0.957439E15,0.876230E15,
      :0.811774E15,0.451896E15,0.400142E15,0.381976E15,0.362850E15,
      :0.366032E15,0.362480E15,0.240134E15,0.217774E15,0.212670E15,
      :0.202689E15,0.205704E15,0.202057E15,0.202703E15,0.199689E15,
      :0.131520E15,0.0913331E15,0.0671018E15,0.0513748E15,
      :0.0405924E15,0.0328799E15,0.0271735E15,0.0228333E15,
      :0.0194556E15,0.0167755E15,0.0146133E15,0.0128437E15/
C
C      IONIZED HELIUM IONIZATION FREQUENCIES
C
       DATA FRQHE2/13.1520E15,3.28799E15,1.46133E15,0.821997E15,
      :0.526078E15,0.365332E15,0.268407E15,0.205499E15,0.162370E15,
      :0.131519E15,0.108694E15,0.0913329E15,0.0778222E15,
      :0.0671018E15,0.0584532E15,0.0513748E15,0.0455085E15,
      :0.0405924E15,0.0364320E15,0.0328799E15,0.0298230E15,
      :0.0271735E15,0.0248619E15,0.0228333E15,0.0210431E15,
      :0.0194556E15,0.0180411E15,0.0167755E15,0.0156385E15,
      :0.0146133E15,0.0136857E15,0.0128437E15/
C
```

```
END
```

C. Program ANDERS

ANDERS is listed in its entirety, except for those routines that are essentially identical to ones in GRAY and LTE and for a very large number of DATA statements in COLRAT, whose inclusion would not contribute to the understanding of the program. The location of the omitted data statements is indicated in the listing. The omitted subroutines are:

DFRE (differs from FRE in that the temperature derivative is calculated)

EDDFAC

FRE

GAUNT

GENER

HEGAUNT

HMINUS

LINSLV

MATINV

In addition, use is made of the following CRAY FORTLIB routines: CREATE, which creates a file; DESTROY, which deletes a file; USERINFO, which returns information about the user's account and job (it is called here so that disk scratch-file names will be unique); IOSTATUS, which has the effect of halting the CPU until a buffered I/O operation is complete; RDABS and WRABS, which perform a function similar to that of the disk I/O routines employed in the MAH code [36]; and XTENDABS, which is used here to extend the disk scratch files to the proper size.

```
C     PROGRAM ANDERS
C
C     AN ADAPTION OF PORTIONS OF THE MIHALAS (1975) CODE TO THE
C     ANDERSON ALGORITHM FOR THE EFFICIENT SOLUTION OF LARGE NUMBERS OF
C     TRANSFER EQUATIONS IN NON-LTE.
C
C     THIS CODE ALSO OPTIONALLY EMPLOYS THE RADIATIVE/COLLISIONAL
C     SWITCHING TECHNIQUE OF HUMMER AND VOELS (1988).
C
C     PARAMETERS:
C
C     MNB       MAXIMUM NUMBER OF FREQUENCY BLOCKS
C     MNDEPTH   MAXIMUM NUMBER OF DEPTH POINTS
C     MNJ       MAXIMUM TOTAL NUMBER OF FREQUENCIES
C     MNJC      MAXIMUM NUMBER OF CONTINUUM FREQUENCY POINTS
C     MNNN      NUMBER OF VARIABLES LINEARIZED
C     MNTRH     MAXIMUM NUMBER OF HYDROGEN TRANSITIONS
C     MNTRHE1   MAXIMUM NUMBER OF NEUTRAL HELIUM TRANSITIONS
C     MNTRHE2   MAXIMUM NUMBER OF IONIZED HELIUM TRANSITIONS
C     MQH       MAXIMUM QUANTUM NUMBER IN PARTITION SUMS OF HYDROGEN
C     MQHE1     MAXIMUM QUANTUM NUMBER IN PARTITION SUMS OF HELIUM I
C     NEQN      TOTAL NUMBER OF ATOMIC STATES
C     NQUAD     NUMBER OF QUADRATURE POINTS PER LINE.
C                   SINCE SIMPSON'S RULE IS USED, THIS MUST BE ODD.
C     NLH       NUMBER OF NON-LTE HYDROGEN LEVELS
C     NLHE1     NUMBER OF NON-LTE HELIUM LEVELS
C     NLHE1S    TOTAL HELIUM LEVELS
C     NLHE2     NUMBER OF NON-LTE IONIZED HELIUM LEVELS
C     NLHE2S    TOTAL IONIZED HELIUM LEVELS
C     NLHS      TOTAL HYDROGEN LEVELS
C
C     ACCOF     SAHA ACTIVITY COEFFICIENT
C     BBCOF     PLANK FUNCTION COEFFICIENT
C     CC        VELOCITY OF LIGHT
C     DELQUAD   FRACTION OF DOPPLER WIDTH PER LINE INTEGRATION INTERVAL
C     DOPCOF    DOPPLER WIDTH COEFFICIENT
C     EMASS     ELECTRON MASS
C     ESU       ELECTRON CHARGE
C     HK        PLANK'S CONSTANT OVER BOLTZMANN'S CONSTANT
C     HP        PLANK'S CONSTANT
C     HYDCOF    HYDROSTATIC EQUATION RADIATIVE COEFFICIENT
C     KB        BOLTZMANN'S CONSTANT
C     MHYD      MASS OF HYDROGEN ATOM
C     PI        PI
C     PIE2MC    CLASSICAL ELECTRON ABSORPTION COEFFICIENT
C     SCOF      RADIATIVE RATE COEFFICIENT
C     SIGE      ELECTRON THOMPSON CROSS SECTION
C
C     VARIABLES:
C
C     FEXIT     FLAG TO EXIT
C     FLTE      FLAG TO ASSUME LTE
C     FPRINT    FLAG TO PRINT DIAGNOSTICS
C     FSWITCH   EMPLOY RADIATIVE-COLLISIONAL SWITCHING
C     RULE      FLAG TO INCLUDE VARIOUS OPACITIES
C                   ONLY CURRENT USE IS RULE(2) TO INCLUDE LINES
C
C     BLOCK     BLOCK ASSIGNMENTS
C     ITPTRH    POINTS TO HYDROGEN LINE FREQUENCIES
C     ITPTRHE1  POINTS TO NEUTRAL HELIUM LINE FREQUENCIES
C     ITPTRHE2  POINTS TO IONIZED HELIUM LINE FREQUENCIES
C     ITRH      TRANSITION INDICES FOR HYDROGEN;
C                       I.E. ITRH(L,U) IS TRANSITION INDEX OF
C                       HYDROGEN L LEVEL TO U LEVEL.
C     ITRHE1    TRANSITION INDICES OF NEUTRAL HELIUM
C     ITRHE2    TRANSITION INDICES OF SINGLY-IONIZED HELIUM
C     LOWH      LOWER LEVEL OF DOMINANT HYDROGEN TRANSITION AT THE
C                       SPECIFIED FREQUENCY
C     LOWHE1    " OF NEUTRAL HELIUM
C     LOWHE2    " OF IONIZED HELIUM
C     LOWERH    LOWER LEVEL OF HYDROGEN TRANSITIONS REQUESTED BY USER
C     LOWERHE1  " OF NEUTRAL HELIUM
C     LOWERHE2  " OF IONIZED HELIUM
C     NB        NUMBER OF FREQUENCY BLOCKS
C     NDEPTH    NUMBER OF DEPTH POINTS
C     NITER     NUMBER OF ITERATIONS TO MAKE
C     NJ        NUMBER OF FREQUENCIES
C     NNE       POINTER TO ELECTRON NUMBER IN MATRICES
C     NNN       " TO END OF MATRIX
```

```
C       NNT        " TO TEMPERATURE
C       NTRH       NUMBER OF HYDROGEN TRANSITIONS
C       NTRHE1 " OF NEUTRAL HELIUM TRANSITIONS
C       NTRHE2 " OF IONIZED HELIUM TRANSITIONS
C       UPH        UPPER LEVEL OF DOMINANT HYDROGEN TRANSITION AT THE
C                          FREQUENCY SPECIFIED.
C       UPHE1      " OF NEUTRAL HELIUM
C       UPHE2      " OF IONIZED HELIUM
C       UPPERH     UPPER HYDROGEN LEVELS OF TRANSITIONS REQUESTED BY USER
C       UPPERHE1   " OF NEUTRAL HELIUM
C       UPPERHE2   " OF IONIZED HELIUM
C
C       A          A MATRIX OF LINEARIZATION
C       AN         LHS OF POPULATION EQUATIONS
C       ANS        RESULT OF POPULATION CALCULATION
C       B          B MATRIX OF LINEARIZATION
C       BN         RHS OF POPULATION EQUATIONS
C       C          C-MATRIX OF LINEARIZATION
C       CHI        OPACITY MATRIX
C       CR         COLLISION RATES FOR HYDROGEN
C       CRHE1      " FOR NEUTRAL HELIUM
C       CRHE2      " FOR IONIZED HELIUM
C       DCRDT      TEMPERATURE DERIVATIVE OF CR
C       DCRHE1     " OF CRHE1
C       DCRHE2     " OF CRHE2
C       DRH        TEMPERATURE DERIVATIVE OF H RADIATIVE BRACKETS
C       DRHE1      " FOR NEUTRAL HELIUM
C       DRHE2      " FOR IONIZED HELIUM
C       DSHDT      TEMPERATURE DERIVATIVE OF H UPPER STATE SUM
C       DSHEDT     " FOR HELIUM
C       ETA        EMISSIVITY MATRIX
C       FF         FREE-FREE CUTOFF (TO ACCOUNT FOR UPPER STATES)
C       FH         EDDINGTON FACTOR FOR FLUX
C       FK         EDDINGTON FACTOR FOR RADIATIVE PRESSURE
C       FREQ       FREQUENCY GRID
C       FRQH       IONIZATION FREQUENCIES OF HYDROGEN
C       FRQHE1     " OF NEUTRAL HELIUM
C       FRQHE2     " OF IONIZED HELIUM
C       GH         STATISTICAL WEIGHTS OF HYDROGENIC LEVELS
C       GHE1       " OF NEUTRAL HELIUM
C       GRAV       SURFACE GRAVITY
C       HO         SURFACE FLUX
C       OSCH       HYDROGEN OSCILLATOR STRENGTHS LISTED BY TRANSITION INDEX
C       OSCHE1     NEUTRAL HELIUM "
C       OSCHE2     IONIZED HELIUM "
C       LAMC       RADIATIVE SWITCHING PARAMETER FOR CONTINUUM
C       LAML       LINE SWITCHING PARAMETER
C       M          MASS GRID
C       MU1        NUCLEI PER PROTON
C       N          HYDROGEN NUMBER DENSITIES
C       NE         ELECTRON DENSITY
C       NHE1       NEUTRAL HELIUM DENSITIES
C       NHE1S      LTE NEUTRAL HELIUM DENSITIES
C       NHE2       IONIZED HELIUM DENSITIES
C       NHE2S      LTE IONIZED HELIUM DENSITIES
C       NHE3       DOUBLY-IONIZED HELIUM DENSITIES
C       NM         FICTIONAL MASSIVE PARTICLE DENSITY
C       NPROT      PROTON DENSITY
C       NS         LTE HYDROGEN DENSITIES
C       NTOT       TOTAL PARTICLE DENSITY
C       Q          RHS OF LINEARIZATION
C       RAD        MEAN INTENSITY OF RADIATION
C       RH         HYDROGEN RADIATIVE BRACKETS
C       RHE1       NEUTRAL HELIUM "
C       RHE2       IONIZED HELIUM "
C       SIG        HYDROGEN CROSS-SECTIONS PLUS FREE-FREE
C       SIGHE1     NEUTRAL HELIUM "
C       SIGHE2     IONIZED HELIUM "
C       SUMH       HYDROGEN UPPER STATE SUM
C       SUMHE      HELIUM UPPER STATE SUMS
C       TEMP       TEMPERATURE
C       TLINE      TEMPERATURE ASSUMED TO DETERMINE LINE FREQUENCY GRID
C       WT         FREQUENCY QUADRATURE WEIGHTS MODIFIED BY SWITCHING
C       WTO        UNMODIFIED FREQUENCY QUADRATURE WEIGHTS
C       Y          RATIO OF HELIUM TO HYDROGEN BY NUMBER
C       ZTOT       TOTAL NUMBERS OF OTHER ELEMENTS
C
C       COMPILER DIRECTIVES:  NUMEROUS SHORT LOOP DIRECTIVES ARE PRESENT
C       IN THE PROGRAM.  THESE ASSUME THAT MNNN IS LESS THAN 64.
C
```

```
      CLICHE COMA
C
C     INCLUDES PARAMETERS
C
      INTEGER MNB, MNDEPTH, MNJC, MQH, MQHE1, NLH, NLHE1, NLHE1S, NLHE2
      INTEGER NLHE2S, NLHS, NQUAD
      PARAMETER (MNB=80, MNDEPTH=70, MNJC=105, MQH=16, MQHE1=31, NLH=5)
      PARAMETER (NLHE1=19, NLHE1S=25, NLHE2=10, NLHE2S=15, NLHS=10)
      PARAMETER (NQUAD=7)
C
      INTEGER MNTRH, MNTRHE1, MNTRHE2
      PARAMETER (MNTRH=10, MNTRHE1=14, MNTRHE2=10)
C
      INTEGER MNJ
      PARAMETER (MNJ=MNJC+NQUAD*(MNTRH+MNTRHE1+MNTRHE2))
C
      INTEGER NEQN, MNNN
      PARAMETER (NEQN=NLHE1+NLHE2+NLH+2, MNNN=MNB+2)
C
      REAL CC, DELQUAD, EMASS, ESU, HP, KB, MHYD, PI
      PARAMETER (CC=2.997925E10, DELQUAD=0.6, EMASS=9.10953E-28)
      PARAMETER (ESU=4.80325E-10, HP=6.62618E-27, KB=1.38066E-16)
      PARAMETER (MHYD=1.67265E-24, PI=3.141592654)
C
      REAL ACCOF, BBCOF, DOPCOF, HK, HYDCOF, PIE2MC, SCOF, SIGE
      PARAMETER (ACCOF=2.074E-16, BBCOF=2.*HP/(CC*CC), DOPCOF=4.286E-7)
      PARAMETER (HK=HP/KB, HYDCOF=4.*PI/(CC*KB))
      PARAMETER (PIE2MC=PI*ESU*ESU/(EMASS*CC), SCOF=4.*PI/HP)
      PARAMETER (SIGE=8.*PI*ESU*ESU*ESU*ESU/
     :               (3.*EMASS*EMASS*CC*CC*CC*CC))
C
      INTEGER BLOCK(MNJ), ITPTRH, ITPTRHE1, ITPTRHE2, NB, NDEPTH, NITER
      INTEGER NJ, NNE, NNN, NNT, NTRH, NTRHE1, NTRHE2
      EQUIVALENCE (NNE,NNN)
C
      LOGICAL FEXIT, FLTE, FPRINT, FSWITCH, RULE(2)
C
      REAL B(MNNN,MNNN), CHI(MNJ,MNB+3)
      REAL DRH(NLH+1,NLH+1), DRHE1(NLHE1+1,NLHE1+1)
      REAL DRHE2(NLHE2+1,NLHE2+1), DSHDT(MNDEPTH), DSHEDT(2,MNDEPTH)
      REAL ETA(MNJ,MNB+3), FF(MNJ,3), FH(MNJ)
      REAL FK(MNJ,MNDEPTH), FREQ(MNJ), GRAV, HO, Q(MNNN), LAMC
      REAL LAML, LINERR, M(MNDEPTH), MU1, N(NLH,MNDEPTH), NE(MNDEPTH)
      REAL NHE1(NLHE1,MNDEPTH), NHE1S(NLHE1S,MNDEPTH)
      REAL NHE2(NLHE2,MNDEPTH), NHE2S(NLHE2S,MNDEPTH), NHE3(MNDEPTH)
      REAL NM(MNDEPTH), NPROT(MNDEPTH), NS(NLHS,MNDEPTH), NTOT(MNDEPTH)
      REAL RAD(MNJ,MNDEPTH), RH(NLH+1,NLH+1)
      REAL RHE1(NLHE1+1,NLHE1+1), RHE2(NLHE2+1,NLHE2+1)
      REAL SIG(NLHS+1,MNJ), SIGHE1(NLHE1S,MNJ)
      REAL SIGHE2(NLHE2S,MNJ), SUMH(MNDEPTH), SUMHE(2,MNDEPTH)
      REAL TEMP(MNDEPTH), TLINE, VV(MNJ), WT(MNJ), Y, ZTOT
C
      COMMON //ITPTRH, ITPTRHE1, ITPTRHE2, BLOCK, NB, NDEPTH, NITER,
     : NJ, NNN, NNT, NTRH, NTRHE1, NTRHE2, FEXIT, FLTE, FPRINT,
     : FSWITCH, RULE, Q, B, CHI, DRH, DRHE1, DRHE2,
     : DSHDT, DSHEDT, ETA, FF, FH, FK, FREQ, GRAV, HO, LAMC,
     : LAML, M, MU1, N, NE, NHE1, NHE1S, NHE2, NHE2S, NHE3, NM, NPROT,
     : NS, NTOT, RAD, RH, RHE1, RHE2, SIG, SIGHE1, SIGHE2, SUMH,
     : SUMHE, TEMP, TLINE, VV, WT, Y, ZTOT
C
      ENDCLICHE

      CLICHE COMAI
C
      INTEGER ITRH(NLH,NLH), ITRHE1(NLHE1,NLHE1), ITRHE2(NLHE2,NLHE2)
      INTEGER LOWERH(MNTRH), LOWERHE1(MNTRHE1), LOWERHE2(MNTRHE2)
      INTEGER LOWH(MNJ), LOWHE1(MNJ), LOWHE2(MNJ), UPH(MNJ), UPHE1(MNJ)
      INTEGER UPHE2(MNJ), UPPERH(MNTRH), UPPERHE1(MNTRHE1)
      INTEGER UPPERHE2(MNTRHE2)
      REAL FRQH(MQH), FRQHE1(MQHE1), FRQHE2(2*MQH), GH(MQH)
      REAL GHE1(MQHE1), GHE2(2*MQH), OSCH(10), OSCHE1(34), OSCHE2(45)
      EQUIVALENCE (GHE2(1),GH(1))
C
      COMMON /COMAI/ITRH, ITRHE1, ITRHE2, LOWERH, LOWERHE1, LOWERHE2,
     : LOWH, LOWHE1, LOWHE2, UPH, UPHE1, UPHE2, UPPERH, UPPERHE1,
     : UPPERHE2, FRQH, FRQHE1, FRQHE2, GHE1, GHE2, OSCH, OSCHE1, OSCHE2
C
      ENDCLICHE

      CLICHE COMC
```

```
C
      REAL AN(NEQN,NEQN), ANS(NEQN), BN(NEQN), CR(NLH,NLH+1)
      REAL CRHE1(NLHE1,NLHE1+1), CRHE2(NLHE2,NLHE2+1)
      REAL DCRDT(NLH,NLH+1), DCRHE1(NLHE1,NLHE1+1)
      REAL DCRHE2(NLHE2,NLHE2+1)
C
      COMMON /COMC/AN, ANS, BN, CR, CRHE1, CRHE2, DCRDT, DCRHE1, DCRHE2
C
      ENDCLICHE

      CLICHE COMF
C
      REAL A(MNNN,MNNN), C(MNNN,MNNN)
C
      COMMON /COMF/A, C
C
      ENDCLICHE

      CLICHE COMW
C
      REAL WTO(MNJ)
      COMMON /COMW/WTO
C
      ENDCLICHE

      PROGRAM ANDERS
C
C     ENTRY POINT
C
      IMPLICIT NONE
C
C     MACROS
C
      COMA
C
C     LOCAL VARIABLES
C
      CHARACTER USER*6, ACC*6, DROP*8, SUFFIX*1
      INTEGER LENGTH
C
C     EXTERNAL PROCEDURES
C
      REAL BLOCKS, CONTROL, CREATE, DESTROY, EXIT, LINK, PUTOUT, SETUP
      REAL USERINFO
C
      EXTERNAL BLOCKS, CONTROL, CREATE, DESTROY, EXIT, LINK, PUTOUT
      EXTERNAL SETUP, USERINFO
C
C     START OF EXECUTABLE STATEMENTS.
C
C     THE FILE input CONTAINS A FIRST-APPROXIMATION MODEL.
C     THE FILE output CONTAINS THE MODEL HEREIN CALCULATED.
C     THE FILE monitor CONTAINS ALL OTHER OUTPUT.
C
      CALL LINK("UNIT5=(input,OPEN,TEXT),UNIT12=(output,CREATE,TEXT),
     : UNIT6=(monitor,CREATE,TEXT)//")
C
C     READ IN THE FIRST APPROXIMATION AND SET UP EVERYTHING
C     PREPARATORY TO BEGINNING CALCULATIONS.
C
      CALL SETUP
C
C     GET USER SUFFIX (SO THAT SCRATCH FILES CAN BE UNIQUELY NAMED)
C
      CALL USERINFO(USER,ACC,DROP,SUFFIX)
C
C     CREATE SCRATCH FILES
C
      LENGTH=MNNN*(MNNN+1)*(NDEPTH-1)
      CALL CREATE(8,'%scr8'//SUFFIX,4,LENGTH)
      LENGTH=MNNN*NDEPTH
      CALL CREATE(9,'%scr9'//SUFFIX,4,LENGTH)
C
C     SET UP FREQUENCY BINNING.
C
      CALL BLOCKS
C
C     ENTER MAIN CONTROL ROUTINE AND CARRY OUT THE CALCULATIONS.
C
      CALL CONTROL
```

```
C
C     WRITE THE RESULTS.
C
      CALL PUTOUT
C
C     DELETE SCRATCH FILES AND EXIT.
C
      CALL DESTROY('%scr8'//SUFFIX)
      CALL DESTROY('%scr9'//SUFFIX)
      CALL EXIT(0)
      END

      SUBROUTINE BLOCKS
C
C     SET UP BLOCK ASSIGNMENTS FOR FREQUENCIES
C
      IMPLICIT NONE
C
C     MACROS
C
      COMA
      COMAI
C
C     LOCAL VARIABLES
C
      INTEGER I, I1, I2, I3, ID, IJ, IL, INDX(100), K
      REAL DEPTH, RAT
C
C     PARAMETER DETERMINING THE MASS RANGE OVER WHICH TO BIN
C     FREQUENCIES.
C
      REAL MRAT
      PARAMETER (MRAT=10.0)
C
C     EXTERNAL PROCEDURES
C
      REAL EXIT, GENER, INDEXX
      EXTERNAL EXIT, GENER, INDEXX
C
C     START OF EXECUTABLE STATEMENTS
C
      DO 10 IJ=1,NJ
         BLOCK(IJ)=0
10    CONTINUE
C
C     CALCULATE OPACITIES
C
      CALL GENER
C
C     CALCULATE DEPTH OF FORMATION FOR EACH FREQUENCY
C
      DO 30 IJ=1,NJ
         DEPTH=CHI(IJ,1)*M(1)
         DO 20 ID=2,NDEPTH
            DEPTH=DEPTH+0.5*(CHI(IJ,ID+1)+CHI(IJ,ID))*(M(ID+1)-M(ID))
            IF (DEPTH.GT.(2./3.))THEN
               VV(IJ)=M(ID)
               GO TO 30
            ENDIF
20       CONTINUE
         WRITE (6,*)'MASS GRID TOO SHALLOW'
         CALL EXIT(1)
30    CONTINUE
C
C     CONTINUUM POINTS FIRST
C
C     ALL FREQUENCIES IN THE SAME IONIZATION CONTINUUM AND WITH A
C     FORMATION DEPTH WITHIN A FACTOR OF MRAT OF EACH OTHER ARE
C     GROUPED TOGETHER.
C
      NB=1
      I2=1
      I3=1
      DEPTH=VV(1)
      DO 40 IJ=1,ITPTRH-1
C
C     ARE WE AT THE EDGE OF A NEW IONIZATION CONTINUUM?
C     IF SO, START A NEW BIN.
C
C     NOTE THAT THE PROGRAM ASSUMES THAT ALL HYDROGEN IONIZATION
```

```
C     CONTINUA START AT THE SAME FREQUENCIES THAT CORRESPONDING
C     HELIUM-II CONTINUA START.  (THIS IS VERY NEARLY TRUE.)
C
C     HYDROGEN AND HELIUM-II LINE TRANSITIONS, ON THE OTHER HAND,
C     ARE TREATED AS HAVING NO OVERLAP.
C
         IF (SIGHE1(I2,IJ).EQ.0.AND.SIGHE1(I2,MAX(1,IJ-1)).NE.0)THEN
            NB=NB+1
            DEPTH=VV(IJ)
            I2=I2+1
         ELSE
     :    IF (SIGHE2(I3,IJ).EQ.0.AND.SIGHE2(I3,MAX(1,IJ-1)).NE.0)THEN
            NB=NB+1
            DEPTH=VV(IJ)
            I3=I3+1
         ENDIF
C
C     SEE IF THE DEPTH OF FORMATION HAS CHANGED ENOUGH TO WARRANT
C     STARTING A NEW BIN.
C
         RAT=MAX(VV(IJ),DEPTH)/MIN(VV(IJ),DEPTH)
         IF (RAT.GT.MRAT)THEN
            NB=NB+1
            DEPTH=VV(IJ)
         ENDIF
         BLOCK(IJ)=NB
40    CONTINUE
      IF (.NOT.RULE(2))GO TO 260
C
C     LINE POINTS
C
C     WE DIVIDE THE LINES INTO THREE CLASSES:  RESONANCE, LOW
C     EXCITATION, AND HIGH EXCITATION.  THE DIVISIONS ARE SOMEWHAT
C     ARBITRARY.
C
C     RESONANCE LINES FIRST:  TREAT EACH LINE SEPARATELY.
C
      IF (NTRH.LT.1)GO TO 65
      I=ITPTRH
50       IF (LOWH(I).EQ.1)THEN
            NB=NB+1
            DEPTH=VV(I+NQUAD-1)
            BLOCK(I+NQUAD-1)=NB
            DO 60 IJ=I+NQUAD-2,I,-1
               RAT=MAX(VV(IJ),DEPTH)/MIN(VV(IJ),DEPTH)
               IF (RAT.GT.MRAT)THEN
                  NB=NB+1
                  DEPTH=VV(IJ)
               ENDIF
               BLOCK(IJ)=NB
60          CONTINUE
            I=I+NQUAD
            GO TO 50
         ENDIF
         I1=I
65    IF (NTRHE1.LT.1)GO TO 85
      I=ITPTRHE1
70       IF (LOWHE1(I).EQ.1)THEN
            NB=NB+1
            DEPTH=VV(I+NQUAD-1)
            BLOCK(I+NQUAD-1)=NB
            DO 80 IJ=I+NQUAD-2,I,-1
               RAT=MAX(VV(IJ),DEPTH)/MIN(VV(IJ),DEPTH)
               IF (RAT.GT.MRAT)THEN
                  NB=NB+1
                  DEPTH=VV(IJ)
               ENDIF
               BLOCK(IJ)=NB
80          CONTINUE
            I=I+NQUAD
            GO TO 70
         ENDIF
         I2=I
C
C     SINCE RESONANCE HELIUM LINES ARE USUALLY ASSUMED TO BE IN DETAILED
C     BALANCE, LINES FROM THE N=2 LEVEL MAY BE TREATED AS RESONANCE LINES.
C     THE PROGRAM DETERMINES WHETHER OR NOT TO DO THIS BASED ON THE LOW
C     LEVEL OF THE FIRST HELIUM LINE READ IN.
C
85    IF (NTRHE2.LT.1)GO TO 105
```

```
       I=ITPTRHE2
          I3=LOWHE2(I)
 90       IF (LOWHE2(I).EQ.I3)THEN
             NB=NB+1
             DEPTH=VV(I+NQUAD-1)
             BLOCK(I+NQUAD-1)=NB
             DO 100 IJ=I+NQUAD-2,I,-1
                RAT=MAX(VV(IJ),DEPTH)/MIN(VV(IJ),DEPTH)
                IF (RAT.GT.MRAT)THEN
                   NB=NB+1
                   DEPTH=VV(IJ)
                ENDIF
                BLOCK(IJ)=NB
 100         CONTINUE
             I=I+NQUAD
             GO TO 90
          ENDIF
          I3=I
C
C      SUBORDINATE LINES
C
C      BIN ALL MOERATE-EXCITATION LINES WITH SAME LOWER LEVEL TOGETHER.
C      FOR HYDROGEN, ONLY THE N=2 LEVEL IS SO TREATED.
C
C      DETERMINE WHICH SET OF FREQUENCIES CORRESPOND TO A SINGLE LOWER
C      LEVEL.
C
 105   IF (NTRH.LT.1)GO TO 140
       DO 110 I=I1,ITPTRHE1-1,NQUAD
          IF (LOWH(I).NE.2)THEN
             K=I
             GO TO 120
          ENDIF
 110   CONTINUE
C
C      NOW PRODUCE AN INDEX ARRAY
C
 120   CALL INDEXX(K-I1,VV(I1),INDX)
C
C      NOW SET UP BINS
C
       NB=NB+1
       IJ=INDX(K-I1)+I1-1
       DEPTH=VV(IJ)
       BLOCK(IJ)=NB
       DO 130 I=K-I1-1,1,-1
          IJ=INDX(I)+I1-1
          RAT=MAX(VV(IJ),DEPTH)/MIN(VV(IJ),DEPTH)
          IF (RAT.GT.MRAT)THEN
             NB=NB+1
             DEPTH=VV(IJ)
          ENDIF
          BLOCK(IJ)=NB
 130   CONTINUE
       I1=K
C
C      SAME IDEA FOR HELIUM IONS, EXCEPT THAT HERE LEVELS 2 THROUGH 5
C      ARE SO TREATED (ALL OF WHICH CORRESPOND TO A PRINCIPAL QUANTUM
C      NUMBER OF 2).
C
 140   IF (NTRHE1.LT.1)GO TO 180
       DO 150 I=I2,ITPTRHE2,NQUAD
             IF (LOWHE1(I).NE.LOWHE1(I2))THEN
                K=I
                GO TO 160
             ENDIF
 150      CONTINUE
C
C      NOW PRODUCE AN INDEX ARRAY
C
 160      CALL INDEXX(K-I2,VV(I2),INDX)
C
C      NOW SET UP BINS
C
          NB=NB+1
          IJ=INDX(K-I2)+I2-1
          DEPTH=VV(IJ)
          BLOCK(IJ)=NB
          DO 170 I=K-I2-1,1,-1
             IJ=INDX(I)+I2-1
```

```
         RAT=MAX(VV(IJ),DEPTH)/MIN(VV(IJ),DEPTH)
         IF (RAT.GT.MRAT)THEN
            NB=NB+1
            DEPTH=VV(IJ)
         ENDIF
         BLOCK(IJ)=NB
170   CONTINUE
C
C     NOW LOOP BACK FOR THE NEXT LINE
C
      I2=K
      IF (LOWHE1(K).LT.6.AND.K.LT.ITPTRHE2)GO TO 140
C
C     NOW IONIZED HELIUM; HERE LEVELS 2 THROUGH 4 ARE SO TREATED.
C     (OR LEVELS 3 THROUGH 4 IF LEVEL 2 LINES ARE TREATED AS RESONANCE.)
C
180   IF (NTRHE2.LT.1)GO TO 215
      IF (I3+NQUAD.GT.NJ) THEN
         K=NJ+1
         GO TO 200
      ENDIF
      DO 190 I=I3,NJ,NQUAD
         IF (LOWHE2(I).NE.LOWHE2(I3))THEN
            K=I
            GO TO 200
         ENDIF
190   CONTINUE
C
C     NOW PRODUCE AN INDEX ARRAY
C
200   CALL INDEXX(K-I3,VV(I3),INDX)
C
C     NOW SET UP BINS
C
      NB=NB+1
      IJ=INDX(K-I3)+I3-1
      DEPTH=VV(IJ)
      BLOCK(IJ)=NB
      DO 210 I=K-I3-1,1,-1
         IJ=INDX(I)+I3-1
         RAT=MAX(VV(IJ),DEPTH)/MIN(VV(IJ),DEPTH)
         IF (RAT.GT.MRAT)THEN
            NB=NB+1
            DEPTH=VV(IJ)
         ENDIF
         BLOCK(IJ)=NB
210   CONTINUE
C
C     NOW LOOP BACK FOR THE NEXT LINE
C
      I3=K
      IF (LOWHE2(K).LT.5.AND.K.LT.NJ)GO TO 180
C
C     NOW FOR HIGH-EXCITATION LINES.
C     THESE ARE ALL BINNED TOGETHER BY ION.
C
C     PRODUCE AN INDEX ARRAY
C
215   IF (NTRH.LT.1)GO TO 225
      CALL INDEXX(ITPTRHE1-I1,VV(I1),INDX)
C
C     NOW SET UP BINS
C
      NB=NB+1
      IJ=INDX(ITPTRHE1-I1)+I1-1
      DEPTH=VV(IJ)
      BLOCK(IJ)=NB
      DO 220 I=ITPTRHE1-I1-1,1,-1
         IJ=INDX(I)+I1-1
         RAT=MAX(VV(IJ),DEPTH)/MIN(VV(IJ),DEPTH)
         IF (RAT.GT.MRAT)THEN
            NB=NB+1
            DEPTH=VV(IJ)
         ENDIF
         BLOCK(IJ)=NB
220   CONTINUE
C
C     SAME FOR OTHER IONS
C
C     PRODUCE AN INDEX ARRAY
```

```
C
225   IF (NTRHE1.LT.1)GO TO 240
      IF (ITPTRHE2-I2.LE.0)GO TO 240
      CALL INDEXX(ITPTRHE2-I2,VV(I2),INDX)
C
C     NOW SET UP BINS
C
      NB=NB+1
      IJ=INDX(ITPTRHE2-I2)+I2-1
      DEPTH=VV(IJ)
      BLOCK(IJ)=NB
      DO 230 I=ITPTRHE2-I2-1,1,-1
         IJ=INDX(I)+I2-1
         RAT=MAX(VV(IJ),DEPTH)/MIN(VV(IJ),DEPTH)
         IF (RAT.GT.MRAT)THEN
            NB=NB+1
            DEPTH=VV(IJ)
         ENDIF
         BLOCK(IJ)=NB
230   CONTINUE
C
C     NOW IONIZED HELIUM
C
C     PRODUCE AN INDEX ARRAY
C
240   IF (NTRHE2.LT.1)GO TO 260
      IF (NJ+1-I3.LE.0)GO TO 260
      CALL INDEXX(NJ+1-I3,VV(I3),INDX)
C
C     NOW SET UP BINS
C
      NB=NB+1
      IJ=INDX(NJ-I3+1)+I3-1
      DEPTH=VV(IJ)
      BLOCK(IJ)=NB
      DO 250 I=NJ-I3,1,-1
         IJ=INDX(I)+I3-1
         RAT=MAX(VV(IJ),DEPTH)/MIN(VV(IJ),DEPTH)
         IF (RAT.GT.MRAT)THEN
            NB=NB+1
            DEPTH=VV(IJ)
         ENDIF
         BLOCK(IJ)=NB
250   CONTINUE
C
C     ALL DONE!  NOW SEE IF THERE ARE TOO MANY BLOCKS.
C
260   IF (NB.GT.MNB)THEN
         WRITE (6,*)NB,' BLOCKS EXCEEDS THE LIMIT OF ',MNB,' BLOCKS.'
         CALL EXIT(1)
      ENDIF
      WRITE (6,*)NB,' BLOCKS ALLOCATED.'
      NNT=NB+1
      NNN=NNT+1
      NNE=NNN
      RETURN
      END


      SUBROUTINE COLRAT(T,C,CHE1,CHE2)
C
C     CALCULATE COLLISION RATE COEFFICIENTS
C
      IMPLICIT NONE
C
      COMA
      COMAI
      COMC
C
C     LOCAL VARIABLES
C
      INTEGER I, J
      REAL C(NLH,NLH+1), CHE1(NLHE1,NLHE1+1), CHE2(NLHE2,NLHE2+1)
      REAL CA, CCON, CCR, E1, E5, EX, GAM, HKT, SRT, T, U0, U1, U2, V
      REAL XX
      PARAMETER (CCON=5.465E-11, CA=4.3144E-6)
C
C     HYDROGEN LINE GAMMA COEFFICIENTS
C
      REAL A1H(NLH,MQH),A2H(NLH,MQH),A3H(NLH,MQH),A4H(NLH,MQH)
      REAL A5H(NLH,MQH)
```

(106 lines of DATA statements omitted)

```
C
C     NEUTRAL HELIUM GAMMA COEFFICIENTS
C
      REAL A1HE1(NLHE1,MQHE1), A2HE1(NLHE1,MQHE1), A3HE1(NLHE1,MQHE1)
      REAL A4HE1(NLHE1,MQHE1), A5HE1(NLHE1,MQHE1)
```

(848 lines of DATA statements omitted)

```
C
C     IONIZED HELIUM GAMMA COEFFICIENTS
C
```

(461 lines of DATA statements omitted)

```
C
C     HYDROGEN IONIZATION WIDTH COEFFICIENTS
C
      REAL C0H(5), C1H(5), C2H(5), C3H(5), C4H(5),C5H(5)
      DATA C0H/-4.992595356440E+3, 3.996944587573E+4, 3.880203529243E+5,
     : 5.783644303438E+5, 9.018970256245E+4/
      DATA C1H/5.419223767021E+3, -3.766618400049E+4,-4.162393006165E+5,
     : -6.708925392728E+5, -2.434499092577E+5/
      DATA C2H/-2.327595426200E+3, 1.355412623572E+4, 1.756073001959E+5,
     : 3.038519700001E+5, 1.635626515436E+5/
      DATA C3H/4.935119971365E+2, -2.283539024763E+3,-3.632413884630E+4,
     : -6.687746329513E+4, -4.535682142076E+4/
      DATA C4H/-5.144533559385E+1, 1.761945551009E+2, 3.689123755488E+3,
     : 7.172363295058E+3, 5.662295249579E+3/
      DATA C5H/2.105014593961E+0, -4.743912931041E+0,-1.475337595301E+2,
     : -3.010521786188E+2, -2.646835159289E+2/
C
C     NEUTRAL HELIUM IONIZATION WIDTH COEFFICIENTS
C
      REAL C0HE1(19),C1HE1(19),C2HE1(19),C3HE1(19),C4HE1(19),C5HE1(19)
      DATA C0HE1/-491.4899465435, 4*52751.32955540, 6*192217.9888246,
     : 8*98300.97872971/
      DATA C1HE1/457.1636407635, 4*-50269.56309656, 6*-203859.6952899,
     : 8*-119238.9072374/
      DATA C2HE1/-158.1006713481, 4*19085.52783860, 6*86170.00350852,
     : 8*57480.84354338/
      DATA C3HE1/24.16267285172, 4*-3569.387671216, 6*-17994.83403347,
     : 8*-13389.89051919/
      DATA C4HE1/-1.360129503102, 4*327.5671481368, 6*1853.319498524,
     : 8*1505.703401430/
      DATA C5HE1/0.000000000000, 4*-11.78735527257, 6*-75.36190623852,
     : 8*-65.72418881018/
C
C     IONIZED HELIUM IONIZATION WIDTH COEFFICIENTS
C
      REAL C0HE2(10),C1HE2(10),C2HE2(10),C3HE2(6),C4HE2(8),C5HE2(6)
      DATA C0HE2/-74.26034311733, -2937.990286080, 19959.12590455,
     : 82886.99750248, 143404.3154991, 196634.6941767, 1181.3516,
     : 1440.1016, 2492.1250, 4663.3129/
      DATA C1HE2/71.67199736470, 3597.535164323, -19738.85742378,
     : -87905.12522636, -159143.2227411, -228914.1199763, -200.71191,
     : -259.75781, -624.84375,-1390.1250/
      DATA C2HE2/-24.25823369433, -1722.038557089, 7589.443988198,
     : 36603.23972799, 69194.12399282, 103961.9966259, 2*0., 30.101562,
     :  97.671874/
      DATA C3HE2/3.562032948247, 403.7822953371, -1407.065655238,
     : -7452.392601514, -14662.30211131, -22877.34494655/
      DATA C4HE2/-.1923825868069, -46.20255731757, 125.6697110972,
     : 743.2677181628, 1517.485430160, 2446.673087330, -2810.7812,
     : -1283.5625/
      DATA C5HE2/0.000000000000, 2.061471799965, -4.324390597780,
     : -29.14243384304, -61.59229459581, -102.2217157435/
C
C     START OF EXECUTABLE STATEMENTS
C
      SRT=SQRT(T)
      HKT=HK/T
      XX=LOG10(T)
C
C     CLEAR RATE COEFFICIENTS
C
      DO 20 I=1,NLH
         DO 10 J=1,NLH+1
            C(I,J)=0.
10       CONTINUE
20    CONTINUE
      DO 40 I=1,NLHE1
         DO 30 J=1,NLHE1+1
            CHE1(I,J)=0.0
```

```
30          CONTINUE
40       CONTINUE
         DO 60 I=1,NLHE2
            DO 50 J=1,NLHE2+1
               CHE2(I,J)=0.0
50          CONTINUE
60       CONTINUE
C
C        HYDROGEN IONIZATION RATES
C
C        FROM LENNON e.a. (1986)
C
         DO 70 I=1,NLH
            GAM=MAX(1.E-15,(C0H(I)/XX+C1H(I))/XX+C2H(I)+XX*(C3H(I)+
     :            XX*(C4H(I)+XX*C5H(I))))
            C(I,NLH+1)=CCON*SRT*GAM*EXP(-HKT*FRQH(I))
70       CONTINUE
C
C        HYDROGEN EXCITATION RATES
C
C        GROUND STATE TO 2ND AND 3RD LEVEL FROM AGGARWAL (1983)
C        2ND TO 3RD LEVEL FROM HATA et al. (1980)
C
C        OTHERS FROM MIHALAS (1978).
C
         DO 100 I=1,NLH
            DO 80 J=I+1,NLH
               U0=FRQH(I)-FRQH(J)
               U0=U0*HKT
               EX=EXP(-U0)
               GAM=MAX(1.E-15,(A1H(I,J)+T*(A2H(I,J)+T*A3H(I,J)))/
     :            (1.0+T*(A4H(I,J)+T*A5H(I,J))))
               CCR=CA*EX*GAM/SRT
               C(I,J)=CCR
               C(J,I)=GH(I)*CCR/(GH(J)*EX)
80          CONTINUE
            DO 90 J=NLH+1,MQH
               U0=FRQH(I)-FRQH(J)
               U0=U0*HKT
               EX=EXP(-U0)
               GAM=MAX(1.E-15,(A1H(I,J)+T*(A2H(I,J)+T*A3H(I,J)))/
     :            (1.0+T*(A4H(I,J)+T*A5H(I,J))))
               CCR=CA*EX*GAM/SRT
               C(I,NLH+1)=C(I,NLH+1)+CCR
90          CONTINUE
100      CONTINUE
C
C        HELIUM IONIZATION RATE COEFFICIENTS FROM LENNON et al. (1986).
C
         DO 110 I=1,NLHE1
            GAM=MAX(1.E-15,(C0HE1(I)/XX+C1HE1(I))/XX+C2HE1(I)+XX*(C3HE1(I)+
     :            XX*(C4HE1(I)+XX*C5HE1(I))))
            CHE1(I,NLHE1+1)=CCON*SRT*GAM*EXP(-HKT*FRQHE1(I))
110      CONTINUE
C
C        HELIUM EXCITATION RATES FROM AGGARWAL et al. (1978) FOR
C        GROUND LEVEL TO FIRST FOUR EXCITED LEVELS.
C        OTHER RATES FROM BERRINTON ET AL. (1985) OR FROM MIHALAS
C        et al. (1975).
C
         DO 140 I=1,NLHE1
            DO 120 J=I+1,NLHE1
               U0=FRQHE1(I)-FRQHE1(J)
               U0=U0*HKT
               EX=EXP(-U0)
               V=GHE1(I)/GHE1(J)
               GAM=MAX(1.E-15,(A1HE1(I,J)+T*(A2HE1(I,J)+T*A3HE1(I,J)))/
     :            (1.0+T*(A4HE1(I,J)+T*A5HE1(I,J))))
               CCR=CA*EX*GAM/SRT
               CHE1(I,J)=CCR
               CHE1(J,I)=V*CCR/EX
120         CONTINUE
            DO 130 J=NLHE1+1,MQHE1
               U0=FRQHE1(I)-FRQHE1(J)
               U0=U0*HKT
               EX=EXP(-U0)
               V=GHE1(I)/GHE1(J)
               GAM=MAX(1.E-15,(A1HE1(I,J)+T*(A2HE1(I,J)+T*A3HE1(I,J)))/
     :            (1.0+T*(A4HE1(I,J)+T*A5HE1(I,J))))
               CCR=CA*EX*GAM/SRT
```

```
          CHE1(I,NLHE1+1)=CHE1(I,NLHE1+1)+CCR
130     CONTINUE
140   CONTINUE
C
C     IONIZED HELIUM IONIZATION RATE COEFFICIENTS
C     FROM LENNON et al. (1986)
C
      DO 150 I=1,6
         GAM=MAX(1.E-15,(C0HE2(I)/XX+C1HE2(I))/XX+C2HE2(I)+XX*(C3HE2(I)+
     :         XX*(C4HE2(I)+XX*C5HE2(I))))
         CHE2(I,NLHE2+1)=CCON*SRT*GAM*EXP(-HKT*FRQHE2(I))
150   CONTINUE
C
C     IONIZATION RATES FROM MIHALAS et al. (1975)
C
      DO 160 I=7,NLHE2
         GAM=MAX(1.E-15,C0HE2(I)+(C1HE2(I)+C2HE2(I)*XX)*XX+
     :           C4HE2(I)/XX/XX)
         U0=HKT*FRQHE2(I)
         CHE2(I,NLHE2+1)=CCON*SRT*EXP(-U0)*GAM
160   CONTINUE
C
C     IONIZATION RATES FROM MIHALAS et al. (1975)
C
      DO 190 I=1,NLHE2
         DO 170 J=I+1,NLHE2
            U0=FRQHE2(I)-FRQHE2(J)
            U0=HKT*U0
            EX=EXP(-U0)
            GAM=MAX(1.E-15,(A1HE2(I,J)+T*(A2HE2(I,J)+T*A3HE2(I,J)))/
     :          (1.0+T*(A4HE2(I,J)+T*A5HE2(I,J))))
            CCR=CA*EX*GAM/SRT
            CHE2(I,J)=CCR
            CHE2(J,I)=GH(I)*CCR/(GH(J)*EX)
170      CONTINUE
         DO 180 J=NLHE2+1,2*MQH
            U0=FRQHE2(I)-FRQHE2(J)
            U0=HKT*U0
            EX=EXP(-U0)
            GAM=MAX(1.E-15,(A1HE2(I,J)+T*(A2HE2(I,J)+T*A3HE2(I,J)))/
     :          (1.0+T*(A4HE2(I,J)+T*A5HE2(I,J))))
            CCR=CA*EX*GAM/SRT
            CHE2(I,NLHE2+1)=CHE2(I,NLHE2+1)+CCR
180      CONTINUE
190   CONTINUE
      RETURN
      END


      SUBROUTINE CONTROL
C
C     OVERALL CONTROL SUBROUTINE FOR LINEARIZATION AND
C     LAMBDA ITERATION
C
      IMPLICIT NONE
C
      COMA
      COMAI
C
      INTEGER ID, I
C
      REAL EDDFAC, EXIT, GAB, GENER, LINEAR, LINIT, NUPOP
      EXTERNAL EDDFAC, EXIT, GAB, GENER, LINEAR, LINIT, NUPOP
C
      FEXIT=.FALSE.
      DO 30 I=1,NITER
C
C     LAMBDA ITERATION.
C     NOTE THAT GENER IS CALLED IN BLOCK PRIOR TO THE FIRST ITERATION;
C     THUS, THE FIRST ITERATION HERE NEED NOT MAKE A CALL TO GENER.
C
         IF (I.GT.1)CALL GENER
         CALL EDDFAC
C
C     ADJUST LAMBDAS IF FSWITCH=.TRUE.
C
         CALL LINIT(I)
         WRITE (6,*)'LAMC= ',LAMC,' LAML= ',LAML
         DO 10 ID=1,NDEPTH
            CALL NUPOP(ID)
10       CONTINUE
```

```
C
C     DIAGNOSTICS
C
         IF (FPRINT)CALL GAB
C
C     DIRECT LINEARIZATION
C
         CALL LINEAR
         DO 20 ID=1,NDEPTH
            CALL NUPOP(ID)
20       CONTINUE
         IF (FEXIT)GO TO 40
30    CONTINUE
      WRITE (6,*)'ITERATIONS FAILED TO CONVERGE'
40    CALL GAB
      RETURN
      END

      SUBROUTINE DGENER(IDD,PR)
C
C     CALCULATE OPACITIES AND DERIVATES
C
      IMPLICIT NONE
C
C     MACROS
C
      COMA
      COMAI
      COMC
C
C     LOCAL VARIABLES
C
      INTEGER I, IB, II, ID, IDD, IL, IJ, L, IT, U, J, JJ, NO
      REAL DD(NEQN,MNB+2), DSKKDT(MNJ,3), PR(MNNN+1), SKK(MNJ,3)
      REAL C(MNJ), DC, DE, DOP, DOPT, DT, E(MNJ), EX(MNJ), FRQO, HKT
      REAL HKTT, S, SIGMA, SIGMAT, SRT, SRT2, T1, X, XO, X1
C
C     EXTERNAL PROCEDURES
C
      REAL COLRAT, DFRE, MATINV, RATEQ
      EXTERNAL COLRAT, DFRE, MATINV, RATEQ
C
      ID=IDD
      CALL DFRE(ID,SKK,DSKKDT)
      T1=1.0/TEMP(ID)
      HKT=HK*T1
      HKTT=HKT*T1
      SRT=SQRT(TLINE/TEMP(ID))
      SRT2=SQRT(TEMP(ID))
C
C     GENERATE COLLISION RATES AND DERIVATIVES.
C
      CALL COLRAT(TEMP(ID),CR,CRHE1,CRHE2)
      DT=1.E-4*TEMP(ID)
      CALL COLRAT(TEMP(ID)+DT,DCRDT,DCRHE1,DCRHE2)
      DO 20 I=1,NLH
         DO 10 J=1,NLH+1
            DCRDT(I,J)=(DCRDT(I,J)-CR(I,J))/DT
10       CONTINUE
20    CONTINUE
      DO 40 I=1,NLHE1
         DO 30 J=1,NLHE1+1
            DCRHE1(I,J)=(DCRHE1(I,J)-CRHE1(I,J))/DT
30       CONTINUE
40    CONTINUE
      DO 60 I=1,NLHE2
         DO 50 J=1,NLHE2+1
            DCRHE2(I,J)=(DCRHE2(I,J)-CRHE2(I,J))/DT
50       CONTINUE
60    CONTINUE
      DO 80 I=1,NEQN
         DO 70 J=1,NNN
            DD(I,J)=0.0
70       CONTINUE
80    CONTINUE
      DO 90 I=1,NNN
         PR(I)=0.0
90    CONTINUE
C
C     OBTAIN RATE EQUATION MATRIX AND INVERT.
```

```
C
      CALL RATEQ(ID,NE(ID),AN,BN)
      CALL MATINV(AN,NEQN,NEQN)
C
C     RADIATIVE FIELD PERTURBATIONS
C     SKIP FOR LTE
C
      DO 100 IJ=1,NJ
         EX(IJ)=EXP(-HKT*FREQ(IJ))
100   CONTINUE
      IF (FLTE)GO TO 180
      DO 170 IJ=1,NJ
         DO 110 I=1,NEQN
            BN(I)=0.
110      CONTINUE
         S=SCOF*WT(IJ)/FREQ(IJ)
         IF (IJ.LT.ITPTRH)THEN
            DO 120 I=1,NLHE1
               BN(I)=S*LAMC*(NHE1S(I,ID)*EX(IJ)-NHE1(I,ID))*SIGHE1(I,IJ)
120         CONTINUE
            DO 130 I=1,NLHE2
               II=I+NLHE1
               BN(II)=S*LAMC*(NHE2S(I,ID)*EX(IJ)-NHE2(I,ID))*
     :                 SIGHE2(I,IJ)
130         CONTINUE
            DO 140 I=1,NLH
               II=I+NLHE1+NLHE2+1
               BN(II)=S*LAMC*(NS(I,ID)*EX(IJ)-N(I,ID))*SIG(I,IJ)
140         CONTINUE
         ELSE IF (IJ.LT.ITPTRHE1)THEN
            I=LOWH(IJ)
            J=UPH(IJ)
            IT=ITRH(I,J)
            JJ=J+NLHE1+NLHE2+1
            II=I+NLHE1+NLHE2+1
            FRQO=FRQH(I)-FRQH(J)
            DOP=SRT2*FRQO*DOPCOF
            X=DELQUAD*MOD(IJ-ITPTRH,NQUAD)*SRT
            SIGMA=PIE2MC*OSCH(IT)*EXP(-X*X)/DOP/1.7724539
            X=S*SIGMA*(N(I,ID)-GH(I)*N(J,ID)/GH(J))
            BN(II)=-X
            BN(JJ)=+X
         ELSE IF (IJ.LT.ITPTRHE2)THEN
            I=LOWHE1(IJ)
            J=UPHE1(IJ)
            IT=ITRHE1(I,J)
            FRQO=FRQHE1(I)-FRQHE1(J)
            DOP=SRT2*FRQO*DOPCOF*.5
            X=DELQUAD*MOD(IJ-ITPTRHE1,NQUAD)*SRT
            SIGMA=PIE2MC*OSCHE1(IT)*EXP(-X*X)/DOP/1.7724539
            X=S*SIGMA*(NHE1(I,ID)-GHE1(I)*NHE1(J,ID)/GHE1(J))
            BN(I)=-X
            BN(J)=+X
         ELSE
            I=LOWHE2(IJ)
            J=UPHE2(IJ)
            JJ=J+NLHE1
            II=I+NLHE1
            IT=ITRHE2(I,J)
            FRQO=FRQHE2(I)-FRQHE2(J)
            DOP=SRT2*FRQO*DOPCOF*.5
            X=DELQUAD*MOD(IJ-ITPTRHE2,NQUAD)*SRT
            SIGMA=PIE2MC*OSCHE2(IT)*EXP(-X*X)/DOP/1.7724539
            X=S*SIGMA*(NHE2(I,ID)-GHE2(I)*NHE2(J,ID)/GHE2(J))
            BN(II)=-X
            BN(JJ)=+X
         ENDIF
         IB=BLOCK(IJ)
         DO 160 IL=1,NEQN
            DO 150 J=1,NEQN
               DD(IL,IB)=DD(IL,IB)+AN(IL,J)*BN(J)*RAD(IJ,ID)
150         CONTINUE
160      CONTINUE
170   CONTINUE
C
C     TEMPERATURE PERTURBATIONS
C
180   DO 190 I=1,NLHE1
         BN(I)=NHE1S(I,ID)*(DRHE1(NLHE1+1,I)-
     :         ((1.5+HKT*FRQHE1(I))/TEMP(ID))*
```

```
     :          (RHE1(NLHE1+1,I)+NE(ID)*CRHE1(I,NLHE1+1)))+
     :          NE(ID)*DCRHE1(I,NLHE1+1)*(NHE1S(I,ID)-NHE1(I,ID))
190   CONTINUE
      DO 230 I=1,NLHE1
         DO 200 J=1,NLHE1
            BN(I)=BN(I)+NE(ID)*
     :            (DCRHE1(J,I)*NHE1(J,ID)-DCRHE1(I,J)*NHE1(I,ID))
200      CONTINUE
         DO 210 J=1,I-1
            FRQ0=FRQHE1(J)-FRQHE1(I)
            X=GHE1(J)*EXP(HKT*FRQ0)/GHE1(I)
            X0=-X*HKTT*FRQ0
            BN(I)=BN(I)-NHE1(I,ID)*(X*DRHE1(I,J)+RHE1(I,J)*X0)+
     :            NHE1(J,ID)*DRHE1(J,I)
210      CONTINUE
         DO 220 J=I+1,NLHE1
            FRQ0=FRQHE1(I)-FRQHE1(J)
            X=GHE1(I)*EXP(HKT*FRQ0)/GHE1(J)
            X0=-X*HKTT*FRQ0
            BN(I)=BN(I)-NHE1(I,ID)*DRHE1(I,J)+
     :            NHE1(J,ID)*(X*DRHE1(J,I)+X0*RHE1(J,I))
220      CONTINUE
230   CONTINUE
      DO 240 I=1,NLHE2
         II=I+NLHE1
         BN(II)=NHE2S(I,ID)*(DRHE2(NLHE2+1,I)-
     :          ((1.5+HKT*FRQHE2(I))/TEMP(ID))*
     :          (RHE2(NLHE2+1,I)+NE(ID)*CRHE2(I,NLHE2+1)))+
     :          NE(ID)*DCRHE2(I,NLHE2+1)*(NHE2S(I,ID)-NHE2(I,ID))
240   CONTINUE
      DO 280 I=1,NLHE2
         II=I+NLHE1
         DO 250 J=1,NLHE2
            BN(II)=BN(II)+NE(ID)*
     :            (DCRHE2(J,I)*NHE2(J,ID)-DCRHE2(I,J)*NHE2(I,ID))
250      CONTINUE
         DO 260 J=1,I-1
            FRQ0=FRQHE2(J)-FRQHE2(I)
            X=GHE2(J)*EXP(HKT*FRQ0)/GHE2(I)
            X0=-X*HKTT*FRQ0
            BN(II)=BN(II)-NHE2(I,ID)*(X*DRHE2(I,J)+X0*RHE2(I,J))+
     :            NHE2(J,ID)*DRHE2(J,I)
260      CONTINUE
         DO 270 J=I+1,NLHE2
            FRQ0=FRQHE2(I)-FRQHE2(J)
            X=GHE2(I)*EXP(HKT*FRQ0)/GHE2(J)
            X0=-X*HKTT*FRQ0
            BN(II)=BN(II)-NHE2(I,ID)*DRHE2(I,J)+
     :            NHE2(J,ID)*(X*DRHE2(J,I)+X0*RHE2(J,I))
270      CONTINUE
280   CONTINUE
      BN(NLHE1+NLHE2+1)=NE(ID)*(Y*DSHDT(ID)*NPROT(ID)-NHE2(1,ID)*
     :            DSHEDT(1,ID)-NHE3(ID)*DSHEDT(2,ID))
      DO 290 I=1,NLH
         II=I+NLHE1+NLHE2+1
         BN(II)=NS(I,ID)*(DRH(NLH+1,I)-((1.5+HKT*FRQH(I))/TEMP(ID))*
     :          (RH(NLH+1,I)+NE(ID)*CR(I,NLH+1)))+
     :          NE(ID)*DCRDT(I,NLH+1)*(NS(I,ID)-N(I,ID))
290   CONTINUE
      DO 330 I=1,NLH
         II=I+NLHE1+NLHE2+1
         DO 300 J=1,NLH
            BN(II)=BN(II)+
     :            NE(ID)*(DCRDT(J,I)*N(J,ID)-DCRDT(I,J)*N(I,ID))
300      CONTINUE
         DO 310 J=1,I-1
            FRQ0=FRQH(J)-FRQH(I)
            X=GH(J)*EXP(HKT*FRQ0)/GH(I)
            X0=-X*HKTT*FRQ0
            BN(II)=BN(II)-N(I,ID)*(X*DRH(I,J)+X0*RH(I,J))+
     :            N(J,ID)*DRH(J,I)
310      CONTINUE
         DO 320 J=I+1,NLH
            FRQ0=FRQH(I)-FRQH(J)
            X=GH(I)*EXP(HKT*FRQ0)/GH(J)
            X0=-X*HKTT*FRQ0
            BN(II)=BN(II)-N(I,ID)*DRH(I,J)+
     :            N(J,ID)*(X*DRH(J,I)+X0*RH(J,I))
320      CONTINUE
330   CONTINUE
```

```
      BN(NEQN)=-NE(ID)*NHE3(ID)*DSHEDT(2,ID)
      DO 350 IL=1,NEQN
         DO 340 J=1,NEQN
            DD(IL,NNT)=DD(IL,NNT)+AN(IL,J)*BN(J)*TEMP(ID)
340      CONTINUE
350   CONTINUE
C
C     ELECTRON DENSITY DERIVATIVES.
C
      DO 360 I=1,NLHE1
         BN(I)=(NHE1S(I,ID)/NE(ID))*
     :         (RHE1(NLHE1+1,I)+2.0*NE(ID)*CRHE1(I,NLHE1+1))-
     :         NHE1(I,ID)*CRHE1(I,NLHE1+1)
360   CONTINUE
      DO 380 I=1,NLHE1
         DO 370 J=1,NLHE1
            BN(I)=BN(I)+CRHE1(J,I)*NHE1(J,ID)-CRHE1(I,J)*NHE1(I,ID)
370      CONTINUE
380   CONTINUE
      DO 390 I=1,NLHE2
         II=I+NLHE1
         BN(II)=(NHE2S(I,ID)/NE(ID))*
     :         (RHE2(NLHE2+1,I)+2.0*NE(ID)*CRHE2(I,NLHE2+1))-
     :          NHE2(I,ID)*CRHE2(I,NLHE2+1)
390   CONTINUE
      DO 410 I=1,NLHE2
         II=I+NLHE1
         DO 400 J=1,NLHE2
            BN(II)=BN(II)+CRHE2(J,I)*NHE2(J,ID)-CRHE2(I,J)*NHE2(I,ID)
400      CONTINUE
410   CONTINUE
      BN(NLHE1+NLHE2+1)=Y*NPROT(ID)*SUMH(ID)-NHE2(1,ID)*SUMHE(1,ID)-
     :         NHE3(ID)*SUMHE(2,ID)
      DO 420 I=1,NLH
         II=I+NLHE1+NLHE2+1
         BN(II)=(NS(I,ID)/NE(ID))*(RH(NLH+1,I)+2.0*NE(ID)*CR(I,NLH+1))-
     :          N(I,ID)*CR(I,NLH+1)
420   CONTINUE
      DO 440 I=1,NLH
         II=I+NLHE1+NLHE2+1
         DO 430 J=1,NLH
            BN(II)=BN(II)+CR(J,I)*N(J,ID)-CR(I,J)*N(I,ID)
430      CONTINUE
440   CONTINUE
      BN(NEQN)=1.0-NHE3(ID)*SUMHE(2,ID)
      DO 460 IL=1,NEQN
         DO 450 J=1,NEQN
            DD(IL,NNE)=DD(IL,NNE)+AN(IL,J)*BN(J)*NE(ID)
450      CONTINUE
460   CONTINUE
C
C     NTOT AND DERIVATIVES THEREOF
C
      PR(NNN+1)=NE(ID)
      PR(NNE)=NE(ID)
      DO 470 IL=1,NLHE1
         PR(NNN+1)=PR(NNN+1)+NHE1(IL,ID)
470   CONTINUE
      DO 490 IL=1,NLHE1
         DO 480 J=1,NNN
            PR(J)=PR(J)+DD(IL,J)
480      CONTINUE
490   CONTINUE
      DO 500 IL=1,NLHE2
         PR(NNN+1)=PR(NNN+1)+NHE2(IL,ID)
500   CONTINUE
      DO 520 IL=1,NLHE2
         DO 510 J=1,NNN
            PR(J)=PR(J)+DD(IL+NLHE1,J)
510      CONTINUE
520   CONTINUE
      PR(NNN+1)=PR(NNN+1)+NHE2(1,ID)*SUMHE(1,ID)*NE(ID)
      PR(NNT)=PR(NNT)+NHE2(1,ID)*DSHEDT(1,ID)*NE(ID)*TEMP(ID)
      PR(NNE)=PR(NNE)+NHE2(1,ID)*SUMHE(1,ID)*NE(ID)
      DO 530 J=1,NNN
         PR(J)=PR(J)+DD(NLHE1+1,J)*SUMHE(1,ID)*NE(ID)
530   CONTINUE
      PR(NNN+1)=PR(NNN+1)+NHE3(ID)*(1.0+SUMHE(2,ID)*NE(ID))
      PR(NNT)=PR(NNT)+NHE3(ID)*DSHEDT(2,ID)*NE(ID)*TEMP(ID)
      PR(NNE)=PR(NNE)+NHE3(ID)*SUMHE(2,ID)*NE(ID)
```

```
      DO 540 J=1,NNN
         PR(J)=PR(J)+DD(NLHE1+NLHE2+1,J)*(1.0+SUMHE(2,ID)*NE(ID))
540   CONTINUE
      DO 550 IL=1,NLH
         PR(NNN+1)=PR(NNN+1)+N(IL,ID)
550   CONTINUE
      DO 570 IL=1,NLH
         DO 560 J=1,NNN
            PR(J)=PR(J)+DD(IL+NLHE1+NLHE2+1,J)
560      CONTINUE
570   CONTINUE
      PR(NNN+1)=PR(NNN+1)+NPROT(ID)*(1.0+SUMH(ID)*NE(ID))
      PR(NNT)=PR(NNT)+NPROT(ID)*DSHDT(ID)*NE(ID)*TEMP(ID)
      PR(NNE)=PR(NNE)+NPROT(ID)*SUMH(ID)*NE(ID)
      DO 580 J=1,NNN
         PR(J)=PR(J)+DD(NLHE1+NLHE2+NLH+2,J)*(1.0+SUMH(ID)*NE(ID))
580   CONTINUE
      NTOT(ID)=PR(NNN+1)
C
C     ELECTRON SCATTERING
C
      NO=NNN+1
      DO 600 J=1,NNN
         DO 590 IJ=1,NJ
            CHI(IJ,J)=0.0
            ETA(IJ,J)=0.0
590      CONTINUE
600   CONTINUE
      DO 610 IJ=1,NJ
         CHI(IJ,NO)=NE(ID)*SIGE
         CHI(IJ,NNE)=CHI(IJ,NO)
         ETA(IJ,NO)=0.
610   CONTINUE
C
C     HELIUM BOUND-FREE OPACITIES
C
      DO 650 IL=1,NLHE1
         DO 620 IJ=1,NJ
            E(IJ)=SIGHE1(IL,IJ)*NHE1S(IL,ID)*EX(IJ)
            ETA(IJ,NO)=ETA(IJ,NO)+E(IJ)
            CHI(IJ,NO)=CHI(IJ,NO)+SIGHE1(IL,IJ)*NHE1(IL,ID)
            ETA(IJ,NNT)=ETA(IJ,NNT)+E(IJ)*(-1.5+HKT*(FREQ(IJ)-
     :                  FRQHE1(IL)))
            ETA(IJ,NNE)=ETA(IJ,NNE)+E(IJ)
620      CONTINUE
         DO 640 J=1,NNN
            DO 630 IJ=1,NJ
               ETA(IJ,J)=ETA(IJ,J)+E(IJ)*DD(NLHE1+1,J)/NHE2(1,ID)
               CHI(IJ,J)=CHI(IJ,J)+SIGHE1(IL,IJ)*DD(IL,J)
630         CONTINUE
640      CONTINUE
650   CONTINUE
      DO 690 IL=NLHE1+1,NLHE1S
         DO 660 IJ=1,NJ
            C(IJ)=SIGHE1(IL,IJ)*NHE1S(IL,ID)
            E(IJ)=C(IJ)*EX(IJ)
            ETA(IJ,NO)=ETA(IJ,NO)+E(IJ)
            CHI(IJ,NO)=CHI(IJ,NO)+C(IJ)
            ETA(IJ,NNT)=ETA(IJ,NNT)+E(IJ)*(-1.5+HKT*(FREQ(IJ)-
     :                  FRQHE1(IL)))
            CHI(IJ,NNT)=CHI(IJ,NNT)+C(IJ)*(-1.5-HKT*FRQHE1(IL))
            ETA(IJ,NNE)=ETA(IJ,NNE)+E(IJ)
            CHI(IJ,NNE)=CHI(IJ,NNE)+C(IJ)
660      CONTINUE
         DO 680 J=1,NNN
            DO 670 IJ=1,NJ
               ETA(IJ,J)=ETA(IJ,J)+E(IJ)*DD(NLHE1+1,J)/NHE2(1,ID)
               CHI(IJ,J)=CHI(IJ,J)+C(IJ)*DD(NLHE1+1,J)/NHE2(1,ID)
670         CONTINUE
680      CONTINUE
690   CONTINUE
      DO 730 IL=1,NLHE2
         DO 700 IJ=1,NJ
            E(IJ)=SIGHE2(IL,IJ)*NHE2S(IL,ID)*EX(IJ)
            ETA(IJ,NO)=ETA(IJ,NO)+E(IJ)
            CHI(IJ,NO)=CHI(IJ,NO)+SIGHE2(IL,IJ)*NHE2(IL,ID)
            ETA(IJ,NNT)=ETA(IJ,NNT)+E(IJ)*(-1.5+HKT*(FREQ(IJ)-
     :                  FRQHE2(IL)))
            ETA(IJ,NNE)=ETA(IJ,NNE)+E(IJ)
700      CONTINUE
```

```
      DO 720 J=1,NNN
         DO 710 IJ=1,NJ
            ETA(IJ,J)=ETA(IJ,J)+E(IJ)*DD(NLHE1+NLHE2+1,J)/NHE3(ID)
            CHI(IJ,J)=CHI(IJ,J)+SIGHE2(IL,IJ)*DD(NLHE1+IL,J)
710      CONTINUE
720   CONTINUE
730 CONTINUE
    DO 770 IL=NLHE2+1,NLHE2S
       DO 740 IJ=1,NJ
          C(IJ)=SIGHE2(IL,IJ)*NHE2S(IL,ID)
          E(IJ)=C(IJ)*EX(IJ)
          ETA(IJ,NO)=ETA(IJ,NO)+E(IJ)
          CHI(IJ,NO)=CHI(IJ,NO)+C(IJ)
          ETA(IJ,NNT)=ETA(IJ,NNT)+E(IJ)*(-1.5+HKT*(FREQ(IJ)-
   :                 FRQHE2(IL)))
          CHI(IJ,NNT)=CHI(IJ,NNT)+C(IJ)*(-1.5-HKT*FRQHE2(IL))
          ETA(IJ,NNE)=ETA(IJ,NNE)+E(IJ)
          CHI(IJ,NNE)=CHI(IJ,NNE)+C(IJ)
740    CONTINUE
       DO 760 J=1,NNN
          DO 750 IJ=1,NJ
             CHI(IJ,J)=CHI(IJ,J)+C(IJ)*DD(NLHE1+NLHE2+1,J)/NHE3(ID)
             ETA(IJ,J)=ETA(IJ,J)+E(IJ)*DD(NLHE1+NLHE2+1,J)/NHE3(ID)
750       CONTINUE
760    CONTINUE
770 CONTINUE
C
C   FREE-FREE OPACITIES
C
    X0=NHE3(ID)*NE(ID)**2
    X1=X0*SUMHE(2,ID)
    DO 780 IL=1,NLHE2
       X1=X1+NE(ID)*NHE2(IL,ID)
780 CONTINUE
    DO 820 J=1,NNN
       DO 790 IJ=1,NJ
          DC=NE(ID)**2*SUMHE(2,ID)*SKK(IJ,2)*DD(NLHE1+NLHE2+1,J)
          CHI(IJ,J)=CHI(IJ,J)+DC
          ETA(IJ,J)=ETA(IJ,J)+DC*EX(IJ)
790    CONTINUE
       DO 810 IL=1,NLHE2
          DO 800 IJ=1,NJ
             DC=NE(ID)*SKK(IJ,2)*DD(NLHE1+IL,J)
             CHI(IJ,J)=CHI(IJ,J)+DC
             ETA(IJ,J)=ETA(IJ,J)+DC*EX(IJ)
800       CONTINUE
810    CONTINUE
820 CONTINUE
    DO 830 IJ=1,NJ
       C(IJ)=X1*SKK(IJ,2)
       E(IJ)=C(IJ)*EX(IJ)
       ETA(IJ,NO)=ETA(IJ,NO)+E(IJ)
       CHI(IJ,NO)=CHI(IJ,NO)+C(IJ)
       DC=X0*DSHEDT(2,ID)*SKK(IJ,2)+X1*DSKKDT(IJ,2)
       DE=(DC+X1*SKK(IJ,2)*HKT*FREQ(IJ)*T1)*EX(IJ)
       ETA(IJ,NNT)=ETA(IJ,NNT)+DE*TEMP(ID)
       CHI(IJ,NNT)=CHI(IJ,NNT)+DC*TEMP(ID)
830 CONTINUE
    X1=2.*NE(ID)*NHE3(ID)*SUMHE(2,ID)
    DO 840 IL=1,NLHE2
       X1=X1+NHE2(IL,ID)
840 CONTINUE
    DO 850 IJ=1,NJ
       DC=X1*SKK(IJ,2)*NE(ID)
       ETA(IJ,NNE)=ETA(IJ,NNE)+DC*EX(IJ)
       CHI(IJ,NNE)=CHI(IJ,NNE)+DC
       C(IJ)=NHE3(ID)*NE(ID)*SKK(IJ,3)
       E(IJ)=C(IJ)*EX(IJ)
       ETA(IJ,NO)=ETA(IJ,NO)+E(IJ)
       CHI(IJ,NO)=CHI(IJ,NO)+C(IJ)
       ETA(IJ,NNT)=ETA(IJ,NNT)+E(IJ)*HKT*FREQ(IJ)+NHE3(ID)*NE(ID)*
   :              DSKKDT(IJ,3)*EX(IJ)*TEMP(ID)
       CHI(IJ,NNT)=CHI(IJ,NNT)+NHE3(ID)*NE(ID)*DSKKDT(IJ,3)*TEMP(ID)
       ETA(IJ,NNE)=ETA(IJ,NNE)+E(IJ)
       CHI(IJ,NNE)=CHI(IJ,NNE)+C(IJ)
       C(IJ)=NPROT(ID)*NE(ID)*SKK(IJ,1)
       E(IJ)=C(IJ)*EX(IJ)
       ETA(IJ,NO)=ETA(IJ,NO)+E(IJ)
       CHI(IJ,NO)=CHI(IJ,NO)+C(IJ)
       ETA(IJ,NNT)=ETA(IJ,NNT)+E(IJ)*HKT*FREQ(IJ)+NPROT(ID)*NE(ID)*
```

```
     :              DSKKDT(IJ,1)*EX(IJ)*TEMP(ID)
          CHI(IJ,NNT)=CHI(IJ,NNT)+NPROT(ID)*NE(ID)*DSKKDT(IJ,1)*TEMP(ID)
          ETA(IJ,NNE)=ETA(IJ,NNE)+E(IJ)
          CHI(IJ,NNE)=CHI(IJ,NNE)+C(IJ)
 850   CONTINUE
       DO 870 J=1,NNN
          DO 860 IJ=1,NJ
             DC=SKK(IJ,3)*DD(NLHE1+NLHE2+1,J)*NE(ID)
             CHI(IJ,J)=CHI(IJ,J)+DC
             ETA(IJ,J)=ETA(IJ,J)+DC*EX(IJ)
             DC=SKK(IJ,1)*DD(NLHE1+NLHE2+NLH+2,J)*NE(ID)
             CHI(IJ,J)=CHI(IJ,J)+DC
             ETA(IJ,J)=ETA(IJ,J)+DC*EX(IJ)
 860      CONTINUE
 870   CONTINUE
C
C      HYDROGEN BOUND-FREE OPACITIES
C
       DO 910 IL=1,NLH
          DO 880 IJ=1,NJ
             E(IJ)=SIG(IL,IJ)*NS(IL,ID)*EX(IJ)
             ETA(IJ,NO)=ETA(IJ,NO)+E(IJ)
             CHI(IJ,NO)=CHI(IJ,NO)+SIG(IL,IJ)*N(IL,ID)
             ETA(IJ,NNT)=ETA(IJ,NNT)+E(IJ)*(-1.5+HKT*(FREQ(IJ)-
     :               FRQH(IL)))
             ETA(IJ,NNE)=ETA(IJ,NNE)+E(IJ)
 880      CONTINUE
          DO 900 J=1,NNN
             DO 890 IJ=1,NJ
                ETA(IJ,J)=ETA(IJ,J)+E(IJ)*DD(NLHE1+NLHE2+NLH+2,J)/
     :                 NPROT(ID)
                CHI(IJ,J)=CHI(IJ,J)+SIG(IL,IJ)*DD(NLHE1+NLHE2+1+IL,J)
 890         CONTINUE
 900      CONTINUE
 910   CONTINUE
       DO 950 IL=NLH+1,NLHS
          DO 920 IJ=1,NJ
             C(IJ)=SIG(IL,IJ)*NS(IL,ID)
             E(IJ)=C(IJ)*EX(IJ)
             ETA(IJ,NO)=ETA(IJ,NO)+E(IJ)
             CHI(IJ,NO)=CHI(IJ,NO)+C(IJ)
             ETA(IJ,NNT)=ETA(IJ,NNT)+E(IJ)*(-1.5+HKT*(FREQ(IJ)-
     :               FRQH(IL)))
             CHI(IJ,NNT)=CHI(IJ,NNT)+C(IJ)*(-1.5-HKT*FRQH(IL))
             ETA(IJ,NNE)=ETA(IJ,NNE)+E(IJ)
             CHI(IJ,NNE)=CHI(IJ,NNE)+C(IJ)
 920      CONTINUE
          DO 940 J=1,NNN
             DO 930 IJ=1,NJ
                ETA(IJ,J)=ETA(IJ,J)+E(IJ)*DD(NLHE1+NLHE2+NLH+2,J)/
     :                 NPROT(ID)
                CHI(IJ,J)=CHI(IJ,J)+C(IJ)*DD(NLHE1+NLHE2+NLH+2,J)/
     :                 NPROT(ID)
 930         CONTINUE
 940      CONTINUE
 950   CONTINUE
C
C      LINE OPACITIES
C
       DO 960 IJ=ITPTRH,ITPTRHE1-1
          L=LOWH(IJ)
          U=UPH(IJ)
          IT=ITRH(L,U)
          FRQ0=FRQH(L)-FRQH(U)
          DOP=SRT2*FRQ0*DOPCOF
          X=DELQUAD*MOD(IJ-ITPTRH,NQUAD)*SRT
          SIGMA=PIE2MC*OSCH(IT)*EXP(-X*X)/DOP/1.7724539
          C(IJ)=SIGMA
          SIGMAT=SIGMA*(X*X-0.5)
          ETA(IJ,NO)=ETA(IJ,NO)+GH(L)*SIGMA*N(U,ID)/GH(U)
          CHI(IJ,NO)=CHI(IJ,NO)+SIGMA*N(L,ID)
          ETA(IJ,NNT)=ETA(IJ,NNT)+GH(L)*SIGMAT*N(U,ID)/GH(U)
          CHI(IJ,NNT)=CHI(IJ,NNT)+SIGMAT*N(L,ID)
 960   CONTINUE
       DO 980 IJ=ITPTRH,ITPTRHE1-1
          L=LOWH(IJ)
          U=UPH(IJ)
          DO 970 J=1,NNN
             ETA(IJ,J)=ETA(IJ,J)+GH(L)*C(IJ)*DD(NLHE1+NLHE2+1+U,J)/
     :               GH(U)
```

```
            CHI(IJ,J)=CHI(IJ,J)+C(IJ)*DD(NLHE1+NLHE2+1+L,J)
970      CONTINUE
980   CONTINUE
      DO 990 IJ=ITPTRHE1,ITPTRHE2-1
         L=LOWHE1(IJ)
         U=UPHE1(IJ)
         IT=ITRHE1(L,U)
         FRQ0=FRQHE1(L)-FRQHE1(U)
         DOP=SRT2*FRQ0*DOPCOF*0.5
         X=DELQUAD*MOD(IJ-ITPTRHE1,NQUAD)*SRT
         SIGMA=PIE2MC*OSCHE1(IT)*EXP(-X*X)/DOP/1.7724539
         C(IJ)=SIGMA
         SIGMAT=SIGMA*(X*X-0.5)
         ETA(IJ,NO)=ETA(IJ,NO)+GHE1(L)*SIGMA*NHE1(U,ID)/GHE1(U)
         CHI(IJ,NO)=CHI(IJ,NO)+SIGMA*NHE1(L,ID)
         ETA(IJ,NNT)=ETA(IJ,NNT)+GHE1(L)*SIGMAT*NHE1(U,ID)/GHE1(U)
         CHI(IJ,NNT)=CHI(IJ,NNT)+SIGMAT*NHE1(L,ID)
990   CONTINUE
      DO 1010 IJ=ITPTRHE1,ITPTRHE2-1
         L=LOWHE1(IJ)
         U=UPHE1(IJ)
         DO 1000 J=1,NNN
            ETA(IJ,J)=ETA(IJ,J)+GHE1(L)*C(IJ)*DD(U,J)/GHE1(U)
            CHI(IJ,J)=CHI(IJ,J)+C(IJ)*DD(L,J)
1000     CONTINUE
1010  CONTINUE
      DO 1020 IJ=ITPTRHE2,NJ
         L=LOWHE2(IJ)
         U=UPHE2(IJ)
         IT=ITRHE2(L,U)
         FRQ0=FRQHE2(L)-FRQHE2(U)
         DOP=SRT2*FRQ0*DOPCOF*0.5
         X=DELQUAD*MOD(IJ-ITPTRHE2,NQUAD)*SRT
         SIGMA=PIE2MC*OSCHE2(IT)*EXP(-X*X)/DOP/1.7724539
         C(IJ)=SIGMA
         SIGMAT=SIGMA*(X*X-0.5)
         ETA(IJ,NO)=ETA(IJ,NO)+GHE2(L)*SIGMA*NHE2(U,ID)/GHE2(U)
         CHI(IJ,NO)=CHI(IJ,NO)+SIGMA*NHE2(L,ID)
         ETA(IJ,NNT)=ETA(IJ,NNT)+GHE2(L)*SIGMAT*NHE2(U,ID)/GHE2(U)
         CHI(IJ,NNT)=CHI(IJ,NNT)+SIGMAT*NHE2(L,ID)
1020  CONTINUE
      DO 1040 IJ=ITPTRHE2,NJ
         L=LOWHE2(IJ)
         U=UPHE2(IJ)
         DO 1030 J=1,NNN
            ETA(IJ,J)=ETA(IJ,J)+GHE2(L)*C(IJ)*DD(NLHE1+U,J)/GHE2(U)
            CHI(IJ,J)=CHI(IJ,J)+C(IJ)*DD(NLHE1+L,J)
1030     CONTINUE
1040  CONTINUE
      DO 1050 IJ=1,NJ
         EX(IJ)=BBCOF*FREQ(IJ)**3
1050  CONTINUE
      DO 1070 J=1,NNN+1
         DO 1060 IJ=1,NJ
            CHI(IJ,J)=CHI(IJ,J)-ETA(IJ,J)
            ETA(IJ,J)=EX(IJ)*ETA(IJ,J)
1060     CONTINUE
1070  CONTINUE
C
C     CONVERT OPACITIES TO MASS COEFFICIENTS
C
      DO 1090 J=1,NNN
         DO 1080 IJ=1,NJ
            CHI(IJ,J)=(CHI(IJ,J)-CHI(IJ,NNN+1)*MU1*PR(J)/NM(ID))/
     :                NM(ID)/MHYD
            ETA(IJ,J)=(ETA(IJ,J)-ETA(IJ,NNN+1)*MU1*PR(J)/NM(ID))/
     :                NM(ID)/MHYD
1080     CONTINUE
1090  CONTINUE
      DO 1100 IJ=1,NJ
         CHI(IJ,NNE)=CHI(IJ,NNE)+CHI(IJ,NNN+1)*MU1*NE(ID)/NM(ID)/
     :               NM(ID)/MHYD
         ETA(IJ,NNE)=ETA(IJ,NNE)+ETA(IJ,NNN+1)*MU1*NE(ID)/NM(ID)/
     :               NM(ID)/MHYD
         CHI(IJ,NNN+1)=CHI(IJ,NNN+1)/NM(ID)/MHYD
         ETA(IJ,NNN+1)=ETA(IJ,NNN+1)/NM(ID)/MHYD
1100  CONTINUE
      RETURN
      END

      SUBROUTINE GAB
```

```
C
C     BE VERBOSE
C
      IMPLICIT NONE
C
      COMA
      COMAI
      COMC
C
      INTEGER I, ID, IJ, IL, J, NO, N1
      REAL DIS, SUM
C
C     PRINT BOUND-FREE JUMPS
C
      WRITE (6,1001)
      WRITE (6,1002)
      WRITE (6,1003)
      IL=1
      DO 10 IJ=2,ITPTRH-1
         IF (SIG(IL,IJ).EQ.0)THEN
            DIS=-2.5*LOG10(FH(IJ-1)*RAD(IJ-1,1)/FH(IJ)/RAD(IJ,1))
            WRITE (6,1004)IL,DIS
            IL=IL+1
         ENDIF
         IF (IL.GT.NLH)GO TO 20
10    CONTINUE
20    WRITE (6,1005)
      IL=1
      DO 30 IJ=2,ITPTRH-1
         IF (SIGHE1(IL,IJ).EQ.0)THEN
            DIS=-2.5*LOG10(FH(IJ-1)*RAD(IJ-1,1)/FH(IJ)/RAD(IJ,1))
            WRITE (6,1004)IL,DIS
            IL=IL+1
         ENDIF
         IF (IL.GT.NLHE1)GO TO 40
30    CONTINUE
40    WRITE (6,1006)
      IL=1
      DO 50 IJ=2,ITPTRH-1
         IF (SIGHE2(IL,IJ).EQ.0)THEN
            DIS=-2.5*LOG10(FH(IJ-1)*RAD(IJ-1,1)/FH(IJ)/RAD(IJ,1))
            WRITE (6,1004)IL,DIS
            IL=IL+1
         ENDIF
         IF (IL.GT.NLHE2)GO TO 60
50    CONTINUE
C
C     PRINT TOTAL FLUX
C
      SUM=0.
60    DO 70 I=1,NJ
         SUM=SUM+WT(I)*FH(I)*RAD(I,1)
70    CONTINUE
      WRITE (6,1007)SUM
C
C     PRINT BASIC MODEL PARAMETERS AND POPULATIONS
C
      WRITE (6,1001)
      WRITE (6,1008)
      WRITE (6,1009)(M(I),TEMP(I),NTOT(I),NE(I),NPROT(I),I=1,NDEPTH)
C
C     PRINT POPULATIONS
C
      WRITE (6,1010)
      NO=1
      N1=MIN(NLH,3)
80    WRITE (6,1011) (I,I,I=NO,N1)
      DO 90 ID=1,NDEPTH
         WRITE (6,1015)M(ID),(N(I,ID),N(I,ID)/NS(I,ID),I=NO,N1)
90    CONTINUE
      IF (N1.LT.NLH)THEN
         NO=N1+1
         N1=MIN(NLH,N1+3)
         GO TO 80
      ENDIF
      WRITE (6,1012)
      NO=1
      N1=MIN(NLHE1,3)
100   WRITE (6,1011) (I,I,I=NO,N1)
      DO 110 ID=1,NDEPTH
```

```
         WRITE (6,1015)M(ID),(NHE1(I,ID),NHE1(I,ID)/NHE1S(I,ID),
     :                 I=NO,N1)
110   CONTINUE
      IF (N1.LT.NLHE1)THEN
         NO=N1+1
         N1=MIN(NLHE1,N1+3)
         GO TO 100
      ENDIF
      WRITE (6,1013)
      NO=1
      N1=MIN(NLHE2,3)
120   WRITE (6,1011) (I,I,I=NO,N1)
      DO 130 ID=1,NDEPTH
         WRITE (6,1015)M(ID),(NHE2(I,ID),NHE2(I,ID)/NHE2S(I,ID),
     :           I=NO,N1)
130   CONTINUE
      IF (N1.LT.NLHE2)THEN
         NO=N1+1
         N1=MIN(N1+3,NLHE2)
         GO TO 120
      ENDIF
      WRITE  (6,1014)
      DO 140 ID=1,NDEPTH
         WRITE (6,1015)M(ID),NHE3(ID)
140   CONTINUE
C
C     PRINT RATES AS A DIAGNOSTIC
C
      IF (.NOT.FPRINT)RETURN
      WRITE (6,1001)
      WRITE (6,1016)
      WRITE (6,1017)
      DO 150 I=1,NLH
         WRITE (6,1018)(CR(I,J),J=1,NLH+1)
150   CONTINUE
      WRITE (6,1019)
      DO 160 I=1,NLHE1
         WRITE (6,1018)(CRHE1(I,J),J=1,NLHE1+1)
160   CONTINUE
      WRITE (6,1020)
      DO 170 I=1,NLHE2
         WRITE (6,1018)(CRHE2(I,J),J=1,NLHE2+1)
170   CONTINUE
      WRITE (6,1001)
      WRITE (6,1021)
      WRITE (6,1017)
      DO 180 I=1,NLH+1
         WRITE (6,1018)(RH(I,J),J=1,NLH+1)
180   CONTINUE
      WRITE (6,1019)
      DO 190 I=1,NLHE1+1
         WRITE (6,1018)(RHE1(I,J),J=1,NLHE1+1)
190   CONTINUE
      WRITE (6,1020)
      DO 200 I=1,NLHE2+1
         WRITE (6,1018)(RHE2(I,J),J=1,NLHE2+1)
200   CONTINUE
      WRITE (6,1001)
      RETURN
C
1001  FORMAT (1H1)
1002  FORMAT (' BOUND-FREE JUMPS:')
1003  FORMAT (' HYDROGEN')
1004  FORMAT (' ',I2,': ',F6.2)
1005  FORMAT (' HELIUM I')
1006  FORMAT (' HELIUM II')
1007  FORMAT (' TOTAL SURFACE FLUX = ',E14.6)
1008  FORMAT (4X,'MASS',5X,'TEMPERATURE',6X,'TOTAL N',11X,'NE',13X,'NP')
1009  FORMAT (E11.3,4E15.8)
1010  FORMAT ('1',55X,'HYDROGEN POPULATIONS')
1011  FORMAT (6X,'MASS',3(6X,'N(',I2,')',5X,'NS(',I2,')'))
1012  FORMAT ('1',32X,'HELIUM I POPULATIONS')
1013  FORMAT ('1',32X,'HELIUM II POPULATIONS')
1014  FORMAT ('1',32X,'HELIUM III POPULATION'//6X,'MASS',6X,'N')
1015  FORMAT (E11.3,6E11.3)
1016  FORMAT (1X,'COLLISION RATES AT DEPTH')
1017  FORMAT (4X,'HYDROGEN')
1018  FORMAT (1X,7E11.4)
1019  FORMAT (4X,'NEUTRAL HELIUM')
1020  FORMAT (4X,'IONIZED HELIUM')
```

```
1021  FORMAT (1X,'RADIATIVE RATES AT DEPTH')
      END

      SUBROUTINE INDEXX(N,ARRIN,INDX)
C
C     FROM "NUMERICAL RECIPES"
C
C     CREATES AN INDEX ARRAY FOR A VECTOR ARRIN.
C
      REAL ARRIN(N)
      INTEGER INDX(N)
      DO 10 J=1,N
         INDX(J)=J
10    CONTINUE
      L=N/2+1
      IR=N
20    CONTINUE
         IF (L.GT.1)THEN
            L=L-1
            INDXT=INDX(L)
            Q=ARRIN(INDXT)
         ELSE
            INDXT=INDX(IR)
            Q=ARRIN(INDXT)
            INDX(IR)=INDX(1)
            IR=IR-1
            IF (IR.EQ.1)THEN
               INDX(1)=INDXT
               RETURN
            ENDIF
         ENDIF
         I=L
         J=L+L
30       IF (J.LE.IR)THEN
            IF (J.LT.IR)THEN
               IF (ARRIN(INDX(J)).LT.ARRIN(INDX(J+1)))J=J+1
            ENDIF
            IF (Q.LT.ARRIN(INDX(J)))THEN
               INDX(I)=INDX(J)
               I=J
               J=J+J
            ELSE
               J=IR+1
            ENDIF
         GO TO 30
         ENDIF
         INDX(I)=INDXT
      GO TO 20
      END

      SUBROUTINE LINEAR
C
C     MAIN LINEARIZATION ROUTINE
C
      IMPLICIT NONE
C
C     MACROS
C
      COMA
      COMF
C
C     LOCAL VARIABLES
C
      INTEGER I, IBASE, ID, J, K
      REAL D(MNNN,MNNN+1), NU(MNNN), NUO(MNNN)
      EQUIVALENCE (D(1,MNNN+1), NU(1))
C
      INTEGER IOSTATUS
      REAL MATGEN, MATINV, OUT, RDABS, WRABS, XTENDABS
      EXTERNAL IOSTATUS, MATGEN, MATINV, OUT, RDABS, WRABS, XTENDABS
C
      DO 130 ID=1,NDEPTH
C
C     GENERATE LEVEL MATRICES
C
         CALL MATGEN(ID)
         IF (ID.EQ.1)GO TO 50
         DO 40 I=1,NNN
            DO 10 K=1,NNN
               Q(I)=Q(I)+A(I,K)*NU(K)
```

```
10          CONTINUE
            DO 30 J=1,NNN
               DO 20 K=1,NNN
                  B(I,J)=B(I,J)-A(I,K)*D(K,J)
20             CONTINUE
30          CONTINUE
40       CONTINUE
50       CALL MATINV(B,NNN,MNNN)
         IBASE=IOSTATUS(8,I)
         DO 60 I=1,NNN
            NU(I)=0.0
60       CONTINUE
         DO 80 I=1,NNN
            DO 70 J=1,NNN
               NU(I)=NU(I)+B(I,J)*Q(J)
70          CONTINUE
80       CONTINUE
         IF (ID.EQ.NDEPTH)GO TO 130
         DO 120 J=1,NNN
            DO 90 I=1,NNN
               D(I,J)=0.
90          CONTINUE
            DO 110 I=1,NNN
               DO 100 K=1,NNN
                  D(I,J)=D(I,J)+B(I,K)*C(K,J)
100            CONTINUE
110         CONTINUE
120      CONTINUE
         IBASE=(ID-1)*MNNN*(MNNN+1)
         CALL WRABS(8,D,MNNN*(MNNN+1),IBASE)
130   CONTINUE
      DO 180 ID=NDEPTH,1,-1
         IBASE=IOSTATUS(8,I)
         IF (ID.EQ.NDEPTH)GO TO 160
         DO 150 I=1,NNN
            DO 140 J=1,NNN
               NU(I)=NU(I)+D(I,J)*NUO(J)
140         CONTINUE
150      CONTINUE
160      IBASE=IOSTATUS(9,I)
         DO 170 I=1,NNN
            NUO(I)=NU(I)
170      CONTINUE
         CALL WRABS(9,NUO,NNN,MNNN*(ID-1))
         IF (ID.EQ.1)GO TO 190
         IBASE=(ID-2)*MNNN*(MNNN+1)
         CALL RDABS(8,D,MNNN*(MNNN+1),IBASE)
180   CONTINUE
190   CALL OUT
      RETURN
      END

      SUBROUTINE LINIT(ITER)
C
C     CALCULATE VALUE FOR SWITCHING PARAMETER
C
      IMPLICIT NONE
C
      COMA
      COMC
C
      INTEGER I, ITER, J
      REAL FAC
C
      REAL COLRAT, NURATE, WTSET
      EXTERNAL COLRAT, NURATE, WTSET
C
C     KEEP LAMBDA=1.0 IF NO SWITCHING EMPLOYED
C
      IF ((.NOT.FSWITCH).AND.(ITER.GT.1))RETURN
C
C     INITIALIZE IF FIRST ITERATION
C
      IF (ITER.LE.1)THEN
C
C     SET LAMBDA=1.0 IF NO SWITCHING EMPLOYED.
C
         IF (.NOT.FSWITCH)THEN
            LAMC=1.0
            LAML=1.0
         ELSE
```

```
C
C     LAMBDA NOT SPECIFIED IN INPUT FILE
C
            LAML=1.0
            IF (LAMC.LE.0)THEN
C
C     FIND MAXIMUM OF R/C (NOT MIN C/R SINCE R MAY BE ZERO)
C
               CALL WTSET(1.)
               LAMC=1.
               CALL COLRAT(TEMP(1),CR,CRHE1,CRHE2)
               CALL NURATE(1)
               LAMC=0.
               DO 20 I=1,NLH
                  DO 10 J=I+1,NLH+1
                     LAMC=MAX(LAMC,RH(I,J)/CR(I,J))
10                CONTINUE
20             CONTINUE
               DO 40 I=1,NLHE1
                  DO 30 J=I+1,NLHE1+1
                     LAMC=MAX(LAMC,RHE1(I,J)/CRHE1(I,J))
30                CONTINUE
40             CONTINUE
               DO 60 I=1,NLHE2
                  DO 50 J=I+1,NLHE2+1
                     LAMC=MAX(LAMC,RHE2(I,J)/CRHE2(I,J))
50                CONTINUE
60             CONTINUE
               LAMC=NE(1)/LAMC
            ENDIF
            IF (RULE(2))THEN
               LAML=LAMC
               LAMC=1.
            ENDIF
         ENDIF
      ELSE
C
C     SUBSEQUENT ITERATIONS WHEN EMPLOYING SWITCHING
C
         FAC=10.
         IF (LINERR.GT.(.02))FAC=2.
         IF (LINERR.GT.(.10))FAC=1.
         LAMC=FAC*LAMC
         IF (LAMC.GT.(.1))LAMC=1.0
         LAML=FAC*LAML
         IF (LAML.GT.(.1))LAML=1.0
      ENDIF
      CALL WTSET(LAML)
      RETURN
      END

      SUBROUTINE MATGEN(IDD)
C
C     GENERATE LINEARIZATION MATRICES
C
      IMPLICIT NONE
C
C     MACROS
C
      COMA
      COMAI
      COMC
      COMF
C
C     LOCAL VARIABLES
C
      INTEGER I, IB, ID, IDD, IJ, J
      REAL BETA(MNB), BETO(MNB,MNNN), BETP(MNB,MNNN)
      REAL CHIM(MNJ,MNNN+1), CHIO(MNJ,MNNN+1), ETAO(MNJ,MNNN+1)
      REAL CHIP(MNJ,MNDEPTH), ETAP(MNJ,MNDEPTH)
      REAL DEL(MNB), DELO(MNB,MNNN), GAMMA(MNB)
      REAL GAMM(MNB,MNNN), GAMO(MNB,MNNN), GAMP(MNB,MNNN)
      REAL KAPM(MNB), KAPP(MNB), KAPPA(MNB)
      REAL PHI(MNB), PRM(MNB+3)
      REAL PRP(MNB+3), PRO(MNB+3), RHO(MNB)
      REAL TT(MNNN)
      REAL XIOPP(MNB), XIOPPO(MNB,MNNN), XIOPPP(MNB,MNNN)
      REAL XIOPO(MNB), XIOPOO(MNB,MNNN), XIOPOP(MNB,MNNN)
      REAL XIMOO(MNB), XIMOOM(MNB,MNNN), XIMOOO(MNB,MNNN)
      REAL XIMOM(MNB), XIMOMM(MNB,MNNN), XIMOMO(MNB,MNNN)
      REAL CONE, DELM, DELP, DELTA, VX, VX1, VY, VZ, VZ1, VZ2
```

```
C
      EQUIVALENCE (CHIP,CHI), (ETAP,ETA), (BETA,GAMMA,DEL)
      EQUIVALENCE (BETO,GAMO,DELO), (BETP,GAMP)
C
C     BE SURE ALL THE LOCAL VARIABLES ARE RETAINED FOR THE
C     NEXT CALL.
C
      SAVE
C
C     EXTERNAL PROCEDURES
C
      REAL DGENER
      EXTERNAL DGENER
C
C     START OF EXECUTABLE STATEMENTS
C
C     CLEAR EVERYTHING FIRST
C
      ID=IDD
      CONE=NE(ID)*SIGE/NM(ID)/MHYD
      DO 10 I=1,NNN
         Q(I)=0.0
10    CONTINUE
      DO 30 I=1,NNN
         DO 20 J=1,NNN
            B(J,I)=0.0
            A(J,I)=0.0
            C(J,I)=0.0
20       CONTINUE
30    CONTINUE
C
C     TRANSFER EQUATIONS
C
C     FIRST DEPTH POINT
C
      IF (ID.GT.1)GO TO 210
C
C     CLEAR SUMMATIONS
C
      DO 40 IB=1,NB
         KAPP(IB)=0.
         KAPPA(IB)=0.
         RHO(IB)=0.
         PHI(IB)=0.
         BETA(IB)=0.
         XIOPP(IB)=0.0
         XIOPO(IB)=0.0
40    CONTINUE
      DO 60 IB=1,NB
         DO 50 J=1,NNN
            BETO(IB,J)=0.
            BETP(IB,J)=0.
            XIOPPO(IB,J)=0.
            XIOPPP(IB,J)=0.
            XIOPOO(IB,J)=0.
            XIOPOP(IB,J)=0.
50       CONTINUE
60    CONTINUE
C
C     ACCUMULATE SOURCE TERMS
C
      DO 70 IJ=1,NJ
         IB=BLOCK(IJ)
         KAPPA(IB)=KAPPA(IB)+WT(IJ)*FK(IJ,1)*RAD(IJ,1)
         KAPP(IB)=KAPP(IB)+WT(IJ)*FK(IJ,2)*RAD(IJ,2)
         RHO(IB)=RHO(IB)+WT(IJ)*RAD(IJ,1)
         PHI(IB)=PHI(IB)+WT(IJ)*RAD(IJ,1)*FH(IJ)
70    CONTINUE
      CALL DGENER(1,PRO)
C
C     SAVE LOWER LEVEL OPACITIES
C
      DO 90 J=1,NNN+1
         DO 80 IJ=1,NJ
            CHIO(IJ,J)=CHI(IJ,J)
            ETAO(IJ,J)=ETA(IJ,J)
80       CONTINUE
90    CONTINUE
      CALL DGENER(2,PRP)
C
```

```
C     SOURCE SUMS
C
      DO 110 IJ=1,NJ
         IB=BLOCK(IJ)
         VY=WT(IJ)*((CHIO(IJ,NNN+1)-CONE)*RAD(IJ,1)-ETAO(IJ,NNN+1))/
     :      CHIO(IJ,NNN+1)
         VZ=CHIO(IJ,NNN+1)+CHIP(IJ,NNN+1)
         BETA(IB)=BETA(IB)+VY*VZ
         DO 100 I=1,NNN
            VX=-CONE*MU1*PRO(I)/NM(1)
            BETO(IB,I)=BETO(IB,I)+WT(IJ)*((CHIO(IJ,I)-VX)*RAD(IJ,1)-
     :                 ETAO(IJ,I))*VZ/CHIO(IJ,NNN+1)+VY*CHIO(IJ,I)*
     :                 (1.-VZ/CHIO(IJ,NNN+1))
            BETP(IB,I)=BETP(IB,I)+VY*CHIP(IJ,I)
100      CONTINUE
         VX=CONE*(1.0+MU1*NE(1)/NM(1))
         BETO(IB,NNE)=BETO(IB,NNE)-WT(IJ)*VX*RAD(IJ,1)*VZ/CHIO(IJ,NNN+1)
         BETO(IB,IB)=BETO(IB,IB)+WT(IJ)*(CHIO(IJ,NNN+1)-CONE)*RAD(IJ,1)*
     :               VZ/CHIO(IJ,NNN+1)
110   CONTINUE
C
C     OPACITY SUMS
C
      DO 130 IJ=1,NJ
         IB=BLOCK(IJ)
         VX1=CHIP(IJ,NNN+1)+CHIO(IJ,NNN+1)
         VX=WT(IJ)*FK(IJ,2)*RAD(IJ,2)/VX1
         XIOPP(IB)=XIOPP(IB)+VX
         VY=WT(IJ)*FK(IJ,1)*RAD(IJ,1)/VX1
         XIOPO(IB)=XIOPO(IB)+VY
         DO 120 I=1,NNN
            XIOPPO(IB,I)=XIOPPO(IB,I)-VX*CHIO(IJ,I)/VX1
            XIOPPP(IB,I)=XIOPPP(IB,I)-VX*CHIP(IJ,I)/VX1
            XIOPOO(IB,I)=XIOPOO(IB,I)-VY*CHIO(IJ,I)/VX1
            XIOPOP(IB,I)=XIOPOP(IB,I)-VY*CHIP(IJ,I)/VX1
120      CONTINUE
130   CONTINUE
C
C     NORMALIZE
C
      DO 150 IB=1,NB
         XIOPP(IB)=2.*XIOPP(IB)/KAPP(IB)
         XIOPO(IB)=2.*XIOPO(IB)/KAPPA(IB)
         PHI(IB)=PHI(IB)/RHO(IB)
         BETA(IB)=0.5*BETA(IB)
150   CONTINUE
      DO 170 IB=1,NB
         DO 160 I=1,NNN
            BETO(IB,I)=0.5*BETO(IB,I)
            BETP(IB,I)=0.5*BETP(IB,I)
            XIOPPO(IB,I)=2.*XIOPPO(IB,I)/KAPP(IB)
            XIOPPP(IB,I)=2.*XIOPPP(IB,I)/KAPP(IB)
            XIOPOO(IB,I)=2.*XIOPOO(IB,I)/KAPPA(IB)
            XIOPOP(IB,I)=2.*XIOPOP(IB,I)/KAPPA(IB)
160      CONTINUE
170   CONTINUE
C
C     NOW CALCULATE THE ACTUAL LINEARIZATION MATRIX ELEMENTS
C
      DELP=M(2)-M(1)
      DO 190 IB=1,NB
         DO 180 I=1,NNN
            B(IB,I)=-(XIOPPO(IB,I)*KAPP(IB)-XIOPOO(IB,I)*KAPPA(IB))/
     :              DELP+0.5*DELP*BETO(IB,I)
            C(IB,I)=(XIOPPP(IB,I)*KAPP(IB)-XIOPOP(IB,I)*KAPPA(IB))/
     :              DELP-0.5*DELP*BETP(IB,I)
180      CONTINUE
190   CONTINUE
      DO 200 IB=1,NB
         B(IB,IB)=B(IB,IB)+XIOPO(IB)*KAPPA(IB)/DELP+PHI(IB)*RHO(IB)
         C(IB,IB)=C(IB,IB)+XIOPP(IB)*KAPP(IB)/DELP
         Q(IB)=(XIOPP(IB)*KAPP(IB)-XIOPO(IB)*KAPPA(IB))/DELP-
     :         PHI(IB)*RHO(IB)-0.5*DELP*BETA(IB)
200   CONTINUE
      GO TO 540
C
C     MOVE SUMS DOWN IF NOT AT SURFACE
C
210   DO 220 IB=1,NB
         KAPM(IB)=KAPPA(IB)
```

```
         KAPPA(IB)=KAPP(IB)
         KAPP(IB)=0.
         GAMMA(IB)=0.
         XIMOO(IB)=XIOPP(IB)
         XIOPP(IB)=0.0
         XIMOM(IB)=XIOPO(IB)
         XIOPO(IB)=0.0
220   CONTINUE
      DO 240 IB=1,NB
         DO 230 J=1,NNN
            GAMM(IB,J)=0.
            GAMO(IB,J)=0.
            GAMP(IB,J)=0.
            XIMOOM(IB,J)=XIOPPO(IB,J)
            XIOPPO(IB,J)=0.
            XIMOOO(IB,J)=XIOPPP(IB,J)
            XIOPPP(IB,J)=0.
            XIMOMM(IB,J)=XIOPOO(IB,J)
            XIOPOO(IB,J)=0.
            XIMOMO(IB,J)=XIOPOP(IB,J)
            XIOPOP(IB,J)=0.
230      CONTINUE
240   CONTINUE
      DO 250 I=1,NNN+1
         PRM(I)=PRO(I)
         PRO(I)=PRP(I)
250   CONTINUE
C
C     SAVE LOWER LEVEL OPACITIES
C
      DO 270 J=1,NNN+1
         DO 260 IJ=1,NJ
            CHIM(IJ,J)=CHIO(IJ,J)
            CHIO(IJ,J)=CHIP(IJ,J)
            ETAO(IJ,J)=ETAP(IJ,J)
260      CONTINUE
270   CONTINUE
C
C     TRANSFER EQUATION AT ORDINARY DEPTH POINT
C
      IF (ID.EQ.NDEPTH)GO TO 380
C
C     ACCUMULATE RADIATION MOMENT SUMS AND OPACITIES THAT
C     DRAW UPON THE PREVIOUS OPACITY CALCULATIONS
C
      DELM=DELP
      DELP=M(ID+1)-M(ID)
      DELTA=0.5*(DELP+DELM)
      CALL DGENER(ID+1,PRP)
C
C     SOURCE SUMS
C
      DO 290 IJ=1,NJ
         IB=BLOCK(IJ)
         KAPP(IB)=KAPP(IB)+WT(IJ)*FK(IJ,ID+1)*RAD(IJ,ID+1)
         VY=DELP*(CHIO(IJ,NNN+1)+CHIP(IJ,NNN+1))+DELM*(CHIO(IJ,NNN+1)+
     :      CHIM(IJ,NNN+1))
         VX=WT(IJ)*((CHIO(IJ,NNN+1)-CONE)*RAD(IJ,ID)-ETAO(IJ,NNN+1))/
     :      CHIO(IJ,NNN+1)
         GAMMA(IB)=GAMMA(IB)+VX*VY
         DO 280 J=1,NNN
            VZ=-CONE*MU1*PRO(J)/NM(ID)
            GAMM(IB,J)=GAMM(IB,J)+VX*DELM*CHIM(IJ,J)
            GAMP(IB,J)=GAMP(IB,J)+VX*DELP*CHIP(IJ,J)
            GAMO(IB,J)=GAMO(IB,J)+VX*((DELM+DELP)*CHIO(IJ,J)-VY*
     :                 CHIO(IJ,J)/CHIO(IJ,NNN+1))+WT(IJ)*VY*((
     :                 CHIO(IJ,J)-VZ)*RAD(IJ,ID)-ETAO(IJ,J))/
     :                 CHIO(IJ,NNN+1)
280      CONTINUE
         VZ=CONE*(1.0+MU1*NE(ID)/NM(ID))
         GAMO(IB,NNE)=GAMO(IB,NNE)-WT(IJ)*VZ*RAD(IJ,ID)*VY/
     :               CHIO(IJ,NNN+1)
         GAMO(IB,IB)=GAMO(IB,IB)+WT(IJ)*(CHIO(IJ,NNN+1)-CONE)*VY*
     :              RAD(IJ,ID)/CHIO(IJ,NNN+1)
290   CONTINUE
C
C     OPACITY SUMS
C
      DO 310 IJ=1,NJ
         IB=BLOCK(IJ)
```

```
            VX1=CHIP(IJ,NNN+1)+CHIO(IJ,NNN+1)
            VX=WT(IJ)*FK(IJ,ID+1)*RAD(IJ,ID+1)/VX1
            XIOPP(IB)=XIOPP(IB)+VX
            VY=WT(IJ)*FK(IJ,ID)*RAD(IJ,ID)/VX1
            XIOPO(IB)=XIOPO(IB)+VY
            DO 300 I=1,NNN
               XIOPPO(IB,I)=XIOPPO(IB,I)-VX*CHIO(IJ,I)/VX1
               XIOPPP(IB,I)=XIOPPP(IB,I)-VX*CHIP(IJ,I)/VX1
               XIOPOO(IB,I)=XIOPOO(IB,I)-VY*CHIO(IJ,I)/VX1
               XIOPOP(IB,I)=XIOPOP(IB,I)-VY*CHIP(IJ,I)/VX1
300         CONTINUE
310      CONTINUE
C
C        NORMALIZE
C
         DO 320 IB=1,NB
            XIOPP(IB)=2.*XIOPP(IB)/KAPP(IB)
            XIOPO(IB)=2.*XIOPO(IB)/KAPPA(IB)
            GAMMA(IB)=0.25*GAMMA(IB)
320      CONTINUE
         DO 340 IB=1,NB
            DO 330 I=1,NNN
               XIOPPO(IB,I)=2.*XIOPPO(IB,I)/KAPP(IB)
               XIOPPP(IB,I)=2.*XIOPPP(IB,I)/KAPP(IB)
               XIOPOO(IB,I)=2.*XIOPOO(IB,I)/KAPPA(IB)
               XIOPOP(IB,I)=2.*XIOPOP(IB,I)/KAPPA(IB)
               GAMM(IB,I)=0.25*GAMM(IB,I)
               GAMO(IB,I)=0.25*GAMO(IB,I)
               GAMP(IB,I)=0.25*GAMP(IB,I)
330         CONTINUE
340      CONTINUE
C
C        NOW CALCULATE THE ACTUAL LINEARIZATION MATRIX ELEMENTS
C
         DO 360 IB=1,NB
            DO 350 J=1,NNN
               A(IB,J)=(XIMOMM(IB,J)*KAPM(IB)-XIMOOM(IB,J)*KAPPA(IB))/
     :                 DELM-GAMM(IB,J)
               B(IB,J)=-(XIOPPO(IB,J)*KAPP(IB)-XIOPOO(IB,J)*KAPPA(IB))/
     :                 DELP+(XIMOOO(IB,J)*KAPPA(IB)-XIMOMO(IB,J)*KAPM(IB))/
     :                 DELM+GAMO(IB,J)
               C(IB,J)=(XIOPPP(IB,J)*KAPP(IB)-XIOPOP(IB,J)*KAPPA(IB))/
     :                 DELP-GAMP(IB,J)
350         CONTINUE
360      CONTINUE
         DO 370 IB=1,NB
            A(IB,IB)=A(IB,IB)+XIMOM(IB)*KAPM(IB)/DELM
            B(IB,IB)=B(IB,IB)+XIOPO(IB)*KAPPA(IB)/DELP+XIMOO(IB)*KAPPA(IB)/
     :              DELM
            C(IB,IB)=C(IB,IB)+XIOPP(IB)*KAPP(IB)/DELP
            Q(IB)=(XIOPP(IB)*KAPP(IB)-XIOPO(IB)*KAPPA(IB))/DELP-
     :           (XIMOO(IB)*KAPPA(IB)-XIMOM(IB)*KAPM(IB))/DELM-
     :           GAMMA(IB)
370      CONTINUE
         GO TO 540
C
C        TRANSFER EQUATION AT LOWER BOUNDARY
C
380      DO 390 IB=1,NB
            DEL(IB)=0.
390      CONTINUE
         DO 410 IB=1,NB
            DO 400 J=1,NNN
               DELO(IB,J)=0.
400         CONTINUE
410      CONTINUE
         DO 430 IJ=1,NJ
            IB=BLOCK(IJ)
            VX=HK*FREQ(IJ)/TEMP(NDEPTH)
            VX1=EXP(VX)
C
C        PLANK FUNCTION
C
            VZ=BBCOF*FREQ(IJ)**3/(VX1-1.0)
C
C        DERIVATIVE OF PLANK FUNCTION WITH TEMPERATURE
C
            VZ1=VZ*VX*VX1/(VX1-1.0)/TEMP(ID)
C
C        DERIVATIVE OF (DERIVATIVE OF PLANK FUNCTION WITH TEMPERATURE) WITH
```

```
C     LOG TEMPERATURE
C
         VZ2=VZ1*(-2.0-VX+2.0*VX*VX1/(VX1-1.0))
         VY=WT(IJ)*VZ1/CHIO(IJ,NNN+1)
         DEL(IB)=DEL(IB)+VY
         DO 420 J=1,NNN
            DELO(IB,J)=DELO(IB,J)-VY*CHIO(IJ,J)/CHIO(IJ,NNN+1)
420      CONTINUE
         DELO(IB,NNT)=DELO(IB,NNT)+WT(IJ)*VZ2/CHIO(IJ,NNN+1)
430   CONTINUE
      VY=0.
      DO 440 IB=1,NB
         VY=VY+DEL(IB)
440   CONTINUE
      DO 450 J=1,NNN
         TT(J)=0.
450   CONTINUE
      DO 470 IB=1,NB
         DO 460 J=1,NNN
            TT(J)=TT(J)+DELO(IB,J)
460      CONTINUE
470   CONTINUE
      DO 480 IB=1,NB
         DEL(IB)=HO*DEL(IB)/VY
480   CONTINUE
      DO 500 IB=1,NB
         DO 490 J=1,NNN
            DELO(IB,J)=(-DEL(IB)*TT(J)+HO*DELO(IB,J))/VY
490      CONTINUE
500   CONTINUE
C
C     NOW BUILD LINEARIZATION MATRICES
C
      DELM=DELP
      DO 520 IB=1,NB
         DO 510 J=1,NNN
            A(IB,J)=-(XIMOOM(IB,J)*KAPPA(IB)-XIMOMM(IB,J)*KAPM(IB))/DELM
            B(IB,J)=(XIMOOO(IB,J)*KAPPA(IB)-XIMOMO(IB,J)*KAPM(IB))/
     :              DELM-DELO(IB,J)
510      CONTINUE
520   CONTINUE
      DO 530 IB=1,NB
         A(IB,IB)=A(IB,IB)+XIMOM(IB)*KAPM(IB)/DELM
         B(IB,IB)=B(IB,IB)+XIMOO(IB)*KAPPA(IB)/DELM
         Q(IB)=DEL(IB)-(XIMOO(IB)*KAPPA(IB)-XIMOM(IB)*KAPM(IB))/
     :        DELM
530   CONTINUE
C
C     RADIATIVE EQUILIBRIUM
C
540   DO 550 IJ=1,NJ
         VX=WT(IJ)*((CHIO(IJ,NNN+1)-CONE)*RAD(IJ,ID)-ETAO(IJ,NNN+1))
         Q(NNT)=Q(NNT)+VX
550   CONTINUE
      DO 570 J=1,NNN
         VZ=-CONE*MU1*PRO(J)/NM(ID)
         DO 560 IJ=1,NJ
            B(NNT,J)=B(NNT,J)-WT(IJ)*((CHIO(IJ,J)-VZ)*RAD(IJ,ID)-
     :                ETAO(IJ,J))
560      CONTINUE
570   CONTINUE
      VZ=CONE*(1.0+MU1*NE(ID)/NM(ID))
      DO 580 IJ=1,NJ
         B(NNT,NNE)=B(NNT,NNE)+WT(IJ)*VZ*RAD(IJ,ID)
580   CONTINUE
      DO 590 IJ=1,NJ
         IB=BLOCK(IJ)
         B(NNT,IB)=B(NNT,IB)-WT(IJ)*(CHIO(IJ,NNN+1)-CONE)*RAD(IJ,ID)
590   CONTINUE
C
C     HYDROSTATIC EQUILIBRIUM
C
C     SURFACE
C
      IF (ID.GT.1)GO TO 630
      DO 610 IJ=1,NJ
         IB=BLOCK(IJ)
         VX=HYDCOF*WT(IJ)*FH(IJ)*RAD(IJ,1)
         Q(NNN)=Q(NNN)-VX*CHIO(IJ,NNN+1)
         DO 600 I=1,NNN
```

```
          B(NNN,I)=B(NNN,I)+VX*CHIO(IJ,I)
600     CONTINUE
        B(NNN,IB)=B(NNN,IB)+VX*CHIO(IJ,NNN+1)
610   CONTINUE
      DO 620 I=1,NNN
        B(NNN,I)=B(NNN,I)+PRO(I)*TEMP(1)/M(1)
620   CONTINUE
      B(NNN,NNT)=B(NNN,NNT)+PRO(NNN+1)*TEMP(1)/M(1)
      IF (ABS(Q(NNN)).GT.GRAV/KB)WRITE (6,*)'EDDINGTON LIMIT WARNING'
      Q(NNN)=Q(NNN)-PRO(NNN+1)*TEMP(1)/M(1)+GRAV/KB
      RETURN
C
C     NORMAL DEPTH POINT
C
630   DO 640 IB=1,NB
        A(NNN,IB)=PRM(IB)*TEMP(ID-1)+HYDCOF*KAPM(IB)
        B(NNN,IB)=PRO(IB)*TEMP(ID)+HYDCOF*KAPPA(IB)
        Q(NNN)=Q(NNN)-HYDCOF*(KAPPA(IB)-KAPM(IB))
640   CONTINUE
      A(NNN,NNT)=(PRM(NNT)+PRM(NNN+1))*TEMP(ID-1)
      A(NNN,NNN)=PRM(NNN)*TEMP(ID-1)
      B(NNN,NNT)=(PRO(NNT)+PRO(NNN+1))*TEMP(ID)
      B(NNN,NNN)=PRO(NNN)*TEMP(ID)
      Q(NNN)=Q(NNN)+(M(ID)-M(ID-1))*GRAV/KB-TEMP(ID)*PRO(NNN+1)+
     :       TEMP(ID-1)*PRM(NNN+1)
      RETURN
      END

      SUBROUTINE NUPOP(IDD)
C
C     RECALCULATE POPULATIONS BASED ON RADIATION FIELD
C
      IMPLICIT NONE
C
      COMA
      COMAI
      COMC
C
      INTEGER IDD, ID, I
C
      REAL COLRAT, LINSLV, RATEQ
      EXTERNAL COLRAT, LINSLV, RATEQ
C
      REAL SB, SBHE1, SBHE2
      SB(I,ID)=ACCOF*EXP(HK*FRQH(I)/TEMP(ID))*GH(I)/TEMP(ID)/
     :         SQRT(TEMP(ID))
      SBHE1(I,ID)=ACCOF*EXP(HK*FRQHE1(I)/TEMP(ID))*GHE1(I)/
     :              TEMP(ID)/SQRT(TEMP(ID))/2.
      SBHE2(I,ID)=ACCOF*EXP(HK*FRQHE2(I)/TEMP(ID))*GH(I)/
     :          TEMP(ID)/SQRT(TEMP(ID))
C
      ID=IDD
      CALL COLRAT(TEMP(ID),CR,CRHE1,CRHE2)
      CALL RATEQ(ID,NE(ID),AN,BN)
      CALL LINSLV(AN,BN,ANS,NEQN,NEQN)
      NTOT(ID)=NE(ID)
      DO 10 I=1,NLH
        N(I,ID)=ANS(I+NLHE1+NLHE2+1)
        NTOT(ID)=NTOT(ID)+N(I,ID)
10    CONTINUE
      NPROT(ID)=ANS(NLHE1+NLHE2+NLH+2)
      NTOT(ID)=NTOT(ID)+NPROT(ID)*(1.0+SUMH(ID)*NE(ID))
      DO 20 I=1,NLHE1
        NHE1(I,ID)=ANS(I)
        NTOT(ID)=NTOT(ID)+ANS(I)
20    CONTINUE
      DO 30 I=1,NLHE2
        NHE2(I,ID)=ANS(I+NLHE1)
        NTOT(ID)=NTOT(ID)+NHE2(I,ID)
30    CONTINUE
      NTOT(ID)=NTOT(ID)+NHE2(1,ID)*SUMHE(1,ID)*NE(ID)
      NHE3(ID)=ANS(NLHE1+NLHE2+1)
      NTOT(ID)=NTOT(ID)+NHE3(ID)*(1.0+NE(ID)*SUMHE(2,ID))
      DO 40 I=1,NLHS
        NS(I,ID)=NE(ID)*NPROT(ID)*SB(I,ID)
40    CONTINUE
      DO 50 I=1,NLHE1S
        NHE1S(I,ID)=NE(ID)*NHE2(1,ID)*SBHE1(I,ID)
50    CONTINUE
      DO 60 I=1,NLHE2S
        NHE2S(I,ID)=NE(ID)*NHE3(ID)*SBHE2(I,ID)
```

```
 60    CONTINUE
       NM(ID)=(NTOT(ID)-NE(ID))*MU1
       RETURN
       END

       SUBROUTINE NURATE(IDD)
C
C      CALCULATE RADIATIVE RATE BRACKETS
C
       IMPLICIT NONE
C
C      MACROS
C
       COMA
       COMAI
C
C      LOCAL VARIABLES
C
       INTEGER I, IDD, ID, IJ, IL, IT, J
       REAL DOP, DOPT, EX, FRQO, HKT, HKTT, P, S, SIGMA, SIGMAT, SRT
       REAL SRT2, X
       REAL DSPXDT(MNJ), SPX(MNJ), SR(MNJ)
C
C      EXTERNAL PROCEDURE
C
       REAL PARTI
       EXTERNAL PARTI
C
C      START OF EXECUTABLE STATEMENTS
C
       ID=IDD
       SRT=SQRT(TLINE/TEMP(ID))
       SRT2=SQRT(TEMP(ID))
       HKT=HK/TEMP(ID)
       HKTT=HKT/TEMP(ID)
C
C      CLEAR RATES
C
       DO 20 I=1,NLH+1
          DO 10 J=1,NLH+1
             RH(I,J)=0.
             DRH(I,J)=0.
10        CONTINUE
20     CONTINUE
       DO 40 I=1,NLHE1+1
          DO 30 J=1,NLHE1+1
             RHE1(I,J)=0.
             DRHE1(I,J)=0.
30        CONTINUE
40     CONTINUE
       DO 60 I=1,NLHE2+1
          DO 50 J=1,NLHE2+1
             RHE2(I,J)=0.
             DRHE2(I,J)=0.
50        CONTINUE
60     CONTINUE
C
C      PREPARE FREQUENCY-DEPENDENT VECTORS
C
       IF (FLTE)GO TO 260
       DO 70 IJ=1,NJ
          S=LAMC*SCOF*WT(IJ)/FREQ(IJ)
          EX=EXP(-HKT*FREQ(IJ))
          P=BBCOF*FREQ(IJ)**3+RAD(IJ,ID)
          SR(IJ)=S*RAD(IJ,ID)
          SPX(IJ)=S*P*EX
          DSPXDT(IJ)=HKTT*FREQ(IJ)*SPX(IJ)
70     CONTINUE
C
C      TEMPERATURE DERIVATIVE OF RATE
C
       DO 90 IL=1,NLH
          DO 80 IJ=1,ITPTRH-1
             DRH(NLH+1,IL)=DRH(NLH+1,IL)+SIG(IL,IJ)*DSPXDT(IJ)
80        CONTINUE
90     CONTINUE
C
C      SOME OF THE FOLLOWING LOOPS ARE DONE IN A CLUMSY WAY
C      IN ORDER TO ENHANCE VECTORIZATION.
C
       DO 120 IT=1,NLH
```

```
         DO 100 IJ=1,ITPTRH-1
            RH(IT,NLH+1)=RH(IT,NLH+1)+SIG(IT,IJ)*SR(IJ)
100      CONTINUE
         DO 110 IJ=1,ITPTRH-1
            RH(NLH+1,IT)=RH(NLH+1,IT)+SIG(IT,IJ)*SPX(IJ)
110      CONTINUE
120   CONTINUE
      DO 140 IL=1,NLHE1
         DO 130 IJ=1,ITPTRH-1
            DRHE1(NLHE1+1,IL)=DRHE1(NLHE1+1,IL)+SIGHE1(IL,IJ)*DSPXDT(IJ)
130      CONTINUE
140   CONTINUE
      DO 170 IL=1,NLHE1
         DO 150 IJ=1,ITPTRH-1
            RHE1(IL,NLHE1+1)=RHE1(IL,NLHE1+1)+SIGHE1(IL,IJ)*SR(IJ)
150      CONTINUE
         DO 160 IJ=1,ITPTRH
            RHE1(NLHE1+1,IL)=RHE1(NLHE1+1,IL)+SIGHE1(IL,IJ)*SPX(IJ)
160      CONTINUE
170   CONTINUE
      DO 190 IL=1,NLHE2
         DO 180 IJ=1,ITPTRH-1
            DRHE2(NLHE2+1,IL)=DRHE2(NLHE2+1,IL)+SIGHE2(IL,IJ)*DSPXDT(IJ)
180      CONTINUE
190   CONTINUE
      DO 220 IL=1,NLHE2
         DO 200 IJ=1,ITPTRH-1
            RHE2(IL,NLHE2+1)=RHE2(IL,NLHE2+1)+SIGHE2(IL,IJ)*SR(IJ)
200      CONTINUE
         DO 210 IJ=1,ITPTRH-1
            RHE2(NLHE2+1,IL)=RHE2(NLHE2+1,IL)+SIGHE2(IL,IJ)*SPX(IJ)
210      CONTINUE
220   CONTINUE
C
C     LINE TRANSITIONS
C
      DO 230 IJ=ITPTRH,ITPTRHE1-1
         I=LOWH(IJ)
         J=UPH(IJ)
         IT=ITRH(I,J)
         FRQ0=FRQH(I)-FRQH(J)
         DOP=SRT2*FRQ0*DOPCOF
         X=DELQUAD*MOD(IJ-ITPTRH,NQUAD)*SRT
         SIGMA=PIE2MC*OSCH(IT)*EXP(-X*X)/DOP/1.7724539
         SIGMAT=SIGMA*(X*X-0.5)/TEMP(ID)
         RH(I,J)=RH(I,J)+SIGMA*SR(IJ)
         DRH(I,J)=DRH(I,J)+SIGMAT*SR(IJ)
         RH(J,I)=RH(J,I)+SIGMA*SPX(IJ)
         DRH(J,I)=DRH(J,I)+SIGMAT*SPX(IJ)+SIGMA*DSPXDT(IJ)
230   CONTINUE
      DO 240 IJ=ITPTRHE1,ITPTRHE2-1
         I=LOWHE1(IJ)
         J=UPHE1(IJ)
         IT=ITRHE1(I,J)
         FRQ0=FRQHE1(I)-FRQHE1(J)
         DOP=SRT2*FRQ0*DOPCOF*0.5
         X=DELQUAD*MOD(IJ-ITPTRHE1,NQUAD)*SRT
         SIGMA=PIE2MC*OSCHE1(IT)*EXP(-X*X)/DOP/1.7724539
         SIGMAT=SIGMA*(X*X-0.5)/TEMP(ID)
         RHE1(I,J)=RHE1(I,J)+SIGMA*SR(IJ)
         DRHE1(I,J)=DRHE1(I,J)+SIGMAT*SR(IJ)
         RHE1(J,I)=RHE1(J,I)+SIGMA*SPX(IJ)
         DRHE1(J,I)=DRHE1(J,I)+SIGMAT*SPX(IJ)+SIGMA*DSPXDT(IJ)
240   CONTINUE
      DO 250 IJ=ITPTRHE2,NJ
         I=LOWHE2(IJ)
         J=UPHE2(IJ)
         IT=ITRHE2(I,J)
         FRQ0=FRQHE2(I)-FRQHE2(J)
         DOP=SRT2*FRQ0*DOPCOF*0.5
         X=DELQUAD*MOD(IJ-ITPTRHE2,NQUAD)*SRT
         SIGMA=PIE2MC*OSCHE2(IT)*EXP(-X*X)/DOP/1.7724539
         SIGMAT=SIGMA*(X*X-0.5)/TEMP(ID)
         RHE2(I,J)=RHE2(I,J)+SIGMA*SR(IJ)
         DRHE2(I,J)=DRHE2(I,J)+SIGMAT*SR(IJ)
         RHE2(J,I)=RHE2(J,I)+SIGMA*SPX(IJ)
         DRHE2(J,I)=DRHE2(J,I)+SIGMAT*SPX(IJ)+SIGMA*DSPXDT(IJ)
250   CONTINUE
260   CALL PARTI(ID)
      RETURN
```

```
      END

      SUBROUTINE OUT
C
C     APPLY CORRECTIONS CALCULATED IN LINEAR.
C     ONLY RAD, NE, AND TEMP CORRECTIONS ARE BOTHERED WITH
C     SINCE THE REST ARE DEFINED BY THE SUBSEQUENT LAMBDA ITERATION.
C
      IMPLICIT NONE
C
      COMA
      COMAI
C
      REAL ERRMAX, ERRJMAX, ERRTMAX, ERRTOT, NU(MNNN)
      INTEGER IB, ID, IJ, IL
      REAL DEL, DELJ
C
      INTEGER IOSTATUS
      REAL RDABS
      EXTERNAL IOSTATUS, RDABS
C
      ERRMAX=0.
      ERRTMAX=0.
      ERRJMAX=0.
      ERRTOT=0.
      DO 20 ID=1,NDEPTH
         CALL RDABS(9,NU,NNN,(ID-1)*MNNN)
         IJ=IOSTATUS(9,IL)
      OT=0.
      DO 20 ID=1,NDEPTH
         CALL RDABS(9,NU,NNN,(ID-1)*MNNN)
         IJ=IOSTATUS(9,IL)
         DO 10 IB=1,NB
            ERRJMAX=MAX(ABS(NU(IB)),ERRJMAX)
10       CONTINUE
         ERRMAX=MAX(ERRMAX,ABS(NU(NNE)))
         ERRTMAX=MAX(ERRTMAX,ABS(NU(NNT)))
20    CONTINUE
      WRITE (6,*)'MAXIMUM J ERROR = ',ERRJMAX
      WRITE (6,*)'MAXIMUM NE ERROR = ',ERRMAX
      WRITE (6,*)'MAXIMUM T ERROR = ',ERRTMAX
      FEXIT=.FALSE.
      ERRTOT=MAX(ERRTMAX,ERRMAX)
      IF (ERRTOT.LT.(.0001).AND.LAMC.EQ.(1.).AND.LAML.EQ.(1.))
     :   FEXIT=.TRUE.
      LINERR=ERRTOT
C
C     SCALE DOWN LARGE CORRECTIONS TO INCREASE CIRCLE OF CONVERGENCE
C
      DEL=MIN(0.2/ERRTOT,1.0)
      WRITE (6,*)'USING CORRECTION FACTOR OF ',DEL
C
C     NOW APPLY CORRECTIONS
C
      DO 60 ID=1,NDEPTH
         CALL RDABS(9,NU,NNN,(ID-1)*MNNN)
         IJ=IOSTATUS(9,IL)
         DO 50 IJ=1,NJ
            IB=BLOCK(IJ)
            RAD(IJ,ID)=RAD(IJ,ID)*MAX(0.1,1.0+DEL*NU(IB))
50       CONTINUE
         TEMP(ID)=TEMP(ID)*(1.0+DEL*NU(NNT))
         NE(ID)=NE(ID)*(1.0+DEL*NU(NNE))
60    CONTINUE
      RETURN
      END

      SUBROUTINE PARTI(IDD)
C
C     CALCULATE UPPER-STATE SUMS
C
      IMPLICIT NONE
C
      COMA
      COMAI
C
      INTEGER IDD, ID, IL, IT
      REAL HKT, X
C
      REAL SB, SBHE1, SBHE2
```

```
      SB(IL,ID)=ACCOF*EXP(HK*FRQH(IL)/TEMP(ID))*GH(IL)/TEMP(ID)/
     :          SQRT(TEMP(ID))
      SBHE1(IL,ID)=ACCOF*EXP(HK*FRQHE1(IL)/TEMP(ID))*GHE1(IL)/
     :             TEMP(ID)/SQRT(TEMP(ID))/2.
      SBHE2(IL,ID)=ACCOF*EXP(HK*FRQHE2(IL)/TEMP(ID))*GH(IL)/
     :             TEMP(ID)/SQRT(TEMP(ID))
C
      ID=IDD
      HKT=HK/TEMP(ID)
      SUMH(ID)=0.
      DSHDT(ID)=0.
      DO 10 IT=1,2
         SUMHE(IT,ID)=0.
         DSHEDT(IT,ID)=0.
10    CONTINUE
      DO 20 IL=NLH+1,MQH
         X=SB(IL,ID)
         SUMH(ID)=SUMH(ID)+X
         DSHDT(ID)=DSHDT(ID)-X*(1.5+HKT*FRQH(IL))/TEMP(ID)
20    CONTINUE
      DO 30 IL=NLHE1+1,MQHE1
         X=SBHE1(IL,ID)
         SUMHE(1,ID)=SUMHE(1,ID)+X
         DSHEDT(1,ID)=DSHEDT(1,ID)-X*(1.5+HKT*FRQHE1(IL))/TEMP(ID)
30    CONTINUE
      DO 40 IL=NLHE2+1,MQH*2
         X=SBHE2(IL,ID)
         SUMHE(2,ID)=SUMHE(2,ID)+X
         DSHEDT(2,ID)=DSHEDT(2,ID)-X*(1.5+HKT*FRQHE2(IL))/TEMP(ID)
40    CONTINUE
      RETURN
      END

      SUBROUTINE PUTOUT
C
C     PUNCH OUT RESULT
C
      IMPLICIT NONE
C
      COMA
      COMAI
C
      INTEGER ID,I
C
      WRITE (12,1001)(M(ID),TEMP(ID),NTOT(ID),NE(ID),ID=1,NDEPTH)
      WRITE (12,1002)1
      WRITE (12,1002)0,0,NLH,NTRH
      WRITE (12,1003)(FRQH(I),I=1,NLH)
      DO 10 I=1,NTRH
         WRITE (12,1002)LOWERH(I),UPPERH(I)
10    CONTINUE
      WRITE (12,1003)((N(I,ID),I=1,NLH),NPROT(ID),ID=1,NDEPTH)
      WRITE (12,1002)2
      WRITE (12,1002)0,0,NLHE1,NTRHE1,NLHE2,NTRHE2
      WRITE (12,1003)(FRQHE1(I),I=1,NLHE1)
      DO 20 I=1,NTRHE1
         WRITE (12,1002)LOWERHE1(I),UPPERHE1(I)
20    CONTINUE
      WRITE (12,1003)(FRQHE2(I),I=1,NLHE2)
      DO 30 I=1,NTRHE2
         WRITE (12,1002)LOWERHE2(I),UPPERHE2(I)
30    CONTINUE
      WRITE (12,1003)((NHE1(I,ID),I=1,NLHE1),(NHE2(I,ID),I=1,NLHE2),
     :                NHE3(ID),ID=1,NDEPTH)
      WRITE (12,1002)0
      RETURN
C
1001  FORMAT (4E15.7)
1002  FORMAT (16I5)
1003  FORMAT (5E15.7)
      END

      SUBROUTINE RATEQ(IDD,ELEC,AAN,BBN)
C
C     PRODUCE RATE EQUATIONS
C
      IMPLICIT NONE
C
      COMA
      COMAI
```

```
      COMC
C
      REAL AAN(NEQN,NEQN), BBN(NEQN)
      INTEGER ID, IDD, I, II, IT, J, JJ, NO
      REAL ELEC, HKT, X
C
      REAL NURATE
      EXTERNAL NURATE
C
      REAL SB, SBHE1, SBHE2
      SB(I,ID)=ACCOF*EXP(HK*FRQH(I)/TEMP(ID))*GH(I)/TEMP(ID)/
     :         SQRT(TEMP(ID))
      SBHE1(I,ID)=ACCOF*EXP(HK*FRQHE1(I)/TEMP(ID))*GHE1(I)/
     :             TEMP(ID)/SQRT(TEMP(ID))/2.
      SBHE2(I,ID)=ACCOF*EXP(HK*FRQHE2(I)/TEMP(ID))*GH(I)/
     :        TEMP(ID)/SQRT(TEMP(ID))
C
      ID=IDD
      CALL NURATE(ID)
      DO 10 I=1,NEQN
         BBN(I)=0.
10    CONTINUE
      DO 30 I=1,NEQN
         DO 20 J=1,NEQN
            AAN(I,J)=0.
20       CONTINUE
30    CONTINUE
      HKT=HK/TEMP(ID)
      DO 40 I=1,NLHE1
         AAN(I,I)=RHE1(I,NLHE1+1)+ELEC*CRHE1(I,NLHE1+1)
         AAN(I,NLHE1+1)=-ELEC*SBHE1(I,ID)*(RHE1(NLHE1+1,I)+
     :                 ELEC*CRHE1(I,NLHE1+1))
40    CONTINUE
      DO 70 I=1,NLHE1
         DO 50 J=1,I-1
            AAN(I,I)=AAN(I,I)+ELEC*CRHE1(I,J)
            AAN(I,J)=-ELEC*CRHE1(J,I)
            X=GHE1(J)*EXP(HKT*(FRQHE1(J)-FRQHE1(I)))/GHE1(I)
            AAN(I,I)=AAN(I,I)+X*RHE1(I,J)
            AAN(I,J)=AAN(I,J)-RHE1(J,I)
50       CONTINUE
         DO 60 J=I+1,NLHE1
            AAN(I,I)=AAN(I,I)+ELEC*CRHE1(I,J)
            AAN(I,J)=-ELEC*CRHE1(J,I)
            X=GHE1(I)*EXP(HKT*(FRQHE1(I)-FRQHE1(J)))/GHE1(J)
            AAN(I,I)=AAN(I,I)+RHE1(I,J)
            AAN(I,J)=AAN(I,J)-X*RHE1(J,I)
60       CONTINUE
70    CONTINUE
      DO 100 I=1,NLHE2
         II=I+NLHE1
         AAN(II,II)=RHE2(I,NLHE2+1)+ELEC*CRHE2(I,NLHE2+1)
         AAN(II,NLHE1+NLHE2+1)=-ELEC*SBHE2(I,ID)*(RHE2(NLHE2+1,I)+
     :         ELEC*CRHE2(I,NLHE2+1))
         DO 80 J=1,I-1
            JJ=NLHE1+J
            AAN(II,II)=AAN(II,II)+ELEC*CRHE2(I,J)
            AAN(II,JJ)=-ELEC*CRHE2(J,I)
            X=GH(J)*EXP(HKT*(FRQHE2(J)-FRQHE2(I)))/GH(I)
            AAN(II,II)=AAN(II,II)+X*RHE2(I,J)
            AAN(II,JJ)=AAN(II,JJ)-RHE2(J,I)
80       CONTINUE
         DO 90 J=I+1,NLHE2
            JJ=J+NLHE1
            AAN(II,II)=AAN(II,II)+ELEC*CRHE2(I,J)
            AAN(II,JJ)=-ELEC*CRHE2(J,I)
            X=GH(I)*EXP(HKT*(FRQHE2(I)-FRQHE2(J)))/GH(J)
            AAN(II,II)=AAN(II,II)+RHE2(I,J)
            AAN(II,JJ)=AAN(II,JJ)-X*RHE2(J,I)
90       CONTINUE
100   CONTINUE
      DO 110 I=1,NLHE1+NLHE2+1
         AAN(NLHE1+NLHE2+1,I)=1.0
110   CONTINUE
      AAN(NLHE1+NLHE2+1,NLHE2+1)=AAN(NLHE1+NLHE2+1,NLHE2+1)+
     :          ELEC*SUMHE(1,ID)
      AAN(NLHE1+NLHE2+1,NLHE1+NLHE2+1)=AAN(NLHE1+NLHE2+1,NLHE1+NLHE2+1)+
     :                                ELEC*SUMHE(2,ID)
      DO 120 I=1,NLH+1
         AAN(NLHE1+NLHE2+1,I+NLHE1+NLHE2+1)=-Y
```

```
120   CONTINUE
      AAN(NLHE1+NLHE2+1,NEQN)=AAN(NLHE1+NLHE2+1,NEQN)-Y*ELEC*SUMH(ID)
      DO 150 I=1,NLH
         II=I+NLHE1+NLHE2+1
         AAN(II,II)=RH(I,NLH+1)+ELEC*CR(I,NLH+1)
         DO 130 J=1,I-1
            JJ=J+NLHE1+NLHE2+1
            AAN(II,II)=AAN(II,II)+ELEC*CR(I,J)
            AAN(II,JJ)=-ELEC*CR(J,I)
            X=GH(J)*EXP(HKT*(FRQH(J)-FRQH(I)))/GH(I)
            AAN(II,II)=AAN(II,II)+X*RH(I,J)
            AAN(II,JJ)=AAN(II,JJ)-RH(J,I)
130      CONTINUE
         DO 140 J=I+1,NLH
            JJ=J+NLHE1+NLHE2+1
            AAN(II,II)=AAN(II,II)+ELEC*CR(I,J)
            AAN(II,JJ)=-ELEC*CR(J,I)
            X=GH(I)*EXP(HKT*(FRQH(I)-FRQH(J)))/GH(J)
            AAN(II,II)=AAN(II,II)+RH(I,J)
            AAN(II,JJ)=AAN(II,JJ)-X*RH(J,I)
140      CONTINUE
         AAN(II,NEQN)=-ELEC*SB(I,ID)*(RH(NLH+1,I)+ELEC*CR(I,NLH+1))
150   CONTINUE
      DO 160 I=1,NLHE2
         J=I+NLHE1
         AAN(NEQN,J)=1.0
160   CONTINUE
      AAN(NEQN,NLHE1+NLHE2+1)=2.0+ELEC*SUMHE(2,ID)
      AAN(NEQN,NEQN)=1.0
      BBN(NEQN)=ELEC
      RETURN
      END

      SUBROUTINE SETUP
C
C     READ IN MODEL APPROXIMATION
C     PERFORM ALL PRECALCULATIONS (E.G. OF CROSS-SECTIONS)
C
      IMPLICIT NONE
C
C     MACROS
C
      COMA
      COMAI
      COMW
C
C     LOCAL VARIABLES
C
      CHARACTER*80 HEADER
      INTEGER  I, II, ID, IJ, IL, IT, J, N0, N1, N2, N3
      REAL ABUND(92)
      REAL CON, DOP, FCON, FRQ3, GLOG, SRT, TEFF, V1, Z
C
C     TABLE OF ATOMIC WEIGHTS
C
      REAL WEIGHT(92)
      DATA WEIGHT/1.0,4.0,6.9,9.0,10.8,12.0,14.0,16.0,19.0,20.2,23.0,
     : 24.3,27.0,28.1,31.0,32.1,35.5,39.9,39.1,40.1,45.0,47.9,50.9,52.0,
     : 54.9,55.8,58.9,58.7,63.5,65.4,69.7,72.6,74.9,79.0,79.9,83.8,85.5,
     : 87.6,88.9,91.2,92.9,95.9,98.9,101.1,102.9,106.4,107.9,112.4,
     : 114.8,118.7,121.8,127.6,126.9,131.3,132.9,137.3,138.9,140.1,
     : 140.9,144.2,145.0,150.4,152.0,157.3,158.9,162.5,164.9,167.3,
     : 168.9,173.0,175.0,178.5,180.9,183.9,186.2,190.2,192.2,195.1,
     : 197.0,200.6,204.4,207.2,209.0,209.0,210.0,222.0,223.0,226.0,
     : 227.0,232.0,231.0,238.0/
C
C     TABLE GIVING QUANTUM NUMBER OF ACTIVE ELECTRON OF EACH NEUTRAL
C     HELIUM STATE TREATED BY THE PROGRAM
C
      INTEGER QN(25)
      DATA QN/1,4*2,6*3,8*4,5,6,7,8,9,10/
C
C     EXTERNAL PROCEDURES
C
      REAL CREATE, EXIT, GAUNT, HEGAUNT, NUPOP, PARTI
      EXTERNAL CREATE, EXIT, GAUNT, HEGAUNT, NUPOP, PARTI
C
C     STATEMENT FUNCTIONS
C
      REAL SB, SBHE1, SBHE2
      SB(IL,ID)=ACCOF*EXP(HK*FRQH(IL)/TEMP(ID))*GH(IL)/TEMP(ID)/
```

```
     :          SQRT(TEMP(ID))
      SBHE1(IL,ID)=ACCOF*EXP(HK*FRQHE1(IL)/TEMP(ID))*GHE1(IL)/TEMP(ID)/
     :              SQRT(TEMP(ID))/2.
      SBHE2(IL,ID)=ACCOF*EXP(HK*FRQHE2(IL)/TEMP(ID))*GH(IL)/TEMP(ID)/
     :              SQRT(TEMP(ID))
C
C     START OF EXECUTABLE STATEMENTS
C
      WRITE (6,*)'NON-LTE ATMOSPHERE PROGRAM'
      READ (5,1001)HEADER
      WRITE (12,1001)HEADER
      READ (5,1002)TEFF,TLINE,GLOG,Y,Z,NITER,FLTE,FPRINT,FSWITCH
      WRITE (12,1002)TEFF,TLINE,GLOG,Y,Z,NITER,FLTE,FPRINT,0
      IF (FSWITCH)READ (5,1007)LAMC
      IF (FLTE)THEN
         WRITE (6,*)'LTE MODEL'
      ELSE
         WRITE (6,*)'NON-LTE MODEL'
      ENDIF
      IF (FSWITCH)WRITE (6,*)'RADIATIVE/COLLISIONAL SWITCHING EMPLOYED'
      IF (FPRINT)WRITE (6,*)'DIAGNOSTICS PRINTED'
      H0=5.6692E-5*TEFF**4/(4.*PI)
      GRAV=EXP(2.302585093*GLOG)
      WRITE (6,1001)HEADER
      WRITE (6,1004)TEFF,GRAV,Y,H0
C
C     ELEMENTAL ABUNDANCES
C
      READ (5,1005)(ABUND(I),I=1,92)
      WRITE (12,1005)(ABUND(I),I=1,92)
      MU1=1.+4.*Y
      ZTOT=1.+Y
      IF (FPRINT)WRITE (6,*)' ELEMENTAL ABUNDANCES'
      DO 10 I=3,92
         V1=1.E-12*EXP(2.302585093*ABUND(I))*Z
         IF (FPRINT)WRITE (6,*)I,V1
         ZTOT=ZTOT+V1
         MU1=MU1+Z*WEIGHT(I)
10    CONTINUE
      MU1=MU1/ZTOT
      WRITE (6,*)'MEAN MOLECULAR WEIGHT = ',MU1
      WRITE (6,*)'NUCLEI PER PROTON = ',ZTOT
C
C     WAVELENGTH GRID
C
      READ (5,1003)NDEPTH,NJ
      WRITE (12,1003)NDEPTH,NJ
      WRITE (6,*)'DEPTH POINTS = ',NDEPTH
      READ (5,1006)(FREQ(I),I=1,NJ)
      WRITE (12,1006)(FREQ(I),I=1,NJ)
C
C     QUADRATURE WEIGHTS
C
      DO 20 IJ=1,MNJ
         WTO(IJ)=0.
20    CONTINUE
30    READ (5,1003)NO,N1
      WRITE (12,1003)NO,N1
      IF (NO.EQ.1)THEN
C
C     SKIP ENTRY
C
         GO TO 30
      ELSE IF (NO.EQ.2)THEN
C
C     TRAPEZOIDAL RULE
C
         V1=ABS(FREQ(N1+1)-FREQ(N1))
         WTO(N1+1)=WTO(N1+1)+0.5*V1
         WTO(N1)=WTO(N1)+0.5*V1
         GO TO 30
      ELSE IF (NO.EQ.3)THEN
C
C     SIMPSON'S RULE
C
         V1=ABS(FREQ(N1+1)-FREQ(N1-1))
         WTO(N1+1)=WTO(N1+1)+V1/6.
         WTO(N1)=WTO(N1)+2.*V1/3.
         WTO(N1-1)=WTO(N1-1)+V1/6.
         GO TO 30
```

```
      ENDIF
C
C     READ OPACITY RULE
C
      READ (5,1003)RULE
      WRITE (12,1003)RULE
      IF (RULE(2))THEN
         WRITE (6,*)'LINES INCLUDED'
      ELSE
         WRITE (6,*)'CONTINUUM ONLY'
      ENDIF
      READ (5,1007)(M(I),TEMP(I),NTOT(I),NE(I),I=1,NDEPTH)
C
C     MASS GRID OR MASS DIFFERENCE GRID?  MAKE SURE OF THE FORMER
C
      IF (M(NDEPTH).LE.M(NDEPTH-1))THEN
         DO 40 ID=2,NDEPTH
            M(ID)=M(ID)+M(ID-1)
40       CONTINUE
      ENDIF
C
C     READ ATOMIC OCCUPATION NUMBERS
C
50       READ (5,1003)I
         IF (I.EQ.1)THEN
C
C     READ NUMBER OF HYDROGEN LEVELS TO USE FOR EACH IONIZATION.
C     ALSO READ LINES TO USE.
C
            READ (5,1003)N0,N1,N2,NTRH
            IF (N1.NE.0)THEN
               WRITE (6,*)'PLEASE DO NOT SPECIFY ANY H- LINES.'
               CALL EXIT(1)
            ENDIF
C
C     THROW AWAY NEGATIVE HYDROGEN IONIZATION FREQUENCY.
C
            IF (N0.NE.0)READ (5,1006)CON
C
C     READ NEUTRAL HYDROGEN IONIZATION FREQUENCIES.
C
            READ (5,1006)(FRQH(IL),IL=1,N2)
C
C     READ LIST OF LINES TO USE FOR HYDROGEN
C
            IF (NTRH.GT.0)
     :          READ (5,1008)(LOWERH(J),UPPERH(J),J=1,NTRH)
C
C     READ ALL HYDROGEN OCCUPATION NUMBERS.
C     NEGATIVE HYDROGEN ION OCCUPATION IS THROWN AWAY.
C
            READ (5,1006)((CON,IL=1,N0),(N(IL,ID),IL=1,N2),
     :                    NPROT(ID),ID=1,NDEPTH)
C
C     CALCULATE LTE POPULATIONS FOR LEVELS NOT INCLUDED IN APPROXIMATE
C     INPUT MODEL.
C
            DO 70 ID=1,NDEPTH
               DO 60 IL=N2+1,NLH
                  N(IL,ID)=NPROT(ID)*NE(ID)*SB(IL,ID)
60             CONTINUE
70          CONTINUE
            GO TO 50
         ELSE IF (I.EQ.2)THEN
C
C     TREAT HELIUM THE SAME WAY AS HYDROGEN.
C
            READ (5,1003)N0,N1,N2,NTRHE1,N3,NTRHE2
            IF (N1.NE.0)THEN
               WRITE (6,*)'PLEASE DO NOT SPECIFY ANY HE- LINES.'
               CALL EXIT(1)
            ENDIF
            IF (N0.NE.0)READ (5,1006)CON
            READ (5,1006)(FRQHE1(IL),IL=1,N2)
            IF (NTRHE1.GT.0)
     :         READ (5,1008)(LOWERHE1(J),UPPERHE1(J),J=1,NTRHE1)
            READ (5,1006)(FRQHE2(IL),IL=1,N3)
            IF (NTRHE2.GT.0)
     :         READ (5,1008)(LOWERHE2(J),UPPERHE2(J),J=1,NTRHE2)
            READ (5,1006)((CON,IL=1,N0),(NHE1(IL,ID),IL=1,N2),
```

```
     :                     (NHE2(IL,ID),IL=1,N3),NHE3(ID),ID=1,NDEPTH)
           DO 100 ID=1,NDEPTH
              DO 80 IL=N2+1,NLHE1
                 NHE1(IL,ID)=NHE2(1,ID)*NE(ID)*SBHE1(IL,ID)
80            CONTINUE
              DO 90 IL=N3+1,NLHE2
                 NHE2(IL,ID)=NHE3(ID)*NE(ID)*SBHE2(IL,ID)
90            CONTINUE
100        CONTINUE
           GO TO 50
        ELSE IF (I.NE.0)THEN
C
C      OTHER ELEMENTS INCLUDED BUT NOT WANTED.
C      PARDON THE ANACHRONISTIC USE OF THE TERMS "CARDS" AND "DECK."
C      THESE ARE PROBABLY REALLY LINES IN AN EDITOR-CREATED DISK FILE.
C
           WRITE (6,*) 'PLEASE REMOVE HEAVY ION CARDS FROM DECK'
           CALL EXIT(1)
        ENDIF
C
C     REMAINDER OF DECK IS IGNORED.
C     PREPARE PRECALCULATED QUANTITIES.
C
      ITPTRH=NJ+1
      ITPTRHE1=ITPTRH
      ITPTRHE2=ITPTRH
      IF (RULE(2))THEN
         SRT=SQRT(TLINE)
C
C     HYDROGEN LINES
C
         N1=NTRH
         DO 130 IL=1,N1
C
C     GET UPPER AND LOWER LEVEL NUMBERS.
C
            J=UPPERH(IL)
            I=LOWERH(IL)
            IT=ITRH(I,J)
C
C     REJECT LINE IF FORBIDDEN
C
            IF (IT.EQ.0)THEN
               WRITE (6,*)'H LINE ',I,' TO ',J,' IS FORBIDDEN.'
               GO TO 130
            ENDIF
C
C     CALCULATE LINE CENTRAL FREQUENCY AND DOPPLER WIDTH.
C     NOTE THAT THIS PROGRAM ASSUMES LINE BROADENING IS DOMINATED
C     BY DOPPLER BROADENING.
C
            FCON=FRQH(I)-FRQH(J)
            DOP=SRT*FCON*DOPCOF
            IF (FCON.LE.0)THEN
               WRITE (6,*) 'H IONIZATION FREQUENCY ERROR FOR LINE ',
     :                     I,',',J
               FCON=-FCON
            ENDIF
            IF (FPRINT)WRITE (6,*)' H ',I,' TO ',J,': FREQ = ',FCON,
     :                            ' DOP = ',DOP
C
C     ADD FREQUENCIES TO FREQUENCY LIST FOR THE LINE.
C
            DO 110 II=1,NQUAD
               FREQ(II+NJ)=FCON
               UPH(II+NJ)=J
               LOWH(II+NJ)=I
110         CONTINUE
C
C     CALCULATE WEIGHTS FOR THE LINE FREQUENCIES.
C     NOTE THAT SIMPSON'S RULE IS USED; THUS NQUAD SHOULD BE ODD.
C
            DO 120 I=1,NQUAD-2,2
               WTO(I+NJ)=WTO(I+NJ)+2.0*DELQUAD*DOP/3.0
               WTO(I+NJ+1)=WTO(I+NJ+1)+8.*DELQUAD*DOP/3.0
               WTO(I+NJ+2)=WTO(I+NJ+2)+2.0*DELQUAD*DOP/3.0
120         CONTINUE
            NJ=NJ+NQUAD
130      CONTINUE
C
```

```
C     NOW DO ALL THE SAME FOR NEUTRAL AND SINGLY IONIZED HELIUM.
C
         ITPTRHE1=NJ+1
         N1=NTRHE1
         DO 160 IL=1,N1
            J=UPPERHE1(IL)
            I=LOWERHE1(IL)
            IT=ITRHE1(I,J)
            IF (IT.EQ.0)THEN
               WRITE (6,*)'HE1 LINE ',I,' TO ',J,' IS FORBIDDEN.'
               GO TO 160
            ENDIF
            FCON=FRQHE1(I)-FRQHE1(J)
            DOP=SRT*FCON*DOPCOF*0.5
            IF (FCON.LE.0)THEN
               WRITE (6,*) 'HE1 IONIZATION FREQUENCY ERROR FOR LINE ',
     :                     I,',',J
               FCON=-FCON
            ENDIF
            IF (FPRINT)WRITE (6,*)' HE1 ',I,' TO ',J,': FREQ = ',FCON,
     :                            ' DOP = ',DOP
            DO 140 II=1,NQUAD
               FREQ(II+NJ)=FCON
               UPHE1(II+NJ)=J
               LOWHE1(II+NJ)=I
140         CONTINUE
            DO 150 I=1,NQUAD-2,2
               WTO(I+NJ)=WTO(I+NJ)+2.0*DELQUAD*DOP/3.0
               WTO(I+NJ+1)=WTO(I+NJ+1)+8.0*DELQUAD*DOP/3.0
               WTO(I+NJ+2)=WTO(I+NJ+2)+2.0*DELQUAD*DOP/3.0
150         CONTINUE
            NJ=NJ+NQUAD
160      CONTINUE
         ITPTRHE2=NJ+1
         N1=NTRHE2
         DO 190 IL=1,N1
            J=UPPERHE2(IL)
            I=LOWERHE2(IL)
            IT=ITRHE2(I,J)
            IF (IT.EQ.0)THEN
               WRITE (6,*)'HE2 LINE ',I,' TO ',J,' IS FORBIDDEN.'
               GO TO 190
            ENDIF
            FCON=FRQHE2(I)-FRQHE2(J)
            DOP=SRT*FCON*DOPCOF*0.5
            IF (FCON.LE.0)THEN
               WRITE (6,*) 'HE2 IONIZATION FREQUENCY ERROR FOR LINE ',
     :                     I,',',J
               FCON=-FCON
            ENDIF
            IF (FPRINT)WRITE (6,*)' HE2 ',I,' TO ',J,': FREQ = ',FCON,
     :                            ' DOP = ',DOP
            DO 170 II=1,NQUAD
               FREQ(II+NJ)=FCON
               UPHE2(II+NJ)=J
               LOWHE2(II+NJ)=I
170         CONTINUE
            DO 180 I=1,NQUAD-2,2
               WTO(I+NJ)=WTO(I+NJ)+2.0*DELQUAD*DOP/3.0
               WTO(I+NJ+1)=WTO(I+NJ+1)+8.0*DELQUAD*DOP/3.0
               WTO(I+NJ+2)=WTO(I+NJ+2)+2.0*DELQUAD*DOP/3.0
180         CONTINUE
            NJ=NJ+NQUAD
190      CONTINUE
      ENDIF
      IF (NJ.GT.MNJ)THEN
         WRITE (6,*) 'TOO MANY FREQUENCY POINTS'
         CALL EXIT(1)
      ENDIF
      WRITE (6,*)' FREQUENCY POINTS = ',NJ
      IF (FPRINT)THEN
         WRITE (6,*)
         WRITE (6,*)' FREQUENCIES AND WEIGHTS'
         WRITE (6,1007)(FREQ(IJ),WTO(IJ),IJ=1,NJ)
      ENDIF
      DO 200 IJ=1,NJ
         WT(IJ)=WTO(IJ)
200   CONTINUE
      DO 240 IJ=1,NJ
         FRQ3=FREQ(IJ)**3
```

```
         V1=2.815E29/FRQ3
         DO 210 IL=1,NLHE2S
            SIGHE2(IL,IJ)=0.
            IF (FREQ(IJ).GT.FRQHE2(IL))
     :          SIGHE2(IL,IJ)=16.*V1*GAUNT(IL,FREQ(IJ)/4.0)/FLOAT(IL)**5
210      CONTINUE
         DO 220 IL=1,NLHE1S
            SIGHE1(IL,IJ)=0.
            IF (FREQ(IJ).GT.FRQHE1(IL))
     :          SIGHE1(IL,IJ)=V1*HEGAUNT(IL,FREQ(IJ))/FLOAT(QN(IL))**5
220      CONTINUE
         DO 230 IL=1,NLHS
            SIG(IL,IJ)=0.
            IF (FREQ(IJ).GT.FRQH(IL))
     :          SIG(IL,IJ)=V1*GAUNT(IL,FREQ(IJ))/FLOAT(IL)**5
230      CONTINUE
         SIG(NLHS+1,IJ)=3.69E8/FRQ3
240   CONTINUE
      DO 250 I=1,NJ
         FF(I,1)=MIN(FRQH(NLHS+1),FREQ(I))
         FF(I,2)=MIN(FRQHE1(NLHE1S+1),FREQ(I))
         FF(I,3)=MIN(FRQHE2(NLHE2S+1),FREQ(I))
250   CONTINUE
      DO 290 ID=1,NDEPTH
         IF (FLTE)THEN
            CALL NUPOP(ID)
         ELSE
            CALL PARTI(ID)
            DO 260 IL=1,NLHS
               NS(IL,ID)=NE(ID)*NPROT(ID)*SB(IL,ID)
260         CONTINUE
            DO 270 IL=1,NLHE2S
               NHE2S(IL,ID)=NE(ID)*NHE3(ID)*SBHE2(IL,ID)
270         CONTINUE
            DO 280 IL=1,NLHE1S
               NHE1S(IL,ID)=NE(ID)*NHE2(1,ID)*SBHE1(IL,ID)
280         CONTINUE
            NM(ID)=(NTOT(ID)-NE(ID))*MU1
         ENDIF
290   CONTINUE
      RETURN
C
1001  FORMAT (A80)
1002  FORMAT (2F9.0,F8.2,F8.3,F8.5,4I5)
1003  FORMAT (16I5)
1004  FORMAT (' EFFECTIVE TEMPERATURE = ',F10.3/' GRAVITY= ',E10.3,/
     :         ' HELIUM ABUNDANCE=',F7.4/' SURFACE FLUX=',E13.5)
1005  FORMAT (8F10.6)
1006  FORMAT (5E15.7)
1007  FORMAT (4E15.7)
1008  FORMAT (2I5)
      END

      BLOCK DATA TABLES
C
C     CONTAINS ALL DATA STATEMENTS FOR COMMON BLOCKS, IN ACCORDANCE
C     WITH THE ANSI STANDARD
C
      COMA
      COMAI
C
      DATA GHE1/1.,3.,1.,9.,3.,3.,1.,9.,5.,15.,3.,3.,1.,9.,5.,15.,3.,
     : 21.,7.,100.,144.,196.,256.,324.,400.,484.,576.,676.,784.,900.,
     : 1024./
C
      DATA GHE2/2.,8.,18.,32.,50.,72.,98.,128.,162.,200.,242.,288.,338.,
     : 392.,450.,512.,578.,648.,722.,800.,882.,968.,1058.,1152.,1250.,
     : 1352.,1458.,1568.,1682.,1800.,1922.,2048./
C
      DATA OSCH/4.162E-1,7.910E-2,2.899E-2,1.394E-2, 6.408E-1,1.193E-1,
     :4.467E-2, 8.420E-1,1.506E-1, 1.038/
C
      DATA (ITRH(1,I),I=1,5)/0,1,2,3,4/
      DATA (ITRH(2,I),I=1,5)/1,0,5,6,7/
      DATA (ITRH(3,I),I=1,5)/2,5,0,8,9/
      DATA (ITRH(4,I),I=1,5)/3,6,8,0,10/
      DATA (ITRH(5,I),I=1,5)/4,7,9,10,0/
C
      DATA OSCHE1/.2762,.0734,.0302, .5391,.06446,.0231, .3764,.1514,
     :.0507, .0693,.6090,.0118,.1250, .0480,.7110,.00834,.1220, .8960,
     :.0429, .6290,.1400, .1110,.1450,.4820, .0139,.00858,1.0100,
```

```
     : .0205,1.0200, .1030,.6470, 1.2100, .8530, .2000/
C
      DATA (ITRHE1(1,I),I=1,19)/0,0,0,0,1,0,0,0,0,0,2,0,0,0,0,0,3,0,0/
      DATA (ITRHE1(2,I),I=1,19)/0,0,0,4,0,0,0,5,0,0,0,0,0,6,0,0,0,0,0/
      DATA (ITRHE1(3,I),I=1,19)/0,0,0,0,7,0,0,0,0,0,8,0,0,0,0,0,9,0,0/
      DATA (ITRHE1(4,I),I=1,19)/0,4,0,0,0,10,0,0,0,11,0,12,0,0,0,13,
     : 3*0/
      DATA (ITRHE1(5,I),I=1,19)/1,0,7,0,0,0,14,0,15,0,0,0,16,0,17,
     : 4*0/
      DATA (ITRHE1(6,I),I=1,19)/0,0,0,10,0,0,0,18,0,0,0,0,0,19,5*0/
      DATA (ITRHE1(7,I),I=1,19)/0,0,0,0,14,0,0,0,0,0,20,0,0,0,0,0,21,
     : 2*0/
      DATA (ITRHE1(8,I),I=1,19)/0,2,0,0,0,18,0,0,0,22,0,23,0,0,0,24,
     : 3*0/
      DATA (ITRHE1(9,I),I=1,19)/0,0,0,0,15,0,0,0,0,0,25,0,0,0,0,0,26,
     : 0,27/
      DATA (ITRHE1(10,I),I=1,19)/0,0,0,11,0,0,0,22,0,0,0,0,0,28,0,0,
     : 0,29,0/
      DATA (ITRHE1(11,I),I=1,19)/2,0,8,0,0,0,20,0,25,0,0,0,30,0,31,
     : 4*0/
      DATA (ITRHE1(12,I),I=1,19)/0,0,0,12,0,0,0,23,0,0,0,0,0,32,5*0/
      DATA (ITRHE1(13,I),I=1,19)/0,0,0,0,16,0,0,0,0,0,30,0,0,0,0,0,
     : 33,0,0/
      DATA (ITRHE1(14,I),I=1,19)/0,6,0,0,0,19,0,0,0,28,0,32,0,0,0,34,
     : 0,0,0/
      DATA (ITRHE1(15,I),I=1,19)/0,0,0,0,17,0,0,0,0,0,31,0,0,0,0,0,0,
     : 0,0/
      DATA (ITRHE1(16,I),I=1,19)/0,0,0,13,0,0,0,24,0,0,0,0,0,34,0,0,
     : 0,0,0/
      DATA (ITRHE1(17,I),I=1,19)/0,0,0,0,0,0,21,0,26,0,0,0,33,0,0,
     : 0,0,0,0/
      DATA (ITRHE1(18,I),I=1,19)/0,0,0,0,0,0,0,0,0,29,0,0,0,0,0,0,
     : 0,0,0/
      DATA (ITRHE1(19,I),I=1,19)/0,0,0,0,0,0,0,0,27,0,0,0,0,0,0,0,
     : 0,0,0/
C
      DATA OSCHE2/4.162E-1,7.910E-2,2.899E-2,1.394E-2,7.800E-3,
     :4.814E-3,3.184E-3,2.216E-3,1.605E-3, 6.408E-1,1.193E-1,
     :4.467E-2,2.209E-2,1.271E-2,8.037E-3,5.429E-3,3.851E-3,
     : 8.420E-1,1.506E-1,5.585E-2,2.768E-2,1.604E-2,1.023E-2,
     :6.981E-3, 1.038,.1794,6.551E-2,3.229E-2,1.872E-2,1.195E-2,
     : 1.231,.2070,7.455E-2,3.644E-2,2.102E-2, 1.424,.234,.08315,
     :.04038, 1.616,.2609,.09163, 1.807,.2876, 1.999/
C
      DATA (ITRHE2(1,I),I=1,10)/ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9/
      DATA (ITRHE2(2,I),I=1,10)/ 1, 0,10,11,12,13,14,15,16,17/
      DATA (ITRHE2(3,I),I=1,10)/ 2,10, 0,18,19,20,21,22,23,24/
      DATA (ITRHE2(4,I),I=1,10)/ 3,11,18, 0,25,26,27,28,29,30/
      DATA (ITRHE2(5,I),I=1,10)/ 4,12,19,25, 0,31,32,33,34,35/
      DATA (ITRHE2(6,I),I=1,10)/ 5,13,20,26,31, 0,36,37,38,39/
      DATA (ITRHE2(7,I),I=1,10)/ 6,14,21,27,32,36, 0,40,41,42/
      DATA (ITRHE2(8,I),I=1,10)/ 7,15,22,28,33,37,40, 0,43,44/
      DATA (ITRHE2(9,I),I=1,10)/ 8,16,23,29,34,38,41,43, 0,45/
      DATA (ITRHE2(10,I),I=1,10)/9,17,24,30,35,39,42,44,45, 0/
C
      DATA FRQH/3.28799E15,0.821997E15,0.365332E15,0.205499E15,
     :0.131519E15,0.0913329E15,0.0671018E15,0.0513748E15,
     :0.0405924E15,0.0328799E15,0.0271735E15,0.0228333E15,
     :0.0194556E15,0.0167755E15,0.0146133E15,0.0128437E15/
C
      DATA FRQHE1/5.94520E15,1.15305E15,0.957439E15,0.876230E15,
     :0.811774E15,0.451896E15,0.400142E15,0.381976E15,0.362850E15,
     :0.366032E15,0.362480E15,0.240134E15,0.217774E15,0.212670E15,
     :0.202689E15,0.205704E15,0.202057E15,0.202703E15,0.199689E15,
     :0.131520E15,0.0913331E15,0.0671018E15,0.0513748E15,
     :0.0405924E15,0.0328799E15,0.0271735E15,0.0228333E15,
     :0.0194556E15,0.0167755E15,0.0146133E15,0.0128437E15/
C
      DATA FRQHE2/13.1520E15,3.28799E15,1.46133E15,0.821997E15,
     :0.526078E15,0.365332E15,0.268407E15,0.205499E15,0.162370E15,
     :0.131519E15,0.108694E15,0.0913329E15,0.0778222E15,
     :0.0671018E15,0.0584532E15,0.0513748E15,0.0455085E15,
     :0.0405924E15,0.0364320E15,0.0328799E15,0.0298230E15,
     :0.0271735E15,0.0248619E15,0.0228333E15,0.0210431E15,
     :0.0194556E15,0.0180411E15,0.0167755E15,0.0156385E15,
     :0.0146133E15,0.0136857E15,0.0128437E15/
C
      END

      SUBROUTINE WTSET(LLAM)
C
```

```
C     SET THE INTEGRATION WEIGHTS
C
      IMPLICIT NONE
C
      COMA
      COMW
C
      INTEGER IJ
      REAL LLAM
C
      DO 10 IJ=ITPTRH,NJ
         WT(IJ)=LLAM*WT0(IJ)
10    CONTINUE
      RETURN
      END
```

D. Program HYD

Only those portions of HYD that differ significantly from ANDERS are listed here.

```
C     PROGRAM HYD
C
C     AN ADAPTION OF PORTIONS OF THE MIHALAS et al. (1975) CODE TO THE
C     ANDERSON ALGORITHM FOR THE EFFICIENT SOLUTION OF LARGE NUMBERS OF
C     TRANSFER EQUATIONS IN NON-LTE.
C
C     PARAMETERS:
C
C     MNB        MAXIMUM NUMBER OF FREQUENCY BLOCKS
C     MNDEPTH    MAXIMUM NUMBER OF DEPTH POINTS
C     MNJ        MAXIMUM TOTAL NUMBER OF FREQUENCIES
C     MNJC       MAXIMUM NUMBER OF CONTINUUM FREQUENCY POINTS
C     MNTRH      MAXIMUM NUMBER OF HYDROGEN TRANSITIONS
C     MNTRHE1    MAXIMUM NUMBER OF NEUTRAL HELIUM TRANSITIONS
C     MNTRHE2    MAXIMUM NUMBER OF IONIZED HELIUM TRANSITIONS
C     MQH        MAXIMUM QUANTUM NUMBER IN PARTITION SUMS OF HYDROGEN
C     MQHE1      MAXIMUM QUANTUM NUMBER IN PARTITION SUMS OF HELIUM I
C     NEQN       TOTAL NUMBER OF ATOMIC STATES
C     NQUAD      SPECIFIES NUMBER OF QUADRATURE POINTS ON EACH SIDE
C                   OF PROFILE.
C     NLH        NUMBER OF NON-LTE HYDROGEN LEVELS
C     NLHE1      NUMBER OF NON-LTE HELIUM LEVELS
C     NLHE1S     TOTAL HELIUM LEVELS
C     NLHE2      NUMBER OF NON-LTE IONIZED HELIUM LEVELS
C     NLHE2S     TOTAL IONIZED HELIUM LEVELS
C     NLHS       TOTAL HYDROGEN LEVELS
C
C     ACCOF      SAHA ACTIVITY COEFFICIENT
C     BBCOF      PLANK FUNCTION COEFFICIENT
C     CC         VELOCITY OF LIGHT
C     DELQUAD    FRACTION OF DOPPLER WIDTH PER LINE INTEGRATION INTERVAL
C     DOPCOF     DOPPLER WIDTH COEFFICIENT
C     EMASS      ELECTRON MASS
C     ESU        ELECTRON CHARGE
C     HK         PLANK'S CONSTANT OVER BOLTZMANN'S CONSTANT
C     HP         PLANK'S CONSTANT
C     HYDCOF     HYDROSTATIC EQUATION RADIATIVE COEFFICIENT
C     KB         BOLTZMANN'S CONSTANT
C     MHYD       MASS OF HYDROGEN ATOM
C     PI         PI
C     PIE2MC     CLASSICAL ELECTRON ABSORPTION COEFFICIENT
C     SCOF       RADIATIVE RATE COEFFICIENT
C     SIGE       ELECTRON THOMPSON CROSS-SECTION
C
C     VARIABLES:
C
C     FEXIT      FLAG TO EXIT
C     RULE       FLAG TO INCLUDE VARIOUS OPACITIES
C
C     BLOCK      BLOCK ASSIGNMENTS
C     ITPTRH     POINTS TO HYDROGEN LINE FREQUENCIES
C     ITRH       TRANSITION INDICES FOR HYDROGEN;
C                        I.E. ITRH(L,U) IS TRANSITION INDEX OF
C                        HYDROGEN L LEVEL TO U LEVEL.
C     LOWH       LOWER LEVEL OF DOMINANT HYDROGEN TRANSITION AT THE
C                        SPECIFIED FREQUENCY
C     LOWERH     LOWER LEVEL OF HYDROGEN TRANSITIONS REQUESTED BY USER
C     LOWERHE1   " OF NEUTRAL HELIUM
C     LOWERHE2   " OF IONIZED HELIUM
C     NB         NUMBER OF FREQUENCY BLOCKS
C     NDEPTH     NUMBER OF DEPTH POINTS
C     NITER      NUMBER OF ITERATIONS TO MAKE
C     NJ         NUMBER OF FREQUENCIES
C     NTRH       NUMBER OF HYDROGEN TRANSITIONS
C     NTRHE1 " OF NEUTRAL HELIUM TRANSITIONS
C     NTRHE2 " OF IONIZED HELIUM TRANSITIONS
C     UPH        UPPER LEVEL OF DOMINANT HYDROGEN TRANSITION AT THE
C                        FREQUENCY SPECIFIED.
C     UPPERH     UPPER HYDROGEN LEVELS OF TRANSITIONS REQUESTED BY USER
C     UPPERHE1   " OF NEUTRAL HELIUM
C     UPPERHE2   " OF IONIZED HELIUM
C
C     A          A MATRIX OF LINEARIZATION
C     AN         LHS OF POPULATION EQUATIONS
C     ANS        RESULT OF POPULATION CALCULATION
C     B          B MATRIX OF LINEARIZATION
C     BN         RHS OF POPULATION EQUATIONS
C     C          C-MATRIX OF LINEARIZATION
C     CHI        OPACITY MATRIX
```

```
C     CHIOV      MATRIX OF NON-HYDROGEN OPACITY
C     CR         COLLISION RATES FOR HYDROGEN
C     ETA        EMISSIVITY MATRIX
C     ETAOV      MATRIX OF NON-HYDROGEN EMISSIVITY
C     FF         FREE-FREE CUTOFF (TO ACCOUNT FOR UPPER STATES)
C     FH         EDDINGTON FACTOR FOR FLUX
C     FK         EDDINGTON FACTOR FOR RADIATIVE PRESSURE
C     FREQ       FREQUENCY GRID
C     FRQH       IONIZATION FREQUENCIES OF HYDROGEN
C     FRQHE1     " OF NEUTRAL HELIUM
C     FRQHE2     " OF IONIZED HELIUM
C     GH         STATISTICAL WEIGHTS OF HYDROGENIC LEVELS
C     GHE1       " OF NEUTRAL HELIUM
C     HO         SURFACE FLUX
C     OSCH       HYDROGEN OSCILLATOR STRENGTHS LISTED BY TRANSITION INDEX
C     M          MASS GRID
C     MU1        NUCLEI PER PROTON
C     N          HYDROGEN NUMBER DENSITIES
C     NE         ELECTRON DENSITY
C     NHE1       NEUTRAL HELIUM DENSITIES
C     NHE1S      LTE NEUTRAL HELIUM DENSITIES
C     NHE2       IONIZED HELIUM DENSITIES
C     NHE2S      LTE IONIZED HELIUM DENSITIES
C     NHE3       DOUBLY-IONIZED HELIUM DENSITIES
C     NM         FICTIONAL MASSIVE PARTICLE DENSITY
C     NPROT      PROTON DENSITY
C     NS         LTE HYDROGEN DENSITIES
C     NTOT       TOTAL PARTICLE DENSITY
C     Q          RHS OF LINEARIZATION
C     RAD        MEAN INTENSITY OF RADIATION
C     RH         HYDROGEN RADIATIVE BRACKETS
C     RHDS       SECONDARY RATES OF HYDROGEN DOWN
C     RHUS       SECONDARY RATES OF HYDROGEN UP
C     SIG        HYDROGEN CROSS-SECTIONS PLUS FREE-FREE
C     SIGFRE     FREE-FREE HYDROGEN OPACITY
C     SIGL       BOUND-BOUND CROSS SECTION FOR HYDROGEN
C     SUMH       HYDROGEN UPPER STATE SUM
C     SUMHE      HELIUM UPPER STATE SUMS
C     TEMP       TEMPERATURE
C     TLINE      TEMPERATURE ASSUMED TO DETERMINE LINE FREQUENCY GRID
C     WT         FREQUENCY QUADRATURE WEIGHTS MODIFIED BY SWITCHING
C     WTO        UNMODIFIED FREQUENCY QUADRATURE WEIGHTS
C     Y          RATIO OF HELIUM TO HYDROGEN BY NUMBER
C     ZTOT       PROTONS PER NUCLEUS
C
      CLICHE COMA
C
C     INCLUDES PARAMETERS
C
      INTEGER MNB, MNDEPTH, MNJC, MQH, MQHE1, NLH, NLHE1, NLHE2
      INTEGER NQUAD
      PARAMETER (MNB=40, MNDEPTH=70, MNJC=105, MQH=16, MQHE1=37)
      PARAMETER (NLH=10, NLHE1=31, NLHE2=15, NQUAD=5)
C
      INTEGER MNTRH, MNTRHE1, MNTRHE2
      PARAMETER (MNTRH=10, MNTRHE1=14, MNTRHE2=10)
C
      INTEGER MNJ
      PARAMETER (MNJ=MNJC+(2*NQUAD+1)*MNTRH)
C
      INTEGER NEQN
      PARAMETER (NEQN=NLH+1)
C
      REAL CC, DELQUAD, EMASS, ESU, HP, KB, MHYD, PI
      PARAMETER (CC=2.997925E10, DELQUAD=0.6, EMASS=9.10953E-28)
      PARAMETER (ESU=4.80325E-10, HP=6.62618E-27, KB=1.38066E-16)
      PARAMETER (MHYD=1.67265E-24, PI=3.141592654)
C
      REAL ACCOF, BBCOF, DOPCOF, HK, HYDCOF, PIE2MC, SCOF, SIGE
      PARAMETER (ACCOF=2.074E-16, BBCOF=2.*HP/(CC*CC), DOPCOF=4.286E-7)
      PARAMETER (HK=HP/KB, HYDCOF=4.*PI/(CC*KB))
      PARAMETER (PIE2MC=PI*ESU*ESU/(EMASS*CC), SCOF=4.*PI/HP)
      PARAMETER (SIGE=8.*PI*ESU*ESU*ESU*ESU/
     :               (3.*EMASS*EMASS*CC*CC*CC*CC))
C
      INTEGER BLOCK(MNJ), ITPTRH, NB, NDEPTH, NITER
      INTEGER NJ, NTRH, NTRHE1, NTRHE2
C
      LOGICAL FEXIT, RULE(2)
C
```

```
C     N.B.:  THE SECOND DIMENSION OF THE VARIABLES CHI AND ETA
C     MUST BE THE GREATER OF MNB+3 OR MNDEPTH.
C
      REAL B(MNB,MNB), CHI(MNJ,MNDEPTH), CHIOV(MNJ,MNDEPTH)
      REAL ETA(MNJ,MNDEPTH), ETAOV(MNJ,MNDEPTH)
      REAL FH(MNJ), FK(MNJ,MNDEPTH), FREQ(MNJ), HO, Q(MNB)
      REAL M(MNDEPTH), MU1, N(NLH,MNDEPTH), NE(MNDEPTH)
      REAL NHE1(NLHE1,MNDEPTH), NHE1S(NLHE1,MNDEPTH)
      REAL NHE2(NLHE2,MNDEPTH), NHE2S(NLHE2,MNDEPTH), NHE3(MNDEPTH)
      REAL NM(MNDEPTH), NPROT(MNDEPTH), NS(NLH,MNDEPTH), NTOT(MNDEPTH)
      REAL RAD(MNJ,MNDEPTH), RH(NLH+1,NLH+1), RHDS(115,MNDEPTH)
      REAL RHUS(115,MNDEPTH), SIG(NLH+1,MNJ), SIGL(MNJ,MNDEPTH)
      REAL SIGFRE(MNJ,MNDEPTH)
      REAL SUMH(MNDEPTH), SUMHE(2,MNDEPTH), TEMP(MNDEPTH), TLINE
      REAL WT(MNJ), Y, ZTOT
C
      COMMON //ITPTRH, BLOCK, NB, NDEPTH, NITER, NJ, NTRH, NTRHE1,
     : NTRHE2, FEXIT, RULE, Q, B, CHI, CHIOV, ETA, ETAOV, FH, FK,
     : FREQ, HO, M, MU1, N, NE, NHE1, NHE1S, NHE2, NHE2S, NHE3, NM,
     : NPROT, NS, NTOT, RAD, RH, RHDS, RHUS, SIG, SIGL, SIGFRE,
     : SUMH, SUMHE, TEMP, TLINE, WT, Y, ZTOT
C
      ENDCLICHE

      CLICHE COMAI
C
      INTEGER ITRH(NLH,MQH+1)
      INTEGER LOWERH(MNTRH), LOWERHE1(MNTRHE1), LOWERHE2(MNTRHE2)
      INTEGER LOWH(MNJ), UPH(MNJ)
      INTEGER UPPERH(MNTRH), UPPERHE1(MNTRHE1)
      INTEGER UPPERHE2(MNTRHE2)
      REAL FRQH(MQH), FRQHE1(MQHE1), FRQHE2(2*MQH), GH(MQH)
      REAL GHE1(MQHE1), GHE2(2*MQH), OSCH(105)
      EQUIVALENCE (GHE2(1),GH(1))
C
      COMMON /COMAI/ITRH, LOWERH, LOWERHE1, LOWERHE2, LOWH, UPH,
     : UPPERH, UPPERHE1, UPPERHE2, FRQH, FRQHE1, FRQHE2, GHE1, GHE2,
     : OSCH
C
      ENDCLICHE

      CLICHE COMC
C
      REAL AN(NEQN,NEQN), ANS(NEQN), BN(NEQN), CR(NLH,NLH+1,MNDEPTH)
C
      COMMON /COMC/AN, ANS, BN, CR
C
      ENDCLICHE

      CLICHE COMF
C
      REAL A(MNB,MNB), C(MNB,MNB)
C
      COMMON /COMF/A, C
C
      ENDCLICHE

      CLICHE COMR
C
      REAL A1H(NLH,MQH),A2H(NLH,MQH),A3H(NLH,MQH),A4H(NLH,MQH)
      REAL A5H(NLH,MQH)
      REAL COH(NLH), C1H(NLH), C2H(NLH), C3H(NLH), C4H(NLH),C5H(5)
C
      COMMON /COMR/A1H,A2H,A3H,A4H,A5H,COH,C1H,C2H,C3H,C4H,C5H
      ENDCLICHE

      CLICHE COMT
C
      LOGICAL FSH(105), FSET
C
      COMMON /COMT/FSH, FSET
C
      ENDCLICHE

      PROGRAM HYD
C
C     ENTRY POINT
C
      IMPLICIT NONE
C
```

```
C     MACROS
C
      COMA
C
C     LOCAL VARIABLES
C
      CHARACTER USER*6, ACC*6, DROP*8, SUFFIX*1
      INTEGER LENGTH
C
C     EXTERNAL PROCEDURES
C
      REAL BLOCKS, CONTROL, CREATE, DESTROY, EXIT, LINK, PUTOUT, SETUP
      REAL USERINFO, OPEN
C
      EXTERNAL BLOCKS, CONTROL, CREATE, DESTROY, EXIT, LINK, PUTOUT
      EXTERNAL SETUP, USERINFO, OPEN
C
C     START OF EXECUTABLE STATEMENTS.
C
C     THE FILE input CONTAINS A FIRST-APPROXIMATION MODEL.
C     THE FILE output CONTAINS THE MODEL HEREIN CALCULATED.
C     THE FILE monitor CONTAINS ALL OTHER OUTPUT.
C
      CALL LINK("UNIT5=(input,OPEN,TEXT),UNIT12=(output,CREATE,TEXT),
     : UNIT6=(monitor,CREATE,TEXT)//")
C
C     SET UP STARK TABLES
C
      LENGTH=50*21*4*105
      CALL OPEN(10,'stark.t',4,LENGTH)
C
C     READ IN THE FIRST APPROXIMATION AND SET UP EVERYTHING
C     PREPARATORY TO BEGINNING CALCULATIONS.
C
      CALL SETUP
C
C     GET USER SUFFIX (SO THAT SCRATCH FILES CAN BE UNIQUELY NAMED)
C
      CALL USERINFO(USER,ACC,DROP,SUFFIX)
C
C     CREATE SCRATCH FILES
C
      LENGTH=MNB*(MNB+1)*(NDEPTH-1)
      CALL CREATE(8,'%scr8'//SUFFIX,4,LENGTH)
      LENGTH=MNB*NDEPTH
      CALL CREATE(9,'%scr9'//SUFFIX,4,LENGTH)
C
C     SET UP FREQUENCY BINNING.
C
      CALL BLOCKS
C
C     ENTER MAIN CONTROL ROUTINE AND CARRY OUT THE CALCULATIONS.
C
      CALL CONTROL
C
C     WRITE THE RESULTS.
C
      CALL PUTOUT
C
C     DELETE SCRATCH FILES AND EXIT.
C
      CALL DESTROY('%scr8'//SUFFIX)
      CALL DESTROY('%scr9'//SUFFIX)
      CALL EXIT(0)
      END


      REAL FUNCTION ASY(NL,NU,ALPHA,TEMP,DEN)
C
      IMPLICIT NONE
C
C     CALCULATES QUASI-STATIC STARK BROADENING OF HYDROGEN.
C     THE DOPPLER CONVOLUTION IS IGNORED; THIS FUNCTION IS VALID
C     ONLY FOR THE WINGS OF THE LINE.
C
      INTEGER I, J, K, NL, NSC(10,9), NU
C
      REAL ALAM, ALPHA, B, CKMAX, CKMIN, CORE, DEN, DNU, DOP, FO, FAC
      REAL FAC1, SHIELD, TEMP, X1, X2, X3, Y1, Y2, Y3
      REAL CK(127,10,9), FF(127,10,9), FIELD(301), W(300,5), XX(300)
      REAL XY(5), XZ(5), YARR(301)
```

```
C
C     STARK COMPONENT TABLES
C
```

(1236 lines of DATA statements omitted)

```
C
      REAL WFLD
      EXTERNAL WFLD
C
      COMMON /MICRO/W, SHIELD
C
C     START OF EXECUTABLE STATEMENTS
C
      SHIELD=0.0898*DEN**(1./6.)/SQRT(TEMP)
      ALAM=1.E8/(109678.758*((1./NL)**2-(1./NU)**2))
      FAC=0.
      DO 10 K=2,NSC(NU,NL)
         FAC=FAC+FF(K,NU,NL)*WFLD(ALPHA/CK(K,NU,NL))/CK(K,NU,NL)
10    CONTINUE
      ASY=FAC
      RETURN
      END


      REAL FUNCTION HEIIPROF(L,U,DNU,T,DEN)
C
C     CALCULATE HELIUM II QUASISTATIC PROFILE
C
      IMPLICIT NONE
C
C     MACROS
C
      COMA
      COMAI
C
C     LOCAL VARIABLES
C
      INTEGER I, IT, L, U
      REAL CORE, DADNU, DEN, DNU, FO, FRQO, STARK, STRENGTH, T, VO
C
      INTEGER ITRHE2(8,10)
      REAL OSCHE2(44)
C
      DATA OSCHE2/
     : 4.162E-1, 7.910E-2, 2.899E-2, 1.394E-2, 7.800E-3,
     : 4.814E-3, 3.184E-3, 2.216E-3, 1.605E-3, 6.408E-1,
     : 1.193E-1, 4.467E-2, 2.209E-2, 1.271E-2, 8.037E-3,
     : 5.429E-3, 3.851E-3, 8.420E-1, 1.506E-1, 5.585E-2,
     : 2.768E-2, 1.604E-2, 1.023E-2, 6.981E-3, 1.038E+0,
     : 1.794E-1, 6.551E-2, 3.229E-2, 1.872E-2, 1.195E-2,
     : 1.231E+0, 2.070E-1, 7.455E-2, 3.644E-2, 2.102E-2,
     : 1.424E+0, 2.340E-1, 8.315E-2, 4.038E-2, 1.616E+0,
     : 2.609E-1, 9.163E-2, 1.807E+0, 2.876E-1/
C
      DATA (ITRHE2(1,I),I=1,10)/ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9/
      DATA (ITRHE2(2,I),I=1,10)/ 1, 0,10,11,12,13,14,15,16,17/
      DATA (ITRHE2(3,I),I=1,10)/ 2,10, 0,18,19,20,21,22,23,24/
      DATA (ITRHE2(4,I),I=1,10)/ 3,11,18, 0,25,26,27,28,29,30/
      DATA (ITRHE2(5,I),I=1,10)/ 4,12,19,25, 0,31,32,33,34,35/
      DATA (ITRHE2(6,I),I=1,10)/ 5,13,20,26,31, 0,36,37,38,39/
      DATA (ITRHE2(7,I),I=1,10)/ 6,14,21,27,32,36, 0,40,41,42/
      DATA (ITRHE2(8,I),I=1,10)/ 7,15,22,28,33,37,40, 0,43,44/
C
C     EXTERNALS
C
      REAL ASY
      EXTERNAL ASY
C
C     START OF EXECUTABLE STATEMENTS
C
      IT=ITRHE2(L,U)
      STRENGTH=OSCHE2(IT)*PIE2MC
      FRQO=FRQHE2(L)-FRQHE2(U)
      FO=1.25E-9*DEN**(2./3.)
C
C     DOPPLER CORE
C
      VO=DOPCOF*FRQO*SQRT(T)*0.5
      CORE=EXP(-(DNU/VO)**2)/VO/1.7724539
C
C     HYDROGENIC STARK PROFILE
C
```

```
      STARK=0.
      IF (DNU.GT.VO)THEN
         DADNU=3.2E9*CC/F0/FRQ0/FRQ0
         STARK=DADNU*ASY(L,U,ABS(DNU-FRQ0*4.03E-4)*DADNU,T,DEN)
      ENDIF
C
C     NOW DECIDE WHICH TO USE:  WE ASSUME DOPPLER PROFILE FOR
C     DNU < VO, AND TAKE THE GREATER OF THE DOPPLER OR STARK PROFILES
C     FOR DNU> VO
C
      HEIIPROF=STRENGTH*MAX(STARK,CORE)
      RETURN
      END


      REAL FUNCTION HPROFILE(L,U,DNU,T,NEL)
C
C     CALCULATE H LINE PROFILES
C
C     HYDROGEN LINES 1-2 TO 2-5 ARE TREATED USING THE FULL UNIFIED
C     STARK THEORY WITH LOWER STATE INTERACTIONS AS OUTLINED BY VIDAL,
C     COOPER, AND SMITH (1973).
C
C     ALL OTHER LINES ARE TREATED APPROXIMATELY USING THE QUASI-STATIC
C     STARK THEORY.
C
      IMPLICIT NONE
C
C     MACRO
C
      COMA
      COMAI
C
C     LOCAL VARIABLES
C
      INTEGER I, J, K, IAO, INO, IT, L, U, TRCUR
      REAL ALPHA, CORE, DADNU, LALPHA, DNU, F0, FRQ0, LNEL, LT, NEL
      REAL STARK, STRENGTH, T, VO
      REAL SPROF(50,21,4), SPI(4), SPN(4), SPT(4), XT(4), XNE(21)
      REAL XALPHA(50)
C
C     EXTERNALS
C
      REAL POLY, ASY, RDABS
      INTEGER IOSTATUS
      EXTERNAL POLY, ASY, RDABS, IOSTATUS
C
C     PROFILE DATA
C
C     LOG TEMPERATURES
C
      DATA XT/4.0, 4.30103, 4.60206, 4.90309/
C
C     LOG ELECTRON NUMBER DENSITIES
C
      DATA XNE/
     : 8.0,  8.5,  9.0,  9.5, 10.0, 10.5, 11.0, 11.5, 12.0, 12.5,
     :13.0, 13.5, 14.0, 14.5, 15.0, 15.5, 16.0, 16.5, 17.0, 17.5,
     :18.0/
C
C     LOG ALPHA (=WAVELENGTH OVER NORMALIZED FIELD STRENGTH)
C
      DATA XALPHA/
     :-5.8, -5.6, -5.4, -5.2, -5.0, -4.8, -4.6, -4.4, -4.2, -4.0,
     :-3.8, -3.6, -3.4, -3.2, -3.0, -2.8, -2.6, -2.4, -2.2, -2.0,
     :-1.8, -1.6, -1.4, -1.2, -1.0, -0.8, -0.6, -0.4, -0.2,  0.0,
     : 0.2,  0.4,  0.6,  0.8,  1.0,  1.2,  1.4,  1.6,  1.8,  2.0,
     : 2.2,  2.4,  2.6,  2.8,  3.0,  3.2,  3.4,  3.6,  3.8,  4.0/
C
C     TRANSITION  1 TO  2 10000.  8.0 (INITIAL TABLE IN MEMORY)
C
```

(1061 lines of DATA statements omitted.)

```
C
C     INDICATE THAT THE DATA FOR THE FIRST TRANSITION ARE
C     ALREADY IN THE ARRAY SPROF.  OTHER TRANSITION TABLES
C     ARE READ INTO MEMORY AS REQUIRED.
C
      DATA TRCUR/1/
C
C     BE SURE THE PROGRAM SAVES TRCUR AND SPROF.  THIS IS A NO-OP
C     FOR MOST COMPILERS, SINCE VERY FEW FORTRAN COMPILERS USE
```

```
C     ANYTHING BUT STATIC MEMORY FOR VARIABLES.
C
      SAVE
C
C     CALCULATE TOTAL LINE OPACITY AND FREQUENCY
C
      IT=ITRH(L,U)
C
C     IF NECESSARY, READ IN THE TABLES FOR THE NEXT TRANSITION.
C
      IF (IT.NE.TRCUR)THEN
         TRCUR=IT
         I=50*21*4
         J=(IT-2)*I
         CALL RDABS(10,SPROF,I,J)
         I=IOSTATUS(10,J)
      ENDIF
      STRENGTH=OSCH(IT)*PIE2MC
      FRQ0=FRQH(L)-FRQH(U)
C
C     CALCULATE NORMALIZED FIELD STRENGTH AND CONVERSION
C     FACTOR FROM FREQUENCY TO THE DIMENSIONLESS QUANTITY ALPHA.
C
      F0=1.25E-9*NEL**(2./3.)
      DADNU=1.E8*CC/F0/FRQ0/FRQ0
      ALPHA=ABS(DNU*DADNU)
C
C     PREPARE TO INTERPOLATE FROM TABLE
C
      LT=LOG10(T)
      LNEL=LOG10(NEL)
      LALPHA=LOG10(MAX(1.E-6,ALPHA))
C
C     CHECK INPUT QUANTITIES TO SEE IF THEY ARE ON THE TABLE
C     GRID.  IF NOT, FOLLOW APPROXIMATE PROCEDURES FOR EACH CASE.
C
      IF (LT.LT.XT(1))THEN
C
C     TEMPERATURE OFF BOTTOM OF SCALE:  THIS SHOULD NEVER HAPPEN.
C
         WRITE (6,*)'TEMPERATURE TOO LOW: U=',U,' L=',L,' T=',T
         STOP
      ELSE IF (LT.GT.XT(4).OR.LNEL.GT.XNE(21))THEN
C
C     TEMPERATURE OR ELECTRON DENSITY OFF THE TOP OF THE SCALE;
C     THIS HAPPENS ONLY AT GREAT OPTICAL DEPTH, THUS AN ACCURATE
C     TREATMENT IS OF NO IMPORTANCE.  WE USE THE QUASI-STATIC
C     PROFILE WITHOUT DOPPLER CONVOLUTION; THIS GIVES ACCURATE
C     WINGS, WHICH ARE THE ONLY PART OF THE PROFILE THAT MIGHT
C     POSSIBLY BE IMPORTANT AT SUCH GREAT DEPTHS.
C
         STARK=ASY(L,U,ALPHA,T,NEL)
         HPROFILE=STRENGTH*STARK*DADNU
      ELSE IF (LNEL.LT.XNE(1))THEN
C
C     VERY LOW ELECTRON DENSITY; WE USE THE DOPPLER PROFILE,
C     SINCE WINGS WILL BE UNIMPORTANT AT SUCH LOW DENSITY.
C
         V0=DOPCOF*FRQ0*SQRT(T)
         HPROFILE=STRENGTH*EXP(-(DNU/V0)**2)/V0/1.7724539
      ELSE IF (LALPHA.GT.XALPHA(50))THEN
C
C     'WAY OUT IN THE WINGS.  WE TAKE THE GREATER OF THE
C      DOPPLER PROFILE OR THE STARK PROFILE.
C
         V0=DOPCOF*FRQ0*SQRT(T)
         CORE=EXP(-(DNU/V0)**2)/V0/1.7724539
         STARK=DADNU*ASY(L,U,ALPHA,T,NEL)
         HPROFILE=STRENGTH*MAX(CORE,STARK)
      ELSE
C
C     TEMPERATURE AND ELECTRON NUMBER ARE BOTH ON THE SCALE.
C
C     FIRST CHECK AND SEE IF ALL THE GRID POINTS REQUIRED ARE THERE.
C
         IN0=MAX(1,MIN(18,INT(2*(LNEL-8.0))))
         IA0=MAX(1,MIN(47,INT(5*(LALPHA+5.8))))
         DO 20 I=1,4
            DO 10 J=IN0,IN0+3
               IF (SPROF(1,J,I).EQ.(0.))GO TO 60
```

```
10          CONTINUE
20       CONTINUE
         DO 50 I=1,4
            DO 40 J=1,4
               DO 30 K=1,4
                  SPN(K)=SPROF(IAO+I-1,INO+K-1,J)
30             CONTINUE
               SPT(J)=POLY(4,XNE(INO),SPN,LNEL)
40          CONTINUE
            SPI(I)=POLY(4,XT,SPT,LT)
50       CONTINUE
         STARK=POLY(4,XALPHA(IAO),SPI,LALPHA)
         HPROFILE=STRENGTH*EXP(2.302585093*STARK)*DADNU
      ENDIF
      RETURN
60    HPROFILE=STRENGTH*DADNU*ASY(L,U,ALPHA,T,NEL)
      END


      BLOCK DATA MFIELD
C
C     MICROFIELD DISTRIBUTION
C
      REAL W(300,5)
      COMMON /MICRO/W, SHIELD
C
```
(659 lines of DATA statements omitted)
```
C
      END


      REAL FUNCTION POLY(NN,XT,YT,X)
C
C     NN-PT POLYNOMIAL INTERPOLATION
C
      REAL XT(NN),YT(NN)
C
      SUM=0.
      DO 20 I=1,NN
         TERM=YT(I)
         DO 10 J=1,NN
            IF (I.NE.J)THEN
               TERM=TERM*(X-XT(J))/(XT(I)-XT(J))
            ENDIF
10       CONTINUE
         SUM=SUM+TERM
20    CONTINUE
      POLY=SUM
      RETURN
      END


      BLOCK DATA COMTT
C
C     DATA STATEMENTS FOR COMMON BLOCK COMT
C
      IMPLICIT NONE
C
C     MACROS
C
      COMT
C
      DATA FSET/.FALSE./
      DATA FSH/105*.TRUE./
C
      END


      SUBROUTINE TRIDAG(A,N,NR)
C
C     INVERT TRIDIAGONAL MATRIX IN PLACE
C
      REAL A(NR,NR)
C
      A(1,2)=A(1,2)/A(1,1)
      A(1,1)=1./A(1,1)
      DO 30 I=2,N-1
         A(I,I)=A(I,I)-A(I,I-1)*A(I-1,I)
         DO 10 J=1,I-1
```

```
            A(I,J)=-A(I,I-1)*A(I-1,J)
10       CONTINUE
         DIV=A(I,I)
         DO 20 J=1,I-1
            A(I,J)=A(I,J)/DIV
20       CONTINUE
         A(I,I)=1./DIV
         A(I,I+1)=A(I,I+1)/DIV
30    CONTINUE
      A(N,N)=A(N,N)-A(N,N-1)*A(N-1,N)
      DO 40 J=1,N-1
         A(N,J)=-A(N,N-1)*A(N-1,J)/A(N,N)
40    CONTINUE
      A(N,N)=1./A(N,N)
      DO 70 I=N-1,1,-1
         DIV=A(I,I+1)
         DO 50 J=1,I
            A(I,J)=A(I,J)-DIV*A(I+1,J)
50       CONTINUE
         DO 60 J=I+1,N
            A(I,J)=-DIV*A(I+1,J)
60       CONTINUE
70    CONTINUE
      RETURN
      END

      SUBROUTINE TWOATM
C
C     ESTIMATES RATE BRACKETS FOR A LINE TRANSITION USING AN EQUIVALENT
C     TWO-LEVEL ATOM APPROACH.  ONLY THE RATE COEFFICIENTS ARE ACTUALLY
C     SAVED FOR USE IN THE MAIN CALCULATION.
C
C     THIS ROUTINE IS CALLED ONCE ONLY TO MAKE AN INITIAL ESTIMATE OF
C     THE RELEVANT RATES.  THIS IS DONE AFTER THE RATES FOR EXPLICIT
C     TRANSITIONS ARE CALCULATED FROM THE INPUT MODEL, WHICH SHOULD BE
C     A NON-LTE MODEL ATMOSPHERE WITH THE MOST IMPORTANT LINES ALREADY
C     REPRESENTED.
C
      IMPLICIT NONE
C
C     MACROS
C
      COMA
      COMAI
      COMC
      COMT
C
C     LOCAL VARIABLES AND PARAMETERS
C
      INTEGER NMU
      PARAMETER (NMU=3)
C
      INTEGER I, IMU, J, ID, IJ, IT, ITT, L, LL, U, UU
      REAL A1(MNDEPTH), A2(MNDEPTH), A3(MNDEPTH), A4(MNDEPTH)
      REAL CHIC(MNDEPTH), CHIL(MNDEPTH), ETAC(MNDEPTH), ETAL(MNDEPTH)
      REAL FM(MNDEPTH), MAT(MNDEPTH,MNDEPTH), MAU(MNDEPTH)
      REAL MAK(MNDEPTH), MAV(MNDEPTH), DT(MNDEPTH), MAQ(MNDEPTH)
      REAL MAW(MNDEPTH,MNDEPTH), A5(MNDEPTH,MNDEPTH), MEANJ(MNDEPTH)
      REAL BB, CHIT, DOP, DTC, EX, FN
      REAL FRQO, MAVSUM(MNDEPTH), SIGMA(NQUAD,MNDEPTH)
      REAL VX, VY, VZ
C
      REAL MU(NMU),WTMU(NMU)
      DATA WTMU/.2777777777777778,.44444444444444,
     : .2777777777777778/,MU/.887298334620742,.5,.112701665379258/
C
      REAL FWT(0:NQUAD-1)
      DATA FWT/.6666667,2.6666667,1.3333333,2.6666667,.6666667/
C
      SAVE
C
C     EXTERNALS
C
      REAL TEDDFAC, HPROFILE, LINSLV, TGENER, TRIDAG
      EXTERNAL TEDDFAC, HPROFILE, LINSLV, TGENER, TRIDAG
C
C     FUNCTIONS
C
      REAL SB, SBHE1, SBHE2
      SB(I,ID)=ACCOF*EXP(HK*FRQH(I)/TEMP(ID))*GH(I)/TEMP(ID)/
```

```
     :          SQRT(TEMP(ID))
      SBHE1(I,ID)=ACCOF*EXP(HK*FRQHE1(I)/TEMP(ID))*GHE1(I)/
     :              TEMP(ID)/SQRT(TEMP(ID))/2.
      SBHE2(I,ID)=ACCOF*EXP(HK*FRQHE2(I)/TEMP(ID))*GHE2(I)/
     :         TEMP(ID)/SQRT(TEMP(ID))
C
C     START OF EXECUTABLE STATEMENTS
C
C     IF NOT ALREADY DONE (I.E. THIS IS FIRST ITERATION) SET UP FS'S
C
      IF (.NOT.FSET)THEN
         DO 10 I=1,NTRH
            L=LOWERH(I)
            U=UPPERH(I)
            IT=ITRH(L,U)
            FSH(IT)=.FALSE.
10       CONTINUE
      ENDIF
C
C     BEGIN ESTIMATING RATES OF SECONDARY TRANSITIONS INVOLVING
C     UPPERMOST LEVELS (N>NLH), WHICH ARE ADDED TO THE CONTINUUM
C     RATES.
C
      DO 350 LL=1,NLH-1
         DO 340  UU=LL+1,NLH
            L=LL
            U=UU
            IT=ITRH(L,U)
            IF (FSH(IT))THEN
               FRQO=FRQH(L)-FRQH(U)
               IF (FRQO.LT.0)THEN
                  FRQO=ABS(FRQO)
                  L=UU
                  U=LL
               ENDIF
C
C     CALCULATE LINE PROFILE
C
               DOP=SQRT(TLINE)*FRQO*DOPCOF
               DO 30 IJ=1,NQUAD
                  DO 20 ID=1,NDEPTH
                     SIGMA(IJ,ID)=HPROFILE(L,U,DOP*DELQUAD*(IJ-1),
     :                                   TEMP(ID),NE(ID))
20                CONTINUE
30             CONTINUE
C
C     ACCUMULATE RATES TO/FROM CONTINUUM
C
               DO 40 ID=1,NDEPTH
                  A1(ID)=RHUS(ITRH(L,MQH+1),ID)+NE(ID)*CR(L,NLH+1,ID)
                  A2(ID)=NS(L,ID)*(RHDS(ITRH(L,MQH+1),ID)+NE(ID)*
     :                   CR(L,NLH+1,ID))
                  A3(ID)=RHUS(ITRH(U,MQH+1),ID)+NE(ID)*CR(U,NLH+1,ID)
                  A4(ID)=NS(U,ID)*(RHDS(ITRH(U,MQH+1),ID)+NE(ID)*
     :                   CR(U,NLH+1,ID))
40             CONTINUE
C
C     ACCUMULATE RATES TO/FROM LEVELS OTHER THAN L AND U
C
               DO 70 I=1,L-1
                  DO 50 ID=1,NDEPTH
                     A1(ID)=A1(ID)+NE(ID)*CR(L,I,ID)
                     A2(ID)=A2(ID)+N(I,ID)*NE(ID)*CR(I,L,ID)
50                CONTINUE
                  ITT=ITRH(I,L)
                  IF (ITT.NE.0)THEN
                     DO 60 ID=1,NDEPTH
                        VX=NS(I,ID)/NS(L,ID)
                        A1(ID)=A1(ID)+VX*RHDS(ITT,ID)
                        A2(ID)=A2(ID)+N(I,ID)*RHUS(ITT,ID)
60                   CONTINUE
                  ENDIF
70             CONTINUE
               DO 100 I=L+1,NLH
                  IF (I.NE.U)THEN
                     DO 80 ID=1,NDEPTH
                        A1(ID)=A1(ID)+NE(ID)*CR(L,I,ID)
                        A2(ID)=A2(ID)+N(I,ID)*NE(ID)*CR(I,L,ID)
80                   CONTINUE
                     ITT=ITRH(L,I)
```

```
                  IF (ITT.NE.0)THEN
                     DO 90 ID=1,NDEPTH
                        VX=NS(L,ID)/NS(I,ID)
                        A1(ID)=A1(ID)+RHUS(ITT,ID)
                        A2(ID)=A2(ID)+N(I,ID)*VX*RHDS(ITT,ID)
90                   CONTINUE
                  ENDIF
               ENDIF
100         CONTINUE
            DO 130 I=1,U-1
               IF (I.NE.L)THEN
                  DO 110 ID=1,NDEPTH
                     A3(ID)=A3(ID)+NE(ID)*CR(U,I,ID)
                     A4(ID)=A4(ID)+N(I,ID)*NE(ID)*CR(I,U,ID)
110               CONTINUE
                  ITT=ITRH(I,U)
                  IF (ITT.NE.0)THEN
                     DO 120 ID=1,NDEPTH
                        VX=NS(I,ID)/NS(U,ID)
                        A3(ID)=A3(ID)+VX*RHDS(ITT,ID)
                        A4(ID)=A4(ID)+N(I,ID)*RHUS(ITT,ID)
120                  CONTINUE
                  ENDIF
               ENDIF
130         CONTINUE
            DO 160 I=U+1,NLH
               DO 140 ID=1,NDEPTH
                  A3(ID)=A3(ID)+NE(ID)*CR(U,I,ID)
                  A4(ID)=A4(ID)+N(I,ID)*NE(ID)*CR(I,U,ID)
140            CONTINUE
               ITT=ITRH(U,I)
               IF (ITT.NE.0)THEN
                  DO 150 ID=1,NDEPTH
                     VX=NS(U,ID)/NS(I,ID)
                     A3(ID)=A3(ID)+RHUS(ITT,ID)
                     A4(ID)=A4(ID)+N(I,ID)*VX*RHDS(ITT,ID)
150               CONTINUE
               ENDIF
160         CONTINUE
C
C     INTEGRATE PROFILE
C
            DO 180 ID=1,NDEPTH
               MAVSUM(ID)=0.
               DO 170 IJ=1,NQUAD
                  MAVSUM(ID)=MAVSUM(ID)+
     :                     SIGMA(IJ,ID)*DELQUAD*DOP*FWT(IJ-1)
170            CONTINUE
180         CONTINUE
C
C     CALCULATE TERMS REPRESENTING LINE SCATTERING AND LINE THERMAL
C     TERMS.
C
            BB=BBCOF*FRQ0*FRQ0*FRQ0
            DO 190 ID=1,NDEPTH
               EX=EXP(HK*FRQ0/TEMP(ID))
               VX=A2(ID)+A4(ID)
               VY=VX*(NE(ID)*CR(L,U,ID)*EX+SCOF*MAVSUM(ID)*BB/FRQ0-
     :            NE(ID)*CR(L,U,ID))+GH(U)*A2(ID)*A3(ID)/GH(L)-
     :            A1(ID)*A4(ID)
               VZ=(VX*NE(ID)*CR(L,U,ID)+A1(ID)*A4(ID))/VY
               VY=VX*SCOF/FRQ0/VY
               A1(ID)=BB*VY
               A2(ID)=BB*VZ
190         CONTINUE
C
C     A1 NOW CONTAINS SCATTERING LINE SOURCE TERM; A2 IS THERMAL
C     (OR OTHER TRANSITION) SOURCE TERM.
C
C     CALCULATE CONTINUUM OPACITIES FOR THIS TRANSITION.
C
            CALL TGENER(FRQ0,CHIC,ETAC)
C
C     CLEAR RYBICKI MATRICES
C
            DO 210 I=1,NDEPTH
               DO 200 J=1,NDEPTH
                  MAW(I,J)=0.
200            CONTINUE
210         CONTINUE
```

```
              DO 220 I=1,NDEPTH
                 MAW(I,I)=-1.
                 MAQ(I)=0.
220           CONTINUE
              DO 320 IJ=0,NQUAD-1
C
C     CALCULATE FREQUENCY INTEGRAL WEIGHTS AND LINE OPACITY
C     FROM PROFILE CALCULATED EARLIER.
C
                 DO 230 ID=1,NDEPTH
                    MAV(ID)=DELQUAD*DOP*FWT(IJ)*SIGMA(IJ+1,ID)
                    ETAL(ID)=GH(L)*N(U,ID)/GH(U)
                    CHIL(ID)=SIGMA(IJ+1,ID)*(N(L,ID)-ETAL(ID))/
     :                       NM(ID)/MHYD
                    ETAL(ID)=SIGMA(IJ+1,ID)*ETAL(ID)*BB/NM(ID)/MHYD
230              CONTINUE
C
C     GENERATE MOMENT FACTORS
C
                 CALL TEDDFAC(FRQ0,CHIC,ETAC,CHIL,ETAL,FM,FN)
C
C     OPTICAL DEPTHS
C
                 DO 240 ID=1,NDEPTH-1
                    DT(ID)=(M(ID+1)-M(ID))*0.5*(CHIL(ID+1)+CHIL(ID)+
     :                     CHIC(ID+1)+CHIC(ID))
240              CONTINUE
C
C     SET UP SOURCE MATRICES
C
                 MAU(1)=0.
                 MAK(1)=0.
                 DO 250 ID=2,NDEPTH-1
                    CHIT=CHIC(ID)+CHIL(ID)
                    MAU(ID)=CHIL(ID)*A1(ID)/CHIT
                    MAK(ID)=-(ETAC(ID)+A2(ID)*CHIL(ID))/CHIT
250              CONTINUE
                 MAU(NDEPTH)=0.
                 MAK(NDEPTH)=BB/(EXP(HK*FRQ0/TEMP(NDEPTH))-1.)
C
C     SET UP DIFFERENCE OPERATOR MATRIX
C
                 DO 270 I=1,NDEPTH
                    DO 260 J=1,NDEPTH
                       MAT(I,J)=0.
260                 CONTINUE
270              CONTINUE
C
C     SURFACE
C
                 MAT(1,1)=-FM(1)/DT(1)-FN
                 MAT(1,2)=FM(1)/DT(1)
C
C     ORDINARY DEPTH POINT
C
                 DO 280 ID=2,NDEPTH-1
                    DTC=0.5*(DT(ID)+DT(ID-1))
                    MAT(ID,ID-1)=FM(ID-1)/DT(ID-1)/DTC
                    MAT(ID,ID+1)=FM(ID)/DT(ID)/DTC
                    MAT(ID,ID)=-MAT(ID,ID-1)-MAT(ID,ID+1)-1.0+
     :                         NE(ID)*SIGE/(CHIC(ID)+CHIL(ID))/
     :                         NM(ID)/MHYD
280              CONTINUE
C
C     LOWER BOUNDARY CONDITION
C
                 MAT(NDEPTH,NDEPTH-1)=0.
                 MAT(NDEPTH,NDEPTH)=1.0
C
C     NOW ACCUMULATE CONTRIBUTION OF THIS (ANGLE,FREQUENCY) POINT
C
                 CALL TRIDAG(MAT,NDEPTH,MNDEPTH)
                 DO 300 I=1,NDEPTH
                    DO 290 J=1,NDEPTH
                       MAW(J,I)=MAW(J,I)-MAV(J)*MAT(J,I)*
     :                          MAU(I)
                       MAQ(I)=MAQ(I)-MAV(I)*MAT(I,J)*
     :                        MAK(J)
290                 CONTINUE
300              CONTINUE
```

```
320             CONTINUE
C
C       NOW SOLVE FOR MEAN SOURCE FUNCTION
C
                CALL LINSLV(MAW,MAQ,MEANJ,NDEPTH,MNDEPTH)
C
C       NORMALIZE MEANJ USING MAVSUM AND CALCULATE RATES.
C
                DO 330 ID=1,NDEPTH
                   EX=EXP(-FRQ0*HK/TEMP(ID))
                   RHUS(IT,ID)=SCOF*MEANJ(ID)/FRQ0
                   RHDS(IT,ID)=SCOF*(MAVSUM(ID)*BB+MEANJ(ID))*EX/FRQ0
330             CONTINUE
             ENDIF
340       CONTINUE
350    CONTINUE
       FSET=.TRUE.
       RETURN
       END


       FUNCTION WFLD(B)
C
C      INTERPOLATION OF MICROFIELD DISTRIBUTION TABLE
C
       COMMON /MICRO/W(300,5), SHIELD
       REAL SPT(5), XX(5)
       DATA XX/0.0, 0.2, 0.4, 0.6, 0.8/
C
       WFLD=0.0
       IF (B.LE.30.0)GO TO 10
       SBS=1./B/SQRT(B)
       WFLD=((21.6*SBS+7.639)*SBS+1.496)*SBS/B
       RETURN
10     IF (B.LE.0.)RETURN
       J=(B+0.2)*10.0
       L=J-1
       IF (J.GT.2)L=J-2
       IF (J.GT.3)L=J-3
       IF (J.GT.300)L=297
       LLL=L+4
       DO 50 I=1,5
          SPT(I)=0.
          DO 40 K=L,LLL
             AK=K-1
             TERM=W(K,I)
             DO 30 M=L,LLL
                IF (K.NE.M)THEN
                   AM=M-1
                   TERM=TERM*(10.*B-AM)/(AK-AM)
                ENDIF
30           CONTINUE
             SPT(I)=SPT(I)+TERM
40        CONTINUE
50     CONTINUE
       WFLD=POLY(5,XX,SPT,SHIELD)
       RETURN
       END
```

E. Program HE

Only the profile functions for neutral helium have been listed for HE, as all other subroutines are similar to ones already listed.

```
      REAL FUNCTION VOIGT(A,V)
C
C     VOIGT FUNCTION CALCULATION; THIS IS SUFFICIENTLY CLEVER TO HANDLE
C     ANY VALUE OF A, NOT JUST SMALL A.  THIS IS DONE BY MEANS OF AN
C     ASYMPTOTIC EXPANSION FOR LARGE A AND A SERIES EXPANSION FOR SMALL
C     A.
C
      PARAMETER (PI=3.141592654)
C
      REAL TERM(100)
C
      COMMON /CONCOM/A1,V1
C
      REAL CONINT
      EXTERNAL CONINT
C
      X=V/A
      T=0.25/A/A
      DET=0.25*X*X/T
      IF (DET.LT.49.)THEN
C
C     SERIES REGIME
C
C     THE FIRST TERM IN THE SEQUENCE IS A BIT TRICKY;
C     WE MUST USE A SERIES EXPRESSION FOR VERY SMALL T.
C     OTHERWISE THE EXPONENTIAL FACTOR OVERFLOWS AS THE ERFC
C     FACTOR UNDERFLOWS.
C
         IF (T.GT.(.0015))THEN
            S0=0.5*SQRT(PI/T)*EXP(0.25/T)*ERFC(0.5/SQRT(T))
         ELSE
            S0=(1.+T*(-2.+T*(12.+T*(-120.+T*1680.))))
         ENDIF
         S1=S0
         N=0
         FAC=1.
10          N=N+1
            S0=(1.0-S0)*0.5/(2.*N-1)/T
            FAC=FAC*DET/N
            TERM(N)=S0*FAC
            IF (TERM(N).GT.(1.E-8*S1).AND.N.LT.100)GO TO 10
         SUM=0.
         DO 20 I=N,1,-1
            SUM=SUM+TERM(I)
20       CONTINUE
         SUM=SUM+S1
         U0=SUM*EXP(-DET)
      ELSE IF (X.GT.10.)THEN
C
C     ASYMPTOTIC REGIME
C
         S1=-1.
         S0=-1./X
         N=1
         TERM(1)=-S0
         SIGN=-1.
30          N=N+1
            S3=S1
            S2=S0
            S1=S2/X-2.*T*(N-1)*S3/X/X
            N=N+1
            S0=S1/X-2.*T*(N-1)*S2/X/X
            SIGN=-SIGN
            TERM(N)=SIGN*S0
            IF (ABS(S0*X).GT.1.E-8.AND.N.LT.99)GO TO 30
         SUM=0.
         DO 40 I=N,1,-2
            SUM=SUM+TERM(I)
40       CONTINUE
         U0=SUM/X
      ELSE
C
C     INTEGRATION REGIME
C
         A1=A
         V1=V
         CALL QROMB(CONINT,0.,1.,U0)
         U0=U0*A*A/1.7724539
      ENDIF
```

```
      VOIGT=MAX(0.,U0/A/1.7724539)
      RETURN
      END


      REAL FUNCTION CONINT(X)
C
      COMMON /CONCOM/A,V
C
      IF (X.LE.0)THEN
         CONINT=0.
      ELSE
         CONINT=X*(1./((V+LOG(X))**2+A*A)+1./((V-LOG(X))**2+A*A))
      ENDIF
      RETURN
      END


      SUBROUTINE QROMB(FUNC,A,B,SS)
C
      PARAMETER (EPS=1.E-6, JMAX=20, JMAXP=JMAX+1, K=5, KM=K-1)
      REAL S(JMAXP), H(JMAXP)
C
      H(1)=1.
      DO 11 J=1,JMAX
         CALL TRAPZD(FUNC,A,B,S(J),J)
         IF (J.GE.K)THEN
            CALL POLINT(H(J-KM),S(J-KM),K,0.,SS,DSS)
            IF (ABS(DSS).LT.EPS*ABS(SS))RETURN
         ENDIF
         S(J+1)=S(J)
         H(J+1)=0.25*H(J)
11    CONTINUE
      WRITE (6,*)'WARNING--TOO MANY STEPS IN HEIPROF INTG '
      RETURN
      END


      SUBROUTINE POLINT(XA,YA,N,X,Y,DY)
C
      PARAMETER (NMAX=10)
      DIMENSION XA(N),YA(N),C(NMAX),D(NMAX)
C
      NS=1
      DF=ABS(X-XA(1))
      DO 11 I=1,N
         DIFT=ABS(X-XA(I))
         IF (DIFT.LT.DF)THEN
            NS=I
            DF=DIFT
         ENDIF
         C(I)=YA(I)
         D(I)=YA(I)
11    CONTINUE
      Y=YA(NS)
      NS=NS-1
      DO 13 M=1,N-1
         DO 12 I=1,N-M
            HO=XA(I)-X
            HP=XA(I+M)-X
            W=C(I+1)-D(I)
            DEN=HO-HP
            IF (DEN.EQ.0.)STOP 'FATAL ERROR IN HEIPROF INTG'
            DEN=W/DEN
            D(I)=HP*DEN
            C(I)=HO*DEN
12       CONTINUE
         IF (2*NS.LT.N-M)THEN
            DY=C(NS+1)
         ELSE
            DY=D(NS)
            NS=NS-1
         ENDIF
         Y=Y+DY
13    CONTINUE
      RETURN
      END


      FUNCTION ERFC(X)
```

```
C
C     ERROR FUNCTION CALCULATION, FROM "NUMERICAL RECIPES."
C
      ERFC=GAMMQ(.5,X**2)
      RETURN
      END


      FUNCTION GAMMLN(XX)
C
C     LOG OF GAMMA FUNCTION; ALSO FROM "NUMERICAL RECIPES."
C
      REAL*8 COF(6),STP,HALF,ONE,FPF,X,TMP,SER
      DATA COF,STP/76.18009173D0,-86.50532033D0,24.01409822D0,
     *    -1.231739516D0,.120858003D-2,-.536382D-5,2.50662827465D0/
      DATA HALF,ONE,FPF/0.5D0,1.0D0,5.5D0/
      X=XX-ONE
      TMP=X+FPF
      TMP=(X+HALF)*LOG(TMP)-TMP
      SER=ONE
      DO 11 J=1,6
        X=X+ONE
        SER=SER+COF(J)/X
11    CONTINUE
      GAMMLN=TMP+LOG(STP*SER)
      RETURN
      END


      FUNCTION GAMMQ(A,X)
C
C     PARTIAL GAMMA FUNCTION FROM "NUMERICAL RECIPES."
C
      IF(X.LT.0..OR.A.LE.0.)PAUSE
      IF(X.LT.A+1.)THEN
        CALL GSER(GAMSER,A,X,GLN)
        GAMMQ=1.-GAMSER
      ELSE
        CALL GCF(GAMMQ,A,X,GLN)
      ENDIF
      RETURN
      END


      SUBROUTINE GCF(GAMMCF,A,X,GLN)
C
C     CONTINUED FRACTION REPRESENTATION OF PARTIAL GAMMA FUNCTION,
C     FROM "NUMERICAL RECIPES."
C
      PARAMETER (ITMAX=100,EPS=3.E-7)
      GLN=GAMMLN(A)
      GOLD=0.
      A0=1.
      A1=X
      B0=0.
      B1=1.
      FAC=1.
      DO 11 N=1,ITMAX
        AN=FLOAT(N)
        ANA=AN-A
        A0=(A1+A0*ANA)*FAC
        B0=(B1+B0*ANA)*FAC
        ANF=AN*FAC
        A1=X*A0+ANF*A1
        B1=X*B0+ANF*B1
        IF(A1.NE.0.)THEN
          FAC=1./A1
          G=B1*FAC
          IF(ABS((G-GOLD)/G).LT.EPS)GO TO 1
          GOLD=G
        ENDIF
11    CONTINUE
      PAUSE 'A too large, ITMAX too small'
1     GAMMCF=EXP(-X+A*LOG(X)-GLN)*G
      RETURN
      END


      SUBROUTINE GSER(GAMSER,A,X,GLN)
C
```

```
C     SERIES REPRESENTATION OF PARTIAL GAMMA FUNCTION,  FROM
C     "NUMERICAL RECIPES."
C
      PARAMETER (ITMAX=100,EPS=3.E-7)
      GLN=GAMMLN(A)
      IF(X.LE.0.)THEN
        IF(X.LT.0.)PAUSE
        GAMSER=0.
        RETURN
      ENDIF
      AP=A
      SUM=1./A
      DEL=SUM
      DO 11 N=1,ITMAX
        AP=AP+1.
        DEL=DEL*X/AP
        SUM=SUM+DEL
        IF(ABS(DEL).LT.ABS(SUM)*EPS)GO TO 1
11    CONTINUE
      PAUSE 'A too large, ITMAX too small'
1     GAMSER=SUM*EXP(-X+A*LOG(X)-GLN)
      RETURN
      END


      SUBROUTINE TRAPZD(FUNC,A,B,S,N)
C
      REAL FUNC
      EXTERNAL FUNC
C
      SAVE IT
C
      IF (N.EQ.1)THEN
         S=0.5*(B-A)*(FUNC(A)+FUNC(B))
         IT=1
      ELSE
         TNM=IT
         DEL=(B-A)/TNM
         X=A+0.5*DEL
         SUM=0.
         DO 11 J=1,IT
            SUM=SUM+FUNC(X)
            X=X+DEL
11       CONTINUE
         S=0.5*(S+(B-A)*SUM/TNM)
         IT=2*IT
      ENDIF
      RETURN
      END
```

F. Program SPECTRUM

The program SPECTRUM is listed here in its entirety. All subroutines beginning with ICH_ are part of the Caltech Astronomy character function library ICH. All subroutines beginning with PG are part of the PGPLOT package [39].

```
      PROGRAM SPECTRUM
C+
C
C     S P E C T R U M
C
C     Takes a set of line data tables and a directory full of
C     theoretical line profiles and allows the user to interactively
C     fit the observed data to the theoretical profiles by chi-square
C     minimization.
C
C     Input files:
C
C     Line data are read from files whose names are prompted for by
C     the program.
C
C     Theoretical data are contained in numerous disk files whose
C     names are in the format 'f<grav><abund><wave>.' where <grav>
C     is a two-numeral string specifying gravity, <abund> is a single
C     numeral specifying ratio of helium to hydrogen, and <wave> is
C     a four-numeral string specifying the line.  Thus, 'f4514863' is
C     a file containing theoretical profiles of the hydrogen 4863 angstrom
C     line (H beta) for log g of 4.5 and He/H of 0.1.  The profiles
C     are listed within the file by temperature; the data consist of
C     a record that starts with the temperature (the remainder of the
C     record may be discarded) followed by a record containing the
C     number of points in the profile, followed by the wavelengths and
C     data for those points.
C
C     If the specified library file does not contain a profile at
C     the appropriate temperature, the program attempts to interpolate
C     in abundance first and then in log gravity.  If neither works,
C     the program so notifies the user.
C
C     Output files:
C
C     The user specifies a soft device (must be something with a cursor)
C     on which lines are displayed and fits may be made.
C     The user is also prompted for a hard device type; when he finishes
C     determining his fit, the final results are written to the disk
C     in a format suitable for the specified hard-device type.  The
C     user may specify the hard device to be a soft device if he wishes
C     to eyeball the fit before making a hard-output plot.
C
C     The remainder of the program is menu-driven.
C
C+
      IMPLICIT NONE
C
C     Functions
C
      INTEGER ICH_LEN, LOCATE, PGBEGIN
C
C     Parameters
C
      INTEGER MPIX, ML
      PARAMETER (MPIX=801, ML=12)
C
C     Local variables
C
      CHARACTER CC*1, CVALUE*32, CWAVE*4, FILE*32, HARD*8, OPTION*1
      CHARACTER SOFT*32, TITLE*80, XLABEL*20, YLABEL*13
C
      LOGICAL EXIST, FETCHED, IFAIL, ROTATED
C
      INTEGER I, IA1, IA2, ICENTER(ML), ICLEFT1(ML), ICLEFT2(ML)
      INTEGER ICRIGHT1(ML), ICRIGHT2(ML), IDCEN, IDUMMY, IG1
      INTEGER IG2, IT1, IT2, J, JJ, K, L, LISTA(40), LTITLE
      INTEGER MA, MFIT, NDPIX(ML), OLDNLINES, PLN
C
      REAL ABUND, AGRID(4), CONTA(ML), CONTB(ML), DEV, FITA, FITB
      REAL FITC, FITD, GGRID(5), GRAV, NABUND, NGRAV, NROT, NTEFF
      REAL OLDROT, OSIG(MPIX,ML)
      REAL OX(MPIX,ML), OY(MPIX,ML), ROT, SIG, SIGFIT(MPIX)
      REAL SUMA, SUMA2, SUMG, SUMG2, SUMT, SUMT2, SUMWA, SUMWG, SUMWT
      REAL T000(MPIX,ML), T001(MPIX,ML), T010(MPIX,ML), T011(MPIX,ML)
      REAL T100(MPIX,ML), T101(MPIX,ML), T110(MPIX,ML), T111(MPIX,ML)
      REAL TCENTER(ML), TEFF, TGRID(8), WAVE(ML), WEIGHT(ML), WT, X
      REAL X1, X2, X3, X4, XCENTER(ML), XFIT(MPIX), YFIT1(MPIX)
      REAL YFIT2(MPIX), YFIT3(MPIX), XPLOT(MPIX), Y, Y1, Y2, YPLOT(MPIX)
```

```
      REAL YFIN1(MPIX), YFIN2(MPIX), YFIN3(MPIX), YFIN4(MPIX)
      REAL YFIN5(MPIX)
C
      REAL*8 CPROJ(40,40), DALPHA(40,40), DCOVAR(40,40), DERR, DET
      REAL*8 DFIT(40), DLAMBDA, DRMS, DXPLOT(MPIX), DYPLOT(MPIX)
      REAL*8 OLDLAMBDA, OLDRMS, PERR(4)
      REAL*8 SUMX, SUMX2, SUMY, SUMXY
C
C     Main data COMMON block
C
      INTEGER DPIX(ML), NLINES
      REAL*8 DOSIG(MPIX,ML), DOX(MPIX,ML), DOY(MPIX,ML)
      REAL*8 RT0000(MPIX,ML), RT0001(MPIX,ML), RT0010(MPIX,ML)
      REAL*8 RT0011(MPIX,ML), RT0100(MPIX,ML), RT0101(MPIX,ML)
      REAL*8 RT0110(MPIX,ML), RT0111(MPIX,ML), RT1000(MPIX,ML)
      REAL*8 RT1001(MPIX,ML), RT1010(MPIX,ML), RT1011(MPIX,ML)
      REAL*8 RT1100(MPIX,ML), RT1101(MPIX,ML), RT1110(MPIX,ML)
      REAL*8 RT1111(MPIX,ML)
      COMMON /MAIN/DPIX, NLINES, DOSIG, DOX, DOY, RT0000, RT0001,
     : RT0010, RT0011, RT0100, RT0101, RT0110, RT0111, RT1000,
     : RT1001, RT1010, RT1011, RT1100, RT1101, RT1110, RT1111
C
C     PLOTFUNC common block
C
      COMMON /PLOTFUNC/DXPLOT, DYPLOT, PLN
C
C     Data statements
C
      DATA XLABEL/'Wavelength/Angstroms'/
      DATA YLABEL/'Relative Flux'/
C
      DATA AGRID/0.05, 0.1, 0.2, 0.5/
      DATA GGRID/2.75, 3.0, 3.5, 4.0, 4.5/
      DATA TGRID/28000., 30000., 32500., 35000., 37500., 40000., 45000.,
     :  50000./
C
C     Externals
C
      EXTERNAL GAUSS, ROTFUNC, PHYSFUNC, PLOTFUNC
C
C     * Start of executable statements *
C
      NLINES=0
      OLDNLINES=0
      IT1=0
      IA1=0
      IG1=0
      ROT=0.
      FETCHED=.FALSE.
      ROTATED=.FALSE.
C
C     Get the top label for all plots
C
70    WRITE (6,*)'Object name? '
      READ (5,2000,ERR=70)TITLE
      LTITLE=ICH_LEN(TITLE)
C
C     Query user for soft device type
C
10    WRITE (6,*)'Soft device type? '
      READ (5,2000,ERR=10)SOFT
      CALL PGQINF('CURSOR',CVALUE,I)
      IF (CVALUE(:1).EQ.'N')THEN
         WRITE (6,*)'Please select a device with a cursor.'
         GO TO 10
      ENDIF
C
C     Open soft device
C
      CALL PGBEGIN(0,SOFT,1,1)
C
C     Get option
C
15    WRITE (6,*)'Option? '
      READ (5,2001)OPTION
      IF (OPTION.EQ.'?')THEN
C
C        List the options
C
         WRITE (6,*)'Options are:'
```

```
         WRITE (6,*)
         WRITE (6,*)'?   Print this message'
         WRITE (6,*)'O   Read in an observed profile'
         WRITE (6,*)'P   Choose values for the three parameters'
         WRITE (6,*)'R   Rotate theoretical profiles'
         WRITE (6,*)'V   Measure VSINI'
         WRITE (6,*)'F   Measure parameters'
         WRITE (6,*)'H   Produce hardcopy'
         WRITE (6,*)'X   Exit'
         GO TO 15
C
      ELSE IF (OPTION.EQ.'O'.OR.OPTION.EQ.'o')THEN
C
C        Prompt user for data file
C
         I=NLINES+1
30       WRITE (6,1000)I
1000     FORMAT (1X,'Data file for line ',I2,'?')
         READ (5,2000,ERR=30)FILE
         OPEN (UNIT=1,FILE=FILE, ACCESS='SEQUENTIAL',STATUS='OLD',
     :           READONLY,ERR=30)
C
C        Read data records to end of file
C
         READ (1,*)
         DO J=1,800
            READ (1,*,END=40)OX(J,I),OY(J,I)
         ENDDO
         WRITE (6,*)'Warning -- Not All of File Read'
40       CLOSE (1)
         NDPIX(I)=J-1
51       WRITE (6,*)'Wavelength of line? '
         READ (5,*,ERR=51)WAVE(I)
         WRITE (CWAVE,1001)INT(WAVE(I))
         INQUIRE (FILE='f450'//CWAVE//'.',EXIST=EXIST)
         IF (.NOT.EXIST)THEN
            WRITE (6,*)'That line is not in the library.'
            GO TO 15
         ENDIF
         IDCEN=LOCATE(OX(1,I),NDPIX(I),WAVE(I))
         IF (OX(1,I).GT.OX(2,I))THEN
C
C        Reverse the order of the arrays if necessary.
C        (They must be ordered from lowest wavelength to highest.)
C
            DO J=1,NDPIX(I)
               YPLOT(J)=OY(NDPIX(I)-J+1,I)
               XPLOT(J)=OX(NDPIX(I)-J+1,I)
            ENDDO
            DO J=1,NDPIX(I)
               OX(J,I)=XPLOT(J)
               OY(J,I)=YPLOT(J)
            ENDDO
         ENDIF
C
C     Now throw up the plots and let the user take a look at them.
C     The user must specify the "shoulders" of the plots, e.g.,
C     the region around the line that will be used to fine-tune
C     the continuum level.  The user is then prompted for the
C     line center; the program tries to refine this guess by
C     fitting a Gaussian to the line.
C
         CALL PGSCI(1)
         CALL PGENV(WAVE(I)-20.0, WAVE(I)+20.0, 0.0, 1.1, 0, 0)
         CALL PGLABEL(XLABEL,YLABEL,TITLE(:LTITLE))
         CALL PGBIN(NDPIX(I),OX(1,I),OY(1,I),.TRUE.)
         WRITE (6,*)'Indicate limits of left shoulder:'
         CALL PGCURSE(X,Y,CC)
         ICLEFT1(I)=LOCATE(OX(1,I),NDPIX(I),X)
         CALL PGCURSE(X,Y,CC)
         ICLEFT2(I)=LOCATE(OX(1,I),NDPIX(I),X)
         WRITE (6,*)'Indicate limits of right shoulder:'
         CALL PGCURSE(X,Y,CC)
         ICRIGHT1(I)=LOCATE(OX(1,I),NDPIX(I),X)
         CALL PGCURSE(X,Y,CC)
         ICRIGHT2(I)=LOCATE(OX(1,I),NDPIX(I),X)
C
C        Make REAL*8 copy of trimmed profile.
C
         DO J=ICLEFT1(I),ICRIGHT2(I)
```

```
            JJ=J-ICLEFT1(I)+1
            DOX(JJ,I)=OX(J,I)
            DOY(JJ,I)=OY(J,I)
         ENDDO
         DPIX(I)=ICRIGHT2(I)-ICLEFT1(I)+1
C
C        Calculate line through shoulders.
C
         CALL LINEFIT(ICLEFT1(I),ICLEFT2(I),ICRIGHT1(I),ICRIGHT2(I),
     :                OX(1,I),OY(1,I),CONTA(I),CONTB(I))
         CALL PGSCI(2)
         X=OX(ICLEFT1(I),I)
         X1=OX(ICRIGHT2(I),I)
         Y=CONTA(I)*X+CONTB(I)
         Y1=CONTA(I)*X1+CONTB(I)
         CALL PGMOVE(X,Y)
         CALL PGDRAW(X1,Y1)
         CALL PGSCI(1)
C
C        Now get approximate line center.
C
50       WRITE (6,*)'Indicate approximate line center.'
         WRITE (6,*)'(Be sure both X and Y are indicated.)'
         CALL PGCURSE(X,Y,CC)
C
C        Evaluate best Gaussian fit to line center.
C
         DFIT(1)=X
         DFIT(2)=1.-Y
         DFIT(3)=1.0
         DFIT(4)=CONTA(I)
         DFIT(5)=CONTB(I)
         DLAMBDA=-1.
         DO J=1,5
            LISTA(J)=J
         ENDDO
         CALL MRQMIN(DOX(1,I),DOY(1,I),DPIX(I),DFIT,
     :               5,LISTA,3,DCOVAR,DALPHA,40,OLDRMS,GAUSS,DLAMBDA)
75          CALL MRQMIN(DOX(1,I),DOY(1,I),DPIX(I),DFIT,
     :               5,LISTA,3,DCOVAR,DALPHA,40,DRMS,GAUSS,DLAMBDA)
            IF (MAX(ABS((DRMS-OLDRMS)/DRMS),ABS(DRMS-OLDRMS)).GT.
     :          (1.E-3))THEN
               OLDRMS=DRMS
               GO TO 75
            ENDIF
         XCENTER(I)=DFIT(1)
         ICENTER(I)=LOCATE(OX(1,I),NDPIX(I),XCENTER(I))
         WRITE (6,*)'Line center calculated as ',XCENTER(I)
C
C        Toss up the line so that the user can see if he likes it.
C
         DO J=1,DPIX(I)
            XPLOT(J)=DOX(J,I)
            YPLOT(J)=DFIT(4)*XPLOT(J)+DFIT(5)
            YPLOT(J)=YPLOT(J)*(1.0-DFIT(2)*EXP(-((DFIT(1)-XPLOT(J))/
     :               DFIT(3))**2))
         ENDDO
         CALL PGSCI(3)
         CALL PGLINE(ICRIGHT2(I)-ICLEFT1(I)+1,XPLOT,YPLOT)
C
C        Estimate V sin I and the equivalent width.
C
         WRITE (6,*)'Line center calculated as ',XCENTER(I)
         WRITE (6,*)'Estimated V SIN I is ',ABS(DFIT(3))*3.E5/WAVE(I)
         WRITE (6,*)'Equivalent width is ',DFIT(2)*ABS(DFIT(3))*1.77245
         WRITE (6,*)'Keep this line? '
         READ (5,2001)OPTION
         IF (OPTION.EQ.'N'.OR.OPTION.EQ.'n')GO TO 15
77       WRITE (6,*)'Theoretical weight factor? '
         READ (5,*,ERR=77)WEIGHT(I)
         NLINES=I
C
C     Estimate the errors in the flux across the profile.
C
         DO J=1,DPIX(I)
            K=0
            SUMX=0.
            SUMX2=0.
            SUMY=0.
            SUMXY=0.
```

```
            DO JJ=J-2,J+2
               IF (JJ.GT.0.AND.JJ.LE.DPIX(I))THEN
                  K=K+1
                  SUMX=SUMX+DOX(JJ,I)
                  SUMX2=SUMX2+DOX(JJ,I)**2
                  SUMY=SUMY+DOY(JJ,I)
                  SUMXY=SUMXY+DOX(JJ,I)*DOY(JJ,I)
               ENDIF
            ENDDO
            DET=K*SUMX2-SUMX*SUMX
            FITA=(SUMY*SUMX2-SUMXY*SUMX)/DET
            FITB=(K*SUMXY-SUMX*SUMY)/DET
            SUMX2=0.
            DO JJ=J-2,J+2
               IF (JJ.GT.0.AND.JJ.LE.DPIX(I))THEN
                  SUMX2=SUMX2+(DOY(JJ,I)-FITA-FITB*DOX(JJ,I))**2
               ENDIF
            ENDDO
            DOSIG(J,I)=SQRT(SUMX2/(K-1))/MAX(1.E-25,WEIGHT(I))
         ENDDO
         FETCHED=.FALSE.
         ROTATED=.FALSE.
         GO TO 15
C
      ELSE IF (OPTION.EQ.'P'.OR.OPTION.EQ.'p')THEN
C
C     Get the estimate of stellar parameters
C
80       WRITE (6,*)'Estimate of TEFF? '
         READ (5,*,ERR=80)TEFF
         I=LOCATE(TGRID,8,TEFF)
         IF (I.LT.1.OR.I.GE.8)THEN
            WRITE (6,*)'Unacceptable temperature range'
            GO TO 80
         ENDIF
91       WRITE (6,*)'Estimate of LOG G? '
         READ (5,*,ERR=91)GRAV
         J=LOCATE(GGRID,5,GRAV)
         IF (J.LT.1.OR.J.GE.5)THEN
            WRITE (6,*)'Unacceptable gravity range'
            GO TO 91
         ENDIF
101      WRITE (6,*)'Estimate of He/H? '
         READ (5,*,ERR=101)ABUND
         K=LOCATE(AGRID,4,ABUND)
         IF (K.LT.1.OR.K.GT.4)THEN
            WRITE (6,*)'Unacceptable abundance range'
            GO TO 101
         ENDIF
C
C     Now try getting the needed libraries.
C
         IF (I.EQ.IT1.AND.J.EQ.IG1.AND.K.EQ.IA1.AND.OLDNLINES.EQ.
     :       NLINES)GO TO 15
         OLDNLINES=NLINES
         IT1=I
         IT2=I+1
         IG1=J
         IG2=J+1
         IA1=K
         IA2=K+1
         IFAIL=.FALSE.
         DO I=1,NLINES
            WRITE (CWAVE,1001)INT(WAVE(I))
            CALL FETCH(IT1,IG1,IA1,CWAVE,I,T000(1,I),FETCHED)
            IFAIL=.NOT.FETCHED
            CALL FETCH(IT1,IG2,IA1,CWAVE,I,T010(1,I),FETCHED)
            IFAIL=IFAIL.OR..NOT.FETCHED
            CALL FETCH(IT2,IG1,IA1,CWAVE,I,T100(1,I),FETCHED)
            IFAIL=IFAIL.OR..NOT.FETCHED
            CALL FETCH(IT2,IG2,IA1,CWAVE,I,T110(1,I),FETCHED)
            IFAIL=IFAIL.OR..NOT.FETCHED
            CALL FETCH(IT1,IG1,IA2,CWAVE,I,T001(1,I),FETCHED)
            IFAIL=IFAIL.OR..NOT.FETCHED
            CALL FETCH(IT1,IG2,IA2,CWAVE,I,T011(1,I),FETCHED)
            IFAIL=IFAIL.OR..NOT.FETCHED
            CALL FETCH(IT2,IG1,IA2,CWAVE,I,T101(1,I),FETCHED)
            IFAIL=IFAIL.OR..NOT.FETCHED
            CALL FETCH(IT2,IG2,IA2,CWAVE,I,T111(1,I),FETCHED)
            IFAIL=IFAIL.OR..NOT.FETCHED
```

```
            IF (IFAIL)THEN
               WRITE (6,*)'Try a different set of parameters'
               FETCHED=.FALSE.
               GO TO 15
            ENDIF
            K=0
            X=1.E38
            DO J=1,DPIX(I)
               IF (T000(J,I).LT.X)THEN
                  K=J
                  X=T000(J,I)
               ENDIF
            ENDDO
            TCENTER(I)=DOX(K,I)
         ENDDO
         FETCHED=.TRUE.
         ROTATED=.FALSE.
         GO TO 15
      ELSE IF (OPTION.EQ.'V'.OR.OPTION.EQ.'v')THEN
C
C     Get a good measure of the rotational velocity by chi-square
C     minimization.  In practice this does not work very well; the
C     user is almost always better off to eyeball it by using
C     the R and H commands to adjust the rotation parameter.
C
         IF (.NOT.FETCHED)THEN
            WRITE (6,*)'Please estimate parameters first.'
            GO TO 15
         ENDIF
         WRITE (6,*)'Value of VSINI to bracket? '
         READ (5,*)ROT
C
C     Calculate profiles for given rotation.
C
         DO I=1,NLINES
            CALL ROTATE(I,T000(1,I),RT0000(1,I),.8333*ROT)
            CALL ROTATE(I,T001(1,I),RT0010(1,I),.8333*ROT)
            CALL ROTATE(I,T010(1,I),RT0100(1,I),.8333*ROT)
            CALL ROTATE(I,T011(1,I),RT0110(1,I),.8333*ROT)
            CALL ROTATE(I,T100(1,I),RT1000(1,I),.8333*ROT)
            CALL ROTATE(I,T101(1,I),RT1010(1,I),.8333*ROT)
            CALL ROTATE(I,T110(1,I),RT1100(1,I),.8333*ROT)
            CALL ROTATE(I,T111(1,I),RT1110(1,I),.8333*ROT)
            CALL ROTATE(I,T000(1,I),RT0001(1,I),1.1667*ROT)
            CALL ROTATE(I,T001(1,I),RT0011(1,I),1.1667*ROT)
            CALL ROTATE(I,T010(1,I),RT0101(1,I),1.1667*ROT)
            CALL ROTATE(I,T011(1,I),RT0111(1,I),1.1667*ROT)
            CALL ROTATE(I,T100(1,I),RT1001(1,I),1.1667*ROT)
            CALL ROTATE(I,T101(1,I),RT1011(1,I),1.1667*ROT)
            CALL ROTATE(I,T110(1,I),RT1101(1,I),1.1667*ROT)
            CALL ROTATE(I,T111(1,I),RT1111(1,I),1.1667*ROT)
         ENDDO
C
C        Now find the best fit.
C
         MFIT=3*NLINES+1
         DO I=1,NLINES
            DFIT(3*I-2)=CONTA(I)
            DFIT(3*I-1)=CONTB(I)
            DFIT(3*I)=XCENTER(I)-TCENTER(I)
         ENDDO
         DFIT(3*NLINES+1)=(TEFF-TGRID(IT1))/(TGRID(IT2)-TGRID(IT1))
         DFIT(3*NLINES+2)=(GRAV-GGRID(IG1))/(GGRID(IG2)-GGRID(IG1))
         DFIT(3*NLINES+3)=LOG(ABUND/AGRID(IA1))/LOG(AGRID(IA2)/
     :                         AGRID(IA1))
         DFIT(3*NLINES+4)=0.5
         DLAMBDA=-1.
         DO I=1,3*NLINES
            LISTA(I)=I
         ENDDO
         LISTA(MFIT)=3*NLINES+4
         CALL MRQMIN2(DFIT,MFIT+3,LISTA,MFIT,DCOVAR,DALPHA,40,DRMS,
     :                DLAMBDA,ROTFUNC)
261         CALL MRQMIN2(DFIT,MFIT+3,LISTA,MFIT,DCOVAR,DALPHA,40,DRMS,
     :                DLAMBDA,ROTFUNC)
            WRITE (6,*)DRMS,DFIT(MFIT+3)
            WRITE (6,*)'Iterate again? '
            READ (5,2001)OPTION
            IF (OPTION.EQ.'Y'.OR.OPTION.EQ.'y')GO TO 261
         NROT=DFIT(MFIT+3)
```

```
C
C        Now estimate probable error.
C
         DLAMBDA=0.0
         CALL MRQMIN2(DFIT,MFIT+3,LISTA,MFIT,DCOVAR,DALPHA,40,DRMS,
     :                    DLAMBDA,ROTFUNC)
         PERR(4)=SQRT(DCOVAR(MFIT+3,MFIT+3)*.3*DRMS)*.3333*ROT
         NROT=.8333*ROT+.3333*ROT*NROT
         WRITE (6,*)'VSINI = ',NROT,' P.E. = ',PERR(4)
         ROTATED=.FALSE.
         GO TO 15
      ELSE IF (OPTION.EQ.'R'.OR.OPTION.EQ.'r')THEN
C
C        Rotate the profiles fetched from the libraries.
C
         WRITE (6,*)'Value of VSINI to use?'
         READ (5,*)ROT
         DO I=1,NLINES
            CALL ROTATE(I,T000(1,I),RT0000(1,I),ROT)
            CALL ROTATE(I,T001(1,I),RT0010(1,I),ROT)
            CALL ROTATE(I,T010(1,I),RT0100(1,I),ROT)
            CALL ROTATE(I,T011(1,I),RT0110(1,I),ROT)
            CALL ROTATE(I,T100(1,I),RT1000(1,I),ROT)
            CALL ROTATE(I,T101(1,I),RT1010(1,I),ROT)
            CALL ROTATE(I,T110(1,I),RT1100(1,I),ROT)
            CALL ROTATE(I,T111(1,I),RT1110(1,I),ROT)
         ENDDO
         ROTATED=.TRUE.
         GO TO 15
      ELSE IF (OPTION.EQ.'F'.OR.OPTION.EQ.'f')THEN
C
C        Do the big fit by chi-square minimization.
C
         MFIT=NLINES*3+3
         DO I=1,NLINES
C
C           Initial estimate of the continuum and wavelength
C           zero point parameters.
C
            DFIT(3*I-2)=CONTA(I)
            DFIT(3*I-1)=CONTB(I)
            DFIT(3*I)=XCENTER(I)-TCENTER(I)
         ENDDO
C
C        Initial estimates of temperature, gravity, and abundance.
C
         DFIT(3*NLINES+1)=(TEFF-TGRID(IT1))/(TGRID(IT2)-TGRID(IT1))
         DFIT(3*NLINES+2)=(GRAV-GGRID(IG1))/(GGRID(IG2)-GGRID(IG1))
         DFIT(3*NLINES+3)=LOG(ABUND/AGRID(IA1))/LOG(AGRID(IA2)/
     :                       AGRID(IA1))
         DLAMBDA=-1.
C
C        Set up and execute the minimization.
C
         DO I=1,MFIT+1
            LISTA(I)=I
         ENDDO
260         CALL MRQMIN2(DFIT,MFIT+1,LISTA,MFIT,DCOVAR,DALPHA,40,DRMS,
     :                    DLAMBDA,PHYSFUNC)
C
C           Let the user decide if he is satisfied with current
C           minimization.
C
            WRITE (6,*)DRMS,DFIT(MFIT-2),DFIT(MFIT-1),DFIT(MFIT)
            WRITE (6,*)'Iterate again? '
            READ (5,2001)OPTION
            IF (OPTION.EQ.'Y'.OR.OPTION.EQ.'y')GO TO 260
         DO I=1,NLINES
            CONTA(I)=DFIT(3*I-2)
            CONTB(I)=DFIT(3*I-1)
            XCENTER(I)=DFIT(3*I)+TCENTER(I)
         ENDDO
         NTEFF=DFIT(3*NLINES+1)
         NGRAV=DFIT(3*NLINES+2)
         NABUND=DFIT(3*NLINES+3)
C
C        Now estimate probable error from covariance matrix.
C
         DLAMBDA=0.0
         CALL MRQMIN2(DFIT,MFIT+1,LISTA,MFIT,DCOVAR,DALPHA,40,DRMS,
```

```
     :                  DLAMBDA,PHYSFUNC)
        DO I=1,3*NLINES+3
           DO J=1,3*NLINES+3
              CPROJ(I,J)=DCOVAR(I,J)
           ENDDO
        ENDDO
        CALL MATINV(CPROJ,3*NLINES+3,40)
        J=3*NLINES
        NTEFF=NTEFF*TGRID(IT2)+(1.-NTEFF)*TGRID(IT1)
        PERR(1)=SQRT(.3*DRMS/CPROJ(J+1,J+1))*(TGRID(IT2)-TGRID(IT1))
        NGRAV=NGRAV*GGRID(IG2)+(1.-NGRAV)*GGRID(IG1)
        PERR(2)=SQRT(.3*DRMS/CPROJ(J+2,J+2))*(GGRID(IG2)-GGRID(IG1))
        NABUND=EXP(NABUND*LOG(AGRID(IA2))+(1.-NABUND)*LOG(AGRID(IA1)))
        PERR(3)=SQRT(.3*DRMS/CPROJ(J+3,J+3))*LOG(AGRID(IA2)/
     :           AGRID(IA1))*NABUND
        WRITE (6,*)
        WRITE (6,*)'Parameters estimated as:'
        WRITE (6,*)'TEFF = ',NTEFF,' P.E. = ',PERR(1)
        WRITE (6,*)'GRAV = ',NGRAV,' P.E. = ',PERR(2)
        WRITE (6,*)'ABUND = ',NABUND,' P.E. = ',PERR(3)
        WRITE (6,*)
        WRITE (6,*)'Inverse covariance matrix:'
        WRITE (6,*)'CHISQ = ',DRMS
        DO I=1,3
           WRITE (6,*)(CPROJ(I,J),J=1,3)
        ENDDO
        GO TO 15
      ELSE IF (OPTION.EQ.'H')THEN
C
C       Make a hard plot of the current fit.
C
        IF (.NOT.FETCHED)THEN
           WRITE (6,*)'Please specify parameters first'
           GO TO 15
        ELSE IF (.NOT.ROTATED)THEN
           WRITE (6,*)'Please specify rotation first'
           GO TO 15
        ENDIF
270     WRITE (6,*)'Hard device type? '
        READ (5,2000,ERR=270)HARD
        WRITE (6,*)'TEFF, GRAV, ABUND to use? '
        READ (5,*)NTEFF,NGRAV,NABUND
        CALL PGBEGIN(0,HARD,1,1)
        DO I=1,NLINES
           CALL PGENV(WAVE(I)-20.0, WAVE(I)+20.0, 0.0, 1.1, 0, 0)
           CALL PGLABEL(XLABEL,YLABEL,TITLE(:LTITLE))
           CALL PGBIN(NDPIX(I),OX(1,I),OY(1,I),.TRUE.)
C
C       Calculate the profile to use.
C
           X=(NTEFF-TGRID(IT1))/(TGRID(IT2)-TGRID(IT1))
           DO J=1,DPIX(I)
              YFIN1(J)=X*RT1000(J,I)+(1.-X)*RT0000(J,I)
              YFIN2(J)=X*RT1010(J,I)+(1.-X)*RT0010(J,I)
              YFIN3(J)=X*RT1100(J,I)+(1.-X)*RT0100(J,I)
              YFIN4(J)=X*RT1110(J,I)+(1.-X)*RT0110(J,I)
           ENDDO
           X=(NGRAV-GGRID(IG1))/(GGRID(IG2)-GGRID(IG1))
           DO J=1,DPIX(I)
              YFIN1(J)=X*YFIN3(J)+(1.-X)*YFIN1(J)
              YFIN2(J)=X*YFIN4(J)+(1.-X)*YFIN2(J)
           ENDDO
           X=LOG(NABUND/AGRID(IA1))/LOG(AGRID(IA2)/AGRID(IA1))
           DO J=1,DPIX(I)
              DXPLOT(J)=DOX(J,I)
              DYPLOT(J)=X*YFIN2(J)+(1.-X)*YFIN1(J)
              XPLOT(J)=DXPLOT(J)
              YPLOT(J)=DYPLOT(J)
           ENDDO
           PLN=DPIX(I)
C
C          Now fit the continuum parameters to the data.
C
           DFIT(1)=CONTA(I)
           DFIT(2)=CONTB(I)
           DFIT(3)=0.
           DLAMBDA=-1.
           DO J=1,3
              LISTA(J)=J
           ENDDO
```

```
            CALL MRQMIN(DOX(1,I),DOY(1,I),DPIX(I),DFIT,3,LISTA,3,DCOVAR,
     :                 DALPHA,40,OLDRMS,PLOTFUNC,DLAMBDA)
76             CALL MRQMIN(DOX(1,I),DOY(1,I),DPIX(I),DFIT,3,LISTA,3,
     :                 DCOVAR,DALPHA,40,DRMS,PLOTFUNC,DLAMBDA)
            IF (ABS((DRMS-OLDRMS)/DRMS).GT.1.E-3.AND.
     :          ABS(DRMS-OLDRMS).GT.(.1))THEN
               OLDRMS=DRMS
               GO TO 76
            ENDIF
            DO J=1,DPIX(I)
               XPLOT(J)=XPLOT(J)+DFIT(3)
               YPLOT(J)=YPLOT(J)*(DFIT(1)*XPLOT(J)+DFIT(2))
            ENDDO
            CALL PGLINE(DPIX(I),XPLOT,YPLOT)
         ENDDO
C
C        Restore the soft device.
C
         CALL PGEND
         CALL PGBEGIN(0,SOFT,1,1)
         GO TO 15
      ELSE IF (OPTION.EQ.'X')THEN
C
C        EXIT
C
         CALL PGEND
         CALL EXIT
      ELSE
         GO TO 15
      ENDIF
C
1001  FORMAT (I4)
2000  FORMAT (A)
2001  FORMAT (A1)
      END

      INTEGER FUNCTION DLOCATE(XX,N,X)
C
C     This is essentially the Numerical Recipes routine LOCATE,
C     but here defined as a function rather than subroutine.
C     Also, the real arguments are REAL*8.
C
      REAL*8 XX(N), X
      JL=0
      JU=N+1
10    IF (JU-JL.GT.1)THEN
         JM=(JU+JL)/2
         IF ((XX(N).GT.XX(1)).EQV.(X.GT.XX(JM)))THEN
            JL=JM
         ELSE
            JU=JM
         ENDIF
      GO TO 10
      ENDIF
      DLOCATE=MAX(1,MIN(N,JL))
      RETURN
      END


      SUBROUTINE LINEFIT(I1,I2,I3,I4,X,Y,A,B)
C
C     Fit a line through two regions of a profile.
C
      IMPLICIT NONE
C
C     Compiler parameters
C
      INTEGER MPIX
      PARAMETER (MPIX=801)
C
C     Parameters
C
      INTEGER I1,I2,I3,I4
      REAL X(MPIX),Y(MPIX),A,B
C
C     Local variables
C
      INTEGER I,J
      REAL*8 SUMY, SUMXY, SUMX, SUMX2, N, RMS, DET
C
C     Start of executable statements
```

```
C
      SUMY=0.0
      SUMXY=0.0
      SUMX=0.0
      SUMX2=0.0
      N=I4-I3+I2-I1+2
      DO I=I1,I2
         SUMY=SUMY+Y(I)
         SUMXY=SUMXY+Y(I)*X(I)
         SUMX=SUMX+X(I)
         SUMX2=SUMX2+X(I)*X(I)
      ENDDO
      DO I=I3,I4
         SUMY=SUMY+Y(I)
         SUMXY=SUMXY+Y(I)*X(I)
         SUMX=SUMX+X(I)
         SUMX2=SUMX2+X(I)*X(I)
      ENDDO
      DET=SUMX*SUMX-SUMX2*N
      A=(SUMY*SUMX-SUMXY*N)/DET
      B=(SUMX*SUMXY-SUMX2*SUMY)/DET
      RETURN
      END


      INTEGER FUNCTION LOCATE(XX,N,X)
C
C     This is essentially the Numerical Recipes routine by the
C     same name, but here defined as a function rather than subroutine.
C
      DIMENSION XX(N)
      JL=0
      JU=N+1
10    IF (JU-JL.GT.1)THEN
         JM=(JU+JL)/2
         IF ((XX(N).GT.XX(1)).EQV.(X.GT.XX(JM)))THEN
            JL=JM
         ELSE
            JU=JM
         ENDIF
      GO TO 10
      ENDIF
      LOCATE=MAX(1,MIN(N,JL))
      RETURN
      END


      SUBROUTINE ROTATE(NL,FLUX,ROT,VVSINI)
C
C     Convolve profile with rotational profile corresponding
C     to VSINI=ROT.  It incidentally reverses the order of
C     the profile points if they are not in ascending order.
C
      IMPLICIT NONE
C
C     Compiler parameters
C
      INTEGER MPIX, ML
      REAL PI
      PARAMETER (MPIX=801, ML=12, PI=3.141592654D0)
C
C     Parameters
C
      INTEGER NL
      REAL FLUX(MPIX), VVSINI
      REAL*8 ROT(MPIX)
C
C     Functions
C
      INTEGER DLOCATE
C
C     Local variables
C
      INTEGER I, LL, LH, J, N
      REAL A, AA, BB, A1, B, B1, VSINI
      REAL*8 XHI, XLO
C
C     Main data COMMON block
C
      INTEGER DPIX(ML), NLINES
```

```
      REAL*8 DOSIG(MPIX,ML), DOX(MPIX,ML), DOY(MPIX,ML)
      REAL*8 RT0000(MPIX,ML), RT0001(MPIX,ML), RT0010(MPIX,ML)
      REAL*8 RT0011(MPIX,ML), RT0100(MPIX,ML), RT0101(MPIX,ML)
      REAL*8 RT0110(MPIX,ML), RT0111(MPIX,ML), RT1000(MPIX,ML)
      REAL*8 RT1001(MPIX,ML), RT1010(MPIX,ML), RT1011(MPIX,ML)
      REAL*8 RT1100(MPIX,ML), RT1101(MPIX,ML), RT1110(MPIX,ML)
      REAL*8 RT1111(MPIX,ML)
      COMMON /MAIN/DPIX, NLINES, DOSIG, DOX, DOY, RT0000, RT0001,
     : RT0010, RT0011, RT0100, RT0101, RT0110, RT0111, RT1000,
     : RT1001, RT1010, RT1011, RT1100, RT1101, RT1110, RT1111
C
      N=DPIX(NL)
      VSINI=VVSINI*DOX(N/2,NL)/2.99792E5
      IF (VSINI.EQ.0.)THEN
         DO I=1,N
            ROT(I)=FLUX(I)
         ENDDO
         RETURN
      ENDIF
C
C     Prepare to convolve semi-analytically.
C
      DO I=1,N
         XLO=DOX(I,NL)-VSINI
         LL=DLOCATE(DOX(1,NL),N,XLO)
         IF (XLO.LT.DOX(1,NL))LL=0
         XHI=DOX(I,NL)+VSINI
         LH=DLOCATE(DOX(1,NL),N,XHI)
         IF (FLUX(LL).EQ.(1.0).AND.FLUX(LH).EQ.(1.0))THEN
            ROT(I)=3.141592654/2.
            GO TO 10
         ENDIF
C
C        Low end of convolution
C
         IF (XLO.LT.DOX(1,NL))THEN
            A=(DOX(I,NL)-DOX(1,NL))/VSINI
            A=MAX(-1.,MIN(1.,A))
            ROT(I)=FLUX(1)*0.5*(0.5*PI-ASIN(A)-A*SQRT(1.-A*A))
         ELSE
            A=(DOX(I,NL)-DOX(LL+1,NL))/VSINI
            BB=(FLUX(LL+1)-FLUX(LL))*VSINI/(DOX(LL,NL)-DOX(LL+1,NL))
            AA=FLUX(LL+1)-BB*A
            ROT(I)=AA*0.5*(ACOS(MAX(-1.,MIN(1.,A)))-
     :             A*SQRT(1.-MIN(1.,A*A)))
            A1=SQRT(1.-MIN(1.,A*A))
            ROT(I)=ROT(I)+BB*A1*A1*A1/3.
         ENDIF
C
C        Main portion of convolution
C
         DO J=LL+1,I-1
            B=-(DOX(J,NL)-DOX(I,NL))/VSINI
            A=-(DOX(J+1,NL)-DOX(I,NL))/VSINI
            BB=(FLUX(J+1)-FLUX(J))/(A-B)
            AA=FLUX(J)-BB*B
            A1=ACOS(MAX(-1.,MIN(1.,A)))
            B1=ACOS(MAX(-1.,MIN(1.,B)))
            ROT(I)=ROT(I)+AA*0.5*(A1-B1-0.5*(SIN(2*A1)-SIN(2*B1)))
            A1=SQRT(1.-MIN(1.,A*A))
            B1=SQRT(1.-MIN(1.,B*B))
            ROT(I)=ROT(I)+BB*(A1*A1*A1-B1*B1*B1)/3.
         ENDDO
         DO J=I,LH-1
            A=(DOX(J,NL)-DOX(I,NL))/VSINI
            B=(DOX(J+1,NL)-DOX(I,NL))/VSINI
            BB=(FLUX(J+1)-FLUX(J))/(B-A)
            AA=FLUX(J)-BB*A
            A1=ACOS(MAX(-1.,MIN(1.,A)))
            B1=ACOS(MAX(-1.,MIN(1.,B)))
            ROT(I)=ROT(I)+AA*0.5*(A1-B1-0.5*(SIN(2*A1)-SIN(2*B1)))
            A1=SQRT(1.-MIN(1.,A*A))
            B1=SQRT(1.-MIN(1.,B*B))
            ROT(I)=ROT(I)+BB*(A1*A1*A1-B1*B1*B1)/3.
         ENDDO
C
C        High end of convolution
C
         IF (XHI.GT.DOX(N,NL))THEN
            A=(DOX(N,NL)-DOX(I,NL))/VSINI
```

```
            A=MAX(-1.0,MIN(1.0,A))
            ROT(I)=ROT(I)+FLUX(1)*0.5*(0.5*PI-ASIN(A)-A*SQRT(1.-A*A))
         ELSE
            A=(DOX(LH,NL)-DOX(I,NL))/VSINI
            BB=(FLUX(LH+1)-FLUX(LH))*VSINI/(DOX(LH+1,NL)-DOX(LH,NL))
            AA=FLUX(LH)-BB*A
            ROT(I)=ROT(I)+AA*0.5*(ACOS(MAX(-1.,MIN(1.,A)))-
     :               A*SQRT(1.-MIN(1.,A*A)))
            A1=SQRT(1.-MIN(1.,A*A))
            ROT(I)=ROT(I)+BB*A1*A1*A1/3.
         ENDIF
10       CONTINUE
      ENDDO
C
C     Normalize
C
      DO I=1,N
         ROT(I)=ROT(I)*2./3.141592654
      ENDDO
      RETURN
      END


      SUBROUTINE MRQMIN(X,Y,NDATA,A,MA,LISTA,MFIT,
     *    COVAR,ALPHA,NCA,CHISQ,FUNCS,ALAMDA)
C
C     NUMERICAL RECIPES routine to perform chi-square minimization.
C
      IMPLICIT REAL*8(A-H,O-Z)
      PARAMETER (MMAX=5)
      DIMENSION X(NDATA),Y(NDATA),A(MA),LISTA(MFIT),
     *  COVAR(NCA,NCA),ALPHA(NCA,NCA),ATRY(MMAX),BETA(MMAX),DA(MMAX)
      IF(ALAMDA.LT.0.)THEN
        KK=MFIT+1
        DO 12 J=1,MA
          IHIT=0
          DO 11 K=1,MFIT
            IF(LISTA(K).EQ.J)IHIT=IHIT+1
11        CONTINUE
          IF (IHIT.EQ.0) THEN
            LISTA(KK)=J
            KK=KK+1
          ELSE IF (IHIT.GT.1) THEN
            PAUSE 'Improper permutation in LISTA'
          ENDIF
12      CONTINUE
        IF (KK.NE.(MA+1)) PAUSE 'Improper permutation in LISTA'
        ALAMDA=0.001
        CALL MRQCOF(X,Y,NDATA,A,MA,LISTA,MFIT,ALPHA,BETA,NCA,CHISQ,F
     *UNCS)
        OCHISQ=CHISQ
        DO 13 J=1,MA
          ATRY(J)=A(J)
13      CONTINUE
      ENDIF
      DO 15 J=1,MFIT
        DO 14 K=1,MFIT
          COVAR(J,K)=ALPHA(J,K)
14      CONTINUE
        COVAR(J,J)=ALPHA(J,J)*(1.+ALAMDA)
        DA(J)=BETA(J)
15    CONTINUE
      CALL GAUSSJ(COVAR,MFIT,NCA,DA,1,1)
      IF(ALAMDA.EQ.0.)THEN
        CALL COVSRT(COVAR,NCA,MA,LISTA,MFIT)
        RETURN
      ENDIF
      DO 16 J=1,MFIT
        ATRY(LISTA(J))=ATRY(LISTA(J))+DA(J)
16    CONTINUE
      CALL MRQCOF(X,Y,NDATA,ATRY,MA,LISTA,MFIT,COVAR,DA,NCA,CHISQ,FU
     *NCS)
      IF(CHISQ.LT.OCHISQ)THEN
        ALAMDA=0.1*ALAMDA
        OCHISQ=CHISQ
        DO 18 J=1,MFIT
          DO 17 K=1,MFIT
            ALPHA(J,K)=COVAR(J,K)
17        CONTINUE
          BETA(J)=DA(J)
          A(LISTA(J))=ATRY(LISTA(J))
```

```
18      CONTINUE
      ELSE
        ALAMDA=10.*ALAMDA
        CHISQ=OCHISQ
      ENDIF
      RETURN
      END


      SUBROUTINE MRQCOF(X,Y,NDATA,A,MA,LISTA,MFIT,ALPHA,BETA,NALP,CH
     *ISQ,FUNCS)
      IMPLICIT REAL*8(A-H,O-Z)
      PARAMETER (MMAX=40)
      DIMENSION X(NDATA),Y(NDATA),ALPHA(NALP,NALP),BETA(MA),
     *    DYDA(MMAX),LISTA(MFIT)
      DO 12 J=1,MFIT
        DO 11 K=1,J
          ALPHA(J,K)=0.
11      CONTINUE
        BETA(J)=0.
12    CONTINUE
      CHISQ=0.
      DO 15 I=1,NDATA
        CALL FUNCS(X(I),A,YMOD,DYDA,MA)
        DY=Y(I)-YMOD
        DO 14 J=1,MFIT
          WT=DYDA(LISTA(J))
          DO 13 K=1,J
            ALPHA(J,K)=ALPHA(J,K)+WT*DYDA(LISTA(K))
13        CONTINUE
          BETA(J)=BETA(J)+DY*WT
14      CONTINUE
        CHISQ=CHISQ+DY*DY
15    CONTINUE
      DO 17 J=2,MFIT
        DO 16 K=1,J-1
          ALPHA(K,J)=ALPHA(J,K)
16      CONTINUE
17    CONTINUE
      RETURN
      END


      SUBROUTINE COVSRT(COVAR,NCVM,MA,LISTA,MFIT)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION COVAR(NCVM,NCVM),LISTA(MFIT)
      DO 12 J=1,MA-1
        DO 11 I=J+1,MA
          COVAR(I,J)=0.
11      CONTINUE
12    CONTINUE
      DO 14 I=1,MFIT-1
        DO 13 J=I+1,MFIT
          IF(LISTA(J).GT.LISTA(I)) THEN
            COVAR(LISTA(J),LISTA(I))=COVAR(I,J)
          ELSE
            COVAR(LISTA(I),LISTA(J))=COVAR(I,J)
          ENDIF
13      CONTINUE
14    CONTINUE
      SWAP=COVAR(1,1)
      DO 15 J=1,MA
        COVAR(1,J)=COVAR(J,J)
        COVAR(J,J)=0.
15    CONTINUE
      COVAR(LISTA(1),LISTA(1))=SWAP
      DO 16 J=2,MFIT
        COVAR(LISTA(J),LISTA(J))=COVAR(1,J)
16    CONTINUE
      DO 18 J=2,MA
        DO 17 I=1,J-1
          COVAR(I,J)=COVAR(J,I)
17      CONTINUE
18    CONTINUE
      RETURN
      END


      SUBROUTINE GAUSSJ(A,N,NP,B,M,MP)
      IMPLICIT REAL*8(A-H,O-Z)
```

```
      PARAMETER (NMAX=40)
      DIMENSION A(NP,NP),B(NP,MP),IPIV(NMAX),INDXR(NMAX),INDXC(NMAX)
      DO 11 J=1,N
        IPIV(J)=0
11    CONTINUE
      DO 22 I=1,N
        BIG=0.
        DO 13 J=1,N
          IF(IPIV(J).NE.1)THEN
            DO 12 K=1,N
              IF (IPIV(K).EQ.0) THEN
                IF (ABS(A(J,K)).GE.BIG)THEN
                  BIG=ABS(A(J,K))
                  IROW=J
                  ICOL=K
                ENDIF
              ELSE IF (IPIV(K).GT.1) THEN
                PAUSE 'Singular matrix'
              ENDIF
12          CONTINUE
          ENDIF
13      CONTINUE
        IPIV(ICOL)=IPIV(ICOL)+1
        IF (IROW.NE.ICOL) THEN
          DO 14 L=1,N
            DUM=A(IROW,L)
            A(IROW,L)=A(ICOL,L)
            A(ICOL,L)=DUM
14        CONTINUE
          DO 15 L=1,M
            DUM=B(IROW,L)
            B(IROW,L)=B(ICOL,L)
            B(ICOL,L)=DUM
15        CONTINUE
        ENDIF
        INDXR(I)=IROW
        INDXC(I)=ICOL
        IF (A(ICOL,ICOL).EQ.0.) PAUSE 'Singular matrix.'
        PIVINV=1./A(ICOL,ICOL)
        A(ICOL,ICOL)=1.
        DO 16 L=1,N
          A(ICOL,L)=A(ICOL,L)*PIVINV
16      CONTINUE
        DO 17 L=1,M
          B(ICOL,L)=B(ICOL,L)*PIVINV
17      CONTINUE
        DO 21 LL=1,N
          IF(LL.NE.ICOL)THEN
            DUM=A(LL,ICOL)
            A(LL,ICOL)=0.
            DO 18 L=1,N
              A(LL,L)=A(LL,L)-A(ICOL,L)*DUM
18          CONTINUE
            DO 19 L=1,M
              B(LL,L)=B(LL,L)-B(ICOL,L)*DUM
19          CONTINUE
          ENDIF
21      CONTINUE
22    CONTINUE
      DO 24 L=N,1,-1
        IF(INDXR(L).NE.INDXC(L))THEN
          DO 23 K=1,N
            DUM=A(K,INDXR(L))
            A(K,INDXR(L))=A(K,INDXC(L))
            A(K,INDXC(L))=DUM
23        CONTINUE
        ENDIF
24    CONTINUE
      RETURN
      END

      SUBROUTINE GAUSS(X,A,YFIT,DYDA,MA)
C
C     Generate Gaussian and derivatives for fit to line profiles.
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(5),DYDA(3)
C
C     Start of executable statements
C
```

```
      DEL=A(1)-X
      CONT=A(4)*X+A(5)
      EX=EXP(-(DEL/A(3))**2)
      YFIT=CONT*(1.D0-A(2)*EX)
      DYDA(1)=+2.D0*DEL*A(2)*CONT*EX/A(3)/A(3)
      DYDA(2)=-CONT*EX
      DYDA(3)=-2.D0*DEL*DEL*A(2)*CONT*EX/A(3)/A(3)/A(3)
      RETURN
      END

      SUBROUTINE FETCH(IT,IG,IA,CWAVE,NL,T,FETCHED)
C
C     Fetch a theoretical profile from the library directory.
C
      IMPLICIT NONE
C
C     Compiler parameters
C
      INTEGER ML, MPIX
      PARAMETER (ML=12, MPIX=801)
C
C     Functions
C
      INTEGER LOCATE
C
C     Parameters
C
      INTEGER IT,IG,IA,NL
      CHARACTER*4 CWAVE
      LOGICAL FETCHED
      REAL T(MPIX)
C
C     Local variables
C
      INTEGER I, J, L, NT, N1, IFILE
      CHARACTER*2 GGRID(5)
      CHARACTER*1 AGRID(4)
      CHARACTER FILE*9
      REAL TINT(MPIX,4), TGRID(8), TEFF, VX1
      REAL VX2, VY1, VY2, WAVE, X1(MPIX), X2(MPIX), XX, YY
C
C     Main data COMMON block
C
      INTEGER DPIX(ML), NLINES
      REAL*8 DOSIG(MPIX,ML), DOX(MPIX,ML), DOY(MPIX,ML)
      REAL*8 RT0000(MPIX,ML), RT0001(MPIX,ML), RT0010(MPIX,ML)
      REAL*8 RT0011(MPIX,ML), RT0100(MPIX,ML), RT0101(MPIX,ML)
      REAL*8 RT0110(MPIX,ML), RT0111(MPIX,ML), RT1000(MPIX,ML)
      REAL*8 RT1001(MPIX,ML), RT1010(MPIX,ML), RT1011(MPIX,ML)
      REAL*8 RT1100(MPIX,ML), RT1101(MPIX,ML), RT1110(MPIX,ML)
      REAL*8 RT1111(MPIX,ML)
      COMMON /MAIN/DPIX, NLINES, DOSIG, DOX, DOY, RT0000, RT0001,
     : RT0010, RT0011, RT0100, RT0101, RT0110, RT0111, RT1000,
     : RT1001, RT1010, RT1011, RT1100, RT1101, RT1110, RT1111
C
C     Data statements
C
      DATA GGRID/'27', '30', '35', '40', '45'/
      DATA AGRID/'0', '1', '2', '5'/
      DATA TGRID/28000., 30000., 32500., 35000., 37500., 40000., 45000.,
     : 50000./
C
C     Start of executable statements
C
C     Construct filename
C
      FILE='f'//GGRID(IG)//AGRID(IA)//CWAVE//'.'
C
C     Try opening the file
C
      OPEN (UNIT=1,FILE=FILE,STATUS='OLD',ACCESS='SEQUENTIAL',ERR=600)
C
C     Look for the temperature desired.
C
10       READ (1,*,END=30)TEFF
         IF (TEFF.EQ.TGRID(IT))GO TO 20
         READ (1,*)NT
         DO I=1,NT
            READ (1,*)TINT(1,1)
         ENDDO
```

```
      GO TO 10
C
C     Read in the library profile.
C
20    READ (1,*)NT
      DO I=1,NT
         READ (1,*)X1(I),TINT(I,1)
      ENDDO
      GO TO 100
C
C     Try to interpolate the necessary profile; first try to do
C     it by abundance.
C
30    WRITE (6,*)'Cannot find entry for ',TGRID(IT),'K in '//FILE
      WRITE (6,*)'Will attempt to interpolate profile.'
      CLOSE(1)
      IF (IA.EQ.1.OR.IA.EQ.4)GO TO 200
      DO IFILE=IA-1,IA+1,2
         FILE='f'//GGRID(IG)//AGRID(IFILE)//CWAVE//'.'
         OPEN (UNIT=1,FILE=FILE,STATUS='OLD',ACCESS='SEQUENTIAL',
     :        ERR=600)
40          READ (1,*,END=200)TEFF
            IF (TEFF.EQ.TGRID(IT-1))GO TO 50
            READ (1,*)NT
            DO I=1,NT
               READ (1,*)TINT(1,IFILE)
            ENDDO
            GO TO 40
C
C     Read in the library profile.
C
50       READ (1,*)NT
         DO I=1,NT
            READ (1,*)X1(I),TINT(I,IFILE)
         ENDDO
      ENDDO
C
C     Now interpolate.
C
      DO I=1,NT
         TINT(I,1)=0.5*(TINT(I,IA-1)+TINT(I,IA+1))
      ENDDO
      GO TO 100
C
C     Abundance interpolation fails--try gravity next.
C
200   IF (IG.EQ.1.OR.IG.EQ.4)THEN
         GO TO 590
      ENDIF
      CLOSE (1)
      DO IFILE=IG-1,IG+1,2
         FILE='f'//GGRID(IFILE)//AGRID(IA)//CWAVE//'.'
C
C     Try opening the file.
C
         OPEN (UNIT=1,FILE=FILE,STATUS='OLD',ACCESS='SEQUENTIAL',
     :        ERR=600)
240         READ (1,*,END=590)TEFF
            IF (TEFF.EQ.TGRID(IT))GO TO 250
            READ (1,*)NT
            DO I=1,NT
               READ (1,*)TINT(1,IFILE)
            ENDDO
            GO TO 240
C
C     Read in the library profile.
C
250      READ (1,*)NT
         DO I=1,NT
            READ (1,*)X1(I),TINT(I,IFILE)
         ENDDO
         CLOSE (1)
      ENDDO
C
C     Now interpolate.
C
      DO I=1,NT
         TINT(I,1)=0.5*(TINT(I,IG-1)+TINT(I,IG+1))
      ENDDO
C
```

```
C     Successful read or interpolate.  Now regrid the profile
C     onto the scale used for the data.
C
100   IF (X1(1).GT.X1(2))THEN
C
C        Reverse order of arrays if necessary.
C
         DO I=1,NT
            X2(I)=X1(NT-I+1)
            TINT(I,2)=TINT(NT-I+1,1)
         ENDDO
         DO I=1,NT
            X1(I)=X2(I)
            TINT(I,1)=TINT(I,2)
         ENDDO
      ENDIF
C
C     Find theoretical profile minimum.
C
      J=0
      WAVE=99.
      DO I=1,NT
         IF (TINT(I,1).LT.WAVE)THEN
            WAVE=TINT(I,1)
            J=I
         ENDIF
      ENDDO
      WAVE=X1(J)
      N1=DPIX(NL)
      DO I=1,N1
         IF (DOX(I,NL).LT.X1(1))THEN
C
C           Interpolate off ends of theoretical profile by
C           assuming -2.5 wings.
C
            XX=ABS((DOX(I,NL)-WAVE)/(X1(1)-WAVE))
            T(I)=1.-(1.-TINT(1,1))/XX/XX/SQRT(XX)
         ELSE IF (DOX(I,NL).GT.X1(NT))THEN
            XX=ABS((DOX(I,NL)-WAVE)/(X1(NT)-WAVE))
            T(I)=1.-(1.-TINT(NT,1))/XX/XX/SQRT(XX)
         ELSE
            L=LOCATE(X1,NT,REAL(DOX(I,NL)))
            L=MAX(2,MIN(NT-2,L))
            XX=DOX(I,NL)
            VX1=X1(L)
            VX2=X1(L+1)
            VY1=TINT(L,1)
            VY2=TINT(L+1,1)
            T(I)=VY1*(XX-VX2)/(VX1-VX2)+VY2*(XX-VX1)/(VX2-VX1)
         ENDIF
      ENDDO
      FETCHED=.TRUE.
      RETURN
C
C     Error exit
C
590   WRITE (6,*)'Unable to interpolate'
      CLOSE (1)
      FETCHED=.FALSE.
      RETURN
600   WRITE (6,*)'Cannot find file '//FILE
      GO TO 590
      END


      SUBROUTINE MRQMIN2(A,MA,LISTA,MFIT,COVAR,ALPHA,NCA,CHISQ,ALAMDA,
     :                    FUNCS)
C
C     Chi-square minimization, from "Numerical Recipes"
C
      IMPLICIT REAL*8(A-H,O-Z)
C
C     Compiler parameter
C
      PARAMETER (MMAX=40)
C
C     Parameters
      DIMENSION A(MA), LISTA(MFIT), COVAR(NCA,NCA), ALPHA(NCA,NCA)
      DIMENSION ATRY(MMAX), BETA(MMAX), DA(MMAX)
C
      IF(ALAMDA.LT.0.)THEN
```

```
         KK=MFIT+1
         DO 12 J=1,MA
           IHIT=0
           DO 11 K=1,MFIT
             IF(LISTA(K).EQ.J)IHIT=IHIT+1
11         CONTINUE
           IF (IHIT.EQ.0) THEN
             LISTA(KK)=J
             KK=KK+1
           ELSE IF (IHIT.GT.1) THEN
             PAUSE 'Improper permutation in LISTA'
           ENDIF
12       CONTINUE
         IF (KK.NE.(MA+1)) PAUSE 'Improper permutation in LISTA'
         ALAMDA=0.001
         CALL MRQCOF2(A,MA,LISTA,MFIT,ALPHA,BETA,NCA,CHISQ,FUNCS)
         OCHISQ=CHISQ
         DO 13 J=1,MA
           ATRY(J)=A(J)
13       CONTINUE
       ENDIF
       DO 15 J=1,MFIT
         DO 14 K=1,MFIT
           COVAR(J,K)=ALPHA(J,K)
14       CONTINUE
         COVAR(J,J)=ALPHA(J,J)*(1.+ALAMDA)
         DA(J)=BETA(J)
15     CONTINUE
       CALL GAUSSJ(COVAR,MFIT,NCA,DA,1,1)
       IF(ALAMDA.EQ.0.)THEN
         CALL COVSRT(COVAR,NCA,MA,LISTA,MFIT)
         RETURN
       ENDIF
       DO 16 J=1,MFIT
         ATRY(LISTA(J))=ATRY(LISTA(J))+DA(J)
16     CONTINUE
       CALL MRQCOF2(ATRY,MA,LISTA,MFIT,COVAR,DA,NCA,CHISQ,FUNCS)
       IF(CHISQ.LT.OCHISQ)THEN
         ALAMDA=0.1*ALAMDA
         OCHISQ=CHISQ
         DO 18 J=1,MFIT
           DO 17 K=1,MFIT
             ALPHA(J,K)=COVAR(J,K)
17         CONTINUE
           BETA(J)=DA(J)
           A(LISTA(J))=ATRY(LISTA(J))
18       CONTINUE
       ELSE
         ALAMDA=10.*ALAMDA
         CHISQ=OCHISQ
       ENDIF
       RETURN
       END

       SUBROUTINE MRQCOF2(A,MA,LISTA,MFIT,ALPHA,BETA,NALP,CHISQ,FUNCS)
C
       IMPLICIT REAL*8 (A-H,O-Z)
C
       PARAMETER (ML=12, MMAX=40, MPIX=801)
       DIMENSION ALPHA(NALP,NALP),BETA(MA),DYDA(MMAX),LISTA(MFIT)
       DIMENSION A(MA)
C
C      Main data COMMON block
C
       INTEGER DPIX(ML), NLINES
       REAL*8 DOSIG(MPIX,ML), DOX(MPIX,ML), DOY(MPIX,ML)
       REAL*8 RT0000(MPIX,ML), RT0001(MPIX,ML), RT0010(MPIX,ML)
       REAL*8 RT0011(MPIX,ML), RT0100(MPIX,ML), RT0101(MPIX,ML)
       REAL*8 RT0110(MPIX,ML), RT0111(MPIX,ML), RT1000(MPIX,ML)
       REAL*8 RT1001(MPIX,ML), RT1010(MPIX,ML), RT1011(MPIX,ML)
       REAL*8 RT1100(MPIX,ML), RT1101(MPIX,ML), RT1110(MPIX,ML)
       REAL*8 RT1111(MPIX,ML)
       COMMON /MAIN/DPIX, NLINES, DOSIG, DOX, DOY, RT0000, RT0001,
      : RT0010, RT0011, RT0100, RT0101, RT0110, RT0111, RT1000,
      : RT1001, RT1010, RT1011, RT1100, RT1101, RT1110, RT1111
C
       DO 12 J=1,MFIT
         DO 11 K=1,J
           ALPHA(J,K)=0.
11       CONTINUE
```

```
          BETA(J)=0.
12      CONTINUE
        CHISQ=0.
        DO IL=1,NLINES
          DO 15 I=1,DPIX(IL)
            CALL FUNCS(IL,DOX(I,IL),A,YMOD,DYDA,MA)
            SIG2I=1./(DOSIG(I,IL)*DOSIG(I,IL))
            DY=DOY(I,IL)-YMOD
            DO 14 J=1,MFIT
              WT=DYDA(LISTA(J))*SIG2I
              DO 13 K=1,J
                ALPHA(J,K)=ALPHA(J,K)+WT*DYDA(LISTA(K))
13            CONTINUE
              BETA(J)=BETA(J)+DY*WT
14          CONTINUE
            CHISQ=CHISQ+DY*DY*SIG2I
15        CONTINUE
        ENDDO
        DO 17 J=2,MFIT
          DO 16 K=1,J-1
            ALPHA(K,J)=ALPHA(J,K)
16        CONTINUE
17      CONTINUE
        RETURN
        END

        SUBROUTINE PHYSFUNC(IL,XX,A,YMOD,DYDA,MA)
C
C       Model function subroutine for MRQMIN2 used in main
C       parameter fitting.
C
        IMPLICIT REAL*8 (A-H,O-Z)
C
C       Compiler parameters
C
        PARAMETER (ML=12, MPIX=801)
C
C       Parameters
C
        INTEGER IL, MA
        DIMENSION A(MA), DYDA(MA)
C
C       Main data COMMON block
C
        INTEGER DPIX(ML), NLINES
        REAL*8 DOSIG(MPIX,ML), DOX(MPIX,ML), DOY(MPIX,ML)
        REAL*8 RT0000(MPIX,ML), RT0001(MPIX,ML), RT0010(MPIX,ML)
        REAL*8 RT0011(MPIX,ML), RT0100(MPIX,ML), RT0101(MPIX,ML)
        REAL*8 RT0110(MPIX,ML), RT0111(MPIX,ML), RT1000(MPIX,ML)
        REAL*8 RT1001(MPIX,ML), RT1010(MPIX,ML), RT1011(MPIX,ML)
        REAL*8 RT1100(MPIX,ML), RT1101(MPIX,ML), RT1110(MPIX,ML)
        REAL*8 RT1111(MPIX,ML)
        COMMON /MAIN/DPIX, NLINES, DOSIG, DOX, DOY, RT0000, RT0001,
     : RT0010, RT0011, RT0100, RT0101, RT0110, RT0111, RT1000,
     : RT1001, RT1010, RT1011, RT1100, RT1101, RT1110, RT1111
C
C       Functions
C
        INTEGER DLOCATE
C
C       Start of executable statements
C
        DO I=1,MA
           DYDA(I)=0.
        ENDDO
        X=XX-A(3*IL)
        L=DLOCATE(DOX(1,IL),DPIX(IL),X)
        L=MIN(DPIX(IL)-2,MAX(2,L))
        X1=DOX(L-1,IL)
        X2=DOX(L,IL)
        X3=DOX(L+1,IL)
        X4=DOX(L+2,IL)
C
C       Calculate interpolant for each mesh point.
C
        Y1=RT0000(L-1,IL)
        Y2=RT0000(L,IL)
        Y3=RT0000(L+1,IL)
        Y4=RT0000(L+2,IL)
        Y000=Y1*(X-X2)*(X-X3)*(X-X4)/((X1-X2)*(X1-X3)*(X1-X4))+
```

```
:      Y2*(X-X1)*(X-X3)*(X-X4)/((X2-X1)*(X2-X3)*(X2-X4))+
:      Y3*(X-X1)*(X-X2)*(X-X4)/((X3-X1)*(X3-X2)*(X3-X4))+
:      Y4*(X-X1)*(X-X2)*(X-X3)/((X4-X1)*(X4-X2)*(X4-X3))
 DY000D3=-Y1*((X-X3)*(X-X4)+(X-X2)*(X-X4)+(X-X2)*(X-X3))/
:          ((X1-X2)*(X1-X3)*(X1-X4))+
:        -Y2*((X-X3)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X3))/
:          ((X2-X1)*(X2-X3)*(X2-X4))+
:        -Y3*((X-X2)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X2))/
:          ((X3-X1)*(X3-X2)*(X3-X4))+
:        -Y4*((X-X2)*(X-X3)+(X-X1)*(X-X3)+(X-X1)*(X-X2))/
:          ((X4-X1)*(X4-X2)*(X4-X3))
 Y1=RT0010(L-1,IL)
 Y2=RT0010(L,IL)
 Y3=RT0010(L+1,IL)
 Y4=RT0010(L+2,IL)
 Y001=Y1*(X-X2)*(X-X3)*(X-X4)/((X1-X2)*(X1-X3)*(X1-X4))+
:      Y2*(X-X1)*(X-X3)*(X-X4)/((X2-X1)*(X2-X3)*(X2-X4))+
:      Y3*(X-X1)*(X-X2)*(X-X4)/((X3-X1)*(X3-X2)*(X3-X4))+
:      Y4*(X-X1)*(X-X2)*(X-X3)/((X4-X1)*(X4-X2)*(X4-X3))
 DY001D3=-Y1*((X-X3)*(X-X4)+(X-X2)*(X-X4)+(X-X2)*(X-X3))/
:          ((X1-X2)*(X1-X3)*(X1-X4))+
:        -Y2*((X-X3)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X3))/
:          ((X2-X1)*(X2-X3)*(X2-X4))+
:        -Y3*((X-X2)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X2))/
:          ((X3-X1)*(X3-X2)*(X3-X4))+
:        -Y4*((X-X2)*(X-X3)+(X-X1)*(X-X3)+(X-X1)*(X-X2))/
:          ((X4-X1)*(X4-X2)*(X4-X3))
 Y1=RT0100(L-1,IL)
 Y2=RT0100(L,IL)
 Y3=RT0100(L+1,IL)
 Y4=RT0100(L+2,IL)
 Y010=Y1*(X-X2)*(X-X3)*(X-X4)/((X1-X2)*(X1-X3)*(X1-X4))+
:      Y2*(X-X1)*(X-X3)*(X-X4)/((X2-X1)*(X2-X3)*(X2-X4))+
:      Y3*(X-X1)*(X-X2)*(X-X4)/((X3-X1)*(X3-X2)*(X3-X4))+
:      Y4*(X-X1)*(X-X2)*(X-X3)/((X4-X1)*(X4-X2)*(X4-X3))
 DY010D3=-Y1*((X-X3)*(X-X4)+(X-X2)*(X-X4)+(X-X2)*(X-X3))/
:          ((X1-X2)*(X1-X3)*(X1-X4))+
:        -Y2*((X-X3)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X3))/
:          ((X2-X1)*(X2-X3)*(X2-X4))+
:        -Y3*((X-X2)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X2))/
:          ((X3-X1)*(X3-X2)*(X3-X4))+
:        -Y4*((X-X2)*(X-X3)+(X-X1)*(X-X3)+(X-X1)*(X-X2))/
:          ((X4-X1)*(X4-X2)*(X4-X3))
 Y1=RT0110(L-1,IL)
 Y2=RT0110(L,IL)
 Y3=RT0110(L+1,IL)
 Y4=RT0110(L+2,IL)
 Y011=Y1*(X-X2)*(X-X3)*(X-X4)/((X1-X2)*(X1-X3)*(X1-X4))+
:      Y2*(X-X1)*(X-X3)*(X-X4)/((X2-X1)*(X2-X3)*(X2-X4))+
:      Y3*(X-X1)*(X-X2)*(X-X4)/((X3-X1)*(X3-X2)*(X3-X4))+
:      Y4*(X-X1)*(X-X2)*(X-X3)/((X4-X1)*(X4-X2)*(X4-X3))
 DY011D3=-Y1*((X-X3)*(X-X4)+(X-X2)*(X-X4)+(X-X2)*(X-X3))/
:          ((X1-X2)*(X1-X3)*(X1-X4))+
:        -Y2*((X-X3)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X3))/
:          ((X2-X1)*(X2-X3)*(X2-X4))+
:        -Y3*((X-X2)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X2))/
:          ((X3-X1)*(X3-X2)*(X3-X4))+
:        -Y4*((X-X2)*(X-X3)+(X-X1)*(X-X3)+(X-X1)*(X-X2))/
:          ((X4-X1)*(X4-X2)*(X4-X3))
 Y1=RT1000(L-1,IL)
 Y2=RT1000(L,IL)
 Y3=RT1000(L+1,IL)
 Y4=RT1000(L+2,IL)
 Y100=Y1*(X-X2)*(X-X3)*(X-X4)/((X1-X2)*(X1-X3)*(X1-X4))+
:      Y2*(X-X1)*(X-X3)*(X-X4)/((X2-X1)*(X2-X3)*(X2-X4))+
:      Y3*(X-X1)*(X-X2)*(X-X4)/((X3-X1)*(X3-X2)*(X3-X4))+
:      Y4*(X-X1)*(X-X2)*(X-X3)/((X4-X1)*(X4-X2)*(X4-X3))
 DY100D3=-Y1*((X-X3)*(X-X4)+(X-X2)*(X-X4)+(X-X2)*(X-X3))/
:          ((X1-X2)*(X1-X3)*(X1-X4))+
:        -Y2*((X-X3)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X3))/
:          ((X2-X1)*(X2-X3)*(X2-X4))+
:        -Y3*((X-X2)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X2))/
:          ((X3-X1)*(X3-X2)*(X3-X4))+
:        -Y4*((X-X2)*(X-X3)+(X-X1)*(X-X3)+(X-X1)*(X-X2))/
:          ((X4-X1)*(X4-X2)*(X4-X3))
 Y1=RT1010(L-1,IL)
 Y2=RT1010(L,IL)
 Y3=RT1010(L+1,IL)
 Y4=RT1010(L+2,IL)
 Y101=Y1*(X-X2)*(X-X3)*(X-X4)/((X1-X2)*(X1-X3)*(X1-X4))+
```

```
     :      Y2*(X-X1)*(X-X3)*(X-X4)/((X2-X1)*(X2-X3)*(X2-X4))+
     :      Y3*(X-X1)*(X-X2)*(X-X4)/((X3-X1)*(X3-X2)*(X3-X4))+
     :      Y4*(X-X1)*(X-X2)*(X-X3)/((X4-X1)*(X4-X2)*(X4-X3))
      DY101D3=-Y1*((X-X3)*(X-X4)+(X-X2)*(X-X4)+(X-X2)*(X-X3))/
     :           ((X1-X2)*(X1-X3)*(X1-X4))+
     :        -Y2*((X-X3)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X3))/
     :           ((X2-X1)*(X2-X3)*(X2-X4))+
     :        -Y3*((X-X2)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X2))/
     :           ((X3-X1)*(X3-X2)*(X3-X4))+
     :        -Y4*((X-X2)*(X-X3)+(X-X1)*(X-X3)+(X-X1)*(X-X2))/
     :           ((X4-X1)*(X4-X2)*(X4-X3))
      Y1=RT1100(L-1,IL)
      Y2=RT1100(L,IL)
      Y3=RT1100(L+1,IL)
      Y4=RT1100(L+2,IL)
      Y110=Y1*(X-X2)*(X-X3)*(X-X4)/((X1-X2)*(X1-X3)*(X1-X4))+
     :      Y2*(X-X1)*(X-X3)*(X-X4)/((X2-X1)*(X2-X3)*(X2-X4))+
     :      Y3*(X-X1)*(X-X2)*(X-X4)/((X3-X1)*(X3-X2)*(X3-X4))+
     :      Y4*(X-X1)*(X-X2)*(X-X3)/((X4-X1)*(X4-X2)*(X4-X3))
      DY110D3=-Y1*((X-X3)*(X-X4)+(X-X2)*(X-X4)+(X-X2)*(X-X3))/
     :           ((X1-X2)*(X1-X3)*(X1-X4))+
     :        -Y2*((X-X3)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X3))/
     :           ((X2-X1)*(X2-X3)*(X2-X4))+
     :        -Y3*((X-X2)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X2))/
     :           ((X3-X1)*(X3-X2)*(X3-X4))+
     :        -Y4*((X-X2)*(X-X3)+(X-X1)*(X-X3)+(X-X1)*(X-X2))/
     :           ((X4-X1)*(X4-X2)*(X4-X3))
      Y1=RT1110(L-1,IL)
      Y2=RT1110(L,IL)
      Y3=RT1110(L+1,IL)
      Y4=RT1110(L+2,IL)
      Y111=Y1*(X-X2)*(X-X3)*(X-X4)/((X1-X2)*(X1-X3)*(X1-X4))+
     :      Y2*(X-X1)*(X-X3)*(X-X4)/((X2-X1)*(X2-X3)*(X2-X4))+
     :      Y3*(X-X1)*(X-X2)*(X-X4)/((X3-X1)*(X3-X2)*(X3-X4))+
     :      Y4*(X-X1)*(X-X2)*(X-X3)/((X4-X1)*(X4-X2)*(X4-X3))
      DY111D3=-Y1*((X-X3)*(X-X4)+(X-X2)*(X-X4)+(X-X2)*(X-X3))/
     :           ((X1-X2)*(X1-X3)*(X1-X4))+
     :        -Y2*((X-X3)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X3))/
     :           ((X2-X1)*(X2-X3)*(X2-X4))+
     :        -Y3*((X-X2)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X2))/
     :           ((X3-X1)*(X3-X2)*(X3-X4))+
     :        -Y4*((X-X2)*(X-X3)+(X-X1)*(X-X3)+(X-X1)*(X-X2))/
     :           ((X4-X1)*(X4-X2)*(X4-X3))
C
C     Now calcute YFIT
C
C     Abundance interpolant
C
      I=3*NLINES+3
      A1=1.-A(I)
      A2=A(I)
      Y00=A1*Y000+A2*Y001
      DY00D3=A1*DY000D3+A2*DY001D3
      DY00DA=Y001-Y000
      Y01=A1*Y010+A2*Y011
      DY01D3=A1*DY010D3+A2*DY011D3
      DY01DA=Y011-Y010
      Y10=A1*Y100+A2*Y101
      DY10D3=A1*DY100D3+A2*DY101D3
      DY10DA=Y101-Y100
      Y11=A1*Y110+A2*Y111
      DY11D3=A1*DY110D3+A2*DY111D3
      DY11DA=Y111-Y110
C
C     Gravity interpolant
C
      I=3*NLINES+2
      A1=1.-A(I)
      A2=A(I)
      Y0=A1*Y00+A2*Y01
      DY0D3=A1*DY00D3+A2*DY01D3
      DY0DA=A1*DY00DA+A2*DY01DA
      DY0DG=Y01-Y00
      Y1=A1*Y10+A2*Y11
      DY1D3=A1*DY10D3+A2*DY11D3
      DY1DA=A1*DY10DA+A2*DY11DA
      DY1DG=Y11-Y10
C
C     Temperature interpolant
C
```

```
      I=3*NLINES+1
      A1=1.-A(I)
      A2=A(I)
      Y=A1*Y0+A2*Y1
      DYD3=A1*DY0D3+A2*DY1D3
      DYDAA=A1*DY0DA+A2*DY1DA
      DYDG=A1*DY0DG+A2*DY1DG
      DYDT=Y1-Y0
C
C     Continuum fit
C
      A1=A(3*IL-2)
      A2=A(3*IL-1)
      YC=A1*X+A2
      YMOD=Y*YC
      DYDA(3*IL-2)=X*Y
      DYDA(3*IL-1)=Y
      DYDA(3*IL)=YC*DYD3
      DYDA(3*NLINES+1)=YC*DYDT
      DYDA(3*NLINES+2)=YC*DYDG
      DYDA(3*NLINES+3)=YC*DYDAA
      RETURN
      END


      SUBROUTINE MATINV(A,N,NR)
C
C     Invert matrix in place by LU decomposition; routine
C     originally from Auer-Mihalas codes.
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      INTEGER N, K, NR, I, L, KO, II, J, JJ
      DIMENSION A(NR,NR)
C
      DO 60 I=2,N
         DIV=A(1,1)
         A(I,1)=A(I,1)/DIV
         DO 30 J=2,I-1
            DIV=A(J,J)
            SUM=0.
            DO 10 L=1,J-1
               SUM=SUM+A(I,L)*A(L,J)
10          CONTINUE
            A(I,J)=(A(I,J)-SUM)/DIV
30       CONTINUE
         DO 50 J=I,N
            SUM=0.
            DO 40 L=1,I-1
               SUM=SUM+A(I,L)*A(L,J)
40          CONTINUE
            A(I,J)=A(I,J)-SUM
50       CONTINUE
60    CONTINUE
      DO 100 I=N,2,-1
         DO 90 J=I-1,1,-1
            SUM=0.
            IF (J+1.EQ.I)GO TO 80
            DO 70 K=J+1,I-1
               SUM=SUM+A(I,K)*A(K,J)
70          CONTINUE
80          A(I,J)=-A(I,J)-SUM
90       CONTINUE
100   CONTINUE
      A(N,N)=1.0/A(N,N)
      DO 140 I=N-1,1,-1
         DIV=A(I,I)
         DO 120 J=N,I+1,-1
            SUM=0.
            DO 110 K=I+1,J
               SUM=SUM+A(I,K)*A(K,J)
110         CONTINUE
            A(I,J)=-SUM/DIV
120      CONTINUE
         A(I,I)=1.0/A(I,I)
140   CONTINUE
      DO 200 I=1,N
         DO 160 J=1,I-1
            SUM=0.0
            DO 150 K=I,N
               SUM=SUM+A(I,K)*A(K,J)
```

```
150         CONTINUE
            A(I,J)=SUM
160      CONTINUE
         DO 190 J=I,N
            SUM=A(I,J)
            IF (J.EQ.N)GO TO 180
            DO 170 K=J+1,N
               SUM=SUM+A(I,K)*A(K,J)
170         CONTINUE
180         A(I,J)=SUM
190      CONTINUE
200   CONTINUE
      RETURN
      END


      SUBROUTINE ROTFUNC(IL,XX,A,YMOD,DYDA,MA)
C
C     Model function subroutine for MRQMIN2 used in
C     rotation fit
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
C     Compiler parameters
C
      PARAMETER (ML=12, MPIX=801)
C
C     Parameters
C
      INTEGER IL, MA
      DIMENSION A(MA), DYDA(MA)
C
C     Main data COMMON block
C
      INTEGER DPIX(ML), NLINES
      REAL*8 DOSIG(MPIX,ML), DOX(MPIX,ML), DOY(MPIX,ML)
      REAL*8 RT0000(MPIX,ML), RT0001(MPIX,ML), RT0010(MPIX,ML)
      REAL*8 RT0011(MPIX,ML), RT0100(MPIX,ML), RT0101(MPIX,ML)
      REAL*8 RT0110(MPIX,ML), RT0111(MPIX,ML), RT1000(MPIX,ML)
      REAL*8 RT1001(MPIX,ML), RT1010(MPIX,ML), RT1011(MPIX,ML)
      REAL*8 RT1100(MPIX,ML), RT1101(MPIX,ML), RT1110(MPIX,ML)
      REAL*8 RT1111(MPIX,ML)
      COMMON /MAIN/DPIX, NLINES, DOSIG, DOX, DOY, RT0000, RT0001,
     : RT0010, RT0011, RT0100, RT0101, RT0110, RT0111, RT1000,
     : RT1001, RT1010, RT1011, RT1100, RT1101, RT1110, RT1111
C
C     Functions
C
      INTEGER DLOCATE
C
C     Start of executable statements
C
      DO I=1,MA
         DYDA(I)=0.
      ENDDO
      X=XX-A(3*IL)
      L=DLOCATE(DOX(1,IL),DPIX(IL),X)
      L=MIN(DPIX(IL)-2,MAX(2,L))
      X1=DOX(L-1,IL)
      X2=DOX(L,IL)
      X3=DOX(L+1,IL)
      X4=DOX(L+2,IL)
C
C     Calculate interpolant for each mesh point.
C
      Y1=RT0000(L-1,IL)
      Y2=RT0000(L,IL)
      Y3=RT0000(L+1,IL)
      Y4=RT0000(L+2,IL)
      Y0000=Y1*(X-X2)*(X-X3)*(X-X4)/((X1-X2)*(X1-X3)*(X1-X4))+
     :      Y2*(X-X1)*(X-X3)*(X-X4)/((X2-X1)*(X2-X3)*(X2-X4))+
     :      Y3*(X-X1)*(X-X2)*(X-X4)/((X3-X1)*(X3-X2)*(X3-X4))+
     :      Y4*(X-X1)*(X-X2)*(X-X3)/((X4-X1)*(X4-X2)*(X4-X3))
      DY0000D3=-Y1*((X-X3)*(X-X4)+(X-X2)*(X-X4)+(X-X2)*(X-X3))/
     :             ((X1-X2)*(X1-X3)*(X1-X4))+
     :          -Y2*((X-X3)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X3))/
     :             ((X2-X1)*(X2-X3)*(X2-X4))+
     :          -Y3*((X-X2)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X2))/
     :             ((X3-X1)*(X3-X2)*(X3-X4))+
     :          -Y4*((X-X2)*(X-X3)+(X-X1)*(X-X3)+(X-X1)*(X-X2))/
     :             ((X4-X1)*(X4-X2)*(X4-X3))
```

```
      Y1=RT0001(L-1,IL)
      Y2=RT0001(L,IL)
      Y3=RT0001(L+1,IL)
      Y4=RT0001(L+2,IL)
      Y0001=Y1*(X-X2)*(X-X3)*(X-X4)/((X1-X2)*(X1-X3)*(X1-X4))+
     :      Y2*(X-X1)*(X-X3)*(X-X4)/((X2-X1)*(X2-X3)*(X2-X4))+
     :      Y3*(X-X1)*(X-X2)*(X-X4)/((X3-X1)*(X3-X2)*(X3-X4))+
     :      Y4*(X-X1)*(X-X2)*(X-X3)/((X4-X1)*(X4-X2)*(X4-X3))
      DY0001D3=-Y1*((X-X3)*(X-X4)+(X-X2)*(X-X4)+(X-X2)*(X-X3))/
     :            ((X1-X2)*(X1-X3)*(X1-X4))+
     :         -Y2*((X-X3)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X3))/
     :            ((X2-X1)*(X2-X3)*(X2-X4))+
     :         -Y3*((X-X2)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X2))/
     :            ((X3-X1)*(X3-X2)*(X3-X4))+
     :         -Y4*((X-X2)*(X-X3)+(X-X1)*(X-X3)+(X-X1)*(X-X2))/
     :            ((X4-X1)*(X4-X2)*(X4-X3))
      Y1=RT0010(L-1,IL)
      Y2=RT0010(L,IL)
      Y3=RT0010(L+1,IL)
      Y4=RT0010(L+2,IL)
      Y0010=Y1*(X-X2)*(X-X3)*(X-X4)/((X1-X2)*(X1-X3)*(X1-X4))+
     :      Y2*(X-X1)*(X-X3)*(X-X4)/((X2-X1)*(X2-X3)*(X2-X4))+
     :      Y3*(X-X1)*(X-X2)*(X-X4)/((X3-X1)*(X3-X2)*(X3-X4))+
     :      Y4*(X-X1)*(X-X2)*(X-X3)/((X4-X1)*(X4-X2)*(X4-X3))
      DY0010D3=-Y1*((X-X3)*(X-X4)+(X-X2)*(X-X4)+(X-X2)*(X-X3))/
     :            ((X1-X2)*(X1-X3)*(X1-X4))+
     :         -Y2*((X-X3)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X3))/
     :            ((X2-X1)*(X2-X3)*(X2-X4))+
     :         -Y3*((X-X2)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X2))/
     :            ((X3-X1)*(X3-X2)*(X3-X4))+
     :         -Y4*((X-X2)*(X-X3)+(X-X1)*(X-X3)+(X-X1)*(X-X2))/
     :            ((X4-X1)*(X4-X2)*(X4-X3))
      Y1=RT0011(L-1,IL)
      Y2=RT0011(L,IL)
      Y3=RT0011(L+1,IL)
      Y4=RT0011(L+2,IL)
      Y0011=Y1*(X-X2)*(X-X3)*(X-X4)/((X1-X2)*(X1-X3)*(X1-X4))+
     :      Y2*(X-X1)*(X-X3)*(X-X4)/((X2-X1)*(X2-X3)*(X2-X4))+
     :      Y3*(X-X1)*(X-X2)*(X-X4)/((X3-X1)*(X3-X2)*(X3-X4))+
     :      Y4*(X-X1)*(X-X2)*(X-X3)/((X4-X1)*(X4-X2)*(X4-X3))
      DY0011D3=-Y1*((X-X3)*(X-X4)+(X-X2)*(X-X4)+(X-X2)*(X-X3))/
     :            ((X1-X2)*(X1-X3)*(X1-X4))+
     :         -Y2*((X-X3)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X3))/
     :            ((X2-X1)*(X2-X3)*(X2-X4))+
     :         -Y3*((X-X2)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X2))/
     :            ((X3-X1)*(X3-X2)*(X3-X4))+
     :         -Y4*((X-X2)*(X-X3)+(X-X1)*(X-X3)+(X-X1)*(X-X2))/
     :            ((X4-X1)*(X4-X2)*(X4-X3))
      Y1=RT0100(L-1,IL)
      Y2=RT0100(L,IL)
      Y3=RT0100(L+1,IL)
      Y4=RT0100(L+2,IL)
      Y0100=Y1*(X-X2)*(X-X3)*(X-X4)/((X1-X2)*(X1-X3)*(X1-X4))+
     :      Y2*(X-X1)*(X-X3)*(X-X4)/((X2-X1)*(X2-X3)*(X2-X4))+
     :      Y3*(X-X1)*(X-X2)*(X-X4)/((X3-X1)*(X3-X2)*(X3-X4))+
     :      Y4*(X-X1)*(X-X2)*(X-X3)/((X4-X1)*(X4-X2)*(X4-X3))
      DY0100D3=-Y1*((X-X3)*(X-X4)+(X-X2)*(X-X4)+(X-X2)*(X-X3))/
     :            ((X1-X2)*(X1-X3)*(X1-X4))+
     :         -Y2*((X-X3)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X3))/
     :            ((X2-X1)*(X2-X3)*(X2-X4))+
     :         -Y3*((X-X2)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X2))/
     :            ((X3-X1)*(X3-X2)*(X3-X4))+
     :         -Y4*((X-X2)*(X-X3)+(X-X1)*(X-X3)+(X-X1)*(X-X2))/
     :            ((X4-X1)*(X4-X2)*(X4-X3))
      Y1=RT0101(L-1,IL)
      Y2=RT0101(L,IL)
      Y3=RT0101(L+1,IL)
      Y4=RT0101(L+2,IL)
      Y0101=Y1*(X-X2)*(X-X3)*(X-X4)/((X1-X2)*(X1-X3)*(X1-X4))+
     :      Y2*(X-X1)*(X-X3)*(X-X4)/((X2-X1)*(X2-X3)*(X2-X4))+
     :      Y3*(X-X1)*(X-X2)*(X-X4)/((X3-X1)*(X3-X2)*(X3-X4))+
     :      Y4*(X-X1)*(X-X2)*(X-X3)/((X4-X1)*(X4-X2)*(X4-X3))
      DY0101D3=-Y1*((X-X3)*(X-X4)+(X-X2)*(X-X4)+(X-X2)*(X-X3))/
     :            ((X1-X2)*(X1-X3)*(X1-X4))+
     :         -Y2*((X-X3)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X3))/
     :            ((X2-X1)*(X2-X3)*(X2-X4))+
     :         -Y3*((X-X2)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X2))/
     :            ((X3-X1)*(X3-X2)*(X3-X4))+
     :         -Y4*((X-X2)*(X-X3)+(X-X1)*(X-X3)+(X-X1)*(X-X2))/
     :            ((X4-X1)*(X4-X2)*(X4-X3))
```

```
 Y1=RT0110(L-1,IL)
 Y2=RT0110(L,IL)
 Y3=RT0110(L+1,IL)
 Y4=RT0110(L+2,IL)
 Y0110=Y1*(X-X2)*(X-X3)*(X-X4)/((X1-X2)*(X1-X3)*(X1-X4))+
:      Y2*(X-X1)*(X-X3)*(X-X4)/((X2-X1)*(X2-X3)*(X2-X4))+
:      Y3*(X-X1)*(X-X2)*(X-X4)/((X3-X1)*(X3-X2)*(X3-X4))+
:      Y4*(X-X1)*(X-X2)*(X-X3)/((X4-X1)*(X4-X2)*(X4-X3))
 DY0110D3=-Y1*((X-X3)*(X-X4)+(X-X2)*(X-X4)+(X-X2)*(X-X3))/
:           ((X1-X2)*(X1-X3)*(X1-X4))+
:         -Y2*((X-X3)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X3))/
:           ((X2-X1)*(X2-X3)*(X2-X4))+
:         -Y3*((X-X2)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X2))/
:           ((X3-X1)*(X3-X2)*(X3-X4))+
:         -Y4*((X-X2)*(X-X3)+(X-X1)*(X-X3)+(X-X1)*(X-X2))/
:           ((X4-X1)*(X4-X2)*(X4-X3))
 Y1=RT0111(L-1,IL)
 Y2=RT0111(L,IL)
 Y3=RT0111(L+1,IL)
 Y4=RT0111(L+2,IL)
 Y0111=Y1*(X-X2)*(X-X3)*(X-X4)/((X1-X2)*(X1-X3)*(X1-X4))+
:      Y2*(X-X1)*(X-X3)*(X-X4)/((X2-X1)*(X2-X3)*(X2-X4))+
:      Y3*(X-X1)*(X-X2)*(X-X4)/((X3-X1)*(X3-X2)*(X3-X4))+
:      Y4*(X-X1)*(X-X2)*(X-X3)/((X4-X1)*(X4-X2)*(X4-X3))
 DY0111D3=-Y1*((X-X3)*(X-X4)+(X-X2)*(X-X4)+(X-X2)*(X-X3))/
:           ((X1-X2)*(X1-X3)*(X1-X4))+
:         -Y2*((X-X3)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X3))/
:           ((X2-X1)*(X2-X3)*(X2-X4))+
:         -Y3*((X-X2)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X2))/
:           ((X3-X1)*(X3-X2)*(X3-X4))+
:         -Y4*((X-X2)*(X-X3)+(X-X1)*(X-X3)+(X-X1)*(X-X2))/
:           ((X4-X1)*(X4-X2)*(X4-X3))
 Y1=RT1000(L-1,IL)
 Y2=RT1000(L,IL)
 Y3=RT1000(L+1,IL)
 Y4=RT1000(L+2,IL)
 Y1000=Y1*(X-X2)*(X-X3)*(X-X4)/((X1-X2)*(X1-X3)*(X1-X4))+
:      Y2*(X-X1)*(X-X3)*(X-X4)/((X2-X1)*(X2-X3)*(X2-X4))+
:      Y3*(X-X1)*(X-X2)*(X-X4)/((X3-X1)*(X3-X2)*(X3-X4))+
:      Y4*(X-X1)*(X-X2)*(X-X3)/((X4-X1)*(X4-X2)*(X4-X3))
 DY1000D3=-Y1*((X-X3)*(X-X4)+(X-X2)*(X-X4)+(X-X2)*(X-X3))/
:           ((X1-X2)*(X1-X3)*(X1-X4))+
:         -Y2*((X-X3)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X3))/
:           ((X2-X1)*(X2-X3)*(X2-X4))+
:         -Y3*((X-X2)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X2))/
:           ((X3-X1)*(X3-X2)*(X3-X4))+
:         -Y4*((X-X2)*(X-X3)+(X-X1)*(X-X3)+(X-X1)*(X-X2))/
:           ((X4-X1)*(X4-X2)*(X4-X3))
 Y1=RT1001(L-1,IL)
 Y2=RT1001(L,IL)
 Y3=RT1001(L+1,IL)
 Y4=RT1001(L+2,IL)
 Y1001=Y1*(X-X2)*(X-X3)*(X-X4)/((X1-X2)*(X1-X3)*(X1-X4))+
:      Y2*(X-X1)*(X-X3)*(X-X4)/((X2-X1)*(X2-X3)*(X2-X4))+
:      Y3*(X-X1)*(X-X2)*(X-X4)/((X3-X1)*(X3-X2)*(X3-X4))+
:      Y4*(X-X1)*(X-X2)*(X-X3)/((X4-X1)*(X4-X2)*(X4-X3))
 DY1001D3=-Y1*((X-X3)*(X-X4)+(X-X2)*(X-X4)+(X-X2)*(X-X3))/
:           ((X1-X2)*(X1-X3)*(X1-X4))+
:         -Y2*((X-X3)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X3))/
:           ((X2-X1)*(X2-X3)*(X2-X4))+
:         -Y3*((X-X2)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X2))/
:           ((X3-X1)*(X3-X2)*(X3-X4))+
:         -Y4*((X-X2)*(X-X3)+(X-X1)*(X-X3)+(X-X1)*(X-X2))/
:           ((X4-X1)*(X4-X2)*(X4-X3))
 Y1=RT1010(L-1,IL)
 Y2=RT1010(L,IL)
 Y3=RT1010(L+1,IL)
 Y4=RT1010(L+2,IL)
 Y1010=Y1*(X-X2)*(X-X3)*(X-X4)/((X1-X2)*(X1-X3)*(X1-X4))+
:      Y2*(X-X1)*(X-X3)*(X-X4)/((X2-X1)*(X2-X3)*(X2-X4))+
:      Y3*(X-X1)*(X-X2)*(X-X4)/((X3-X1)*(X3-X2)*(X3-X4))+
:      Y4*(X-X1)*(X-X2)*(X-X3)/((X4-X1)*(X4-X2)*(X4-X3))
 DY1010D3=-Y1*((X-X3)*(X-X4)+(X-X2)*(X-X4)+(X-X2)*(X-X3))/
:           ((X1-X2)*(X1-X3)*(X1-X4))+
:         -Y2*((X-X3)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X3))/
:           ((X2-X1)*(X2-X3)*(X2-X4))+
:         -Y3*((X-X2)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X2))/
:           ((X3-X1)*(X3-X2)*(X3-X4))+
:         -Y4*((X-X2)*(X-X3)+(X-X1)*(X-X3)+(X-X1)*(X-X2))/
:           ((X4-X1)*(X4-X2)*(X4-X3))
```

```
 Y1=RT1011(L-1,IL)
 Y2=RT1011(L,IL)
 Y3=RT1011(L+1,IL)
 Y4=RT1011(L+2,IL)
 Y1011=Y1*(X-X2)*(X-X3)*(X-X4)/((X1-X2)*(X1-X3)*(X1-X4))+
:      Y2*(X-X1)*(X-X3)*(X-X4)/((X2-X1)*(X2-X3)*(X2-X4))+
:      Y3*(X-X1)*(X-X2)*(X-X4)/((X3-X1)*(X3-X2)*(X3-X4))+
:      Y4*(X-X1)*(X-X2)*(X-X3)/((X4-X1)*(X4-X2)*(X4-X3))
 DY1011D3=-Y1*((X-X3)*(X-X4)+(X-X2)*(X-X4)+(X-X2)*(X-X3))/
:           ((X1-X2)*(X1-X3)*(X1-X4))+
:         -Y2*((X-X3)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X3))/
:           ((X2-X1)*(X2-X3)*(X2-X4))+
:         -Y3*((X-X2)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X2))/
:           ((X3-X1)*(X3-X2)*(X3-X4))+
:         -Y4*((X-X2)*(X-X3)+(X-X1)*(X-X3)+(X-X1)*(X-X2))/
:           ((X4-X1)*(X4-X2)*(X4-X3))
 Y1=RT1100(L-1,IL)
 Y2=RT1100(L,IL)
 Y3=RT1100(L+1,IL)
 Y4=RT1100(L+2,IL)
 Y1100=Y1*(X-X2)*(X-X3)*(X-X4)/((X1-X2)*(X1-X3)*(X1-X4))+
:      Y2*(X-X1)*(X-X3)*(X-X4)/((X2-X1)*(X2-X3)*(X2-X4))+
:      Y3*(X-X1)*(X-X2)*(X-X4)/((X3-X1)*(X3-X2)*(X3-X4))+
:      Y4*(X-X1)*(X-X2)*(X-X3)/((X4-X1)*(X4-X2)*(X4-X3))
 DY1100D3=-Y1*((X-X3)*(X-X4)+(X-X2)*(X-X4)+(X-X2)*(X-X3))/
:           ((X1-X2)*(X1-X3)*(X1-X4))+
:         -Y2*((X-X3)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X3))/
:           ((X2-X1)*(X2-X3)*(X2-X4))+
:         -Y3*((X-X2)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X2))/
:           ((X3-X1)*(X3-X2)*(X3-X4))+
:         -Y4*((X-X2)*(X-X3)+(X-X1)*(X-X3)+(X-X1)*(X-X2))/
:           ((X4-X1)*(X4-X2)*(X4-X3))
 Y1=RT1101(L-1,IL)
 Y2=RT1101(L,IL)
 Y3=RT1101(L+1,IL)
 Y4=RT1101(L+2,IL)
 Y1101=Y1*(X-X2)*(X-X3)*(X-X4)/((X1-X2)*(X1-X3)*(X1-X4))+
:      Y2*(X-X1)*(X-X3)*(X-X4)/((X2-X1)*(X2-X3)*(X2-X4))+
:      Y3*(X-X1)*(X-X2)*(X-X4)/((X3-X1)*(X3-X2)*(X3-X4))+
:      Y4*(X-X1)*(X-X2)*(X-X3)/((X4-X1)*(X4-X2)*(X4-X3))
 DY1101D3=-Y1*((X-X3)*(X-X4)+(X-X2)*(X-X4)+(X-X2)*(X-X3))/
:           ((X1-X2)*(X1-X3)*(X1-X4))+
:         -Y2*((X-X3)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X3))/
:           ((X2-X1)*(X2-X3)*(X2-X4))+
:         -Y3*((X-X2)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X2))/
:           ((X3-X1)*(X3-X2)*(X3-X4))+
:         -Y4*((X-X2)*(X-X3)+(X-X1)*(X-X3)+(X-X1)*(X-X2))/
:           ((X4-X1)*(X4-X2)*(X4-X3))
 Y1=RT1110(L-1,IL)
 Y2=RT1110(L,IL)
 Y3=RT1110(L+1,IL)
 Y4=RT1110(L+2,IL)
 Y1110=Y1*(X-X2)*(X-X3)*(X-X4)/((X1-X2)*(X1-X3)*(X1-X4))+
:      Y2*(X-X1)*(X-X3)*(X-X4)/((X2-X1)*(X2-X3)*(X2-X4))+
:      Y3*(X-X1)*(X-X2)*(X-X4)/((X3-X1)*(X3-X2)*(X3-X4))+
:      Y4*(X-X1)*(X-X2)*(X-X3)/((X4-X1)*(X4-X2)*(X4-X3))
 DY1110D3=-Y1*((X-X3)*(X-X4)+(X-X2)*(X-X4)+(X-X2)*(X-X3))/
:           ((X1-X2)*(X1-X3)*(X1-X4))+
:         -Y2*((X-X3)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X3))/
:           ((X2-X1)*(X2-X3)*(X2-X4))+
:         -Y3*((X-X2)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X2))/
:           ((X3-X1)*(X3-X2)*(X3-X4))+
:         -Y4*((X-X2)*(X-X3)+(X-X1)*(X-X3)+(X-X1)*(X-X2))/
:           ((X4-X1)*(X4-X2)*(X4-X3))
 Y1=RT1111(L-1,IL)
 Y2=RT1111(L,IL)
 Y3=RT1111(L+1,IL)
 Y4=RT1111(L+2,IL)
 Y1111=Y1*(X-X2)*(X-X3)*(X-X4)/((X1-X2)*(X1-X3)*(X1-X4))+
:      Y2*(X-X1)*(X-X3)*(X-X4)/((X2-X1)*(X2-X3)*(X2-X4))+
:      Y3*(X-X1)*(X-X2)*(X-X4)/((X3-X1)*(X3-X2)*(X3-X4))+
:      Y4*(X-X1)*(X-X2)*(X-X3)/((X4-X1)*(X4-X2)*(X4-X3))
 DY1111D3=-Y1*((X-X3)*(X-X4)+(X-X2)*(X-X4)+(X-X2)*(X-X3))/
:           ((X1-X2)*(X1-X3)*(X1-X4))+
:         -Y2*((X-X3)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X3))/
:           ((X2-X1)*(X2-X3)*(X2-X4))+
:         -Y3*((X-X2)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X2))/
:           ((X3-X1)*(X3-X2)*(X3-X4))+
:         -Y4*((X-X2)*(X-X3)+(X-X1)*(X-X3)+(X-X1)*(X-X2))/
:           ((X4-X1)*(X4-X2)*(X4-X3))
```

```
C
C     Now calcute YFIT
C
C     Rotation interpolant
C
      I=3*NLINES+4
      A1=1.-A(I)
      A2=A(I)
      Y000=A1*Y0000+A2*Y0001
      DY000D3=A1*DY0000D3+A2*DY0001D3
      DY000DR=Y0001-Y0000
      Y001=A1*Y0010+A2*Y0011
      DY001D3=A1*DY0010D3+A2*DY0011D3
      DY001DR=Y0011-Y0010
      Y010=A1*Y0100+A2*Y0101
      DY010D3=A1*DY0100D3+A2*DY0101D3
      DY010DR=Y0101-Y0100
      Y011=A1*Y0110+A2*Y0111
      DY011D3=A1*DY0110D3+A2*DY0111D3
      DY011DR=Y0111-Y0110
      Y100=A1*Y1000+A2*Y1001
      DY100D3=A1*DY1000D3+A2*DY1001D3
      DY100DR=Y1001-Y1000
      Y101=A1*Y1010+A2*Y1011
      DY101D3=A1*DY1010D3+A2*DY1011D3
      DY101DR=Y1011-Y1010
      Y110=A1*Y1100+A2*Y1101
      DY110D3=A1*DY1100D3+A2*DY1101D3
      DY110DR=Y1101-Y1100
      Y111=A1*Y1110+A2*Y1111
      DY111D3=A1*DY1110D3+A2*DY1111D3
      DY111DR=Y1111-Y1110
C
C     Abundance interpolant
C
      I=3*NLINES+3
      A1=1.-A(I)
      A2=A(I)
      Y00=A1*Y000+A2*Y001
      DY00D3=A1*DY000D3+A2*DY001D3
      DY00DR=A1*DY000DR+A2*DY001DR
      Y01=A1*Y010+A2*Y011
      DY01D3=A1*DY010D3+A2*DY011D3
      DY01DR=A1*DY010DR+A2*DY011DR
      Y10=A1*Y100+A2*Y101
      DY10D3=A1*DY100D3+A2*DY101D3
      DY10DR=A1*DY100DR+A2*DY101DR
      Y11=A1*Y110+A2*Y111
      DY11D3=A1*DY110D3+A2*DY111D3
      DY11DR=A1*DY110DR+A2*DY111DR
C
C     Gravity interpolant
C
      I=3*NLINES+2
      A1=1.-A(I)
      A2=A(I)
      Y0=A1*Y00+A2*Y01
      DY0D3=A1*DY00D3+A2*DY01D3
      DY0DR=A1*DY00DR+A2*DY01DR
      Y1=A1*Y10+A2*Y11
      DY1D3=A1*DY10D3+A2*DY11D3
      DY1DR=A1*DY10DR+A2*DY11DR
C
C     Temperature interpolant
C
      I=3*NLINES+1
      A1=1.-A(I)
      A2=A(I)
      Y=A1*Y0+A2*Y1
      DYD3=A1*DY0D3+A2*DY1D3
      DYDR=A1*DY0DR+A2*DY1DR
C
C     Continuum fit
C
      A1=A(3*IL-2)
      A2=A(3*IL-1)
      YC=A1*X+A2
      YMOD=Y*YC
      DYDA(3*IL-2)=X*Y
      DYDA(3*IL-1)=Y
```

```
      DYDA(3*IL)=YC*DYD3
      DYDA(3*NLINES+4)=YC*DYDR
      RETURN
      END


      SUBROUTINE PLOTFUNC(XX,A,YFIT,DYDA,MA)
C
C     Generate continuum parameters for fit to line profiles.
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER PLN, PLI
      DIMENSION A(5),DYDA(3), DXPLOT(801), DYPLOT(801)
C
C     Externals
C
      INTEGER DLOCATE
      EXTERNAL DLOCATE
C
C     PLOTFUNCA common block
C
      COMMON /PLOTFUNC/DXPLOT, DYPLOT, PLN
C
C     Start of executable statements
C
      X=XX-A(3)
      L=DLOCATE(DXPLOT,PLN,X)
      L=MIN(PLN-2,MAX(2,L))
      X1=DXPLOT(L-1)
      X2=DXPLOT(L)
      X3=DXPLOT(L+1)
      X4=DXPLOT(L+2)
      Y1=DYPLOT(L-1)
      Y2=DYPLOT(L)
      Y3=DYPLOT(L+1)
      Y4=DYPLOT(L+2)
      YMOD=Y1*(X-X2)*(X-X3)*(X-X4)/((X1-X2)*(X1-X3)*(X1-X4))+
     :      Y2*(X-X1)*(X-X3)*(X-X4)/((X2-X1)*(X2-X3)*(X2-X4))+
     :      Y3*(X-X1)*(X-X2)*(X-X4)/((X3-X1)*(X3-X2)*(X3-X4))+
     :      Y4*(X-X1)*(X-X2)*(X-X3)/((X4-X1)*(X4-X2)*(X4-X3))
      DYMODD3=-Y1*((X-X3)*(X-X4)+(X-X2)*(X-X4)+(X-X2)*(X-X3))/
     :           ((X1-X2)*(X1-X3)*(X1-X4))+
     :        -Y2*((X-X3)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X3))/
     :           ((X2-X1)*(X2-X3)*(X2-X4))+
     :        -Y3*((X-X2)*(X-X4)+(X-X1)*(X-X4)+(X-X1)*(X-X2))/
     :           ((X3-X1)*(X3-X2)*(X3-X4))+
     :        -Y4*((X-X2)*(X-X3)+(X-X1)*(X-X3)+(X-X1)*(X-X2))/
     :           ((X4-X1)*(X4-X2)*(X4-X3))
      CONT=A(1)*X+A(2)
      YFIT=YMOD*CONT
      DYDA(1)=X*YMOD
      DYDA(2)=YMOD
      DYDA(3)=DYMODD3*CONT
      RETURN
      END
```