Accepted: DECEMBER 12, 2008
Preprint typeset using LaTeX style emulateapj v. 12/01/06

MODELING OF STELLAR ATMOSPHERES

SEBASTIAN TRUJILLO[1],
Accepted: December 12, 2008

ABSTRACT

We develop and implement a 1-D, plane-parallel, stellar atmosphere model in local thermodynamic equilibrium using the Grey case in Eddington's approximation. We obtain the relevant physical parameters and the ionization fractions and excitation ratios of all the species as a function of optical depth into the atmosphere in the range $10^{-3} < \tau < 10^{3}$, for which the atmosphere thickness is negligible compared to the star's radius. The Grey flux is also computed for several representative layers. Using mass fractions $X = 0.70, Y = 0.28, Z = 0.02$, the model is applied to a solar type star with an atmosphere composed entirely of H, He and C. A possible candidate for a PopIII star, a large mass and very low metallicity O star is also analyzed using the model.

Subject headings:

1. INTRODUCTION

In Astronomy and all other sciences, the complexity of the physical system under study usually dictates the extent to which it can be represented analytically and the onset of the need to use numerical computer models to reproduce the observations. Even though each of the many physical processes that take place in the atmosphere of a star have been known for decades and are well understood, the complex interplay of these processes and their interdependence leave little hope of using simple analytical models to describe the conditions inferred from spectroscopic observations of real stars. There comes a point as the observational tools become more accurate, where the interpretation of the data becomes so sensitive to the details of the physical processes that a crude model is no longer sufficient to discern between different scenarios.

In this work, we set out to implement the first step towards "realistic" modeling of a stellar atmosphere. We use both macro and microscopic physical laws, whichever is best suited for describing the rough behavior of the gas in a computationally inexpensive way. For instance, the pressure balance across each thin layer of the atmosphere is controlled by the macroscopic description of fluid parcels in hydrostatic equilibrium, whereas the opacity of each layer is determined by the abundance of each species modulated by the ionization fractions that themselves depend critically on the electron content of the gas and the temperature.

In the next section we describe the physical processes, assumptions and approximations that went into the model, and briefly describe the computational tools that were employed in the coding. Some preliminary convergence analysis are discussed in § 3. The modeled physical behavior of a solar type and a low metallicity high mass (i.e. Pop III) star is analyzed in § 4, including some comparison with observed values and departures due to simplifying approximations. Lastly, conclusions are presented in § 5.

2. THE MODEL

In our model the plane parallel atmosphere of the star is subdivided into many adyacent layers at regular intervals in optical depth (in logarithmic space). In Eddington's approximation, the intensity of radiation from a single layer is approximated by the sum of two "streams", where the amplitude of each is constant over its corresponding hemisphere:

$$I_\lambda(\tau_\lambda) = I^{in}_\lambda(\tau_\lambda) + I^{out}_\lambda(\tau_\lambda), \tag{1}$$

where *in* refers to radiation traveling towards the center of the star and *out* refers to radiation traveling towards the surface. Further approximating, we assume the Grey case where all the wavelength-dependent quantities are replaced by their average over the full spectrum. Solving for the moments of the radiative tranfer equation yields a relationship between the optical depth of a layer and the flux emanating from it. Integrating the Plank radiation law we obtain the temperature of a layer as a function of its optical depth,

$$T^4(\tau) = \frac{3}{4} T^4_{eff}[\tau + q(\tau)], \tag{2}$$

where $q(\tau)$ is known as the Hopf function, T_{eff} is the effective or observed black-body temperature of the photosphere and the optical depth is constant for all wavelengths. The Hopf function is a slowly varying, monotonically increasing function that corrects the optical depth by $\approx 15\%$ at most. Its values were obtained using the bicubic spline interpolation technique (Press et al. 2007) on a table provided by C. Churchill.

Since the temperature of each layer only depends on the optical depth, it is preset for each layer of the model atmosphere. Using hydrostatic equilibrium, we can solve for the gas pressure increment from layer to layer,

$$dP_{tot} = g \frac{d\tau}{\chi(\rho_{tot}, T)}, \tag{3}$$

where ρ_{tot} is the gas density of the layer and $\chi(\rho_{tot}, T)$ is the Rossland Mean Opacity, defined as

$$\chi_R^{-1} = \frac{\int_0^\infty \chi_\lambda^{-1} \frac{dK_\lambda}{dz} d\lambda}{\int_0^\infty \frac{dK_\lambda}{dz} d\lambda}, \tag{4}$$

where χ_λ is the opacity of the gas and dK_λ/dz is the gradient of the the second moment of the intensity with depth into the atmosphere. For our model, we were able to obtain customized opacity tables for specific atmospheric elemental abundances using the OPAL code (Iglesias & Rogers 1996). Bicubic 2D spline interpolation (Press et al. 2007) was used to obtain continuous opacities. For each layer, the opacity depends on temperature and gas density. The latter can only be obtained from the ideal gas law

$$N_{tot} = \frac{P_{tot}}{kT}, \tag{5}$$

[1] New Mexico State University, Las Cruces, NM 88003 st@astronomy.nmsu.edu

where N_{tot} is the particle number density in a given layer. The total particle number density depends critically on the ionization ratios of each species in the gas (given by the Saha equation),

$$\frac{N_{j+1,k}}{N_{jk}} = \frac{C_\Phi}{n_e}\frac{U_{j+1}(T)}{U_j(T)}T^{3/2}\exp\left(-\frac{\chi_{Ijk}}{kT}\right), \quad (6)$$

where n_e is the electron number density and U_{jk} are the partition functions (from bicubic spline interpolation on tables found in Gray (2005)). Using the principles of particle and charge conservation, we find a transcendental equation for the electron density,

$$n_e = (N_{tot} - n_e)\sum_{k}^{allspecies}\alpha_k\sum_{j=1}^{J_k} jf_{jk}(n_e, T). \quad (7)$$

Here, f_{jk} are the ionization fractions of species k in stage j as obtained from recursive use of the Saha equation. We used the ZBRENT root-solving routine described in Press et al. (2007) to calculate the electron density of each layer.

Without a priori knowledge of the electron density in a given layer, one can only guess what the opacity will be in the next one. This means the pressure differential has to be solved for iteratively by first "guessing" it using the parameters of the previous layer and then comparing this guess with a more accurate estimate obtained by averaging the opacity (as it appears in the ideal gas law) over the previous and current steps. This pressure is then checked to converge to an accuracy of $1 - P_{avg}/P_{guess} \approx 10^{-5}$. Once the macroscopic properties of the gas (i.e. temperature, pressure, total and electron density) are obtained for each step in τ, microscopic properties such as ionization and excitation ratios, electron contributions and flux are easily calculated as a function of depth in the atmosphere.

In summary, our stellar atmosphere model simplifies the physical behavior of the system using several approximations:

1. Plane parallel atmosphere: thickness of the atmosphere is small compared to the star's radius
2. Eddington + Grey Case: one isotropic intensity per hemisphere and all parameters independent of wavelength
3. LTE: radiation near the top of the atmosphere assumed to participate in energy balance without escaping
4. Hydrostatic equilibrium: assumes absence of bulk fluid motions, convection and radiation pressure
5. Metals in the atmosphere: assume correct metallicity but use single metal (opacity table accounts for this)

The full version of the computer code is reproduced in Appendix A.

3. CONVERGENCE

As a way to assure the validity of the results obtained using the computer model, we tested the code in two ways.

First, we adjusted the initial pressure guess for the top layer of Model 1 (see below) starting at a value $P_{tot} = 10^{-3}$dyne cm^{-2} and increasing it in small steps all the way up to $P_{tot} = 10^3$dyne cm^{-2}. We found that the model is quite insensitive to the initial choice of pressure, quickly converging to the same value (to 3 significant digits) within the first few steps in optical depth. Even though the convergence is extremely fast, we decided to start the models approximately five steps prior to $\tau = 10^{-3}$, and simply discarded this transitional period in out analysis.

Second, since the algorithm relies on calculations of parameters for each advancing layer from finite differencing, we investigated the sensitivity of the code to the thickness of each layer, in other words, the number of τ steps required for convergence of the physics. By varying the number of logarithmic steps from 100 to 10000 we saw a maximum deviation of 0.1% in all the macroscopic variables all the way to $\tau = 10^3$. Thus, we concluded that the iterative pressure loop may have lifted the need for high resolution in optical depth (and equivalently in physical depth). Since the running time was extremely short for 1000 layers, we settled on this number.

TABLE 1
PARAMETERS OF MODEL 1 AND MODEL 2

	Model 1	Model 2
Spectral Type	G	O
Mass [$M_\odot$]	1	120
Radius [$R_\odot$]	1	30
T_{eff} [K]	5800	51000
log(g)	4.44	3.56
Composition	H,He,C [a]	H,He
Mass Fractions, X,Y,Z	0.70,0.28,0.02	0.70,0.30

[a] In the case, of carbon, we assume only the first two possible ionizations

4. RESULTS AND DISCUSSION

We run our computer model for two sets of parameters corresponding to two very different stellar populations, a solar type main sequence star and a O type very low metallicity blue giant candidate for a Pop III star. The physical input parameters of each model are summarized in Table 1 [2].

In the next sections we discuss the results of the two models

4.1. *Model 1*

In Figure 1*a*, we show the physical depth into the atmosphere (where $z = 0$ is the top) as a function of optical depth.

At first glance, we confirm one of the assumptions in our physical model: that the thickness of the atmosphere is much smaller that the radius of the star with $z(\tau = 10^3) \approx 1300$km $\approx 0.002 R_\odot$. The atmosphere of Model 1 is indeed plane-parallel. Naively, we would expect a smooth proportionality between the photon mean free path (the inverse of the optical depth) and the height through the atmosphere. Instead, the physical depth changes abruptly within the top dilute layers and it transitions to a plateau for $1 < \tau < 10^2$, spiking up again for $\tau > 10^2$. This behavior is better understood by examining the physics behind the optical depth. By definition,

$$d\tau = \chi(\rho_{tot}, T)\rho_{tot}dz \quad (8)$$

in a Grey atmosphere. This relationship tells us that the incremental change in physical depth across layers will be $dz = d\tau/\chi\rho_{tot}$, so as the density of the gas increases with depth (as expected), even for a constant opacity term, the incremental change in the physical depth will continuously decrease and produce a plateau in the physical depth which is seen up to $\tau \approx 1$ in Figure 1*a* . The subsequent changes in the slope are better understood in the context of the density and optical depth vs temperature relations.

[2] atomic data from http://www.nist.gov/

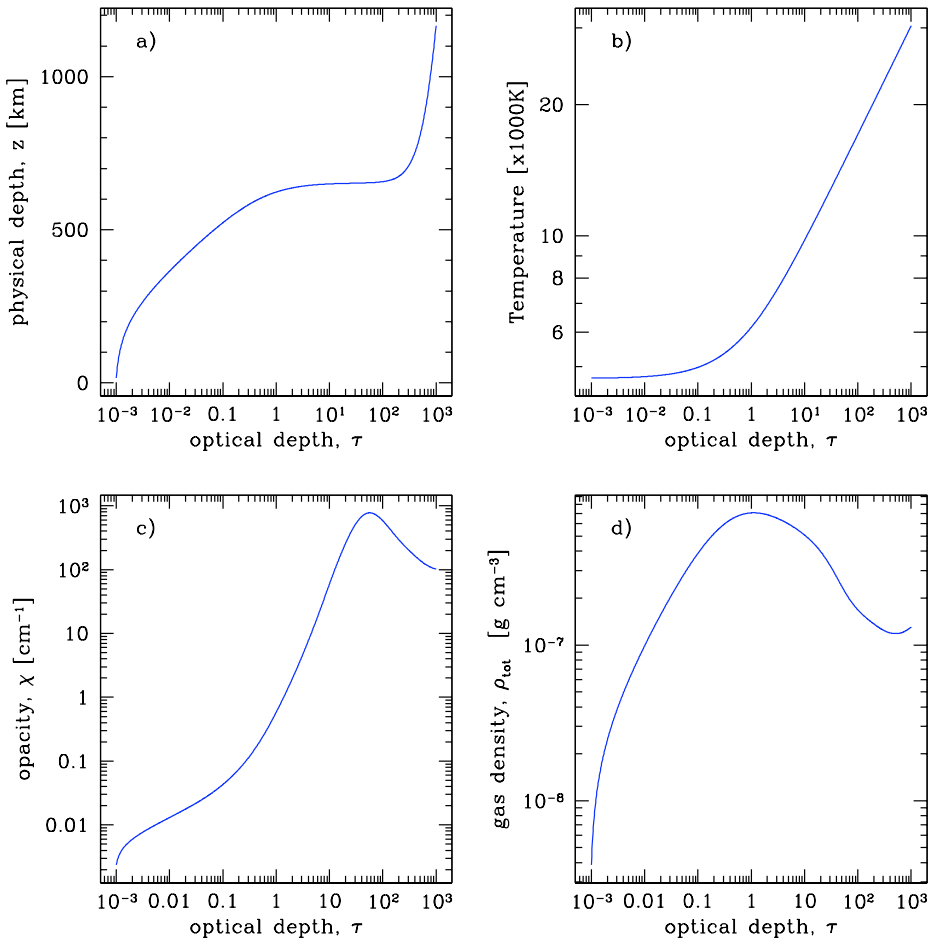

FIG. 1.— (a) The physical depth into the atmosphere of Model 1 vs. optical depth. The stellar radius is $R = 6.96 \times 10^{\vec{9}}$km. — (b) The temperature of the gas as a function of optical depth into the atmosphere of Model 1. Notice $T_{eff} = 5800$K occurs at $\tau \approx 2/3$. — (c) The Rossland Mean Opacity, χ, as a function of optical depth into the atmosphere of Model 1. — (d) The total mass density, ρ_{ot}, of the gas as a function of optical depth into the atmosphere of Model 1.

In Figure 1*b*, we plot the optical depth dependence of the temperature. As discussed in § 2, the evaluation of the moments of the intensity in a Grey atmosphere in LTE yields a simple power law relation, Eq.2 with a correction term in the form of the Hopf function. This power law behavior is displayed in Figure 1*b*, and being independent of all the other physical parameters in the model, it simply tells us that the uppermost layer of the atmosphere at $\tau = 10^{-3}$ is at $T \approx 4700$K, with the temperature steeply rising up to $T = 30000$K at $\tau = 10^3$. Also, the effective photospheric temperature matches that of the Sun at $T(\tau = 2/3) \approx 5700$K.

Figure 1*c* shows the opacity of the gas in the atmosphere of Model 1. A clue to the origin of the turnover in the opacity at $\tau \approx 50$ is found in the temperature plot in Figure 1*b*. As we travel deeper into the atmosphere of the star, the temperature increases resulting in higher ionization fractions for all the species. At about 10000K or $\tau \approx 10$, Hydrogen (which dominates in number over all other species) becomes mostly ionized and it can no longer absorb radiation through bound-bound transitions, resulting in a sharp decrease in the total absorption cross section of the gas. This produces the observed turnover in the opacity.

Figure 1*d* examines the total gas density as a function of optical depth. Once again, we observe a turn over in the density when the optical depth becomes considerable, at $\tau \approx 1$. At first glance this drop off might seem unphysical compared to what is observed in planetary atmospheres (i.e. the monotonic density increase with decreasing height from the surface), but a closer look at the ideal gas law, Eq.5 shows the particle number density is the ratio of the total pressure to the temperature of the gas. Clearly, for a slowly increasing pressure and a steeply increasing temperature there will be a depth at which the temperature effect dominates and lowers the density (see Figure 2 and discussion below). All the physical parameters discussed above depend critically on the temperature gradient in the Grey case of Eddington's approximation. This approximation relies on the homogeneity of each gas layer and does not include the effects of turbulent or bulk motions of the gas, convection and radiation pressure support. These are all expected to be small contributions for a solar type star, but they might produce serious deviations in the atmosphere of Model 2. It is now evident why there is a sharp increase in the physical depth in Figure 1*a*. Both the opacity, χ, and the density ρ_{tot}, in Eq.8 turnover and begin decreasing at high optical depth. This gives rise to the observed spike in physical depth, z. We are careful to note that the exact shape of the the curves discussed above might differ slightly for more realistic compositions than the single metal used in Model 1. As the

opacity tables were modified to include all the metal in the solar atmosphere, the Rossland Mean opacity changed slightly producing a departure of about 10 – 20% from Model 1.

In Figure 2, we investigate the evolution of the gas and electron pressures in Model 1.

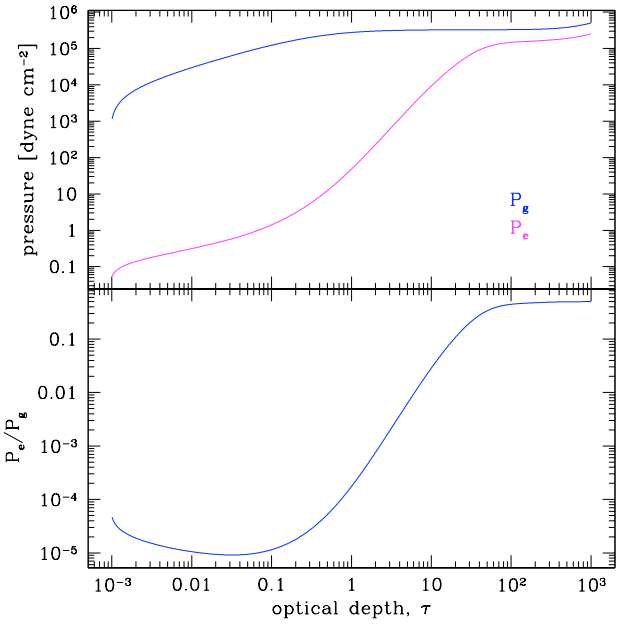

FIG. 2.— *Top panel*: The behavior of the total gas pressure, P_g, and the electron pressure, P_e, as functions of optical depth into the atmosphere of Model 1. *Lower panel*: the ratio of the pressures approaches 0.5 as hydrogen becomes fully ionized at high temperatures deep inside the atmosphere.

As outlined in § 2, the electron pressure is obtained in each layer using charge conservation and ionization equilibrium, which are both functions of the temperature and the total pressure in the gas (Eq.7). The total pressure varies by almost 3 orders of magnitude from the top of the atmosphere to $\tau = 10^3$ and increases steadily up to $\tau \approx 1$, where it remains at a near constant value due to the fast rise in the opacity in this regime (see Eq.3). As the opacity peaks near $\tau = 50$ and then starts to decrease, hydrogen becomes ionized in large fractions making the contribution to the total pressure from electrons approach %50 (due to the number density dominance of hydrogen over other species, close to half of all the particles are electrons and half are protons). This is even more evident in the negligible contribution of the electron pressure at the top of the atmosphere (where hydrogen is neutral) and its fast rise after $\tau \approx 10$, which corresponds to $T \approx 10000$K.

Moving on to the microscopic regime, Figure 3 shows the ionization fractions, N_{jk}/N_k of each species k at each ionization stage j, as a function of optical depth as obtained from recursive use of Eq.6.

As expected, as we dive deeper into the atmosphere and the temperature increases, the neutral hydrogen (H I) fraction begins to drop, accompanied by a rise in H II at about $\tau = 10$ or $T = 10000$K. More interesting behaviors are observed for Heand C. Since neutral helium has a higher ionization potential than hydrogen ($\chi_{I,0,2} = 24.59$eV), it displays the same trend with an increase in the abundance of He Iand a corresponding rise in He II at $\tau \approx 10^2$. Doubly ionized Helium is even harder to obtain, with its relative abundance $\approx 10^{-70}$ up to $\tau = 1$ and slowly rising to $\approx 10^{-5}$ at the bottom of the at-

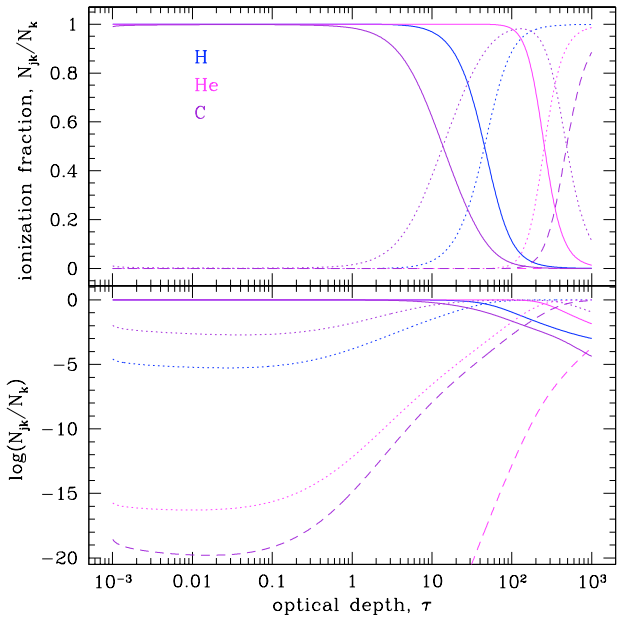

FIG. 3.— *Top panel*: The ionization fractions N_{jk}/N_k for each ionization stage j of each species k (H,He,C) in the atmosphere of Model 1. Solid lines represent neutral atoms, dotted lines represent singly ionized atoms, and dashed lines represent doubly ionized atoms. *Lower panel*: same plot in logarithmic space covering several orders of magnitude.

mosphere. Neutral carbon is slightly more prone to ionization than hydrogen ($\chi_{I,0,3} = 11.26$eV), so we observe C I falling at around $\tau = 1$ and C II taking over until the temperature becomes too high for the two lowest ionization stages. At about $\tau = 10^2$ or $T = 17000$K C III, takes over, dominating the carbon contribution for $\tau > 10^3$. As was observed in the plots of the pressures, the high abundance of hydrogen over all other species makes its microscopic properties (excitation and ionization) dominate the macroscopic gas physics and the opacity of the gas. This behavior is quantitatively presented in Figure 5, which plots the electron densities contributed by each ionization stage of each species. These were obtained by simply multiplying the abundace of each ion by the number of electrons lost and the ionization fractions from Eq.6,

$$n_{e,jk} = (N_{tot} - n_e)\alpha_k j f_{jk}(n_e, T). \quad (9)$$

As we expected, singly ionized hydrogen dominates at all temperatures or optical depths due to its high abundance in the gas. Carbon, being almost as easily ionized as hydrogen, contributes almost the same electron density up to a depth $\tau \approx 1$, where the rise in the ionization fraction of hydrogen takes over as the dominant effect. At $\tau \approx 100$, helium starts to become singly ionized and its electrons contribute about 10% of the total. Electrons from doubly ionized species, He II and C II make up at most 1% of the total pool in the atmosphere at the deepest layer.

Lastly, we examine the line strengths of the hydrogen Lyman, Balmer and Paschen absorption features indirectly by calculating the excitation ratios of each series as a function of optical depth. The Boltzmann equation gives

$$\frac{n_{ijk}}{N_{jk}} = \frac{g_{ijk}}{U_j(T)} \exp\left(-\frac{\chi_{ijk}}{kT}\right), \quad (10)$$

where n_{ijk}/N_{jk} is the fraction of atoms of species k in ionization stage j that are excited to energy level i, g_{ijk} are the statistical weights, and χ_{ijk} is the excitation energy of level i.

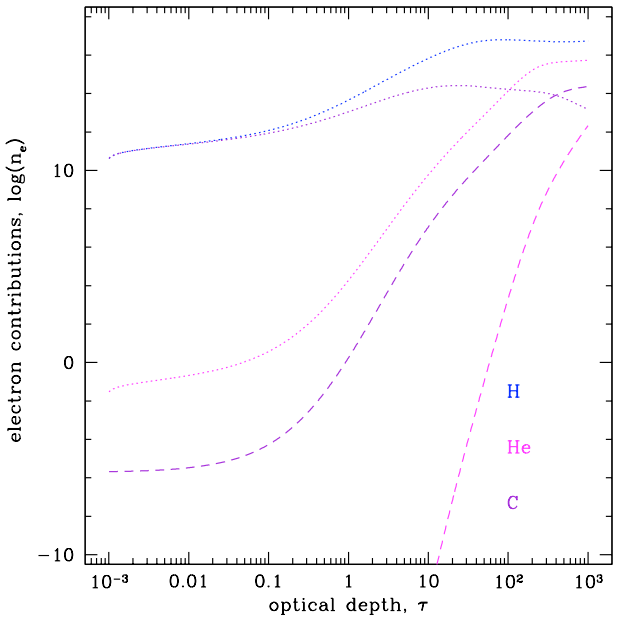

FIG. 4.— The partial contributions of each ion each species to the total electron density in the atmosphere of Model 1. As before, solid lines represent neutral atoms, dotted lines represent singly ionized atoms, and dashed lines represent doubly ionized atoms.

In Figure **??**, we show the results for the first three series of hydrogen.

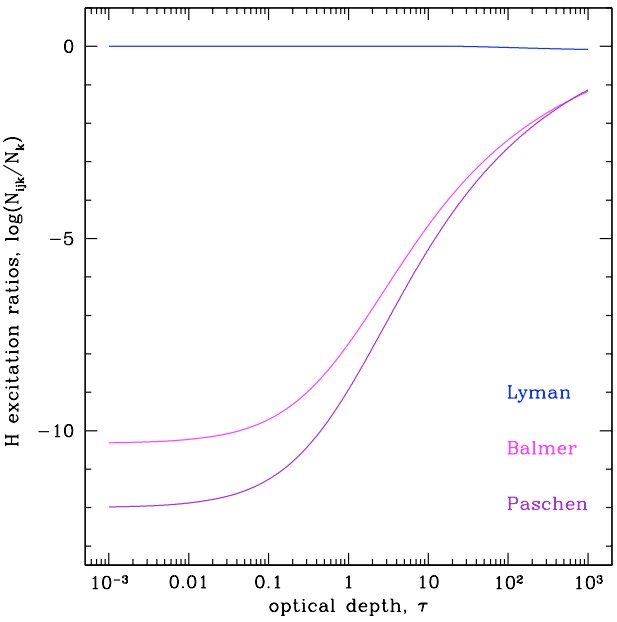

FIG. 5.— The relative excitation fractions of the neutral hydrogen Lyman, Balmer and Paschen series as a function of optical depth into the atmosphere of Model 1.

For the shallow layers, the temperature of the gas is well below 10000K , which is unsufficient to give most ground level electrons the 10.2eV they need to jump to the first excited state. In this range most of the electrons will be able to produce Lyman absorption lines and the Balmer and Paschen transitions will be about 10 orders of magnitude less likely to occur. As is the case with hydrogen ionization, and since the energies are close, the transition to higher excitations begins at $T \approx 10000$K or $\tau = 100$, with the Balmer and Paschen fractions increasing as we go deeper into the atmosphere (the Balmer fraction leading due to its more accesible excitation potential). By the time we reach the deepest layer of the atmosphere, the Lyman series starts to weaken slightly and the Balmer and Paschen electrons comprise about 10% of the neutral hydrogen electrons. The more ambitious task of computing the actual absorption line profiles was attempted but was not finished due to time constraints.

4.2. *Model 2*

This section could not to be finished due to time constraints.

5. CONCLUSIONS

Overall, Model 1 was succesful in reproducing the expected behavior of a stellar atmosphere like that of the Sun. The computer code showed a rapid convergence with the number of layers and predicted an atmospheric thickness of about 1% of the radius of the star. This confirmed our assumption of a plane parallel atmosphere with a constant gravitational acceleration. The temperature dependence on the optical depth was not modeled but obtained analitically from Eq.2 and as such it is an approximation based on local thermodynamic equilibrium. Turbulence or convection in the atmosphere are not accounted for and may introduce small deviations from the ideal model. As mentioned before the model is based on a motionless fluid in hydrostatic equilibrium and does not account for radiation pressure support or inhomogeneities in the gas layers. These effects might result in serious departures for models of high temperature stars. The next level of realism in the model would include the full composition of the star (in Model 1, the two dozen or so metals present in the Sun) and all the possible ionizations of the metals. Again, as mentioned before, these simplifications might have introduced an error of about 20% compared to the more elaborate models . Despite its simplistic nature, the model seemed to reproduce the features expected both in the macroscopic parameters of the gas and in the microscopic particle balance.

The authors wish to thank Prof. C. W. Churchill for his important contributions to the undestanding of the physical concepts involved in this work and for providing important guidance in the use of numerical algorithms for the computer code.

REFERENCES

Iglesias C.A. & Rogers F.J. 1996, ApJ, 464, 943
Press, W.H., Teukolsky, S.A., Vetterling, W. T., Flannery, B. P.(2007), Numerical Recipes
Gray, D.F. (2005), The Observation and Analysis of Stellar Photospheres

APPENDIX
COMPUTER CODE

```
MODULE routines

REAL*8, PARAMETER                     :: k_eV=8.62e-5, k_erg=1.38e-16, G_const=6.67e-8, m_H=1.66e-24, m_e=9.11e-28
INTEGER, PARAMETER                    :: jmax(3)=(/1,2,2/)
REAL*8                                :: alpha(3), T, tau, N_tot, n_e, N_n, rho_n, rho_e, rho_tot, opacity
INTEGER, PARAMETER                    :: Npoints=10
REAL*8, DIMENSION(0:2,1:3,1:Npoints)  :: U_table=0

INTEGER, parameter                    :: Npnts=19, Mpnts=57
REAL*8, DIMENSION(1:Npnts,1:Mpnts)    :: op_tab=0
real*8, dimension(1:Npnts)            :: rho_table=0
real*8,dimension (1:Mpnts)            :: T_table=0



CONTAINS
!-------------------------------------------------------------------------------------------
      SUBROUTINE SPLINE(X,Y,N,YP1,YPN,Y2)
!     computes 1D 2nd derivatives of grid
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (NMAX=600)
      DIMENSION X(N),Y(N),Y2(N),U(NMAX)
      IF (YP1.GT..99D30) THEN
        Y2(1)=0.
        U(1)=0.
      ELSE
        Y2(1)=-0.5
        U(1)=(3./(X(2)-X(1)))*((Y(2)-Y(1))/(X(2)-X(1))-YP1)
      ENDIF
      DO 11 I=2,N-1
        SIG=(X(I)-X(I-1))/(X(I+1)-X(I-1))
        P=SIG*Y2(I-1)+2.
        Y2(I)=(SIG-1.)/P
        U(I)=(6.*((Y(I+1)-Y(I))/(X(I+1)-X(I))-(Y(I)-Y(I-1))/(X(I)-X(I-1)))/(X(I+1)-X(I-1))-SIG*U(I-1))/P
11    CONTINUE
      IF (YPN.GT..99D30) THEN
        QN=0.
        UN=0.
      ELSE
        QN=0.5
        UN=(3./(X(N)-X(N-1)))*(YPN-(Y(N)-Y(N-1))/(X(N)-X(N-1)))
      ENDIF
      Y2(N)=(UN-QN*U(N-1))/(QN*Y2(N-1)+1.)
      DO 12 K=N-1,1,-1
        Y2(K)=Y2(K)*Y2(K+1)+U(K)
12    CONTINUE
      RETURN
      END SUBROUTINE


!-------------------------------------------------------------------------------------------
      SUBROUTINE SPLINT(XA,YA,Y2A,N,X,Y)
!     performs 1D interpolation
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION XA(N),YA(N),Y2A(N)
      KLO=1
      KHI=N
1     IF (KHI-KLO.GT.1) THEN
        K=(KHI+KLO)/2
        IF(XA(K).GT.X)THEN
          KHI=K
        ELSE
          KLO=K
        ENDIF
      GOTO 1
      ENDIF
      H=XA(KHI)-XA(KLO)
      IF (H.EQ.0.) PAUSE 'Bad XA input.'
      A=(XA(KHI)-X)/H
      B=(X-XA(KLO))/H
      Y=A*YA(KLO)+B*YA(KHI)+((A**3-A)*Y2A(KLO)+(B**3-B)*Y2A(KHI))*(H**2)/6.
      RETURN
      END SUBROUTINE

!-------------------------------------------------------------------------------------------
      SUBROUTINE SPLIE2(X1A,X2A,YA,M,N,Y2A)
!     computes 2D 2nd derivatives of grid
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (NN=600)
```

```
      DIMENSION X1A(M),X2A(N),YA(M,N),Y2A(M,N),YTMP(NN),Y2TMP(NN)
      DO 13 J=1,M
        DO 11 K=1,N
          YTMP(K)=YA(J,K)
11      CONTINUE
        CALL SPLINE(X2A,YTMP,N,1.D30,1.D30,Y2TMP)
        DO 12 K=1,N
          Y2A(J,K)=Y2TMP(K)
12      CONTINUE
13    CONTINUE
      RETURN
      END SUBROUTINE

!-------------------------------------------------------------------------------------------
      SUBROUTINE SPLIN2(X1A,X2A,YA,Y2A,M,N,X1,X2,Y)
!     performs 2D interpolation
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (NN=600)
      DIMENSION X1A(M),X2A(N),YA(M,N),Y2A(M,N),YTMP(NN),Y2TMP(NN),YYTMP(NN)
      DO 12 J=1,M
        DO 11 K=1,N
          YTMP(K)=YA(J,K)
          Y2TMP(K)=Y2A(J,K)
11      CONTINUE
        CALL SPLINT(X2A,YTMP,Y2TMP,N,X2,YYTMP(J))
12    CONTINUE
      CALL SPLINE(X1A,YYTMP,M,1.D30,1.D30,Y2TMP)
      CALL SPLINT(X1A,YYTMP,Y2TMP,M,X1,Y)
      RETURN
      END SUBROUTINE

!-------------------------------------------------------------------------------------------
DOUBLE PRECISION FUNCTION U(j,k,Theta)

implicit none

INTEGER, INTENT(IN)                :: j, k
REAL*8, INTENT(IN)                 :: Theta
INTEGER                            :: i,io
REAL*8                             :: slope1=2e30, slope2=2e30, Uint
REAL*8, DIMENSION(1:Npoints)       :: deriv, U_tab=0, Theta_table=(/0.2,0.4,0.6,0.8,1.0,1.2,1.4,1.6,1.8,2.0/)


DO i=1,Npoints
U_tab(i) = U_table(j,k,i)
ENDDO

 CALL SPLINE(Theta_table,U_tab,Npoints,slope1,slope2,deriv)
 CALL SPLINT(Theta_table,U_tab,deriv,Npoints,Theta,Uint)

U = 10**Uint
IF (U < 0) write(*,*) 'bad interpolation'

!write(*,'(a,f10.3)') 'U = ', U

END FUNCTION U

!-------------------------------------------------------------------------------------------
DOUBLE PRECISION FUNCTION hopf(tau)

implicit none

REAL*8, INTENT(IN)                 :: tau
INTEGER                            :: i,io
integer, parameter                 :: Npnts=20
REAL*8                             :: slope1=2e30, slope2=2e30, q
REAL*8, DIMENSION(1:Npnts)         :: deriv, tau_tab, q_tab


tau_tab = (/0.00,0.01,0.03,0.05,0.10,0.20,0.30,0.40,0.50,0.60,0.8,1.0,1.5,2.0,2.5,3.0,3.5,4.0,5.0,1e30/)
q_tab = (/0.577351,0.588236,0.601242,0.610758,0.627919,0.649550,0.663365,0.673090,0.680240,0.685801, &
          0.693534,0.698540,0.705130,0.707916,0.709191,0.709806,0.710120,0.710270,0.710398,0.710446/)

 CALL SPLINE(tau_tab,q_tab,Npnts,slope1,slope2,deriv)
 CALL SPLINT(tau_tab,q_tab,deriv,Npnts,tau,q)

write(*,'(a,es12.3)') '   q =', q
hopf = q

END FUNCTION hopf

!-------------------------------------------------------------------------------------------
DOUBLE PRECISION FUNCTION op(rho,T)
```

```
implicit none

REAL*8, INTENT(IN)                 :: rho, T
INTEGER                            :: i,io
REAL*8                             :: op_int, logT, logrho
REAL*8, DIMENSION(1:Npnts,1:Mpnts)  :: deriv


logT = log10(T)
logrho = log10(rho/((1e-6*T)**3))

 CALL SPLIE2(rho_table,T_table,op_tab,Npnts,Mpnts,deriv)
 CALL SPLIN2(rho_table,T_table,op_tab,deriv,Npnts,Mpnts,logrho,logT,op_int)

op = 10**op_int
IF (op < 0) write(*,*) 'bad opacity interpolation'


write(*,'(a,es10.3,a,f7.3,a,f7.3,a,f7.3)') &
'   opacity =', op, '    log =', log10(op),'    log(T) =',logT,'    log(R) =',logrho

END FUNCTION op

!-----------------------------------------------------------------------------------------
DOUBLE PRECISION FUNCTION Boltz(i,j,k,T)

implicit none

INTEGER, INTENT(IN)               :: i, j, k
REAL*8, INTENT(IN)                :: T
REAL*8                            :: Theta
REAL*8, DIMENSION(0:2,0:1,1:3)    :: chi_exc, g

chi_exc(0,0,1) = 0.000
chi_exc(1,0,1) = 10.198
chi_exc(2,0,1) = 12.088
g(0,0,1) = 2.0
g(1,0,1) = 8.0
g(2,0,1) = 18.0

Theta = 5040.0/T
Boltz = (g(i,j,k)/U(j,k,Theta))*10**(-Theta*chi_exc(i,j,k))


END FUNCTION Boltz

!-----------------------------------------------------------------------------------------
DOUBLE PRECISION FUNCTION SAHA(j,k,n_e,T)

implicit none

INTEGER, INTENT(IN)               :: j, k
REAL*8, INTENT(IN)                :: n_e, T
REAL*8, PARAMETER                 :: C_phi=4.83e15
REAL*8                            :: Theta
REAL*8, DIMENSION(0:1,1:3)        :: chi

chi(0,1) = 13.6
chi(0,2) = 24.59
chi(0,3) = 11.26
chi(1,1) = 0.0
chi(1,2) = 54.42
chi(1,3) = 24.38

Theta = 5040.0/T
IF (j <= jmax(k)) THEN
SAHA = C_phi*n_e**(-1)*T**(3./2)*(U(j,k,Theta)/U(j-1,k,Theta))*10**(-Theta*chi(j-1,k))
ELSE
SAHA = 0
ENDIF

END FUNCTION SAHA

!-----------------------------------------------------------------------------------------
DOUBLE PRECISION FUNCTION recsaha(j,k,n_e,T)

implicit none

INTEGER, INTENT(IN)               :: j, k
REAL*8, INTENT(IN)                :: n_e, T
REAL*8                            :: P, S
INTEGER                           :: l, jj

S =1
```

```
DO jj=1,jmax(k)
P =1
DO l=1,jj
P = P*SAHA(l,k,n_e,T)
ENDDO
S = S + P
ENDDO

P =1
IF (j == 0) THEN
P =1
ELSE
DO l=1,j
P = P*SAHA(l,k,n_e,T)
ENDDO
ENDIF

recsaha = P/S


END FUNCTION recsaha

!-----------------------------------------------------------------------------------------------
DOUBLE PRECISION FUNCTION myfunc(n_e)

implicit none

INTEGER                          :: j, k
REAL*8, INTENT(IN)               :: n_e
REAL*8                           :: tot, tot2

tot2 =0
DO k=1,3
tot =0
DO j=1,jmax(k)
tot = tot + j*recsaha(j,k,n_e,T)
ENDDO
  tot2 = tot2 + alpha(k)*tot
ENDDO
myfunc = (N_tot - n_e)*tot2 - n_e

END FUNCTION myfunc

!-----------------------------------------------------------------------------------------------
      DOUBLE PRECISION FUNCTION ZBRENT(FUNC,X1,X2,TOL)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (ITMAX=100,EPS=3.E-16)
      A=X1
      B=X2
      FA=FUNC(A)
      FB=FUNC(B)
      IF(FB*FA.GT.0.) PAUSE 'Root must be bracketed for ZBRENT.'
      FC=FB
      DO 11 ITER=1,ITMAX
        IF(FB*FC.GT.0.) THEN
          C=A
          FC=FA
          D=B-A
          E=D
        ENDIF
        IF(ABS(FC).LT.ABS(FB)) THEN
          A=B
          B=C
          C=A
          FA=FB
          FB=FC
          FC=FA
        ENDIF
!        call dummy_call( eps )
        TOL1=2.*EPS*ABS(B)+0.5*TOL
        XM=.5*(C-B)
        IF(ABS(XM).LE.TOL1 .OR. FB.EQ.0.)THEN
          ZBRENT=B
          RETURN
        ENDIF
        IF(ABS(E).GE.TOL1 .AND. ABS(FA).GT.ABS(FB)) THEN
          S=FB/FA
          IF(A.EQ.C) THEN
            P=2.*XM*S
            Q=1.-S
          ELSE
            Q=FA/FC
            R=FB/FC
            P=S*(2.*XM*Q*(Q-R)-(B-A)*(R-1.))
```

```
            Q=(Q-1.)*(R-1.)*(S-1.)
          ENDIF
          IF(P.GT.0.) Q=-Q
          P=ABS(P)
          IF(2.*P .LT. MIN(3.*XM*Q-ABS(TOL1*Q),ABS(E*Q))) THEN
            E=D
            D=P/Q
          ELSE
            D=XM
            E=D
          ENDIF
        ELSE
          D=XM
          E=D
        ENDIF
        A=B
        FA=FB
        IF(ABS(D) .GT. TOL1) THEN
          B=B+D
        ELSE
          B=B+SIGN(TOL1,XM)
        ENDIF
        FB=FUNC(B)
11    CONTINUE
      PAUSE 'ZBRENT exceeding maximum iterations.'
      ZBRENT=B
      RETURN
      END FUNCTION

!-----------------------------------------------------------------------------------------------
END MODULE routines

!_______________________________________________________________________________________________

PROGRAM main

USE routines

IMPLICIT NONE

REAL*8                              :: dtau, delta_tau, delta_z, g, Ntot(3)=0
REAL*8, PARAMETER                   :: Am(3)=(/1.01,4.00,12.01/), eps=1e-12, &
                                       minv=1e10, maxv=1e18, tau_min=0.99e-3, tau_max=1e3, &
                                       Xm=0.70, Ym=0.28, Zm=0.02, T_eff=5800, M_star=1.99e33, R_star=6.96e10
INTEGER                             :: ii, jj, kk, i, j, k
REAL*8, DIMENSION(0:2,1:3)          :: N=0, n_e_partial=0
character(7)                        :: junk
integer, parameter                  :: Nsteps=1000
real*8                              :: P_1, P_2, P_e, N_boltz00, N_boltz10, N_boltz20
real*8, dimension(1:Nsteps)         :: tau_a, T_a, P_tot_a, opacity_a, rho_tot_a, z !-----arrays for storing
   !parameters of each layer



!---------read partition functions table
OPEN(UNIT=5,FILE='partitionfunctions.dat')
READ(5,*)
READ(5,*)
DO ii=1,Npoints
READ(5,*) ((U_table(jj,kk,ii), jj=0,2), kk=1,3)
ENDDO
 CLOSE(5)
write(*,*) 'partition funcs read ok'

DO i=1,Npoints
WRITE(*,'(9(f12.3,1x))') ((U_table(jj,kk,i), jj=0,2), kk=1,3)
ENDDO

!---------read opacity table
OPEN(UNIT=5,FILE='opacitytable.dat')
READ(5,*)
READ(5,*)
READ(5,*)
READ(5,*)
READ(5,*) junk, (rho_table(ii), ii=1,Npnts)
READ(5,*)
DO jj=1,Mpnts
READ(5,*) T_table(jj), (op_tab(ii,jj), ii=1,Npnts)   !----reads row by row where
                                                     !ii gives density and jj gives temp.
ENDDO
 CLOSE(5)

write(*,*) 'opacities read ok'
write(*,'(a,1x,19(f4.1,4x))') junk, (rho_table(ii), ii=1,Npnts)
```

```
DO jj=1,5
write(*,'(f5.2,1x,19(f7.3,1x))') T_table(jj), (op_tab(ii,jj), ii=1,Npnts)
ENDDO
write(*,*)

!-----calculate alphas:
alpha(1) = 1/(1 + (Am(1)/Am(2))*(Ym/Xm) + (Am(1)/Am(3))*(Zm/Xm))
alpha(2) = alpha(1)*(Am(1)/Am(2))*(Ym/Xm)
alpha(3) = alpha(1)*(Am(1)/Am(3))*(Zm/Xm)

g = G_const*M_star/(R_star**2)   !---calculate surface gravity
write(*,'(a,es10.2)') 'g =', g
write(*,*)

!---open output files:
OPEN(6,FILE='results/model.output1.dat')
OPEN(7,FILE='results/model.output2.dat')
OPEN(8,FILE='results/model.output3.dat')
write(6,'(20(a,4x))')
' tau     ','   z        ','Temp     ','opacity  ','rho_tot  ','P_tot    ','P_e      ','P_e/P_tot'
write(7,'(20(a,4x))')
' tau     ','   Boltz00  ','Boltz10  ','Boltz20  '
write(8,'(20(a,2x))')
' tau     ','     N_01/N_1 ','N_11/N_1 ','N_02/N_2 ','N_12/N_2 ','N_22/N_2 ','N_03/N_3 ', &
'N_13/N_3 ','N_23/N_3 ', &
'n_e_11   ','n_e_12   ','n_e_22   ','n_e_13   ','n_e_23   '


tau_a=0; T_a=0; P_tot_a=0; opacity_a=0; rho_tot_a=0; z=0   !---initialize arrays

!-------------------------------------------------------------------------start tau loop
T_a=0; P_tot_a=0; opacity_a=0; rho_tot_a=0

dtau = log10(tau_max/tau_min)/(real(Nsteps)-1)

P_tot_a(1) = 9.895e2
z(1) =0

ii =1   !---first layer:

write(*,'(a,i5)') 'ii =', ii
tau = tau_min*10**((ii-1)*dtau)
tau_a(ii) = tau
write(*,'(a,es12.3)') '   tau =', tau_a(ii)
T = T_eff*((3./4)*(tau + hopf(tau)))**(1./4)
T_a(ii) = T
write(*,'(a,f8.2)') '   T =', T_a(ii)
write(*,'(a,es12.3)') '   P_tot =', P_tot_a(ii)
N_tot = P_tot_a(ii)/(k_erg*T)
write(*,'(a,es12.3)') '   N_tot =', N_tot
n_e = ZBRENT(myfunc,minv,maxv,eps)
write(*,'(a,es12.3)') '   n_e =', n_e
N_n = N_tot - n_e
write(*,'(a,es12.3)') '   N_n =', N_n
rho_n =0
do k=1,3
        rho_n = rho_n + alpha(k)*Am(k)
enddo
rho_n = m_H*N_n*rho_n
write(*,'(a,es12.3)') '   rho_n =', rho_n
rho_e = m_e*n_e
write(*,'(a,es12.3)') '   rho_e =', rho_e
rho_tot = rho_n + rho_e
rho_tot_a(ii) = rho_tot
write(*,'(a,es12.3)') '   rho_tot =', rho_tot_a(ii)
opacity = op(rho_tot,T)
opacity_a(ii) = opacity
write(*,'(a,es10.3)') '   z =', z(ii)

ii =2   !---loop from second layer to last layer:
DO
        write(*,'(a,i5)') 'ii =', ii
        tau = tau_min*10**((ii-1)*dtau)
        tau_a(ii) = tau
        write(*,'(a,es12.3)') '   tau =', tau_a(ii)
        T = T_eff*((3./4)*(tau + hopf(tau)))**(1./4)
        T_a(ii) = T
        write(*,'(a,f8.2)') '   T =', T_a(ii)
        delta_tau = tau_a(ii) - tau_a(ii-1)
        write(*,'(a,es12.3)') '   delta_tau =', delta_tau
        P_1 = P_tot_a(ii-1) + g*delta_tau/opacity_a(ii-1)   !----pressure from previous opacity
        !---------------------------------
        do      !----------------------------------------iterative pressure loop
                write(*,'(a,es12.3)') '   P_1 =', P_1
```

```
                N_tot = P_1/(k_erg*T)
                write(*,'(a,es12.3)') '   N_tot =', N_tot
                n_e = ZBRENT(myfunc,minv,maxv,eps)
                write(*,'(a,es12.3)') '   n_e =', n_e
                N_n = N_tot - n_e
                write(*,'(a,es12.3)') '   N_n =', N_n
                rho_n =0
                do k=1,3
                        rho_n = rho_n + alpha(k)*Am(k)
                enddo
                rho_n = m_H*N_n*rho_n
                write(*,'(a,es12.3)') '   rho_n =', rho_n
                rho_e = m_e*n_e
                write(*,'(a,es12.3)') '   rho_e =', rho_e
                rho_tot = rho_n + rho_e
                rho_tot_a(ii) = rho_tot
                write(*,'(a,es12.3)') '   rho_tot =', rho_tot_a(i)
                opacity = op(rho_tot,T)
                opacity_a(ii) = opacity
                !---pressure from avg opacity
                P_2 = P_tot_a(ii-1) + g*delta_tau/((opacity_a(ii) + opacity_a(ii-1))/2.)
                write(*,'(a,es12.3)') '   P_2 =', P_2
                if (abs(1 - P_2/P_1) < 1e-5) then
                 P_tot_a(ii) = P_1
                        write(*,'(a,a,es12.3)') '   pressures match!', '-----> P_tot =', P_tot_a(ii)
                        P_e = n_e*k_erg*T_a(ii)
                        write(*,'(a,es10.3)') '   P_e =', P_e
                        delta_z = delta_tau/(rho_tot_a(ii)*(opacity_a(ii) + opacity_a(ii-1))/2.)
                        z(ii) = z(ii-1) + delta_z
                        write(*,'(a,es10.3)') '   z =', z(ii)
                        !---Boltzmann ratios:
                        k =1; j =0
                        i =0
                        N_boltz00 = Boltz(i,j,k,T)
                        i =1
                        N_boltz10 = Boltz(i,j,k,T)
                        i =2
                        N_boltz20 = Boltz(i,j,k,T)
                        write(*,'(4(a,es10.3,5x))')
                         '   N_boltz00 =', N_boltz00,'N_boltz10 =', N_boltz10,'N_boltz20 =', N_boltz20, &
                         'total =', N_boltz00+N_boltz10+N_boltz20
                        !---Saha ratios:
                        Ntot=0
                        DO k=1,3
DO j=0,jmax(k)
N(j,k) = recsaha(j,k,n_e,T) !----ionization fractions
Ntot(k) = Ntot(k) + N(j,k)  !----check totals add up to unity for
                            !each species
if (N(j,k) < dble(1e-35)) N(j,k) = dble(1e-35) !----fix very small
                                               !numbers for supermongo input
ENDDO
ENDDO
write(*,'(a,1x,es10.3,1x,es10.3,1x,es10.3)') '   ionization fraction totals:', (Ntot(k), k=1,3)
Ntot =0
!---electron contributions per j and k:
do k=1,3
do j=1,jmax(k)
n_e_partial(j,k) = alpha(k)*N_n*j*recsaha(j,k,n_e,T)
if (n_e_partial(j,k) < dble(1e-35)) n_e_partial(j,k) = dble(1e-35) !----fix
                                                     !very small numbers for SM
enddo
enddo
!----write parameters of layer to file:
                        write(6,'(es10.3,5x,es10.3,3x,f10.2,5(3x,es10.3))') &
                         tau_a(ii), z(ii), T_a(ii), opacity_a(ii), rho_tot_a(ii), P_tot_a(ii), P_e, P_e/P_tot_a(ii)
                        write(7,'(es10.3,5x,3(es10.3,3x))') &
                         tau_a(ii), N_boltz00, N_boltz10, N_boltz20
                        write(8,'(es10.3,5x,16(es10.3,1x))') &
                         tau_a(ii), ((N(j,k), j=0,jmax(k)), k=1,3), ((n_e_partial(j,k), j=1,jmax(k)), k=1,3)
                        write(*,*)
                        ii =ii+1  !----advance to next layer
                        exit
                else
                        write(*,*) '   pressures not consistent'
                        P_1 = P_2
                 endif
        enddo
        !-----------------------------------end iterative pressure loop
        !if (ii > 5) EXIT
        if (ii > Nsteps) EXIT
ENDDO
!-------------------------------------------------------------------------end tau loop
write(*,*)
```

```
close(6); close(7); close(8)

write(*,'(a,3x,(3(f8.3,3x)))') 'abundance fractions (H,He,C):', (alpha(k), k=1,3)
write(*,*)



END PROGRAM main
```