Program LINLTE.FOR (LINFOR)
Fine analysis and spectrum synthesis of stellar spectra

M. Steffen W. Steenbock H. Holweger M. Lemke

Kiel 1981/Austin 1991

Program based on BHT (Baschek, Holweger, Trawing, 1966).
Input files:

1. ATM.DAT from ATMOS
   Model atmosphere
2. LINE.DAT
   Line data and commands
3. FALT.DAT
   Convolution (german: Faltung) data and output control of spectrum synthesis (file only needed for synthesis)
4. DEPART.DAT from NLTE system
   Departure coefficients for given element (file only needed if NLTE calculations requested)

Output files:

1. FOR003.DAT; in UNIX usually fort.3; FORTRAN channel number 3
   Line printer output of results
2. PLOT.OUT
   Synthetic spectrum (optional)

The first section of the following lists the subroutines in the sequence they are called. The second section describes 'auxiliary' routines called by routines of the first section.
Global variables, *i.e.* those which are in COMMON blocks, are printed in caps, local variables in lower case, and formal parameters to functions or subroutines in caps but in a different type face like THIS.

Flow of Program and Formulae

READIN(1) (Subroutine)

Reads model atmosphere.

- Start with reading those elements and ions from `LINE.DAT` (Part 1) for which line calculations are requested in Part 2 of the file.

`ijma`	number of elements and ions; H I always counted
`NAMIJ(I)`	1... `ijma` element and ion codes; H I = 100 always included, even with continuum.
`IFATM`	0: don't print model atmosphere; 1: do print.
`IFDEP`	0: don't print departure coefficients; 1: do print.

- Continue with model atmosphere from `ATM.DAT`:

`NMA`	number of depth points
`TEFF`	T_{eff} [K]
`GLOG`	$\log g$ [cgs]
`REFLAM`	λ_{ref} [Å] for $\tau_{\mathrm{ref}}, \kappa_{\mathrm{ref}}, \sigma_{\mathrm{ref}}$
`IMA`	number of elements
`NAMI(I)`	1... `IMA` element code = atomic number * 100
`WTI(I)`	1... `IMA` μ_i; atomic weights (^{12}C=12.000)
`ABUI(I)`	1... `IMA` $\log \varepsilon_i$; element abundances ($\log \varepsilon_{\mathrm{H}} = 12.0$)
`RTAU(N)`	1... `NMA` $\tau_{\lambda_{\mathrm{ref}}}$
`THETA(N)`	1... `NMA` $\Theta = \frac{5{,}400}{T}$
`PELG(N)`	1... `NMA` $\log p_e$; electron pressure [dyn cm^{-2}]
`PGLG(N)`	1... `NMA` $\log p_g$; gas pressure [dyn cm^{-2}]
`ALGK(N)`	1... `NMA` $\log \kappa_{\lambda_{\mathrm{ref}}}$
`ALGS(N)`	1... `NMA` $\log \sigma_{\lambda_{\mathrm{ref}}}$
`ROPTOT(N)`	1... `NMA` $\kappa_{\lambda_{\mathrm{ref}}} + \sigma_{\lambda_{\mathrm{ref}}}$
`RSU`	$R_*/R_\odot$
`R(N)`	1... `NMA` stellar radius [cm] at depth point `N`. ⇒ `R(1)` = stellar radius in [cm]
`XI(N)`	1... `NMA` ξ_{micro}; microturbulence [km/s]
`LMA`	number of wavelengths
`OLAM(L)`	1... `LMA` wavelengths λ [Å] for which continuous ...
`ALKL(N,L)`	1... `NMA, LMA` ... opacities $\log \kappa_\lambda$ and ...
`ALSL(N,L)`	1... `NMA, LMA` ... $\log \sigma_\lambda$ are read in (depth dependent).
`JMA`	= `ijma` only particle concentrations needed for the line calculations are read in
`CHIJ(J)`	1... `JMA` χ_j [eV] ionization potential
`NAMJ(J)`	1... `JMA` = `NAMIJ(I)` maybe in different sequence
`ZETAJ(N,J)`	1... `NMA, JMA` $\log \left(\frac{N_{j,0}/g_{j,0}}{N_{\mathrm{nuc}}} \right)$; particle concentrations (depth dependent); see subroutine `IONDIS`
`WTJ(J)`	1... `JMA` = `WTI(I)` (atomic weights) for `NAMJ(J)` = `NAMI(I)`

- Calculate:

HE HE $= \frac{\varepsilon_{\rm He}}{\varepsilon_{\rm H}} = 10^{\texttt{ABUI(2)}-\texttt{ABUI(1)}}$

DEL precision for equivalent width iteration; = 0.001 in DATA statement

RMIN minimum line absorption depth up to which profile will be calculated; = DEL $*$ DEL $+ 0.003$

NTMIN depth point N with lowest temperature

taum mean line formation depth; = 0.11 in DATA statement

MT depth point N with $\tau_{\rm ref} \sim$ taum

ITMA maximum number of iterations for equivalent width match; = 9 in DATA statement

PUTOUT(1) (Subroutine)

IFATM print model atmosphere if equal to 1

IFDEP print departure coefficients if equal to 1

KAPGAM(1) (Subroutine)

Calculation of line independent parts of γ.

$$\eta(\Delta\lambda) = \eta_0 \, \Phi(\Delta\lambda) = \frac{\kappa_l}{\kappa_{\rm ref}}$$

η_0 : Line opacity; frequency independent part

$\Delta\lambda$: distance from line center

$$\Phi = \begin{cases} H(\alpha, v) & \text{for metal lines} \\ \texttt{PROFHY}(\alpha', \Delta\lambda) & \text{for Hydrogen lines} \end{cases}$$

with

$$\alpha = \frac{\gamma}{2\,\Delta\omega_D} = \frac{\gamma\,\lambda_k}{2 \cdot 2\pi\,c\,v_D}$$

$$v = \frac{\Delta\lambda}{\Delta\lambda_D} = \frac{\Delta\lambda}{\lambda_k v_D}$$

$$\Delta\lambda_D = \frac{\lambda^2}{2\pi\,c}\Delta\omega_D = \lambda v_D$$

α' see KAPGAM(2), page 8

`VDOP(n,j)` Doppler velocity in units of c

$$\begin{aligned}\mathtt{VDOP}(\mathtt{n},\mathtt{j}) = v_D &= \frac{1}{c}\sqrt{\xi^2 + \frac{2RT}{N}} \\ &= \frac{1}{299{,}792.5}\sqrt{\mathtt{XI}(\mathtt{n})^2 + \frac{\mathtt{x}}{\mathtt{THETA}(\mathtt{n})}}\end{aligned}$$

with

$$\mathtt{x} = \frac{2\,R\cdot 5{,}040}{\mathtt{WTJ}(\mathtt{j})} = \frac{83.808}{\mathtt{WTJ}(\mathtt{j})}$$

Damping:

$$\gamma = \gamma_{\mathrm{rad}} + \gamma_e + \gamma_v$$

$$\begin{aligned}\gamma_v &= \gamma_{\mathrm{H}} + \gamma_{\mathrm{He}} \\ &= \gamma_{\mathrm{H}}\,(1 + G\frac{\varepsilon_{\mathrm{He}}}{\varepsilon_{\mathrm{H}}})\end{aligned}$$

with

$$G = \left(\frac{\alpha_{\mathrm{He}}}{\alpha_{\mathrm{H}}}\right)^{2/5}\left(\frac{\mu_{\mathrm{H}}}{\mu_{\mathrm{He}}}\right)^{3/10} = \frac{1}{2.4194} \quad \alpha\text{: polarizability}$$

$n = 6$:

$$\gamma_{\mathrm{H}} = x\,C_6^{2/5}\,v^{3/5}\,N_{\mathrm{H}} \quad \text{Lindholm: } x = 8.08 \tag{1}$$

$n = 4$:

$$\gamma_e = x\,C_4^{2/3}\,v^{1/3}\,N_e \quad \text{Lindholm: } x = 11.37 \tag{2}$$

$$v = \left[\frac{8\,kT}{\pi\,m_{\mathrm{H}}}\left(\frac{1}{A_1} + \frac{1}{A_2}\right)\right]^{1/2} \tag{3}$$

$$\left(\frac{1}{A_1} + \frac{1}{A_2}\right) = \begin{cases} 1 & \mathrm{H\,I} \text{ — metal} \quad A_2 \gg A_1 \\ \frac{1}{m_e/m_{\mathrm{H}}} = 1{,}837 & e \text{ — metal} \quad A_2 \gg A_1 \end{cases} \tag{4}$$

`vdwlg(n)`

$$\gamma_{\mathrm{H}} = C_6^{2/5} p_{\mathrm{H}}\,\Theta^{7/10}\left(\frac{8.08}{k\cdot 5{,}040}\right)\left(\frac{8\,k\cdot 5{,}040}{\pi\,m_{\mathrm{H}}}\right)^{3/10}$$

with (1), (3), (4), and $N_{\mathrm{H}} = \dfrac{p_{\mathrm{H}}\,\Theta}{k\cdot 5{,}040}$

$$\log \gamma_{\mathrm{H}} = \frac{2}{5}\log C_6 + \log p_{\mathrm{H}} + \frac{7}{10}\log\Theta + 8.6724 \quad [10^8\,\mathrm{sec}^{-1}]$$

$$\texttt{vdwlg(n)} = \log p_{\mathrm{H}} + \frac{7}{10}\log\Theta + 8.6724 + \log(1 + G\frac{\varepsilon_{\mathrm{He}}}{\varepsilon_{\mathrm{H}}})$$

`eldplg(n)`

$$\gamma_e = C_4^{2/3} p_e\, \Theta^{5/6} \left(\frac{8\,k\cdot 5{,}040\cdot 1{,}837}{\pi\, m_{\mathrm{H}}}\right)^{1/6} \frac{11.37}{k\cdot 5{,}040}$$

with (2), (3), (4), and $N_e = \dfrac{p_e\Theta}{k\cdot 5{,}040}$

$$\log\gamma_e = \frac{2}{3}\log C_4 + \underbrace{\log p_e + \frac{5}{6}\log\Theta + 7.7614}_{=\texttt{eldplg(n)}} \; [10^8\,\mathrm{sec}^{-1}]$$

`PHLG(n)` partial pressure of Hydrogen

$$\frac{p_{\mathrm{HI},0}}{p_g} = \frac{N_{\mathrm{HI},0}}{N_g} = \frac{N_{\mathrm{HI},0}}{N_{\mathrm{nuc}}}$$

$$N_{\mathrm{HI},0} = N_{\mathrm{nuc}}\, 10^{\texttt{ZETAJ(HI)}}\, g_{\mathrm{HI},0}$$

$$p_{\mathrm{H}} = p_g\, 10^{\texttt{ZETAJ(HI)}}\, g_{\mathrm{HI},0}; \qquad g_{\mathrm{HI},0} = 2$$

$$\texttt{PHLG(n)} = \log p_{\mathrm{H}} = \texttt{PGLG(n)} + \log 2 + \texttt{ZETAJ(n,1)}$$

READIN(2)

Reads line data and commands from `LINE.DAT` (Part 2).

Commands

`IFSURF` $= \left\{\begin{array}{lll} 2 & \cos\vartheta \geq 0 & \text{compute intensities} \\ 1 & \cos\vartheta < 0 & \text{compute fluxes} \end{array}\right\}$ control of subroutine `FLUX`

`IFSCAT` $= \left\{\begin{array}{ll} 0 & \text{scattering is true absorption} \\ 1 & \text{correct treatment} \end{array}\right\}$ scattering in the continuum; control of subroutine `FLUX`

`IFSPHA` $= \left\{\begin{array}{ll} 0 & \text{plane parallel} \\ 1 & \text{sphericity in 1st approximation} \end{array}\right.$

`IFNLTE` $= \left\{\begin{array}{ll} 0 & \text{LTE calculation} \\ 1 & \text{NLTE calculation} \end{array}\right\}$ control of subroutine `SUMETA`

IFABS $= \begin{cases} 0 & \text{write relative intensity to } \texttt{PLOT.OUT} \\ 1 & \text{absolute intensity } I_\lambda \\ 2 & \text{absolute intensity } I_\nu \end{cases}$

CONTI $= \left\{ \begin{array}{ll} \texttt{.TRUE.} & \text{compute continuum} \\ \texttt{.FALSE.} & \text{compute lines} \end{array} \right\}$ control of subroutines FLUX, SUMETA, CONT

KCONTR $= \begin{cases} \geq 0 = \texttt{K} & \text{output of depth dependent quantities for blend} \\ & \text{line } \texttt{K} \text{ or for continuum if } \texttt{K} = 1 \text{ and } \texttt{MLAM} < 0 \\ < 0 & \text{no depth dependent output} \end{cases}$

ia code for line computations; see description of LINE.DAT.

BOA(1) = .TRUE. if ia odd — W_λ from $\log gf$; output for each iteration (ia $= 3, 5, 7$)

(2) = .TRUE. if ia $= 2, 3, 6, 7$ — Fine analysis; iterate $\log gf$ to match equivalent width

(3) = .TRUE. if ia $= 4, 5, 6, 7$ — Coarse analysis; rough estimate of $\log gf$ to W_λ

(4) = .TRUE. if ia $= 8$ — Spectrum synthesis

DLOGGR $\Delta \log \gamma_{\text{rad}}$ not used for H lines

DLOGC4 $\Delta \log C_4$ not used for H lines

DLOGC6 $\Delta \log C_6$ not used for H lines

new_abu new abundance; adds new_abu$-\varepsilon_{\texttt{ATM.DAT}}$ to $\log gf$ for all lines of given element/ion

COSTHE $\cos \vartheta$ ($= -1$ for flux computation; see IFSURF)

KMA number of blend lines; for synthesis: total number of lines

BLEND(k) .TRUE. if f value of blend line k should be kept fixed (see also NEW, page 17, and *Fine analysis*, page 15)

ANFLAM λ_{start} [Å] for synthesis

ENDLAM λ_{end} [Å] for synthesis

DELLAM $\Delta\lambda$ [Å] for synthesis

ITV $= \begin{cases} 0 & \text{no terminal output during synthesis} \\ 1 & \text{do print on terminal} \end{cases}$

MLAM $\bar{\lambda}$ [Å] for continuum calculation; if < 0: depth dependent output (KCONTR $= 1$)

SINGLE .TRUE. for single line computation (*i.e.* not a blend). Compute only half of profile (WCAL $= 2$ WCAL)

WLIM minimum W_λ [Å] for suppressing output of lines with WCAL $<$ WLIM

ISYN number of point of synthetic profile for which depth dependent information will be given

Line data for blend component k, $k = 1 \ldots$ KMA

NAMK(k) element code

MULT(k) multiplet number; if MULT(k) < 0: depth dependent output of line quantities (KCONTR $=$ k)

`LAMK(k)` λ_k [Å]; wavelength at line center
`CHIK(k)` χ_k [eV]; excitation potential of lower level

`GFLG(k)` = { Metal lines: $\log gf$
Hydrogen lines: $\log K + 17$ (see Traving, 1962, *Ap. J.*, **135**, 439)

`DRRCA(k)` = { Metal lines: $\Delta\bar{r}^2/a_0^2$ for van der Waals broadening; see `KAPGAM(2)`, page 8
Hydrogen lines: n_l; principal quantum number of lower level

`C4LG(k)` = { Metal lines: $-\log C_4$; if < 0 $\gamma_e = 0$; if $= 0$ use approximation according to Griem (1968, *Phys. Rev.*, **165**, 258) and Cowley (1971, *Obs.*, **91**, 139) (see `KAPGAM(2)`, page 8).
Hydrogen lines: n_u; principal quantum number of upper level

`RAD(k)` = { Metal lines: $\gamma_{\rm rad}$; if < 0 classical formula: $\gamma_{\rm rad} = 2.22 \cdot 10^{15}\,\lambda^{-2}$ (see `KAPGAM(2)`, page 8)
Hydrogen lines: $C_e/C_{\rm H}$ (see Cayrel, Traving, 1960, *Z. Astrophys.*, **50**, 239)

`WNOTE(k)` measured W_λ; ignored if `k` $\neq$ `KMA` (*i.e.*, for all components of the blend except the last one) and in spectrum synthesis. Printed, though.
`WOBS` measured W_λ of the blend

If `DRRCA` < 0 (Unsöld approximation):
`LU(k)` l_l; orbital quantum number of valence electron of lower level
`DIU(k)` ΔI_l; excitation energy [eV] of parent term for lower level
`LO(k)` l_u; orbital quantum number of valence electron of upper level
`DIO(k)` ΔI_u; excitation energy [eV] of parent term for upper level

If `IFNLTE` $= 1$:
`LOW(k)` number of lower NLTE level in model atom
`UP(k)` number of upper NLTE level in model atom
`lowk` name (label) of lower NLTE level in model atom (set only if `LOW(k)` not numeric)
`upk` name (label) of upper NLTE level in model atom (set only if `UP(k)` not numeric)

`HY(k)` `.TRUE.` if H line (`NAMK(k)` = 100)
`ABUK(k)` abundance of element of line k (= `ABUI(i)` if `NAMK(k)` = `NAMI(i)`)
`KJ(k)` = `j` if `NAMK(k)` = `NAMJ(j)`; *i.e.*, position of line element `NAMK(k)` in array `NAMJ(j)`

- Branching synthesis—line analysis:

Spectrum synthesis

`DLK(k)` distances of line centers from start of synthesis interval: $1{,}000 \cdot (\lambda_k - \lambda_{\rm start})$ [mÅ]; see also figures on page 14
`DELLAM` rescaling of `DELLAM` to mÅ: `DELLAM` $= 1{,}000 *$ `DELLAM`

Line analysis see also figures page on 14

`MLAM` $\bar{\lambda} = \sum_{n=1}^{\mathtt{KMA}} \lambda_k / \mathtt{KMA}$

`DLK(k)` $\Delta\lambda_k = 1{,}000 \cdot (\lambda_k - \lambda_{k,\min})$; distances of line centers (in a blend) from blend line with smallest wavelength [mÅ]

`DLDOP` Doppler width for estimate of integration end: $\Delta\lambda_D = 1{,}000 \cdot \bar{\lambda}\, v_D$ [mÅ] with v_D taken at depth point `MT` (typical line formation depth) for element `KJ`(|`KCONTR`|) (Remark: |`KCONTR`| chooses different v_D if `MULT(k)` < 0. Intended?)

`ENDLAM` $\mathtt{DLK(k)}_{\max} + 2\,\Delta\lambda_D$; wavelength for integration end

LINLTE (Main program)

- For continuum calculation (`CONTI = .TRUE.`):

`FCLAM` `= CONT(MLAM)`; $I_{\bar{\lambda},c}$ or $F_{\bar{\lambda},c}$

Then do

- `CALL PUTOUT(2)`; output continuum quantities

and

- `GO TO READIN(2)` (page 5); continue reading `LINE.DAT`

- else for line calculations `CONTI = .FALSE.`:

KAPGAM(2)

Calculation of line dependent parts of γ, α, and η. See also `KAPGAM(1)`, page 3. NLTE accounted for in function `SUMETA`, page 34.

Metal lines

`ETA0(n,k)`

$$\begin{aligned}
\eta^*_{0,k} &= \frac{\kappa^*_{j,s}\, N^*_{j,s}}{(\kappa_{\mathrm{ref}} + \sigma_{\mathrm{ref}})\, N_{\mathrm{nuc}}} \quad \text{atom/ion } j\text{, lower level } s \\
&= \frac{\kappa^*_{j,s}}{(\kappa_{\mathrm{ref}} + \sigma_{\mathrm{ref}})} g_{j,s}\, 10^{-\chi_{rs}\,\Theta}\, \underbrace{\frac{N^*_{j,0}/g_{j,0}}{N_{\mathrm{nuc}}}}_{10^{\mathtt{ZETAJ}}} \\
&= \frac{2\,\pi^{3/2}\, e^2}{m_e\, c\ \underbrace{\Delta\omega_D}_{\frac{2\,\pi\,c}{\lambda_k} v_D}} \left(1 - e^{-c_2/\lambda_k\, T}\right) \frac{f}{(\kappa_{\mathrm{ref}} + \sigma_{\mathrm{ref}})}\, g_{j,s}\, 10^{-\chi_k \Theta}\, 10^{\mathtt{ZETAJ}} \\
&= \frac{\pi^{1/2} e^2 10^{-8}\, gf}{m_e\, c^2} \frac{\lambda_k}{v_D}\, 10^{\mathtt{ZETAJ(N,J)}}\, 10^{-\chi_k \Theta} \frac{(1 - 10^{-\mathtt{v}\,\Theta})}{(\kappa_{\mathrm{ref}} + \sigma_{\mathrm{ref}})} \quad (\lambda_k \text{ in [Å]})
\end{aligned}$$

with

$$\mathtt{v} = \frac{12{,}398.54}{\lambda_k\ [\text{Å}]}\ [\text{eV}]$$

Stepwise calculation:

$$\mathtt{w}' = \log \frac{\pi^{1/2} e^2 10^{-8}\, gf}{m_e\, c^2} = -20.3015 + \log gf$$

$$\mathtt{w} = \mathtt{w}' + \log \lambda_k$$

$$\mathtt{z} = 1 - 10^{-\mathtt{v}\,\Theta}$$

$$\mathtt{ETA0(n,k)}' = 10^{(\mathtt{w+ZETAJ(n,KJ(k))-CHIK(k)\,THETA(n)})} \mathtt{z} / \underbrace{\mathtt{ROPTOT(n)}}_{\kappa_{\mathrm{ref}}+\sigma_{\mathrm{ref}}}$$

$$\mathtt{ETA0(n,k)} = \mathtt{ETA0(n,k)}'/\mathtt{VDOP(n,KJ(k))}$$

If `DRRCA` ≤ -0.1:

`RRCA` Function; Unsöld approximation

$$\frac{\bar{r}^2}{a_0^2} = \frac{n_{\mathrm{eff}}^2}{2\,(Z+1)^2}\left(5\,n_{\mathrm{eff}}^2 + 1 - 3\,l(l+1)\right)$$

$$n_{\mathrm{eff}}^2 = \frac{(Z+1)^2 \cdot 13.598}{\chi_j + \Delta I_k - \chi_k}$$

with χ_j = ionization energy of element j, χ_k level energy, ΔI_k excitation energy of parent term, and Z charge of ion.

`RRCAU(k)` $= \mathtt{RRCA}(\chi_k, \Delta I_l + \chi_j, l_l, \mathtt{iz})$; $\mathtt{iz} = Z + 1 = 1$ for neutrals, 2 for singly ionized etc.

`RRCAO(k)` $= \mathtt{RRCA}(\chi_k + v, \Delta I_u + \chi_j, l_u, \mathtt{iz})$

$v = \frac{12398.54}{\lambda_k\ [\mathrm{Å}]}$ [eV] $= h\nu \quad \Rightarrow \chi_k + v =$ energy of upper level

`DRRCA(k)`

$$\begin{aligned} \frac{\Delta\bar{r}^2}{a_0^2} &= \left|\frac{\Delta\bar{r}_u^2}{a_0^2}\right| - \left|\frac{\Delta\bar{r}_l^2}{a_0^2}\right| \\ &= |\mathtt{RRCAO(k)}| - |\mathtt{RRCAU(k)}| \end{aligned}$$

$$\mathtt{DRRCA(k)} = \left|\frac{\Delta\bar{r}^2}{a_0^2}\right| \cdot 10^{\Delta \log C_6}$$

`C6LG(k)`
$$\begin{aligned} &= -\log C_6 - \Delta \log C_6 \\ &= -\log \frac{\Delta\bar{r}^2}{a_0^2} - \log \frac{e^2 \alpha_{\mathrm{H}}\, a_0^2}{\hbar} - \Delta \log C_6 \\ &= \mathtt{DRRCA(k)} - 32.3867 \end{aligned}$$

`C4LG(k)`
$$= -\log C_4 - \Delta \log C_4$$

$$= \texttt{C4LG(k)} - \texttt{DLOGC4}$$

For $\texttt{C4LG(k)} = 0$ use Griem/Cowley approximation at 10,000 K:

$$\gamma_e = \begin{cases} \underbrace{\frac{1}{2\pi\, v_{10{,}000}} \left(\frac{h}{m_e\,(Z+1)}\right)^2 \frac{5}{3}}_{\texttt{c4fak}} n_{\mathrm{eff}}^4\, N_e & \text{neutrals, } \textit{i.e.}, Z = 0 \\ \underbrace{\frac{1}{2\pi\, v_{10{,}000}} \left(\frac{h}{m_e\,(Z+1)}\right)^2 4}_{\texttt{c4fak}}\, n_{\mathrm{eff}}^4\, N_e & \text{ions} \end{cases}$$

with (3) and (4) for $T_{\mathrm{eff}} = 10{,}000\,\mathrm{K}$:

$$v_{10{,}000} = \left(\frac{8\,k \cdot 10{,}000}{\pi\, m_{\mathrm{H}}}\, 1{,}837\right)^{\frac{1}{2}} = 6.213 \cdot 10^7$$

$\Rightarrow$

c4fak

$$= \begin{cases} 2.259 \cdot 10^{-7} & \text{neutrals} \\ 5.421 \cdot 10^{-7}/(Z+1)^2 & \text{ions} \end{cases}$$

and with $N_e = \dfrac{P_{\mathrm{H}}\,\Theta}{k \cdot 5{,}040}$ (in units of 10^8 rad/sec)

$\Rightarrow$

$$\gamma_e = \texttt{c4fak}\, n_{\mathrm{eff}}^4\, p_{\mathrm{H}}\, \Theta\, \underbrace{\frac{10^{-8}}{k \cdot 5{,}040}}_{\texttt{c4konst}} \tag{5}$$

With (2) and (5) we obtain:

$$\begin{aligned} C_4^{2/3} &= \frac{\texttt{c4fak}}{11.37}\, n_{\mathrm{eff}}^4\, v^{-\frac{1}{3}} \\ C_4 &= \left(\frac{\texttt{c4fak}}{11.37}\, n_{\mathrm{eff}}^4\right)^{\frac{3}{2}} v^{-\frac{1}{2}} \end{aligned}$$

or

$$\log C_4 = 1.5 \log(\texttt{c4fak}/11.37) + 1.5 \log(n_{\mathrm{eff}}^2) - 3.9$$

`RAD(k)` $\gamma_{\mathrm{rad}} \cdot 10^{\Delta \log \gamma_{\mathrm{rad}}}$ $[10^8\ \mathrm{sec}^{-1}]$

For `RAD(k)` < -0.001 use classical formula:

$$\begin{aligned}\gamma_{\mathrm{rad}} &= \frac{2\,e^2\omega^2}{3\,m\,c^3}\\ &= \frac{2\,e^2\,4\,\pi^2}{3\,m\,c\,\lambda_k^2}\\ &= \frac{2.22\cdot 10^7}{\lambda_k^2}\quad \left[10^8\,\mathrm{sec}^{-1}\right];\ \ \lambda \text{ in Å}\end{aligned}$$

`ALPHA(n,k)`

$$\begin{aligned}\alpha(\mathtt{n},\mathtt{k}) &= \frac{\gamma}{2\,\Delta\omega_D} = \frac{\gamma\,\lambda_k}{4\,\pi\,c\,v_D};\quad \gamma = \gamma_e + \gamma_{\mathrm{rad}} + \gamma_v\\ &= \left(10^{-2/3\ \log C_4 + \mathtt{eldplg}} + \gamma_{\mathrm{rad}} + 10^{-4/10\ \log C_6 + \mathtt{vdwlg}}\right)\frac{\overbrace{\lambda_k}^{\text{[Å]: }10^{-8}\text{ against }\gamma\ (10^8)}}{v_D\,\underbrace{4\,\pi\,c}_{37.673\cdot 10^{10}}}\end{aligned}$$

see also `KAPGAM(1)`, page 3

Hydrogen lines; if `HY(k) = .TRUE.`

`GFLG(k)` $\log K + 17$; see Traving, 1962, *Ap. J.*, **135**, 439

`DRRCA(k)` n_l

`ETA0(n,k)`

$$\eta_0 = (2.603\,e)^{3/2}\,gK\,10^{\mathtt{ZETAJ(n,1)}}\,\frac{1 - e^{-c_2/\lambda_k T}}{\kappa_{\mathrm{ref}} + \sigma_{\mathrm{ref}}}\,\frac{p_e}{kT}\,10^{-\chi_k\,\Theta}$$

$$g = 2\,n_l^2$$

$$\begin{aligned}\mathtt{w} &= \underbrace{\mathtt{w}'}\ + 2.4054 + 2\log n_l\\ \text{see page 9; } &= -20.3015 + \mathtt{GFLG};\ \mathtt{GFLG} = \log K + 17\\ &= -20.3015 + \log K + 17 + 2\log n_l + 2.4054\\ &= \frac{3}{2}\log(2.603\,e) + \log 2 - \log(k\cdot 5{,}040) + 2\log n_l + \log K\end{aligned}$$

Original write up contains a pretty mess trying to explain the numerical constants. Good luck! Try BHT or Traving, 1962, *Ap. J.*, **135**, 439.

$$\mathtt{ETA0(n,k)} = \underbrace{\mathtt{ETA0(n,k)}'}_{\text{see page 9; new }\mathtt{w}}\ \Theta\,10^{\log p_e}$$

Set

RRCAO(k) $= 0$
RRCAU(k) $= 0$
DRRCA(k) $= 0$ (but why?)
C4LG(k) n_u
ANM see BHT:

$$\texttt{ANM} = A_{nm} = 0.06479\frac{\frac{n_u^3}{n_l^2}+\frac{n_l^3}{n_u^2}}{\sqrt{n_u^2+n_l^2}}$$

ALPHA(n,k)

$$\begin{aligned} \alpha' &= A_{nm}\sqrt{\Theta}\left(4.1007-\log(\sqrt{p_e}n_u^2\,\Theta^{3/2})\right) \\ &= A_{nm}\sqrt{\Theta}(4.1007-\frac{1}{2}\log p_e - 2\log n_u - \frac{3}{2}\log\Theta) \end{aligned}$$

Main Program: Branch synthesis—line analysis

Synthesis

MLAM

$$\bar\lambda = \begin{cases} \frac{1}{2}(\lambda_{\text{end}}-\lambda_{\text{start}}) & \text{for } \lambda_{\text{end}}-\lambda_{\text{start}} \le 1\,\text{Å} \\ \text{center of } 1\,\text{Å wide subintervals for which} & \\ F_{\lambda,c},\ B_{\lambda,c},\ \eta_{\lambda,c}, \text{ and } \sigma_{\lambda,c} \text{ are recalculated} & \text{else} \\ \text{in progress of synthesis} & \end{cases}$$

FCLAM $I_{\bar\lambda,c}$ or $F_{\bar\lambda,c}$; $=$ CONT(MLAM)
dl distance from λ_{start} in mÅ

□ Do for whole synthesis interval $[\lambda_{\text{start}}, \lambda_{\text{end}}]$:
LASCAN(m) wavelength array [Å]; $= \lambda_{\text{start}} +$ dl/1,000
RSCAN(m) calculated spectrum; continuum normalized to 10,000:

RSCAN(m) = 10,000 · FLUX(dl)/FCLAM

- Printout on terminal if ITV = 1
- dl = dl + DELLAM

□ when done:
MMA number of calculated points (LASCAN, RSCAN)

- `CALL PUTOUT(3)`: printout of line data
- `CALL FALT`: convolve spectrum and write/print spectrum
- `GO TO READIN(2)` (page 5): read new data

Line analysis

- first

`FCLAM` $F_{\bar\lambda,c} = \texttt{CONT(MLAM)}$
with

`MLAM` $\texttt{MLAM} = \bar\lambda = \sum_{k=1}^{\texttt{KMA}} \lambda_k / \texttt{KMA}$

Coarse analysis if `BOA(3) = .TRUE.`

1. reduce f value (and η_0) until flux at center of first blend line (λ_0) is less than flux 5 mÅ shortward of λ_0, *i.e.*, no emission line.

   Done with `NEW(−1)` (*i.e.*, reduce f value by a factor of 10; see subroutine `NEW`, page 17) until `FLUX(0)` $\leq$ `FLUX(−5)`.

2. repeat calling `NEW(v)` with $\texttt{v} = \Delta \log gf = \log\left(\frac{R_c}{r(0)} - 1\right)$ until

$$\begin{aligned} 0.1 \geq \log\left(\frac{F_{\bar\lambda,c} - B(\texttt{NTMIN})}{F_{\bar\lambda,c} - F_{\lambda_0}} - 1\right) &= \log\left(\frac{1 - \frac{B}{F_{\bar\lambda,c}}}{1 - \frac{F_{\lambda_0}}{F_{\bar\lambda,c}}} - 1\right) \\ &= \log\left(\frac{R_c}{r(0)} - 1\right) \end{aligned}$$

   i.e.

   $R_c \sim r(0)$

   then call `NEW( UCG(WOBS) )` to get new f value from observed equivalent width.

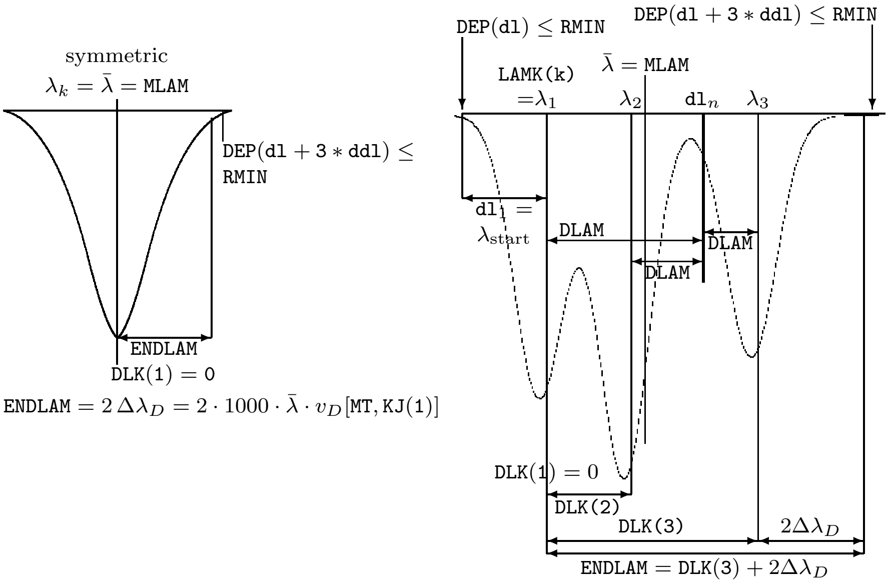
This procedure adjusts the abundance so that the abscissa of the universal curve of growth (function `UCG`), $\log C = \log gf\varepsilon + \text{const}$, equals 0. See also BHT eqn. 32 a,b.

Calculation of equivalent width

Start values

Set `IT = 0`, `WCAL = 0`, `nit = 0`.
Profile is calculated at wavelength point `dl` initialized as

	SINGLE = .TRUE.	SINGLE = .FALSE.
`dl`	`dl` = 0; line center [mÅ]	`dl` = -5, -10, -20, -40, -80, -160… until $$\mathtt{DEP(dl)} = 1 - \frac{\mathtt{FLUX(dl)}}{F_{\bar{\lambda},c}} \le \mathtt{RMIN},$$ that is, start left from first blend line where absorption depth is less than `RMIN`.
`ddl`	`ddl` $= 10\,\text{mÅ} = \Delta x$ [mÅ]; start value for wavelength increment in integration. Half of profile is integrated.	`ddl` $= -\mathtt{dl}/16$ Whole profile is integrated.


symmetric
$\lambda_k = \bar{\lambda} = \mathtt{MLAM}$
$\mathtt{DEP(dl + 3 * ddl)} \le$ RMIN
ENDLAM
$\mathtt{DLK(1)} = 0$
$\mathtt{ENDLAM} = 2\,\Delta\lambda_D = 2 \cdot 1000 \cdot \bar{\lambda} \cdot v_D[\mathtt{MT,KJ(1)}]$
$\mathtt{DEP(dl)} \le \mathtt{RMIN}$
$\mathtt{DEP(dl + 3 * ddl)} \le \mathtt{RMIN}$
$\bar{\lambda} = \mathtt{MLAM}$
LAMK(k)
$=\lambda_1$
λ_2
$\mathtt{dl}_n$
λ_3
$\mathtt{dl}_1 =$
λ_{start}
DLAM
DLAM
DLAM
$\mathtt{DLK(1)} = 0$
DLK(2)
DLK(3)
$2\Delta\lambda_D$
$\mathtt{ENDLAM} = \mathtt{DLK(3)} + 2\Delta\lambda_D$

Integration

`WCAL`

$$W_\lambda = \mathtt{WCAL} = \int_{\mathtt{dl}}^{\cdots} \mathtt{DEP(dl)}\, d\mathtt{dl}$$

`DEP(dl)`: absorption depth; see function `DEP` on page 17.

$$\begin{aligned}
\texttt{WCAL} &= \int y(x)\,dx \\
x_{v+n} &= x_v + n\,\underbrace{\Delta x}_{\texttt{ddl}} \\
y_{v+n} &= \texttt{DEP}(x_{v+n}) \\
&= \texttt{DEP}(\texttt{dl} + n\,\Delta x) \\
\texttt{WCAL} &= \texttt{WCAL} + \frac{\Delta x}{3}\left(\frac{14}{15}y_v + \frac{64}{15}y_{v+1} + \frac{24}{15}y_{v+2} + \frac{64}{15}y_{v+3} + \frac{14}{15}y_{v+4}\right) \\
x_v &= x_v + 4\,\Delta x \quad (\texttt{dl} = \texttt{dl} + 4 * \texttt{ddl})
\end{aligned}$$

4-step Simpson rule with automatic adjustment of increment Δx:

$$\Delta x = \begin{cases}
2\,\Delta x & \text{if } \texttt{dl} = x_v > -1 \\
0.5\,\Delta x & \text{if } |y_{v+4} - y_{v+2}| > 0.1 \qquad \text{change of absorption} \\
 & \text{or } |4\,y_{v+3} - 4\,y_{v+1}| > 0.4 \qquad \text{depth too large; } > 0.1 \\
 & \text{or } |y_{v+2} - y_v| > 0.1 \\
 & \text{and } \Delta x > 19 \qquad \text{but keep minimum step} \\
 & \qquad\qquad\qquad\qquad \text{at about } 10\,\text{mÅ}
\end{cases}$$

Integration end

`ENDLAM` $2\,\Delta\lambda_D + \texttt{DLK(KMA)}$
Stop integration when $\texttt{dl} \geq \texttt{ENDLAM}$ and $\texttt{DEP}(\texttt{dl} + 3 * \texttt{ddl}) \leq \texttt{RMIN}$ (see figure).
If single line set $\texttt{WCAL} = 2 * \texttt{WCAL}$.

Fine analysis if `BOA(2) = .TRUE.`

`IT`
`ITMA`

while $\texttt{IT} \leq \underbrace{\texttt{ITMA}}_{=9}$

and

$$\left|\log\frac{\texttt{WOBS}}{\texttt{WCAL}}\right| < 0.87 \cdot \underbrace{\texttt{DEL}}_{=0.001}\ :$$

Using subroutine `ORDER`, sort the first $\log gf$ which is not to be kept fixed (*i.e.*, `BLEND(k) = .FALSE.`) and the corresponding W_{cal} in arrays `GFLGIT` and `WCLGIT`, with increasing W_{cal}.

Calculation of $\Delta \log gf$

1^{st} Iteration	$\Delta \log gf$ = `UCG(WOBS)` − `UCG(WCAL)`; estimate for 2nd f value
$2^{\text{nd}} + \text{n}^{\text{th}}$ Iteration	Interpolate or extrapolate in `GFLGIT` for log `WOBS`; linearly if `IT` $= 2$ and quadratic for `IT` > 2. See function `POL`, page 18.

Then call `NEW(`$\Delta \log gf$`)` and repeat integration (page 14).

End

when $\left|\log \frac{\texttt{WOBS}}{\texttt{WCAL}}\right| < 0.87 \cdot$ `DEL` or `IT` $>$ `ITMA`:

Output of results and line data; if `KCONTR` ≥ 0 depth dependent.

- `GO TO READIN(2)` (page 5): read new data

Subroutines and Functions

The following subroutines and functions are called from various places of the program sections listed above and not in the sequence as it appears here.

CONT(MLAM) (Function)

Calculates $B_{\bar\lambda,c}(T)$, $\sigma_{\bar\lambda,c}$, $\eta_{\bar\lambda,c}$, $I_{\bar\lambda,c}$ or $F_{\bar\lambda,c}$ for all depth points.

MLAM	`REAL`	input	$\bar\lambda$ [Å]

`BLAM(n)`

$$B_{\bar\lambda,c} = \frac{2\,h\,c^2 \cdot 10^{40}}{\bar\lambda^5}\,\frac{1}{e^{\frac{\Theta}{\bar\lambda}\frac{h\,c\cdot 10^8}{k\cdot 5{,}040}} - 1} = \frac{a}{\bar\lambda^5}\,\frac{1}{e^{b\,\Theta/\bar\lambda} - 1}$$

with

$a = 1.1911 \cdot 10^{35}$ $\qquad b = 28{,}548$

`SIGMAC(n)` $\sigma_{\bar\lambda,c}$; interpolated in `ALSL(n,1)` for $\bar\lambda$ (`MLAM`, parabolic).

`ETAC(n)`

$$= \eta_{\bar\lambda,c} = \frac{\kappa_{\bar\lambda,c} + \sigma_{\bar\lambda,c}}{\underbrace{\kappa_{\text{ref}} + \sigma_{\text{ref}}}_{\texttt{ROPTOT(n)}}}$$

with $\kappa_{\bar{\lambda},c}$ interpolated in `ALKL(n,l)` for `MLAM`.

CONT $(I_{\bar{\lambda},c}$ or $F_{\bar{\lambda},c}) =$ `FLUX(0)` with `CONTI = .TRUE.`.
`FLUX` is the astrophysical flux, *i.e.,*

$$F_\lambda = \frac{1}{\pi}\mathcal{F}_\lambda = 4\,H_\lambda \qquad \left[\frac{\mathrm{erg}}{\mathrm{s}\cdot\mathrm{cm}^2\cdot\text{Å}}\right]$$

NEW(DEPS) (Subroutine)

DEPS	`REAL`	input	$\Delta\log gf$

Computes for all lines k with `BLEND(k) = .FALSE.`:

$$\begin{array}{lllll} 1) & \log gf & = & \log gf + \Delta\log gf \\ 2) & \mathtt{ETA0(n,k)} & = & \mathtt{ETA0(n,k)}\cdot 10^{\Delta\log gf} \end{array}$$

DEP(DL) (Function)

Calculates absorption depth.

DL	`REAL`	input	λ

`LASCAN(m)`
`RSCAN(m)`
`MMA`

Looks in array `LASCAN(1:MMA)` for λ. If found return in DEP absorption depth from array `RSCAN(1:MMA)`. If λ not there calculate absorption depth as

$$\mathtt{RSCAN(m)} = 1 - \mathtt{FLUX}(\lambda)/\mathtt{FCLAM}$$

and sort results into arrays `LASCAN(m)` and `RSCAN(m)`. Thus these arrays contain the line profile points sorted by wavelength.

DEP absorption depth obtained as described above

ORDER(X,Y,A,B,N) (Subroutine)

Sorts table.

X	`REAL`	input
Y	`REAL`	input
A(N)	`REAL`	modified
B(N)	`REAL`	modified
N	`INTEGER`	modified

Inserts function value $y(x)$ into table $b_i(a_i) \quad (1 \le i \le n)$. Table size increased by 1, *i.e.,* $\mathsf{N} = \mathsf{N} + 1$.

UCG(W) (Function)

Universal Curve of Growth; see Hunger, 1956, *Z. f. Astrophys.* **39**, 36.

W	`REAL`	input	Equivalent width

$$R_c = 1 - \frac{B_\lambda(\mathtt{NTMIN})}{\mathtt{FCLAM}}$$

$$\Omega = \log \frac{\mathsf{W}}{2\,\Delta\lambda_D\,R_c}$$

$$\mathsf{UCG} = \begin{cases} \Omega & \Omega < -1 \\ \Omega + [\Omega - \log(\,2\,\alpha(\mathtt{MT},1)\,)]\,(\Omega+1)\dfrac{2}{3} & -1 \le \Omega < 0.5 \\ 2\,\Omega - \log(\,2\,\alpha(\mathtt{MT},1)\,) & \Omega \ge 0.5 \end{cases}$$

POL(X,A,B,MI,MA,L) (Function)

Inter- and extrapolation routine.

X	`REAL`	input	place x to evaluate $b_i(a_i)$
A(1)	`REAL`	input	independent variable a_i
B(1)	`REAL`	input	dependent variable b_i
MI	`INTEGER`	input	lower i where function defined
MA	`INTEGER`	input	upper i where function defined
L	`INTEGER`	input	control parameter

$$\mathsf{POL} = \begin{cases} b_i(x) & \mathsf{MI} \le i \le \mathsf{MA} \\ \text{extrapolated} & \text{else} \end{cases}$$

Extrapolation is always linear. Interpolation controled by L:

L =

1 : linear interpolation

2 : linear, parabolic, hyperbolic interpolation:
If MA − MI $= 0$, *i.e.*, two points only, linear interpolation.
Else if b is monotonic in $[a_0, a_2]$ do hyperbolic interpolation:

$$b(x) = b(a_0) + (x - a_0)[b_0 b_1] \left\{ 1 - (x - a_1) \frac{[b_0 b_1 b_2]}{[b_0 b_2]} \right\}^{-1}$$

else parabolic interpolation:

$$b(x) = b(a_0) + (x - a_0) \left\{ [b_0 b_1] + (x - a_1)[b_0 b_1 b_2] \right\}$$

with

$$[b_i b_j] = \frac{b(a_i) - b(a_j)}{a_i - a_j}$$

$$[b_i b_j b_k] = \frac{[b_i b_j] - [b_j b_k]}{a_i - a_k}$$

FLUX(DL) (Function)

Computes flux or intensity.
Controled by `IFSCAT`, `IFSURF`, `IFSPHA`, and `CONTI` (Function `SUMETA`).

DL	`REAL`	input	Distance ([mÅ]) from line/synthesis center

- call `SUMETA(n,DL,sumet2)`, controled by `CONTI`

`BETA(n)` Underlined terms taken only for line calculation, *i.e.* `CONTI = .FALSE.`. Also, $\lambda = \bar{\lambda}$ for continuum calculation.

$$\begin{aligned}\beta_\lambda &= \frac{\sigma_{\bar{\lambda},c}}{\kappa_{\bar{\lambda},c} + \sigma_{\bar{\lambda},c} + \underline{\sum_{i=1}^{\mathtt{KMA}} \kappa_{\lambda,l_i}}} \\ &= \frac{\sigma_{\bar{\lambda},c}}{(\kappa_{\text{ref}} + \sigma_{\text{ref}}) \sum_{i=1}^{\mathtt{KMA}} \eta_i}\end{aligned}$$

with

`dum(n)`

$$\sum_{i=1}^{\mathtt{KMA}} \eta_i = \frac{\kappa_{\bar{\lambda},c} + \sigma_{\bar{\lambda},c} + \underline{\sum_{i=1}^{\mathtt{KMA}} \kappa_{\lambda,l_i}}}{\kappa_{\text{ref}} + \sigma_{\text{ref}}}$$

`sc`

$$\frac{\sigma_{\bar{\lambda},c}}{\kappa_{\text{ref}} + \sigma_{\text{ref}}}$$

`sumet2`

$$\frac{\kappa_{\bar{\lambda},c} + \sigma_{\bar{\lambda},c} + \sum_{k=1}^{\mathtt{KMA}} \kappa^*_{\lambda,l_k} b_{u_k}}{\kappa_{\text{ref}} + \sigma_{\text{ref}}}$$

`strue(n)` $\bar{S}_\lambda$; source function for true absorption, *i.e.*, without scattering term:

$$
\begin{aligned}
\bar{S}_\lambda &= C\,\frac{1}{\frac{b_l}{b_u}e^{+c}-1} \\
&= C\,\frac{b_u}{b_l e^{+c}-b_u} \\
&= C\,\frac{b_u}{\frac{b_l}{e^{-c}}-b_u} \\
&= C\,\frac{b_u e^{-c}}{b_l-b_u e^{-c}}
\end{aligned}
$$

$\Rightarrow$

$$B_\lambda = C\frac{e^{-c}}{1-e^{-c}}$$

$\Rightarrow$

$$\frac{\bar{S}_\lambda}{B_\lambda} = \frac{b_u(1-e^{-c})}{b_l-b_u e^{-c}}$$

with

$$
\begin{aligned}
\kappa &= C'\,(b_l-b_u e^{-c}) \\
\kappa^* &= C'\,(1-e^{-c})
\end{aligned}
$$

$\Rightarrow$

$$
\begin{aligned}
\frac{\bar{S}_\lambda}{B_\lambda} &= \frac{b_u\kappa^*}{\kappa} \\
\bar{S}_\lambda &= \frac{b_u\kappa^*}{\kappa}B_\lambda
\end{aligned}
$$

$\Rightarrow$

$$\texttt{strue(n)} = \frac{\texttt{sumet2}-\texttt{sc}}{\texttt{dum(n)}-\texttt{sc}}\,\texttt{BLAM(n)}$$

`taulam(n)` $$\tau_\lambda(n) = \int\sum_{i=1}^{\texttt{KMA}}\eta_i(n)\,d\tau_{\rm ref}$$

$$= \int_{\tau_1}^{\tau} \frac{\kappa_{\bar{\lambda},c} + \sigma_{\bar{\lambda},c} + \sum_{i=1}^{\mathtt{KMA}} \kappa_{\lambda,l_i}}{\kappa_{\mathrm{ref}} + \sigma_{\mathrm{ref}}} \, d\tau_{\mathrm{ref}}$$

with

$$\tau_1 = \sum_{i=1}^{\mathtt{KMA}} \eta_i(1)\tau_{\mathrm{ref}}(1) \quad \text{finite starting } \tau$$

Integration with subroutine `INTEG` by Kurucz, see page 36.
$\tau_1 = 0$ after integration.

Scattering treated as true absorption (`IFSCAT = 0`)

`SLAM(n)` $S_\lambda(n) = (1 - \beta_\lambda)\bar{S}_\lambda + \beta_\lambda B_\lambda(n)$

For flux calculation (`IFSURF = 1`):

For spherical approximation set

$$S_\lambda = \left(\frac{R}{R_\odot}\right)^2 S_\lambda$$

Interpolate S_λ to fixed τ scale with subroutine `MAP1` by Kurucz for pretabulated Φ operator in matrix form.

$$S_\lambda(\tau_\lambda)_{1\ldots\mathtt{NMA}} \rightarrow S'_\lambda(\tau'_\lambda)_{1\ldots\mathtt{NXTAU}}$$

`NXTAU` Number of points in fixed τ scale (43)
`XTAU(l)` 1...`NXTAU` τ'_λ; fixed τ scale for matrix operators, see also page 28.
`xs(l)` 1...`NXTAU` S'_λ; S_λ interpolated to fixed τ scale τ'_λ.

Scattering treated correctly (`IFSCAT = 1`)

For spherical approximation in flux calculation set

$$\bar{S}_\lambda = \left(\frac{R}{R_\odot}\right)^2 \bar{S}_\lambda$$

S_λ is calculated on fixed τ scale for integration matrices, which depend only on τ and $\Delta\tau$. For doing this, interpolate $\bar{S}_\lambda$ and β_λ to fixed τ scale τ'_λ with subroutine `MAP1` by Kurucz, see page 39.

$$\bar{S}_\lambda(\tau_\lambda)_{1\ldots\mathtt{NMA}} \quad \longrightarrow \quad \bar{S}'_\lambda(\tau'_\lambda)_{1\ldots\mathtt{NXTAU}} \tag{6}$$

$$\beta_\lambda(\tau_\lambda)_{1\ldots\mathtt{NMA}} \quad \longrightarrow \quad \beta'_\lambda(\tau'_\lambda)_{1\ldots\mathtt{NXTAU}} \tag{7}$$

`NXTAU` Number of points in fixed τ scale (43)

`XTAU(l)` 1. . . `NXTAU` τ'_λ; fixed τ scale for matrix operators, see also page 28.

Then calculate $S'_\lambda(\tau'_\lambda)$ (see below) and do inverse transformation:

$$S'_\lambda(\tau'_\lambda)_{1\ldots\mathtt{NXTAU}} \longrightarrow S_\lambda(\tau_\lambda)_{1\ldots\mathtt{NMA}} \tag{8}$$

Calculation of S_λ

We omit primes in the following, *i.e.*, we write S_λ instead of S'_λ etc.

$$S_\lambda = (1-\beta_\lambda)\,\bar{S}_\lambda + \beta_\lambda J_\lambda \tag{9}$$

with

$$\begin{array}{rcl}
\bar{S}_\lambda & & \text{Source function for true absorption} \\
\beta_\lambda & = & \dfrac{\sigma_\lambda}{\kappa_\lambda + \sigma_\lambda} \\
1-\beta_\lambda & = & \dfrac{\kappa_\lambda}{\kappa_\lambda + \sigma_\lambda} \\
\kappa_\lambda & = & \kappa_{\bar{\lambda},c} \quad \text{if continuum only} \\
 & = & \kappa_{\bar{\lambda},c} + \kappa_{\lambda,l} \quad \text{if lines included}
\end{array}$$

Substitute $J_\lambda = \Lambda S_\lambda$ in (9) $\Rightarrow$

$$S_\lambda = (1-\beta_\lambda)\bar{S}_\lambda + \beta_\lambda \Lambda\, S_\lambda$$

$\vec{S} = (\mathbf{I}-\boldsymbol{\beta})\vec{\bar{S}} + \boldsymbol{\beta}\boldsymbol{\Lambda}\vec{S}$ $\quad$ $\boldsymbol{\beta}$ diagonal; $\mathbf{I}$: unity matrix

$$(\mathbf{I}-\boldsymbol{\beta}\,\boldsymbol{\Lambda})\vec{S} = (\mathbf{I}-\boldsymbol{\beta})\vec{\bar{S}} \tag{10}$$

$\Rightarrow$

$$(\mathbf{I}-\boldsymbol{\beta}\,\boldsymbol{\Lambda})\vec{S} - (\mathbf{I}-\boldsymbol{\beta})\vec{\bar{S}} = 0 \tag{11}$$

Use Gauß-Seidel iteration scheme to solve eqn. 11

Iteration j:

$$\Delta^j = (\mathbf{I}-\boldsymbol{\beta}\,\boldsymbol{\Lambda})\vec{S}^{j+1} - (\mathbf{I}-\boldsymbol{\beta})\vec{\bar{S}} \tag{12}$$

$$\text{(10) for } \vec{S}^j = \vec{S}^{j-1} + \Delta\vec{S}^j\ : \tag{13}$$

$$(\mathbf{I}-\boldsymbol{\beta}\,\boldsymbol{\Lambda})(\vec{S}^{j-1} + \Delta\vec{S}^j) = (\mathbf{I}-\boldsymbol{\beta})\vec{\bar{S}} \tag{14}$$

Substitute (14) for $(\mathbf{I}-\boldsymbol{\beta})\vec{\bar{S}}$ in (12)

$$\begin{aligned}
\Delta^j &= (\mathbf{I}-\boldsymbol{\beta}\,\boldsymbol{\Lambda})\vec{S}^{j-1} - (\mathbf{I}-\boldsymbol{\beta}\,\boldsymbol{\Lambda})(\vec{S}^{j-1} + \Delta\vec{S}^j) \qquad (15)\\
&= (\mathbf{I}-\boldsymbol{\beta}\,\boldsymbol{\Lambda})\left[\vec{S}^{j-1} - (\vec{S}^{j-1} + \Delta\vec{S}^j)\right] \qquad (16)\\
&= -(\mathbf{I}-\boldsymbol{\beta}\,\boldsymbol{\Lambda})\Delta\vec{S}^j \qquad (17)\\
\Delta\vec{S}^j &= -(\mathbf{I}-\boldsymbol{\beta}\,\boldsymbol{\Lambda})^{-1}\Delta^j \qquad (18)
\end{aligned}$$

$(\mathbf{I} - \boldsymbol{\beta}\,\mathbf{\Lambda})$ diagonal

$$(\mathbf{I} - \boldsymbol{\beta}\,\mathbf{\Lambda})^{-1} \simeq \frac{1}{1 - \beta_i \Lambda_{ii}} \tag{19}$$

(18) with (19) ⇒

$$\Delta \vec{S}^j = -\frac{\Delta^j}{1 - \beta_i \Lambda_{ii}} \tag{20}$$

$$= \frac{(\mathbf{I} - \boldsymbol{\beta})\vec{\vec{S}} - (\mathbf{I} - \boldsymbol{\beta}\,\mathbf{\Lambda})\vec{S}^{j-1}}{1 - \beta_i \Lambda_{ii}} \tag{21}$$

Repeat calculating new $\Delta \vec{S}^j$ until

$$\left| \frac{\Delta \vec{S}^j}{\vec{S}^{j-1}} \right| \leq 0.00001$$

in all depth points but do not more than 43 iterations (`NXTAU`, arbitrary choice).

In program start with

$$\vec{S}^0 = \vec{\vec{S}}$$

`COEFJ(l,l)`	Λ_{ii}
`diag(l)`	$1\ldots$`NXTAU` $1 - \beta_i\,\Lambda_{ii}$
`xbeta(l)`	$1\ldots$`NXTAU` $\boldsymbol{\beta}$; β_λ on fixed τ scale τ'_λ, *i.e.*, β'_λ.
`xblam(l)`	$1\ldots$`NXTAU` $(\mathbf{I} - \boldsymbol{\beta})\vec{\vec{S}}$
`xs(l)`	$1\ldots$`NXTAU` $\vec{S}^{j-1}$

$$\mathtt{delxs}' = \mathbf{\Lambda}\vec{S}^{j-1} = \sum_{i=1}^{\mathtt{NXTAU}} \Theta_{li} S_i; \quad \text{see below, page 27} \tag{22}$$

`delxs`

$$\Delta \vec{S}^j = \frac{\mathbf{\Lambda}\vec{S}^{j-1}\boldsymbol{\beta} + (\mathbf{I} - \boldsymbol{\beta})\vec{\vec{S}} - \vec{S}^{j-1}}{1 - \beta_i \Lambda_{ii}}$$

$$= \frac{\mathtt{delxs}' * \mathtt{xbeta(k)} + \mathtt{xblam(k)} - \mathtt{xs(k)}}{\mathtt{diag(k)}}$$

if $\left|\frac{\texttt{delxs}}{\texttt{xs}}\right| > 0.00001$ set

$\texttt{xs} = \texttt{xs} + \texttt{delxs}$

and repeat from eqn. (22).

Finally, do inverse transformation:

$S'_\lambda(\tau'_\lambda)_{1\ldots\texttt{NXTAU}} \rightarrow S_\lambda(\tau_\lambda)_{1\ldots\texttt{NMA}}$

The inverse transformation fails for $\tau_\lambda > 20 = \tau'_\lambda$ which means it ends for $\texttt{maxj} < \texttt{NMA}$. The asymptotic forms for Λ and Φ will be used in this region:

For $\texttt{maxj} + 1 < \texttt{NMA}$ to $\texttt{NMA}$:

$$\begin{array}{rcll} \texttt{Start:}\ S_\lambda(\tau_\lambda) & = & \bar{S}_\lambda(\tau_\lambda) = S_\lambda^{\text{old}} & \\ H_\lambda & = & \dfrac{1}{3}\dfrac{dS_\lambda}{d\tau_\lambda} & \text{Derivatives calculated with function } \texttt{DERIV} \text{ (Kurucz), see page 39.} \\ J_\lambda & = & \dfrac{dH_\lambda}{d\tau_\lambda} + S_\lambda & \\ S_\lambda^{\text{new}} & = & (1-\beta_\lambda)\bar{S}_\lambda + \beta_\lambda J_\lambda & \\ S_\lambda(\tau_\lambda) & = & S_\lambda^{\text{new}} & \end{array} \tag{23}$$

If $\sum_{\texttt{maxj}+1}^{\texttt{NMA}} \frac{|S^{\text{new}} - S^{\text{old}}|}{S^{\text{new}}} \geq 0.00001$ go to $\texttt{Start}$ with $S_\lambda = S_\lambda^{\text{new}}$.

Matrix operators

See also Mihalas, page 156, first edition.

$$\begin{aligned} J &= \Lambda S \\ &= M_1(\tau) \\ &= J(\tau) = \Lambda\, S(\tau) \end{aligned} \tag{24}$$

$$M_1(\tau) = \frac{1}{2}\int_0^\infty S(t)\, E_1|t-\tau|\, dt = M(\tau)$$

N subintervals, depth point τ_l:

$$M(\tau_l) = M_l = \frac{1}{2}\sum_{j=1}^{N}\int_{\tau_j}^{\tau_{j+1}} S(t)\, E_1|t-\tau_l|\, dt$$

Approximate $S(t)$ in interval (τ_j, τ_{j+1}) by parabola through points S_j, S_{j+1} and determine coefficients C_{jki} from least squares fit including S_{j-1} and S_{j+2} (see page 29 [Kurucz]):

$$S_j(t) = \sum_{k=1}^{3} t^{k-1}\sum_{i=1}^{N} C_{jki}\, S_i$$

$$\begin{aligned} \Rightarrow\; M_{lj} &= \frac{1}{2}\int_{\tau_j}^{\tau_{j+1}} dt\, E_1|t-\tau_l| \sum_{k=1}^{3} t^{k-1}\sum_{i=1}^{N} C_{jki} S_i \\ &= \sum_{k=1}^{3}\eta_{ljk}\sum_{i=1}^{N} C_{jki}\, S_i \end{aligned}$$

with

$$\eta_{ljk} = \frac{1}{2}\int_{\tau_j}^{\tau_{j+1}} t^{k-1}\, E_1|\tau_l - t|\, dt \quad \text{(calculation see page 30 [Kurucz])}$$

$$\begin{aligned} \Rightarrow\; \boxed{M_l} &= \sum_{j=1}^{N}\sum_{k=1}^{3}\eta_{ljk}\sum_{i=1}^{N} C_{jki}\, S_i \\ &= \sum_{j=1}^{N}\sum_{k=1}^{3}\sum_{i=1}^{N}\eta_{ljk}\, C_{jki}\, S_i \\ &= \sum_{j=1}^{N}\sum_{k=1}^{3}\sum_{i=1}^{N}\eta_{lik}\, C_{ikj}\, S_j \\ &= \boxed{\sum_{j=1}^{N}\Theta_{lj} S_j} = J(\tau_l) \end{aligned} \tag{25}$$

`C1...C43` Θ_{lj}

$\Rightarrow \; M_1(\tau) = \Theta \, S(\tau)$

$\Rightarrow \; \Lambda = \Theta \quad$ matrix operator

Since the matrix $\Lambda = \Theta$ is difficult to calculate it has been pretabulated with program `PRETAB` by Kurucz.
Lit: Kurucz, ATLAS5, *Smithsonian Astrophys. Obs. Special Report* **309**, pp. 17-18:

> ‘Since the integration matrices are rather complicated to evaluate, they have been pretabulated for a fixed τ set, where the values of τ have been chosen to give accurate integrations. The program that does this, `PRETAB`, is listed in Section 9.2. The 43 points currently used are the following:

1	0	12	0.01	23	0.63	34	3.65
2	0.000032	13	0.016	24	0.78	35	4.15
3	0.000056	14	0.025	25	0.95	36	4.9
4	0.0001	15	0.042	26	1.15	37	6.1
5	0.00018	16	0.065	27	1.35	38	7.7
6	0.00032	17	0.096	28	1.6	39	10
7	0.00056	18	0.139	29	1.85	40	12.5
8	0.001	19	0.196	30	2.15	41	15
9	0.0018	20	0.273	31	2.45	42	17.5
10	0.0032	21	0.375	32	2.75	43	20
11	0.0056	22	0.5	33	3.15		

> Note that since there are few points near the surface, integrals at monochromatic optical depths of 10^{-4} and less cannot be very reliable if the source function varies there. …

Least-squares parabolic interpolation coeffcients (Kurucz)

$$D = \tau_{j-1}^2 + \tau_{j+2}^2 - \tau_j\,\tau_{j-1} - \tau_j\,\tau_{j+1} - \tau_{j+1}\,\tau_{j-1} - \tau_{j+1}\,\tau_{j+2} + 2\,\tau_j\,\tau_{j+1}$$

$$C_{j\,l\,l-1} = \frac{\tau_j\,\tau_{j+1}}{D}$$

$$C_{j\,l\,j} = \frac{\tau_{j+1}(-\tau_{j-1}^2 - \tau_{j+2}^2 + \tau_{j+1}\,\tau_{j-1} + \tau_{j+1}\,\tau_{j+2}}{(\tau_j - \tau_{j+1})\,D}$$

$$C_{j\,l\,j+1} = \frac{\tau_j(\tau_{j-1}^2 + \tau_{j+2}^2 - \tau_j, \tau_{j-1} - \tau_j\,\tau_{j+2})}{(\tau_j - \tau_{j+1})\,D}$$

$$C_{j\,l\,j+2} = \frac{\tau_j\,\tau_{j+1}}{D}$$

$$C_{j\,2\,j-1} = \frac{-(\tau_j + \tau_{j+1})}{D} \qquad C_{j\,3\,j-1} = \frac{1}{D}$$

$$C_{j\,2\,j} = \frac{\tau_{j-1}^2 - 2\tau_{j+1}^2 + \tau_{j+2}^2}{(\tau_j - \tau_{j+1})\,D} \qquad C_{j\,3\,j} = \frac{-\tau_{j-1} + 2\tau_{j+1} - \tau_{j+2}}{(\tau_j - \tau_{j+1})\,D}$$

$$C_{j\,2\,j+1} = \frac{-\tau_{j-1}^2 + 2\tau_j^2 - \tau_{j+2}^2}{(\tau_j - \tau_{j+1})\,D} \qquad C_{j\,3\,j+1} = \frac{\tau_{j-1} - 2\tau_j + \tau_{j+2}}{(\tau_j - \tau_{j+1})\,D}$$

$$C_{j\,2\,j+2} = \frac{-(\tau_j + \tau_{j+1})}{D} \qquad C_{j\,3\,j+2} = \frac{1}{D}$$

Integration matrix

Quoted from Kurucz (here $n = 1$ for J_λ and $n = 2$ for flux):

$$\eta_{nljk} = \frac{1}{2}\,\mathrm{sign}(\tau_j - \tau_l)^{n-1} \int_{\tau_j}^{\tau_{j+1}} t^{k-1} E_n|\tau_l - t|\,dt$$

is an integral that can be evaluated analytically. To evaluate η_{nljk}, we use the indefinite integral

$$\int E_n(x)\,dx = -E_{n+1}(x)$$

and integrate by parts, obtaining

$$\begin{aligned}\eta_{nljk} \quad = \quad & \frac{1}{2}\,\mathrm{sign}(\tau_j - \tau_l)^{n-1} \\ & \big|\big(\tau_j^{k-1}\,E_{n+1}\,|\tau_l - \tau_j| - \tau_{j+1}^{k-1}\,E_{n+1}\,|\tau_l - \tau_{j+1}|\big) \\ & \quad + (k-1)\,\mathrm{sign}(\tau_j - \tau_l) \\ & \quad\quad \big(\tau_j^{k-2}\,E_{n+2}\,|\tau_l - \tau_j| - \tau_{j+1}^{k-2}\,E_{n+2}\,|\tau_l - \tau_{j+1}|\big) \\ & \quad + (k-1)(k-2) \\ & \quad\quad \big(\tau_j^{k-3}\,E_{n+3}\,|\tau_l - \tau_j| - \tau_{j+1}^{k-3}\,E_{n+3}\,|\tau_l - \tau_{j+1}|\big)\big|\,.\end{aligned} \tag{2.56}$$

We must evaluate η_{nljk} carefully when the τ's are small because of two cases of numerical cancellation. First, for $|\tau_l - \tau_j|$ small, we write out the expression for η_{nljk} explicitly using the power-series expansion for the exponential integrals and grouping terms in such a way that no loss of accuracy occurs. The power series is

$$E_n(x) = \frac{(-x)^{n-1}}{(n-1)!}\left(-\log x - \gamma + \sum_{m=1}^{n-1}\frac{1}{m}\right) - \sum_{\substack{m=0\\ m\neq n-1}}^{\infty} \frac{(-x)^m}{(m-n+1)\,m!}, \tag{2.57}$$

where $\gamma = 0.577215664901533$ is Euler's constant. Second, for τ_j/τ_l small, we use the power-series expansion around τ_l and evaluate only those terms that do not cancel analytically. The expansion is

$$\begin{aligned}E_n(x-y) = {} & E_n(x) + y\,E_{n-1}(x) + \frac{1}{2}y^2 E_{n-2}(x) + \cdots + \\ & + \frac{y^{n-2}}{(n-2)!}E_2(x) + \frac{y^{n-1}}{(n-1)!}E_1(x) + \\ & + e^{-x}\sum_{m=0}^{\infty}\frac{y^{m+n}}{(m+n)!}\left[1 + \frac{m}{x} + \frac{m\,(m-1)}{x^2} + \cdots + \frac{m!}{x^{m-1}} + \frac{m!}{x^m}\right]\end{aligned}$$

For more information see program `PRETAB` or Mihalas, page 156, first edition.

Flux integration (`IFSURF = 1`)

$$\begin{aligned} H &= \Phi S \\ &= M_2(\tau) \\ &= H(\tau) = \Phi\, S(\tau) \end{aligned}$$

$$M_2(\tau) = \frac{1}{2}\int_0^\infty \mathrm{sign}(t-\tau)\, S(t)\, E_2|t-\tau|\, dt$$

Similar formulation as for $M_1(\tau)$ (page 27) yields:

$$M_2 = \sum_{j=1}^{N} \Theta^*_{lj} S_j$$

Surface flux: $\tau = 0 \mathrel{\hat{=}} l = 1$

$$\Rightarrow\ H_{\mathrm{surface}} = \sum_{j=1}^{N} \Theta^*_{1j} S_j$$

`COEFH(1,1)`	Θ^*_{1j}
FLUX	$4 \cdot \sum_{\mathtt{i=1}}^{\mathtt{NXTAU}} \Theta^*_{1j} S'_{\lambda,i}$

Intensity integration (`IFSURF = 2`)

$$\begin{aligned} I(\mu) &= \int_0^\infty S(\tau)\, e^{-\tau/\mu}\, \frac{d\tau}{\mu} \\ &= \sum_{i=1}^{N-1} \int_{\tau_i}^{\tau_{i+1}} S(\tau)\, e^{-\tau/\mu}\, \frac{d\tau}{\mu} + \int_{\tau_N}^{\infty} S(\tau)\, e^{-\tau/\mu}\, \frac{d\tau}{\mu} \qquad \text{with } N = \text{number of depth points, } \texttt{NMA} \end{aligned}$$

Approximate $S(\tau)$ in interval (τ_i, τ_{i+1}) with weighted backward and forward parabolas (see `PARCOE`, page 37):

$$S_i = a_i + b_i\tau + c_i\tau^2$$

$$\begin{aligned} I(\mu) &= \sum_{i=1}^{N-1} \int_{\tau_i}^{\tau_{i+1}} e^{-\tau/\mu}\,(a_i + b_i\tau + c_i\tau^2)\, \frac{d\tau}{\mu} + \int_{\tau_N}^{\infty} e^{-\tau/\mu}\,(a_i + b_i\tau + c_i\tau^2)\, \frac{d\tau}{\mu} \\ &= \sum_{i=1}^{N-1} \left\{ -e^{-\tau/\mu} \left[a_i + b_i(\tau+\mu) + c_i \left((\tau+\mu)^2 + \mu^2 \right) \right] \right\}_{\tau_i}^{\tau_{i+1}} + \\ &\quad \left\{ -e^{-\tau/\mu} \left[a_i + b_i(\tau+\mu) + c_i \left((\tau+\mu)^2 + \mu^2 \right) \right] \right\}_{\tau_N}^{\infty} \\ &= \sum_{i=1}^{N-1} \left\{ e^{-\tau_i/\mu} \left[a_i + b_i(\tau_i+\mu) + c_i \left((\tau_i+\mu)^2 + \mu^2 \right) \right] - \right. \\ &\quad \left. e^{-\tau_{i+1}/\mu} \left[a_i + b_i(\tau_{i+1}+\mu) + c_i \left((\tau_{i+1}+\mu)^2 + \mu^2 \right) \right] \right\} + \\ &\quad e^{-\tau_N/\mu} \left[a_N + b_N(\tau_N+\mu) + c_N \left((\tau_N+\mu)^2 + \mu^2 \right) \right] - 0 \end{aligned}$$

$$\begin{aligned} \textsf{FLUX} \qquad &= \sum_{i=1}^{N-1} \left\{ e^{-\tau_i/\mu}[S_i + \underbrace{(b_i + 2c_i\tau_i)}_{\texttt{b2ct(n)}}\mu + \underbrace{2c_i}_{\texttt{ctwo(n)}}\ \mu^2] - \right. \\ &\quad \left. e^{-\tau_{i+1}/\mu}[S_{i+1} + \underbrace{(b_i + 2c_i\tau_{i+1})}_{\texttt{b2ct1(n)}}\mu + 2c_i\mu^2] \right\} + \\ &\quad e^{-\tau_N/\mu}[S_N + (b_N + 2c_N\tau_N)\mu + 2c_N\mu^2] \end{aligned}$$

Summation stops prematurely if $\frac{\tau_{i+1}}{\mu} > 70$.

For small optical depths, *i.e.* $\Delta = \frac{\tau_{i+1}}{\mu} - \frac{\tau_i}{\mu} \le 0.03$:

$$\begin{aligned}\int_{\tau_i}^{\tau_{i+1}} S(\tau)\, e^{-\tau/\mu} \frac{d\tau}{\mu} \quad = \quad & e^{-\tau_{i+1}/\mu} \left\{ S_i\Delta + [S_i + (b_i + 2\, c_i\, \tau_i)\mu] \frac{\Delta^2}{2} + \right. \\ & \left. [S_i + (b_i + 2\, c_i\, \tau_i)\mu + 2\, c_i\, \mu^2] \sum_{n=3}^{m} \frac{\Delta^n}{n!} \right\} \end{aligned} \tag{26}$$

with

$$m = \begin{cases} 4 & \text{if } \Delta \le 0.001 \\ 9 & \text{if } 0.001 < \Delta \le 0.03 \end{cases}$$

`d` Δ

`ddddd` help variable used in recursive expansion of eqn. (26):

$$\begin{cases} 1 & \text{if } \Delta \le 0.001 \\ \frac{\sum_{n=4}^{9} \frac{\Delta^n}{n!}}{\Delta^4/24} & \text{if } 0.001 < \Delta \le 0.03 \end{cases}$$

Calculation of contributions from individual layers to emergent flux/intensity

Done if `KCONTR` ≥ 0 and either continuum calculation or at center of desired line

`ch` $\mu = \begin{cases} \cos\vartheta & \text{if } I_\lambda(\mu) \\ \frac{2}{3} & \text{if } F_\lambda \end{cases}$

`a(n)` 1... `NMA`

$$a_i = \begin{cases} \frac{1}{2}(S_i\, e^{-\tau_i/\mu} + S_{i+1}\, e^{-\tau_{i-1}/\mu})(\frac{\tau_i}{\mu} - \frac{\tau_{i-1}}{\mu}) & \text{if } \frac{\tau_i}{\mu} \text{ or } \frac{\tau_{i-1}}{\mu} \le 87 \\ 0 & \text{else} \end{cases}$$

`f` $\sum_{i=1}^{\texttt{NMA}} a_i$

For continuum calculation (`CONTI = .TRUE.`):

`TAULAC(n)` 1... `NMA` $\tau_{\lambda,c}$; optical depth of continuum (= `TAULAM(n)`)

`CONTRC(n)` 1... `NMA` = 100*`a(n)/f`; contribution of layer `n` to emergent continuum flux/intensity

Line calculations (`CONTI = .FALSE.`):

`CONTRL(n)` 1... `NMA` = 100*`a(n)/f`; contribution of layer `n` to emergent line flux/intensity

SUMETA(N,DL,SUMET2) (Function)

Calculates $\sum \eta$.

N	`INTEGER`	input	Depth point
DL	`REAL`	input	Distance [mÅ] from line/synthesis center
SUMET2	`REAL`	output	see below

For continuum calculation (`CONTI = .TRUE.`):

SUMETA $$= \texttt{ETAC}(\mathsf{N}) = \eta_c = \frac{\kappa_{\bar\lambda,c} + \sigma_{\bar\lambda,c}}{\kappa_{\rm ref} + \sigma_{\rm ref}}$$

with η_c calculated beforehand in Function `CONT`.

SUMET2 = SUMETA

In line calculations (`CONTI = .FALSE.`):

`dlam` $\Delta\lambda$ Distance from line center of blend line k:

$$= |\mathsf{DL} - \texttt{DLK(k)}|/1{,}000 \text{ [Å]}$$

$$\begin{aligned}\kappa_l &= C\,b_l\left(1 - \frac{b_u}{b_l}e^{c'}\right)\\ &= C\left(b_l - b_u e^{c'}\right)\end{aligned}$$

$\Rightarrow$

$$\kappa_l^* = C\left(1 - e^{c'}\right)$$

$\Rightarrow$

$$\frac{\kappa_l}{\kappa_l^*} = \frac{b_l - b_u\, e^{c'}}{1 - e^{c'}}$$

$\Rightarrow$

SUMETA $$\begin{aligned} &= \eta_c + \sum_{k=1}^{\texttt{KMA}} \eta^*_{0,k}\,\Phi\,\frac{b_{l_k} - b_{u_k} e^{-h\nu_k/kT}}{1 - e^{-h\nu_k/kT}}\\ &= \frac{\kappa_{\bar\lambda,c} + \sigma_{\bar\lambda,c} + \sum_{i=1}^{\texttt{KMA}} \kappa_{\lambda,l_i}}{\kappa_{\rm ref} + \sigma_{\rm ref}} \quad \text{for depth point } \mathsf{N}\end{aligned}$$

with

$$\Phi = \begin{cases} H(\alpha_k, \frac{\Delta\lambda}{\lambda_k \underbrace{v_D}_{\texttt{VDOP}(\mathsf{N},\texttt{KJ(k)})}}) & \text{Metal lines} \\ \texttt{PROFHY}(\alpha_k, \Delta\lambda, \mathsf{N}, \texttt{k}) & \text{Hydrogen lines} \end{cases}$$

SUMET2 $= \eta_c + \sum_{k=1}^{\texttt{KMA}} \eta^*_{0,k}\, \Phi\, b_{u_k}$

PROFHY(R,D,N,K) (Function)

Profile for Hydrogen lines.

R	`REAL`	input	α; damping constant
D	`REAL`	input	$\Delta\lambda$; distance [mÅ] from line/synthesis center
N	`INTEGER`	input	Depth point
K	`INTEGER`	input	Line number in blend

`dld` $\Delta\lambda_D = v_D\, \lambda_k (\texttt{VDOP(N,1)*LAMK(K)})$

PROFHY
$$= \begin{cases} \left\{ \begin{array}{l} \frac{1}{17.7245} \left(\texttt{ASH}(\mathsf{R}, \Delta\lambda, \mathsf{N}, \mathsf{K}) + \right. \\ \quad \left[\sum_{i=1}^{25} \texttt{ASH}(\mathsf{R}, \Delta\lambda + i \cdot 0.1\Delta\lambda_D, \mathsf{N}, \mathsf{K}) + \right. \\ \quad \left. \left. \texttt{ASH}(\mathsf{R}, \Delta\lambda - i \cdot 0.1\Delta\lambda_D, \mathsf{N}, \mathsf{K})\, e^{-\frac{i \cdot 0.1\Delta\lambda_D^2}{\Delta\lambda_D^2}} \right] \right) \end{array} \right\} & \Delta\lambda < 5\,\Delta\lambda_D \\ \texttt{ASH}(\mathsf{R}, \mathsf{D}, \mathsf{N}, \mathsf{K}) & \Delta\lambda \geq 5\,\Delta\lambda_D \end{cases}$$

ASH(R,D,N,K) (Function)

Asymptotic form of Hydrogen line profile for line wings

R	`REAL`	input	α; damping constant
D	`REAL`	input	$\Delta\lambda$; distance [mÅ] from line/synthesis center
N	`INTEGER`	input	Depth point
K	`INTEGER`	input	Line number in blend

ASH $= \dfrac{1 + R^* \sqrt{\Delta}\lambda + \frac{C_e}{C_{\mathrm{H}}} \frac{p_{\mathrm{H}}}{p_e}}{\Delta\lambda^{5/2}}$ following Griem; see BHT

$$\left(\frac{1}{R^*\sqrt{\Delta\lambda}}\right)^4 = 1 + \left(\frac{1}{R\sqrt{\Delta\lambda}}\right)^4$$

INTEG(X,F,FINT,N,START) (Subroutine)

Integration routine (Kurucz).

X(1)	`REAL`	input	independent variable
F(1)	`REAL`	input	integrand
FINT(1)	`REAL`	output	integral
N	`INTEGER`	input	number of points
START	`REAL`	input	start value; added to integral

‘Simple integrals like (2.44) are performed by fitting parabolas to the integrand for each depth interval in the atmosphere, as follows:

$$\begin{aligned}
\int_0^{x_N} f(x)\,dx &= \sum \int_{x_j}^{x_{j+1}} f(x)\,dx \\
&= \sum \int_{x_j}^{x_{j+1}} (a_j + b_j x + c_j x^2)\,dx \\
&= \sum \left[a_j(x_{j+1} - x_j) + \frac{b_j(x_{j+1}^2 - x_j^2)}{2} + \frac{c_j(x_{j+1}^3 - x_j^3)}{3} \right] \\
&= \sum \left[a_j + \frac{b_j(x_{j+1} + x_j)}{2} + \frac{c_j(x_{j+1}^2 + x_{j+1}x_j + x_j^2)}{3} \right] (x_{j+1} - x_j)
\end{aligned}$$

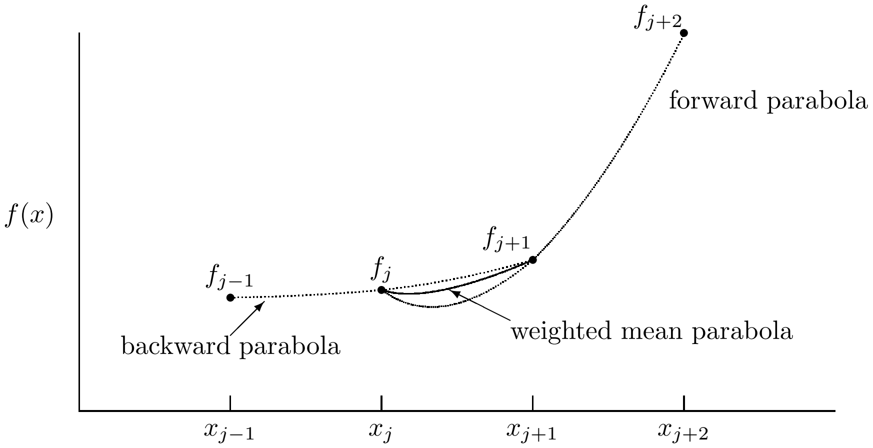
Calls `PARCOE` to determine the interpolation coefficients a_j, b_j, and c_j.

PARCOE(F,X,A,B,C,N) Subroutine

Computes parabola coefficients.

F(1)	`REAL`	input	function to fit
X(1)	`REAL`	input	independent variable
A(1)	`REAL`	output	fit coefficient
B(1)	`REAL`	output	fit coefficient
C(1)	`REAL`	output	fit coefficient
N	`INTEGER`	input	number of points

‘In ATLAS, the interpolation coefficients a_j, b_j, and c_j are determined by weighting forward and backward parabolas inversely by their second derivatives, as shown in the following diagram:



This weighting avoids large overshoots, which are a problem with simple parabolic interpolation. The unweighted parabolas are given by the expressions

$$\begin{aligned}
c_j &= \frac{f_{j+1}}{(x_{j+1}-x_j)(x_{j+1}-x_{j-1})} - \frac{f_j}{(x_j-x_{j-1})(x_{j+1}-x_j)} + \\
&\quad \frac{f_{j-1}}{(x_j-x_{j-1})(x_{j+1}-x_{j-1})}, \\
b_j &= \frac{f_j-f_{j-1}}{x_j-x_{j-1}} - (x_j+x_{j-1})\,c_j,
\end{aligned}$$

and

$$a_j = f_{j-1} - x_{j-1}\frac{f_j-f_{j-1}}{x_j-x_{j-1}} + x_j\,x_{j-1}\,c_j.$$

The weight is

$$w_j = \frac{|c_{j+1}|}{|c_{j+1}|+|c_j|},$$

so the weighted mean parabola is

$$\begin{aligned}
\bar{a}_j &= w_j a_j + (1-w_j)\,a_{j+1} \\
\bar{b}_j &= w_j b_j + (1-w_j)\,b_{j+1} \\
\bar{c}_j &= w_j c_j + (1-w_j)\,c_{j+1}
\end{aligned}$$

MAP1(XOLD,FOLD,NOLD,XNEW,FNEW) Function

Parabolic interpolation for entire vector. Same formalism as `PARCOE`.

XOLD(1)	`REAL`	input	independent variable
FOLD(1)	`REAL`	input	function to interpolate
NOLD	`INTEGER`	input	number of points
XNEW(1)	`REAL`	input	independent variable to interpolate
FNEW(1)	`REAL`	output	interpolated function

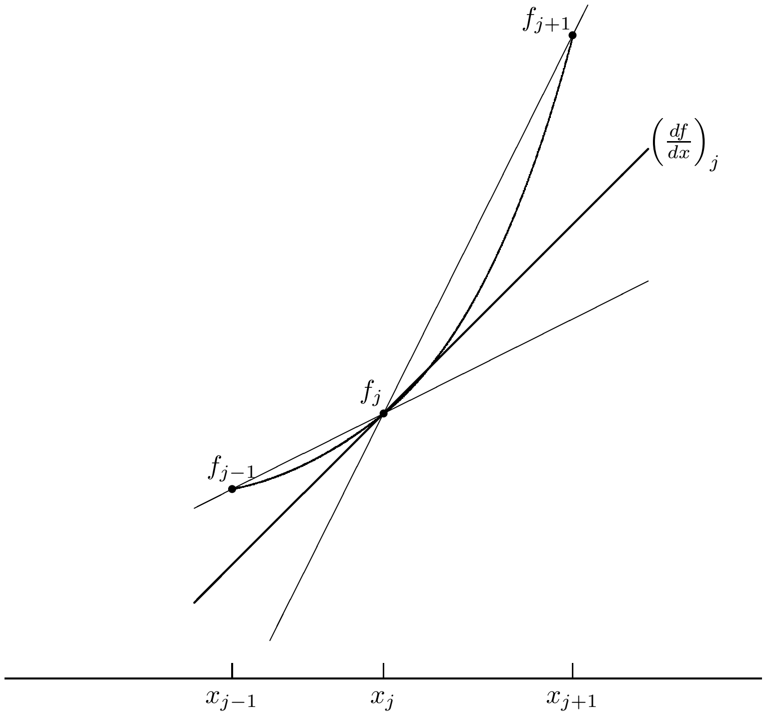
MAP1	number of points needed for inverse interpolation

DERIV(X,F,DFDX,N) Subroutine

Computes derivatives. Used only for calculation of S_λ in large optical depths.

X(1)	`REAL`	input	independent variable
F(1)	`REAL`	input	function values
DFDX(1)	`REAL`	output	derivatives at `X`(i)
N	`INTEGER`	input	number of points

Code taken from Kurucz:
'Derivatives are taken by bisecting the angles formed by the extension of the line segments as shown in the following diagram:



The result is

$$\left(\frac{df}{dx}\right)_j = \frac{S\,(T_{j+1}+T_j)}{1-T_{j+1}T_j},$$

where

$$T_j = \frac{D_j}{1+\sqrt{1+D_j^2}},$$

with

$$D_j = \frac{1}{S}\frac{f_j-f_{j-1}}{x_j-x_{j-1}},$$

and where

$$S = \frac{\max\left(|f_{j-1}|,|f_j|,|f_{j+1}|\right)}{|x_j|}.$$

The scaling factor S converts f and x to the same magnitude.

Index

Contents