

Interpolation of Stellar Model Grids and Application to the NEMO Grid

J. Nendwich¹, U. Heiter², F. Kupka³, N. Nesvacil^{4,1}, W.W. Weiss¹

¹ Institut für Astronomie, Türkenschanzstrasse 17, 1180 Vienna, Austria

² Department of Astronomy and Space Physics, Uppsala University, Box 515, SE-75120 Uppsala, Sweden

³ Max-Planck-Institute for Astrophysics, Karl-Schwarzschild Str. 1, 85741 Garching, Germany

⁴ European Southern Observatory (ESO), Alonso de Cordova 3107, Vitacura, Santiago, Chile

Abstract

NEMO.2003¹ is a DVD with 91,520 stellar model atmospheres representing a 5D grid of modified ATLAS9 atmospheres; the purpose of the modifications was to include different treatments of convection and higher vertical resolution. In addition, for every model fluxes are provided and color indices for 14 different photometric systems. Because the model grid contained gaps due to non-converging models, we developed and applied a set of 4D interpolation routines to complete the grid.

All the data, which will continuously be updated, can be found on the NEMO homepage <http://ams.astro.univie.ac.at/nemo/> and are available via DVD.

Introduction

As described in Heiter et al. (2002, hereafter H02) several new sets of grids of model stellar atmospheres were computed with modified versions of the ATLAS9 code. The individual sets differed from each other, and from previous ones, essentially in the physics used for the treatment of the convective energy transport, in the higher vertical resolution of the atmospheres, and in a finer grid in the (T_{eff} , $\log g$) plane. The improvements related to resolution enabled the

¹Vienna New Model Grid of Stellar Atmospheres

computation of derivatives of color indices, as well as limb darkening coefficients and derivatives (Garrido et al. 2001, 2002; Barban et al. 2003), accurate enough for pulsation mode identification, and of smooth pre-main sequence evolutionary tracks (Montalbán et al. 2004). A description of the modifications of ATLAS9 with respect to their treatment(s) of convection was also provided by H02. While the grids published by Kurucz (1993b (K93b), 1998) and Castelli et al. (1997) favor a rather efficient convection in stellar atmospheres due to their calibration based on solar central intensities, the new sets of grids were computed for convection models which predict a much lower efficiency, at least within the photosphere. The motivation behind this has been to allow a choice between the two assumptions, as none of the presently implemented local, homogeneous models are able to reproduce all observations, whether for the sun or for groups of stars spread over wider regions of the HR diagram (see, e.g., Smalley et al. 2002). In addition, model atmospheres with lower convective efficiency than those in Kurucz (K93b, 1998) and Castelli et al. (1997) distributions allow recovering a larger number of observations for a given choice of convection model and its parameters (H02, D’Antona et al. 2002, D’Antona & Montalbán 2003, Montalbán et al. 2004, Stassun et al. 2004). Schuler et al. (2004) used different sets of ATLAS9 grids, including the 72 layer models presented here, to derive oxygen abundances in open cluster dwarfs. Their work shows that the differences in abundances due to different models are small compared to the typical error in the relative oxygen triplet abundances.

Parameter range

As discussed in H02, the computations for all values of the grid parameters have been finished and we can present the whole 5D grid (Table 1).

The convection models CM (Canuto & Mazzitelli, 1991) and CGM (Canuto, Goldman & Mazzitelli, 1996) and their implementation into ATLAS9 are summarized in H02. As is shown there as well, the convective flux can be neglected for temperatures higher than 8600 K. Thus, convection has been turned off for such models and they are computed only once for models with 288 layers and once for models with 72 layers and are included in the CGM subgrids. This explains the two different temperature ranges in Table 1 and the total number of models of 91,520. We also repeat here that the uppermost layers of our models are located at $\log \tau_{\text{Ross}} = -6.875$ and the difference of consecutive layers in $\log \tau_{\text{Ross}}$ is 0.125 and 0.03125 for models with 72 and 288 layers, respectively.² For all models, the opacity distribution functions calculated by Kurucz (1993a) were used.

²The lowermost layer is located at $\log \tau_{\text{Ross}} = -6.875 + (72 - 1)/8 = +2.0$ and $-6.875 + (288 - 1)/32 = +2.09375$ for models with 72 and 288 layers, respectively.

Parameter	From	To	Increment (List of Values)	Unit	#
T_{eff}	4,000	10,000 8,600 ^a	200	K	31 24 ^a
$\log g$	2.0	5.0	0.2	{cgs} ^b	16
$[M/H]$	-2.0	+1.0	(-2.0, -1.5, ± 1.0 , ± 0.5 , ± 0.3 , ± 0.2 , ± 0.1 , 0.0)	dex	13
v_{micro}	0	4	(0, 1, 2, 4)	km s ⁻¹	4
Convec. /resol.	CGM($\alpha^*=0.09$) / 72, MLT($\alpha=0.5$) / 72, CGM($\alpha^*=0.09$) / 288, CM / 288				4
Total number of model atmospheres					91,520

Table 1: Parameter range of the grid

'Convec.' = convection model, 'resol.' = vertical resolution (number of layers)

^a for the non CGM models, see also text

^b log cm s⁻²

Colors

Colors and color indices in 14 photometric systems have been calculated with the program "colors", which is based on the programs of Kurucz CD-ROM 13 (<http://kurucz.harvard.edu/programs/COLORS/>) revised and rewritten in Fortran90 by JN. This program is also available on the NEMO website as well as on DVD. The color indices for specific photometric systems have been normalized using model fluxes and measured indices for standard stars as follows: β Leo ($T_{\text{eff}}=8\,850$, $\log g=4.16$, $[M/H]=0.0$, $v_{\text{micro}}=2$) for Walraven VBLUW, HD 83373 (9 250, 4.00, +0.0, 2) for the *hk194* index of Strömgren uvbyCa, a B star (30 000, 4.00, +0.0, 2) for Vilnius, and Vega (9 550, 3.95, -0.5, 2) for Cousins (Cape) VRI, Δ a, Geneva, JHKcit, (V)RIJKL, thirteen, and UBV. DDO and HST color indices have not been normalized. The standard star colors and resulting zero points, as well as the references for the filter transmission curves that have been used, are given in Tables 2 - 4. Colors in the DDO system were calculated with data from Lamla (1982, L82). The HST filter curves were taken from ftp://ftp.stsci.edu/cdbs/cdbs8/synphot_tables/. The filter transmission curves for all systems are shown in Figs. 1 - 5. The symbols represent the values given by K93b or found in the literature, the lines show the results of the (curvature weighted) parabolic interpolations from the Ku-

ruz programs, which are sometimes convoluted with other response functions (atmosphere, optics, receiver).

H-Beta (Vega)							
Kurucz 1993b, 1998 (.../programs/COLORS/)							
		H_{β}					
		2.903					
		2.695					
Using additional response functions 'OneP21', 'Air', and 'Reflect'. Output "468nm.dat" are not correct H_{β} values. (see text below)							
Delta a (Vega)							
Kupka et al. (2003), Crawford & Barnes (1970), Maitzen & Vogt (1983)							
		a	$b-y$				
		-0.009	-0.005				
		0.600	0.500				
$a = g_2 - (g_1 + y)/2$, $\Delta a = a - a_0[(b - y); (B - V); (g_1 - y)]$ Using additional response functions 'OneP21' and 'Reflect'.							
DDO							
L82 p. 55							
		35-38	38-41	41-42	42-45	45-48	
		-	-	-	-	-	
		0.000	0.000	0.000	0.000	0.000	
No zero point star given. Using additional response functions 'OneP21', 'Air', and 'Reflect'.							
uvbyCa (Vega, HD83373)							
Crawford & Barnes (1970), Hauck & Mermilliod (1980), Strömgren (1966)							
	$u-b$	u_0-b	$b-y$	m_1	c_1	c_{10}	hk
	1.411	1.411	0.004	0.157	1.089	1.089	<i>0.194</i>
	0.802	0.780	0.500	-0.070	-0.058	-0.080	<i>-0.133</i>
Model for HD83373 ($hk=0.194$): $(T_{\text{eff}}, \log g, [M/H], v_{\text{micro}}) = (9250, 4.0, +0.0, 2)$ Using additional response functions 'OneP21', 'Air', and 'Reflect'.							

Table 2: Names, zero point stars, references, standard and correction values of indices of the photometric systems. The last line contains remarks concerning the Kurucz programs among others.

It turned out that the standard output of ATLAS9 for the flux files is meshed too coarse in λ to give accurate results for H_{β} . So the color output for this system is called "468nm.dat" instead of "beta.dat".

Walraven (β Leo)							
L82 p. 76, Lub & Pel (1977)							
	V-B	B-U	U-W	B-L			
	0.034	0.436	0.108	0.198			
	-0.561	1.232	0.106	0.460			
$\lambda = 372 \text{ nm}: l_1(\text{K93b}) = 0.468, l_1(\text{L82}) = 0.488$ $\lambda = 462 \text{ nm}: b_1(\text{K93b}) = 0.282, b_1(\text{L82}) = 0.287$ Model for β Leo: $(T_{\text{eff}}, \log g, [M/H], v_{\text{micro}}) = (8850, 4.16, +0.0, 2)$							
Geneva (Vega)							
L82 p. 61							
	U-B	V-B	B ₁ -B	B ₂ -B	V ₁ -B	G-B	
	1.505	0.959	0.900	1.510	1.662	2.168	
	0.759	0.293	0.942	1.443	1.045	1.318	
K93b has some additional response function values.							
Geneva3 (Vega)							
Cramer & Maeder (1979), Künzli et al. (1997), L82 p. 61, North & Kupka (1997), Rufener & Nicolet (1988)							
	U-B	V-B	B ₁ -B	B ₂ -B	V ₁ -B	G-B	
	1.505	0.959	0.900	1.510	1.662	2.168	
	0.004	0.017	0.017	0.005	0.018	0.027	
Additional (derived) indices are calculated: U-B ₂ , B ₁ -B ₂ , B ₂ -V ₁ , V ₁ -G, d, Δ , g, m ₂ , X, Y, Z, pT, pG							
UBV (USA Vilnius Buser) (Vega)							
Arp (1961), Ažusienis & Straizys (1969), Buser (1978), Buser & Kurucz (1978), L82 p. 48, Matthews & Sandage (1963)							
	U-B			B-V			
	-0.005			-0.003			
	u ₁ -b ₁	u _{1.3} -b _{1.3}	U ₃ -B ₂	b ₁ -v ₁	b ₀ -v ₀	B ₃ -v ₀	
	-0.475	-0.468	-0.408	0.619	0.601	0.601	
Vilnius (7 color) (Vega)							
L82 p. 66							
	U-P	P-X	X-Y	Y-Z	Z-V	V-S	T-S
	0.630	0.830	0.290	0.100	0.050	0.140	0.040
	0.304	0.302	0.543	0.414	0.218	0.826	0.267
Normalization in L82: all indices are 0.000 for $(T_{\text{eff}}, \log g, [M/H], v_{\text{micro}}) = (30000, 4.00, +0.0, 2)$							

Table 3: see Table 2

HST ('External')						
ftp://ftp.stsci.edu/cdbs/cdbs8/synphot_tables/						
	555	606	814			
	-	-	-			
	0.000	0.000	0.000			
No zero point star. Response functions of cameras 'pcf', 'wf2f', 'wf3f', 'wf4f', and 'wfpc2f' used (same color index for one wavelength region).						
Thirteen (Vega)						
L82 p. 68, Johnson & Mitchell (1975)						
33-52	35-52	37-52	40-52	45-52	52-58	52-58'
0.052	0.035	0.043	0.013	0.001	-0.008	0.001
-0.202	-0.259	0.090	0.584	0.377	0.356	0.390
	52-63	52-72	52-80	52-86	52-99	52-110
	-0.011	0.008	0.010	-0.004	0.011	0.000
	0.664	1.103	1.451	1.683	2.019	2.441
58'(K93b) = 58(L82), 58(K93b): no reference found; (52-58) _{Vega} = -0.008 (K93b) vs. 0.001 (L82)						
Cousins (Cape) (Vega)						
Cousins (1976), L82 p. 47, 79, 82						
	V-R	V-I				
	-0.009	-0.005				
	0.593	1.228				
V(K93b)=v ₀ (UBV _{Vilnius}), R(K93b)=r _c (L82), I(K93b)=i _c (L82) (= 0.84 vs. 0.86 at λ = 840 nm), additional values for RCA31034 ('Cathode75'): (λ, S(λ)) = (880, 0.03), (890, 0.01)						
JHKcit (Vega)						
Kurucz 1993b, 1998 (.../programs/COLORS/)						
	V-J	V-H	V-K			
	0.030	0.030	0.030			
	19.181	16.444	14.226			
Using additional response functions 'atmos', 'detec', and 'block'.						
(V)RIJKL (Vega)						
L82 p. 71						
	V-R	V-I	V-J	V-K	V-L	
	0.000	0.000	0.000	0.000	0.000	
	0.717	1.488	2.651	4.850	6.699	
K93b uses L ^{II} of L82.						

Table 4: see Table 2

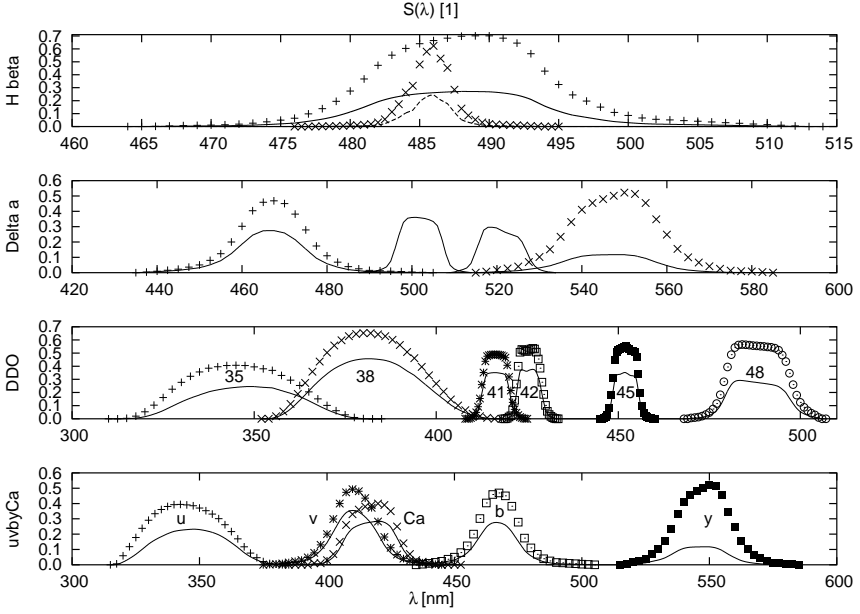


Figure 1: Response functions of four out of the 14 photometric systems (1/4). The symbols represent the filter values, the lines show the results of the convolution of their curvature weighted parabolic interpolations with other response functions (atmosphere: ‘Air’ (not in Δa), optics: ‘Reflect’, receiver: ‘OneP21’).

Missing models - grid gaps

The grid computations were performed with the perl package SMGT (Stellar Model Grid Tool), described in Schmidt (1999, S99). Due to bad initial conditions or because the parameter region is physically and/or mathematically ‘difficult’, 1,110 out of the 91,520 models did not converge with respect to the SMGT convergence criteria (S99; H02, their Table 3). Their distribution in the grid is shown in Figures 6 - 10 (p. 53 - 57). For the single parameters we find difficult regions for low T_{eff} values (Fig. 6), high (and low) $\log g$ values (Fig. 7), low $[M/H]$ values (Fig. 8), and for CGM_{288} whereas MLT_{72} is ‘easiest’ among the different convection models (Fig. 10); v_{micro} seems not to be as relevant for creating problems to achieve convergence (Fig. 9). A more detailed insight—for combinations of two parameters—can be obtained from the 3D plots of Figures 11 - 16 (p. 58 - 63). More figures are available on the NEMO website.

The total numbers of models fulfilling the primary and secondary SMGT

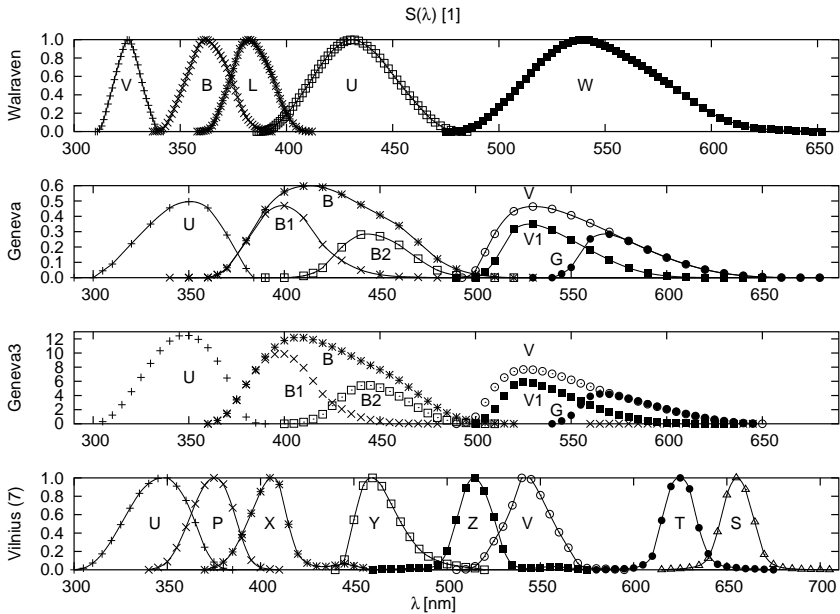


Figure 2: Response functions (filter values and their curvature weighted parabolic interpolations) of four out of the 14 photometric systems (2/4).

convergence criteria are given in Table 5 (p. 65). In this sense the models are also called *fully* and *partially converged*. Note that before interpolation (next section) about 3/4 of the (90,410) models are fully converged and 1/4 partially, whereas with the (527) interpolated and consecutively converged files it is the other way round: 1/4 fulfil the primary convergence criteria and 3/4 only the secondary ones.

Interpolation

A Fortran90 program was written to close the gaps in the grid through interpolation (Ip1). Since an extended version of the program should allow the interpolation of models (and their fluxes and colors) between the grid nodes (Ip2), the following requirements are desired:

1. Multidimensionality: As there are four numerical parameters— T_{eff} , $\log g$, $[M/H]$, and v_{micro} (the convection model is non-numerical)—a 4D interpolation is necessary.

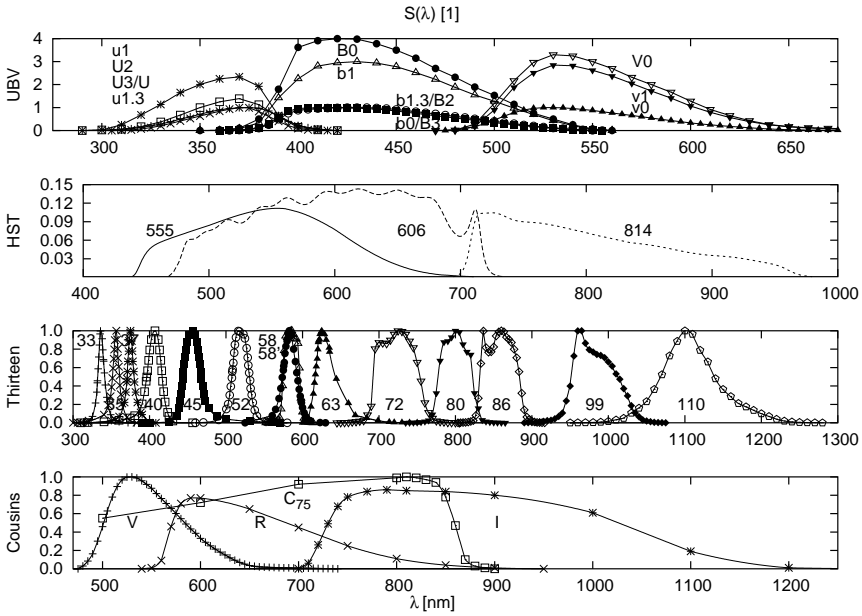


Figure 3: Response functions (filter values and their curvature weighted parabolic interpolations) of four out of the 14 photometric systems (3/4).

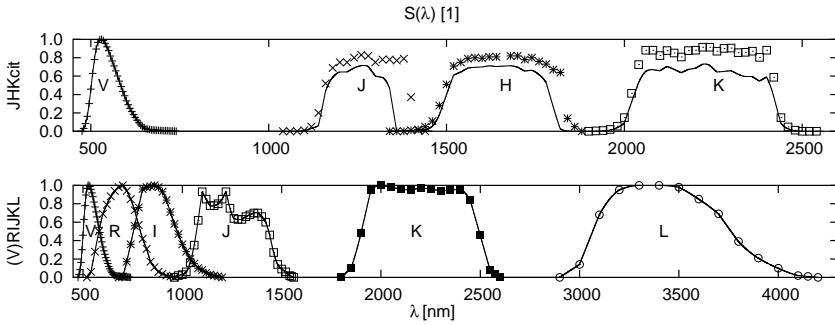


Figure 4: Response functions (filter values and their curvature weighted parabolic interpolations) of two out of the 14 photometric systems (4/4). JHKcit is convoluted with the additional response functions 'atmos', 'detec', and 'block'.

This is realized by means of four 1D interpolations, one for each numerical parameter, because in the surroundings of a gap there are frequently further gaps. Hence, in general it is difficult to get a suitable 4D subgrid

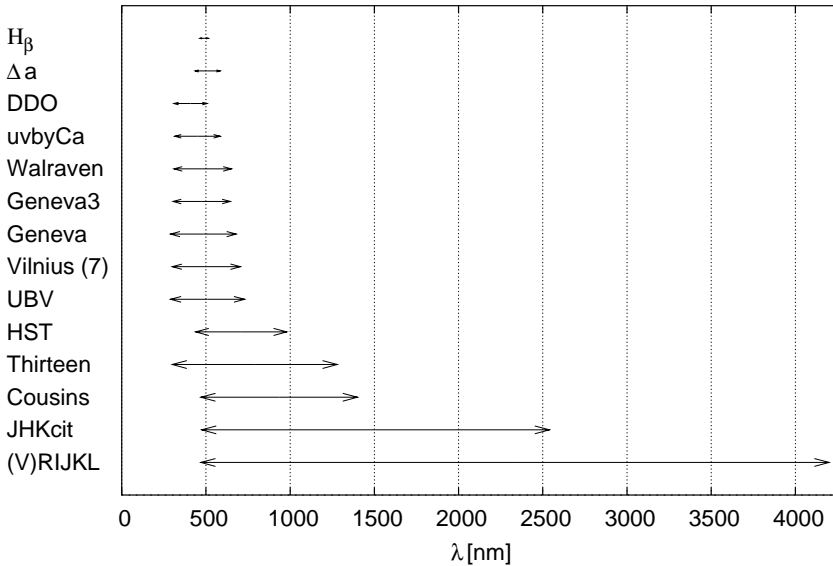


Figure 5: Wavelength range of the photometric systems

neighborhood.

2. Precision: mmag in colors.

Tests (comparisons of interpolated values with SMGT calculated ones) have shown that linear interpolation results in errors of some ten millimag, so at least parabolic interpolation is necessary to shrink down the deviation to the desired accuracy of one millimag.

3. Local procedure: A global procedure would involve 25,000 grid nodes and thus make the program complex and slow, and it requires a complete grid without gaps. The physics does not demand a global method either.

This excludes splines which would satisfy the precision and differentiability requirements.

4. Differentiability: First derivatives of color indices with respect to T_{eff} and $\log g$ are used for pulsation mode identification (Garrido et al. 1990); they should be computable and *continuous* at the grid nodes.

Unlike curvature weighted parabolas as are used in the Fortran77 color programs by Kurucz (K93b), *distance weighted parabolas* can provide

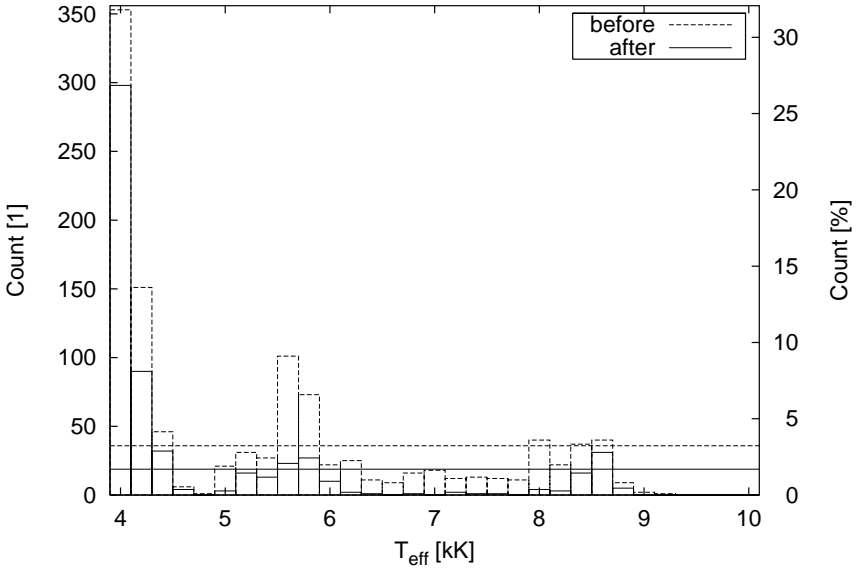


Figure 6: Number of missing models versus T_{eff} (added up for the other four parameters) before and after the interpolation. The two horizontal lines indicate the average values per T_{eff} step of 35.8 ($=1,110/31$) and 18.8 ($=583/31$).

continuous derivatives at the grid nodes and fulfill also the other requirements.

Distance Weighted Parabolas for Interpolation

Fig. 17 (p. 64) illustrates the interpolation procedure in 1D. For an interpolation in the interval $x_2 \leq x \leq x_3$ we need two left neighboring grid nodes x_1 and x_2 and two right ones x_3 and x_4 . An interpolating function $h(x)$ is constructed for the unknown function $f(x)$ using the four given data points $(x_i, f_i), i = 1, \dots, 4$, where

$$f_i = f(x_i) \tag{1}$$

We define a (quadratic) ‘left’ parabola $p_L(x)$ by means of $x_{1,2,3}$ and a ‘right’ one $p_R(x)$ by means of $x_{2,3,4}$, and for comparison a cubic polynomial $p_C(x)$ by means of all four:

$$\begin{aligned} p_L(x_1) &= f_1 = p_C(x_1) \\ p_L(x_2) &= f_2 = p_C(x_2) = p_R(x_2) \end{aligned} \tag{2}$$

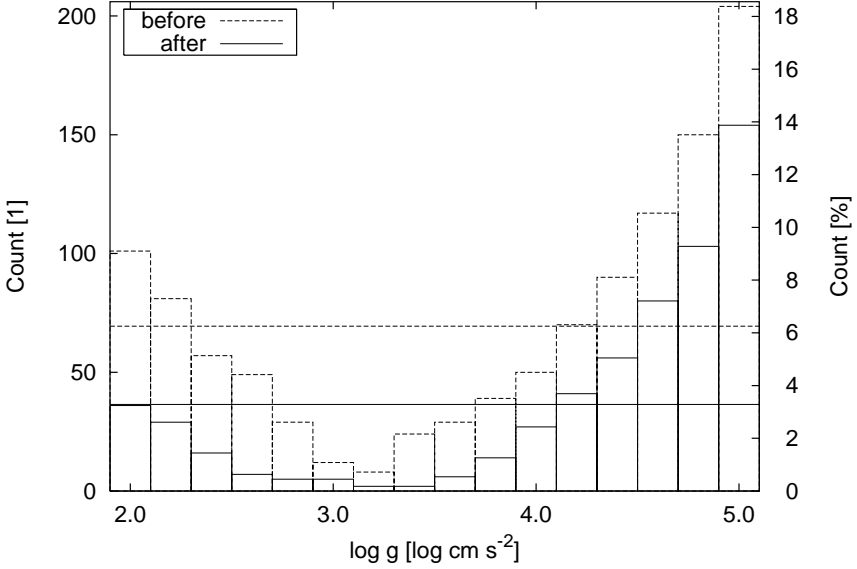


Figure 7: Number of missing models versus $\log g$ (added up for the other four parameters) before and after the interpolation. The two horizontal lines indicate the average values per $\log g$ step of 69.4 ($=1,110/16$) and 36.4 ($=583/16$).

$$\begin{aligned}
 p_L(x_3) &= f_3 = p_C(x_3) = p_R(x_3) \\
 f_4 &= p_C(x_4) = p_R(x_4)
 \end{aligned}$$

The interpolating function for $x_2 \leq x \leq x_3$ is defined as distance weighted mean of the two parabolas:

$$\begin{aligned}
 h(x) &= \frac{w_L(x) \cdot p_L(x) + w_R(x) \cdot p_R(x)}{w_L + w_R} = \\
 &= \frac{(x_3 - x) \cdot p_L(x) + (x - x_2) \cdot p_R(x)}{x_3 - x_2} = \\
 &= \frac{\Delta_{30} \cdot p_L(x) + \Delta_{02} \cdot p_R(x)}{\Delta_{32}} =
 \end{aligned} \tag{3}$$

which is a polynomial of order 3 in x . Note, that this method for interpolation is not restricted to equidistant grids and that $w_L + w_R$ is constant with respect to x . (The definition of Δ_{ij} is given in Eq. 8 on page 57.)

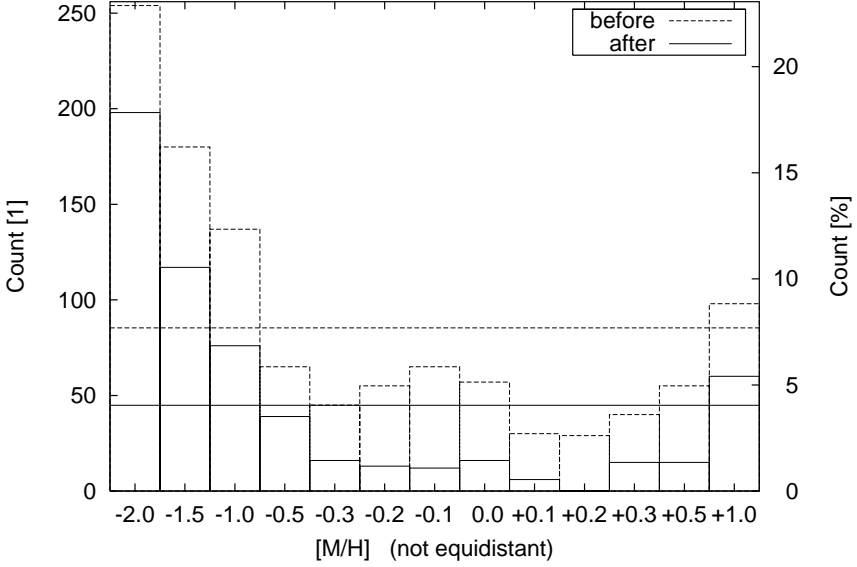


Figure 8: Number of missing models versus $[M/H]$ (added up for the other four parameters) before and after the interpolation. The two horizontal lines indicate the average values per $[M/H]$ step of 85.4 ($=1,110/13$) and 44.8 ($=583/13$).

It holds by (2) that

$$\begin{aligned}
 h(x_2) &= p_L(x_2) = f_2 & (= p_R(x_2) = p_C(x_2)) \\
 h(x_3) &= p_R(x_3) = f_3 & (= p_L(x_3) = p_C(x_3))
 \end{aligned}
 \tag{4}$$

For the first derivative we obtain

$$\begin{aligned}
 h'(x) &= \frac{-1 \cdot p_L(x) + (x_3 - x) \cdot p'_L(x) + 1 \cdot p_R(x) + (x - x_2) \cdot p'_R(x)}{x_3 - x_2} = \\
 &= \frac{w_L(x) \cdot p'_L(x) + w_R(x) \cdot p'_R(x)}{w_L + w_R} + \frac{[p_R - p_L](x)}{x_3 - x_2}
 \end{aligned}
 \tag{5}$$

which can be interpreted as the sum of the distance weighted derivatives and the interval averaged 'slope' between the right and left parabolas. The second term vanishes for $x = x_2$ and $x = x_3$ and hence

$$\begin{aligned}
 h'(x_2) &= p'_L(x_2) \\
 h'(x_3) &= p'_R(x_3)
 \end{aligned}
 \tag{6}$$

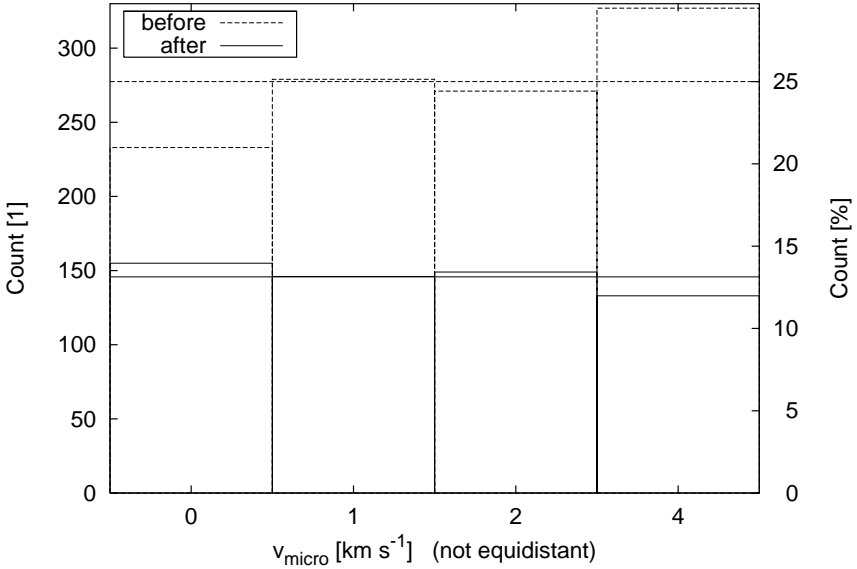


Figure 9: Number of missing models versus v_{micro} (added up for the other four parameters) before and after the interpolation. The two horizontal lines indicate the average values per v_{micro} step of 278 ($=1,110/4$) and 146 ($=583/4$).

Analogously, if we define left and right parabolas $\tilde{p}_L(x; x_{2,3,4})$ and $\tilde{p}_R(x; x_{3,4,5})$ (with $\tilde{p}_L(x) = p_R(x)$ because of the same three underlying grid nodes x_2, x_3 , and x_4), and the interpolating function $h(x)$ as the distance weighted mean of the two parabolas for the interval $x_3 \leq x \leq x_4$, we obtain

$$\begin{aligned} h'(x_3) &= \tilde{p}'_L(x_3) = p'_R(x_3) \\ h'(x_4) &= \tilde{p}'_R(x_4) \end{aligned} \quad (7)$$

Eq. 7 shows the continuity of the first derivative of the interpolating function $h'(x)$ at the grid nodes, and as (5) yields a uniquely defined quadratic polynomial in-between grid nodes, the continuity of the first derivative as obtained from distance weighted parabolic interpolation (2)-(3) holds for the entire of $x_1 \leq x \leq x_N$ for which interpolation is required. Due to this property and the fact of being third order in x , (2)-(3) is sometimes also called a “local spline interpolation”.

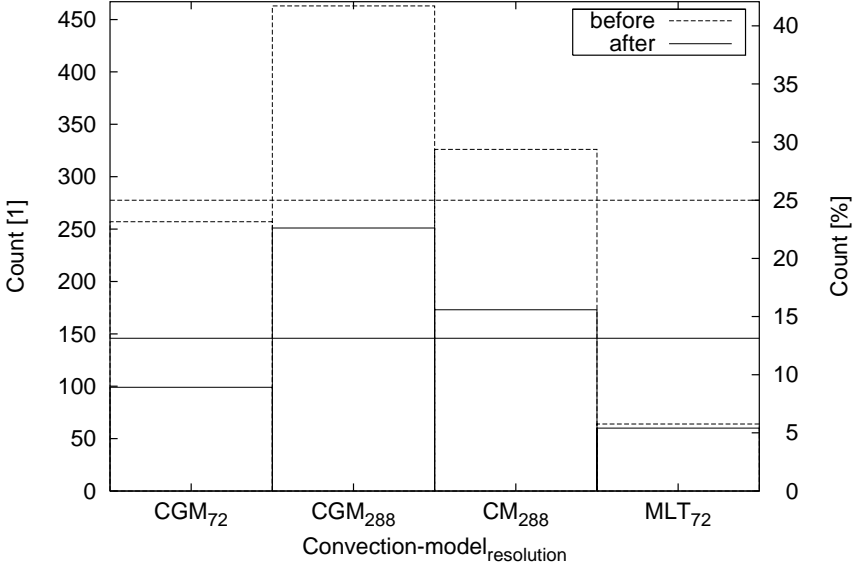


Figure 10: Number of missing models versus convection model (added up for the other four parameters) before and after the interpolation. The two horizontal lines indicate the average values per convection model of 278 ($=1,110/4$) and 146 ($=583/4$).

Using the definitions³

$$\begin{aligned}\Delta_{0j}(x) &= x - x_j & (8) \\ \Delta_{ij} &= x_i - x_j = -\Delta_{ji} = \Delta_{ik} + \Delta_{kj} = \Delta_{0j}(x_i)\end{aligned}$$

and Lagrange polynomials ($i, j, k = 1, \dots, 3$ or $2, \dots, 4$),⁴

$$L_{ijk}(x) = \frac{\Delta_{0j} \Delta_{0k}}{\Delta_{ij} \Delta_{ik}} = \frac{x - x_j}{x_i - x_j} \frac{x - x_k}{x_i - x_k} = L_{ikj}(x) \quad (9)$$

$$L_{ijk}(x_l) = \delta_{il} = \begin{cases} 1, & l = i \\ 0, & l \neq i \end{cases}, \quad l \in \{i, j, k\}$$

$$\begin{aligned}p_L(x) &= f_1 L_{123}(x) + f_2 L_{231}(x) + f_3 L_{312}(x) & (10) \\ p_R(x) &= f_2 L_{234}(x) + f_3 L_{342}(x) + f_4 L_{423}(x)\end{aligned}$$

³We will never use it in the following ambiguous manner: $x_0 - x_j = \Delta_{0j} \neq \Delta_{0j}(x_i) = x_i - x_j$, except for the unique case $i = 0$.

⁴ $L_{ijkl}(x) = (\Delta_{0j}/\Delta_{ij}) \cdot (\Delta_{0k}/\Delta_{ik}) \cdot (\Delta_{0l}/\Delta_{il})$,
 $p_C(x) = f_1 L_{1234}(x) + f_2 L_{2341}(x) + f_3 L_{3412}(x) + f_4 L_{4123}(x)$

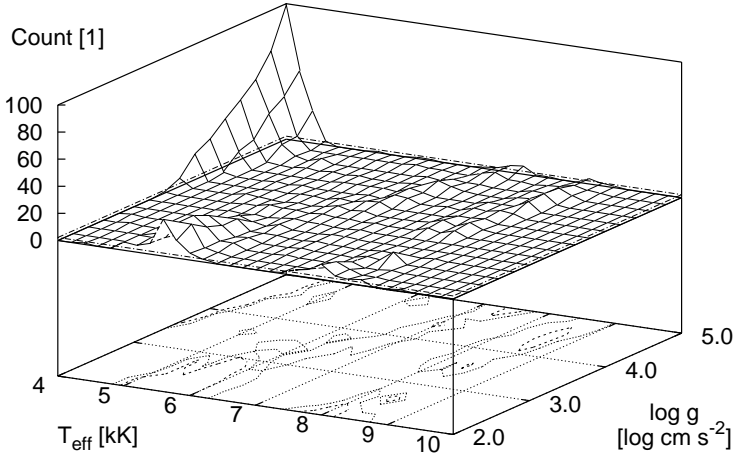


Figure 11: Number of missing models versus T_{eff} and $\log g$ (added up for the other three parameters) **before** the interpolation. The contour lines are drawn for the half and the double of the average value of 2.24 ($=1,110/31/16$) which itself is indicated by the bordering dashed line in the 3D part of the plot.

we obtain

$$\begin{aligned}
 \vec{h}(x) &= \frac{1}{\Delta_{32}} \left\{ \vec{f}_1 \underbrace{w_L(x)L_{123}(x)}_{\bar{L}_1(x)} + \vec{f}_4 \underbrace{w_R(x)L_{423}(x)}_{\bar{L}_4(x)} + \right. & (11) \\
 &+ \vec{f}_2 \underbrace{[w_L(x)L_{231}(x) + w_R(x)L_{234}(x)]}_{\bar{L}_2(x)} + \\
 &\left. + \vec{f}_3 \underbrace{[w_L(x)L_{312}(x) + w_R(x)L_{342}(x)]}_{\bar{L}_3(x)} \right\} = \\
 &= \frac{1}{\Delta_{32}} \left\{ \frac{1}{\Delta_{32}} \left[\vec{f}_2 \Delta_{30} \left(\frac{\Delta_{30}\Delta_{01}}{\Delta_{21}} + \frac{\Delta_{40}\Delta_{02}}{\Delta_{42}} \right) + \right. & (12) \\
 &\left. + \vec{f}_3 \Delta_{02} \left(\frac{\Delta_{30}\Delta_{01}}{\Delta_{31}} + \frac{\Delta_{40}\Delta_{02}}{\Delta_{43}} \right) \right] - \right.
 \end{aligned}$$

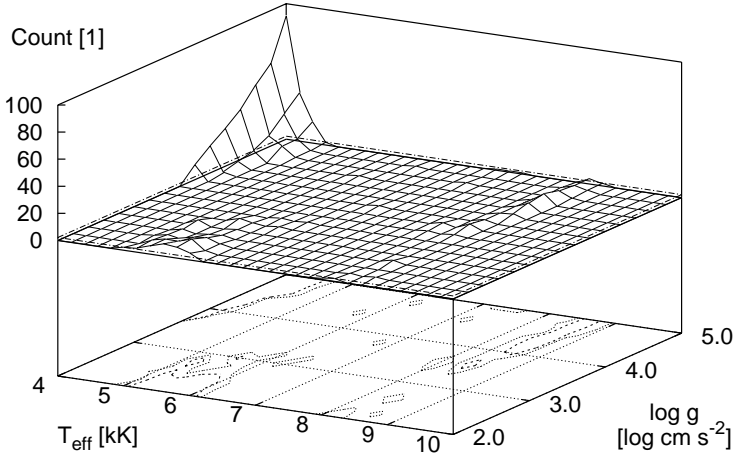


Figure 12: Number of missing models versus T_{eff} and $\log g$ (added up for the other three parameters) **after** the interpolation. The contour lines are drawn for the half and the double of the average value of 1.18 ($=583/31/16$) which itself is indicated by the bordering dashed line in the 3D part of the plot.

$$-\Delta_{30}\Delta_{02} \left[\vec{f}_1 \frac{\Delta_{30}}{\Delta_{31}\Delta_{21}} + \vec{f}_4 \frac{\Delta_{02}}{\Delta_{43}\Delta_{42}} \right] \Bigg\}$$

where we have taken into account that we also want to interpolate vector functions in the model grid, because every \vec{f}_i represents a model atmosphere with up to $6 \cdot 288 = 1728$ numerical values. Therefore, the formula is optimized for computational performance⁵ such that each \vec{f}_i occurs only once (and all $\Delta_{ij} \geq 0$ for $x_1 < x_2 \leq x = x_0 \leq x_3 < x_4$).

Interpolation at grid nodes (Ip1)

Formula (11) is used for closing the gaps in the model atmosphere grids. Assume a missing model at

$$\vec{x}_0 = (t_0, g_0, z_0, v_0, c_0) \quad (13)$$

⁵Fortran90 array arithmetic

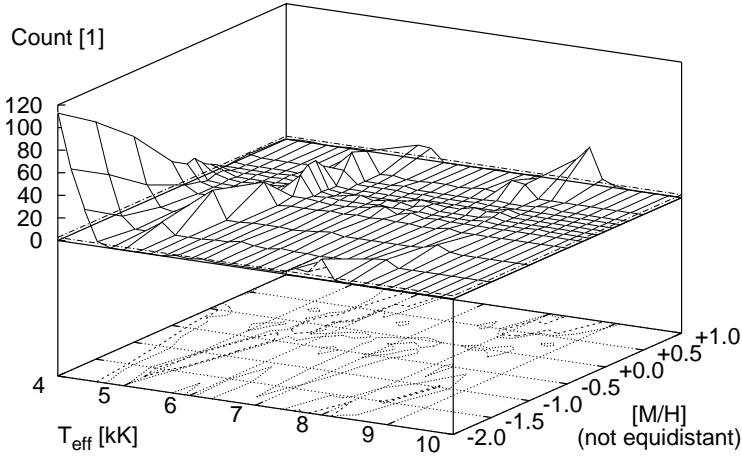


Figure 13: Number of missing models versus T_{eff} and $[M/H]$ (added up for the other three parameters) **before** the interpolation. The contour lines are drawn for the half and the double of the average value of 2.75 ($=1,110/31/13$) which itself is indicated by the bordering dashed line in the 3D part of the plot.

where

$$\begin{aligned}
 t &= x_{(1)} = T_{\text{eff}} \\
 g &= x_{(2)} = \log g \\
 z &= x_{(3)} = [M/H] \\
 v &= x_{(4)} = v_{\text{micro}} \\
 [c &= x_{(5)} = \text{Convection model}_{\text{resolution}}]
 \end{aligned} \tag{14}$$

Note, that $c = x_{(5)}$ is a non-numerical parameter of the grid and hence can not be used for interpolation. Therefore, we have to consider four 4D subgrids (for each value of c) and neglect c hereafter when talking about interpolation.

For each $x_{(s)}$, $s = 1, \dots, 4$, an independent 1D interpolation is done in the following manner:

First we look for the next two left and right neighbors in the grid. Their parameters and the corresponding (vector) function values⁶ are:

⁶In the case of ATLAS9 model atmospheres the quantities representing the atmospheric

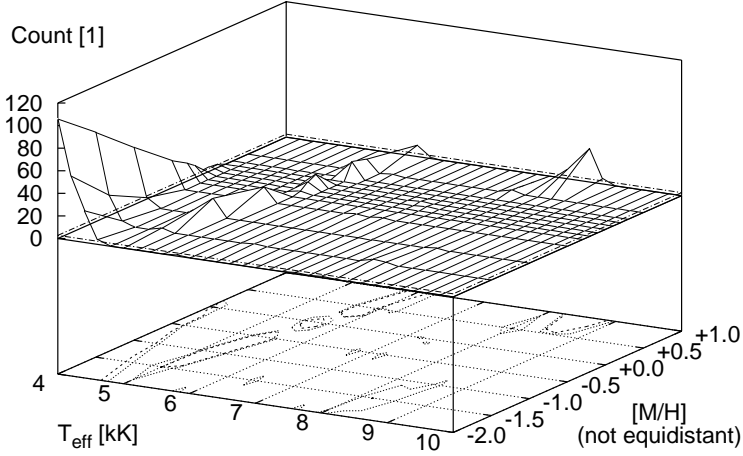


Figure 14: Number of missing models versus T_{eff} and $[M/H]$ (added up for the other three parameters) **after** the interpolation. The contour lines are drawn for the half and the double of the average value of 1.45 ($=583/31/13$) which itself is indicated by the bordering dashed line in the 3D part of the plot.

for $s = 1$: $x_{(s)} = x_{(1)} = t$ ($t_1 < t_2 \leq t_0 \leq t_3 < t_4$):

$$\begin{aligned}
 \text{left} & : \vec{x}_{1(1)} = (t_1, g_0, z_0, v_0), \vec{x}_{2(1)} = (t_2, g_0, z_0, v_0) \\
 \text{right} & : \vec{x}_{3(1)} = (t_3, g_0, z_0, v_0), \vec{x}_{4(1)} = (t_4, g_0, z_0, v_0) \\
 \vec{f}_{i(1)} & = \vec{f}(\vec{x}_{i(1)}) = \vec{f}(t_i, g_0, z_0, v_0)
 \end{aligned} \tag{15}$$

for $s = 2$: $x_{(s)} = x_{(2)} = g$ ($g_1 < g_2 \leq g_0 \leq g_3 < g_4$):

$$\begin{aligned}
 \text{left} & : \vec{x}_{1(2)} = (t_0, g_1, z_0, v_0), \vec{x}_{2(2)} = (t_0, g_2, z_0, v_0) \\
 \text{right} & : \vec{x}_{3(2)} = (t_0, g_3, z_0, v_0), \vec{x}_{4(2)} = (t_0, g_4, z_0, v_0) \\
 \vec{f}_{i(2)} & = \vec{f}(\vec{x}_{i(2)}) = \vec{f}(t_0, g_i, z_0, v_0)
 \end{aligned} \tag{16}$$

for $s = 3$: $x_{(s)} = x_{(3)} = z$ ($z_1 < z_2 \leq z_0 \leq z_3 < z_4$):

$$\text{left} : \vec{x}_{1(3)} = (t_0, g_0, z_1, v_0), \vec{x}_{2(3)} = (t_0, g_0, z_2, v_0) \tag{17}$$

structure are the integrated mass, the temperature, the gas pressure, the electron density, the Rosseland mean opacity and the radiative acceleration at each depth.

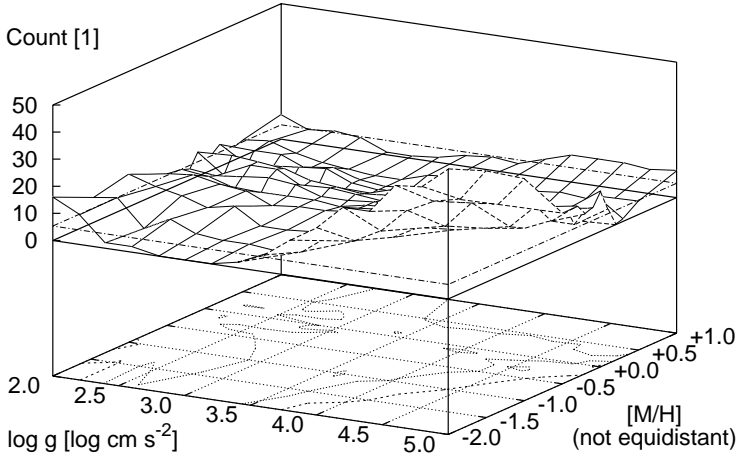


Figure 15: Number of missing models versus $\log g$ and $[M/H]$ (added up for the other three parameters) **before** the interpolation. The contour lines are drawn for the half and the double of the average value of 5.34 ($=1,110/16/13$) which itself is indicated by the bordering dashed line in the 3D part of the plot.

$$\begin{aligned} \text{right} & : \vec{x}_{3(3)} = (t_0, g_0, z_3, v_0), \vec{x}_{4(3)} = (t_0, g_0, z_4, v_0) \\ \vec{f}_{i(3)} & = \vec{f}(\vec{x}_{i(3)}) = \vec{f}(t_0, g_0, z_i, v_0) \end{aligned}$$

for $s = 4$: $x_{(4)} = x_{(4)} = v$ ($v_1 < v_2 \leq v_0 \leq v_3 < v_4$):

$$\begin{aligned} \text{left} & : \vec{x}_{1(4)} = (t_0, g_0, z_0, v_1), \vec{x}_{2(4)} = (t_0, g_0, z_0, v_2) \\ \text{right} & : \vec{x}_{3(4)} = (t_0, g_0, z_0, v_3), \vec{x}_{4(4)} = (t_0, g_0, z_0, v_4) \\ \vec{f}_{i(4)} & = \vec{f}(\vec{x}_{i(4)}) = \vec{f}(t_0, g_0, z_0, v_i) \end{aligned} \quad (18)$$

Subsuming the third lines we can write⁷

$$\vec{f}_{i(s)} = \vec{f}(\vec{x}_{i(s)}) \quad (19)$$

⁷One remark about notation: $\vec{x} = (x_{(1)}, x_{(2)}, x_{(3)}, x_{(4)}, x_{(5)}) = (t, g, z, v, c)$, $\vec{x}_0 = (x_{(1)0}, x_{(2)0}, x_{(3)0}, x_{(4)0}, x_{(5)0}) = (t_0, g_0, z_0, v_0, c_0)$; e.g. $s=3$: $\vec{x}_{i(3)} = (t_0, g_0, z_i, v_0, c_0) = (x_{(1)0}, x_{(2)0}, \mathbf{x}_{(3)i}, x_{(4)0}, x_{(5)0})$. I.e. $x_{(s)i}$ is a component of $\vec{x}_{i(s)}$. Thus, we could define $m_{(s)}(\vec{x}_{i(s)}) = m_{x_{(s)i}}$ (Eqs. 21, 26).

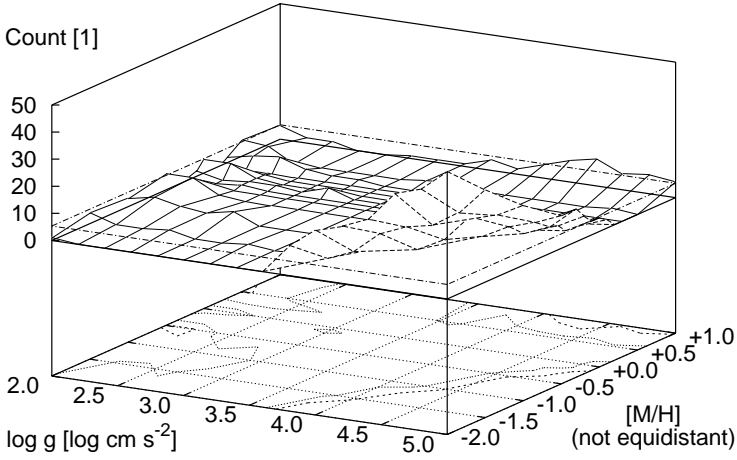


Figure 16: Number of missing models versus $\log g$ and $[M/H]$ (added up for the other three parameters) **after** the interpolation. The contour lines are drawn for the half and the double of the average value of 2.80 ($=583/16/13$) which itself is indicated by the bordering dashed line in the 3D part of the plot.

With Eq. 11 (p. 58, $\Delta_{ij} \rightarrow \Delta_{ij(s)}$) the 1D interpolation for $x_{(s)}$ at \vec{x}_0 can be done:

$$\vec{h}_{(s)}(\vec{x}_0) = \vec{h}_{(s)}(t_0, g_0, z_0, v_0) = \frac{1}{\Delta_{32(s)}} \sum_{i=1}^4 \vec{f}_{i(s)} \bar{L}_i(x_{(s)0}) \quad (20)$$

We introduce weighting factors to allow for neighbors which are not at the next grid positions because there are further gaps in between:

$$w_{(s)} = \frac{6}{d_{(s)}} \quad (21)$$

$$d_{(s)} = \sum_{i=1}^4 |m_{x_{(s)0}} - m_{x_{(s)i}}|$$

where $m_{x_{(s)i}}$ is the grid position number of $x_{(s)i}$ (number corresponding to $x_{(s)i}$ when counting every grid step from the minimum to the maximum value

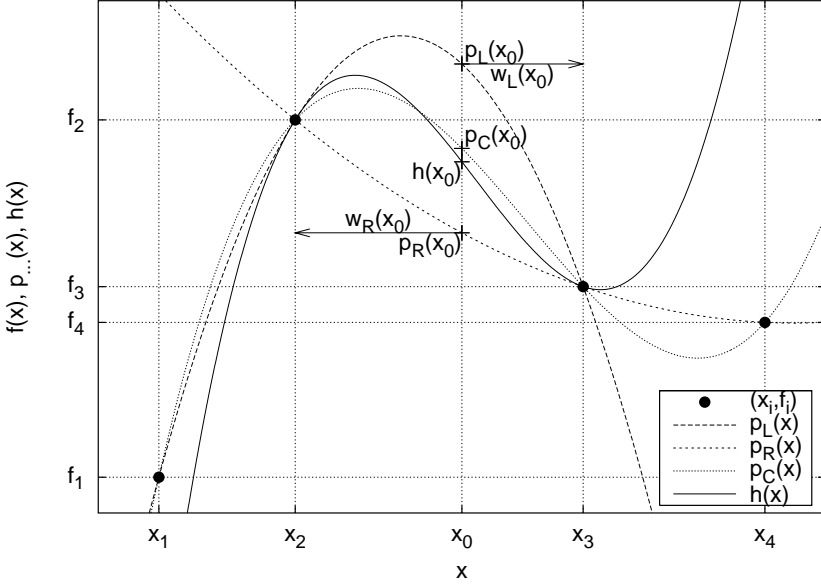


Figure 17: Example for distance weighted parabolas as interpolating function compared to a cubic polynomial. $h(x)$ is only used for $x = x_0 \in [x_2, x_3]$. Within this interval it is quite close to $p_C(x)$ while outside the difference between the two grows rapidly.

of parameter $x_{(s)}$).⁸

Thus, $d_{(s)}$ is the sum of distances in grid step units of the interpolating point $x_{(s)0}$ from the neighbors $x_{(s)i}$ (or in point of view of 4D: sum of distances of \vec{x}_0 from $\vec{x}_{i(s)}$). In the optimal case it is $2 + 1 + 1 + 2 = 6$ and the maximum weighting factor is 1.0.

It can happen that only three neighbors are available (2:1 left:right or 1:2 or 3:0 or 0:3). This is always the case for $x_{(4)} = v = v_{\text{micro}}$, because there are only four different values in total (0, 1, 2, and 4). In this case, a simple quadratic (2nd order polynomial) interpolation is done and the weighting factor is given by:

$$w_{(s)[3]} = \frac{1}{2} \cdot \frac{4}{d_{(s)[3]}} = \frac{2}{d_{(s)[3]}} \leq \frac{1}{2} \quad (22)$$

⁸E.g. for $s = 1$, $x_{(s)i} = x_{(1)i} = t_i = T_{\text{eff},i} = 4000$: $m_{x_{(s)i}} = 1$ (2 for 4200, 3 for 4400, ..., 31 for 10000). Note, that we are considering interpolation at grid nodes (missing models) and consequently $m_{x_{(s)0}}$ is also defined uniquely.

Interpolation	Number of models			Percentage		
	before	after	Δ	before	after	Δ
Primary (fully)	67,674	67,804	+130	73.94	74.09	+0.14
Secondary (part.)	22,736	23,133	+397	24.84	25.28	+0.43
Missing / Interpol.	1,110	583	-527	1.21	0.64	-0.58
Sum	91,520	91,520	± 0	100.00	100.00	± 0.00

Table 5: Numbers and percentages of models fulfilling the primary and secondary SMT convergence criteria before and after the interpolation. Due to rounding the percentages may not sum up to 100.00.

with a factor 1/2 to account for the lower accuracy of this interpolation due to one fewer neighbor and 4 as minimal (optimal) distance sum.

In case of only two neighbors (2:0, 1:1, or 0:2) linear interpolation is applied and

$$w_{(s)[2]} = \left(\frac{1}{2}\right)^2 \cdot \frac{2}{d_{(s)[2]}} = \frac{1}{2d_{(s)[2]}} \leq \frac{1}{4} \tag{23}$$

With only one neighbor (1:0, 0:1) we use identical interpolation (as an initial guess) and

$$w_{(s)[1]} = \left(\frac{1}{2}\right)^3 \cdot \frac{1}{d_{(s)[1]}} = \frac{1}{8d_{(s)[1]}} \leq \frac{1}{8} \tag{24}$$

If there is no neighbor at all (0:0) we set

$$w_{(s)[0]} = 0 \tag{25}$$

The general expressions for $n = 1, \dots, 4$ are:

$$w_{(s)[n]} = \left(\frac{1}{2}\right)^{4-n} \cdot \frac{(-n^3 + 9n^2 - 14n + 12)/6}{d_{(s)[n]}}, \tag{26}$$

$$d_{(s)[n]} = \sum_{i=1}^n |m_{x_{(s)0}} - m_{x_{(s)i}}|$$

Once the 1D interpolations in all four dimensions $x_{(s)}(t, g, z, v)$ have been completed, the final result of grid gap closing interpolation is given by

$$\vec{h}(\vec{x}_0) = \frac{1}{W} \cdot \sum_{s=1}^4 w_{(s)} \cdot \vec{h}_{(s)}(\vec{x}_0) \tag{27}$$

$$q(\vec{x}_0) = \frac{W}{4}$$

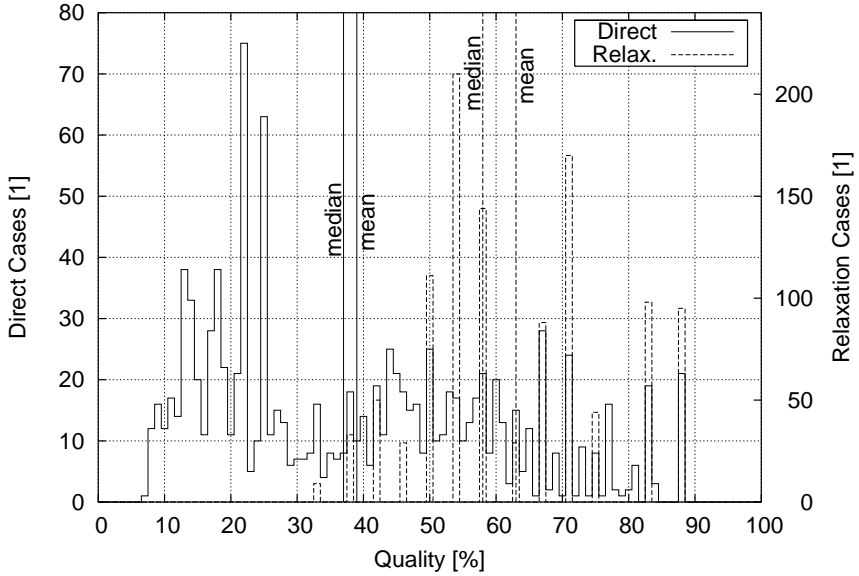


Figure 18: Histogram of neighbor/interpolation qualities in the grid with 1,110 missing/interpolated models. For relaxation interpolation there are only 13 possible values.

$$W = \sum_{s=1}^4 w_{(s)}$$

Thus, $w_{(s)}$ not only takes into account the distance of neighbors in one dimension but also the weighting among the 4 dimensions. Strictly speaking it should be written as $w_{(s)[n_{(s)}(\vec{x}_0)]}$.

$q(\vec{x}_0)$ is the *interpolation/neighbor quality*. The maximum possible value of it is 1.0 (100%) provided that in all four dimensions there is an optimal 2:2 neighbor situation. In our grid $q(\vec{x}_0)$ is never 0.0,⁹ hence $\vec{h}(\vec{x}_0)$ is always well defined.

This procedure (Eq. 13 - 27) is applied to all missing models in each of the four convection model subgrids. There are two ways to do so. We call the first one *direct interpolation*: make one run for every missing model as described above and take only converged models as neighbors. The second one

⁹That would happen in case of no neighbors at all in all four dimensions. The maximum value is 87.5% and not 100% because of the maximal three neighbors in v .

Neighbors	Dimension $s : x_{(s)}$				#	Distance sum	Weight
left:right	1:t	2:g	3:z	4:v	n	$d_{(s)[n]}$	$w_{(s)[n]}$
3:0, 0:3	+	+	+	+	3	$3+2+1=6$	$1/2 \cdot 4/6 = 1/3$
2:1, 1:2	+	+	+	+	3	$2+1+1=4$	$1/2 \cdot 4/4 = 1/2$
2:2	+	+	+	-	4	$2+1+1+2=6$	$6/6 = 1$
$\sum_{s=1}^4 w_{(s)[n]}$	$\{1/3, 1/2, 1\}_t + \{1/3, 1/2, 1\}_g + \{1/3, 1/2, 1\}_z + \{1/3, 1/2\}_v =$ $= \{1\frac{1}{3}, 1\frac{1}{2}, 1\frac{2}{3}, 1\frac{5}{6}, 2, 2\frac{1}{6}, 2\frac{2}{3}, 2\frac{1}{2}, 2\frac{2}{3}, 2\frac{5}{6}, 3, 3\frac{1}{3}, 3\frac{1}{2}\}$						
$q(\vec{x}_0)$ [%]	$\{33\frac{1}{3}, 37\frac{1}{2}, 41\frac{2}{3}, 45\frac{5}{6}, 50, 54\frac{1}{6}, 58\frac{1}{3}, 62\frac{1}{2}, 66\frac{2}{3}, 70\frac{5}{6}, 75, 83\frac{1}{3}, 87\frac{1}{2}\}$						

Table 6: Possible neighbor situations and values of distance sum $d_{(s)[n]}$ and weighting factor $w_{(s)[n]}$ for the four parameters $x_{(s)}$ and resulting 13 possible values of the neighbor/interpolation quality $q(\vec{x}_0)$ for relaxation interpolation.

we call *relaxation interpolation*: make one run after the other, and take also interpolated models as neighbors,¹⁰ as long as the (values in the) model files change.¹¹

Fig. 18 shows the distribution of the qualities in the grid before the interpolation. In the case of relaxation interpolation there are no gaps in the grid anymore, because also the interpolated files are valid neighbors, hence there are only 13 possible values of neighbor quality (Table 6).

We save the directly interpolated files and the relaxation interpolated ones in different directories and apply SMGT to them. Different numbers of models converge (fully and partially) in the two directories. As there are more converged files in the relaxation directory, which we therefore rate as the ‘better’ one, we replace the interpolated files in our grid with one of the appropriate newly converged ones in following order:

1. Fully converged from relaxation interpolated model
2. Fully converged from directly interpolated model
3. Partially converged from relaxation interpolated model
4. Partially converged from directly interpolated model

¹⁰Alternatively, we could save the interpolated files from the current run elsewhere and replace all the old interpolated models from the last run with the new ones from the current run only after the current run has been completed. But as we are not interested in the intermediate results and to save time a recently interpolated file replaces the old one immediately.

¹¹The comparison is done character by character (csh-script) which corresponds to a numerical significance of at most 8 digits, depending on the variable to be interpolated.

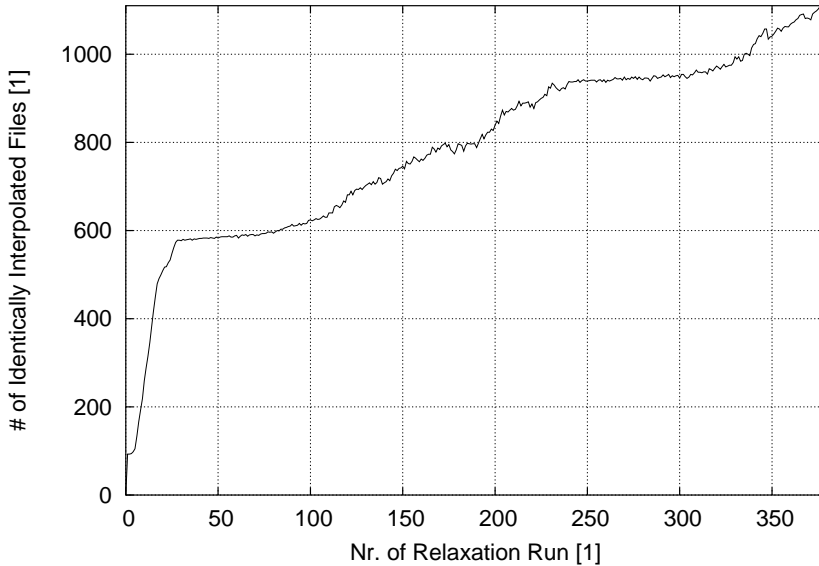


Figure 19: Relaxation progress in loop 1: Number of identically interpolated files in consecutive relaxation runs versus number of relaxation run (added up over the four convection model subgrids). There are actually stages with decreasing number. 379 runs are needed to converge the procedure for the 1110 missing/interpolated files which makes an average slope of about 3 files per run.

In this way we can replace 427 (115, 12, 267, and 33 from group 1, 2, 3, and 4, respectively) interpolated files by converged models.

Fig. 19 shows the relaxation progress of the 1,100 missing models in our grid. 379 runs are needed to converge the procedure.

Now with this partially new grid¹² we can repeat the whole procedure: interpolating directly and relaxationally, replacing interpolated files with converged models. After 6 such loops no further progress can be achieved in our grid. A total of 527 (=427+62+20+10+6+2+0 in the 7 loops) out of the originally 1,110 missing models converged, 583 are left interpolated. (They are called “missing models after the interpolation” in Figs. 11 - 16.)

The relaxation progress of the other loops (2 - 7) looks similar to that of loop 1 (Fig. 19).

328 out of the 527 models converged both starting from a relaxationally

¹²Converged files contain different values compared to the interpolated ones taken as their SMTG starting models.

interpolated model and a directly interpolated one, 137 converged only in the relaxation case and 62 only in the direct one, so without relaxation interpolation only 390 would have converged (720 left missing/interpolated) and without direct interpolation 465 (645).

To sum up, the whole grid gap interpolation consists of seven loops, each loop of one direct interpolation and one relaxation interpolation, each relaxation interpolation of some hundred runs,¹³ each direct interpolation and each run consists of four 1D interpolations for every missing model. This gives a total of 5,893,792 1D interpolations (of vectors/arrays with $6 \cdot 288 = 1728$ or $6 \cdot 72 = 432$ components). After each loop SMGT is applied to the interpolated files, altogether 4,774 times.

Interpolation between grid nodes (Ip2)

The Ip2 is real 4D, it expects a complete 4^4 (256) neighborhood¹⁴ of grid nodes (existing model files, converged or interpolated)

$$\vec{f}_{ijkl} = \vec{f}(\vec{x}_{ijkl}) = \vec{f}(x_{(1)i}, x_{(2)j}, x_{(3)k}, x_{(4)l}) = \vec{f}(t_i, g_j, z_k, v_l) \quad (28)$$

where $i, j, k, l = 1, \dots, 4$ around the interpolation point $\vec{x}_0 = (t_0, g_0, z_0, v_0)$ which now is not a grid node but still we assume $x_{(s)1} < x_{(s)2} \leq x_{(s)0} \leq x_{(s)3} < x_{(s)4}$ for $s = 1, \dots, 4$, i.e. 2 left and 2 right 'neighbors', in each of the 4 'directions' (dimensions, parameters) $x_{(s)}$.

Using again Lagrange's formalism for polynomial interpolation (cf. Eq. 8, p. 57)

$$\Delta_{0j(s)} = x_{(s)} - x_{(s)j} \quad (29)$$

$$\Delta_{ij(s)} = x_{(s)i} - x_{(s)j}$$

$$L_{ijk}(x_{(s)}) = \frac{\Delta_{0j(s)} \Delta_{0k(s)}}{\Delta_{ij(s)} \Delta_{ik(s)}} \quad (30)$$

16 (2^4) left-right combined parabolas can be defined, e.g. (cf. Eq. 9, p. 57):

$$\begin{aligned} p_{LRRLL}(\vec{x}) &= \vec{f}_{1221} \cdot L_{123}(t) \cdot L_{234}(g) \cdot L_{234}(z) \cdot L_{123}(v) + \\ &+ \vec{f}_{1222} \cdot L_{123}(t) \cdot L_{234}(g) \cdot L_{234}(z) \cdot L_{231}(v) + \\ &+ \vec{f}_{1223} \cdot L_{123}(t) \cdot L_{234}(g) \cdot L_{234}(z) \cdot L_{312}(v) + \\ &+ \vec{f}_{1231} \cdot L_{123}(t) \cdot L_{234}(g) \cdot L_{342}(z) \cdot L_{123}(v) + \end{aligned} \quad (31)$$

¹³379, 365, 340, 331, 280, 216, and 166 in the loops 1 to 7, respectively.

¹⁴The base 4 is due to $i = 1, \dots, 4$ (4 neighbors), the exponent 4 is due to $s = 1, \dots, 4$ (4D).

$$\times \left(L_{123}(v) \cdot \vec{f}_{3441} + L_{231}(v) \cdot \vec{f}_{3442} + L_{312}(v) \cdot \vec{f}_{3443} \right) \Big] \Big\}$$

The complete formula contains 81 (3^4) \vec{f}_{ijkl} : $i = 1, 2, 3, j = 2, 3, 4, k = 2, 3, 4, l = 1, 2, 3$. The order of bracketing (v is innermost) is arbitrary. The distance (4D hypervolume) weight for this example is

$$\begin{aligned} w_{\text{LRRL}}(\vec{x}) &= \Delta_{30(1)} \cdot \Delta_{02(2)} \cdot \Delta_{02(3)} \cdot \Delta_{30(4)} = & (33) \\ &= w_{\text{L}}(t) \cdot w_{\text{R}}(g) \cdot w_{\text{R}}(z) \cdot w_{\text{L}}(v) = \\ &= (t_3 - t)(g - g_2)(z - z_2)(v_3 - v) \end{aligned}$$

The general expressions are given by

$$\begin{aligned} p_{\hat{i}\hat{j}\hat{k}\hat{l}}(\vec{x}) &= \sum_{i=1+\hat{i}}^{3+\hat{i}} \sum_{j=1+\hat{j}}^{3+\hat{j}} \sum_{k=1+\hat{k}}^{3+\hat{k}} \sum_{l=1+\hat{l}}^{3+\hat{l}} \vec{f}_{ijkl} \times & (34) \\ &\times \underbrace{\prod_{\substack{i'=1+\hat{i} \\ i' \neq i}}^{3+\hat{i}} \frac{\Delta_{0i'(1)}}{\Delta_{ii'(1)}}}_{L_{ii'_1i'_2}(t)} \underbrace{\prod_{\substack{j'=1+\hat{j} \\ j' \neq j}}^{3+\hat{j}} \frac{\Delta_{0j'(2)}}{\Delta_{jj'(2)}}}_{L_{jj'_1j'_2}(g)} \underbrace{\prod_{\substack{k'=1+\hat{k} \\ k' \neq k}}^{3+\hat{k}} \frac{\Delta_{0k'(3)}}{\Delta_{kk'(3)}}}_{L_{kk'_1k'_2}(z)} \underbrace{\prod_{\substack{l'=1+\hat{l} \\ l' \neq l}}^{3+\hat{l}} \frac{\Delta_{0l'(4)}}{\Delta_{ll'(4)}}}_{L_{ll'_1l'_2}(v)} = \\ &= \sum_{i=1+\hat{i}}^{3+\hat{i}} L_{ii'_1i'_2}(t) \times \left\{ \sum_{j=1+\hat{j}}^{3+\hat{j}} L_{jj'_1j'_2}(g) \times \right. \\ &\quad \left. \times \left[\sum_{k=1+\hat{k}}^{3+\hat{k}} L_{kk'_1k'_2}(z) \times \left(\sum_{l=1+\hat{l}}^{3+\hat{l}} L_{ll'_1l'_2}(v) \times \vec{f}_{ijkl} \right) \right] \right\} \end{aligned}$$

$$w_{\hat{i}\hat{j}\hat{k}\hat{l}}(\vec{x}) = w_{\hat{i}}(t) \cdot w_{\hat{j}}(g) \cdot w_{\hat{k}}(z) \cdot w_{\hat{l}}(v). \quad (35)$$

where \hat{i} reaches the value $L = 0$ in case of a left parabola (in $x_{(1)} = t$) and $R = 1$ in case of a right one, and analogously for \hat{j} ($x_{(2)} = g$), \hat{k} ($x_{(3)} = z$), and \hat{l} ($x_{(4)} = v$), and

$$w_{\text{L}}(x_{(s)}) = \Delta_{30(s)} \quad (36)$$

$$w_{\text{R}}(x_{(s)}) = \Delta_{02(s)}$$

$$W = \sum_{\hat{i}=\text{L,R}} \sum_{\hat{j}=\text{L,R}} \sum_{\hat{k}=\text{L,R}} \sum_{\hat{l}=\text{L,R}} w_{\hat{i}\hat{j}\hat{k}\hat{l}}(\vec{x}) = \quad (37)$$

$$= \prod_{s=1}^4 \Delta_{32(s)} = (t_3 - t_2)(g_3 - g_2)(z_3 - z_2)(v_3 - v_2)$$

For the interpolating function we obtain (cf. Eq. 3, p. 54)

$$\begin{aligned}
\vec{h}(\vec{x}) &= \frac{1}{W} \sum_{\hat{i}=0}^1 \sum_{\hat{j}=0}^1 \sum_{\hat{k}=0}^1 \sum_{\hat{l}=0}^1 w_{\hat{i}\hat{j}\hat{k}\hat{l}}(\vec{x}) \cdot p_{\hat{i}\hat{j}\hat{k}\hat{l}}(\vec{x}) = & (38) \\
&= \frac{1}{W} \sum_{\hat{i}=0}^1 w_{\hat{i}}(t) \sum_{\hat{j}=0}^1 w_{\hat{j}}(g) \sum_{\hat{k}=0}^1 w_{\hat{k}}(z) \sum_{\hat{l}=0}^1 w_{\hat{l}}(v) \cdot p_{\hat{i}\hat{j}\hat{k}\hat{l}}(t, g, z, v) = \\
&= \frac{1}{W} \sum_{\hat{i}=0}^1 w_{\hat{i}}(t) \sum_{\hat{j}=0}^1 w_{\hat{j}}(g) \sum_{\hat{k}=0}^1 w_{\hat{k}}(z) \sum_{\hat{l}=0}^1 w_{\hat{l}}(v) \times \\
&\quad \times \sum_{i=1+\hat{i}}^{3+\hat{i}} L_{ii'i_2'}(t) \sum_{j=1+\hat{j}}^{3+\hat{j}} L_{jj'j_2'}(g) \sum_{k=1+\hat{k}}^{3+\hat{k}} L_{kk'k_2'}(z) \sum_{l=1+\hat{l}}^{3+\hat{l}} L_{ll'l_2'}(v) \times \\
&\quad \times \vec{f}_{ijkl} = \\
&= \frac{1}{W} \sum_{\hat{i}=0}^1 w_{\hat{i}}(t) \sum_{i=1+\hat{i}}^{3+\hat{i}} L_{ii'i_2'}(t) \cdot \sum_{\hat{j}=0}^1 w_{\hat{j}}(g) \sum_{j=1+\hat{j}}^{3+\hat{j}} L_{jj'j_2'}(g) \times \\
&\quad \times \sum_{\hat{k}=0}^1 w_{\hat{k}}(z) \sum_{k=1+\hat{k}}^{3+\hat{k}} L_{kk'k_2'}(z) \cdot \underbrace{\sum_{\hat{l}=0}^1 w_{\hat{l}}(v) \sum_{l=1+\hat{l}}^{3+\hat{l}} L_{ll'l_2'}(v)}_{\vec{H}_{ijk}(v)} \cdot \vec{f}_{ijkl} \\
&\quad \underbrace{\hspace{10em}}_{\vec{H}_{ij}(z,v)}
\end{aligned}$$

This can be transformed into a sum of (tetracubic) separable expressions again by the definitions

$$\begin{aligned}
\bar{L}_1(x) &= w_L(x) \cdot L_{123}(x) & (39) \\
\bar{L}_2(x) &= w_L(x) \cdot L_{231}(x) + w_R(x) \cdot L_{234}(x) \\
\bar{L}_3(x) &= w_L(x) \cdot L_{312}(x) + w_R(x) \cdot L_{342}(x) \\
\bar{L}_4(x) &= w_R(x) \cdot L_{423}(x)
\end{aligned}$$

And so

$$\begin{aligned}
\vec{H}_{ijk}(v) &= \sum_{\hat{l}=0}^1 w_{\hat{l}}(v) \sum_{l=1+\hat{l}}^{3+\hat{l}} L_{ll'l_2'}(v) \cdot \vec{f}_{ijkl} = & (40) \\
&= \left[w_0(v) \sum_{l=1}^3 + w_1(v) \sum_{l=2}^4 \right] L_{ll'l_2'}(v) \cdot \vec{f}_{ijkl} =
\end{aligned}$$

$$\begin{aligned}
&= w_L(v) \left\{ L_{123}(v) \vec{f}_{ijk1} + L_{231}(v) \vec{f}_{ijk2} + L_{312}(v) \vec{f}_{ijk3} \right\} + \\
&+ w_R(v) \left\{ L_{234}(v) \vec{f}_{ijk2} + L_{342}(v) \vec{f}_{ijk3} + L_{423}(v) \vec{f}_{ijk4} \right\} = \\
&= \vec{f}_{ijk1} \underbrace{w_L(v) L_{123}(v)}_{\bar{L}_1(v)} + \\
&+ \vec{f}_{ijk2} \underbrace{\{w_L(v) L_{231}(v) + w_R(v) L_{234}(v)\}}_{\bar{L}_2(v)} + \\
&+ \vec{f}_{ijk3} \underbrace{\{w_L(v) L_{312}(v) + w_R(v) L_{342}(v)\}}_{\bar{L}_3(v)} + \\
&+ \vec{f}_{ijk4} \underbrace{w_R(v) L_{423}(v)}_{\bar{L}_4(v)} = \\
&= \sum_{l=1}^4 \bar{L}_l(v) \cdot \vec{f}_{ijkl} \tag{41}
\end{aligned}$$

Analogously¹⁵

$$\begin{aligned}
\vec{H}_{ij}(z, v) &= \sum_{\hat{k}=0}^1 w_{\hat{k}}(z) \sum_{k=1+\hat{k}}^{3+\hat{k}} L_{kk'_1k'_2}(z) \cdot \vec{H}_{ijk}(v) = \\
&= \sum_{k=1}^4 \bar{L}_k(z) \cdot \vec{H}_{ijk}(v) \tag{42}
\end{aligned}$$

$$\begin{aligned}
\vec{H}_i(g, z, v) &= \sum_{\hat{j}=0}^1 w_{\hat{j}}(g) \sum_{j=1+\hat{j}}^{3+\hat{j}} L_{jj'_1j'_2}(g) \cdot \vec{H}_{ij}(z, v) = \\
&= \sum_{j=1}^4 \bar{L}_j(g) \cdot \vec{H}_{ij}(z, v) \tag{43}
\end{aligned}$$

$$\begin{aligned}
\vec{H}(t, g, z, v) &= \sum_{\hat{i}=0}^1 w_{\hat{i}}(t) \sum_{i=1+\hat{i}}^{3+\hat{i}} L_{ii'_1i'_2}(t) \cdot \vec{H}_i(g, z, v) = \tag{44}
\end{aligned}$$

¹⁵In Fortran90 the (one line) command for $\vec{H}_{ijk}(v) \rightarrow \vec{H}_{ij}(z, v)$ is e.g.: `hij(:, :, :) = Lz(1)*hijk(:, :, :, 1) + Lz(2)*hijk(:, :, :, 2) + Lz(3)*hijk(:, :, :, 3) + Lz(4)*hijk(:, :, :, 4)`, where the first ‘:’ stands for the ‘ $\vec{\cdot}$ ’ of \vec{H} , and the second and third for i and j , respectively; no DO loops have to be written for this. $k = 1, \dots, 4$ is written explicitly.

$$\begin{aligned}
= \vec{H}(\vec{x}) &= \sum_{i=1}^4 \bar{L}_i(t) \cdot \vec{H}_i(g, z, v) \\
\vec{h}(\vec{x}) &= \frac{1}{W} \vec{H}(\vec{x})
\end{aligned} \tag{45}$$

Thus, we finally can write (cf. Eq. 11, p. 58)

$$\begin{aligned}
\vec{h}(\vec{x}) &= \frac{1}{W} \sum_{i=1}^4 \sum_{j=1}^4 \sum_{k=1}^4 \sum_{l=1}^4 \vec{f}_{ijkl} \bar{L}_i(t) \bar{L}_j(g) \bar{L}_k(z) \bar{L}_l(v) = \\
&= \frac{1}{W} \sum_{i=1}^4 \bar{L}_i(t) \underbrace{\sum_{j=1}^4 \bar{L}_j(g) \sum_{k=1}^4 \bar{L}_k(z) \underbrace{\sum_{l=1}^4 \bar{L}_l(v) \vec{f}_{ijkl}}_{\vec{H}_{ijk}(v)}}_{\vec{H}_{ij}(z,v)} \\
&\quad \underbrace{\hspace{10em}}_{\vec{H}_i(g,z,v)} \\
&\quad \underbrace{\hspace{15em}}_{\vec{H}(t,g,z,v)}
\end{aligned} \tag{46}$$

The separability is of great advantage for the differentiation, e.g. ($s = 2$):

$$\begin{aligned}
\frac{\partial \vec{h}(\vec{x})}{\partial g} &= \frac{1}{W} \sum_{i=1}^4 \sum_{j=1}^4 \sum_{k=1}^4 \sum_{l=1}^4 \vec{f}_{ijkl} \bar{L}_i(t) \frac{d\bar{L}_j(g)}{dg} \bar{L}_k(z) \bar{L}_l(v) = \\
&= \frac{1}{W} \sum_{i=1}^4 \bar{L}_i(t) \underbrace{\sum_{j=1}^4 \bar{L}'_j(g) \sum_{k=1}^4 \bar{L}_k(z) \underbrace{\sum_{l=1}^4 \bar{L}_l(v) \vec{f}_{ijkl}}_{\vec{H}_{ijk}(v)}}_{\vec{H}_{ij}(z,v)} \\
&\quad \underbrace{\hspace{10em}}_{\vec{H}_i^{(g)}(g,z,v)} \\
&\quad \underbrace{\hspace{15em}}_{\vec{H}^{(g)}(t,g,z,v)}
\end{aligned} \tag{47}$$

with

$$w'_L(x) = -1 \tag{48}$$

$$w'_R(x) = +1$$

$$L'_{ijk}(x) = \frac{\Delta_{0j} + \Delta_{0k}}{\Delta_{ij} \cdot \Delta_{ik}} \tag{49}$$

$$\bar{L}'_1(x) = w_L(x) \cdot L'_{123}(x) \quad - L_{123}(x) \tag{50}$$

$$\begin{aligned}
\bar{L}'_2(x) &= w_L(x) \cdot L'_{231}(x) + w_R(x) \cdot L'_{234}(x) - L_{231}(x) + L_{234}(x) \\
\bar{L}'_3(x) &= w_L(x) \cdot L'_{312}(x) + w_R(x) \cdot L'_{342}(x) - L_{312}(x) + L_{342}(x) \\
\bar{L}'_4(x) &= w_R(x) \cdot L'_{423}(x) + L_{423}(x)
\end{aligned}$$

Derivatives of order higher than one (which are not continuous at the grid nodes!) and mixed derivatives can also be computed easily in this way.

If $x_{(s)0} = x \in [x_1, x_2]$ or $x \in [x_{N-1}, x_N]$ (no two neighbors on each side) then simple quadratic interpolation is applied: $p_Q(x; x_1, x_2, x_3)$ or $p_Q(x; x_{N-2}, x_{N-1}, x_N)$.

For grids with fewer or more (numerical) parameters/dimensions $x_{(s)}$ the interpolation works analogously.¹⁶

Discussion and outlook

Improvement of convergence

The rate of success of the grid gap interpolation procedure described in the previous section depends on the location in the parameter space. By success we mean that newly converged models are obtained when using the interpolated ones as starting models. From Figs. 6 to 10 one can see that the procedure removed most of the isolated gaps for parameters where most of the surrounding models are converged (i.e. with high neighbor quality). It was also successful for a part of the transition region in the HR diagram where convection switches from thin superadiabatic zones as in A stars to deep quasi-adiabatic zones as in solar type stars. This region extends along a line from around 5500 K at $\log g = 2$ to around 7500 K for $\log g = 5$ and shifts to slightly higher temperatures for increasing metallicity. For the part of this region with high $\log g$, the interpolation presumably produces models which are close enough to the 'correct' structure so that the subsequent model calculation is safe from temporarily converging to a wrong solution. The gap remaining at the low $\log g$ end of this region (for all metallicities considered here) indicates that the interpolation is not accurate enough to overcome this 'transition problem' there.

Interpolation is also ineffective in the corner of our parameter space characterized by low metallicity, low temperature and high $\log g$, where the temperature correction algorithm applied in ATLAS9 cannot cope with adiabatic convection in the atmosphere. The third region where models remain missing after the interpolation concerns models with 288 layers and $[M/H] = +1.0$. The region is confined to temperatures between 8400 and 8800 K and $\log g$

¹⁶The number of neighbors will not change: $i = 1, \dots, 4$, but the number of dimensions/'directions': $s = 1, \dots, D$.

larger than 3.6. For those models the flux derivative errors high in the atmosphere ($\log \tau_{\text{Ross}} \lesssim -5$) do not decrease below 10% and continuously increase to about 40% in the outermost layers. The flux errors and temperature corrections are very small (lower than the primary convergence criteria). The same phenomenon occurs in the models with 72 layers, but is confined to even higher layers ($\log \tau_{\text{Ross}} \lesssim -6$) and therefore they have been classified as converged according to the secondary criteria. We cannot give a definitive explanation for this problem at the moment.

For a part of the models in the grids the gas pressure decreases to values below 10^{-2} dyn cm $^{-2}$ for the outermost layers. These values are outside of the range of the ODF tables by Kurucz (1993a), and ATLAS9 uses the constant value corresponding to the limiting pressure instead. For models with $[M/H] = -2.0$ this is the case for a nearly triangular region with $T_{\text{eff}} \geq 6500$ K and $\log g \leq 4.2$, where at these limits only the uppermost layer is affected, while the extension of this zone increases gradually towards the inner layers when moving towards the ‘corner’ of our parameter space. In the most extreme case ($T_{\text{eff}}, \log g$) = (10 000, 2.2) it extends to $\log \tau_{\text{Ross}} \lesssim -4.5$ and the pressure at the outermost layer is $\approx 10^{-4.5}$ dyn cm $^{-2}$. The affected region is smaller for higher metallicities. This inaccurate description of the opacity at high layers nevertheless does not seem to present a problem for the convergence of these models, as the temperature corrections, flux and flux derivative errors do not show a correlated behavior.¹⁷ The only exception might be the $[M/H] = -2$ subgrids where the hottest models show increasingly larger temperature corrections for low $\log g$ (but small flux and flux derivative errors).

Interpolation routines

At present the interpolation routines work only with the NEMO grid and ATLAS9 atmospheres with 72 or 288 layers. Future versions are intended to deal with a variety of different grid structures and resolutions in a flexible manner.

Acknowledgments. This work is supported by the Austrian Fonds zur Förderung der wissenschaftlichen Forschung (FWF) within the project Stellar Atmospheres and Pulsating Stars (P14984), by the Bundesministerium für Bildung, Wissenschaft und Kultur (BMBWK) and the Bundesministerium für Verkehr, Innovation und Technologie (BMVIT) via the Austrian Space Agency (ASA).

Many thanks to Barry Smalley for his numerous advices and corrections!

¹⁷See the figures at the NEMO website http://ams.astro.univie.ac.at/nemo/dvd/??????/vmicro_?/, where ‘??????’ is one of ‘cgm072’, ‘cgm288’, ‘mlt072’, or ‘cm_288’, and ‘?’ is one of 0,1,2,4; files ‘info_p.ps’ and ‘info_m.ps’.

References

- Arp, H. 1961, A.J. 133, 874
- Ažusienis, A., Straizys, V. 1969, Soviet Astron. 13, 316
- Barban, C., Goupil, M. J., van't Veer-Menneret, C., Garrido, R., Kupka, F., Heiter, U. 2003, A&A 405, 1095
- Buser, R. 1978, A&A 62, 411
- Buser, R., Kurucz, R. L. 1978, A&A 70, 555
- Cousins, A. W. J. 1976, Mem. R. Astron. Soc. 81, 25
- Cramer, N., Maeder, A. 1979, A&A 78, 395
- Crawford, D. L., Barnes, J. V. 1970, A.J. 75, 978
- D'Antona, F., Montalbán, J., Kupka, F., Heiter, U. 2002, ApJ 564, L93
- D'Antona, F., Montalbán, J. 2003, A&A 412, 213
- Garrido, R., Garcia-Lobo, E., Rodriguez, E. 1990, A&A 234, 262
- Garrido, R., Claret, A., Moya, A., Kupka, F., Heiter, U., Barban, C., Goupil, M. J., van't Veer-Menneret, C. 2001, in Proc. 1st Eddington workshop *Stellar Structure and Habitable Planet Finding*, ed. F. Favata, I.W. Roxburgh and D. Galadi (ESA SP-485, Noordwijk: ESA Publications Division), 103
- Garrido, R., Moya, A., Goupil, M. J., Barban, C., van't Veer-Menneret, C., Kupka, F., Heiter, U. 2002, CoAst 141, 48
- Hauck, B., Mermilliod, M. 1980, A&AS 40, 1
- Heiter, U., van't Veer-Menneret, C., Barban, C., Weiss, W. W., Goupil, M. J., Schmidt W., Katz, D., Garrido, R. 2002, A&A 392, 619
- Künzli, M., North, P., Kurucz, R. L., Nicolet, B. 1997, A&AS 122, 51
- Johnson, H. L., Mitchell, R. I. 1975, Rev. Mex. Astron. Astrofis. 1, 299
- Kupka F., Paunzen E., Maitzen H. M. 2003, MNRAS 341, 849-854
- Kurucz, R. L. 1993a, *Opacities for Stellar Atmospheres*, Kurucz CD-ROMs No. 2-6, Cambridge, Mass.: Smithsonian Astrophysical Observatory
- Kurucz, R. L. 1993b, *ATLAS9 Stellar Atmosphere Programs and 2 km/s grid*, Kurucz CD-ROM No. 13, Cambridge, Mass.: Smithsonian Astrophysical Observatory
- Kurucz, R. L. 1998, <http://kurucz.harvard.edu/>
- Lamla, E. 1982, in Landolt-Börnstein New Series, vol. VI 2b, ed. K. Schaifers and H. H. Voigt (Berlin: Springer), p.35-90
- Lub, J., Pel, J. W. 1977, A&A 54, 137
- Maitzen, H. M., Vogt, N. 1983 A&A 123, 48
- Matthews, Th. A., Sandage, A. R. 1963, A.J. 138, 30
- Montalbán, J., D'Antona, F., Kupka, F., Heiter, U. 2004, A&A 416, 1081
- Nendwich, J., Nesvacil, N., Heiter, U., Kupka, F. 2003, in IAU Symp. 210, *Modelling of Stellar Atmospheres*, ed. N.E. Piskunov, W.W. Weiss and D.F. Gray (San Francisco: ASP), p. 423
- North, P. private communication with F. Kupka, 1997
- Rufener, F., Nicolet, B. 1988, A&A 206, 357
- Schuler, S. C., King, J. R., Hobbs, L. M., Pinsonneault, M. H. 2004, ApJ 602, L117
- Smalley, B., Gardiner, R. B., Kupka, F., Bessell, M. S. 2002, A&A 395, 601

- Smalley, B., Kupka, F. 2003, in IAU Symp. 210, *Modelling of Stellar Atmospheres*, ed. N.E. Piskunov, W.W. Weiss and D.F. Gray (San Francisco: ASP), p. 425
- Stassun, K. G., Mathieu, R. D., Vaz, L. P. R., Stroud, N., Vrba, F. J. 2004, *ApJS* 151, 357
- Strömgren, B. 1966, *Ann. Rev. Astron. Astrophys.* 4, 433