Interpolation of Stellar Model Grids and Application to the NEMO Grid

J. Nendwich¹, U. Heiter², F. Kupka³, N. Nesvacil⁴, 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 (\(T_{\text{eff}}\), \(\log g\)) 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 \textsc{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 \textsc{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 \textsc{ATLAS9} are summarized in H02. As is shown there as well, the convective flux can be neglected for temperatures higher than 8 600 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.\footnote{The 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.}

For all models, the opacity distribution functions calculated by Kurucz (1993a) were used.
Table 1: Parameter range of the grid

| Parameter | From | To | Increment (List of Values) | Unit | # |
|-----------|------|----|----------------------------|------|---|
| $T_{\text{eff}}$ | 4,000 | 10,000 | 8,600$^a$ | 200 K | 31 24$^a$ |
| $\log g$ | 2.0 | 5.0 | 0.2 | $\{\text{cgs}\}^b$ | 16 |
| $[M/H]$ | $-2.0$ | $+1.0$ | $(-2.0, -1.5, \pm 1.0, \pm 0.5, \pm 0.3, \pm 0.2, \pm 0.1, 0.0)$ | dex | 13 |
| $v_{\text{micro}}$ | 0 | 4 | (0, 1, 2, 4) | $\text{km s}^{-1}$ | 4 |
| Convect. /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 |

$^a$ 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: $eta$ Leo ($T_{\text{eff}}=8850$, $\log g=4.16$, $[M/H]=0.0$, $v_{\text{micro}}=2$) for Walraven VBLUW, HD 83373 (9250, 4.00, +0.0, 2) for the $hk194$ index of Strömgren uvbyCa, a B star (30000, 4.00, +0.0, 2) for Vilnius, and Vega (9550, 3.95, −0.5, 2) for Cousins (Cape) VRI, $\Delta a$, 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 Kurucz programs, which are sometimes convoluted with other response functions (atmosphere, optics, receiver).
Interpolation of Stellar Model Grids and Application to the NEMO Grid

H-Beta (Vega)

Kurucz 1993b, 1998 (/programs/COLORS/)

| $H_{\beta}$ |
|------------|
| 2.903      |
| 2.695      |

Using additional response functions ‘OneP21’, ‘Air’, and ‘Reflct’. Output ”468nm.dat” are not correct $H_{\beta}$ 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 ‘Reflct’.

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 ‘Reflct’.

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    | 0.194|
| 0.802  | 0.780   | 0.500  | −0.070 | −0.058 | −0.080   | −0.133|

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 ‘Reflct’.

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 $\lambda$ to give accurate results for $H_{\beta}$. So the color output for this system is called ”468nm.dat” instead of ”beta.dat”.

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

**Walraven (β Leo)**

|       | V−B | B−U | U−W | B−L |
|-------|-----|-----|-----|-----|
|       | 0.034 | 0.436 | 0.108 | 0.198 |
| λ = 372 nm: | 1.361 | 1.232 | 0.106 | 0.460 |
| λ = 462 nm: | b_1(K93b) = 0.468, b_1(L82) = 0.488 |

Model for β Leo: \((T_{\text{eff}}, \log g, [M/H], v_{\text{micro}}) = (8850, 4.16, +0.0, 2)\)

**Geneva (Vega)**

|       | 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:

**UBV (USA | Vilnius | Buser) (Vega)**

Arp (1961), Ažusienis & Stražys (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)**

|       | 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 | 0.392 | 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
Interpolation of Stellar Model Grids and Application to the NEMO Grid

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)

|   | 33–52 | 35–52 | 37–52 | 40–52 | 45–52 | 52–58 | 52–58 | 52–58 | 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(J93b) = 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)=v0(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 \(\Delta\alpha\)), 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. 11 - 15). For the single parameters we find difficult regions for low \(T_{\text{eff}}\) 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); \(v_{\text{micro}}\) 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. 16 - 21). 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. 23). 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—$T_{\text{eff}}$, log $g$, $[M/H]$, and $v_{\text{micro}}$ (the convection model is non-numerical)—a 4D interpolation is necessary.
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 millimags, 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 \( T_{\text{eff}} \) 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
J. Nendwich, U. Heiter, F. Kupka, N. Nesvacil and W.W. Weiss

Distance Weighted Parabolas for Interpolation

Fig. 17 (p. 22) 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 $(x_i, f_i), i = 1, \ldots, 4$, where

$$f_i = f(x_i)$$

(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:

$$p_L(x_1) = f_1 = p_C(x_1)$$

$$p_L(x_2) = f_2 = p_C(x_2) = p_R(x_2)$$

(2)

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

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 \((=583/16)\).

\[
p_L(x_3) = f_3 = p_C(x_3) = p_R(x_3)
\]

\[
f_4 = p_C(x_4) = p_R(x_4)
\]

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

\[
h(x) = \frac{w_L(x) \cdot p_L(x) + w_R(x) \cdot p_R(x)}{w_L + w_R} = \frac{(x_3 - x) \cdot p_L(x) + (x - x_2) \cdot p_R(x)}{x_3 - x_2} = \frac{\Delta_{30} \cdot p_L(x) + \Delta_{02} \cdot p_R(x)}{\Delta_{32}}
\]

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 \( \Delta_{ij} \) is given in Eq. 8 on page 15.)
It holds by (2) that
\[ h(x_2) = p_L(x_2) = f_2 \quad (= p_R(x_2) = p_C(x_2)) \quad (4) \]
\[ h(x_3) = p_R(x_3) = f_3 \quad (= p_L(x_3) = p_C(x_3)) \]

For the first derivative we obtain
\[ h'(x) = \frac{-1 \cdot p_L(x) + (x_3 - x) \cdot p'_L(x) + 1 \cdot p_R(x) + (x - x_2) \cdot p'_R(x)}{x_3 - x_2} = \]
\[ = \frac{w_L(x) \cdot p'_L(x) + w_R(x) \cdot p'_R(x)}{w_L + w_R} + \frac{[p_R - p_L](x)}{x_3 - x_2} \quad (5) \]

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
\[ h'(x_2) = p'_L(x_2) \quad (6) \]
\[ h'(x_3) = p'_R(x_3) \]
Interpolation of Stellar Model Grids and Application to the NEMO Grid

Figure 9: Number of missing models versus $v_{\text{micro}}$ (added up for the other four parameters) before and after the interpolation. The two horizontal lines indicate the average values per $v_{\text{micro}}$ 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 $\tilde{p}_L(x) = \tilde{p}_R(x)$ 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

$$
\begin{align*}
\tilde{h}'(x_3) &= \tilde{\tilde{p}}'_L(x_3) = \tilde{\tilde{p}}'_R(x_3) \\
\tilde{h}'(x_4) &= \tilde{\tilde{p}}'_R(x_4)
\end{align*}
$$

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”. 

Using the definitions\(^3\)

\[
\Delta_{0j}(x) = x - x_j \\
\Delta_{ij} = x_i - x_j = -\Delta_{ji} = \Delta_{ik} + \Delta_{kj} = \Delta_{0j}(x_i)
\]

and Lagrange polynomials \((i,j,k = 1, \ldots , 3 \text{ or } 2, \ldots , 4)\).\(^4\)

\[
L_{i,j,k}(x) = \frac{\Delta_{0j} \Delta_{0k}}{\Delta_{ij} \Delta_{ik}} \cdot \frac{x - x_j}{x_i - x_j} \cdot \frac{x - x_k}{x_i - x_k} = L_{ikj}(x)
\]

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

\[
pl(x) = f_1 L_{123}(x) + f_2 L_{231}(x) + f_3 L_{312}(x)
\]

\[
pl(x) = f_2 L_{234}(x) + f_3 L_{342}(x) + f_4 L_{423}(x)
\]

\(^3\)We 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 \(i = 0\).

\(^4\)\(L_{ij,k}(x) = (\Delta_{0j}/\Delta_{ij}) \cdot (\Delta_{0k}/\Delta_{ik}) \cdot (\Delta_{0l}/\Delta_{il})\).

\(p_C(x) = f_1 L_{1234}(x) + f_2 L_{2341}(x) + f_3 L_{3412}(x) + f_4 L_{4123}(x)\)
Figure 11: Number of missing models versus $T_{\text{eff}}$ 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

\[
\tilde{h}(x) = \frac{1}{\Delta_{32}} \left\{ \tilde{f}_1 \underbrace{w_L(x)L_{123}(x)}_{L_1(x)} + \tilde{f}_4 \underbrace{w_R(x)L_{423}(x)}_{L_4(x)} + \right. \\
+ \tilde{f}_2 \underbrace{[w_L(x)L_{231}(x) + w_R(x)L_{234}(x)]}_{L_2(x)} + \\
+ \tilde{f}_3 \underbrace{[w_L(x)L_{312}(x) + w_R(x)L_{342}(x)]}_{L_3(x)} \right\} = \\
= \frac{1}{\Delta_{32}} \left\{ \frac{1}{\Delta_{42}} \left[ \tilde{f}_2 \Delta_{40} \left( \frac{\Delta_{30}\Delta_{01}}{\Delta_{21}} + \frac{\Delta_{40}\Delta_{02}}{\Delta_{42}} \right) + \\
+ \tilde{f}_3 \Delta_{02} \left( \frac{\Delta_{30}\Delta_{01}}{\Delta_{31}} + \frac{\Delta_{40}\Delta_{02}}{\Delta_{43}} \right) \right] - \right. \\
+ \tilde{f}_3 \Delta_{02} \left( \frac{\Delta_{30}\Delta_{01}}{\Delta_{31}} + \frac{\Delta_{40}\Delta_{02}}{\Delta_{43}} \right) \right\} 
\]
Figure 12: Number of missing models versus $T_{\text{eff}}$ 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.

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

where we have taken into account that we also want to interpolate vector functions in the model grid, because every $\hat{f}_i$ represents a model atmosphere with up to $6 \cdot 288 = 1728$ numerical values. Therefore, the formula is optimized for computational performance\(^5\) such that each $\hat{f}_i$ 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

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

\(^5\)Fortran90 array arithmetic
Figure 13: Number of missing models versus $T_{\text{eff}}$ 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{align}
  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 resolution}
\end{align}

Note, that $c = x(5)$ is a non-numerical parameter of the grid and hence cannot 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 $x(s)$, $s = 1, \ldots, 4$, 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 values\(^6\) are:
Figure 14: Number of missing models versus $T_{\text{eff}}$ 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 $s = 1$: $x(s) = x_1 = t$ \(t_1 < t_2 \leq t_0 \leq t_3 < t_4\):

\[
\text{left} : \quad \vec{x}_1 = (t_1, g_0, z_0, v_0), \quad \vec{x}_2 = (t_2, g_0, z_0, v_0) \\
\text{right} : \quad \vec{x}_3 = (t_3, g_0, z_0, v_0), \quad \vec{x}_4 = (t_4, g_0, z_0, v_0) \\
\vec{f}_i(1) = \vec{f}(\vec{x}_i(1)) = \vec{f}(t_i, g_0, z_0, v_0)
\]

for $s = 2$: $x(s) = x_2 = g$ \(g_1 < g_2 \leq g_0 \leq g_3 < g_4\):

\[
\text{left} : \quad \vec{x}_1 = (t_0, g_1, z_0, v_0), \quad \vec{x}_2 = (t_0, g_2, z_0, v_0) \\
\text{right} : \quad \vec{x}_3 = (t_0, g_3, z_0, v_0), \quad \vec{x}_4 = (t_0, g_4, z_0, v_0) \\
\vec{f}_i(2) = \vec{f}(\vec{x}_i(2)) = \vec{f}(t_0, g_i, z_0, v_0)
\]

for $s = 3$: $x(s) = x_3 = z$ \(z_1 < z_2 \leq z_0 \leq z_3 < z_4\):

\[
\text{left} : \quad \vec{x}_1 = (t_0, g_0, z_1, v_0), \quad \vec{x}_2 = (t_0, g_0, z_2, v_0)\\
\text{right} : \quad \vec{x}_3 = (t_0, g_0, z_3, v_0), \quad \vec{x}_4 = (t_0, g_0, z_4, v_0)
\]

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.

\[
\text{right: } \vec{x}_{3(3)} = (t_0, g_0, z_3, v_0), \quad \vec{x}_{4(3)} = (t_0, g_0, z_4, v_0)
\]
\[
\vec{f}_{i(3)} = \vec{f}(\vec{x}_{i(3)}) = \vec{f}(t_0, g_0, z_i, v_0)
\]

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

\[
\text{left: } \quad \vec{x}_{1(4)} = (t_0, g_0, z_0, v_1), \quad \vec{x}_{2(4)} = (t_0, g_0, z_0, v_2)
\]
\[
\text{right: } \quad \vec{x}_{3(4)} = (t_0, g_0, z_0, v_3), \quad \vec{x}_{4(4)} = (t_0, g_0, z_0, v_4)
\]
\[
\vec{f}_{i(4)} = \vec{f}(\vec{x}_{i(4)}) = \vec{f}(t_0, g_0, z_0, v_i)
\]

Subsuming the third lines we can write\(^7\)
\[
\vec{f}_{i(s)} = \vec{f}(\vec{x}_{i(s)})
\]

\(^7\)One remark about notation: \( \vec{x} = (x_{(1)}, x_{(2)}, x_{(3)}, x_{(4)}, x_{(5)}) = (t, g, z, v[t, c]) \), \( \vec{x}_0 = (x_{(1)0}, x_{(2)0}, x_{(3)0}, x_{(4)0}, x_{(5)0}) = (t_0, g_0, z_0, v_0[0, c_0]) \); e.g. \( s = 3 \): \( \vec{x}_{i(3)} = (t_{i0}, g_{i0}, z_{i0}, v_{i0}) = (x_{(1)i0}, x_{(2)i0}, x_{(3)i0}, x_{(4)i0}, x_{(5)i0}) \). I.e. \( x_{(s)} \) is a component of \( \vec{x}_{i(s)} \). Thus, we could define \( m_{i(s)}(\vec{x}_{i(s)}) = m_{s(i)} \). (Eqs. 21, 26.)
With Eq. 11 (p. 16, $\Delta_{ij} \rightarrow \Delta_{ij(s)}$) the 1D interpolation for $x_i(s)$ at $\vec{x}_0$ can be done:

$$\vec{h}_{ij}(\vec{x}_0) = \vec{h}_{ij}(t_0, g_0, z_0, v_0) = \frac{1}{\Delta_{32(s)}} \sum_{i=1}^{4} \tilde{f}_{ij}(s) L_i(x_i(s)0)$$ (20)

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

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

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

where $m_{x_i(s)}$ is the grid position number of $x_i(s)$ (number corresponding to $x_{i(i)}$ when counting every grid step from the minimum to the maximum value).
Interpolation of Stellar Model Grids and Application to the NEMO Grid

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 \( \vec{x}_{(s)i} \)). 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:

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

\(^8\)E.g. for \( s = 1, x_{(s)} = x_{(1)} = t_i = T_{\text{eff},i} = 4\,000; m_{x_{(s)i}} = 1 \) (2 for 4,200, 3 for 4,400, ..., 31 for 10,000). Note, that we are considering interpolation at grid nodes (missing models) and consequently \( m_{x_{(s)i}} \) is also defined uniquely.
Table 5: Numbers and percentages of models fulfilling the primary and secondary SMGT convergence criteria before and after the interpolation. Due to rounding the percentages may not sum up to 100.00.

| Interpolation                  | Number of models | Percentage   |
|--------------------------------|------------------|--------------|
|                                | before | after | ∆   | before | after | ∆   |
| 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   |

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

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

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

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

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

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

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

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

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

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

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

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

\(^9\)That 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 \(v\).


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

\(^{10}\) Alternatively, 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.

\(^{11}\) The 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.
Interpolation of Stellar Model Grids and Application to the NEMO Grid

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 need 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 grid\(^{12}\) 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

\(^{12}\)Converged files contain different values compared to the interpolated ones taken as their SMGT 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, 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 ($I_{p2}$)

The $I_{p2}$ is real 4D, it expects a complete $4^4 (256)$ neighborhood of grid nodes (existing model files, converged or interpolated)

$$\tilde{f}_{ijkl}(x) = \tilde{f}(x_{ijkl}) = \tilde{f}(x_{(1)i}, x_{(2)j}, x_{(3)k}, x_{(4)l}) = \tilde{f}(t, g, z, v)$$

where $i, j, k, l = 1, \ldots, 4$ around the interpolation point $\tilde{x}_0 = (t_0, g_0, z_0, v_0)$ which now is not a grid node but still we assume $x_{(s)1} < x_{(s)2} < x_{(s)0} < x_{(s)4}$ for $s = 1, \ldots, 4$, 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. 15)

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

16 ($2^4$) left-right combined parabolas can be defined, e.g. (cf. Eq. 9, p. 15):

$$p_{LRRRL}(x) = f_{1221} \cdot L_{1234}(t) \cdot L_{234}(g) \cdot L_{234}(z) \cdot L_{123}(v) +$$
$$+ f_{1222} \cdot L_{1234}(t) \cdot L_{234}(g) \cdot L_{234}(z) \cdot L_{231}(v) +$$
$$+ f_{1223} \cdot L_{1234}(t) \cdot L_{234}(g) \cdot L_{234}(z) \cdot L_{312}(v) +$$
$$+ f_{1231} \cdot L_{1234}(t) \cdot L_{234}(g) \cdot L_{342}(z) \cdot L_{123}(v) +$$

$^{14}$The base 4 is due to $i = 1, \ldots, 4$ (4 neighbors), the exponent 4 is due to $s = 1, \ldots, 4$ (4D).
\[ \begin{align*}
+ \ & f_{1232} \cdot L_{123}(t) \cdot L_{234}(g) \cdot L_{342}(z) \cdot L_{231}(v) + \\
+ \ & f_{1233} \cdot L_{123}(t) \cdot L_{234}(g) \cdot L_{342}(z) \cdot L_{312}(v) + \\
+ \ & f_{1241} \cdot L_{123}(t) \cdot L_{234}(g) \cdot L_{423}(z) \cdot L_{123}(v) + \\
+ \ & f_{1242} \cdot L_{123}(t) \cdot L_{234}(g) \cdot L_{423}(z) \cdot L_{231}(v) + \\
+ \ & f_{1243} \cdot L_{123}(t) \cdot L_{234}(g) \cdot L_{423}(z) \cdot L_{312}(v) + \\
+ \ & f_{3441} \cdot L_{312}(t) \cdot L_{423}(g) \cdot L_{423}(z) \cdot L_{123}(v) + \\
+ \ & f_{3442} \cdot L_{312}(t) \cdot L_{423}(g) \cdot L_{423}(z) \cdot L_{231}(v) + \\
+ \ & f_{3443} \cdot L_{312}(t) \cdot L_{423}(g) \cdot L_{423}(z) \cdot L_{312}(v) = \\
= \ & L_{123}(t) \times \left\{ L_{234}(g) \times \left[ L_{234}(z) \times \\
& \times \left( L_{123}(v) \cdot f_{1221} + L_{231}(v) \cdot f_{1222} + L_{312}(v) \cdot f_{1223} \right) + \\
& + L_{342}(z) \times \\
& \times \left( L_{123}(v) \cdot f_{1231} + L_{231}(v) \cdot f_{1232} + L_{312}(v) \cdot f_{1233} \right) + \\
& + L_{423}(z) \times \\
& \times \left( L_{123}(v) \cdot f_{1241} + L_{231}(v) \cdot f_{1242} + L_{312}(v) \cdot f_{1243} \right) + \\
& + L_{342}(g) \times \left[ L_{234}(z) \times \\
& \times \left( L_{123}(v) \cdot f_{1321} + L_{231}(v) \cdot f_{1322} + L_{312}(v) \cdot f_{1323} \right) + \\
& + L_{342}(z) \times \\
& \times \left( L_{123}(v) \cdot f_{1331} + L_{231}(v) \cdot f_{1332} + L_{312}(v) \cdot f_{1333} \right) + \\
& + L_{423}(z) \times \\
& \times \left( L_{123}(v) \cdot f_{1341} + L_{231}(v) \cdot f_{1342} + L_{312}(v) \cdot f_{1343} \right) \right) \right\} + \\
+ \ & L_{312}(t) \times \left\{ L_{423}(g) \times \left[ L_{234}(z) \times \\
& \times \left( L_{123}(v) \cdot f_{1441} + L_{231}(v) \cdot f_{1442} + L_{312}(v) \cdot f_{1443} \right) \right] \right\} + \\
+ \ & \ldots
\end{align*} \]
The general expressions are given by

\[ \Delta^{(s)} \] in case of a right parabola (in \( R = 1 \)), and analogously for \( R = 1 \).

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

\[ w_{LRRL}(\vec{x}) = \Delta_{30(1)} \cdot \Delta_{02(2)} \cdot \Delta_{02(3)} \cdot \Delta_{30(4)} = \]

\[ = w_L(t) \cdot w_R(g) \cdot w_L(z) \cdot w_L(v) = \]

\[ = (t_3 - t)(g - g_2)(z - z_2)(v_3 - v) \]

The general expressions are given by

\[ p_{ijkl}^{\vec{x}}(\vec{x}) = \]

\[ = \sum_{i=1}^{3+i} \sum_{j=1}^{3+j} \sum_{k=1}^{3+k} \sum_{l=1}^{3+l} f_{ijkl}^t \times \]

\[ \prod_{i'=1}^{3+i} \prod_{j'=1}^{3+j} \prod_{k'=1}^{3+k} \prod_{l'=1}^{3+l} \frac{\Delta_{0i'j'(1)}}{\Delta_{ii'j'(1)}} \frac{\Delta_{0j'k'(2)}}{\Delta_{jj'k'(2)}} \frac{\Delta_{0k'l'(3)}}{\Delta_{kk'l'(3)}} \frac{\Delta_{0l'i'(4)}}{\Delta_{ll'i'(4)}} = \]

\[ = \sum_{i=1}^{3+i} L_i^{t_1} \cdot \sum_{j=1}^{3+j} L_j^{t_2} \cdot \sum_{k=1}^{3+k} L_k^{t_3} \cdot \sum_{l=1}^{3+l} L_l^{t_4} \times \]

\[ \left( \sum_{i'=1}^{3+i} L_i^{t_1} \cdot \sum_{j'=1}^{3+j} L_j^{t_2} \cdot \sum_{k'=1}^{3+k} L_k^{t_3} \cdot \sum_{l'=1}^{3+l} L_l^{t_4} \right) \]

\[ w_{ii'j'k'}(\vec{x}) = w_i^t(t) \cdot w_j^g(g) \cdot w_k^z(z) \cdot w_l^v(v). \]

where \( i \) 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 \( j \) \( x_2 = g \), \( k \) \( x_3 = z \), and \( l \) \( x_4 = v \), and

\[ w_L(x_1) = \Delta_{02(s)} \]

\[ w_R(x_2) = \Delta_{02(s)} \]

\[ W = \sum_{i=L,R} \sum_{j=L,R} \sum_{k=L,R} \sum_{l=L,R} w_{ijkl}(\vec{x}) = \]

\[ = \prod_{s=1}^{4} \Delta_{32(s)} = (t_3 - t_2)(g_3 - g_2)(z_3 - z_2)(v_3 - v_2) \]
For the interpolating function we obtain (cf. Eq. 3, p. 12)

\[
\vec{h}(\vec{x}) = \frac{1}{W} \sum_{i=0}^{3} \sum_{j=0}^{3} \sum_{k=0}^{3} \sum_{l=0}^{3} w_{ijkl}(\vec{x}) \cdot p_{ijkl}(\vec{x}) = \]

\[
= \frac{1}{W} \sum_{i=0}^{3} w_i(t) \sum_{j=0}^{3} w_j(g) \sum_{k=0}^{3} w_k(z) \sum_{l=0}^{3} w_l(v) \cdot p_{ijkl}(t, g, z, v) = \]

\[
= \frac{1}{W} \sum_{i=0}^{3} w_i(t) \sum_{j=0}^{3} w_j(g) \sum_{k=0}^{3} w_k(z) \sum_{l=0}^{3} w_l(v) \times \]

\[
\times \sum_{i=1+1}^{3+1} L_{ii'j'k'}(t) \sum_{j=1+1}^{3+1} L_{jj'k'k'}(g) \sum_{k=1+1}^{3+1} L_{kk'k'k'}(z) \sum_{l=1+1}^{3+1} L_{ll'k'k'}(v) \times \]

\[
\times \vec{f}_{ijkl} = \]

\[
= \frac{1}{W} \sum_{i=0}^{3} w_i(t) \sum_{i=1+1}^{3+1} L_{ii'j'k'}(t) \cdot \sum_{j=0}^{3} w_j(g) \sum_{j=1+1}^{3+1} L_{jj'k'k'}(g) \times \]

\[
\times \sum_{k=0}^{3} w_k(z) \sum_{k=1+1}^{3+1} L_{kk'k'k'}(z) \sum_{l=0}^{3} w_l(v) \sum_{l=1+1}^{3+1} L_{ll'k'k'}(v) \cdot \vec{f}_{ijkl} \]

\[
\vec{H}_{ijkl}(v) \]

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

\[
\vec{L}_1(x) = w_L(x) \cdot L_{123}(x) \]

\[
\vec{L}_2(x) = w_L(x) \cdot L_{231}(x) + w_R(x) \cdot L_{234}(x) \]

\[
\vec{L}_3(x) = w_L(x) \cdot L_{312}(x) + w_R(x) \cdot L_{342}(x) \]

\[
\vec{L}_4(x) = w_R(x) \cdot L_{423}(x) \]

And so

\[
\vec{H}_{ijkl}(v) = \sum_{l=0}^{3} w_l(v) \sum_{l=1+1}^{3+1} L_{ll'k'k'}(v) \cdot \vec{f}_{ijkl} = \]

\[
= \left[ w_0(v) \sum_{l=1}^{3} + w_1(v) \sum_{l=2}^{4} \right] L_{ll'k'k'}(v) \cdot \vec{f}_{ijkl} = \]
\[ \begin{align*}
&= w_L(v) \left\{ L_{123}(v) \vec{f}_{ijk1} + L_{231}(v) \vec{f}_{ijk2} + L_{312}(v) \vec{f}_{ijk3} \right\} + \\
&+ w_R(v) \left\{ L_{234}(v) \vec{f}_{ijk2} + L_{342}(v) \vec{f}_{ijk3} + L_{423}(v) \vec{f}_{ijk4} \right\} = \\
&= \vec{f}_{ijk1} \ w_L(v) L_{123}(v) + \\
&+ \vec{f}_{ijk2} \ \underbrace{w_L(v) L_{231}(v) + w_R(v) L_{234}(v)}_{L_1(v)} + \\
&+ \vec{f}_{ijk3} \ \underbrace{w_L(v) L_{312}(v) + w_R(v) L_{342}(v)}_{L_2(v)} + \\
&+ \vec{f}_{ijk4} \ w_R(v) L_{423}(v) = \\
&= \sum_{l=1}^{4} L_l(v) \cdot \vec{f}_{ijkl} \tag{41}
\end{align*} \]

Analogously\textsuperscript{15}

\[ \vec{H}_{ij}(z, v) = \sum_{k=0}^{1} w_{k}(z) \sum_{k=1+k}^{3+k} L_{kk'k''}(z) \cdot \vec{H}_{ijk}(v) = \tag{42} \]

\[ = \sum_{k=1}^{4} L_k(z) \cdot \vec{H}_{ijk}(v) \]

\[ \vec{H}_{i}(g, z, v) = \sum_{j=0}^{1} w_{j}(g) \sum_{j=1+j}^{3+j} L_{jj'j''}(g) \cdot \vec{H}_{ij}(z, v) = \tag{43} \]

\[ = \sum_{j=1}^{4} L_j(g) \cdot \vec{H}_{ij}(z, v) \]

\[ \vec{H}(t, g, z, v) = \sum_{i=0}^{1} w_{i}(t) \sum_{i=1+i}^{3+i} L_{ii'i''}(t) \cdot \vec{H}_{i}(g, z, v) = \tag{44} \]

\textsuperscript{15}In Fortran90 the (one line) command for $\vec{H}_{ijk}(v) \rightarrow \vec{H}_{ij}(z, v)$ is e.g.: $h_{ij}(\cdot, \cdot, \cdot) = Lz(1) \cdot h_{ijk}(\cdot, \cdot, 1) + Lz(2) \cdot h_{ijk}(\cdot, \cdot, 2) + Lz(3) \cdot h_{ijk}(\cdot, \cdot, 3) + Lz(4) \cdot h_{ijk}(\cdot, \cdot, 4)$, where the first `:' stands for the $z$ of $H_i$, and the second and third for $i$ and $j$, respectively; no DO loops have to be written for this. $k = 1, \ldots, 4$ is written explicitly.
Interpolation of Stellar Model Grids and Application to the NEMO Grid

\[
\vec{H}(\vec{x}) = \sum_{i=1}^{4} L_i(t) \cdot \vec{H}_i(g, z, v)
\]

\[
\vec{h}(\vec{x}) = \frac{1}{W} \vec{H}(\vec{x})
\]

Thus, we finally can write (cf. Eq. 11, p. 16)

\[
\vec{h}(\vec{x}) = \frac{1}{W} \sum_{i=1}^{4} \sum_{j=1}^{4} L_i(t) L_j(g) \sum_{k=1}^{4} \sum_{l=1}^{4} L_k(z) L_l(v) f_{ijkl} \vec{h}_{ijkl}(g, z, v) = \frac{1}{W} \vec{H}(\vec{x})
\]

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

\[
\frac{\partial \vec{h}(\vec{x})}{\partial g} = \frac{1}{W} \sum_{i=1}^{4} \sum_{j=1}^{4} \sum_{k=1}^{4} \sum_{l=1}^{4} \vec{h}_{ijkl}(g, z, v) \frac{\partial L_j(g)}{\partial g} L_k(z) L_l(v) = \frac{1}{W} \vec{H}^{(s)}(g, z, v)
\]

\[
\vec{h}_{ijkl}(g, z, v) = \vec{h}_{ijkl}(g, z, v)
\]

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

\[
\frac{\partial \vec{h}(\vec{x})}{\partial g} = \frac{1}{W} \sum_{i=1}^{4} \sum_{j=1}^{4} \sum_{k=1}^{4} \sum_{l=1}^{4} \vec{h}_{ijkl}(g, z, v) \frac{\partial L_j(g)}{\partial g} L_k(z) L_l(v) = \frac{1}{W} \vec{H}^{(s)}(g, z, v)
\]

with

\[
w_L(x) = -1
\]

\[
w_L(x) = +1
\]

\[
L'_{ijk}(x) = \frac{\Delta_{ij} + \Delta_{ik}}{\Delta_{ij} \cdot \Delta_{ik}}
\]

\[
\vec{L}'_i(x) = w_L(x) \cdot L'_{123}(x) - L_{123}(x)
\]
\[
\begin{align*}
\bar{L}'_2(x) &= w_L(x) \cdot L'_{231}(x) + w_R(x) \cdot L'_{234}(x) - L_{231}(x) + L_{234}(x) \\
\bar{L}'_3(x) &= w_L(x) \cdot L'_{312}(x) + w_R(x) \cdot L'_{342}(x) - L_{312}(x) + L_{342}(x) \\
\bar{L}'_4(x) &= w_R(x) \cdot L'_{423}(x) + L_{423}(x)
\end{align*}
\]

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 5 500 K at \(\log g = 2\) to around 7 500 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 8 400 and 8 800 K and \(\log g\).

\(^{16}\)The number of neighbors will not change: \(i = 1, \ldots, 4\), but the number of dimensions/’directions’: \(s = 1, \ldots, D\).
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 \((T_{\text{eff}}, \log g) = (10000, 2.2)\) 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 in 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!

\(^{17}\) See the figures at the NEMO website http://ams.astro.univie.ac.at/nemodvdf/??????/vmicro\(?????\)/, where ‘??????’ is one of ‘cgm072’, ‘cgm288’, ‘mlt072’, or ‘cm288’, 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. Schäfers 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
Interpolation of Stellar Model Grids and Application to the NEMO Grid

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