APOGEE

TECHNICAL NOTE

Basic tests with Ferre using a Gaussian LSF

Carlos Allende Prieto

Instituto de Astrofísica de Canarias

April 7, 2011

Abstract

Ferre is the f90 code used at the core of the APOGEE Stellar Parameter and Chemical Abundance Pipeline (ASPCAP). The code has been extensively upgraded to 1) run in parallel using OPENMP 2) use Principal Component Analysis (PCA) to compress the library of theoretical spectra, and 3) included additional optimization algorithms. Here we report on results for tests extracting the atmospheric parameters ($T_{\rm eff}$, $\log g$, [Fe/H], [C/Fe], [N/Fe], [α/Fe], and micro-turbulence) for cool ($3500 < T_{\rm eff} < 5000$K) stars assuming a Gaussian line spread function (LSF).

1 Introduction

Ferre works by loading a library of model spectra, and finding the values of the model parameters that lead to the best possible agreement with an observed spectrum. The model evaluations are performed by interpolation in the library. By default the code employs the Nelder-Mead algorithm (Nelder & Mead 1965), but recently several other optimization algorithms have been implemented. The computing time is dominated by the interpolations in the library, which can be held in memory or in a database. The fastest speeds are achieved by using linear interpolation, and holding the library in memory. The size of the library crucially affects speed. PCA compression can be used on the library and very effectively reduces the grid size, typically by up to a factor 10, and accelerates the calculations.

After many tests, we have settled on using Ferre to derive up to seven atmospheric parameters simultaneously: $T_{\rm eff}$, $\log g$, [Fe/H], [C/Fe], [N/Fe], [α/Fe], and microturbulence. The target stars are to be divided in a number of categories or classes, depending mainly on $T_{\rm eff}$, with overlapping temperature ranges. When a star is close to the limits of one class, or there is no secure assignment, its spectrum will be considered in two or more classes, and the finally assigned class will be decided depending on the success of fitting the spectrum for each of the candidate classes. Only after a star has been assigned a set of atmospheric parameters ASPCAP will proceed to determine other chemical abundances from its spectrum.

ASPCAP considers several possibilities for the analysis, with different levels of complexity. In the most simple scenario, a single LSF is adopted for all fibers, and the spectra are resampled to match the array of wavelengths of the model library. In this case, the preprocessing code in ASPCAP homogenizes the observations, and the optimization process proceeds fast. For a more detailed analysis, FERRE will adapt the model spectra, convolving the input library with LSF models customized for each fiber, and resampling the model spectra to match each observation. In this, more involved, situation, the calculations will be slower. All the tests described here refer to the the simple case of a fiber-independent wavelength-independent Gaussian LSF.

Below we describe a number of basic tests on the code performance for a test grid with a size that is similar to what we plan to use for APOGEE spectra. Using seven parameters for cool stars ($3500 < T_{\rm eff} < 5000$K), we have a library with 336,875 nodes (spectra) and about 12,000 frequencies per spectrum, adding up to 27 GB of data (45 GB in an ASCII-formatted file). This grid includes 7 values for $T_{\rm eff}$, 11 for $\log g$, 5 for the micro-turbulence (ξ), 7 for the the overall metallicity, and 5 values for each of the following abundances: α, C, and N. The library used in this test is based on model atmospheres calculated by Castelli & Kurucz (2004), available from Kurucz's web site (ODFNEW models), for which we have calculate H-band spectra using the APOGEE linelist v4 (codename *m201007052154*) and the ASSET spectral synthesis code (Koesterke, Allende Prieto & Lambert 2008; Koesterke 2009). Note that for the spectral synthesis the micro-turbulence and the abundances of C, N and the α elements are changed, while their values remain unchanged in the model atmospheres.

2 Performance

Running Ferre with a library as big as 27 GB requires a machine with at least as much memory, or accessing the library as a direct-access file, which is slower. Running with the library in memory for a modern linux workstation takes of the order of 10 minutes to read the library and then about 3-5 minutes per spectrum. We found that the performance was good only after cranking up the parameters for the Nelder-Mead algorithm to carry out 10 searches per spectrum, with random initial starting points, and optimizing the spectral weights: weighting frequencies according to the derivatives of the spectrum relative to the parameters (Allende Prieto 2004).

Applying PCA to the spectral library reduces effectively the number of frequencies by a factor of ~ 10, and the overall size of the library by a factor ~ 4. In addition, our tests indicate that the performance is similar to working with an uncompressed library but without the need to *optimize* the spectral weights as described above. Overall we find that using PCA compression for a library of this size has a very significant impact on speed, and the time per spectrum is reduced from a few minutes to a few seconds or less.

Fig. 1 shows the performance of the code using the Nelder-Mead algorithm on a subsection of the same library, with Gaussian noise added to have a signal-to-noise (S/N) ratio of 50 per pixel. The testing data correspond to the nodes of the library, except for spectra that are at the edges in one or more of the parameters. The robust scatter for each parameter is derived by excluding the 15% upper and lower tails of the distribution of residuals, so that if a distribution were Gaussian, it would cover from minus to plus σ. In this way, we find uncertainties (see also Table 1) of 0.008 0.02 0.06 0.008 dex in [Fe/H], [C/Fe], [N/Fe], and [α/Fe], respectively, and 0.017 dex, 6.8 K, and 0.026 dex in micro-turbulence, $T_{\rm eff}$, and $\log g$, respectively.

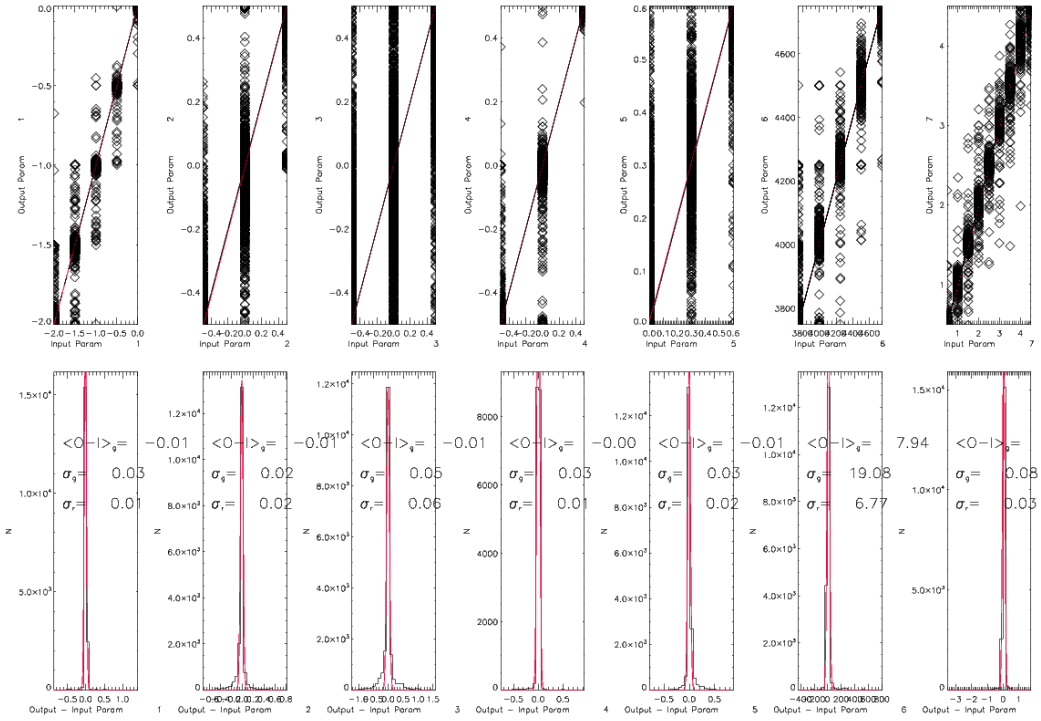

Figure 1: *Upper panels*: Input (truth) and output (derived) parameters. *Lower panels*: distribution of residuals for each parameter; the red lines show Gaussian curves fit to the data; the labels indicate average values and 1-σ uncertainties derived by fitting Gaussians (σ_g) or the robust procedure described in the text (σ_r). Params 1 through 7 are: [Fe/H], [C/Fe], [N/Fe], [α/Fe], micro-turbulence, $T_{\rm eff}$, $\log g$.

Similarly, we have repeated this test using the UOBYQA algorithm (Powell 2002), finding a very similar performance, although this algorithm is several times slower[1]. Since the analyzed spectra are at the grid nodes, interpolation errors are not fairly represented, and these uncertainties are mainly associated with the loss of information due to the degradation by the Gaussian noise added.

It is interesting to examine how much errors increase by analyzing spectra for parameters off the grid nodes. Since we do not have at this point model atmospheres for parameters off the grid nodes, we have simply interpolated linearly the PCA coefficients on the grid. The interpolated spectra were obtained for 18225 cases, with parameters derived at random with a uniform probability over the parameter space, and the data were not degraded with random Gaussian noise, which was the case for the testing spectra on the grid nodes. The results are included in Table 1; a small but significant degradation is visible.

[1]We make use of the FORTRAN implementations by A. Miller available from `http://jblevins.org/mirror/amiller/`

Table 1: Robust uncertainties in the derived atmospheric parameters

Algorithm	σ([Fe/H])	σ([C/Fe])	σ([N/Fe])	σ([α/Fe])	$\sigma(\log \xi)$ (km/s)	$\sigma(T_{\rm eff})$ (K)	$\sigma(\log g)$ (cm/s^2)
Nelder-Mead (N-M)	0.0080	0.0215	0.0610	0.0080	0.0165	6.80	0.0260
UOBYQA	0.0070	0.0190	0.0755	0.0065	0.0140	5.536	0.0225
Nelder-Mead Interpolated	0.0197	0.0517	0.0856	0.0266	0.0354	23.78	0.0566
N-M R=10,000	0.0245	0.0520	0.1825	0.0250	0.0650	18.98	0.0665
N-M R=3,000	0.0725	0.1575	0.4225	0.0735	0.2185	52.02	0.1765

3 Parallelization

Ferre now includes parallelization using OPENMP (within a shared-memory node), for most of the algorithms it uses. We have tested how parallelizing improves the performance using a 2-processor quad-core Intel XEON X5355 2.66GHz GHZ machine with 4MB cache and 33 GB RAM. Fig. 2 compares the ratio of the time for a serial calculation (30-50 minutes for 1000 spectra, depending on the compiler and the compiler options) with that for a parallel calculation with a different number of cpus, i.e. the speedup. The computing time is drastically reduced when using two cpus, with smaller but substantial speedups for 3 and 4 cpus. Beyond that, despite the machine has 8 cores, the system resources saturate and the code begins to run slower. By the time we use the 8 cores in parallel, the running (wall) time for the calculation is similar to that obtained using only one CPU. This study needs to be repeated with the actual hardware where the pipeline will be run.

4 Spectral resolution

To examine the effect of reducing the resolution of the spectra, we repeated the simple test above with Gaussian noise (S/N=50) for resolving powers of $R = 10,000$ and 3,000. The results have been included in Table 1. The degradation in the extracted parameters is modest for 10,000 but significant at 3,000, in particular the distributions of residuals exihibit marked departures from a Normal curve, with extended wings, and therefore the statistics in Table 1 are no longer representative of the scatter at such low resolution.

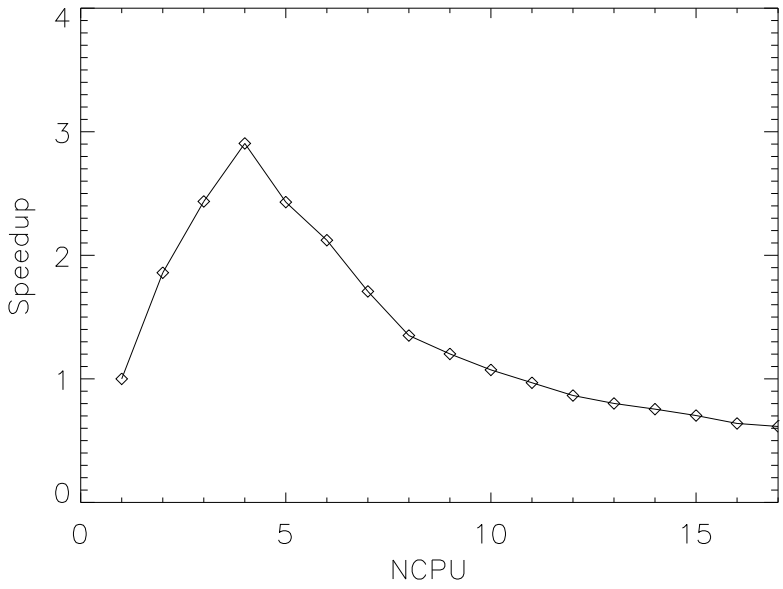

Figure 2: Speedup in the calculations by running Ferre in parallel on a 2 x 4 core Intel machine.

5 Conclusions

We have settled on the basic configuration for running `ferre` to derive atmospheric parameters for cool (3500-5000 K) giants: a 7-parameter grid considering effective temperature ($T_{\rm eff}$), surface gravity ($\log g$), micro-turbulence, overall metallicity ([Fe/H]), carbon ([C/Fe]), nitrogen ([N/Fe]) and α- element ([O Mg Si S Ca Ti,/Fe]) abundances. Our tests considered the spectrum sampled with two frequencies per resolution element and a FWHM resolving power of $\lambda/\delta\lambda = 30,000$, with about 12,000 frequencies between 1.5 and 1.7 μm, and considering the gaps between the three chips.

From our basic tests including Gaussian noise, we find no fundamental issue to extract these parameters, although nitrogen is the most weakly constrained parameter at low metallicities. The computing time per spectrum amounts to a few minutes on modern processors, but can be reduced to a few seconds per spectrum by taking advantange of Principal Component Analysis (PCA) compression for the spectral library. PCA also reduces by about a factor 10 the RAM requirements.

In our tests, interpolation errors in the theoretical spectra are found to contribute modestly to the errors in the atmospheric parameters, even though simple linear interpolation is used. In addition to the Nelder-Mead algorithm, we have found that UOBYQA algorithm of Powell (2002) also gives similar results, although it is slower for our problem. Only a limited loss of information is apparent in our tests

when the resolving power is reduced to $R = 10,000$, but an important degradation is found for $R = 3,000$.

The optimization code has been parallelized over the input spectra using OpenMP. The performance in parallel scales linearly for 2 CPUs, and more slowly for 3 and 4 CPUs, but it ceases to improve for more than 4 CPUs on a 8 CPU Intel computer.

6 References

- Allende Prieto, C. 2004, Astronomische Nachrichten, 325, 604
- Castelli, F., & Kurucz, R. L. 2004, IAU Symp. No 210; IAU Symp. No 210, Modelling of Stellar Atmospheres, eds. N. Piskunov et al. 2003, poster A20 (arXiv:astro-ph/0405087)
- Koesterke, L. 2009, American Institute of Physics Conference Series, 1171, 73
- Koesterke, L., Allende Prieto, C., & Lambert, D. L. 2008, ApJ, 680, 764
- Nelder, J. A., & Mead, R. 1965, Computer Journal, 7, 308-313
- Powell, M. J. D. 2002, Math. Progm., Ser. B 92, 555-582