



# ILIUM: An iterative local interpolation method for parameter estimation

---

prepared by: Coryn A.L. Bailer-Jones  
Max Planck Institute for Astronomy, Heidelberg  
Email: calj@mpia.de

approved by:

reference: GAIA-C8-TN-MPIA-CBJ-042

issue: 1

revision: 1

date: 2009-02-10

status: Issued

## Abstract

I introduce a new algorithm for parameter estimation with multidimensional data. While a general method, I develop it on the problem of estimating multiple astrophysical parameters (APs) from spectra. ILIUM iteratively interpolates over a training grid of labelled spectra to derive successively better estimates of the APs which are therefore not limited by the grid resolution. This interpolation uses nonlinear forward models to uniquely predict the spectra given the APs. ILIUM uses estimates of the sensitivity of the flux to the APs to weight the contribution of each flux band to the AP. Particular attention is paid to the fact that some APs are much “weaker” than others, in the sense that they are responsible for much less variance in the spectra and are therefore typically much harder to estimate. I demonstrate ILIUM on a grid of BP/RP stellar spectra showing variance in  $T_{\text{eff}}$  and  $\log g$ . Applied to  $G=15$  synthetic data, the mean absolute errors in  $\log g$  and  $\log(T_{\text{eff}})$  are 0.065 dex and 0.0010 dex respectively (RMS errors of 0.093 dex and 0.0014 dex) which are 7 and 5 times smaller (respectively) than that achieved with the nearest neighbours algorithm applied to noise-free data. At  $G=18.5$  the ILIUM errors are 0.35 dex in  $\log g$  and 0.0057 in  $\log(T_{\text{eff}})$  (equivalent to 1.3% in  $T_{\text{eff}}$ ). The algorithm also estimates individual AP errors and provides a goodness-of-fit for each spectrum. Extension of the algorithm to more APs is foreseen, and a simple extension should also allow it to identify AP degeneracies and so provide multiple solutions.

## Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Algorithm</b>	<b>6</b>
2.1	Forward modelling . . . . .	6
2.2	Core algorithm . . . . .	7
2.3	Generalization to multiple APs and multiple bands . . . . .	8
2.4	The forward model . . . . .	9
2.5	Performance statistics . . . . .	12
2.6	Uncertainty estimates . . . . .	12
2.7	Practical algorithm . . . . .	13
2.7.1	Preprocess data . . . . .	13
2.7.2	Sensitivity estimation . . . . .	13
2.7.3	Lower limit on sensitivities . . . . .	13
2.7.4	AP update contribution clipping . . . . .	13
2.7.5	AP update decay . . . . .	14
2.7.6	Upper limit on AP update step size . . . . .	14
2.7.7	Limit AP extrapolation . . . . .	15
2.7.8	Stopping criterion . . . . .	15
2.7.9	Implementation . . . . .	15
2.8	Alternative solution . . . . .	15

<b>3</b>	<b>Application</b>	<b>17</b>
3.1	The data . . . . .	17
3.1.1	The grid . . . . .	17
3.1.2	The spectra . . . . .	18
3.1.3	Train and test data sets . . . . .	18
3.2	The forward model . . . . .	18
3.3	Results at $G=15$ . . . . .	19
3.4	Results at $G=20$ . . . . .	26
3.5	Other results . . . . .	30
<b>4</b>	<b>Summary and discussion</b>	<b>32</b>
<b>5</b>	<b>Future work</b>	<b>33</b>

## 1 Introduction

The problem is to estimate the astrophysical parameters (APs,  $\phi$ ) of an object given a measurement of its spectrum ( $\mathbf{p}$ ). As there are multiple APs and multiple dimensions in the spectrum, this is sometimes referred to as a multidimensional multiple regression problem. A common approach is to use nearest neighbours. It gives low bias estimates, but often with large variance because its accuracy is limited by the AP resolution of the grid. Moreover, it suffers from the curse of dimensionality as the number of APs grows: the number of training points must grow exponentially in order to retain the same accuracy, with consequent penalties on the amount of data needed and run time.

For these reasons, one often resorts to fitting a regression model directly to a grid of labelled data to yield estimates  $\hat{\phi} = h(\mathbf{p}; \alpha)$ , where  $\alpha$  are the parameters of the fitted function. Machine learning methods such as neural networks (e.g. Re Fiorentin et al. 2007) and support vector machines (e.g. Tsalmantza et al. 2007, Bailer-Jones et al. 2008) can be employed when the dimensionality is high, which is the case for Gaia where there are several APs and the number of flux bands in the spectrum is of order 100.

One of the major limitations of this direct fitting approach is that the function  $h$  is not unique. Indeed, we know for stars that the APs are not single-valued functions of the fluxes – peek ahead to Fig. 7 (although multiple bands may result in this degeneracy being broken). For this reason, the “brute force” inverse fitting approach we usually adopt may actually be theoretically flawed, even though in practice it often provides good estimates. Furthermore, machine learning methods are often unable to produce AP uncertainty estimates or goodness-of-fit measures (at least not easily; see Tiede 2008 section 5.2), and it is often hard to understand in individual cases how the data were used to arrive at the AP estimates.

Most of these problems with these inverse methods arise from the absence of a *forward model*, a model which predicts the flux given the APs. Unlike  $h$ , the forward model is unique and can in principle be fitted easily using the training data. Here I introduce a new method which uses the forward model explicitly in AP estimation. For each new spectrum presented to classify, it interpolates the grid locally around the nearest neighbour estimate in an iterative fashion until some convergence is reached.

This technical note is organized as follows. In section 2 I introduce the algorithm as well as some practical modifications necessary for working on real data. Section 3 demonstrates it using Gaia BP/RP data. I summarize and discuss the results in section 4 and conclude with ideas for future work in section 5. Table 1 summarizes my notation.

TABLE 1: Notation

$a$	scalar
$a_i$	scalar with index
$\mathbf{a}$	vector (bold face). Compare vector $\phi$ with scalar $\phi$
$A$	matrix (upper case). $T$ denotes its transpose
$\hat{a}$	the hat indicates an estimated quantity
$i$	counter over band (spectrum pixel), $i = 1 \dots I$
$p_i$	photon counts (flux) in band $i$ ( $\mathbf{p}$ is a spectrum)
$j$	counter over AP, $j = 1 \dots J$
$\phi_j$	AP $j$
$\phi^S, \phi^W$	“strong” and “weak” AP respectively
$k, l$	counters over the strong and weak APs respectively
$s_{ij}$	sensitivity of band $i$ to AP $j$ , $\frac{\partial p_i}{\partial \phi_j}$
$S$	sensitivity matrix, $I \times J$ matrix with elements $s_{ij}$
$f_i(\phi)$	forward model for band $i$
$n$	iteration; e.g. $\phi(n)$ is the AP at iteration $n$

† column vectors in matrix equations

## 2 Algorithm

### 2.1 Forward modelling

Let us call the true relationship between the APs and the flux in a band <sup>1</sup>the *generative function*,  $g_i(\phi)$ . This encapsulates the underlying stellar physics, radiative transfer, instrument model etc. (There is a separate function for each band, but for simplicity I will refer to it in the singular.) We do not know the generative function: all we have for doing AP estimation is the grid, which is a sample of points selected from the generative function (typically via model atmospheres and synthetic spectra). Instead we approximate the generative function using a *forward model*, which is a (nonlinear) parametrized fit to the grid points. With this we can predict the flux in a band at arbitrary values of the APs, i.e. off the grid. (A separate forward model is fit independently for each band.) As the forward models are continuous functions, we can also use them to calculate the sensitivities, which by definition are the gradients of the flux with respect to each AP. The forward model fitting is done just once for a given grid and is kept fixed when predicting APs, i.e. it is a training procedure.

<sup>1</sup>“Band” is the term I used for a measured flux in the spectrum. It may be a photometric band, a pixel in the spectrum, or the sum over several pixels in a spectrum.

## 2.2 Core algorithm

The ILIUM parameter estimation algorithm in outline as follows. For a given measured spectrum, we find its nearest neighbour in the grid and calculate the residuals in all bands,  $\delta p$ . We then calculate the sensitivities,  $\{\partial p_i / \partial \phi_j\}$ , at the nearest neighbour and together with the residuals use a first order Taylor expansion of the forward model to determine the offset to apply to the current AP estimate. This provides a new AP estimate which we use in the forward model to predict the flux in each band. We then iterate the procedure.

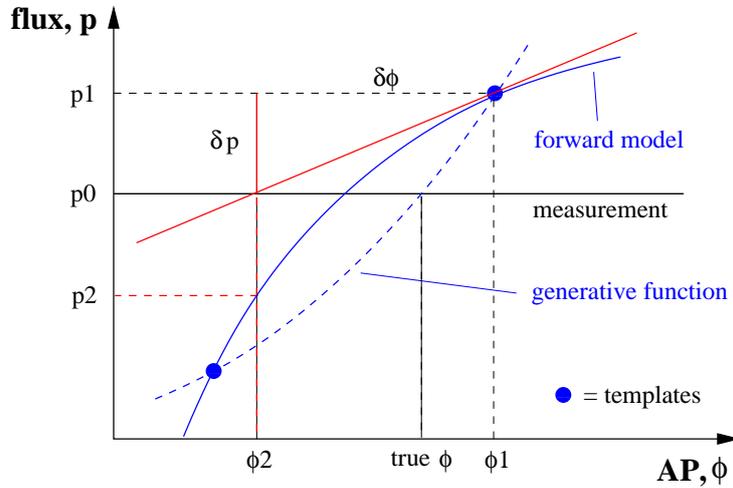


FIGURE 1: Sketch of the local iterative interpolation principle for one band ( $I=1$ ) and one AP ( $J=1$ ). The dashed blue line is the (unknown) generative function, and the solid curved blue line is the forward model (our approximation to the generative function) formed by fitting a function to the templates (grid points). The red line shows the local linear approximation of the forward model used to calculate the AP step. (In the case show the forward model could be inverted. But this is not generally the case, not even in one dimension if it has a turning point.)

In detail, the algorithm is as follows. We initially consider the situation with just one AP and one band. We first fit the forward model to the grid,  $\hat{p} = f(\phi)$ . This remains fixed. Let  $p(0)$  be the measured band for which we want to estimate the APs. The procedure (see also Fig. 1) is as follows ( $n$  is the iteration number)

1. Initialize: find nearest grid neighbour to  $p(0)$ . Call this  $[p(1), \phi(1)]$ .  $\phi(1)$  is the initial AP estimate.
2. Use the forward model to calculate the local sensitivities,  $\frac{\partial p}{\partial \phi}$ , at the current AP estimate.
3. Calculate the discrepancy (residual) between the predicted flux and the measured flux,  $\delta p(n) = p(n) - p(n-1)$ . (This is equal to  $p(1) - p(0)$  for the first iteration.)

4. Estimate the AP offset as  $\delta\phi(n) = \left(\frac{\partial\phi}{\partial p}\right)_{\phi(n)} \times \delta p(n)$ , i.e. a Taylor expansion truncated to the linear term. (Note that the partial derivative is the *reciprocal* sensitivity.) This is the offset between the true AP value and the current estimate in a local linear approximation.
5. Make a step in AP space,  $\phi(n+1) = \phi(n) - \delta\phi(n)$ . (Note the sign, which steps *toward* the better estimate.) This is the new AP prediction.
6. Use the forward model to predict the corresponding (off-grid) flux,  $p(n+1)$
7. Iterate steps 2–7 until convergence is achieved or a stop is imposed.

At each iteration we obtain an estimate of the APs (step 5) and the corresponding spectrum (step 6). The algorithm alternates between estimating the APs and calculating an update to apply to the current estimate. Convergence could be defined in several ways, e.g. when changes in the spectrum or the APs (or their rate of change) drop below some threshold. Alternatively we could simply stop after some fixed number of iterations.

If the generative function were locally linear, and if the forward model were exact (and the data were noise free), then step 5 would yield the exact APs in the first iteration and the algorithm would converge (as  $\delta p$  would be zero in the next iteration). As neither the generative function nor its approximation, the forward model, is linear, there is no guarantee of convergence. For example, if the AP steps were sufficiently large to move to a part of the function with a sensitivity of the opposite sign, then the model could diverge or get stuck in a limit cycle. For this, and other reasons, the algorithm in practice has some additional features (discussed in section 2.7).

### 2.3 Generalization to multiple APs and multiple bands

In general we have several bands and several APs.

For multiple bands ( $I > 1$ ) and one AP we have  $I$  independent forward models and must replace  $p$  with  $\mathbf{p}$ . The estimated AP residual in step 5 is then the sum over all bands (I now drop the iteration index)

$$\begin{aligned} \delta\phi &= \sum_i \left(\frac{\partial\phi}{\partial p_i}\right)_{\phi} \delta p_i \\ &= \sum_i \left(\frac{\partial p_i}{\partial\phi}\right)_{\phi}^{-1} \delta p_i \end{aligned} \quad (1)$$

As we still have just one AP, we can simply use the inverse of the sensitivities in the above equation. This is not the case for multiple APs, because in general  $\partial\phi_j/\partial p_i \neq (\partial p_i/\partial\phi_j)^{-1}$ . Without a forward model we also could not calculate the  $\{\partial p_i/\partial\phi_j\}$  from the grid, because

in general there is no function  $\phi_j(\mathbf{p})$  (as Fig. 7 demonstrates). This is one of the motivations behind ILIUM.

For multiple APs ( $J > 1$ ) as well as multiple bands, we write the flux perturbation in band  $i$  due to small changes in the APs as

$$\delta p_i = \sum_j \left( \frac{\partial p_i}{\partial \phi_j} \right) \delta \phi_j \quad (2)$$

For  $I$  bands we write this in matrix form as

$$\delta \mathbf{p} = S \delta \phi \quad (3)$$

where  $S$  is the  $I \times J$  sensitivity matrix with elements  $s_{ij} = \partial p_i / \partial \phi_j$ . We assume that  $I > J$ . If we now multiply equation 3 on the left by  $(S^T S)^{-1} S^T$  we get

$$\delta \phi = (S^T S)^{-1} S^T \delta \mathbf{p} \quad (4)$$

The AP update equation (step 5 in the algorithm) is then

$$\phi(n+1) = \phi(n) - (S^T S)^{-1} S^T \delta \mathbf{p}(n) \quad (5)$$

The basic procedure is otherwise unchanged. Of course, the  $I$  forward models are now functions of  $J$  variables, and this turns out to be a critical matter.

## 2.4 The forward model

The basic algorithm described so far can make use of any forward model, on the condition that we can calculate the function values and its first derivatives for arbitrary values of the APs.

The most obvious forward model would be a multidimensional, nonlinear regression of the form  $\hat{\mathbf{p}} = f(\phi)$ , which in principle works for any number of APs. However, I found that it was very hard to get a model which simultaneously fit both  $T_{\text{eff}}$ , a strong AP, and  $\log g$ , a weak AP: The former could be fit well, but the latter not. *Strong* here means that it explains much of the variance in the flux data, i.e. is a strong predictor of the flux. *Weak* is a relative term, indicating that the AP explains much less of the variance. The reason for the poor fit over the weak AP was that the model had to be fit by minimizing a single objective function, namely the error in reproducing the flux. As the weak AP has very little impact on the flux, its influence has little impact on the error, so the model optimization does little to produce a good fit over this AP. I experimented extensively with many models and with ways of improving the sensitivity of an error function to the weak APs, but to little avail.

To overcome this problem, I use a two-component forward model, which currently is only defined for two APs. The first component fits a single 1D nonlinear function to the strong AP,

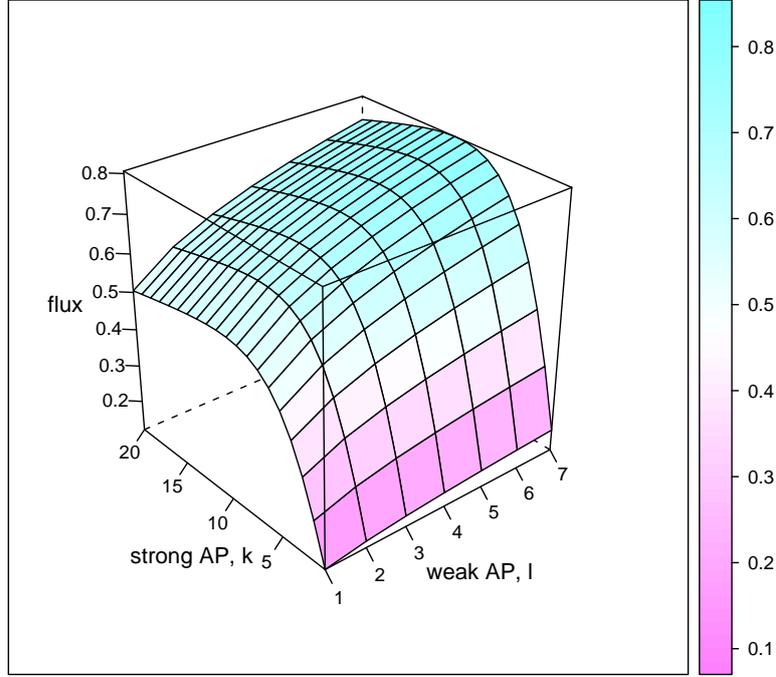


FIGURE 2: Schematic of a two-component forward model. Note the much stronger variation in the flux in the direction of the strong AP. For  $T_{\text{eff}}$  and  $\log g$  the relative variation with respect to  $T_{\text{eff}}$  is much larger than that shown here for most, if not all, bands.

marginalizing over the weak AP. This fits most of the flux variation. Then, for each discrete value of the strong AP in the grid, I separately fit the residual flux as a function of the weak AP; these are the second components (also 1D). They provide a flux increment dependent on the weak AP, which is added to the flux predicted given the strong AP. A schematic illustration of such a two-component forward model is illustrated in Fig. 2.

The two-component model is fit as follows. Let  $\phi^S$  denote a strong AP,  $\phi^W$  a weak AP and  $f_i(\phi^S, \phi^W)$  the complete (2D) forward model for band  $i$ . Let subscripts  $k$  and  $l$  denote specific values in the grid of the strong and weak APs respectively (see Fig. 3). We model the flux at some arbitrary AP value as

$$f_i(\phi^S, \phi^W) = f_i^S(\phi^S) + f_{i,k}^W(\phi^W; \phi^S = \phi_k^S) \quad (6)$$

where both  $f_i^S$  and  $f_{i,k}^W$  are 1D functions.  $f_i^S(\phi^S)$  is the single strong component, which fits the flux variation with respect to the strong AP, marginalizing over all  $\phi^W$ .  $f_{i,k}^W$  is the  $k^{\text{th}}$  weak component, which is a fit of the flux variation with respect to the weak AP with the strong AP fixed at the nearest value of  $\phi^S$ , i.e.  $\phi_k^S$ . To fit this we therefore use those points in the grid

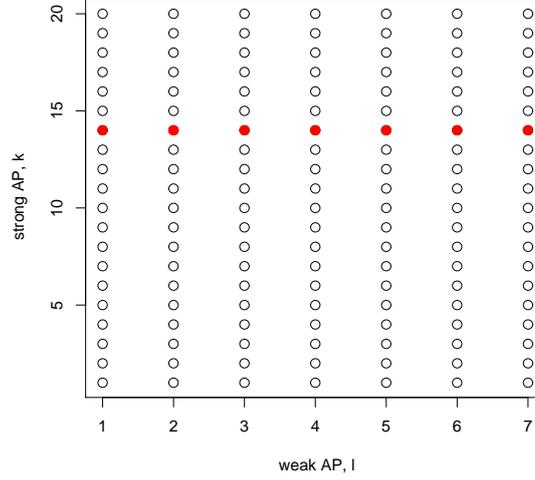


FIGURE 3: Schematic of the grid of AP values,  $k = 1 \dots 20$ ,  $l = 1 \dots 7$ . The red points denote those used to fit the weak component of the forward model for one value of the strong AP.

at fixed  $\phi^S$  and variable  $\phi^W$  (e.g. the red points in Fig. 3). Clearly, this requires us to have a “semi-regular” grid, one consisting of spectra with a range of values of the weak AP for each value of the strong AP. (This requirement is easily fulfilled when using synthetic-based data.) The two components are illustrated in Fig. 4. The number of weak components is equal to the number of values of  $\phi^S$ , which is 20 in this schematic case.

$f_i^S(\phi^S)$  is a fit to the whole grid of flux values,  $\{p_i\}$ . As this fit marginalizes over  $\phi^W$  it is implicitly a fit at the average value of  $\phi^W$  at each  $\phi^S$ . For this reason, each weak component is fit to the grid flux data minus the average flux at that value of  $\phi^S$ , i.e. to the set of values

$$\{p(\phi^S = \phi_k^S) - \overline{p(\phi^S = \phi_{k'}^S)}\}$$

Currently both the strong and weak components of the forward model are fit using smoothing splines, with the number of degrees-of-freedom (DoF) set according to the grid (see section 3.2).

Applying the forward model is easy. Given  $(\phi^S, \phi^W)$  we

1. evaluate  $f_i^S$ , the strong component
2. find the nearest value,  $\phi_k^S$ , in the grid to  $\phi^S$ , i.e. identify the closest weak component
3. evaluate  $f_{i,k}^W$ , the increment from the weak component
4. sum the two components,  $f_i^S + f_{i,k}^W$ , to give the forward model prediction.

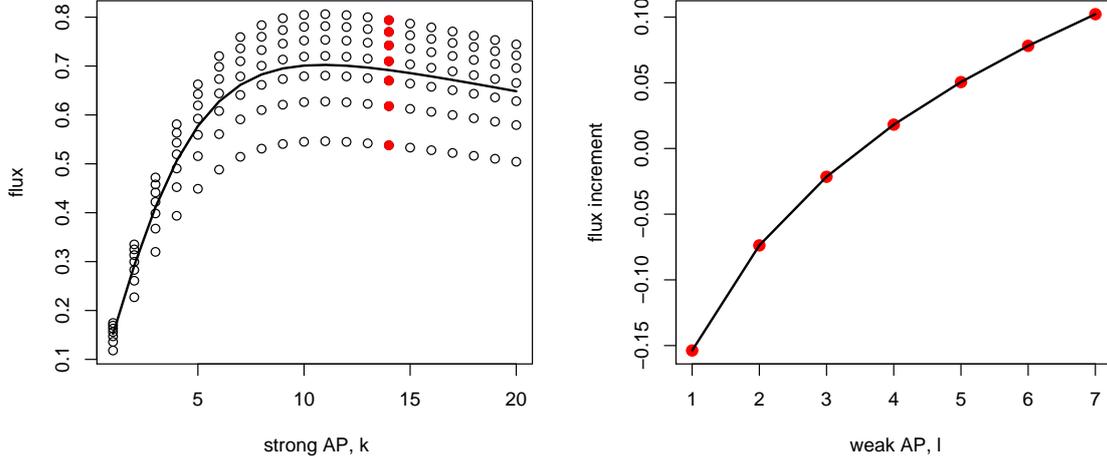


FIGURE 4: Schematic illustration of the two components of the forward model. Left: the fit over the strong APs. Right: one of the fits over the weak APs (the red points in the left panel at  $k=14$ ).

## 2.5 Performance statistics

Model performance is assessed primarily via the AP residuals on a test data set. In addition to examining these as a function of the APs, I report three statistics : (1) the RMS,  $\sqrt{\overline{\delta\phi^2}}$ , which we abbreviate with  $\sigma_\phi$ , (2) the mean absolute residual,  $\overline{|\delta\phi|}$  (which is less sensitive to outliers than the RMS) and (3) the mean residual,  $\overline{\delta\phi}$  (a measure of the systematic error).

## 2.6 Uncertainty estimates

We can estimate the uncertainties on the AP estimates of unlabelled data using the same formulation for estimating the APs. Specifically,  $|\delta\phi|$  is the AP uncertainty estimate, where  $\delta\phi$  is given by equation 4.  $S$  is evaluated at the estimated AP value, and  $\delta\mathbf{p}$  is the difference between the forward model predicted flux at this estimate and the measured flux. This is essentially the final step not made, but would be just what we want if the model gets stuck in a limit cycle.

We can also calculate a goodness-of-fit (GoF) for unlabelled data, because ILIUM predicts the spectrum from the estimated APs using the forward model. We use the reduced- $\chi^2$

$$\chi^2 = \frac{1}{I-1} \sum_{i=1}^{i=I} \left( \frac{p_i - \hat{p}_i}{\sigma_{p_i}} \right)^2 \quad (7)$$

where  $\hat{p}_i = f_i(\phi_j)$  is the spectrum predicted by the forward model. The  $\{\sigma_{p_i}\}$  are estimates of

the standard deviation of the noise in the measured flux.

## 2.7 Practical algorithm

The practical realities of working with real (noisy) data mean that the basic algorithm should be supplemented with a few additional features in order to make it more robust.

### 2.7.1 Preprocess data

We standardize each band and each AP, i.e. we linearly scale each to have zero mean and unit standard deviation.

### 2.7.2 Sensitivity estimation

If the forward model is a simple analytical function, then it may have analytical first derivatives which can be used to calculate the sensitivities. But in the general case we can use the method of first differences, i.e.

$$\frac{\partial p}{\partial \phi_j} \simeq \frac{f(\phi + \delta \phi_j) - f(\phi - \delta \phi_j)}{2 \delta \phi_j} \quad (8)$$

I choose  $2 \delta \phi_j$  to be 0.1 dex for  $\log g$  and 0.5% for  $T_{\text{eff}}$  [0.0022 for  $\log(T_{\text{eff}})$ ], which translate to 0.03 and 0.0034 in standardized units respectively for the grid used in section 3.

### 2.7.3 Lower limit on sensitivities

Given that the AP updates depend upon the inverse of the sensitivity matrix (equation 4), it seems prudent to prevent the sensitivities (calculated in equation 8) from being too small. For this reason, a lower limit is placed on the absolute value of each sensitivity,  $s_{ij}$ , of 0.001 (with  $p$  and  $\phi$  in standardized units). On the data set presented in section 3, no band fell below this limit in most iterations for most objects, and even when it did it was only for a few bands. Removing this lower limit actually never resulted in singular matrix and had negligible impact on the results, so for all practical purposes its existence can be ignored.

### 2.7.4 AP update contribution clipping

Equation 4 can be written

$$\phi(n+1) = \phi(n) - M \delta \mathbf{p}(n) \quad (9)$$

where  $M = (S^T S)^{-1} S^T$  is a  $J \times I$  matrix. The above equation is a set of  $J$  update equations, where the update for AP  $j$  is the dot product of two vectors, the  $j^{\text{th}}$  row of  $M$ ,  $\mathbf{m}_j$ , with  $\mathbf{p}(n)$ ,

i.e.

$$\begin{aligned}
 \phi_j(n+1) &= \phi_j(n) - \mathbf{m}_j \delta \mathbf{p}(n) \\
 &= \phi_j(n) - \sum_i m_{ij} \delta p_i(n) \\
 &= \phi_j(n) - \sum_i u_{ij}(n)
 \end{aligned} \tag{10}$$

which defines  $u_{ij}$ . The update to AP  $j$  is a sum over  $I$  individual updates, which we can view as an update “spectrum”. In section 3 we inspect these updates, and find that on occasion some are much larger than the others. This is undesirable, as it means that the update is dominated by just one or a few components (which may be adversely affected by noise, because  $\delta \mathbf{p}(n)$  is a noisy measurement). For this reason, I clip outliers in the spectrum of the  $I$  updates. (It is valid to compare the updates for different bands, because we work with standardized fluxes.) To be robust, I set an upper (lower) limit which is a multiple  $c$  of the median of those points above (below) the median. Using the notation  $\theta(\cdot)$  to denote median, the limits are

$$\begin{aligned}
 u_{\text{upper}} &= \theta(u_i) + c[\theta(u_i > \theta(u_i)) - \theta(u_i)] \\
 u_{\text{lower}} &= \theta(u_i) + c[\theta(u_i < \theta(u_i)) - \theta(u_i)]
 \end{aligned} \tag{11}$$

I somewhat arbitrarily set  $c = 10$  so as to be relatively conservative in clipping. In the baseline example (section 3), typically several updates are clipped per band/iteration.<sup>2</sup>

### 2.7.5 AP update decay

At faint magnitudes ( $G > 17$ ), I found that the AP updates would sometimes get stuck in limit cycles, i.e. the updates would oscillate between two values, occasionally with a large amplitude. To overcome this, I introduce a modification which decreases the magnitude of the AP updates with increasing iteration. Specifically I use an exponential decay such that the AP step,  $\delta \phi = \{\delta \phi_{\text{logg}}, \delta \phi_{\text{logteff}}\}$ , at iteration  $n$  is multiplied by

$$2^{-(n-1)/s}$$

where  $s$  is the decay time scale, i.e. the number of iterations over which the nominal AP update is halved. Convergence (AP updates tending to zero) is thereby enforced. The default is not to use this decay (and in fact all results shown in this TN are without decay).

### 2.7.6 Upper limit on AP update step size

The AP steps at each iteration (equation 4) could in principle be very large. This is undesirable, because the updates are based on a local *linear* approximation to the generative function. The code therefore imposes upper limits on the  $\log g$  and  $\log(T_{\text{eff}})$  AP updates of 2.0 dex and 0.04

<sup>2</sup>Note that it only makes sense to clip if there are more than three bands; in practice many more.

(10%) respectively (which correspond to 1.3 and 0.13 respectively in standardized units). Note that a larger step size is permitted for the weaker AP, because the initial nearest neighbour offset can be quite incorrect. In practice these limits are rarely imposed.

### 2.7.7 Limit AP extrapolation

We do not expect the forward model to make good predictions beyond the AP extremes of the grid, so I set upper and lower limits on the AP estimates which ILIUM can provide. These are set as  $e$  times the range of each AP, i.e.

$$\begin{aligned} \text{upper limit} &= \max \phi_j + e(\max \phi_j - \min \phi_j) \\ \text{lower limit} &= \min \phi_j - e(\max \phi_j - \min \phi_j) \end{aligned} \quad (12)$$

The default is to permit 10% extrapolation, i.e.  $e = 0.1$ .

### 2.7.8 Stopping criterion

The algorithm is simply run for a fixed number of iterations. As we see good natural convergence after about 20 iterations, a more sophisticated stopping criterion is not necessary at present.

### 2.7.9 Implementation

ILIUM is currently written in R (around 400 lines of code, with little error trapping). No effort has been made to write it efficiently, so the application phase currently requires around 1 second per object (with 20 iterations and 137 objects in grid). The forward modelling (training), requires just a few seconds. In comparison, a cross validation nearest neighbour code also written in R (just a few lines) takes just a second or so for all 274 objects, and even this is dominated by the file reading. Although ILIUM presently uses nearest neighbours for its initialization, the grid necessary for this probably does not need to be as dense as a grid for AP estimation via nearest neighbours. Therefore, ILIUM should not be as exposed to the curse of dimensionality.

The nominal values of the ILIUM model parameters are given in Table 2. Decay is not normally used, the minimum absolute sensitivity is not needed,  $c$  and  $e$  are the same for both APs and the delta for the sensitivity is not fundamental to the method. ILIUM can therefore be considered to have 6 adjustable parameters.

## 2.8 Alternative solution

In an earlier version of this algorithm (Bailer-Jones 2008), the matrix  $(S^T S)^{-1} S^T$  in equation 4 was replaced with  $R = [r_{ji}]$ , the  $J \times I$  matrix of reciprocal sensitivities ( $r_{ji} = s_{ij}^{-1}$ ). This is

TABLE 2: Model parameters, divided according to subalgorithm in which they are used.  $N_{\text{Teff}}$  is the number of unique  $T_{\text{eff}}$  values in the grid,  $N_{\text{logg}}$  is the number of unique  $\log g$  values in the grid at the nearest neighbour temperature (used in the two-part forward model).

Parameter	Nominal settings	
<i>General (interp.update):</i>		
number of iterations	20	
	$\log g$	$\log(T_{\text{eff}})$
<i>Forward model calculation (calc.formod):</i>		
DoF in forward model fit	$4^\dagger$	$N_{\text{Teff}}/2$
<i>Sensitivity calculation (est.sens):</i>		
delta for sensitivity calculation via first differences*	0.03	0.0034
minimum absolute sensitivity*	0.001	0.001
<i>AP updates (step.aps, interp.update):</i>		
decay scale <sup>^</sup> , $s$	–	–
update contribution clipping factor, $c$	10	10
max. step size*	1.3	0.13
AP extrapolation limit factor, $e$	0.1	0.1

<sup>†</sup> linear interpolation is used if  $N_{\text{logg}} \leq 4$

\* in units of the standardized variables

<sup>^</sup> default is not to use decay

obviously not mathematically correct, because  $\partial\phi_j/\partial p_i \neq (\partial p_i/\partial\phi_j)^{-1}$ , but maybe it was adequate and is potentially faster. I found that this approach yielded reasonable results (competitive with nearest neighbours), but only if the AP step size was modified (made smaller), and if the update decay was used. If decay was not used the model sometimes converged anyway, but often drifted away or (more often) became stuck in limit cycles (sometimes with a superimposed drift). The best results obtained with this approach on the G=15 data described in section 3 were of order

	$\log g$	$\log(T_{\text{eff}})$
$\overline{\delta\phi}$	-0.15	0.0010
$ \overline{\delta\phi} $	0.50	0.0046
$\sigma_\phi$	0.91	0.0070

## 3 Application

### 3.1 The data

#### 3.1.1 The grid

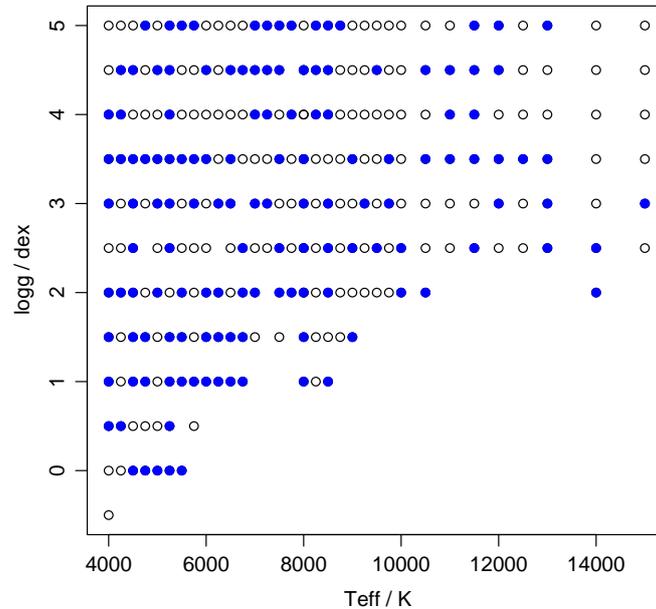


FIGURE 5: The AP grid used for the experiments. The filled points are the random selection used for the forward model fitting.

The AP grid is built from the MARCS and BASEL GOG simulations of single stars (cycle 3

data) using the nominal grid of the main stellar library (Sordo & Vallenari 2007). The strong AP is  $T_{\text{eff}}$  and the weak AP is  $\log g$ . For the main experiments I use just those spectra with  $[\text{Fe}/\text{H}]=0$  and  $A_V=0$ . This leaves a grid of 274 unique AP combinations, as show in Fig. 5. There are 33 discrete  $T_{\text{eff}}$  values ( $N_{T_{\text{eff}}} = 33$ ) and the number of discrete  $\log g$  values at each  $T_{\text{eff}}$ ,  $N_{\log g}(T_{\text{eff}})$ , varies between 6 and 12.  $\log(T_{\text{eff}})$  rather than  $T_{\text{eff}}$  is used internally to ILIUM.

### 3.1.2 The spectra

The spectra are non-oversampled end-of-mission BP/RP spectra truncated to remove low SNR pixels (bands). The original cycle 3 BP/RP have  $2 \times 180$  bands, from which we select numbers 1,4,7 etc. in order to remove the factor 3 oversampling. Of this set of 120 bands, we retain bands 12 : 45 and 75 : 108, a total of 68 bands of which 34 bands are in BP covering 338–634 nm and 34 bands are in RP covering 667–1035 nm. Some experiments have been done varying the data range, but not exhaustively and are not reported here. Note that the values in the corresponding sigma spectra have to be reduced by  $\sqrt{3}$  in order to consistently remove the oversampling, as summarized in Smith (2008). (Compared to the pixel range used in Bailer-Jones et al. 2008, we have now removed three pixels from each end of BP and RP and an additional six pixels from the red end of BP.)

### 3.1.3 Train and test data sets

Training data are used to build the forward model and the performance assessed on the test set. The training set is a random selection of half of the grid, i.e. 137 objects (the filled points in Fig. 5), and is noise-free. The test set is the complement (the other 137). For a given experiment the test set are at a single magnitude with zero mean Gaussian noise added via random sampling (with the standard deviation given by the sigma spectra).

## 3.2 The forward model

The 1D forward model over the strong AP,  $T_{\text{eff}}$ , is a smoothing spline with  $N_{T_{\text{eff}}}/2 = 16.5$  degrees-of-freedom (DoF) (I use the R package `sreg`). Each of the 1D weak forward models are also smoothing splines. As the maximum number of  $\log g$  points is 10 (for the training data), and because the variation with  $\log g$  is quite smooth, I set the DoF for these fits to be 4. However, many of the  $T_{\text{eff}}$  values in the training grid have fewer  $\log g$  points: To avoid overfitting, if  $N_{\log g}(T_{\text{eff}}) \leq 4$  then a linear fit is used (R package `polreg` with `degree=1`). If  $N_{\log g}(T_{\text{eff}}) = 1$ , then no fit is performed and this weak component of the forward model is zero.

Getting a good fit for the forward model is a critical part of the algorithm. Figs. 6 and 7 show its fit (to the noise-free grid, of course) as a function of  $\log(T_{\text{eff}})$  and  $\log g$ . The fits are generally very good, also showing the degree of smoothness we would expect for these data plus a robust extrapolation. Smoothness is important for a reliable calculation of the sensitivities via first

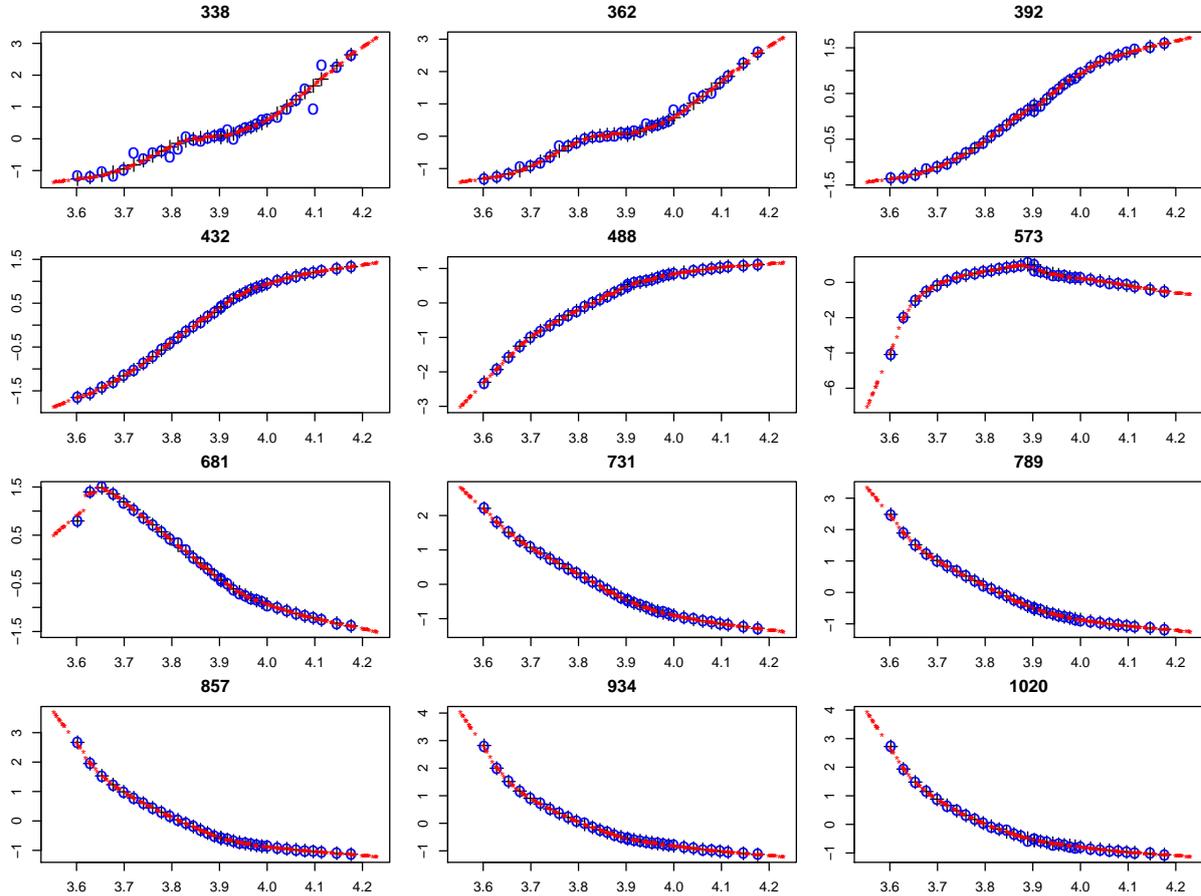


FIGURE 6: Predictions of the full forward model as a function of  $\log(T_{\text{eff}})$  at constant  $\log g=4.0$  in 12 different bands (with wavelength in nm at the top of each panel). The black crosses are the (noise-free) grid points, the red stars are the forward model predictions (at randomly selected AP values) and the blue circles the noisy ( $G=15$ ) grid points. The flux plotted on the ordinate is in standardized units.

differences. Compare the flux scales between Figs. 6 and 7 and notice how small the flux variation is as  $\log g$  varies over its full range compared to  $T_{\text{eff}}$ : This is what it means to be a weak AP!

### 3.3 Results at $G=15$

I apply ILIUM to the  $G=15$  test data with the nominal ILIUM parameter settings with decay turned off. Fig. 8 shows how the AP estimates evolve, for 5 different stars. The first iteration is the nearest neighbour estimate; the final is the adopted AP estimate. The red lines show the true APs. We see a range of convergence behaviours. Sometimes ILIUM very rapidly converges, for other stars it takes longer, sometimes it is smooth, other times not, and it can be quite different

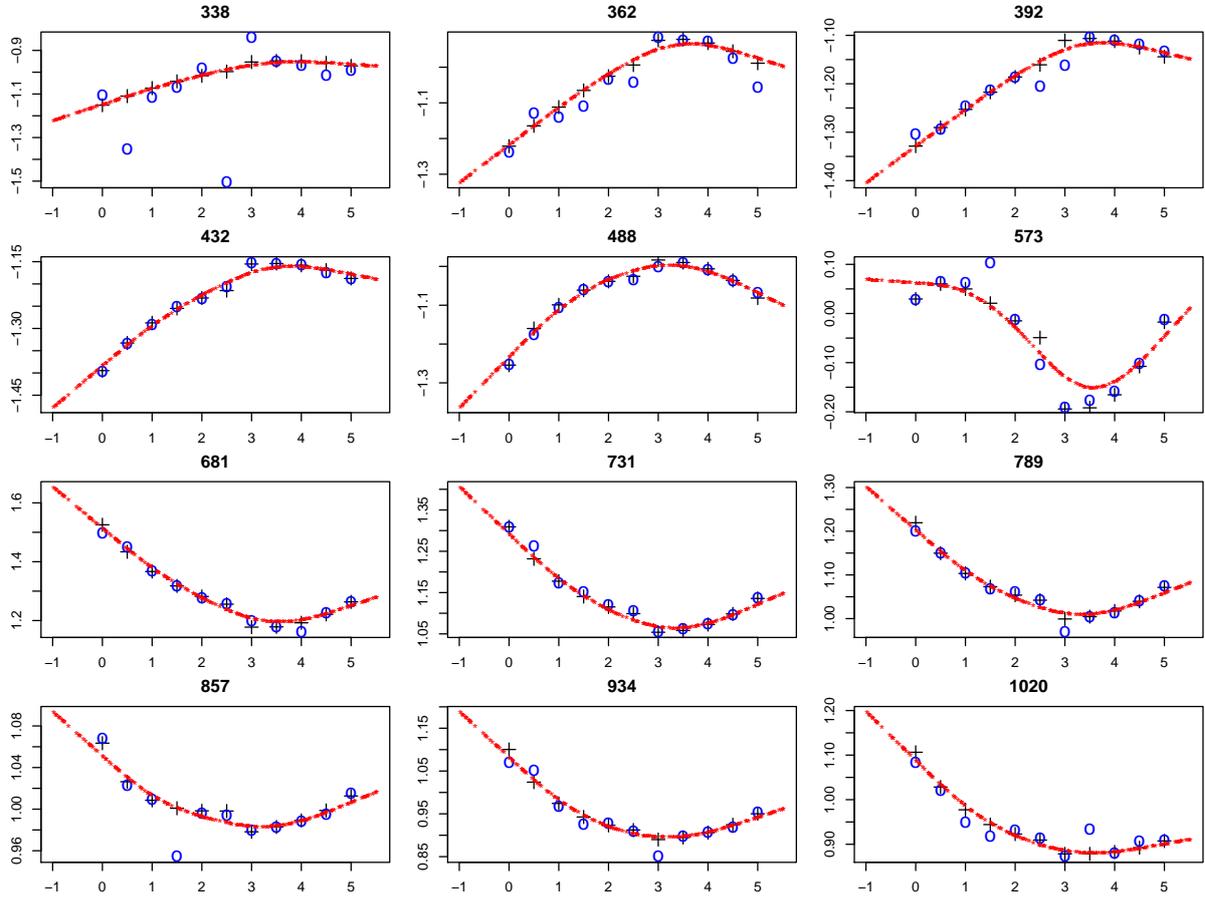


FIGURE 7: As Fig. 6, but now showing predictions of the full forward model as a function of  $\log g$  at constant  $T_{\text{eff}}=5000$  K.

for the two APs for a given star. This behaviour of course depends not only the APs, but also the specific measurements (as they are noisy). Sometimes the nearest neighbour estimate for one or other of the stars is the correct one, and ILIUM may actually iterate away from this and converge on a different value (although usually quite close). Convergence is almost always achieved, even though decay is turned off (so there is nothing in the algorithm to enforce convergence). This is a very encouraging property. Limit cycles in the APs are seen in a handful of cases, but the oscillation amplitude is very small, say a few hundredths of a dex in  $\log g$  or a few K in  $T_{\text{eff}}$ . The upper limits on the AP estimations as well as the AP updates, although in place, never had to be enforced for any star by the algorithm.

As described in section 2.7.4, we can calculate the contribution to the update of AP  $j$  for each band  $i$ : this is  $u_{ij}$  in equation 10. Fig. 9 and Fig. 10 show the spectrum of these AP updates at each iteration for one particular star (no. 67). Points which lie outside of the limits (dashed lines) are clipped. The numbers printed in each panel are the actual update (in standardized

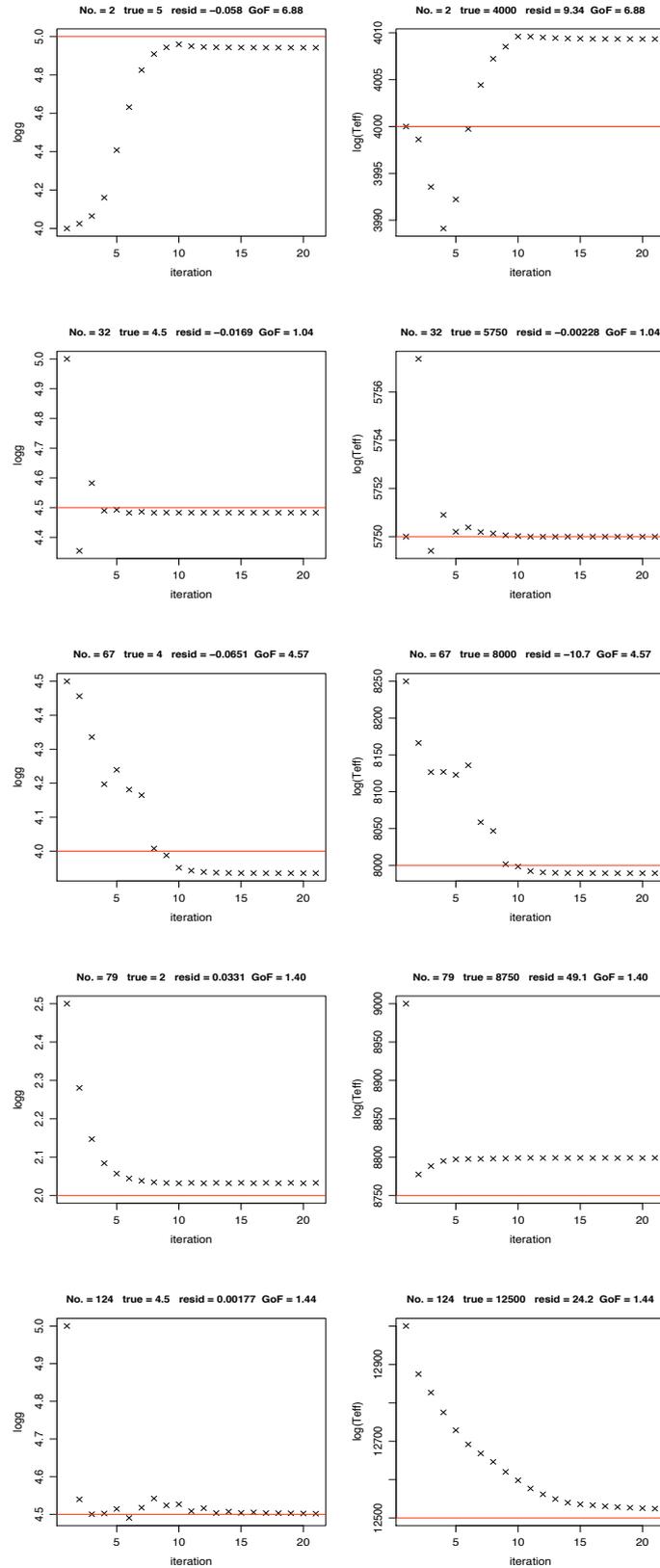


FIGURE 8: AP evolution for 5 stars in the test data set at  $G=15$  ( $\log g$  left,  $T_{\text{eff}}$  right). The true APs are written at the top of each panel pair and plotted as the red horizontal line. GoF is the  $\chi^2$  goodness-of-fit (equation 7).

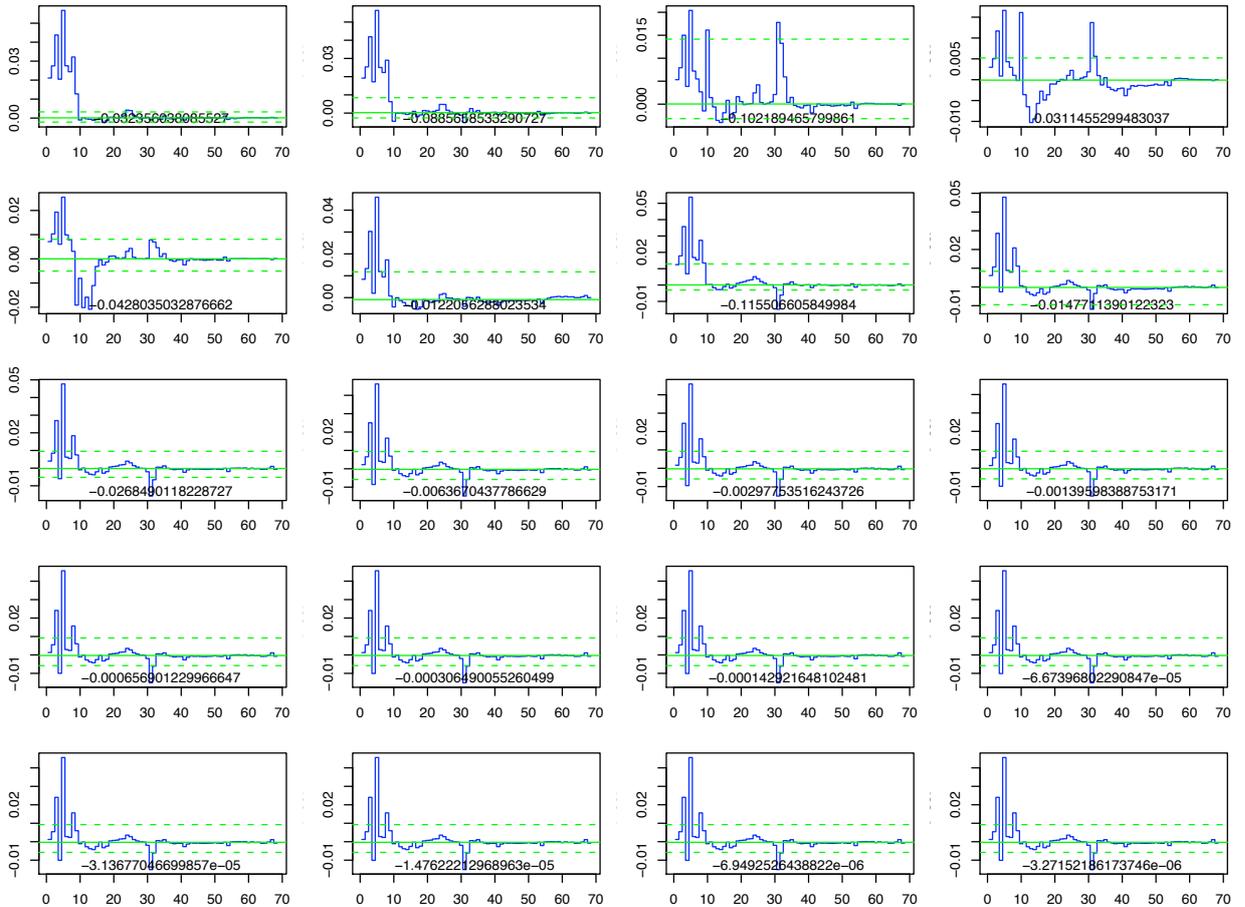


FIGURE 9: The spectrum of  $\log g$  updates ( $u_{ij}$  in equation 10) for star 67 ( $\log g=4$  dex,  $T_{\text{eff}}=8000$  K) at each iteration, with iteration increasing first across columns then down the rows from the top-left. In each panel (iteration), the horizontal axis is the band (pixel) number and the vertical axis the AP update, in standardized units. The solid horizontal line is the mean, and the dashed lines the upper and lower clipping limits

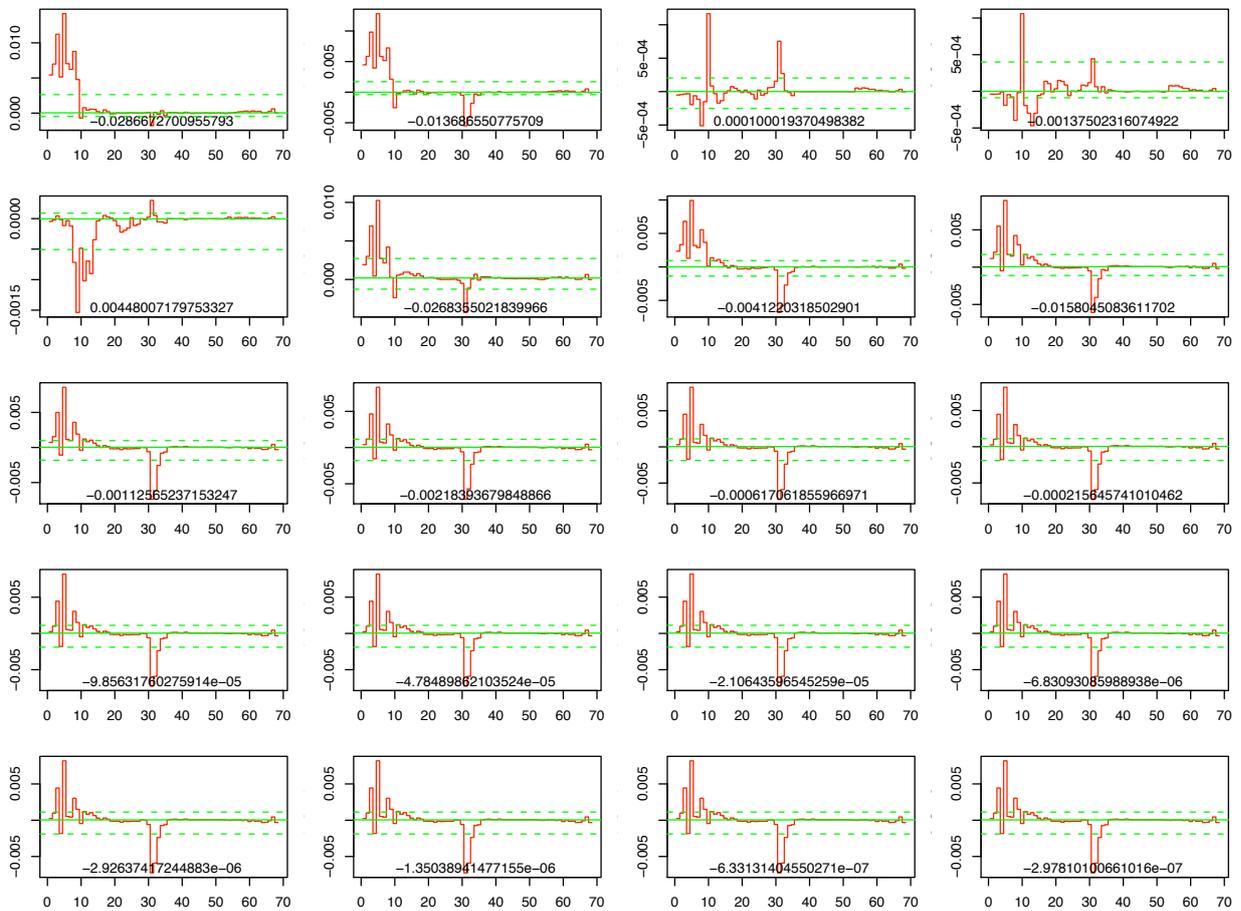


FIGURE 10: As Fig. 9 but for  $\log(T_{\text{eff}})$ .

units) calculated from the spectrum, i.e. the sum after the clipping. We can see how the spectrum of updates evolves with time and how this correlates with the AP evolution in Fig. 8

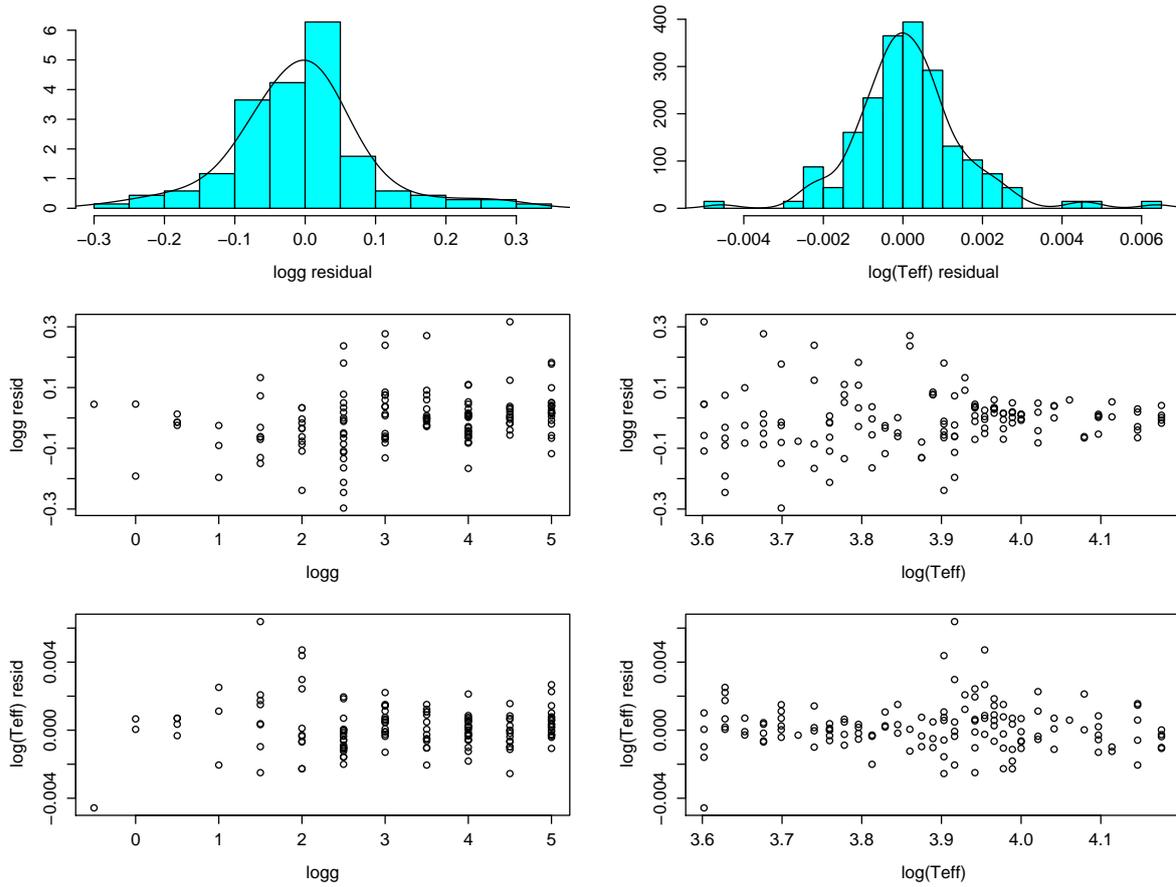


FIGURE 11: AP residuals for the test data at G=15

The residuals are shown in Fig. 11. The summary statistics are

	$\log g$	$\log(T_{\text{eff}})$	
$\overline{\delta\phi}$	-0.0073	$1.0e^{-5}$	ILIUM, G=15
$ \overline{\delta\phi} $	0.065	0.0010	
$\sigma_\phi$	0.093	0.0014	

In comparison, the results below are from a nearest neighbour method which has two advantages with respect to ILIUM: (1) it uses leave-one-out cross validation, i.e. has almost the complete grid for each estimation; (2) it is for noise-free test data

	$\log g$	$\log(T_{\text{eff}})$	
$\overline{ \delta\phi }$	0.026	$-9.4e^{-5}$	Nearest neighbour, noise-free
$\overline{ \delta\phi }$	0.43	0.0034	
$\sigma_\phi$	0.47	0.0067	

ILIUM achieves absolute errors in  $\log g$  and  $\log(T_{\text{eff}})$  which are a factor of 7 and 5 smaller respectively. Thus ILIUM has achieved its main goal of overcoming the grid resolution to perform significantly better than nearest neighbours.

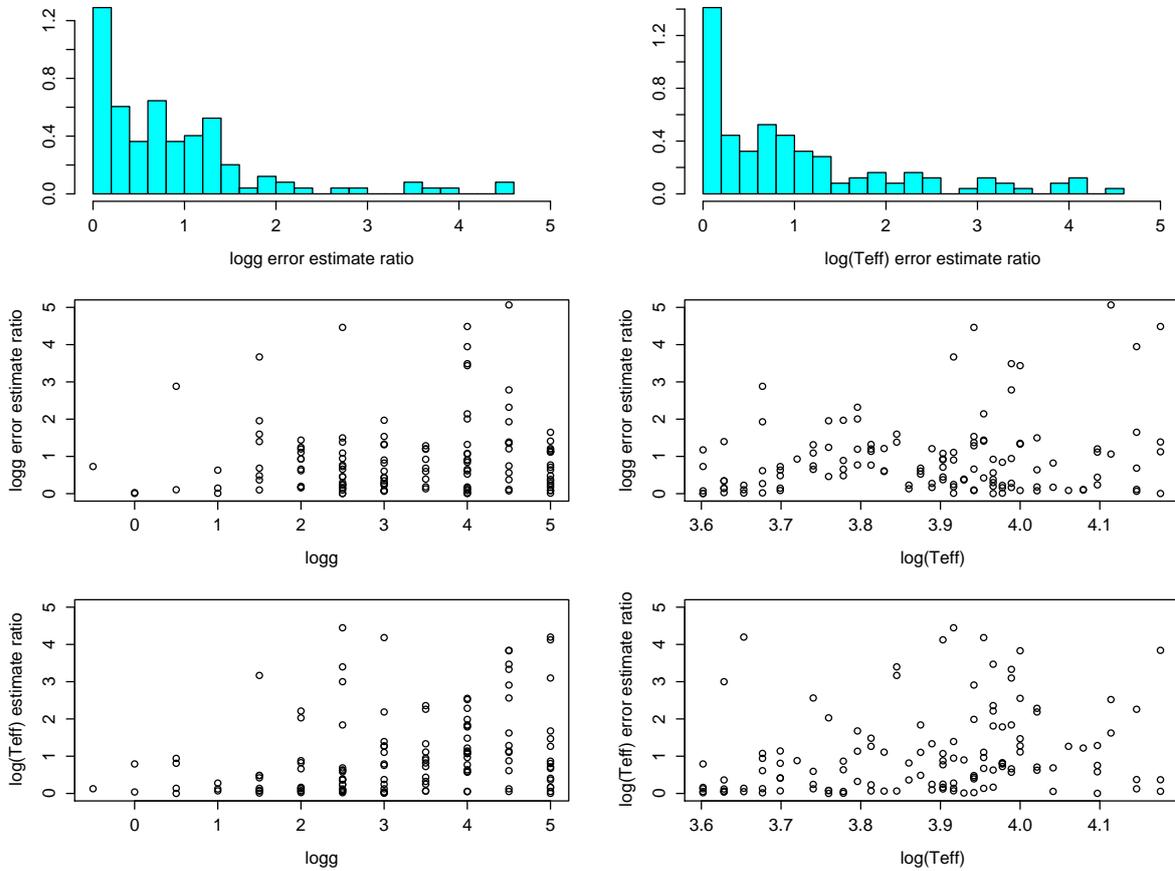


FIGURE 12: Estimated AP uncertainties expressed as a ratio of the absolute value of the true residuals, for  $G=15$

As described in section 2.6, ILIUM can estimate individual AP uncertainties. We examine the accuracy of these by comparing them to the true residuals on the test set. Fig. 12 plots the ratio of the AP uncertainties to the absolute value of the true residuals. This is plotted in the same way as the AP residuals in Fig. 11 to show any variation with AP. (We perhaps see a tendency towards a smaller ratio for  $\log g$  at low  $\log g$ .) The plots are limited to ratios below 5: there are 26 cases (from 274) where the uncertainty estimate is more than five times the true residual, in all cases because the residual is very small rather than the uncertainty estimate unreasonably

large. Indeed, the largest estimated uncertainties are 0.27 dex in  $\log g$  and 0.0072 in  $\log(T_{\text{eff}})$ . 41% of the uncertainty estimates are within a factor of 2 of the true residual, which shows the uncertainty estimates to be order-of-magnitude correct. (We obviously would not expect the distribution of the ratios to closely peak around 1, as this would imply a systematic error which can be accurately predicted and corrected for!)

### 3.4 Results at G=20

I now apply ILIUM to the G=20 test data again with the nominal ILIUM parameter settings with decay turned off. Fig. 13 shows how the AP estimates evolve, for the same 5 stars as shown in Fig. 8 for G=15. Notice how the algorithm gets stuck in limit cycles for some objects. Turning on update decay would help in these cases. Due to the larger noise, the calculated AP updates are much less accurate (the noise results in an error in the term  $\delta p$  in equation 4). This is probably the reason why the AP extrapolation limits are now enforced at many iterations for many stars. The upper limit on the AP step sizes were also imposed on a few occasions. The AP residuals are plotted in Fig. 14 and the corresponding summary statistics are as follows

	$\log g$	$\log(T_{\text{eff}})$	
$\overline{\delta\phi}$	-0.21	0.0045	ILIUM, G=20
$ \overline{\delta\phi} $	1.14	0.019	
$\sigma_\phi$	1.53	0.027	

There is now a mild systematic error in  $\log g$  and the  $\log g$  errors are very large. However,  $T_{\text{eff}}$  can still be estimated to an average uncertainty of 4.3% (the percentage error in  $T_{\text{eff}}$  is 2.3 times the error in  $\log(T_{\text{eff}})$ ). In comparison, the performance of the nearest neighbour algorithm using leave-one-out cross validation on the same data (i.e. all points with noise corresponding to G=20) is

	$\log g$	$\log(T_{\text{eff}})$	
$\overline{\delta\phi}$	-0.01	0.002	Nearest neighbour, G=20
$ \overline{\delta\phi} $	1.43	0.035	
$\sigma_\phi$	1.80	0.047	

Thus even though the nearest neighbour grid is twice as dense (yet noisy), it does slightly worse than ILIUM in  $\log g$  and more than 50% worse for  $\log(T_{\text{eff}})$ .

Fig 15 shows how the predicted spectrum (in black) varies with the iterations, for star 32, which has  $\log g=4.5$  dex and  $T_{\text{eff}}=5750$  K. These can be compared to the AP predictions at each iteration show in Fig. 13. As the predicted spectrum is predicted by the smooth forward model, it is much smoother than the noisy true spectrum: We see that ILIUM has some power to smooth over noise spikes in the original spectrum, particularly given that AP contribution clipping is used. (I did not show a corresponding spectrum prediction plot for G=15, because there the flux differences can hardly be seen.)

The goodness-of-fit (GoF) values (equation 7) are shown in Fig. 16. Many values are near unity,

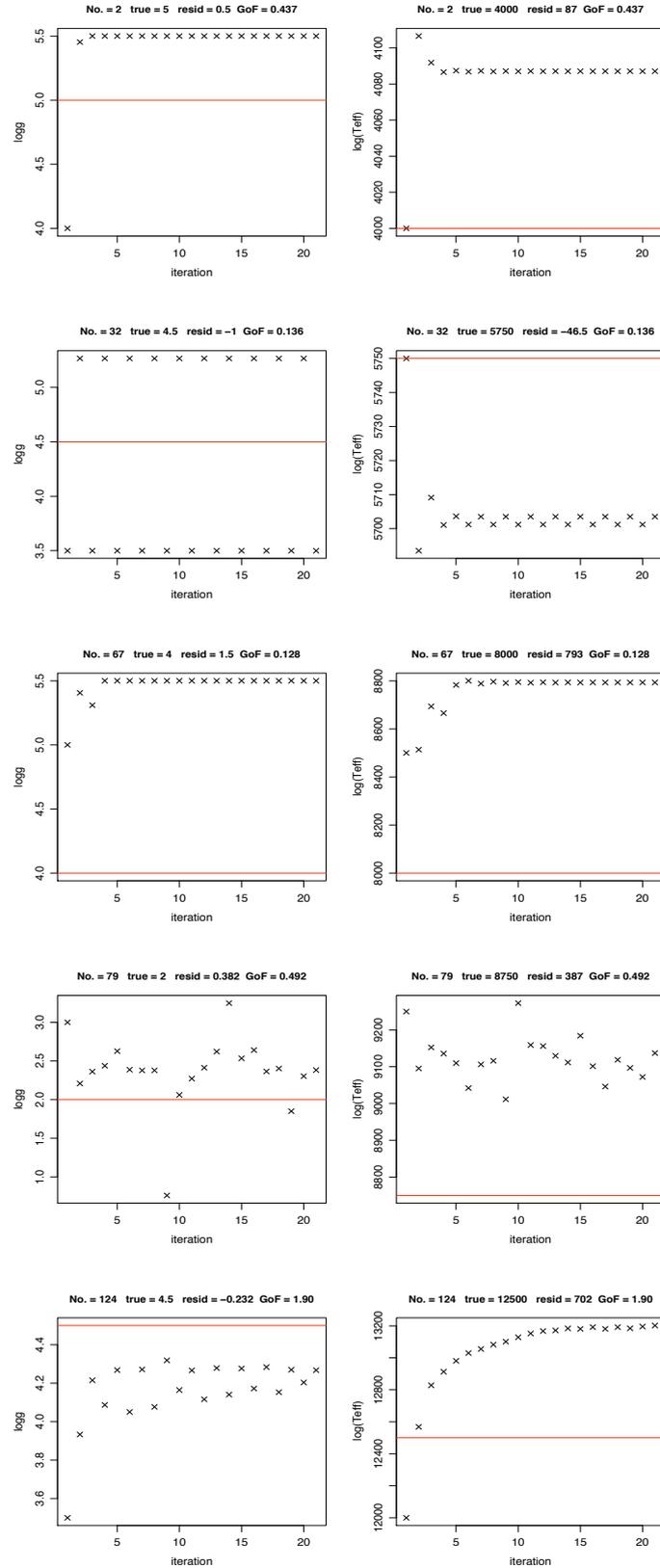


FIGURE 13: AP evolution for 5 stars in the test data set at  $G=20$  ( $\log g$  left,  $T_{\text{eff}}$  right). The true APs are written at the top of each panel pair and plotted as the red horizontal line. GoF is the  $\chi^2$  goodness-of-fit (equation 7).

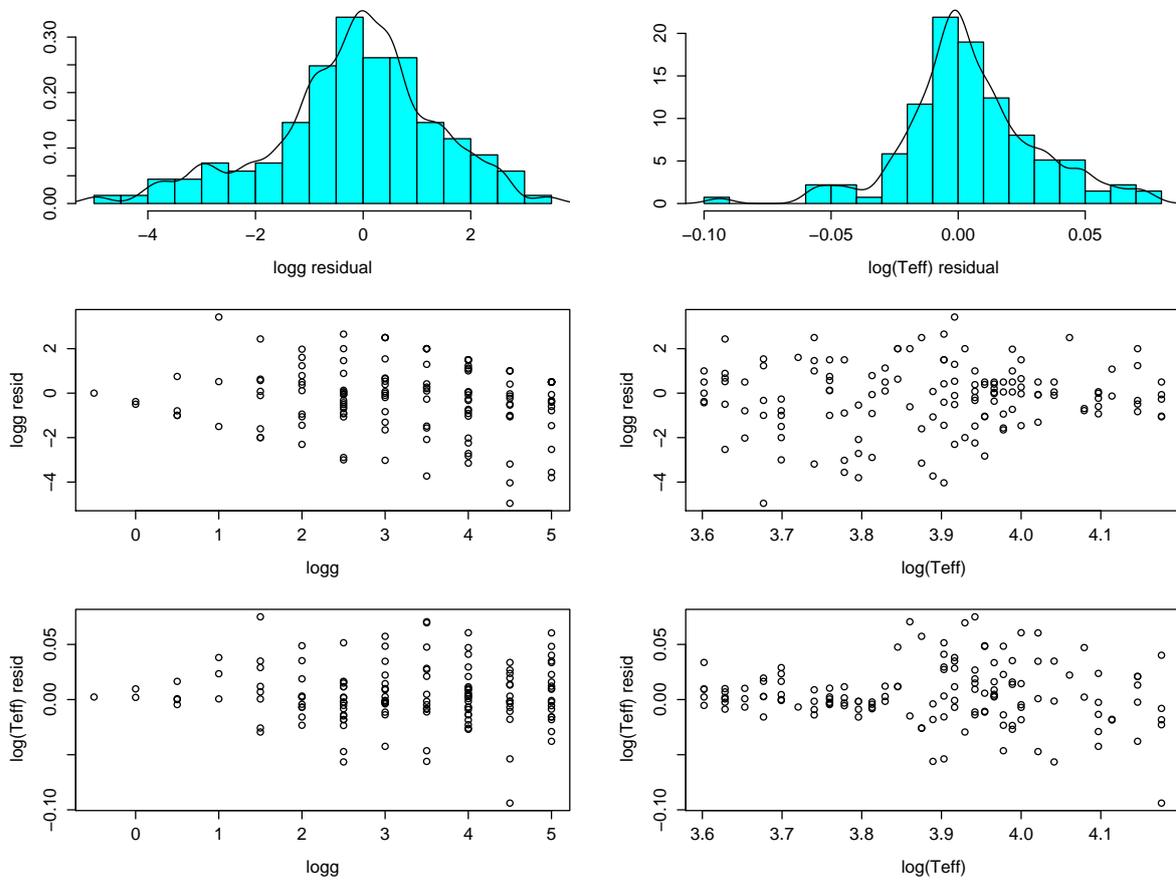


FIGURE 14: AP residuals for the test data at G=20

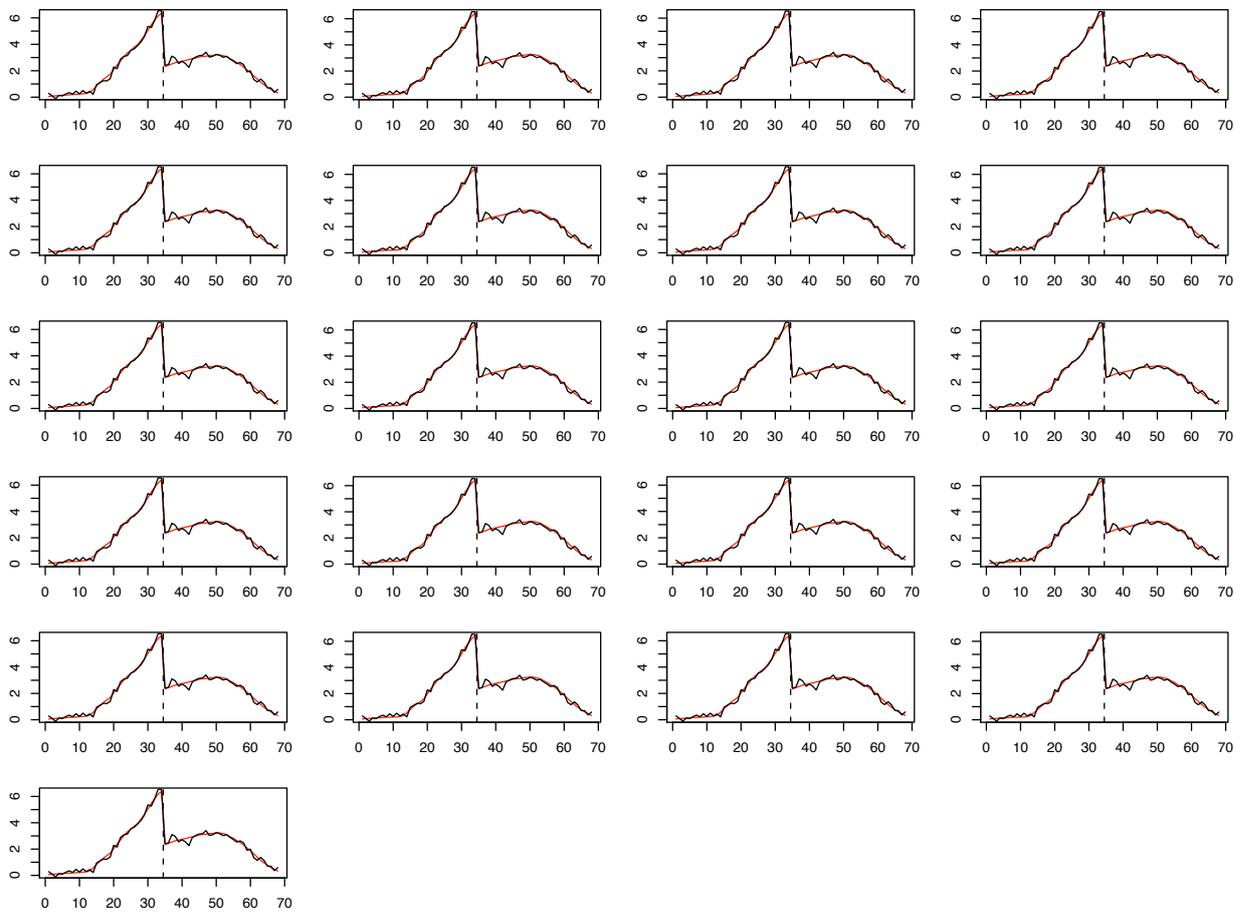


FIGURE 15: Predicted spectrum at each iteration (red) compared to the measured spectrum (black) for star number 32 ( $\log g=4.5$  dex and  $T_{\text{eff}}=5750$  K). The vertical dashed line marks the boundary between BP and RP.

indicating that the spectra are fit well by ILIUM, which is also a reflection of the forward model being a good approximation of the generative function. Large GoF values for unseen data would indicate that ILIUM has not made a good fit, which in turn would suggest that the object is not explained by the generative function. This could occur if the forward model (our approximation of the generative function) is built from stars and ILIUM is then presented with a quasar.

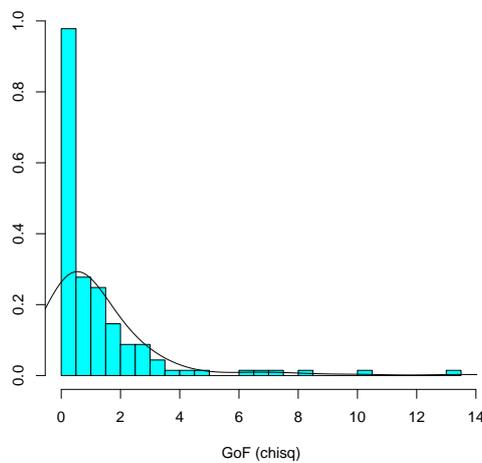


FIGURE 16: Distribution of the goodness-of-fit (reduced  $\chi^2$ ) values for the G=20 test data

### 3.5 Other results

Results at other magnitudes (all with the nominal algorithm parameters and no decay) are as follows<sup>3</sup>

<sup>3</sup>GOG cycle 3 BP/RP data were only provided at G=15, 18.5 and 20.0. To simulate noisy spectra at other magnitudes I scaled the flux and sigma spectra either from the G=15 data (to achieve G=16.2) or from the G=18.5 data (to achieve G=17.5 and G=19.0) assuming the “background limit”, i.e.  $\text{SNR} \propto \text{flux}$ . (Given the way the data are represented, this actually involves scaling the flux spectra and leaving the sigma spectra unchanged: see Mantis issue 3531.) This underestimates the SNR (is conservative) when scaling to fainter magnitudes (i.e. for G=16.2 and 19.0) and overestimates when scaling to brighter magnitudes (i.e. for G=17.5).

	$\log g$	$\log (T_{\text{eff}})$	
$\overline{\delta\phi}$	-0.0043	$-6.7e^{-5}$	ILIUM, G=16.2
$ \overline{\delta\phi} $	0.078	0.0013	
$\sigma_\phi$	0.12	0.0018	
	$\log g$	$\log (T_{\text{eff}})$	
$\overline{\delta\phi}$	-0.0028	$5.3e^{-4}$	ILIUM, G=17.5
$ \overline{\delta\phi} $	0.15	0.0025	
$\sigma_\phi$	0.218	0.0033	
	$\log g$	$\log (T_{\text{eff}})$	
$\overline{\delta\phi}$	-0.051	$9.8e^{-4}$	ILIUM, G=18.5
$ \overline{\delta\phi} $	0.35	0.0057	
$\sigma_\phi$	0.51	0.0080	
	$\log g$	$\log (T_{\text{eff}})$	
$\overline{\delta\phi}$	-0.032	0.0022	ILIUM, G=19.0
$ \overline{\delta\phi} $	0.55	0.0090	
$\sigma_\phi$	0.77	0.013	

At G=18.5, extrapolation limits were only imposed for two (of 137) objects. Around 10% of objects show limit cycles in one or other of the APs but generally of a small amplitude. The trend with magnitude is as we would expect. We continue to get good results down to G=19.0.

Finally, we now build a new, larger, grid (for training and testing) which additionally shows variance in [Fe/H]. We do not attempt to estimate it, so it represents a “nuisance parameter” which will make the estimation of  $T_{\text{eff}}$  and  $\log g$  harder. The [Fe/H] values range from  $-5$  dex to  $+1$  dex in steps of 0.5 dex. The full grid has 4361 stars of which we randomly select 75% for training and test the performance on the remaining 25%. The performance at G=15 is

	$\log g$	$\log (T_{\text{eff}})$	
$\overline{\delta\phi}$	-0.037	$7.4e^{-4}$	ILIUM, G=15 with wide [Fe/H] variation
$ \overline{\delta\phi} $	0.53	0.0058	
$\sigma_\phi$	0.90	0.0078	

The results are, of course, somewhat worse than that achieved when [Fe/H] is fixed at zero, being similar to what was achieved around G=18.5 or 19 for [Fe/H]. However, at  $T_{\text{eff}} > 8000$  K or so, the  $\log g$  residuals are just as small as the fixed [Fe/H] case, because metallicity variations have hardly any impact on the spectrum of hot stars. Around 10% of objects show limit cycles (sometimes now with periods of several iterations).

Results of ILIUM on  $T_{\text{eff}}$ , [Fe/H] estimation at constant  $\log g$  will be demonstrated in a future technical note (GAIA-C8-TN-MPIA-CBJ-043).

## 4 Summary and discussion

ILIUM produces very good results on end-of-mission non-oversampled  $G=15$  BP/RP spectra: 0.07 dex in  $\log g$  and 0.2% in  $T_{\text{eff}}$  (mean absolute errors) for stars with a wide range of  $T_{\text{eff}}$  (4000 to 15 000 K) and  $\log g$  ( $-0.5$  to  $5.0$  dex). The only reason it hasn't been extended to hotter stars is that I have not yet added the OB star library. The results are as reasonable as can be expected at  $G=20$ , with a  $T_{\text{eff}}$  accuracy of 4%, while  $\log g$  is quite poor at 1.1 dex. Of course, these are all “synthetic” errors, in the sense that there is no mismatch between the training grid and the test data.

The results at  $G=15$  are several times better than nearest neighbours. This demonstrates that ILIUM can significantly improve upon the AP resolution limit set by the grid, which sets the lower limit on the accuracy for the nearest neighbours method.<sup>4</sup> The results validate the forward modelling procedure and the iterative approach to local linear interpolation. At  $G=20$  ILIUM still has superior accuracy, but only around 50% better, presumably because the noise level is now similar in size to the differences between spectra at neighbouring grid points.<sup>5</sup>

Central to ILIUM is the idea of the forward model, which predicts the photometry given estimates of the APs. This is always a unique function, and can be calculated independently for each band, so in theory is a simple smoothing problem to fit it. It is complicated in practice, however, by the presence of strong and weak APs, which have very dissimilar impacts on the variance in the flux data. This is solved by splitting the forward model into two components.

Implicit in the method is the use of the local sensitivities to weight the contributions of the flux bands to the AP updates at each iteration. (The sensitivities are the derivatives of the fluxes with respect to the APs and are local in the sense that they vary with the AP values themselves.) Bands with low sensitivities automatically make a small contribution to the APs. (In principle, therefore, we may add any number of bands and ILIUM will down weight the irrelevant ones.) The two component approach to the forward model is critical here, because it ensures that the flux variation over the weak AP is smooth and hence that its sensitivities can be estimated accurately.

Using the forward model we can provide a goodness-of-fit even for unlabelled data, thus permitting detection of outliers which do not fit the model. As the forward model can be extrapolated, sensible AP estimates and GoF measures can be obtained beyond the limits of the grid. Finally, the linearization in ILIUM allows us to estimate individual AP uncertainties. Comparison of these with the true residuals demonstrates these estimates to have the right order of magnitude,

<sup>4</sup>One could average over  $k$  nearest neighbours, thereby reducing the variance, but at the cost of increased bias in the AP errors (this is the standard bias–variance tradeoff).

<sup>5</sup>If the error ellipsoid were large compared to the grid spacing, then we could simply use nearest neighbours and estimate the APs as the of the neighbours inside the error ellipsoid, with the standard deviation providing an error estimate (although finding them all is time consuming). ILIUM brings the most benefit in the low grid density / low noise limit, although it may always be preferable on account of its sensitivity weighting.

with 40% of the uncertainties lying within a factor 2 of the true residual.

## 5 Future work

The major limitation of ILIUM at present is that it only works on two APs, currently  $T_{\text{eff}}$  and  $\log g$ , although it could be any two. Extension to  $N_S$  strong APs should be simple by extending the 1D strong forward model to be an  $N_S$ -D interpolation. Provided these strong APs (e.g.  $T_{\text{eff}}$  and  $A_V$ ) explain a similar level of variance in the flux data, then a good fit via standard regression methods should be possible. If we still had just one weak AP, then we need just 1D weak forward models, but now one at each *combination* of the strong APs. With  $N_W$  weak APs, we could presumably replace each of these with a  $N_W$ -D interpolation, which requires that the set of weak APs explain a similar level of variance as each other (to within a few factors of 10). The key, then, is to be able to split the APs into strong and weak categories, according to the variance they explain in the spectra. From previous work and experience, it seems that  $T_{\text{eff}}$  and  $A_V$  on the one hand, and  $\log g$  and  $[\text{Fe}/\text{H}]$  on the other, fit into these two categories. ( $R_V$  and  $[\alpha/\text{Fe}]$  still need to be examined.) The grid would need to be semi-regular, i.e. show a range of values of all weak APs for each point in the grid of strong APs. There is no requirement for the strong APs to be on a regular grid.

As ILIUM shows good convergence properties – even without using decay – it may be able to identify multiple solutions if initialized at various (random) AP values (rather than the nearest neighbour). In practice we would want to ensure that the same solution is located more than once (within some tolerance). This has not yet been tested because I do not expect such degeneracies with just  $T_{\text{eff}}$  and  $\log g$ .

To go much further with ILIUM – so that it can efficiently handle much larger amounts of data as well as multiple APs – it will be necessary to convert it into a high level language (Java if it is to be used for Gaia). The code is quite simple and can make use of standard matrix and regression libraries. A fast k-D tree implementation of the nearest neighbours initialization will also be desirable as the grid grows.

## References

- Bailer-Jones C.A.L., 2008, *A method for exploiting domain information in astrophysical parameter estimation*, ASP Conf. Ser. 394, 169
- Bailer-Jones C.A.L., Smith K.W., Tiede C., Sordo R., Vallenari A., 2008, *Finding rare objects and building pure samples: Probabilistic quasar classification from low resolution Gaia spectra*, MNRAS 391, 1838

Re Fiorentin P., Bailer-Jones C.A.L., Lee Y.S., Beers T.C., Sivarani T., 2007, *Estimating stellar atmospheric parameters from SDSS/SEGUE spectra*, A&A, 467, 1373

Smith K., 2008, *How to use oversampled simulated BP/RP data*, GAIA-C8-TN-MPIA-KS-007

Sordo R., Vallenari A., 2008, *Description of CU8 cycle 3 simulated data*, GAIA-C8-DA-OAPD-RS-002

Tiede C., 2008, *GWP-S-822 - cycle 4 - GSP-PHOT - Software Requirements Specication, Software Design and User Manual*, GAIA-C8-SP-MPIA-CT-008-4

Tsalmantza P., Kontizas M., Bailer-Jones C.A.L., Rocca-Volmerange B., Korakitis R., Kontizas E., Livanou E., Dapergolas A., Bellas-Velidis I., Vallenari A., Fioc M., 2007, *Towards a library of synthetic galaxy spectra and preliminary results of the classification and parametrization of unresolved galaxies from Gaia*, A&A, 470, 761