

# A platform independent, parallel version of ATLAS12

**Katharina M. Bischof**

Institut f. Astronomie, Universität Wien, Türkenschanzstr. 17, 1180 Wien, Austria  
email: bischof@astro.univie.ac.at

**Abstract.** Opacity sampling with ATLAS12, the stellar atmosphere code developed by R.L. Kurucz, cannot always be carried out with the desired frequency or depth resolution because of the limited computing power of even the fastest monoproductors. There are also known problems of portability which make it difficult to run ATLAS12 with various compilers on different operating systems.

We first created a Fortran77 version that can be compiled using the g77 compiler, a useful feature for astronomers having no access to VMS compatible Fortran compilers. As a further step, ATLAS12 was successfully ported to Ada95, an object-oriented parallel language. ATLAS12 is now platform independent, split up in modules and running in parallel on multi-processor machines. Any limitations as to the maximum number of depth and frequency grid-points and the number of lines that can be treated have been pushed far beyond what is possible with the original version.

We intend to incorporate the continuous opacity routines of ATLAS12 in our new CAMAS code for magnetic atmospheres (but also in the existing COSSAM and CARAT codes) to be able to compare our results with Kurucz's de facto stellar atmosphere standard.

**Keywords.** Stars: atmospheres

---

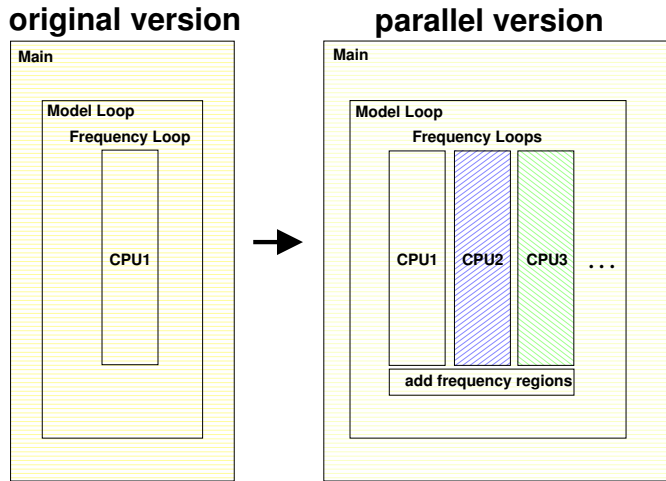
## 1. Introduction

ATLAS12 is a stellar atmosphere code developed by R. L. Kurucz. This opacity sampling code can be used as a (non-magnetic) standard for stars with exotic abundances Kurucz (1996), albeit without stratification. In its original version the code is able to calculate models with 30000 points in frequency and a depth-grid with up to 72 points. For some applications (e.g., convective models) a higher resolution in depth would be desirable.

We decided to port the code to the object-oriented parallel language Ada95 because of the need for:

- improved frequency resolution
- improved depth resolution
- ensuing necessity of parallel computation
- modularisation, portability, readability
- compatibility with our Ada95 based codes
- solution for the Little / Big Endian problem

One goal was to incorporate the ATLAS opacities into our codes. Thus we needed a modularised version of the code with well defined interfaces.



**Figure 1.** ATLAS12 is well suited for parallelisation since the frequency integration can easily be divided into subintervals.

## 2. Resolution

The obvious first step towards better resolution was to get rid of hard coded maximum vector lengths. Fixed array lengths have the big drawback as they usually fill more of the memory than necessary. This problem is easily overcome by the use of dynamic arrays. The maximum required length of all arrays used is evaluated at the beginning of the program by scanning the model file.

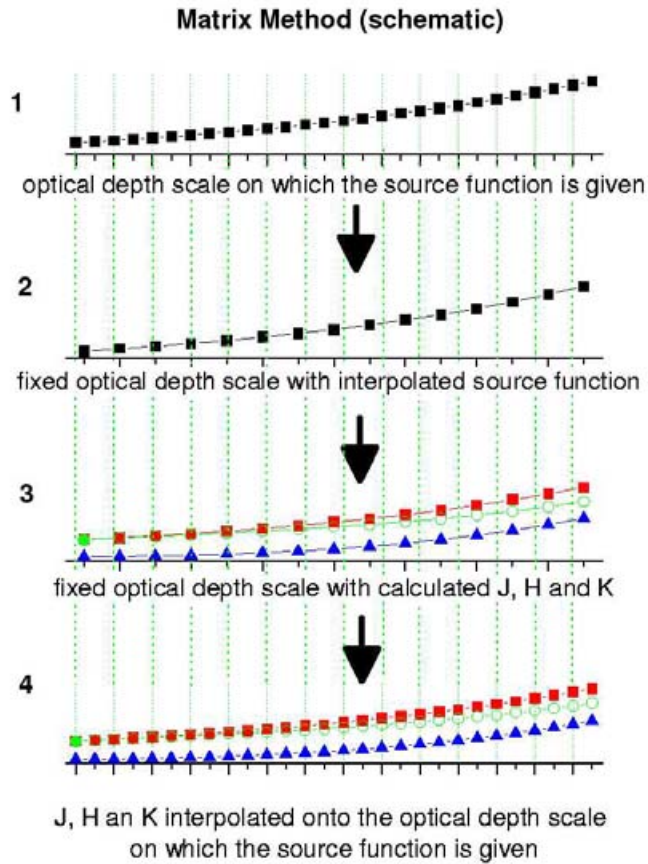
But enlarging the number of depth points is not enough to increase the achievable resolution of the model. Originally the ATLAS code calculated the mean intensity ( $J$ ), flux ( $H$ ) and radiation pressure ( $K$ ) by using the matrix method developed by Kurucz (1969, 1970). This method involves so called "integration matrices" which depend on the optical depth point distribution. ATLAS12 uses pretabulated integration matrices and, therefore,  $J$ ,  $H$  and  $K$  can only be derived on a fixed optical depth scale. This requires the interpolation of the source function onto this grid and the interpolation of the results onto the optical depth scale at a given frequency. Consequently the maximum number of depth points is also limited by the step width of the fixed optical depth scale of the integration matrices as one will obtain only "empty resolution" by interpolating a large number of points into a coarse grid.

### There are two methods to overcome this problem:

- Calculate integration matrices with the required step width for  $J$ ,  $H$  and  $K$  at the start of the program (see Figure 2).
- Use a Feautrier scheme (Feautrier 1964, Rybicki & Hummer 1991) to calculate  $J$ ,  $H$  and  $K$  (see Figure 3).

A third possibility, calculating such integration matrices at every frequency point for the given optical depth scale, turned out to be slow and unstable at the same time.

The Feautrier scheme has the advantage, that it does not need any pretabulation on an optical depth scale. Thus it does not limit the number of depth points. The accuracy (and the CPU-time) of the method depends on both the number of depth points and of angle points.

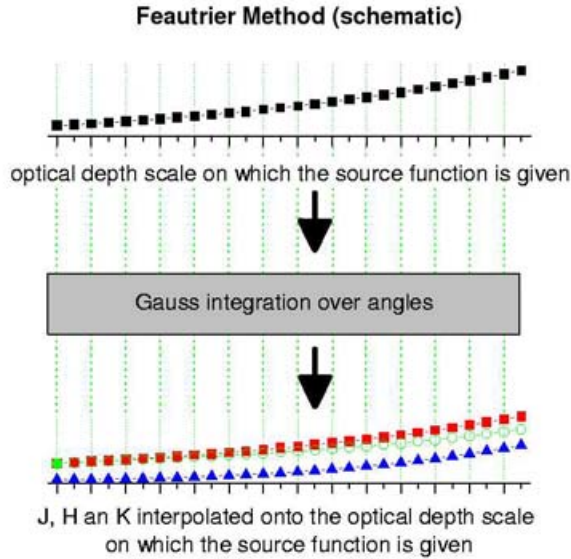


**Figure 2.** The use of the Matrix Method involves the interpolation of the source function (1) onto the grid on which the integration matrices are pretabulated (2). The resulting J, H and K vectors (3) are interpolated onto the optical depth scale at the frequency (4). Thus the maximum number of depth points that will make sense is limited by the spacing of the pretabulated optical depth scale.

### 3. Parallelisation and runtime

The calculation of stellar atmospheres is well suited for parallel computation. About two-thirds of the computing time is spent on frequency integration (see Figure 4 and Table 1). We simply need to subdivide the frequency range into subintervals and pass these to different CPUs. The results of these computations are summed up and sent back to the model loop (see Figure 1).

For a comparison of run-times we have executed 30 iterations on a 12000 K ATLAS9 model with 72 depth points and 30000 frequency points. We also changed some abundances (C,N,O -1 dex and Cr, Fe, Ni +1 dex) to test the convergence behaviour of our codes.



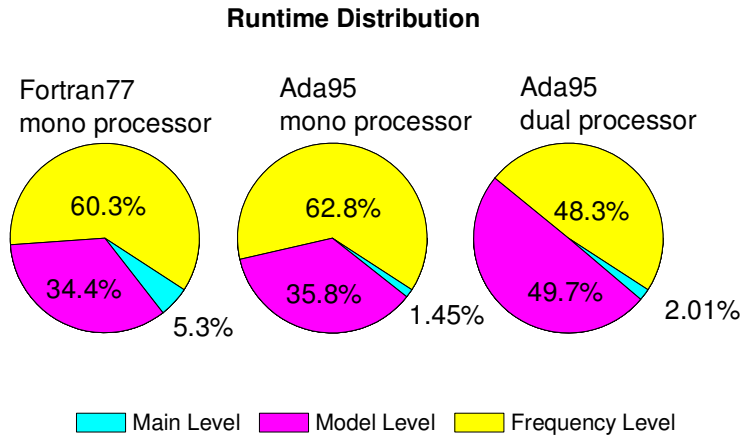
**Figure 3.** The Feautrier scheme calculates a flux-like and an intensity-like quantity on the given optical depth scale for rays with different angles to the line of sight. A Gauss integration is used to obtain J, H and K. The accuracy of J, H and K can be improved by increasing the number of angles. This method does not carry out any interpolations.

Runtime Table						
	total	Main Level	Model Level		Frequency Level	
	mm:ss	mm:ss	total mm:ss	single iteration mm:ss	total mm:ss	single iteration mm:ss
g77, clone	11:38	0:37	4:00	0:08	7:01	0:14
Ada95, clone	19:34	0:17	7:00	0:14	12:17	0:24
parallel (dual processor)	14:05	0:17	7:00	0:14	6:48	0:14
Ada95, Feautrier, 6 points	23:11	0:17	7:00	0:14	15:54	0:32
parallel (dual processor)	16:00	0:17	7:00	0:14	8:43	0:17
Ada95, Feautrier, 10 points	26:52	0:17	7:00	0:14	19:33	0:39
parallel (dual processor)	17:53	0:17	7:00	0:14	10:36	0:21

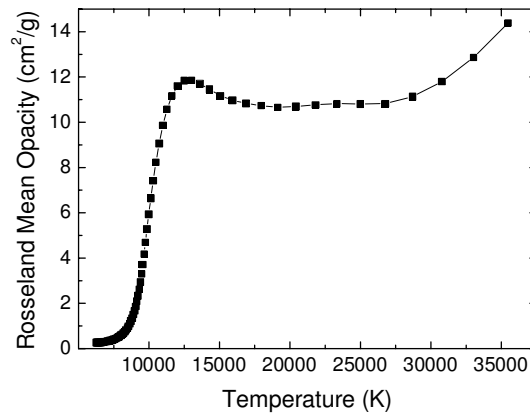
**Table 1.** The calculations were carried out on an Intel Xeon (2.660 MHz) dual processor machine. We used the compiler switches -O3 for g77 and -O3 -gnatp for the Ada95 compiler, to get comparable optimisation. At the moment the Ada95 code needs more time than the g77 code in the model level because the file handling (in LINOP) takes more time if formatted in/output is used.

The code versions compared were:

- a Fortran version which is the original version with some minor changes to make it compilable with the g77 compiler
- an Ada95 version using the matrix method to derive J, H and K
- an Ada95 version using the Feautrier method to derive J, H and K



**Figure 4.** This plot shows that the original code and its Ada95 clone spent about two thirds of the computation time on the frequency integration. The rightmost graph shows how much a second processor can decrease the duration of the frequency integration (see also Table 1).



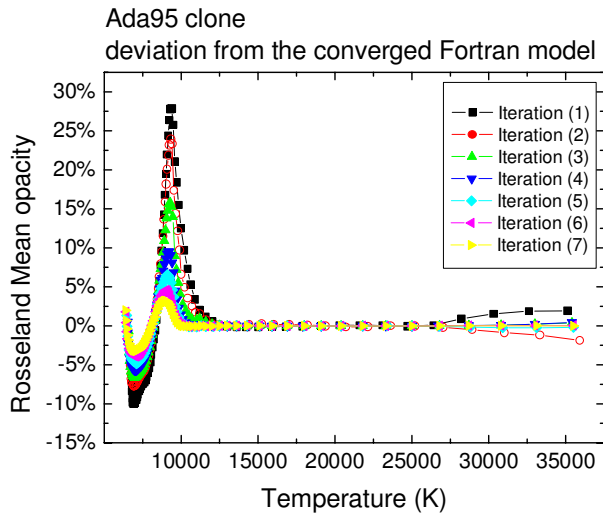
**Figure 5.** The converged Fortran model for a 12000 K star with 72 depth points, 30000 frequency points, decreased C,N,O (-1 dex) and increased Cr, Fe, Ni (+1 dex) abundances.

When we compiled the codes we used the `-O3` switch for `g77` and `-O3 -gnatp` for the Ada95 compiler to get comparable optimisation. We used a multi-processor machine with two Intel 2660 MHz CPUs to compare the run-times of the different versions of the code.

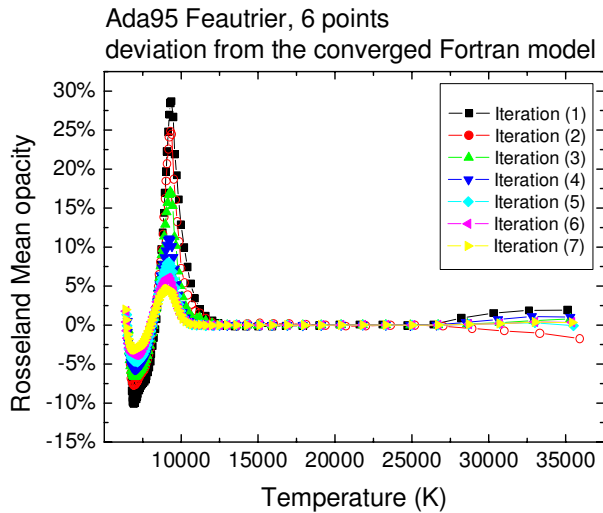
At the moment the Ada95 code needs more time than the `g77` code in the model level because the file handling in LINOP takes more time when formatted in/output is used. The same can be said about the frequency level.

#### 4. Results and current status

Our calculations showed that our 12000 K model converged to the same values with comparable convergence speed. Figures 5 to 8 show the decreasing deviations from the



**Figure 6.** The Ada95 clone deviations from the converged Fortran model



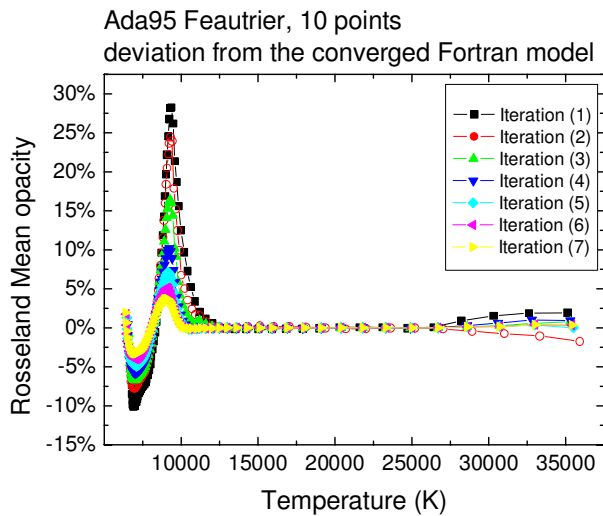
**Figure 7.** The Feautrier model with 6 points deviations from the converged Fortran model

converged model. After 7 iterations all the parameters of the models are within 5% relative deviation from the converged model.

The advantages of the Ada95 code are its aptitude for multiprocessor machines and the possibility to calculate more depth points. The Ada95 code is not yet fully optimised as regards CPU time since so far we have focused on portability and the establishment of well defined interfaces to incorporate some parts of the code into our other codes.

## 5. Conclusion

Our version of the code will be useful for the study of phenomena which need improved resolution in depth or frequency. The new code is parallel and fairly platform-independent. We have tested it with various Linux distributions (Debian, Redhat and Suse) and various processor types (Intel Xeon / AMD Athlon).



**Figure 8.** The Feautrier model with 10 points deviations from the converged Fortran model

### Acknowledgements

The author thanks M. J. Stift and R. L. Kurucz for useful discussions and P. Fierlinger for taming the printer. This work was supported by the Austrian Science Fund (FWF) Project P16003, the Austrian Research Association (OeFG) and an IAU travel grant.

### References

- Feautrier, P. 1964, in: E.H. Avrett, O. Gingerich & C.A. Whitney (eds.), *SAO Special Report No. 167* 80
- Kurucz, R.L. 1996, in: S.J. Adelman, F. Kupka & W.W. Weiss (eds.), *Model Atmospheres and Spectrum Synthesis* ASP Conference Series **108**, 160
- Kurucz, R.L. 1970, *SAO Special Report No. 308*
- Kurucz, R.L. 1969, *ApJ* **156**, 235
- Rybicki, G.B. & Hummer D.G. 1991, *A&A* **245**, 171 (Appendix A and B)