

Lester 版 atlas12 (L-atlas12)

星学館・加藤 賢一

1. 概要と今回の作業内容

■ Lester & Neilson(2008) の atlas_os

Kurucz (<http://kurucz.harvard.edu/>) の恒星大気モデル ATLAS9、ATLAS12 を Lester & Neilson (2008、<http://www.astro.utoronto.ca/~lester/>) が fortran2003 に従って書き変えた atlas_odef、atlas_os は最新のフォートラン・コンパイラに対応した ATLAS9、ATLAS12 の新しいバージョンである。Kurucz の元のプログラムを Castelli (<http://wwwuser.oats.inaf.it/castelli/>) や Bischof (Katharina M. Bischof, Inst. f. Astronomy, Vienna) が Fortran77 に対応すべく書き変えた版があるが、これら全てが今や新しい gfortran や intel-fortran コンパイラに対応できなくなっていることから、貴重なバージョンと言える。そして、Lester らの版は単に fortran2003 に対応させたというレベルにとどまらず、積極的に fortran2003 の新しい文法に合わせて全面的に書き変えているため厳密なプログラム化となっていて、信頼性が格段に上がっている。

さらに、ATLAS9、ATLAS12 を球状大気に拡張した S-ATLAS9、S-ATLAS12、スペクトル合成法の synthe も同梱されており、ATLAS モデルの拡張新版として現在の決定版と言えるのではないかと思う。

■ L-atlas12 のバグと修正

ここでは Lester & Neilson による ATLAS12 大気モデル計算コードを L-atlas12 と書くことにする。

さて、上のように記したが、Lester が公開しているソース・プログラムにはバグがあり、そのままではコンパイルできない。実は2015年には修正なしに intel-fortran でコンパイル、実行できたが、今回、2023年版の intel-fortran コンパイラを採用したところ、コンパイルできないことがわかった。gfortran でも同様に、最新の gfortran コンパイラではコンパイルできなかった。これらのバグは文法チェックが厳格になった新しいコンパイラによって見つかったが、中には論理的な誤りもあり、それは危険な誤りであることが判明した。

コンパイラでは見つからないバグに元のソースファイルから書き変える時に起きた単純な、しかし危険な誤りがあった。たとえば subroutine xlinop で rewind 19 文が抜けていたのはその例で、水素を含む軽元素線データを2回目以降の逐次近似では読み込まずに進めてしまい、最終結果に水素等の影響が抜けてしまっていた。今回はソースファイルを全面的に見直したわけではないので、こうした欠陥がなお残っていると覚悟しておかねばならないようである。ただし、大気モデルを作成する上では温度・圧力構造に大きな差が出ているようではないので、通常の用法ではさほど気にすることはないと思われる。

バグの種類と対応は表1のとおりである。

さらにコンパイラが警告を出している箇所があるが、これらは相当数が残ったままである。

表1. バグとその修正

バグの箇所	内容	対処
sub xlinop	軽元素線データ nltelines.bin を読み込む際に先頭位置にポインタを戻す命令 <code>rewind 19</code> が抜けてい	致命的な誤り。書き換え時の見落とし。 <code>read(19)</code> の前に追記。

	た。	
sub xlinop	if(iw<1 .or. waveset(iw)<wcon) iw<1 の場合、waveset(iw)は境界エラー を起し、この if 文自体がエラーとなる。	2015 年では通った。 2つの if 文に分ける。
sub xlinop	do if(iw < max(nu - i, 1)) exit i が定義されていない。	2015 年では通った。i は不定で、確かに誤り。書き換え時 の見落とし。 do i = 1, 2000 と Kurucz に従い修正。
sub josh_r	do i_mu = 1, n_mu i_mu は 2 重定義。	2015 年では通った。 i_mu は arr_tlu(ij, 1:ndepth, i_mu) という形で既 に配列の指標として使われている。 k_mu とした。
fun mapcs	n_last の初期値エラー	2015 年では通った。 n_last の初期値が未定義で、コンパイラは 0 を与える。 これが配列の指標になっているのでエラーとなる。 初期値を n_last=81 とした。
sub solvit	intent(inout)::b(:)がエラー	2015 年では通った。 下記のように修正。 intent(out)::b(:)
sub si2op	index_t(ij)の初期値エラー	上の n_last と同じ誤り。
sub josh	knu(1) = sum(ck(1:nxtau) * xs(1:ndepth))のエラー	コンパイルは通るが、境界越えを起こす可能性がある。 xs(1:ndepth) は xs(1:nxtau) に。

■軽元素線データ nltelines.bin の扱い

HI, HeI, HeII, CI, CII, OI, NaI, MgI, MgII, AlI, SiI, SiII, KI, CaI, CaII の各元素のデータを収納している重要なファイルが nltelines.bin である。元データは nltelines.asc。

(1) 変換プログラム

元データ nltelines.asc から rnlte.for や rnlteall.for を用いてバイナリデータにするように Kurucz は指示しているが、これらには別の意味不明のデータ・ファイルが必要とされ、このままでは変換できない。そこで、Castelli が与えている nltelinesascctobin.for を使うのが良い。

(2) 錯綜するバイナリデータ・ファイル

ATLAS12 では nltelines.bin を下記のように読み込む。

```
DO 900 LINE=1,100000
READ(19,END=901)WLVAC,ELO,GF,NBLO,NBUP,NELION,TYPE,NCON,NELIONX,
1GAMMAR,GAMMAS,GAMMAW,IWL
```

一方、rnlte.for や nltelinesascctobin.for では

```
1253 write(19)wlvac,elo,gf,nblo,nbup,nelion,type,ncon,nelionx,
gammar,gammas,gammaw,nbuff,lim
```

としてバイナリデータを作る。

最後の 2 つに違いがあり、こうしてできた nltelines.bin を ATLAS12 にかければ当然ながらエラーとなる。Kurucz

や Castelli が提供しているバイナリファイルには要注意である。

(3) アスキーファイル `gfnlte.asc` を採用

バイナリファイルは容易に内容を見ることができないので、それが ATLAS12 に適合するものか、にわかには判断できない。そこで、今回、内容が即座に見分けられるように、バイナリではなく、アスキーで作り変えて名称を `gfnlte.asc` とし、最後の **IWL** に **nbuf** だけを採用して ATLAS12 に適合するようにした。

■ ソースファイルの統合とコンパイル法

Lester が提供しているソースファイルはモジュールが 44、プログラム本体が 8 と 52 本に分かれている。筆者にとっては扱いにくいので

```
1.module.for
2.programs.for
```

の 2 つに統合した。デバッグ・オプションを付ける場合は以下のように設定した。

```
• ifort -qmkl -check all -gen_interfaces -fpe0 -ftrapuv -traceback -o L-atlas12.exe
1.module.f 2.programs.f
```

win-ifort によるコンパイルでは Linux-ifort より文法チェックが厳しいようで、`intent` 属性の指定で混乱があると指摘があり、それを修正してコンパイルした。ただ、今回は Linux-ubuntu 上の実行ファイルだけを添付しており、windows 版は省いた。

2. システム構成、テスト計算の結果

■ システムとコンパイラ

(1) os、コンパイラ

OSとしてはLinux、具体的にはubuntu16.04を採用した。Windows10でも良いが、実行時間がubuntuの3倍以上要するためubuntuとした。

コンパイラはintel-fortran (ifort、バージョン: 2021.8.0)。しばらく有料版しか提供されていなかったが、最近また無料版が利用できるようになった。

ubuntu14、ubuntu18ではintel-fortran (ifort と略) のインストールに失敗した。最新のubuntu22でもifortとの相性は良いが、ubuntu16に比べて厳格になっていて使い勝手が悪く、Windowsとのファイル交換も素直にできないのでubuntu16を採用した。

gfortranではL-atlas12のソースファイル (ifortでコンパイル可) はコンパイルできなかった。gfortranはifortに比べて変数の扱いが厳密であることや、ifortが暗黙の了解としている事項を認めず、注意深い取り扱いが求められることに起因すると思われる。

Linux: Ubuntu 16.04 LTS

ifort/Linux: l_BaseKit_p_2023.0.0.25537.sh, l_HPCKit_p_2023.0.0.25400.sh

ifort/Windows10: visual_studio_2019, w_BaseKit_p_2023.0.0.25940.exe,
w_HPCKit_p_2023.0.0.25931.exe

(2) ハードウェア

筆者の構成は以下のとおり:

Epson AT993E (商品名 Endeavour) - intel i7-6700 CPU 3.40GHz RAM 16GB
Win10 Home + VMware Player 16 + ubuntu 16.04

以上のシステムによる15回の逐次近似を行った場合の実行時間を表3にまとめておいた。

表2. 実行時間比較—ubuntu上とwindows上

モデル (Teff log g)	実行時間 (分) ifort	デバッグモード	Windows ifort
3000K +0.0	4.5		
4260K +0.9	2.0		14.4
10000K +4.0	1.7	9.8	
20300K +2.7	1.7		

(3) 計算結果について

図1にubuntu上での計算結果とwindows上での結果 (フラックス、nm 当り) を示しておく。両者に結果の違いはないが、表2のとおりでWindows上では7倍の計算時間だった。

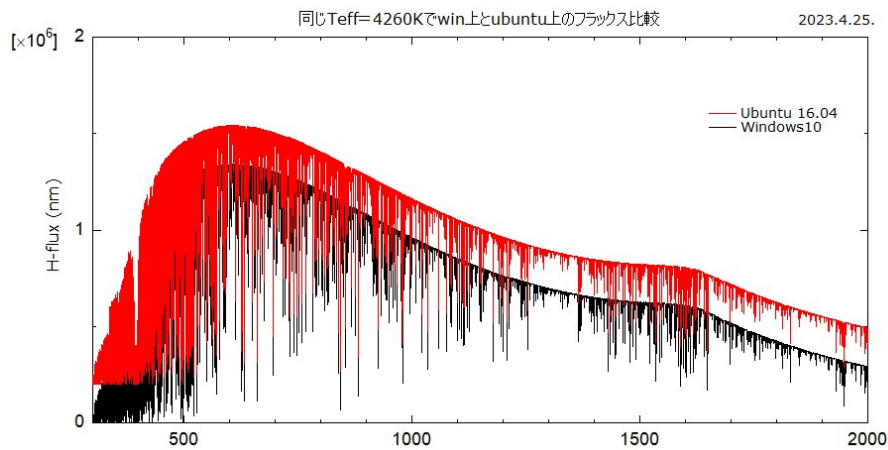


図 1. Ubuntu での結果と Windows 上での結果の比較

■ gfortran での不適切な振舞い

今回は採用していない gfortran コンパイラだが、Windows 版ではここでのソース・プログラムをコンパイル、実行することができる。つまり同じ gfortran と称しながらも Linux 版 gfortran コンパイラとは大きな違いがあり、両者は別物と見た方が良さそうである。

以下で紹介するように Windows 版 gfortran コンパイラの結果は ifort の結果と異なっていることと、これまでの ATLAS9 の結果と乖離していることから不適切なコードが出力されたものと思われるので、ここでは gfortran を採用していない。

	Windows10	Linux
ifort	○	○
gfortran	△	×

同じソース・プログラムに同じ大気パラメータを採用し、ifort と gfortran によって計算した結果を比べてみると、下記の図 2、図 3 のように明らかな差が出てしまった。

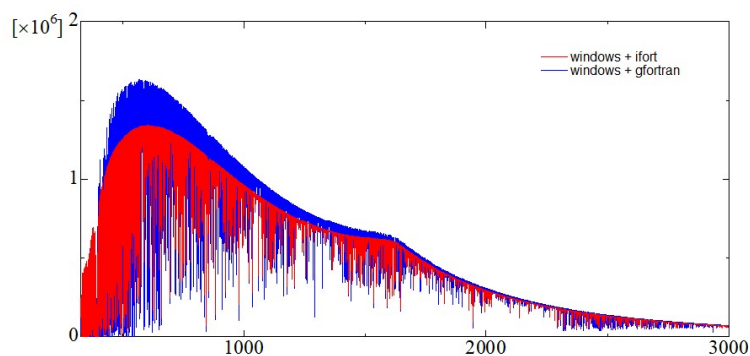


図 2. コンパイラによる結果の違い (1) フラックス 4260K

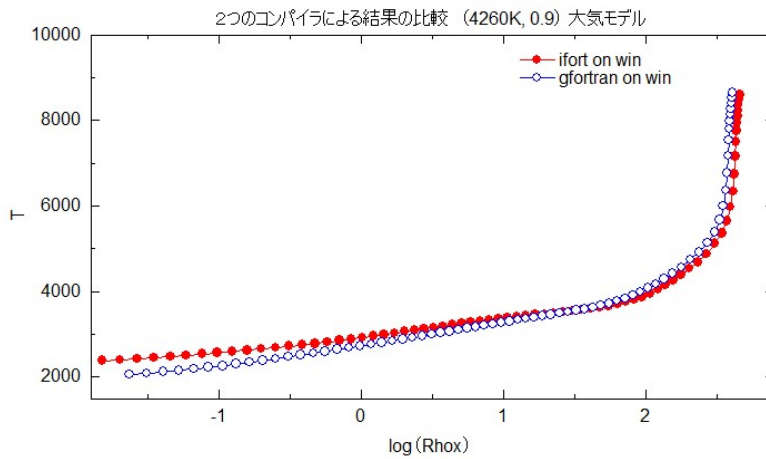


図3. コンパイラによる結果の違い (2) 温度分布 4260K

ここまででは優劣をにわかに判定することができないので、ATLAS9 による(3500K, 1.0)モデルから ifort と gfort によって同じ (3500K, 1.0) のモデルを計算してみた。その結果が図4である。ifort による結果は元データに酷似しているが、gfort による結果は両者からひどくずれている。それも線形成領域での乖離が大きく、フラックスにも大きな違いが生まれていることが予想される。同じパラメータでの計算であり、元の ATLAS9 モデルから大きくずれることは考えにくいところから、gfort による計算は適切ではなかったと判断される。

これが単に処理系の違いに起因するものか、あるいはコンパイルできたとしてもまだ論理的な誤りが残っていることに起因するものか、その他にあるのか、不明である。

なお、ここで殊更 gfortran に言及したのは Castelli や Bischof 版では gfortran によるコンパイルが ifort 同様に扱われているからである。

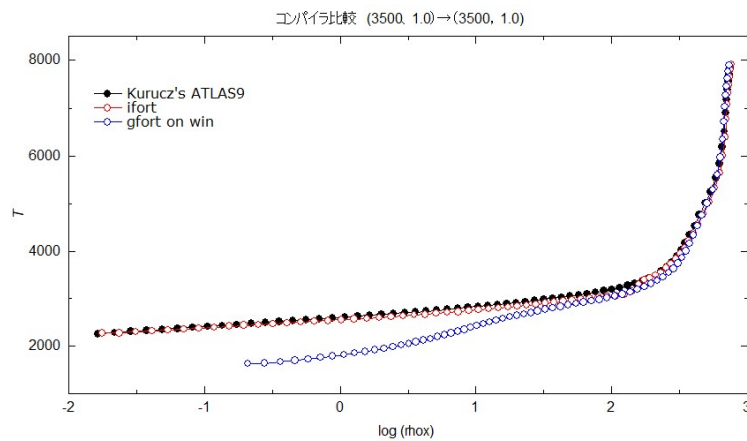


図4. コンパイラによる結果の違い (3) 温度分布 3500K

3. 動かし方

■ 動かし方

Ubuntu 上に計算実行用フォルダーを作り、次の諸ファイルを置き、

```
#!/L-atlas12.exe
```

として実行。

```
L-atlas12.exe
```

表3の線データ

分子データ molecules.dat

初期値ファイル input.dat, model.dat

■ スペクトル線データ

ATLAS12 で使用される線データをまとめておく。

表3. ATLAS12 で使用される線データ

名称	内容	本数	サイズ
gfnlte.asc	HI, HeI, HeII, CI, CII, OI, NaI, MgI, MgII, AlI, SiI, SiII, KI, CaI, CaII	38,298	4 MB
lowliness.bin	gfnlte.asc で扱う元素を除いた全元素の4階電離イオンの線	7,915,144	494 MB
highlines.bin		2,566,183	160 MB
diatomic.bin		1,975,559	123 MB
tiolines.bin		9,436,125	590 MB
h2o.bin		16,478,089	515 MB

■ ファイル構成一覧（線データは除く）

input.dat におけるオプションによっていくつかのファイルや中間ファイルが作成される。中間ファイルは計算の最後に削除される。

表4. 関係ファイル一覧（線データは除く）

種別	名称	用途等	サイズ他
実行プログラム	L-atlas12.exe	実行用ファイル	2.5 MB
分子線データ	molecules.dat		
入力データ	input.dat	入出力用パラメータ	
	model.dat	出発となる大気モデル。計算する T_{eff} , $\log g$ et al. に近い atlas9 モデルなどを採用。	
出力データ	print.txt	計算の途中経過等	
	punch.txt	計算された大気モデル	
	surf.txt	input.dat のオプションによって作成	

■ input.dat 用各種パラメータとその設定法

多くのパラメータが用意されているが、あまり試していないので実態が良くわからない。

次のマニュアル類を参照願いたい。

•Kurucz, R. L. 1970, 'ATLAS:A computer program for calculationong model stellar atmopsheres', SAO Special report No.309 (Kurucz のホームページ <http://kurucz.harvard.edu/> から)

•Castelli, F. 1988, 'Kurucz's models, Kurucz's fluxes and the ATLAS code. Models and fluexes available at OAT', Osservatorio Astronomico di Trieste Pubblicazione No.1164 (Castelli のホームページ <https://wwwuser.oats.inaf.it/castelli/> から)

■ 3種類の温度補正ルーチン

温度補正のための方式が3種類用意されていて input.dat での指示で選択することができる。Kurucz の方式、Feautrie の方式、Rybicki 方式で、次のように先頭に#を付けてコメントアウトしておけば Kurucz の方式が選択される。

```
#josh feautrier
#josh rybicki
```

(3600K, 0.2)での計算時間はそれぞれ、3.8分、2.9分、82分と、大きな違いがあった。図1のとおりで、最終的に得られた $\rho_{\text{ox}}-T$ 関係に大きな差異はなかった。

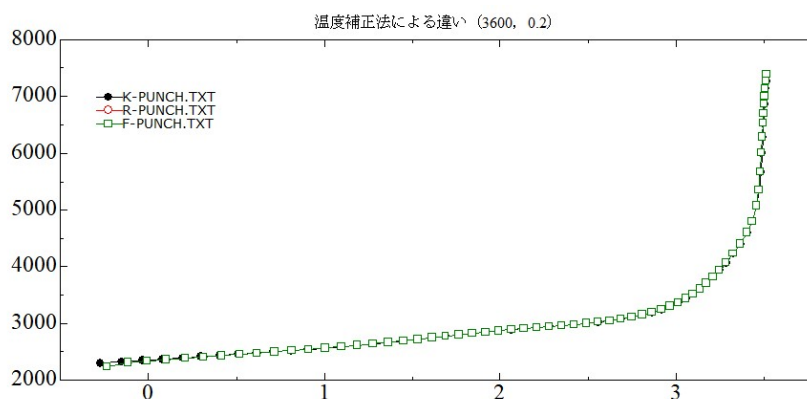


図5. 3種類の温度補正法による $\rho_{\text{ox}}-T$ 関係

■ 初期値入力ファイルの例 input.dat, model.dat

L-atlas の計算に際して、初期値ファイルとして2種類用意しなければならない。制御用の input.dat と大気モデル model.dat である。model.dat は計算したい新しい大気モデルのパラメータに近いモデルを持ってくる。

表5. 初期値 input.dat, model.dat の例

```
read punch
read lines
opacity on Xline
#ttaup ode
#josh feautrier
#josh rybicki
molecules on
#surface intensity 12 1.0 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.05 0.01
```



```

#surface intensity
iterations 15
print      1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 2 0 0 0 0 1 0 0 0 0 1
punch      0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
surface flux 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 フラックス表示
#resolution 10000
begin
TEFF 4260. GRAVITY 0.90000 LTE
TITLE SDSC GRID [+0.0] VTURB 2.0 KM/S L/H 1.25
OPACITY IFOP 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 0 0 0
CONVECTION ON 1.25 TURBULENCE OFF 0.00 0.00 0.00 0.00
ABUNDANCE SCALE 1.00000 ABUNDANCE CHANGE 1 0.91100 2 0.08900
ABUNDANCE CHANGE 3 -10.88 4 -10.89 5 -9.44 6 -3.48 7 -3.99 8 -3.11
. . .
READ DECK6 72 RHOX,T,P,XNE,ABROSS,ACCRAD,VTURB
1.40710741E-02 2350.3 1.407E-01 3.051E+06 9.477E-06 3.282E-03 2.000E+05
1.86715553E-02 2376.4 1.867E-01 4.076E+06 9.858E-06 2.980E-03 2.000E+05
2.45594180E-02 2402.8 2.455E-01 5.410E+06 1.029E-05 2.716E-03 2.000E+05
. . .

```

表6. 出力結果 `punch_file`

```

<Linux 版>
read deck6 72 rhoz,t,p_gas,xne,abross,accrad,vturb,flxcnv,vconv,velsnd
1.545E-02 2381.1 1.227E-01 2.862E+06 8.631E-06 1.138E-03 2.000E+05 0.000E+00 0.000E+00 1.219E+06
2.051E-02 2406.6 1.629E-01 3.835E+06 8.939E-06 1.064E-03 2.000E+05 0.000E+00 0.000E+00 1.090E+06
2.702E-02 2432.1 2.146E-01 5.105E+06 9.278E-06 9.901E-04 2.000E+05 0.000E+00 0.000E+00 9.845E+05
. . .
4.567E+02 8414.1 3.616E+03 1.831E+14 3.483E+00 1.770E-01 2.000E+05 9.195E-01 4.523E+05 8.158E+05
4.616E+02 8553.8 3.654E+03 2.129E+14 4.252E+00 1.628E-01 2.000E+05 9.394E-01 4.446E+05 8.250E+05
4.670E+02 8673.9 3.696E+03 2.412E+14 5.027E+00 1.552E-01 2.000E+05 9.498E-01 4.388E+05 8.336E+05
pradk 4.3226E-01

<Windows 版>
read deck6 72 rhoz,t,p_gas,xne,abross,accrad,vturb,flxcnv,vconv,velsnd
1.545E-02 2381.1 1.227E-01 2.862E+06 8.631E-06 1.138E-03 2.000E+05 0.000E+00 0.000E+00 1.219E+06
2.051E-02 2406.6 1.629E-01 3.835E+06 8.939E-06 1.064E-03 2.000E+05 0.000E+00 0.000E+00 1.090E+06
2.702E-02 2432.1 2.146E-01 5.105E+06 9.278E-06 9.901E-04 2.000E+05 0.000E+00 0.000E+00 9.845E+05
. . .
4.568E+02 8414.1 3.616E+03 1.831E+14 3.483E+00 1.770E-01 2.000E+05 9.195E-01 4.523E+05 8.158E+05
4.616E+02 8553.8 3.654E+03 2.129E+14 4.253E+00 1.628E-01 2.000E+05 9.394E-01 4.446E+05 8.250E+05
4.670E+02 8673.9 3.696E+03 2.412E+14 5.027E+00 1.552E-01 2.000E+05 9.498E-01 4.388E+05 8.336E+05
pradk 4.3226E-01

```

■ サンプル計算

次の3種類を添付した。(1)ではベテルギウスのフラックスの観測値 (Indo、Kitt Peak National Observatory

のホームページから) と計算値との比較も行っている。図6を参照のこと。

- (1) (3500K, 0.0)
- (2) (10000K, 4.0)
- (3) (20300K, 2.7)

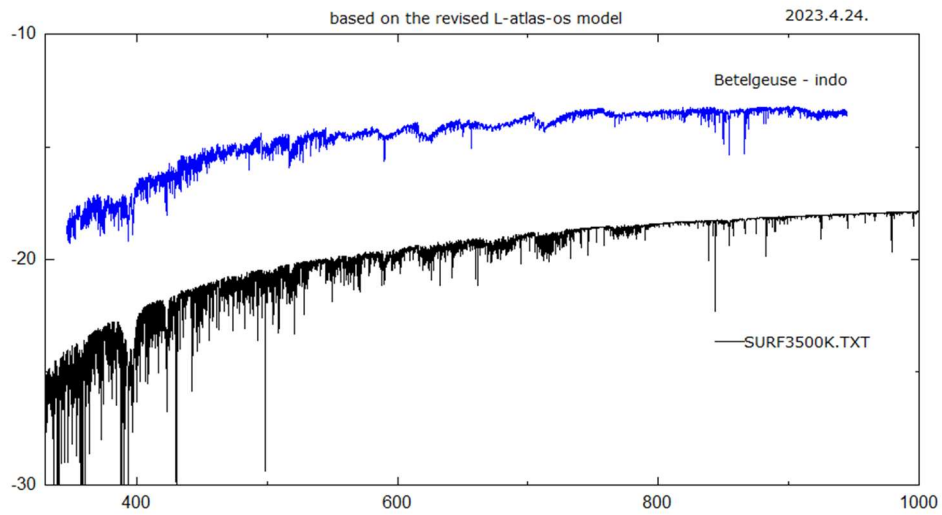


図6. サンプル計算の一例

references:

Lester, J. B. & Neilson, H. R. 2008 A&A 491, 633-641, 'Satlas: Spherical versions of the atlas stellar atmosphere program'

付 録

■ 球状大気用の s-atlas について

球状大気用の s-atlas では Linux 版でも 90 分ほどかかるので Windows 版は実用的ではない。

なお、超巨星に対応するパラメータでも計算してみたが、平行平板大気の L-atlas12 のフラックスと比較すると顕著な差はなかった。

■ コンパイル用シュルスクリプト - Ubuntu

Lester のファイルに付属している Makefile はそのままでは動かないので（従来のようにコンパイラ本体の場所を指定するだけではいけないらしいが、詳細不明）、次のようなシェルスクリプトに置き換えた。

表 4. コンパイル用シュルスクリプト

```
#!/bin/bash
command /opt/intel/oneapi/compiler/2023.0.0/linux/bin/intel64/ifort -c module_var_types.f
command /opt/intel/oneapi/compiler/2023.0.0/linux/bin/intel64/ifort -c module_code_dimensions.f
command /opt/intel/oneapi/compiler/2023.0.0/linux/bin/intel64/ifort -c module_physical_constants.f
command /opt/intel/oneapi/compiler/2023.0.0/linux/bin/intel64/ifort -o atlas_os.exe module_var_types.o
module_code_dimensions.o module_physical_constants.o module_abross_vars.o module_abtot_vars.o
module_abundances.o module_atmosphere_parameters.o module_continuum_tables.o module_conv_vars.o
module_depart_vars.o module_edensity_vars.o module_elements_vars.o module_flux_vars.o
module_freq_set.o module_freq_vars.o module_gaussian_parameters.o module_height_vars.o
module_if_vars.o module_intensity_vars.o module_isotope_vars.o module_iter_vars.o
module_junk_vars.o module_logtab.o module_molecular_vars.o module_odeint_vars.o module_opacity.o
module_opacity_switches.o module_potion_vars.o module_program_switches.o module_put_vars.o
module_pzero_vars.o module_rad_pressure.o module_rad_vars.o module_rhoz_vars.o
module_state_vars.o module_tabex.o module_tau_std.o module_temp_vars.o module_total_opacity.o
module_total_pressure.o module_tsmooth.o module_turbpr_vars.o module_txnzn_vars.o
module_xlines_vars.o module_xnf_dop.o module_xnf_vars.o module_xnmol.o atlas_os_main.o
atlas_os_conv.o atlas_os_josh.o atlas_os_kapp.o atlas_os_pops.o atlas_os_read.o
atlas_os_tcorr.o atlas_os_utils.o
```

以上