

Guide How to use:

**shellspec17gen1.f90**

---

© 2015 Klára Šejnová

The goal of this work is to find out the characteristics of a disk around the star, this is called inverse problem. An inverse problem is a general framework that is used to convert observed measurements into information about a physical object or system that we are interested in. For this problem we use Shellspec code. I present here description of the modification of the Shellspec code. I nicely ask user to read original manual of the Shellspec code at first. User can find it at Budaj & Richards (2004). To understand modification more it is also necessary to know something about genetic algorithms.

## 1 Current modification

Firstly we translated the code into programming language Fortran 90 (Metcalf et al. 2004) which improved the readability and usability of the code and made it more effective in combination with genetic algorithm subroutine PIKAIA (explained below). This translation also enable the code to be more efficient in the respect of the upgrading the compilers which still count with the programming language Fortran 90 unlike Fortran 77. Modified code follows simple computational scheme Fig. 1. Namely in the initial run we have population of  $n_i$  individuals with randomly generated set of parameters (genome). For every individual from population we calculate normalized synthetic spectrum  $F_{\text{synth}}$  using modified Shellspec code and compare it with normalized observed spectrum  $F_{\text{obs}}$ . As a measure of the quality of the fit we define fitness function

$$f_{\text{fit}} = \frac{1}{1 + \chi^2}, \quad (1)$$

based on the  $\chi^2$  merit function

$$\chi^2(A) = \sum_{i=1}^N \frac{[F_{\text{obs}}(x_i) - F_{\text{synth}}(x_i; A)]^2}{F_{\text{synth}}(x_i; A)}, \quad (2)$$

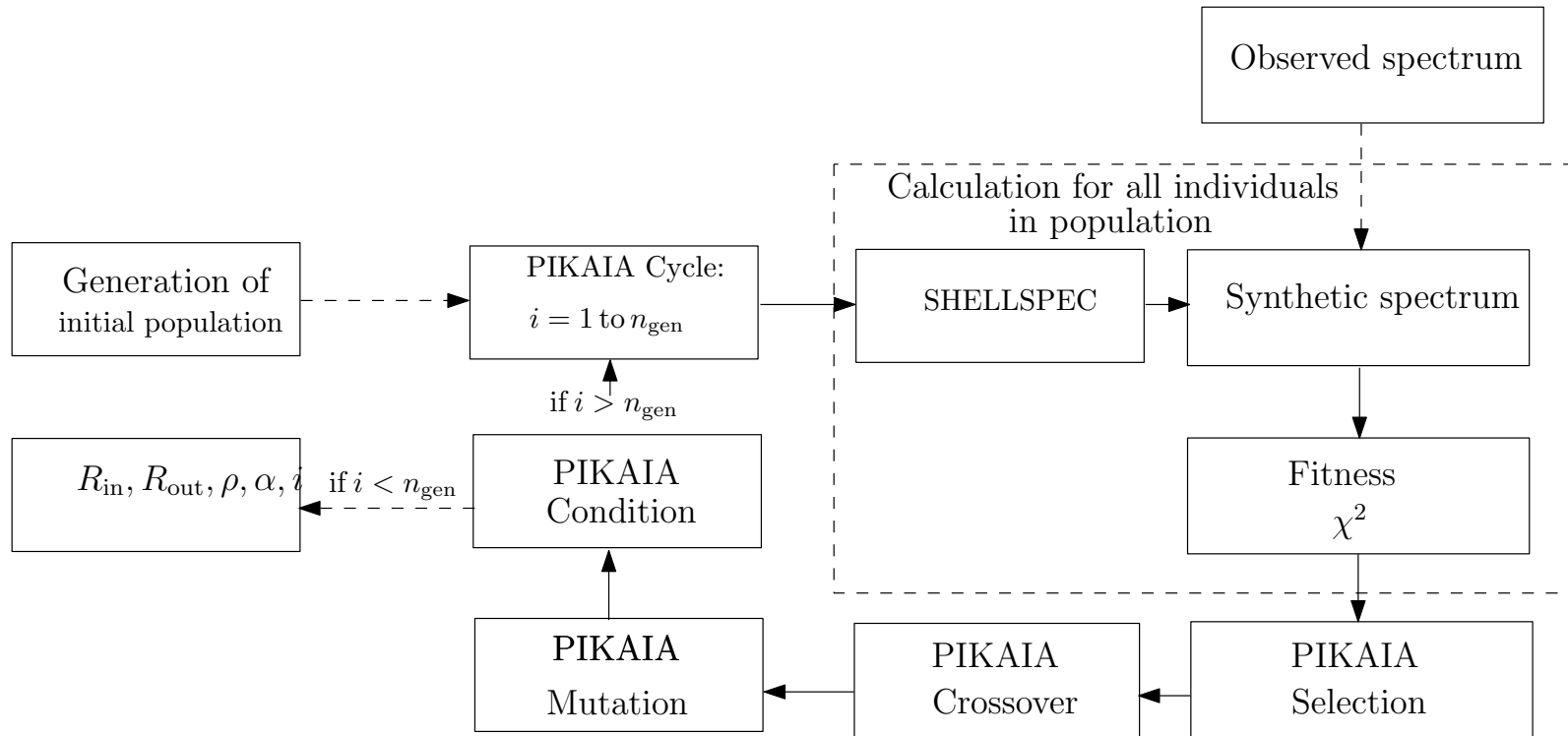
where  $F_{\text{obs}}(x_i)$  is the flux of the observed spectrum at the wavelength  $x_i$  and  $A(a_1, \dots, a_j)$  is set of free input parameters like inner(outer) disk radius, density of the disk, opening angle or inclination. For determination of merit function (2) it was necessary to interpolate calculated synthetic spectrum to the same wavelengths as the observed spectrum.

In the next step, input parameters describing the geometry and the physics of the problem are optimized via genetic algorithms PIKAIA. Namely, all individuals (every individual represents different combination of parameters) from the given population were sorted according to their value of fitness function (Eq. (1)) from the best to the worst individual. Only the best individuals survive and generate new population via mutation and crossover operations and thus generated the new sets of parameters. Loop continue until the best match is found. Values of the estimated free parameters and  $\chi^2$  are saved to the output file. Number of individuals in the population  $n_i$  and number of generation  $n_g$  which are driving behavior of PIKAIA genetic algorithm must be defined before calculation and are stored in the control vector `ctrl`. Users can also define boundaries for their own free parameters if it is necessary.

As it is clear from the above, to use our modified code user needs to have observed spectrum of the studied star. Shellspec code itself has not needed it before but the modified code uses the observed spectrum as an input.

### 1.1 More detailed description

The original program code was transformed into function which is contained in module called *shellgen*, this module is used by the program *shellspec17gen1* which is the main program



**Figure 1:** Layout chart of the code: 1. Creation of the initial population (generation zero) of  $n_i$  individuals with the different set of parameters. 2. Starting of the generation cycle 3. For every individuals we calculate synthetic spectrum using modified SHELLSPEC 4. This synthetic spectrum is compared with the observed spectrum using fitness function 5. Selection of the best individuals using fitness values. 6. Reproduction for the new generation using crossover applied on the best individuals. 7. Mutation applied on the individuals from the new generation. 8. If the number of the generation is greater than  $n_g$  then cycle will end else cycle will continue.

after modification. The module *shellgen* includes function *shell.calculation* which calculates normalized synthetic spectrum.

We also changed some of the subroutine (instead *shellspec.in* we defined module *initsh.f90* - the concept of the new file is the same as before, only the definition and values of parameters are in the same folder, so it is more clear), we also put some subroutine into the module (*raphson.f90*, *shellfunc.f90*, *statfun.f90*) and made it more systematic.

In *shellgen* module user can define free parameters and boundaries for parameters. PIKAIA subroutine then choose random number in interval of boundaries (for example `rinmax = 5.D0`, `rindc=rstar+xpar(1)*rinmax`) and run original shellspec code. When the run is over results are saved (for this moment we are interested in synthetic spectrum (flux, wavelength) but user can choose different output if necessary), observed spectrum is read and subroutine *interpolace(nfreq,b,matice,inter)* is done. To upload data for observed spectrum we use command: `open(unit=16, file='name of the file with data for observed spectrum', status='old')`. Subroutine *interpolace* interpolates values of the synthetic data so it could be possible to compare them with the observed data (we want to know synthetic flux at observed wavelength)

$$F_{\text{obs}}(x) = F_{\text{obs0}} + \frac{(F_{\text{obs1}} - F_{\text{obs0}})}{(x_1 - x_0)}(x - x_0) \quad (3)$$

where  $F_{\text{obs1}}$ ,  $F_{\text{obs0}}$  is synthetic flux,  $x_1$ ,  $x_0$  synthetic wavelength and  $F_{\text{obs}}(x)$  flux at the wavelength. The subroutine *chii(matice,inter,chi2)* determine value of the fitness for synthetic and observed spectrum. The result of user-defined fitness function which PIKAIA seeks is  $1/(1 + \chi^2)$ . This function is called *shell.calculation*. This procedure is done several times (it depends how big number of generations user defines) and the result is the best fit. All the preliminary results are saved into the file *vysledky.dat*. Again user can define what should be saved here.

## 2 How to use the modified code

- In *shellspec17gen1.f90* user can define number of the individuals in population `ctrl(1)` and number of the generation `ctrl(2)`. The meaning of the control vector can be found in the manual for PIKAIA. User can define other properties by this vector. The bigger the number of the generation the longer the computing time. With increasing number of the generation the fitness is getting better but it is not that much significant. In our experience it is not necessary to have big number of the generation. User can also define number of the free parameters by `npar`.
- In *shellgen.f90* user defines free parameters. For example to define opening angle of a disk we use

`adisc=1.D0+xpar(4)* adiscmax,`

where `xpar()` lies in the interval  $[0, 1]$ . This also means that lower boundary for opening angle is  $1^\circ$  and upper boundary is  $1^\circ + \text{adiscmax}$  and `adiscmax` is defined by user. Caution must be made when defining `adiscmax`. If the lower boundary is not zero then the value for `adiscmax` does not mean upper boundary, the upper boundary is lower boundary plus `adiscmax`. Number 4 in `xpar(4)` means that it is the fourth free parameter, this information is useful for reading the results.

- In file *initsh.f90* user defines values for the object to be study, usually parameters which are known, values for free parameters do not have to be filled up.

- **Example**

The example is a model presented by a star with a disk, the properties of the primary star are usually known unlike the properties of the disk. Thus the parameters of primary star are defined in file *initsh.f90*. And as the free parameters we choose inner, outer radius of the disk, density, opening angle, microturbulent velocity which are defined in module *shellgen*. The boundaries of some of these parameters were found in different articles. User should find its own boundaries for different type of the model. We have five free parameters in this example model  $\Rightarrow$  `npar= 5`. Number of the individuals is 100  $\Rightarrow$  `ctrl(1)= 100` and number of generations is 30  $\Rightarrow$  `ctrl(2)= 30`. As was mentioned before these free values can be defined in *shellspec17gen1.f90*. Definition of the free parameters is in the beginning of the module *shellgen.f90*:

*Normalization of free parameters for optimalization problem*

```
rinmax = 5.D0
drmax = 12.D0
densmax=1.D-10
adiscmax=45.D0
vtrbdcmax=150.D0
```

*Definition of free parameters*

```
rindc = rstar + xpar(1)*rinmax
dr = 1.D0 + xpar(2)*drmax
routdc = rindc+dr
densdc=1.D-13+xpar(3)*densmax
adisc=1.D0+xpar(4)*adiscmax
vtrbdc=10.D0+xpar(5)*vtrbdcmax
```

- To run the code we have prepared *Makefile* with further instruction:

```
make clean
make mod
make shellspecgen
./shellspecgen
```

- If user would like to save all the output from PIKAIA then do command `./shellspecgen > results.txt` and the output will be saved to the file *results.txt*. To understand all of the output from PIKAIA please see Charbonneau & Knapp (1995) here we present only the most important output. Example of the *results.txt* file follows:

```
*****
*                PIKAIA Genetic Algorithm Report                *
*****

Number of Generations evolving: 30
  Individuals per generation: 100
Number of Chromosome segments: 5
  Length of Chromosome segments: 6
    Crossover probability: 0.8500
    Initial mutation rate: 0.0500
    Minimum mutation rate: 0.0005
    Maximum mutation rate: 0.2500
  Relative fitness differential: 1.0000
    Mutation Mode:

Variable                input and values of free parameters +
Reproduction Plan:      |
Full generational replacement |
Parameters :           5 / 0.775036097  0.766041458  0.58724206-----|
0.761822581  0.939142346  0.00000000  0.00000000  0.00000000 |
```

```

0.00000000    0.00000000    0.00000000    0.00000000    0.00000000    |
0.00000000    0.00000000    0.00000000    0.00000000    0.00000000    |
0.00000000    0.00000000    0.00000000    0.00000000    0.00000000    |
0.00000000    0.00000000    0.00000000    0.00000000    0.00000000    |
0.00000000    0.00000000    0.00000000    0.00000000    0.00000000    |
Test :   19.167677736282350   8.9751804828643795   5.8824206053462074E-011-----|
35.282016754150391         150.87135314941406

```

Shellspec

```

INPUT:
shellspec.in - main input (geometry,objects...)
line.dat - atomic data for the lines
shellspec.mod - 3D model of the shell
                (optional IF imodel=2)
abundances - abundances (optional IF ichemc=1)
phases - orbital phases (optional IF nphase=0)
starspec1 - star spectrum (optional IF lunt1>0)
starspec2 - star spectrum (optional IF lunt2>0)      → start of Shellspec
starspec3 - star spectrum (optional IF lunt3>0)
albedo1 - albedo (optional IF albst<0)
albedo2 - albedo (optional IF albcs<0)
OUTPUT:
shellspec.out - more detail output
fort.xx - 2D images at some frequency
shellspectrum - spectrum of the shell
lightcurve - LC or trailed spectrogram

nphase=          1
No. of sp. lines READ from line.dat=          1
nstar1=          1          31006
i= 40.50 alpha=  0.00
chi2 38.263639636012677          0.960224926 → fitness of the syntetic
                                                and observed spectrum
                                                with input parameters
.
. → shellspec runs several times
.

          656609   657335   636841
          358352   354352   423062
           12089    12089   110006
          603457   808037   367174
          403011   346258   408248

status:          0
x:  0.656608999 0.358352005 1.20890001E-02 0.603456974 0.403010994
f:  0.998377800 → this is the result with the value of the best
                  individual (x:) and its fitness function (f:)

ctrl: 100.000000 30.000000 6.000000 0.850000 2.000000 0.050000
       0.000500 0.250000 1.000000 1.000000 1.000000 3.000000
       →these are just values for ctrl vector

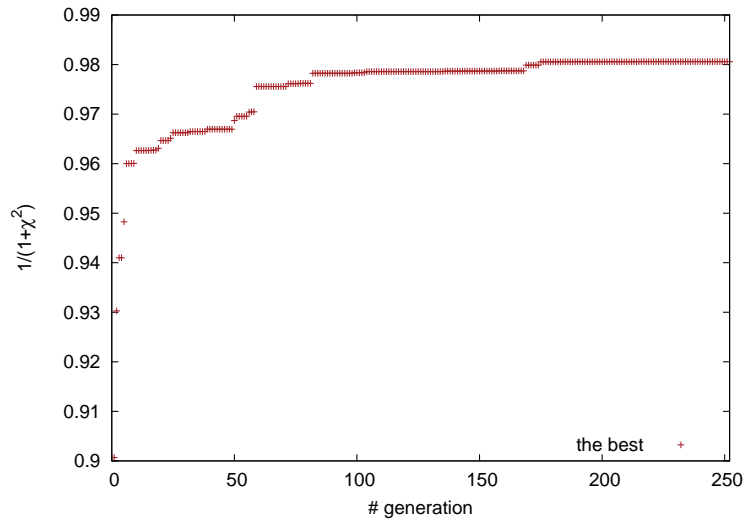
```

- Values for the best individuals are saved into the file *generace.dat*. By plotting the first and the fourth column of the file user will be able to see how the best individual for each generation varies and getting better by each generation. This file is generated by the `write` command in module *pikaia.f90* (`/home/shellspecf90/source`). Example of the *generace.dat* is below (in columns from the first: generation, individuals, initial mutation rate, fitness of the first best, second best, and the average) and data of the first and the fourth column (number of generation and fitness of the first best individual for the generation) are plotted in the Fig. 1.

```

#generace.dat
1  99 0.0033333 0.3286170 0.3286168 0.0252947
2  99 0.0022222 0.3286170 0.1962448 0.0554000
3  99 0.0014815 0.3286170 0.3286168 0.0736583
4  99 0.0009877 0.4598798 0.3485152 0.0920731

```



**Table 1:** Evolution of the value for  $1/(1 + \chi^2)$  for the best individuals for each generation

5	99	0.0006584	0.5018177	0.4845585	0.1004014
6	99	0.0005000	0.5018177	0.4845585	0.1254106
. . .					
495	99	0.0432488	0.8062923	0.8056455	0.5158235
496	99	0.0432488	0.8062923	0.8062890	0.5158592
497	99	0.0432488	0.8062923	0.8062800	0.6942428
498	99	0.0432488	0.8063483	0.8063009	0.7174796
499	99	0.0648732	0.8063483	0.8063152	0.7344757
500	99	0.0648732	0.8063483	0.8062189	0.6891267

## References

- Budaj, J. & Richards, M. T. 2004, Contributions of the Astronomical Observatory Skalnaté Pleso, 34, 167
- Charbonneau, P. & Knapp, B. 1995, A User's Guide to PIKAIA 1.0, Tech. rep.
- Metcalf, M., Reid, J., & Cohen, M. 2004, Fortran 95/2003 Explained
- Šejnová, K., Votruba, V., & Koubský, P. 2012a, in Astronomical Society of the Pacific Conference Series, Vol. 464, Astronomical Society of the Pacific Conference Series, ed. A. C. Carciofi & T. Rivinius, 219
- Šejnová, K., Votruba, V., & Koubský, P. 2012b, in IAU Symposium, Vol. 282, IAU Symposium, ed. M. T. Richards & I. Hubeny, 261–262
- Šejnová, K. 2010, Master's thesis, Masarykova Univerzita v Brně