AsfGrid

ACKNOWLEDGEMENT:

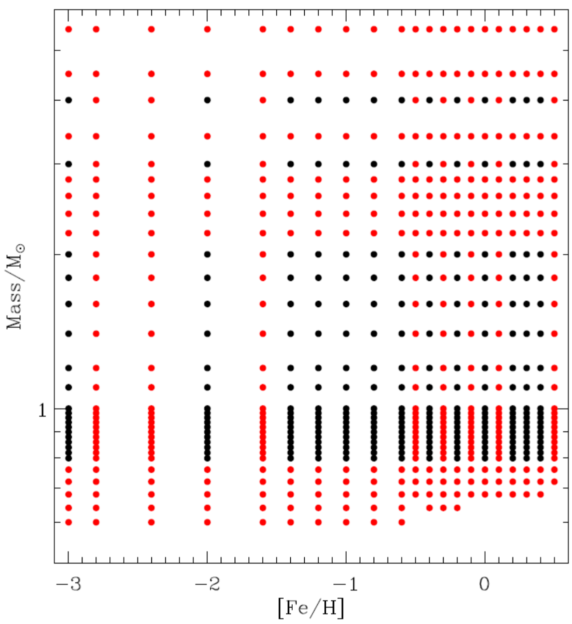
If you use the current version of AsfGrid (v0.0.6) for your research we ask you to cite both Sharma et al., 2016, *ApJ*, 822, 15 and Stello & Sharma, 2022, *Res. Notes AAS*, 6, 168.

DESCRIPTION:

asfgrid 0.0.6: computes

1. Corrected Δv to be used in the Δv scaling relation,
2. Correction factor $f_{\Delta v}$ for Δv scaling relation (see Eq 5 in Sharma et al. 2016, ApJ, 822,15),
3. Mass and radius using Δv correction, by interpolation over a grid.

The interpolation grid covers all combinations of the following values of metallicity and mass:
log(Z/0.019) = [-3.00, -2.80, -2.40, -2.00, -1.60, -1.40, -1.40, -1.20, -1.00, -0.80, -0.60, -0.50, -0.40, -0.30, -0.20, -0.10, 0.00, 0.10, 0.20, 0.30, 0.40]
mass= [0.60, 0.64, 0.68, 0.72, 0.76, 0.80, 0.82, 0.84, 0.86, 0.88, 0.90, 0.92, 0.94, 0.96, 0.98, 1.00, 1.10, 1.20, 1.40, 1.60, 1.80, 2.00, 2.20, 2.40, 2.60, 2.80, 3.00, 3.40, 4.00, 4.50, 5.50]
except for a few of the highest-metallicity lowest-mass combinations as shown in the following figure from Stello & Sharma, 2022, *Res. Notes AAS*, 6, 168:



WHATS NEW IN VERSION:

0.06	• The grid has been substantially upgraded with an extended range and finer spacing in mass and metallicity. To upgrade to the new version, simply download the directory with the data files and change the default data directory in line 150 of asfgrid.py to this directory. • However, the code asfgrid.py is the same as before.
0.05	• Minor bug fixes. • Changes are related to the use of the code as a python module. • Now also handles scalar input that is outside the grid range. • For scalar input, output is also scalar.
0.04	• To be able to handle `out of grid range' input values; the default here is to use nearest grid point. • Now the program works for input files with only one line (star). • Added function get_mass_radius() . • When run in script mode, the text output files now include radius along with mass. • Now [Fe/H] is allowed as input instead of Log(Z) in the text input file. • A flag 'isfeh', which can be True or False, is introduced in the functions get_mass_radius() and get_dnu_numax(). • Finer grid spacing in age during the He core burning phase.
0.03	• Changes in the following files: asfgid.py and interp_grid2.ebf. • Version 0.0.1 had a bug in line 165 of asfgrid.py, which could result in wrong answers. • Also, the file interp_grid2.ebf, was of lower resolution than required to get reliable answers.

INSTALLATION:

1) Download the latest zip file, e.g. , `asfgrid_v0.0.6.zip`

```
unzip asfgrid_v0.0.6.zip
cd asfgrid_v0.0.6
./asfgrid.py -help
OR
>>>import asfgrid
```

Ideally the data files directory should be in the current directory of the installed code. If you want to put the data files directory that come with the program in a different directory than where the code is installed, you need to set the appropriate datadir path by editing line 150 to
`self.datadir='asfgrid_data_aug22/'` (if in current directory)
or
`self.datadir='/your_path/asfgrid_data_aug22/'    (if custom dir)`

2) requirements: scipy__version__ >= 0.14.0

3) You will need the python ebf module to read the grid interpolation files. These can be installed by

For non root do:

```
pip install ebfpy --user
```

For root do:

```
sudo pip install ebfpy
```

If required see http://ebfformat.sourceforge.net/ for further details on installation of ebf.

RUNNING THE CODE:

asfgrid.py is the main code. It can be run as

- a script or as
- a python module.

1. For script mode

To run make sure it is executable by e.g.

```
chmod 755 asfgrid.py
```

The input and output is via ascii files. Check usage with

```
./asfgrid.py -help
```

USAGE:

```
asfgrid.py inputfile
```

DESCRIPTION:

Output file name is constructed from filename with suffix .out
Input file should be ascii as follows

```
evstate logz   teff     dnu    numax
1       -1.97  4659.8   8.81   92.36
1       -1.98  4903.2   13.1   157.3
```

First line must contain column names
Column names can be in any order but need to follow names given below

OPTIONS:

Possible input/outputs are
1) (evstate, logz, teff, dnu, numax) ->(mass, radius)
2) (evstate, logz, teff, mass, logg) ->(dnu, numax, fdnu)
3) (evstate, logz, teff, mass, logg, mini) ->(dnu, numax, fdnu)

If using feh instead of logz
4) (evstate, feh, teff, dnu, numax) ->(mass, radius)
5) (evstate, feh, teff, mass, logg) ->(dnu, numax, fdnu)
6) (evstate, feh, teff, mass, logg, mini) ->(dnu, numax, fdnu)

(1) and (4) are typically used to go from observations to resulting mass and radius. The rest are typically used to go from any stellar model to predicted dnu (corrected), numax and correction factor. If your input is from a model that includes mass loss, where mass<mini, the correct interpolation is assured by using option (3) or (6).

VARIABLES:

Name	Description	Unit
evstate	1=Pre RGB tip 2=Post RGB tip	
logz	log metallcity Log(Z)	
feh	If input is feh, program use [Fe/H]=log(Z/$Z_{\odot}$) with $Z_{\odot}$=0.019 to convert to logz.	
teff	Effective temperature T_{eff}.	K
mass	Actual mass; when written as output, it is the mass obtained using the dnu-scaling relation corrected with $f_{\Delta\nu}$ (fdnu).	$M_{\odot}$
radius	Radius; corresponds to the radius obtained using the dnu-scaling relation corrected with $f_{\Delta\nu}$ (fdnu).	$R_{\odot}$
mini	Initial mass. Useful for cases with mass loss, where actual mass is <= mini.	
logg	Log(surface gravity / [cm s^{-2}])	
dnu	$\Delta\nu$, observed large frequency separation (μHz). When written as output, it corresponds to the radial mode frequency-based $\Delta\nu$ from the grid.	μHz
numax	ν_{max} , observed frequency of max power (μHz). When written as output, it corresponds to the scaling-based ν_{max} from the grid.	μHz
fdnu	$f_{\Delta\nu}$, correction factor for $\Delta\nu$ scaling relation.	

RUN EXAMPLES:

Using the three test input files provided

```
./asfgrid.py test1.txt
./asfgrid.py test2.txt
./asfgrid.py test3.txt
```

2. For module mode:

Start python or ipython. Then

```
>>> import asfgrid
>>> help(asfgrid)
```

Run example:

```
>>> import asfgrid
>>> evstate=[1,1]
>>> logz=[-1.97,-1.98]
>>> teff=[4659.8,4903.2]
>>> dnu=[8.81,13.1]
```

```
>>> numax=[92.36,157.3]
>>> s=asfgrid.Seism()
>>> mass,radius=s.get_mass_radius(evstate,logz,teff,dnu,numax)
>>> print(mass,radius)
>>> logg=s.mr2logg(mass,radius)
>>> dnu,numax,fdnu=s.get_dnu_numax(evstate,logz,teff,mass,mass,logg)
>>> print(dnu,numax)
```

DATA TABLES AND FILES:

The data tables and files are packed in a directory as follows

- `asfgrid_data_aug2022/`
  - `dnu_grid_evstate.txt`: Raw stellar grid (MESA) in ascii format (not used by asfgrid)
  - `dnu_grid_evstate.ebf`: Same as above but in ebf format (not used by asfgrid)
  - `grid_interp1.ebf` : Stellar grid in format for asfgrid to interpolate in (evstate, logz, teff, mass, logg_teff)
  - `grid_interp2.ebf` : Stellar grid in format for asfgrid to interpolate in (evstate, logz, teff, mass_nu, logg_teff)

Table: dnu_grid_evstate

The main grid table containing all models sorted by (z, mass, model). Model increases monotonically with age.

Column name	Description	Units
z	Metallicity	
mass	Stellar mass M. Since the grid has no mass loss this is the same as mini	$M_\odot$
model	Consecutive model number increases with age	
evstate	Evolutionary state -2: Post RGB wiggles between 1 and 2 -1: PreZAMS 0: MS [central_he4<0.945] 1: Subgiant+RGB [central_he4>0.945] 2: Stable He core burning (RC) [0.005 ≤ central_he4 ≤ 0.945] 3: AGB [central_he4<0.005] In this file, evstate has more states than used in the code	
age	Stellar age	Gyr
teff	Effective Temperature	K
numax	ν_{max} , from the scaling relation using mass, radius, and T_{eff} and assuming $\nu_{max,\odot}$=3090 μHz.	μHz
dnu_frq	Δν . From fit to radial mode frequencies.	μHz
dnu_int	$\Delta\nu_{int}$. From sound-speed integral.	μHz

dnu_sc	$\Delta\nu_{scaling}$ = 135.1 $\sqrt{(\rho/\rho_\odot)}$obtained from density scaling relation.	μHz
dp	Period spacing from buoyancy integral.	s
central_he4	Central He^4 fraction	
lum	Luminosity L	$L_\odot$
sradius	Stellar radius R	$R_\odot$

Table: grid_interp1

Derived table for estimating correction factor $f_{\Delta\nu}$ by interpolating over (z, mass, evstate, logg_teff).

Column name	Description	Units
logz	Log Metallicity, log(z)	
mass	Stellar mass. Since the grid has no mass loss this is the same as mini	$M_\odot$
evstate	Evolutionary state (see dnu_grid_evstate table)	
logg_teff	A function of logg and log(teff), which is almost monotonic with age. log(teff)+0.05* [tanh((logg-4.5)/0.25)+1]*logg For logg<<4.2, logg_teff ~ log(teff).	
model	Consecutive model number increases with age	
age	Stellar age	Gyr
teff	Effective Temperature T_{eff}	K
logg	log (Surface gravity / cm s^{-2})	
fdnu	Correction factor, $f_{\Delta\nu} = \Delta\nu/\Delta\nu_{scaling}$, ($\Delta\nu$ from fit to radial mode frequencies)	
fdnu_int	Correction factor, $f_{\Delta\nu, int} = \Delta\nu_{int}/\Delta\nu_{scaling}$, ($\Delta\nu_{int}$ from sound speed integral)	
mass_nu	Mass from $\Delta\nu$ and log(g) using scaling relation $M_{\Delta\nu}=(g/g_\odot)^3 / (\Delta\nu/135.1)^4$	
dp	Period spacing from buoyancy integral.	s
central_he4	Central He^4 fraction	

Table: grid_interp2

Derived table for estimating mass by interpolating over (z, mass_nu, evstate, logg_teff).

Column name	Description	Units

logz	Log Metallicity, log(z)	
mass_nu	Mass from Δv and log(g) using scaling relation $M_{\Delta v}=(g/g_{\odot})^3$ / $(\Delta v/135.1)^4$	
evstate	Evolutionary state (see dnu_grid_evstate table)	
logg_teff	A function of logg and log(teff), which is almost monotonic with age. log(teff)+0.05* [tanh((logg-4.5)/0.25)+1]*logg For logg<<4.2, logg_teff ~ log(teff).	
model	Model number increases with age	
age	Stellar age	Gyr
teff	Effective Temperature	K
logg	log (Surface gravity / cm s^{-2})	
fdnu	Correction factor, $f_{\Delta v} = \Delta v/\Delta v_{scaling}$, (Δv from fit to radial mode frequencies)	
fdnu_int	Correction factor, $f_{\Delta v,\, int} = \Delta v_{int}/\Delta v_{scaling}$, ($\Delta v_{int}$ from sound speed integral)	
mass	Stellar mass. Since the grid has no mass loss this is the same as mini.	$M_{\odot}$
dp	Period spacing from buoyancy integral.	s
central_he4	Central He^4 fraction	