

# Summary of Database

Yutong Ji

August 2019

## 1 Work has been done:

I wrote a ASCII file(model.txt, see attachment) which contains basic model information and a procedure (model.pro, see attachment) which can read model information into structures as what rhdmodeldata.pro did of Bernd's code.

With this procedure, one parameter can have more than one values (e.g. Quality: 1.0,2.0,3.0) which will be stored in an array; Different variable types can be distinguished(string, integer or float); And input can be more flexible: for example if 'Organization' is not specified in one entry then this whole line could be left out. It's also okay if a new parameter need to be added.

The ASCII file roughly looks like this(figure. 1):

```
=== ModelTable_CIFIST ===

==entry==
Model:      'd3t28g50mm00du4';
Quality:    1.0,2.0,3.0;
Author:     'Hans-Guenter Ludwig';
ParModelIdent: '.00';
nFile:      41;
mMeanAvg:   0;
nMeanAvg:   99999;
mTimeAvg:   0.0;
nTimeAvg:   0.0;
Opta_Table: 'g2v_marcs_idmean3x_12.opta';
Opacity_bins: 12;
EOS_file:   'eos_cifist2006_m00_a00_15.eos';
GAS_file:   'gas_cifist2006_m00_a00_15.eos';
LHD_file:   'yes';
LHD_directory: 'lhdmodels';
LHD_filename: 't5780g44mm00m13a15ob12_marcs.150','t5780g44mm00m13a20ob12_marcs.150';
readme_file: 'yes';
nx:         140;
ny:         140;
nz:         150;
Teff:       2800;
logg:       5.0;
M_H:        0.0;
parameter_filename: 'parameter.in';
```

Figure 1: one entry from model.txt

After discussion with Andy, he suggested some parameters like Opta\_Table, EOS\_file, to be added.

Once reading data is done, the model structure (stored as point array) we get will look like this(figure. 2):

```
[IDL> help,/str,*modDB[0]
** Structure <1354d68>, 24 tags, length=240, data length=238, refs=1:
MODEL          STRING      'd3t28g50mm00du4'
QUALITY        FLOAT       Array[3]
AUTHOR         STRING      'Hans-Guenter Ludwig'
PARMODELIDENT  STRING      '.00'
NFILE          INT         41
MMEANAVG       INT         0
NMEANAVG       LONG        99999
MTIMEAVG       FLOAT       0.00000
NTIMEAVG       FLOAT       0.00000
OPTA_TABLE     STRING      'g2v_marcs_idmean3x_12.opta'
OPACITY_BINS   INT         12
EOS_FILE       STRING      'eos_cifist2006_m00_a00_15.eos'
GAS_FILE       STRING      'gas_cifist2006_m00_a00_15.eos'
LHD_FILE       STRING      'yes'
LHD_DIRECTORY  STRING      'lhdmodels'
LHD_FILENAME   STRING      Array[2]
README_FILE    STRING      'yes'
NX             INT         140
NY             INT         140
NZ             INT         150
TEFF           INT         2800
LOGG           FLOAT       5.00000
M_H            FLOAT       0.00000
PARAMETER_FILENAME STRING  'parameter.in'
```

Figure 2: the resulting data structure

```

[IDL> restore, 'allbasdb.sav'
[IDL> help, /str, basDB
** Structure <319d8f8>, 34 tags, length=192, data length=192, refs=1:
MODEL          STRING      'at70g43n01'
DATEADDED      DOUBLE      2458179.2
GRAV_MODE      STRING      'constant'
EOSFILE        STRING      'eos_mm00_15.eos'
OPAFILE        STRING      'phoenix_opal_grey.opta'
MTIME          LONG        4000
NTIME          LONG        99999
ITIME_FIRST    LONG        283146
ITIME_LAST     LONG        0
TIME_FIRST     FLOAT       100000.
TIME_LAST      FLOAT       0.00000
RADIUS         FLOAT       0.00000
RADIUST8K      FLOAT       0.00000
MASS_STAR      FLOAT       0.00000
MASS_ENVELOPE  FLOAT       0.00000
TEFF           FLOAT       7000.00
TEFFOUT        FLOAT       7098.32
LUMINOSITYOUT  DOUBLE      0.0000000
S_INFLOW       FLOAT       2.85000e+09
S_CORE         FLOAT       2.62071e+09
GRAV           FLOAT       19952.6
ABUX           FLOAT       0.908510
ABUY           FLOAT       0.0908500
ABUZ           FLOAT       0.000640000
QMOL           FLOAT       1.30180
XB1TOT         FLOAT       7.40813e+09
XB2TOT         FLOAT       1.23469e+07
XB3TOT         FLOAT       3.08678e+09
NQUC           LONG        0
PRECISION      LONG        4
DIMENSION      LONG        2
NX1            LONG        600
NX2            LONG        1
NX3            LONG        500

```

Figure 3: structure of basDB (this is derived directly from 'allbasdb.sav' which you sent to me in email before)

```

modellist = udb_ListAll(modDB)
;modellist contains all the model names
modellist = modellist[where(udb_extract('Quality', modellist, modDB) ge 0.5)]
;select certain models
logg =alog10(udb_extract('grav', modellist, basDB))

```

Figure 4: an example of usage

## 2 Further idea:

### 2.1 How to fit fill the data base with actual data from lx40:/d/hgl/snaps?

Since the data structure we get is the same as what Bernd's code get from rhdmodeldata.pro, so theoretically there shouldn't be any problem to fill data from lx40:/d/hgl/snaps. To do this, I think we need rhd\_seq\_extractpar.pro which is to read the correspondent full files and par files into basDB and parDB. basDB is an anonymous structure containing basic data such as model name, Teff, radius. It looks like this (figure. 3):

Apparently this basDB contains much more information than modDB. parDB is similar but contains more information.

Once we have modDB, parDB, basDB, then we can calculate variables as we want, for example (figure. 4):

But now key problem is how do we get basDB and parDB from full files? I tried to run, but it shows to me there are some bugs (figure. 5) in the code. So I still haven't figured it out how to get basDB from full files.

```

[IDL> .com rhd_seq_extractpar.pro

grav=abs(spheregrav(Radius, r0=(*parDB[iFile]).r0_grav, r1=(*parDB[iFile]).r1_grav, r2=r2_grav, C=!con.G*(parDB[iFile]).mass_star))
^
% Syntax error.
At: /home/yutong/rhd_seq_extractpar.pro, Line 198

basDB[iFile].mass_envelope=mean(aarsg_seqavg_data('mass1', (*seqDB[iFile]), qavg=-1))*!con.MSun
^
% Syntax error.
At: /home/yutong/rhd_seq_extractpar.pro, Line 217

q=seqavg('s', (*seqDB[iFile]), units='cgs')
^
% Syntax error.
At: /home/yutong/rhd_seq_extractpar.pro, Line 231

```

Figure 5: Syntax error in rhd\_seq\_extractpar.pro