AUG python tools - an overview

G. Tardini, A. Bock, T. Lunt, M. Reich

aug_sfutils overview, October 12th 2021

Goal

Overview of the **aug_sfutils** package, intended for working with AUG shotfiles (r/w) and equilibria

- **Read AUG shotfiles ("DD")**
- **Read/manipulate equilibrium shotfiles**: mapping, contours, surface crossing points... ("KK")
- Write shotfiles: diagnostics, codes (WW)
- Edit Shotfile-headers (SFH)

Not presented: python-based "end products" such as `trview`, `pyspecview`, `idaview`.

If you have valuable AUG-related **modules to share**, email us or add to `www.aug.ipp.mpg.de/foswiki/bin/view/PythonAtAUG`

Why python

(On top of the general arguments pro python)
Shotfile r/w is possible in Fortran, c, IDL, matlab... but py-reading is rapidly increasing, especially for AUG-newbies:

- **High-level** (cumbersome stuff far from user, compact coding for users)
- Easier to code: often 1-line, array with correct shape + metadata (type, unit...)
- Easier to maintain (source is readable, very compact) $\Longrightarrow$ easier to enhance

Several “dd” and “kk”-like packages written by several people so far.
Today I present **new native classes**, which I maintain and “guarantee”.
Attempt: to merge all good features into one “optimum” package.

My fazit: if you start a new project/code with shotfiles r/w, **use py**.

Reading shotfiles with SFREAD

The class **aug_sfutils.SFREAD** (current version: 0.2.8) reads shotfiles:

- Works on: **lxts, tok(i), PCs** (tested on Linux, Mac, Windows)
- **py2/3 compatible**
- **no wrapper** around libddww.so, digesting shotfiles directly
- no fancy py-libs needed

Requirements:

- python with numpy, scipy, matplotlib (e.g. **anaconda**)
- pip if you wish local installation
- on PCs: **afsopen with kerberos authentication** to access shotfiles

Installing/loading aug_sfutils

On IPP's tok(i) and lxts:

module load aug_sfutils

Or easy local install (only option for PCs):

pip install aug_sfutils

Advantages of `pip install`:

- Smooth installation into a py-ready path
- Easy access to the source files, stored in
  `~/.local/lib/<current_py_vers>/site-packages/<package>`

Beware:

- if you use another py (even 3.8 instead of 3.7) you need another `pip install` in that environment (anaconda/xx)

aug_sfutils first test

Quick SFREAD test:

```
module load aug_sfutils
python
>>>import aug_sfutils as sf
>>>cez = sf.SFREAD(39649, ’cEZ’, ed=1)
>>># case+order insensitive, ’ed’ or ’edition’ keyword
>>># No data read yet at this stage, only SFH
>>>ticez = cez(’Ti’)
>>>ticez # python assumes print-statement
>>>dir(ticez) # list attributes and methods of Ti-object
```

SFREAD features

- SFREAD can read **any object** (Sig, SGR, ParSet, AB, TB...), **any data types** (int, float, char1-72...), **raw or calibrated** Sig/SGR, “difficult” TB, **private or ’AUGD’** shotfiles

- Some methods: `getlastshot`, `getgc`, `getpreviousshot`, `getcti_ts06`, `getlist`, `getlist_by_type`, `time_first`

- ParSets are read as whole, Parameters are dict entries

- Arrays retain **metadata: phys_units, obj_type, relations, indices, level, status, data_format, descr**

- **3-5 times faster** than previous dd_20200525.py, even more compared to dd.py (and **no segmentation** for large SGR!)

- Examples of typical user’s applications:

  www.aug.ipp.mpg.de/∼git/aug_sfutils

  Please email me in case of bugs or wishes

SFREAD practical info

```
>>>a=sf.SFREAD(39649, ’cEZ’)
>>>b1 = a.getobject(’Ti’)   # Array to SGR ’Ti’
>>>b2 = a(’Ti’)             # Array to SGR ’Ti’, caching
>>>t1 = a.gettimebase(’Ti’) # TB related to ’Ti’
>>>t2 = a(’time’)           # TB by name, caching
>>>a.time_first(’Ti’)       # True if t is 1st dim of ’Ti’
>>>a.getlist_by_type(obj_type=7) # Lists CEZ Sig names
```

- **Cached** arrayd so won’t read twice! **a() reads also ParSets+Dev**; `getobject` reads only Sig, SGR, TB, AB
- But: **getobject** takes optional **arguments nbeg, nend** for limited time range, for Sig + SGR(..., t)
- `gettimebase (getareabase)` works both with the related Sig or SGR name, or with the actual TB (AB) name
- t2 works only with the proper TB (AB) name

SFREAD by full-path

SFREAD has a special functionality: you can access a shotfile or a shotfile-header by giving the full afs-path as argument of **keyword sf or sfh**. E.g.:

```
>>>import aug_sfutils as sf
>>>a=sf.SFREAD(sf=\
...'/afs/ipp/home/a/augd/shots/3978/L1/TOT/39789.1')
# For Windows '\/afs/ipp/home...
>>>a('Wfi')
>>>b=sf.SFREAD(sfh=\
...'/afs/ipp/home/g/git/tr_client/AUGD/TRA00000.sfh')
>>>b('Setup').keys()
```

Good for parsing a shotfile header! Also as plan-B for shotfiles if the path-search fails (say different path-string syntax in a different OS).

“kk” functionality

“kk” is split into:

- The EQU **class** reads specifically “from AUG shotfile” (EQI, EQH, IDE) into an equilibrium object, suited for the mapeq toolbox

- mapeq is an abstract collection of methods: it requires an input equilibrium object, be it from AUG shotfile, or IMAS, eqdsk...
  It maps between spatial coordinates, evaluates magnetic surface contours, calculates B-field components...

EQU class

Quick test/syntax:

```
>>>import aug_sfutils as sf
>>>eq = sf.EQU(28053) # Reads almost whole EQ* shotfile
>>>eq.pfm.shape       # Displays dims of PFM SGR
```

The EQU class is based on SFREAD, hence fast and stable. It contains also the shotfile class (attributes and methods) in EQU.sf, so one can always add attributes a posteriori, e.g. CLE:

```
>>>nt = len(eq.time)
>>>eq.cle = eq.sf.getobject(’CLE’)[:, :nt]
```

On top of the EQ*-shotfile quantities (pfl, tfl, pfm, q, jpol, djpol, pres, dpres...) EQU objects have attributes for **derived quantities, e.g.** ρ_{tor} **and the COCO number**. The **q-profile value at the separatrix** is replaced by the **extrapolation** of the closest 2 inside.

mapeq methods

Example: fetching separatrix contour at t=3 s:

```
>>>import aug_sfutils as sf
>>>eq = sf.EQU(28053) # Reads almost whole EQ* shotfile
>>>r, z = sf.rho2rz(eq, 1., t_in=3, coord_in='rho_pol')
>>>r_sep = r[0]
>>>z_sep = z[0]
```

Typical AUG applications again at
`www.aug.ipp.mpg.de/~git/aug_sfutils`:

- Intersections line / magnetic surface
- Mapping R, z to ρ_{tor} for a given diagnostic (e.g. CEZ)
- Calculating magnetic fields at any R, z
- Getting ρ of a magnetic surface at given q

COCO class

The class

- Identifies the current COCO
- Converts an equilibrium object from a COCO to any other

Example:

```
>>>import aug_sfutils as sf
eq_in = sf.EQU(28053, diag='EQH')
# cocos_in is recognised automatically, CLISTE has 17 btw
eq_out = sf.cocos.coco2coco(eq_in, cocos_out=1)
print(np.average(eq_in.pfl), np.average(eq_out.pfl))
```

ww.py for shotfiles writing

More and more diagnosticians prefer to write their shotfiles with python. How to:

```
>>>import aug_sfutils as sf
>>>nshot = 38783
>>>data_d = {...}
>>>sf.write_sf(nshot, data_d, sfhdir, ’DGN’, exp=’AUGD’)
```

The ww.py class is ctypes-wrapped around the official lib
/afs/ipp/aug/ads/lib64/@sys/libddww8.so.8.1

- **No plan to write a native one**, due to the high flexibilty and amount of user’s options
- **Not available for OS != linux/unix**
- **py2-3 compatible**
- Docu + example: www.aug.ipp.mpg.de/∼git/aug_sfutils

ww.py features

- ww.py works nicely with a lot of object types and formats, I have used it for a decade for TOT/TTH and TRANSP, now also RABBIT
- Probably incomplete, I can add wrapper-methods if needed
- The **method** `write_sf` takes a dictionary and automatically writes the whole shotfile, with good flexibility
- You need a (correct) **shotfile header beforehand**.
  Hereby `sfh.py` can help, at least for reshaping the SFH's objects

SFH

- py-SFH is less developed than dd, ww, kk
- **Reading shotfile headers:** `aug_sfutils.sfhread` native, quick & detailed
- For **modifying shotfile-headers:** `aug_sfutils.sfh`**, wrapper** around `/afs/ipp/aug/ads/lib64/amd64_sles11/libsfh8.so.20201118`
- Feeling: there might be several “private” py-sfh around I am not aware of

“Vision”: **py-xsfed** based on aug_sfutils sfhread.py + extended sfh.py. Werkstudent? Interest?

Miscellany of py utilities

We have a small unburocratic group (A. Bock, T. Lunt, M. Reich) for exchange about py-tools (potentially relevant for several users) available at IPP.

If you have code

- Mature to be shared (good coding practices)
- Documented
- To be imported (no "end-product" like trview, ideview, pyspecview, cdfplayer)

feel free to share it at the wiki

`www.aug.ipp.mpg.de/foswiki/bin/view/PythonAtAUG`

Existing examples: eqdsk r/w; ufiles r/w; output parsers for RABBIT, TORIC, ASCOT-fast ions; separatrix-fit; removing time-frames close to ELMs; exp-profiles Gaussian/spline fits...

Conclusions

- `aug_sfutils` **is a comprehensive package for shotfile r/w and equilibrium manipulation**
- **High compatibility**: py2-3; IPP lxts, tok (SUN: any good py?); Mac, Windows, Linux PCs
- Easy to install: `module load aug_sfutils` on IPP clusters, `pip install aug_sfutils` else/anywhere
- **Fast and stable sf-reading**
- Proper versioning

Homework: run the examples at
`www.aug.ipp.mpg.de/~git/aug_sfutils`
As for most homeworks, copy-paste will do.
Looking forward to feedback