CheasePy

Reconstruct MHD Equilibrium for Modified Plasma Profiles and Geometry

Ehab Hassan[1], Gabrielle Merlo, and David Hatch

IFS, University of Texas at Austin

[1]Currently at Oak Ridge National Laboratory

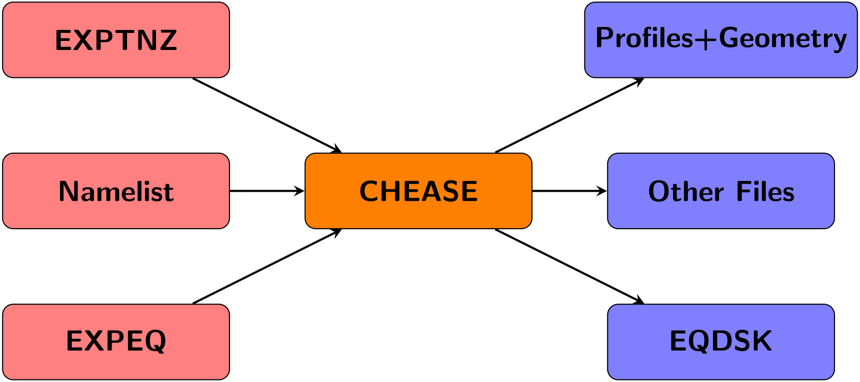
Motivation

The growing need to build new fusion reactors based on various engineering and physics increasing the need for reactor system codes which requires MHD equilibrium for profiles and geometry. This can be achieved using CHEASE code. CheasePy is developed to facilitate running CHEASE code using different types of input files or by importing any user-defined profiles and geometry.

CHEASE Code - INPUTS/OUTPUTS

CHEASE code solves the **Grad-Shafranov** equation which is given by:[2]

$$\nabla \cdot \frac{1}{R^2}\nabla\Psi = \frac{j_\phi}{R} = -p'(\Psi) - \frac{1}{R^2}TT'(\Psi)$$



[2]Lütjens *et al.* CPC1996

Input/Output Types and Data Structures

NAMELIST

Measurement	Description
NS(NT)	Number of radial (poloidal) equilibrium-$\sigma(\theta)$ intervals.
NPSI	Number of radial stability-s intervals.
NCHI	Number of poloidal nodes for ballooning.
NRBOX(NZBOX)	Number of R (Z) points used to save equilibrium in EQDSK.
NSTTP(NPROPT)	Input (output) current profiles (1=ff', 2=I^*, 3=$I_{\parallel}$, 4=$J_{\parallel}$).
NPPFUN	Input pressure profiles (4=P', 8=P).
NEQDSK	Source of equilibrium geometry (0=EXPEQ, 1=EQDSK).
NRHOMESH	Input grid (0=ρ_{ψ_N}, 1=ρ_{ϕ_N}).

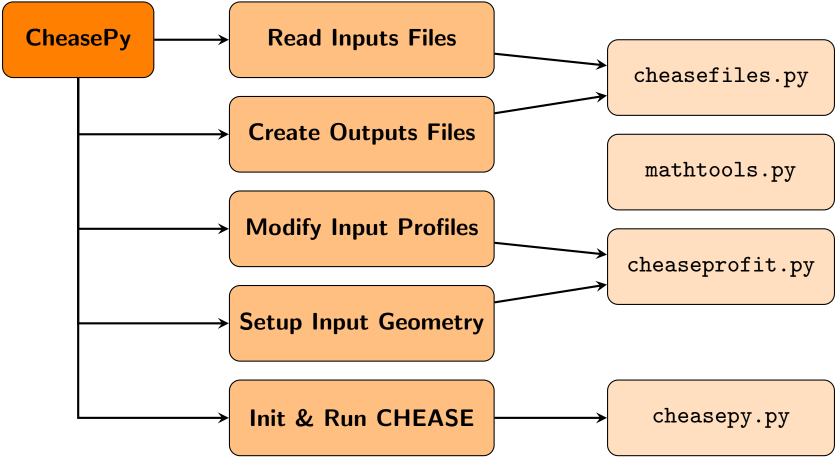
EXPTNZ

Measurement	Description
ρ_{ψ_N} or ρ_{ϕ_N}	Grid
Z_{eff}	Effective Atomic Number
T_i and n_i	Ion Temperature and Density
T_e and n_e	Electron Temperature and Density

EXPEQ

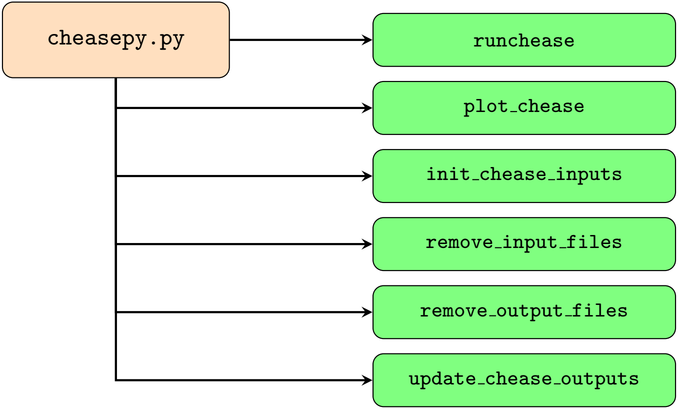
Measurement	Description
ρ_{ψ_N} or ρ_{ϕ_N}	Grid
ϵ	Inverse aspect ratio
r_{bound} and z_{bound}	Boundary Coordinates
P or P'	Pressure Profile
ff', $I_{\parallel}$, $J_{\parallel}$, or I^*	Current (Flux) Density

CheasePy Package Structure



https://github.com/ehabhassan/CheasePy

Modules of CheasePy - `cheasepy.py`



https://github.com/ehabhassan/CheasePy

Modules of CheasePy - cheasepy.py

Example

```
srcVals= {}
srcVals['gfname'] = 'DIIID_162940_EQDSK'
srcVals['iterdbfname'] = 'DIII_162940_ITERDB'
srcVals['inputpath'] = 'shots/DIIID_KEFITD_162940'
srcVals['rhomesh_src'] = 'eqdsk'
srcVals['current_src'] = 'eqdsk'
srcVals['pressure_src'] = 'eqdsk'
srcVals['eprofiles_src'] = 'iterdb'
srcVals['iprofiles_src'] = 'iterdb'
srcVals['boundary_type'] = 'asis'

namelistVals= {}
namelistVals['NS'] = 64
namelistVals['NT'] = 64
namelistVals['NPSI'] = 128
namelistVals['NCHI'] = 128
namelistVals['NRBOX'] = 60
namelistVals['NZBOX'] = 60
namelistVals['NSTTP'] = 3
namelistVals['NPPFUN'] = 8

importedVals= {}
importedVals['Iprl'] = Iprl
```

https://github.com/ehabhassan/CheasePy

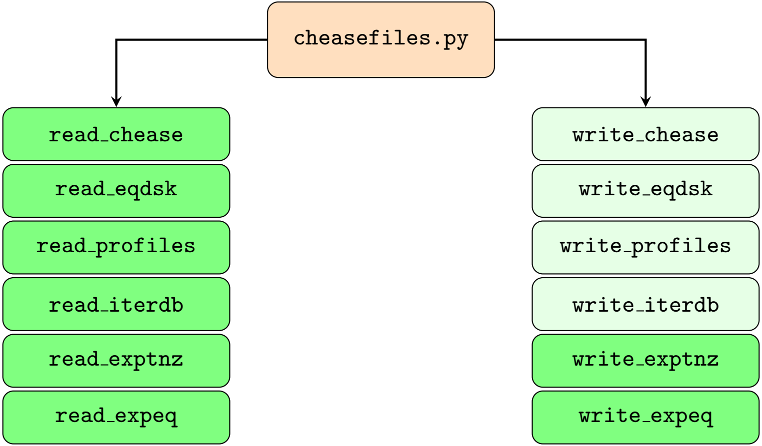
Modules of CheasePy - cheasepy.py

Example

```
remove_input_files()
remove_output_files()
init_chease_inputs(srcVals,namelistVals,importedVals)
runchease()
update_chease_outputs(suffix=0)

namelistVals['NRBOX'] = 513
namelistVals['NZBOX'] = 513

srcVals['rhomesh_src'] = 'chease'
srcVals['current_src'] = 'chease'
srcVals['pressure_src'] = 'chease'
srcVals['inputpath'] = './'
srcVals['cheasefname'] = "chease_iter000.dat"

del importedVals['Iprl']

init_chease_inputs(srcVals,namelistVals,importedVals)
runchease()
update_chease_outputs(suffix=1)

plot_chease(fpath='chease_iter001.dat',skipfigs=0,eqdskfname="EQDSK_iter001")
```

https://github.com/ehabhassan/CheasePy

Modules of CheasePy - `cheasefiles.py`



https://github.com/ehabhassan/CheasePy

Modules of CheasePy - `cheasefiles.py` (read files)

Generally, the **read_xxxx**() methods take the following arguments:

read_xxxx(**fpath**, **setParam**, **kwargs)

With the differences between the read functions embedded in the **setParam** argument.

Argument	Default	Description
fpath	user-input	The path to *TARGET* file.
setParam	{}	nrhopsi=0 for ρ_{ψ_N} and nrhopsi=1 for ρ_{ϕ_N} grid.
		Zeff=True for global Zeff or Zeff=False for local Zeff.
		norm=True for Normalized or norm=False for SI units.
**kwargs	None	Choose a source for the ρ_{ψ_N} and ρ_{ϕ_N} grid to interpolate on.

Example

```
setParam = {'nrhomesh':1,'norm':True}
cheasepath = 'chease_iter000.dat'
eqdskpath = 'g162940.2334'
read_chease(fpath=cheasepath,setParam,eqdsk=eqdskpath)
```

https://github.com/ehabhassan/CheasePy

Modules of CheasePy - `cheasefiles.py` (write files)

Generally, the **write_xxxx**() methods take the following arguments:

write_xxxx(**setParam**, ∗∗**kwargs**)

With the differences between the read functions embedded in the **setParam** argument.

Argument	Default	Description
setParam	{}	Specify the types and sources for different grids and profiles. outfile=True to create EXPTNZ file or outfile=False to return EXPTNZ data
**kwargs	None	Specify the path to the source files.

Example

```
exptnzParam = {}
exptnzParam['outfile'] = False
exptnzParam['nrhomesh'] = [0,'eqdsk']
exptnzParam['eprofiles'] = 'profiles'
exptnzParam['iprofiles'] = 'profiles'

gpath = 'g162940.2334'
pppath = 'p162940.2334'

exptnzDATA = write_exptnz(setParam=exptnzParam,profiles=ppath,eqdsk=gpath)
```

https://github.com/ehabhassan/CheasePy

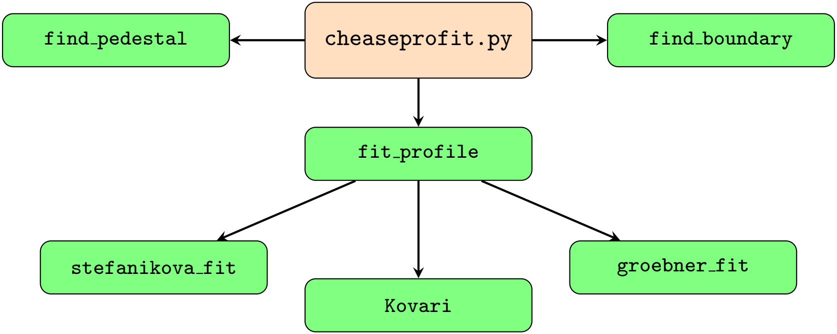
Modules of CheasePy - cheasefiles.py (write files)

Example

```
external = {}
external['Iprl'] = Iprl       % Should be predefined
external['R0EXP'] = 1.7
external['B0EXP'] = 2.2
external['rbound'] = rbound   % Should be predefined
external['zbound'] = zbound   % Should be predefined

expeqParam = {}
expeqParam['outfile'] = True
expeqParam['geometry'] = ['imported']
expeqParam['nrhomesh'] = [0,'expeq'] or ['rhopsi','expeq']
expeqParam['nppfun'] = [8,'expeq'] or ['pressure','expeq']
expeqParam['nsttp'] = [3,'imported'] or ['Iprl','imported']

expeqpath = 'expeq_iter000'

write_expeq(setParam=expeqParam,expeq=expeqpath,imported=external)
```

https://github.com/ehabhassan/CheasePy

Modules of CheasePy - `cheaseprofit.py`



https://github.com/ehabhassan/CheasePy

Modules of CheasePy - `cheaseprofit.py`

Groebner and Stefanikova Fitting Methods and Parameters

$$F_{groebner} = \frac{1}{2}\left(F_{ped\_height} - F_{ped\_sol}\right)\left[mtanh(\alpha_{ped}, z) + 1\right] + F_{ped\_sol}$$

$$F_{stefanikova} = F_{groebner} + \left[F_{cor\_height} - F_{groebner}\right] e^{-\left(\frac{\rho_\phi}{\Delta\rho_{cor}}\right)^{\alpha_{cor}}}$$

$$z = 2\left(\frac{\rho_{ped\_mid} - \rho_\phi}{\Delta\rho_{ped}}\right)$$

$$mtanh(\alpha_{ped}, z) = \frac{(1+\alpha_{ped} z)e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

- α_{ped} = slope of the inner pedestal
- α_{cor} = exponential degree at the core
- ρ_{ped_mid} = ρ_ϕ at the middle of the pedestal
- $\Delta\rho_{ped}$ = width of the pedestal ($\Delta\rho_\phi$)
- $\Delta\rho_{cor}$ = width of the core region ($\Delta\rho_\phi$)
- F_{ped_height} = profile value at the pedestal top
- F_{ped_sol} = profile value at the scrap-off layer
- F_{cor_height} = profile value at core top

Modules of CheasePy - `cheaseprofit.py`

Kovari Fitting Method and Parameters

$$n(\rho) = \begin{cases} n_{ped} + (n_0 - n_{ped})\left(1 - \frac{\rho^2}{\rho_{ped,n}^2}\right)^{\alpha_n} & 0 \leq \rho \leq \rho_{ped,n} \\ n_{sep} + (n_{ped} - n_{sep})\left(\frac{1-\rho}{1-\rho_{ped,n}}\right) & \rho_{ped,n} < \rho \leq 1 \end{cases}$$

$$T(\rho) = \begin{cases} T_{ped} + (T_0 - T_{ped})\left(1 - \frac{\rho^{\beta_T}}{\rho_{ped,T}^{\beta_T}}\right)^{\alpha_T} & \leq \rho \leq \rho_{ped,T} \\ T_{sep} + (T_{ped} - T_{sep})\left(\frac{1-\rho}{1-\rho_{ped,T}}\right) & \rho_{ped,T} < \rho \leq 1 \end{cases}$$

$$n_0 = \frac{(\alpha_n + 1)}{3\rho_{ped,n}^2}\left[3\langle n_e\rangle + n_{sep}\left(-2 + \rho_{ped,n} + \rho_{ped,n}^2\right) - n_{ped}\left(\left(1 + \rho_{ped,n}\right) + \frac{(\alpha_n - 2)}{(\alpha_n + 1)}\rho_{ped,n}^2\right)\right]$$

$$T_0 = T_{ped} + \left[T_{ped}\rho_{ped,T}^2 - \langle T_e\rangle + {}^1\!/_3\left(1 - \rho_{ped,T}\right)\left[\left(1 + 2\rho_{ped,T}\right)T_{ped} + \left(2 + \rho_{ped,T}\right)T_{sep}\right]\right]\gamma$$

$$\gamma = \begin{cases} \frac{-\Gamma(1+\alpha_T+2/\beta_T)}{\rho_{ped,T}^2\Gamma(1+\alpha_T)\Gamma(1+2/\beta_T)} & if\,\alpha_T \in \mathbb{N} \\ \frac{\sin(\pi\alpha_T)\Gamma(-\alpha_T)\Gamma(1+\alpha_T+2/\beta_T)}{\pi\rho_{ped,T}^2\Gamma(1+2/\beta_T)} & if\,\alpha_T \in \mathbb{R} \end{cases}$$

Modules of CheasePy - cheaseprofit.py

Example

```
gpath = "g162940.2334"
ppath = "p162940.2334"
gdata = read_eqdsk(fpath=gpath)
pdata = read_profiles(fpath=ppath,setParam={'nrhomesh':1},eqdsk=gpath)

setparam = {}
setparam['plot'] = True
setparam['norm'] = False

pres_fit_param = fit_profile(pdata['rhotor'],pdata['pressure'],method='groebner',
                   setParam=setparam,fitParam={},fitBounds={})

setparam['norm'] = True
fitparam = {}
fitparam['ped_mid'] = pres_fit_param['ped_mid']
fitparam['ped_width'] = pres_fit_param['ped_width']+0.05

ne_fit_param = fit_profile(pdata['rhotor'],pdata['ne'],method='groebner',
                   setParam=setparam,fitParam=fitparam)

neProfile = groebner_fit(gdata['rhotor'],*ne_fit_param)
```

https://github.com/ehabhassan/CheasePy

Future Development Plan:

- Add more models for profile fitting.
- Add read (write) capabilities from (to) databases.
- Add read (write) capabilities from (to) files of different formats.

Thank You

Questions are Welcome ...