A brief user guide to OpenPOPCON

Richard Nies

September 2020

Contributors to the code are Sam Frank, Richard Nies, Tal Rubin, YOUR NAME HERE.

1 Introduction

OpenPOPCON is a useful 0D scoping tool of tokamak performance. With many properties prescribed, e.g. profile shapes, confinement time, etc, the plasma can be parametrised by only density and temperature and a vast amount of information can be obtained from only these two parameters. An example plot is shown in Fig. 1 to demonstrate the usefulness and richness of POPCONs.

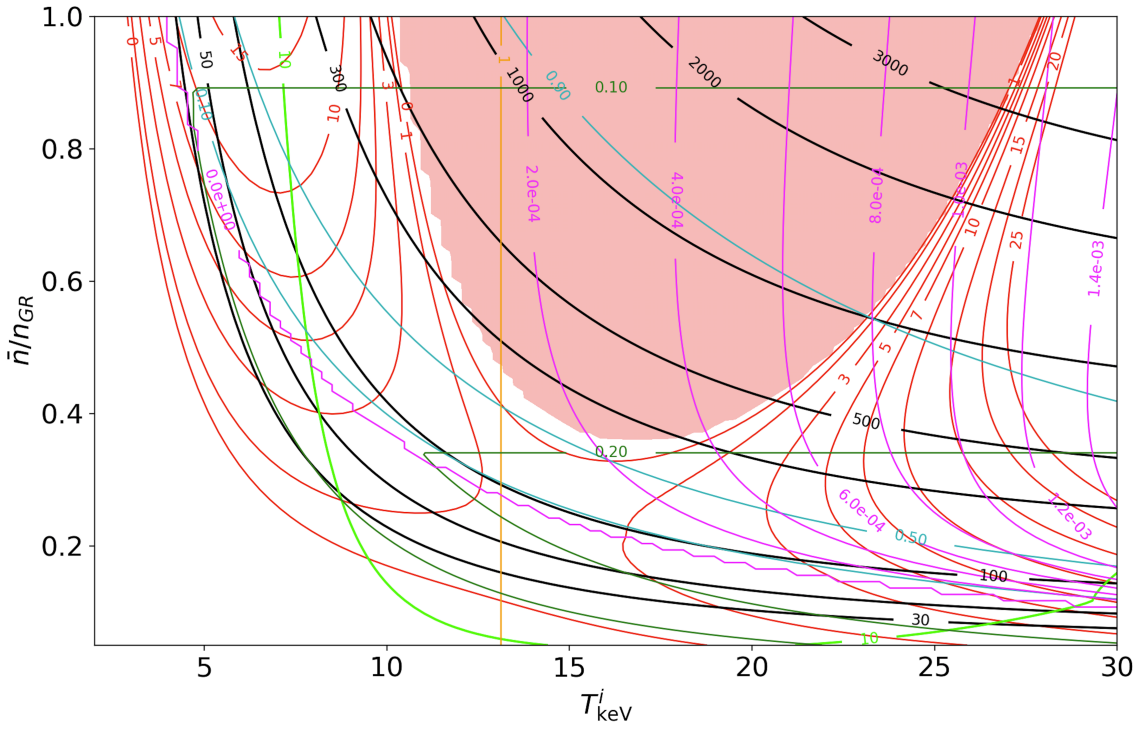

Figure 1: Example POPCON for SPARC-x2 parameters and $P_{\mathrm{SOL}} \leq 20$ MW. Plotted are the ignition region (red shaded), auxiliary power in MW (red), fusion power in MW (black), Q (lime), $P_{\mathrm{rad}}/P_{\mathrm{loss}}$ (cyan), P_{ohm} in MW (orange), f_{LH} (dark green) and n_{imp}/n_e (magenta).

2 Getting started

From the root ARCH-CORE directory on GitHub (`https://github.com/ARCH-reactor/CORE`), the code can be found in `Open_POPCON/openpopcon.py`. An example file to run the code can be found at `Open_POPCON/test_POPCON/run_popcon.py`. Going into this directory and running the file (e.g. `python3 run_popcon.py`) should provide you with your first POPCON!

You can now get started running the POPCON for your own desired set of parameters. When running from another directory, make sure to add the path of the ARCH root directory with `sys.path.append(...)` in the `run_popcon.py` file. This will enable loading of all the required modules, i.e. OpenPOPCON but also the fitting coefficients for $L_{\rm rad}$ and $P_{\rm fus}$.

There are many input parameter. We could list them all here but the inevitable passage of time is bound to make this task redundant. Instead, a good place to start is to inspect the list of parameters output by the code when running it, most of which can be adjusted directly following the example in `run_popcon.py` (e.g. `mypops.Ip=17.4`). More avid readers or those confused by the variable names should open the source file `openpopcon.py` with their favourite text editor, and search for the `__init__` function (probably around line 30 of the code), below which the parameters are named and hopefully commented well enough to understand. If not, get in touch with the friendly and helpful developers and suggest improvements to the code and/or this user guide.

One input parameter deserving special consideration is `fixed_quantity`, which sets the method to determine the radiated power $P_{\rm rad}$. More details on this can be found in Sec. 3.

The parameter `matlen` sets the number of points for which to run the POPCON. If this number exceeds 3, a contour plot as shown in Fig. 1 is automatically displayed. If it is equal to or smaller than 3, radial profiles of different quantities are displayed (currently, $P_{\rm rad}, P_{\rm fus}, P_{\rm ohm}$), as examplified in Fig. 2.

3 More details on code function

The way the code works is to solve power balance at each operating point (n,T). In other words, the code determines the auxiliary power $P_{\rm aux}$ until force balance is achieved, i.e. $P_{\rm tot} = W_{\rm th}/\tau_E$, where $P_{\rm tot}$ contains contributions from the fusion power, ohmic power, auxiliary power (e.g. RF, NBI) and radiated power. This problem is highly nonlinear, e.g. τ_E might depend on heating power in the ITER89 scaling. The solver iteratively adjusts $P_{\rm aux}$ until force balance is achieved.

The radiated power can be determined in different ways, regulated through the parameter `fixed_quantity`. Currently, three methods are implemented:

`fixed_quantity="impfrac"` : the impurity fraction `impfrac` is prescribed, which straightforwardly prescribes $P_{\rm rad}$ for a given n and T.

`fixed_quantity="f_LH"` : the radiated power is iteratively adjusted to keep the power at the scrape-off-layer $P_{\rm SOL}$ certain fraction of the LH threshold (the power at which the plasma goes from L- to H-mode), set by the parameter `f_LH`.

`fixed_quantity="Psol"` : the radiated power is iteratively adjusted to keep the power at the scrape-off-layer $P_{\rm SOL}$ below a value given by `Psol_target` (in MW).

For these two latter cases, the impurity fraction has to be adjusted iteratively to obtain the desired radiated power. This in turn impacts the plasma dilution and $Z_{\rm eff}$, which themselves impact $P_{\rm fus}, P_{\rm ohm}$, etc.

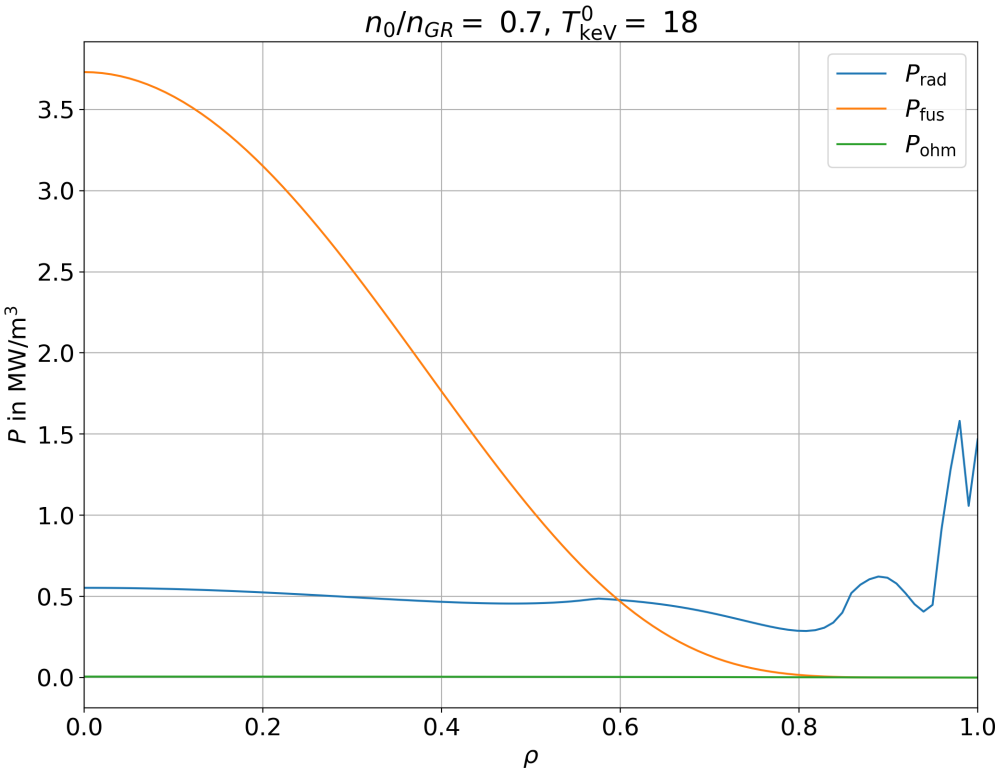

Figure 2: Example POPCON radial profiles for SPARC-x2 parameters.