FreeGS November 2020 Update Documentation

Chris Winnard

November 2020

1 General New Features

- Several new general-use features have been added to equilibrium.py.
- None of the following require input arguments. E.g to print a result: print(eq.AveragePlasmaPressure()).
- To call one of these functions within equilibrium.py you'd simply use: *self*.plasmaArea() + *self*.AveragePlasmaPressure(), as an example where you are adding the plasma area to the average plasma pressure.
- A function plasmaArea(), which operates very similarly to the already existing plasmaVolume(), has been added.
- A function AveragePlasmaPressure() has been added.
- plasmaW(), which calculates the internal energy (the plasma W), has been added.

2 New Functions for ITER-Like Tokamaks

- In addition to the above changes, several other functions have been added, but for the time being **these are tailored to ITER-like tokamaks operating in H-mode**.
- However, **this can be changed by changing the front factor of AverageElDensity() and the scaling parameters of tauE_Coeffs**.
- These functions also do not require input arguments, and can be called like those in the previous section.
- The following functions calculate average densities: AverageElDensity(); AverageDT_IonDensities() (returns a matrix with the two densities); AverageImpurityDensity(); AverageTotDensity().
- ElDensityRatio() calculates the electron peak-to-average density ratio.

- Working from those functions, the following calculate the peak densities: PeakElDensity(); PeakDT_IonDensities() (returns a matrix with the two values); PeakImpurityDensity(); PeakTotDensity().
- AverageTemperature() calculates the average particle temperature, by using the ideal gas law considering all particles.
- The function PowerL() calculates the loss power of the plasma.
- The function tauE() calculates the confinement time.
- The function tauE_Coeffs is required for PowerL() and tauE() to run.
- By default, AverageElDensity() uses 300 npoints (i.e, 300 points on the plasma's outer edge) when it calls on the minorRadius function, but this can be changed as appropriate. As a guide, with npoints = 20 the results were observed to diverge at the second significant figure.

3 Additional Changes to the Code

- Some bugs in the last update to equilibrium.py have been patched.
- In this endeavour, changes have been made to the following functions: the three internalInductance functions, poloidalBeta2(), intersectsWall(), and effectiveElongation().
- effectiveElongation(), which calculates the elongation of the plasma using its volume, does not require inner and outer wall positions as input arguments anymore.