UQ for Atomic Data

2021-12

2nd Meeting of the Global Network for the Atomic and Molecular Physics of Plasmas
6 – 9 December 2021

David Coster
Max Planck Institute for Plasma Physics

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 800925.

Summary

- Apply UQ to the coronal average charge state problem
- Use 3 parameters to vary the ionisation rates and 3 for the recombination rates
- Get the rates either from the ITER AMNS library (for this data, based on ADAS) or directly from ADAS
- Explore the variance in the results and the Sobol indices which explain the variance
- Compare the variances with those arising from different ADAS data-sets

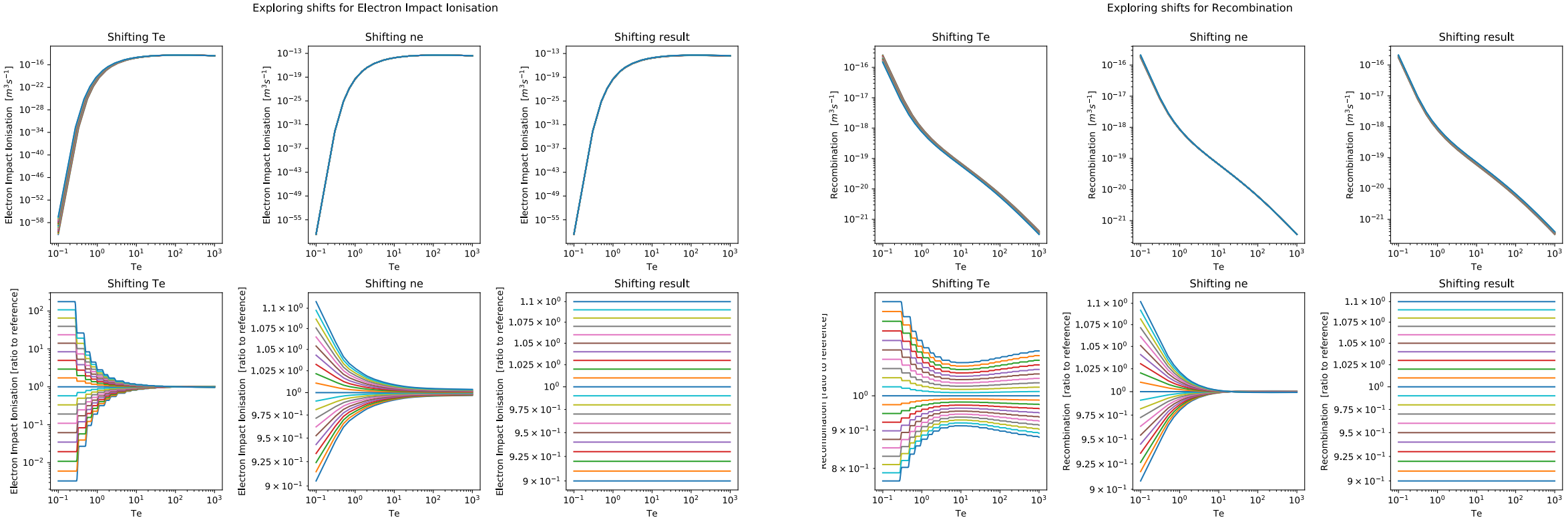
- **Need a better way of parametrising the uncertainty in the atomic data**
- **Need better data**
- **Look to apply this to SOLPS runs to explore the impact of uncertainty in the atomic data**

How to add UQ?

- Approach taken here
  - Have Rate(te, ne) from, for example, ADAS
    - Change this to Rate(Te*$\mathbf{v}_{Te}$, ne*$\mathbf{v}_{ne}$) * $\mathbf{v}_{rate}$
      - With the **v**'s varying around 1
  - Do this for ionisation and recombination rates
  - Then solve for the coronal equilibrium average charge distribution, as a function of these varying **v**'s
- We can use the same methodology if we have better ways of parameterising the uncertainties in the atomic data

Results for applying these **v**'s …

- Use (here)
  - **v** = [0.9,1.1] in steps of 0.01



Now use the VECMA EasyVVUQ toolkit

VECMA

- https://github.com/UCL-CCS/EasyVVUQ
- Suleimenova, Diana, Hamid Arabnejad, Wouter N. Edeling, David Coster, Onnie O. Luk, Jalal Lakhlili, Vytautas Jancauskas, et al. 'Tutorial Applications for Verification, Validation and Uncertainty Quantification Using VECMA Toolkit'. Journal of Computational Science, June 2021, 101402. https://doi.org/10.1016/j.jocs.2021.101402. [And references therein]
- Python package capable of running UQ "campaigns"

Some details

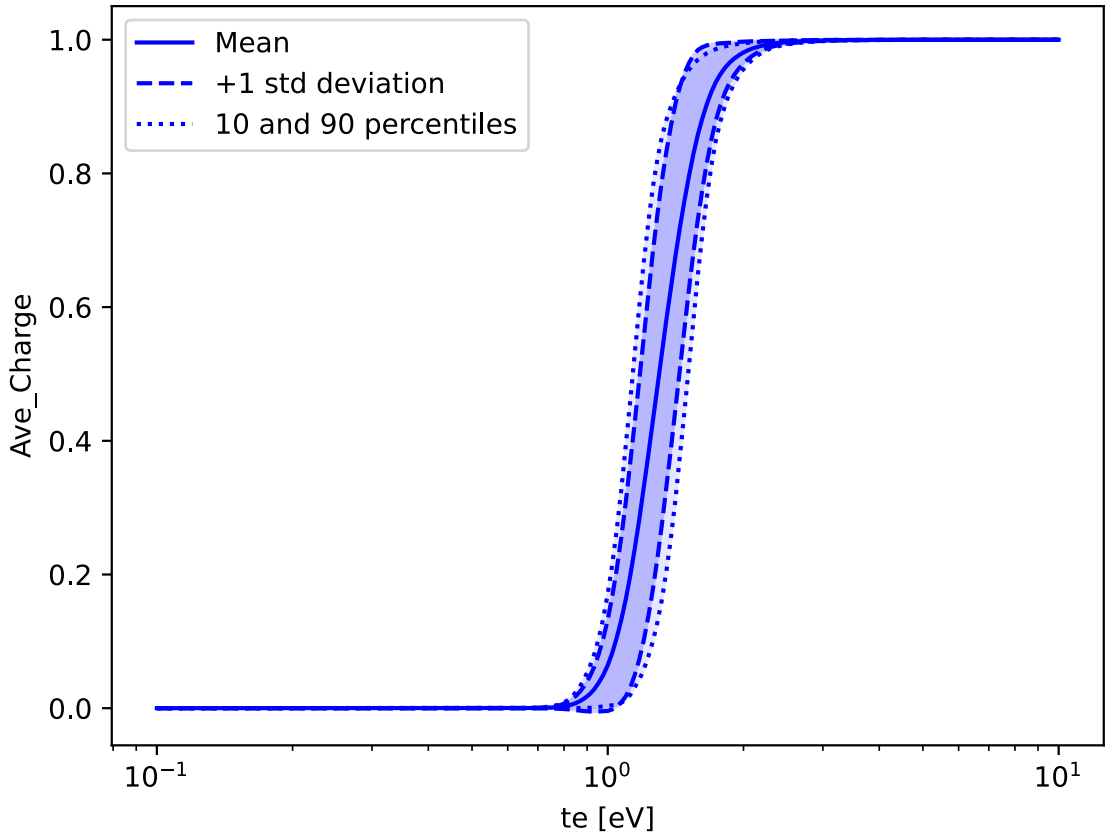
- Here we assume an uniform distribution of the **v**'s in the interval [0.8, 1.2]

```
vary = {
    "El_te_vary":   cp.Uniform(0.8,   1.2),
    "El_ne_vary":   cp.Uniform(0.8,   1.2),
    "El_rate_vary": cp.Uniform(0.8,   1.2),
    "RC_te_vary":   cp.Uniform(0.8,   1.2),
    "RC_ne_vary":   cp.Uniform(0.8,   1.2),
    "RC_rate_vary": cp.Uniform(0.8,   1.2)}
```

- Use Polynomial Chaos Expansion with varying order to evaluate the statistical information

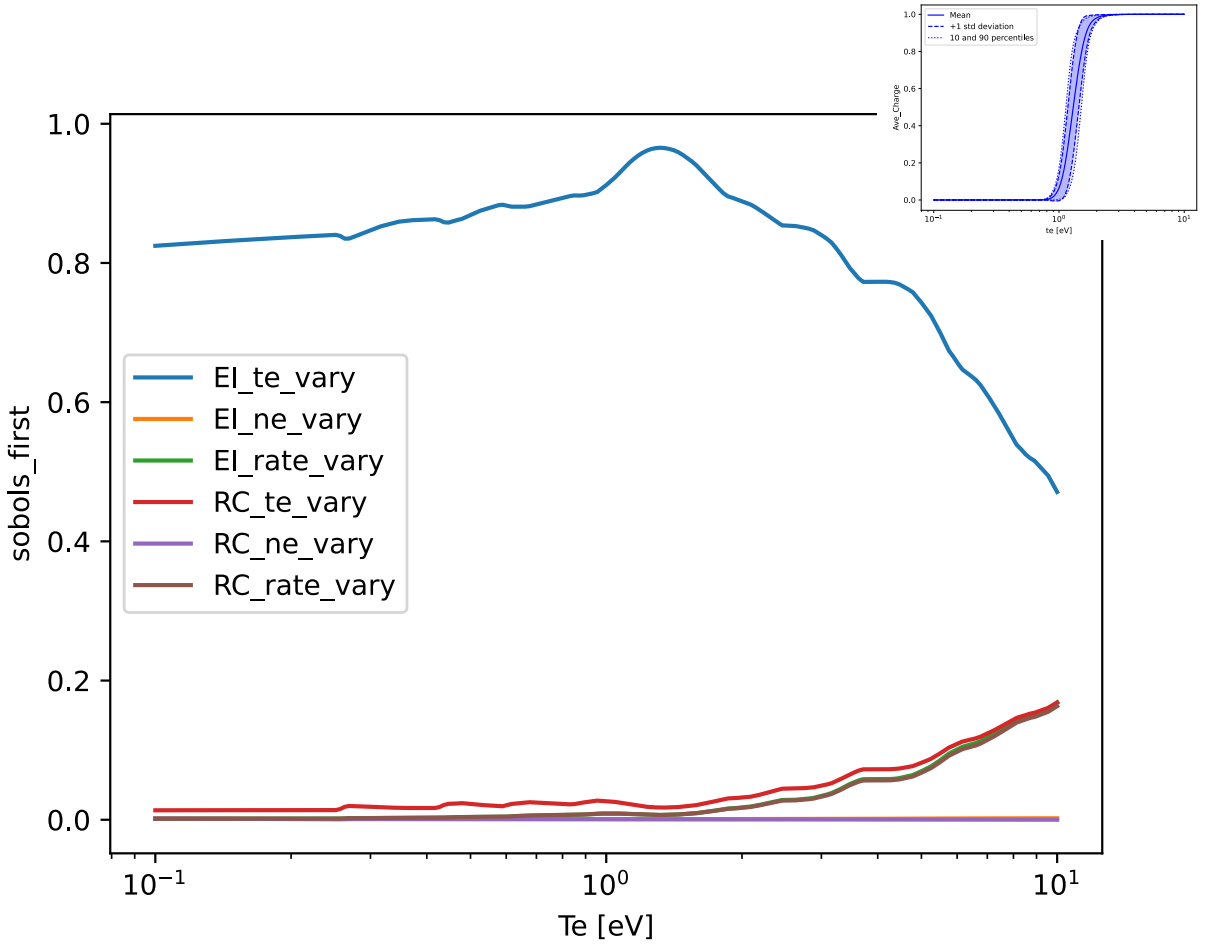
Results for H, ne=3e19

- Mean average charge as a function of Te
- With + and - 1 standard deviation
- And the 10 and 90 percentiles
- To understand where the variance is coming from, we need to look at the Sobol indices …



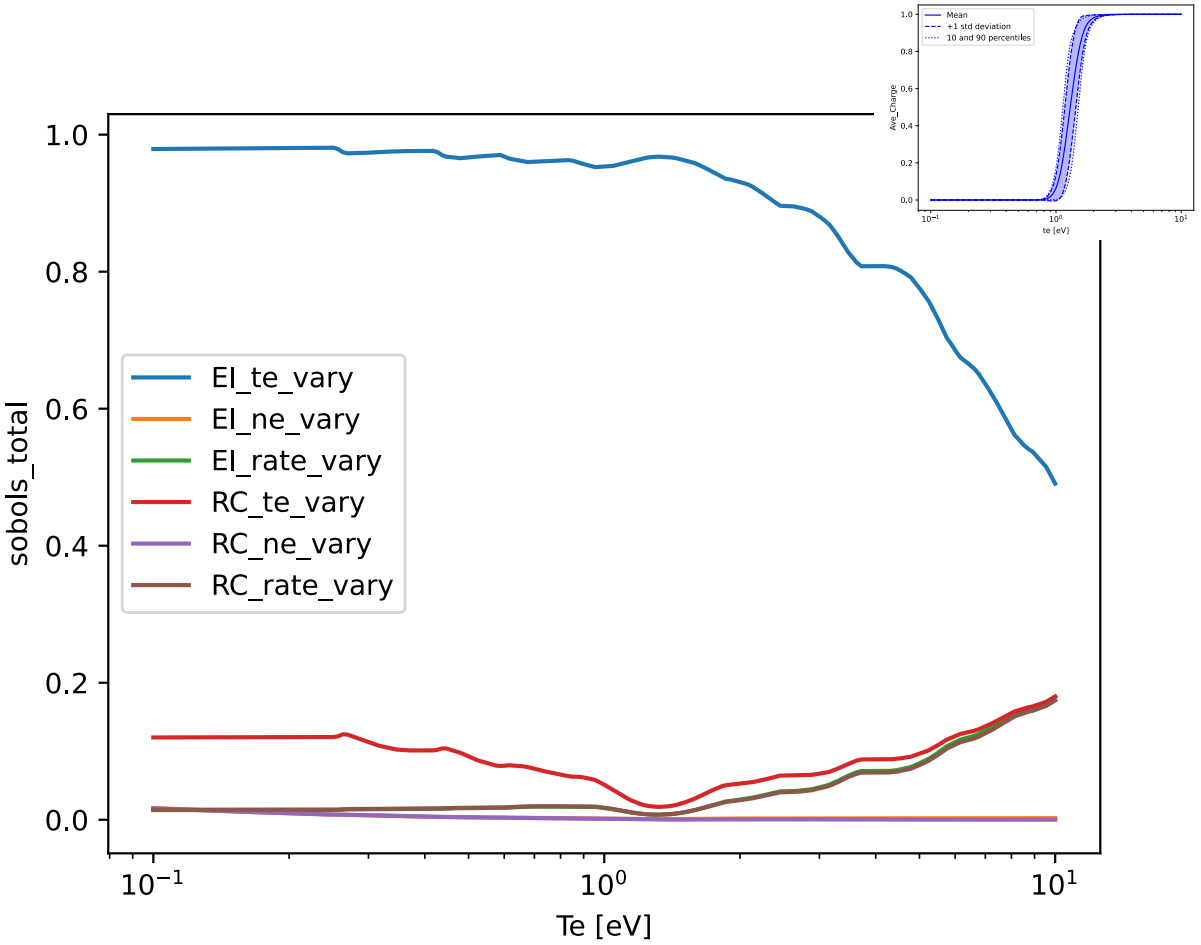
Results for H, ne=3e19

- For the first Sobol
  - We see that over the whole domain, contribution of varying the Te argument to the EI rate is the most important
  - At higher temperatures, we start to see increasing contributions from the RC variation: the Te argument variation and the variation in the rate



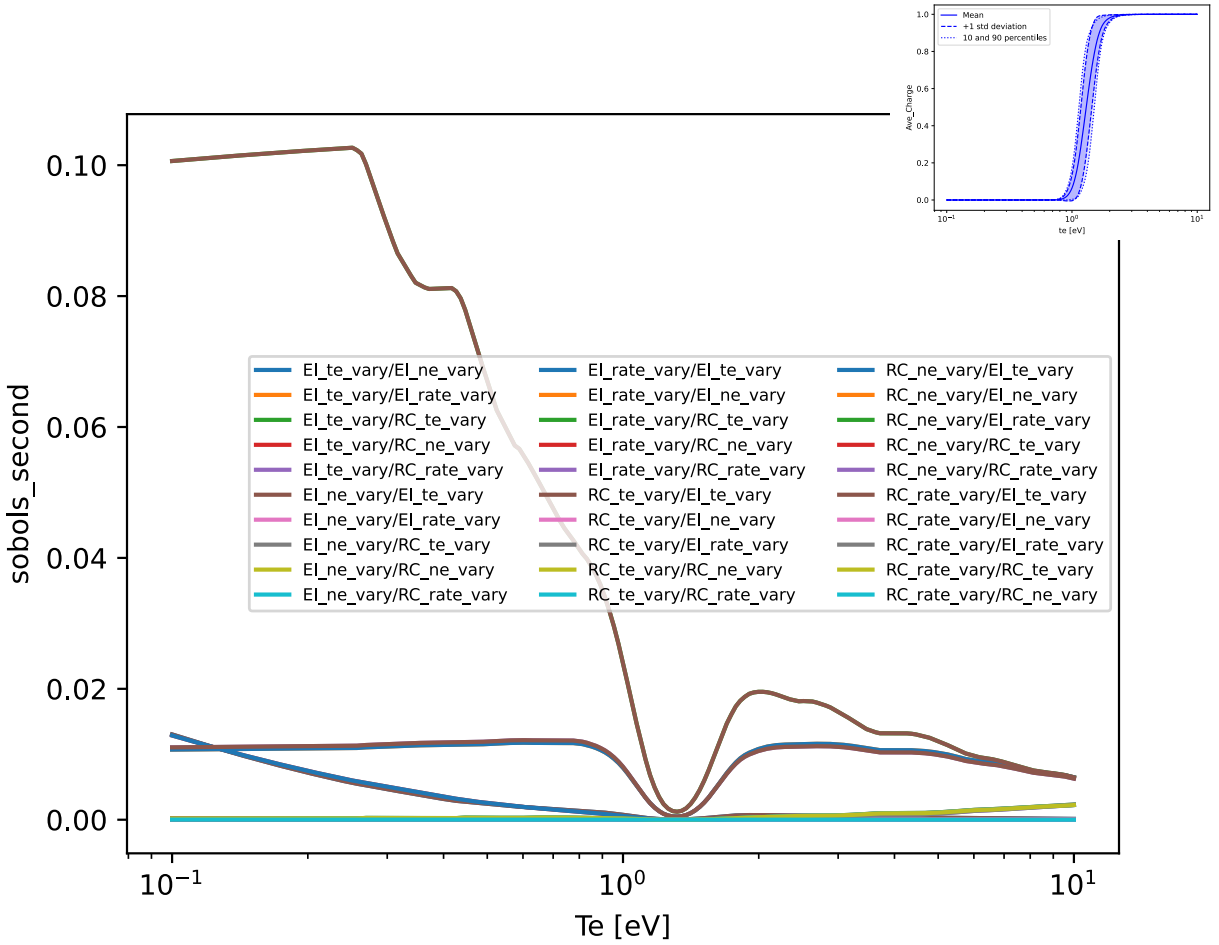
Results for H, ne=3e19

- Similar results for the total Sobol
  - We see that over the whole domain, contribution of varying the Te argument to the EI rate is the most important
  - At higher temperatures, we start to see increasing contributions from the RC variation: the Te argument variation and the variation in the rate
  - Also see a contribution of the Te varying RC contribution at lower temperatures



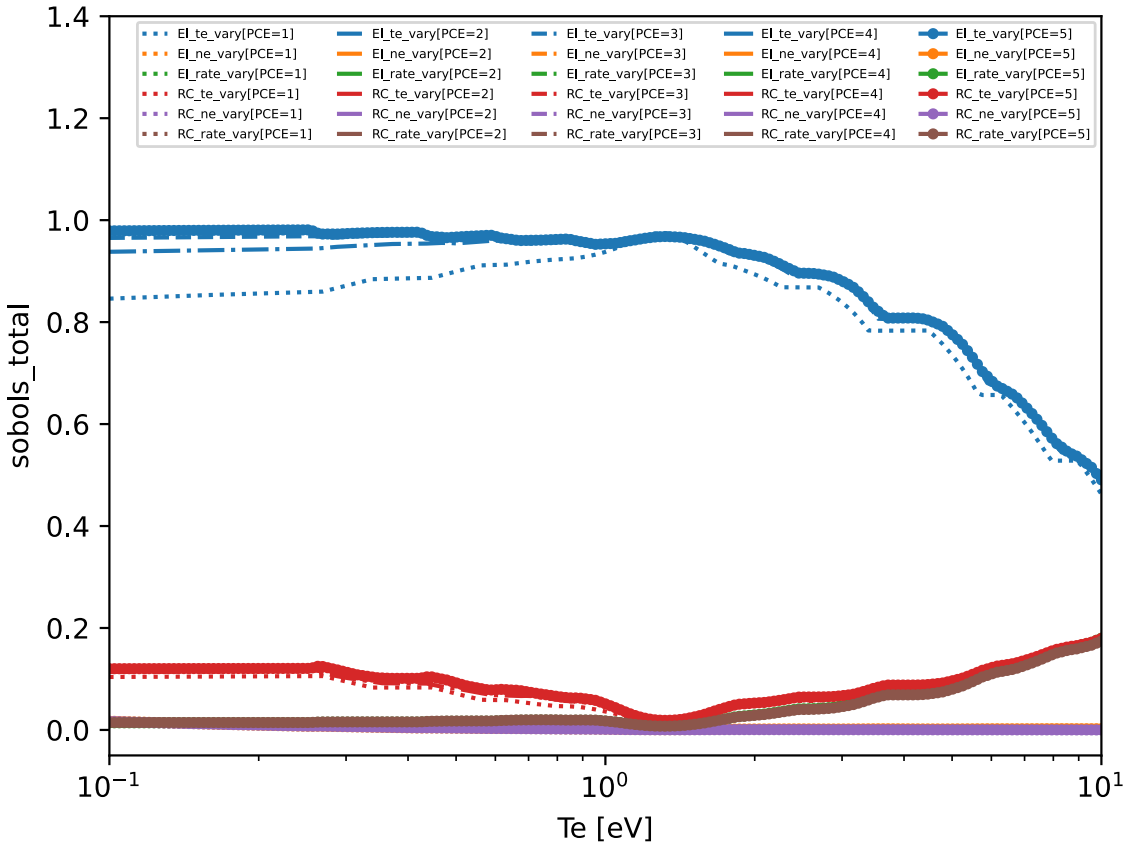
Results for H, ne=3e19

- Explanation for the difference can be seen from the second Sobols
- Largest contributions from
  - RC_te_vary / EI_te_vary
  - EI_te_vary / RC_te_vary



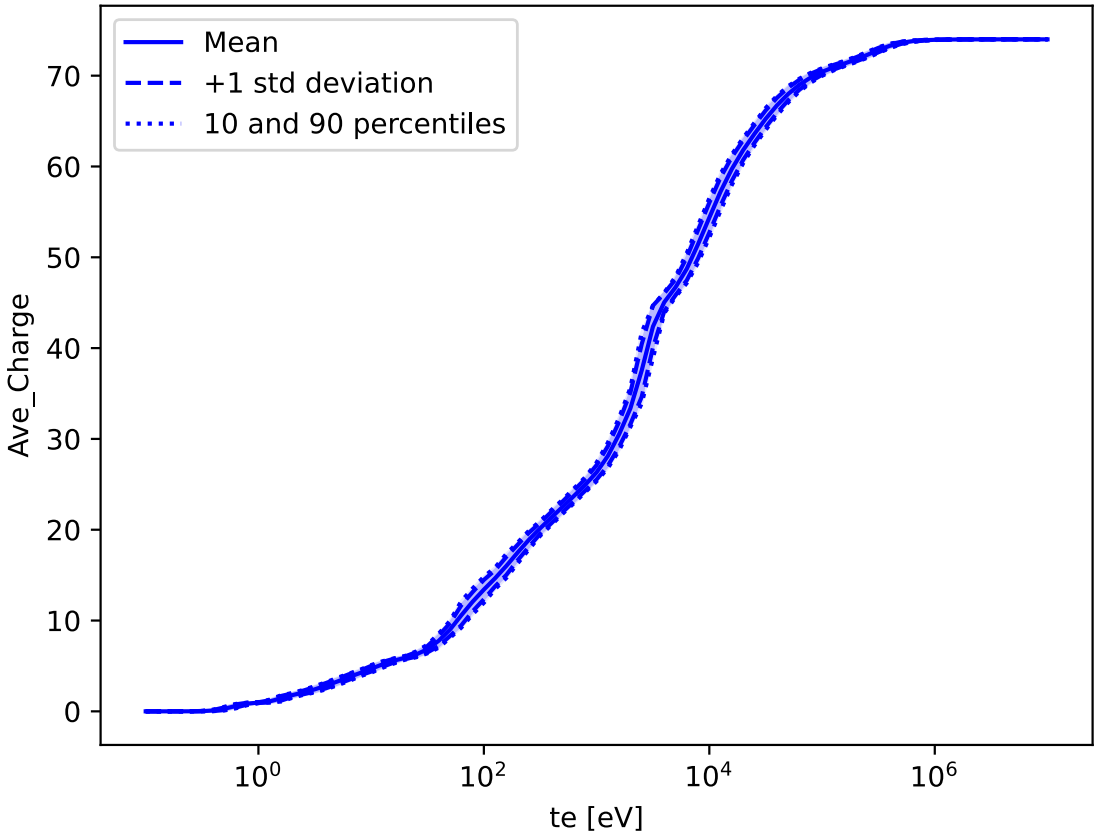
Need to check the convergence with PCE order

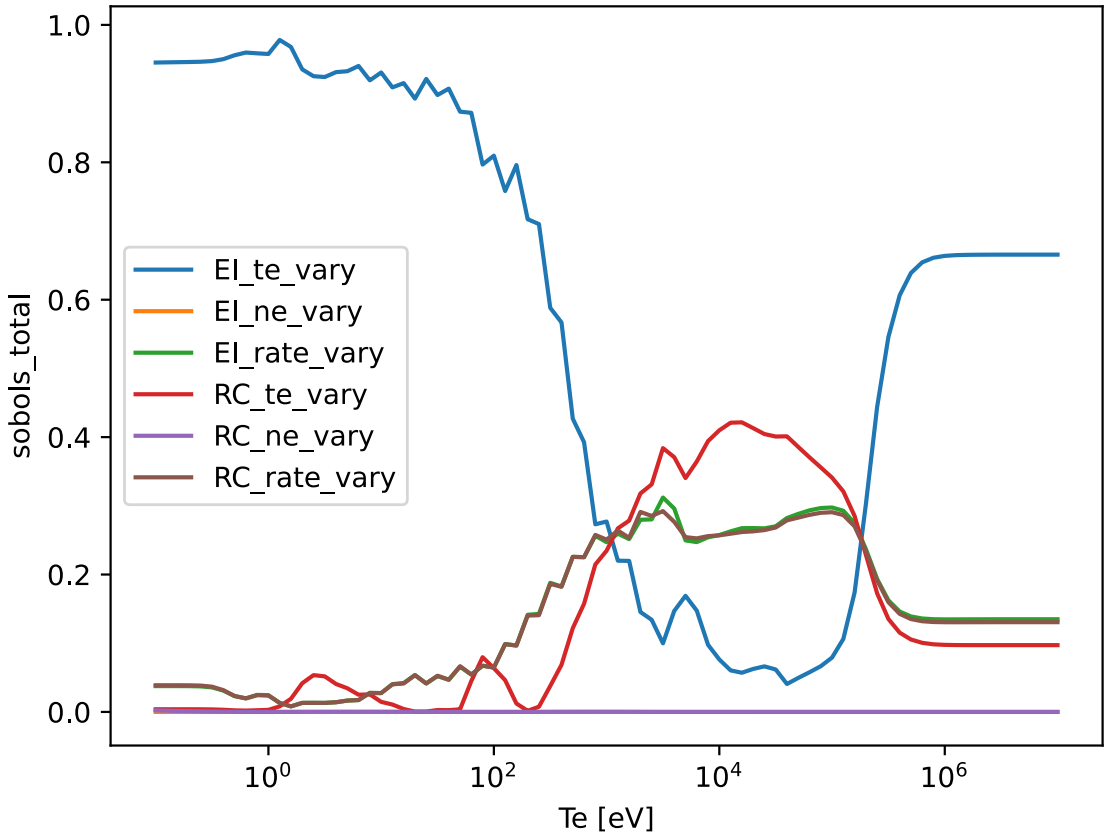
- Scan PCE order 1..5
- The Sobols seem to converge quite well, and order 3 seems to be a good compromise between accuracy and speed
  - Order 3 requires 4096 samples
  - Order 5 requires 46656 samples



Now consider the results for W

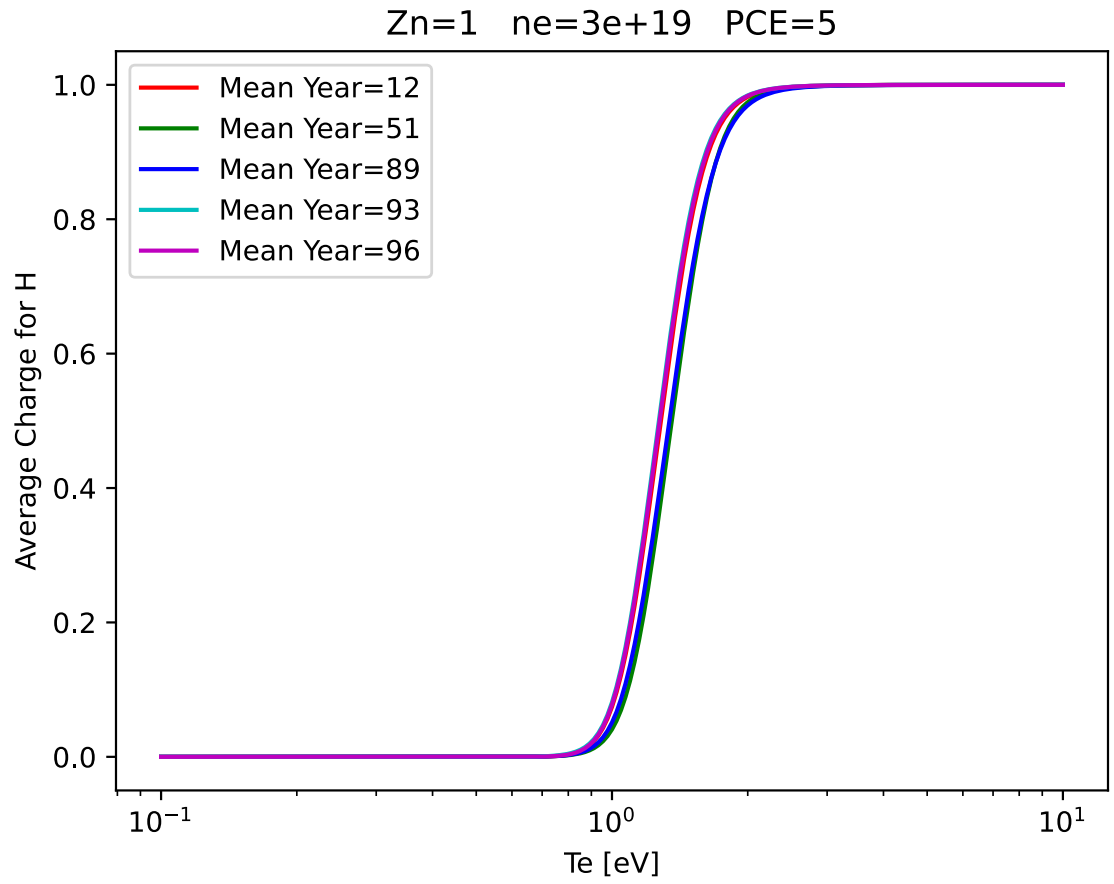
- Average charge and total Sobols: EI_te_vary dominant over most of the range except for an important Te range where RC_te_vary takes over …





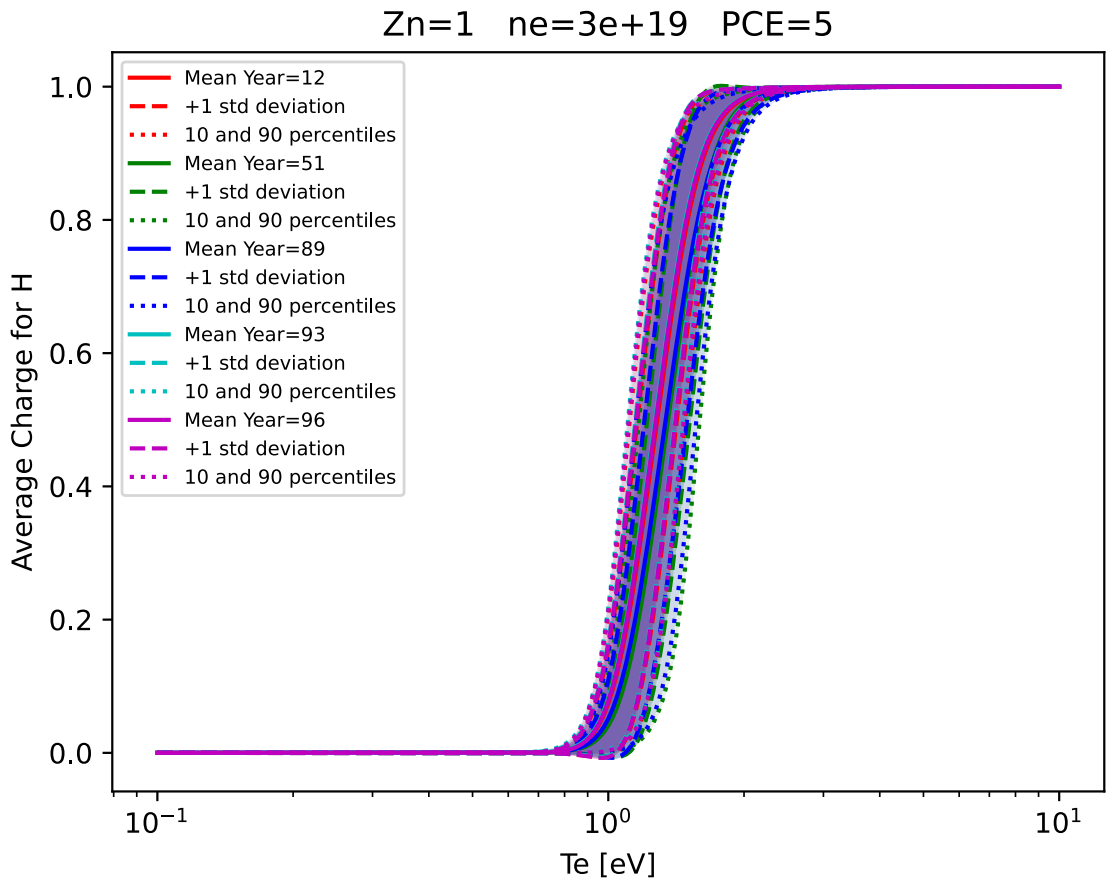
ADAS has multiple data sets: compare UQ variation to those arising from the different data sets: H

- Different ADAS data-sets for H are very similar
- Plotting the distributions doesn't tell us much more …



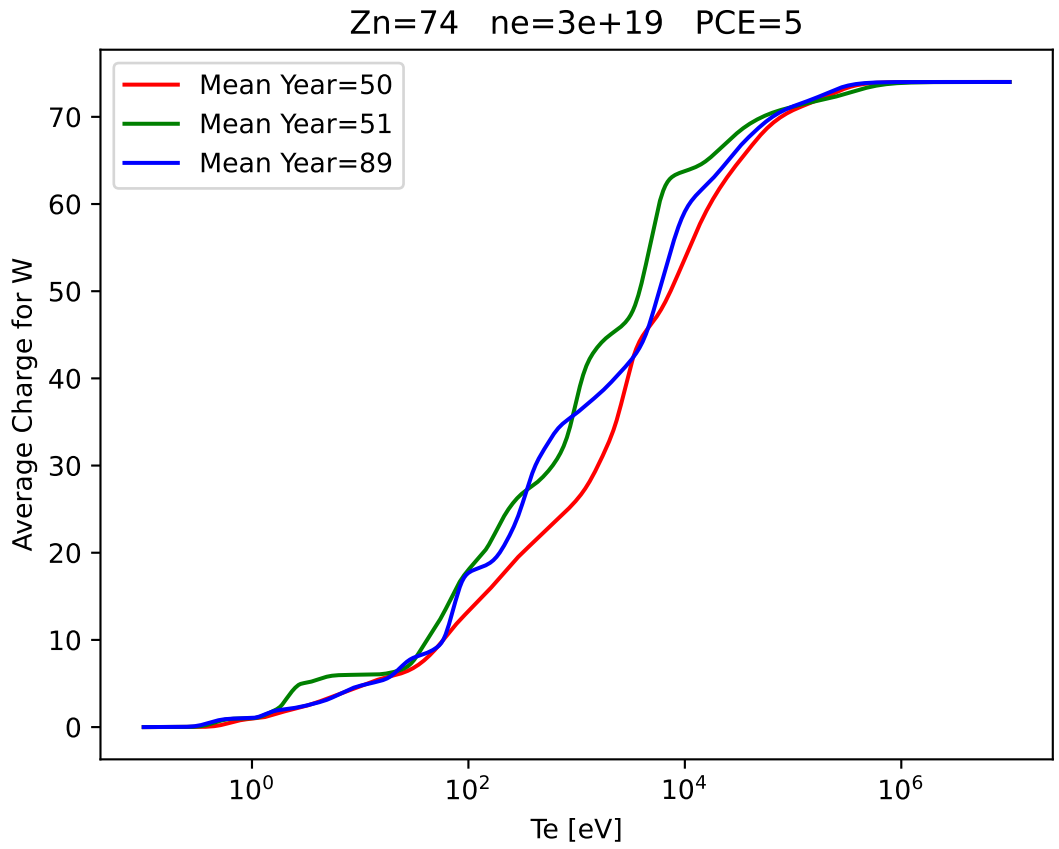
ADAS has multiple data sets: compare UQ variation to those arising from the different data sets: H

- Different ADAS data-sets for H are very similar
- Plotting the distributions doesn't tell us much more …



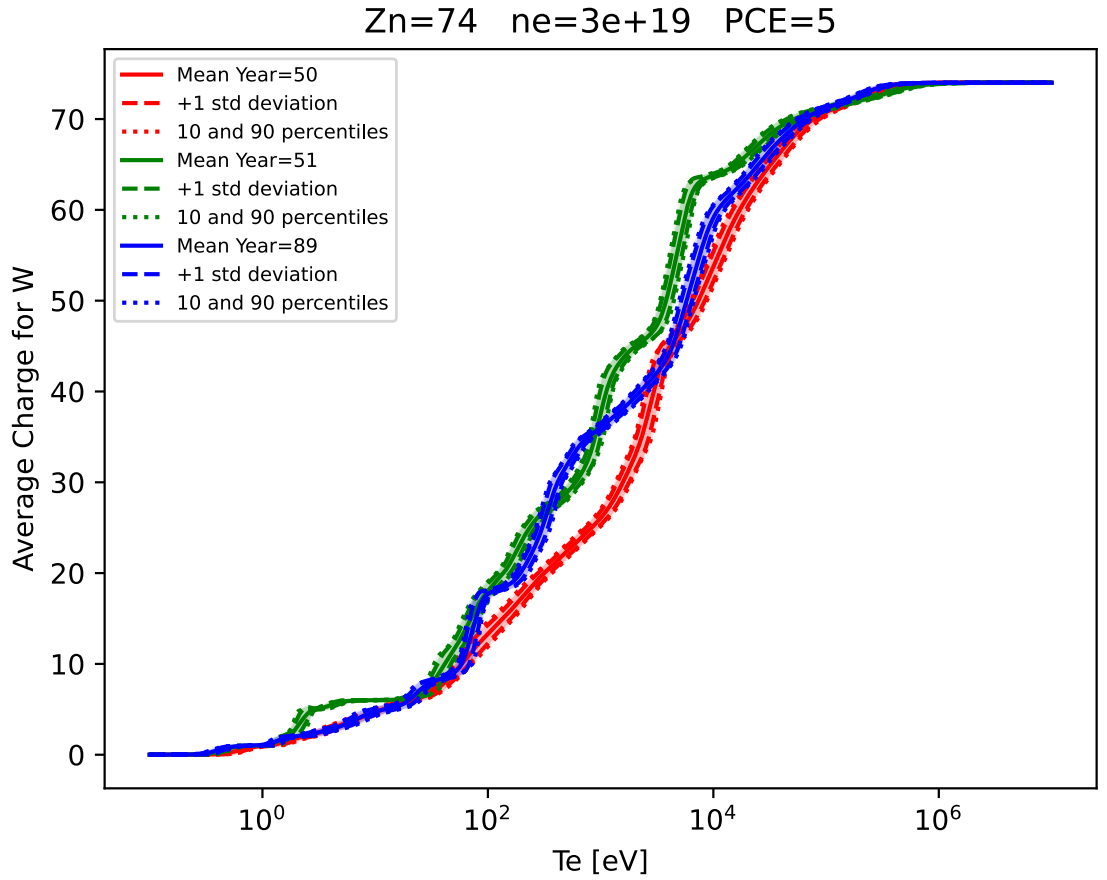
ADAS has multiple data sets: compare UQ variation to those arising from the different data sets: W

- A lot of variation between the W data sets



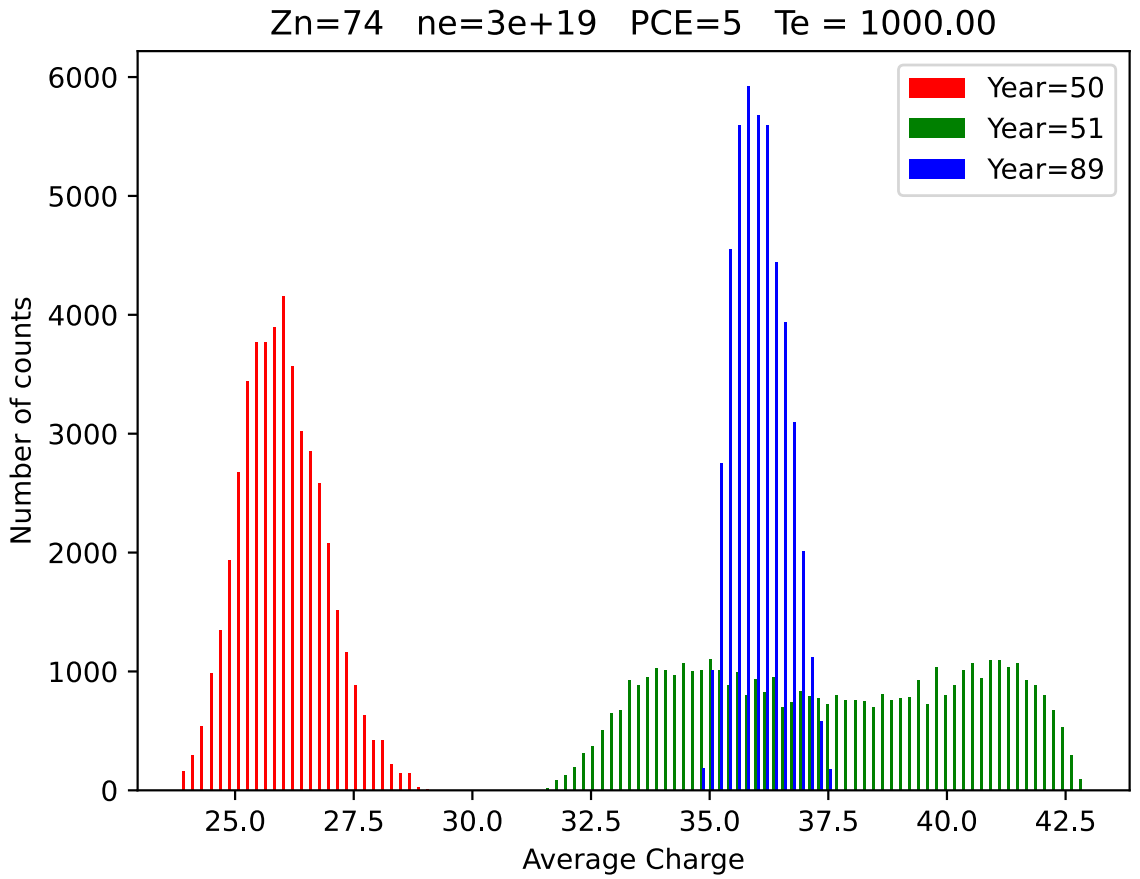
ADAS has multiple data sets: compare UQ variation to those arising from the different data sets: W

- A lot of variation between the ADAS W data sets
  - Much larger than the variance coming from our choice of varying coefficients
  - Would be good to get agreement on “better” W data!



ADAS has multiple data sets: compare UQ variation to those arising from the different data sets: W

- A lot of variation between the ADAS W data sets
  - Much larger than the variance coming from our choice of varying coefficients
  - Would be good to get agreement on “better” W data!
  - At 1000eV the average charge is somewhere between 24 and 43!



In conclusion

- Need a better prescription for how to vary the atomic physics data in order to do a sensitivity study
  - Ideally with a few parameters that can alter the information from ADAS
- While we can do a 5th order PCE campaign for the coronal cases, we want to look at the impact on edge simulations
  - 46656 samples took 10767.549 seconds (~0.23 seconds per case)
  - SOLPS run with just H might take a few hours without drifts and a week with drifts for each sample
  - Can do dimension adaptive analysis, though
    - In a different context 1 048 576 samples was reduced to 1245, 40 days to 6 hours