Max-Planck-Institut für Plasmaphysik

Building a Turbulence-Transport workflow incorporating uncertainty quantification for predicting core profiles in a tokamak plasma.

D. P. Coster*, R. Fischer, O. Hoenen, J. Lakhlili, O. O. Luk, R. Preuss, B. D. Scott, U. von Toussaint and the ASDEX Upgrade Team

Max Planck Institute for Plasma Physics, Boltzmannstr., D-85748 Garching, Germany

Introduction

- Magnetically confined plasmas hold out the hope for a nearly inexhaustible supply of energy
- Important to understand energy transport in the core plasma
- 1-D core plasma treatment is a common approach
- Challenge is to determine the fluxes / transport coefficients
- Here do this by building a workflow that includes a turbulence code

Workflow

- Use a multi-scale approach to address the disparity of time- and space-scales
- Implemented a workflow incorporating
  - a transport code (in this case, the ETS [1])
  - a fixed boundary equilibrium code (here, CHEASE [2]
  - a turbulence code (GEM [3])
  - an utility code that converts fluxes to transport coefficients

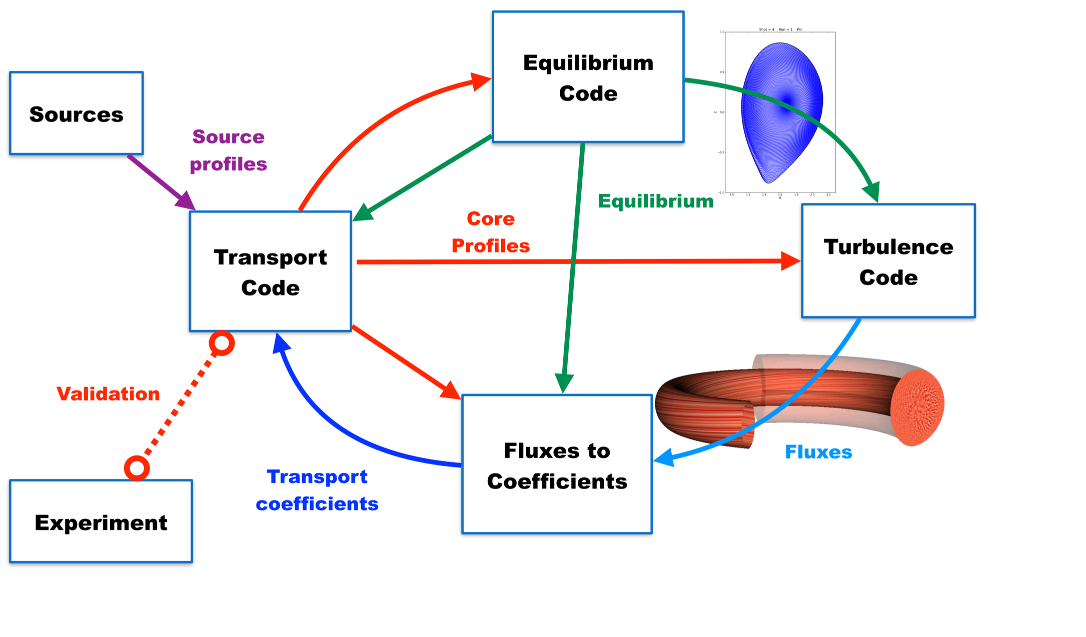

Figure: The Transport, Turbulence and Equilibrium Workflow.

- start from some initial profiles for the temperatures and densities
- repeat until converged
  - repeat until out of time in the queue
    - equilibrium code calculates new metric coefficients
    - these, together with the temperature and density profiles, are used by the turbulence code to advance the state of the turbulence simulation for a time window $\Delta t_{turbulence}$ that allows for some evolution of these fluxes
    - these fluxes are then converted to transport coefficients
    - these transport coefficients are used by the transport code to evolve the profiles over a time window $\Delta t_{transport}$
- genesis of this approach is described in [4]
- but was moved from the infrastructure described in that article to using the MUSCLE2 coupling framework [5] to implement the workflow in a tightly coupled, modular and extensible manner, with the unit of exchange between the codes the appropriate Consistent Physical Object (CPO) [6]
- development of the workflow is described in [7], [8] and [9]
- Work is ongoing to switch to the new MUSCLE3 [10], framework [https://github.com/multiscale/muscle3]

Uncertainty Quantification (UQ)

- use the EasyVVUQ library [11] developed as part of the VECMA project [https://www.vecma.eu/]
- start by treating the workflow as a black-box and then apply one of the standard UQ techniques to the inputs and outputs of the black-box
  - applied to a simplified heat transport model as a tutorial for the fusion workflow
    - more information can be found at the VECMA EasyVVUQ tutorial website [https://easyvvuq.readthedocs.io/en/dev/fusion_tutorial.html]
  - applied to the fusion workflow with GEM0 (an analytic replacement for GEM)
  - not yet applied to the fusion workflow with GEM (too expensive given current computing resources)
- Work has now started on opening up the workflow so that UQ is applied at the code level within the workflow, and rather than just passing profile between codes, distributions of profiles are passed.

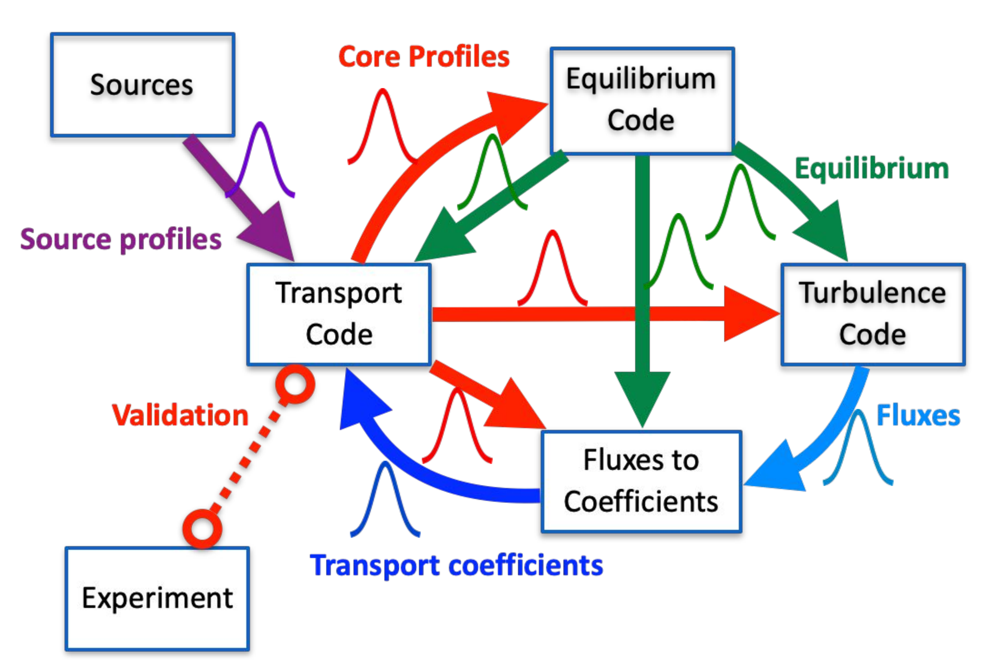

Figure: The planned Transport, Turbulence and Equilibrium Workflow passing distributions.

- plan is to have each code box produce a distribution of results which would be sampled by the next code box
- could use PCE to generate these distributions

Results

Uncertainty Quantification (UQ) results are shown for

- the simple fusion tutorial example
- Transport, Turbulence and Equilibrium Workflow using the GEM0 code in place of the GEM turbulence code

Physics results are shown for

- the full Transport, Turbulence and Equilibrium Workflow and a comparison with experimental data
- results from the full Transport, Turbulence and Equilibrium Workflow comparing the usual positive triangularity with an artificially created negative triangularity AUG discharge are shown

Uncertainty quantification for the simple fusion tutorial example

The advantage of the simple fusion tutorial implementation is that the source code is open and it is fast. It finds the steady-state heat conduction equation in cylindrical geometry using the "Finite Volume PDE Solver Using Python", Fipy [12], in about 20ms allowing for fast investigations of UQ options.

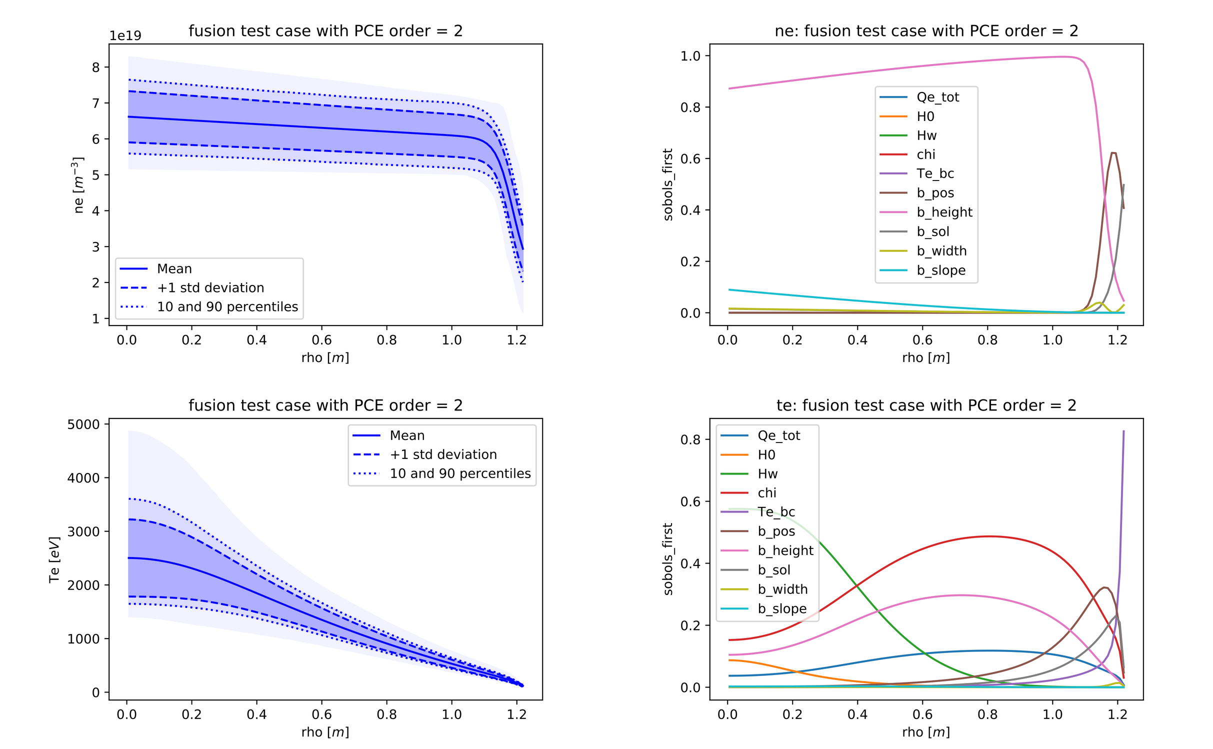

Figure: Upper left, profile of the electron density showing the uncertainty arising from uncertainties in the parameters of the mtanh function (modified tanh function, [13] which refers to [14]) used to parameterize the density profile. Upper right, the profiles of the first Sobol indices for the electron density associated with the various sources of uncertainty (with the main contribution arising from the height of the pedestal in the mtanh specification of the density profile). Lower left, the profile of the electron temperature profile and measures of the uncertainty arising from uncertainties in the heating profile, density profile, transport coefficient and boundary condition. Lower right, the profiles of the first Sobol indices for the electron temperature associated with the various sources of uncertainty.

- a PCE [15] (and references therein) investigation using 10 varying parameters with a polynomial order of 2 (resulting in $(2+1)^{10} = 59049$ samples)
- Sobol results shows that
  - most of the variance at the centre of the plasma comes from the heating profile width
  - at mid-radius the most important source of the variance is the transport coefficient
  - right at the edge most of the variance comes from the boundary condition
- Note that the variance chosen for the input parameters was somewhat arbitrary and so no physics conclusions should be drawn from the results in this section (in addition the model is very simple).

Profiles and uncertainties from the Transport, Turbulence and Equilibrium Workflow with a simplified transport model

- in order to do exploratory studies closer to the ultimate goal, a variant of the Transport, Turbulence and Equilibrium Workflow is used
  - where the turbulence code GEM is replaced by a simpler model GEM0 which gives the output fluxes at the same positions that GEM does, but based on a simple analytic model
- this model is not expected to give quantitatively similar results to GEM, but is designed to be plug-and-play compatible in the workflow
- this means that the runs no longer require 1024 cores, and run in minutes rather than days. Runs usually converge after about 10 cycles around the workflow, taking about 30 seconds

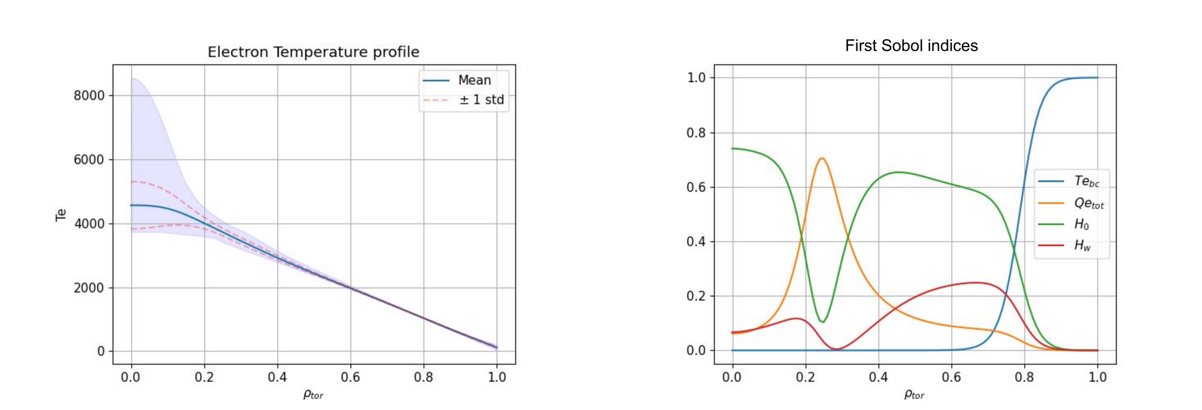

Figure: Left, profile of the electron temperature showing the uncertainty arising from uncertainties in the heating profile and boundary condition. On the right the profiles of the first Sobol indices associated with the boundary condition and the integral, position and width of the heating profile.

- the workflow has been run with 4 uncertain parameters
  - 3 characterizing the width, position and magnitude of the electron heating profile
  - 1 characterising the boundary electron temperature
- normal distribution is used of $\pm 20\%$
- (More computational details can be found in [16].)

Profiles from the Transport, Turbulence and Equilibrium Workflow with a turbulent transport model

The full workflow uses GEM for the calculation of turbulence fluxes. With eight flux tubes, GEM usually runs on 1024 cores. A typical run consists of approximately 1500 cycles of the workflow and takes about 19 hours. Multiple runs are needed before a quasi-steady-state is achieved.

The figure below shows a a comparison between the full workflow simulations and a few "Standard H-Mode" shots

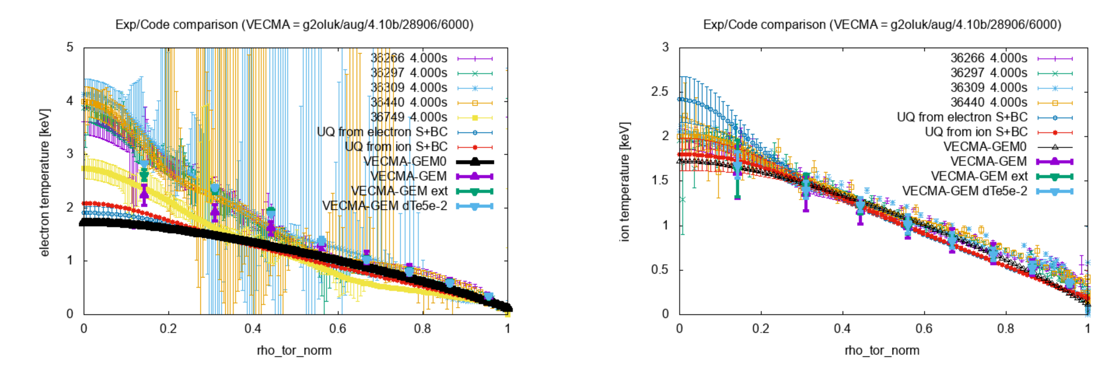

Figure: Profile of the electron temperature (left) and ion temperature (right) for a few ASDEX Upgrade "Standard H-mode" shots, together with simulation results.

- the simulation was set up for an earlier Standard H-Mode shot
  - but not all of the diagnostic signals are available for that shot
  - similar discharges performed later were used as the basis of comparison
- The uncertainty intervals indicated for GEM are those resulting from the turbulence — not those arising from parameter uncertainties
- also plotted are some GEM0 results with uncertainty information

The figure below shows heavily smoothed fluxes from the simulation for the eight flux-tubes, together with a measure of the time-variation, compared to the integral energy source inside each of the flux surface for each flux-tube.

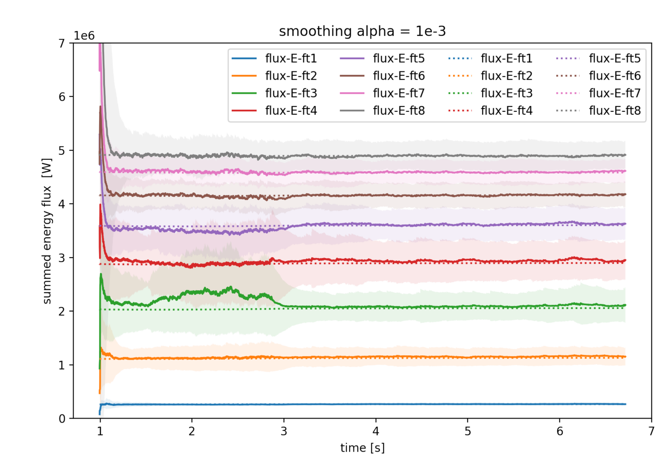

Figure: Comparison of the energy flux from the transport code (solid lines) compared to the integration of energy sources inside the flux surfaces (dotted lines) for the eight flux-tubes.

- one measure of convergence is that the time-average of the calculated fluxes should approach the source integrals — a process that seems to happen relatively quickly
- (More details on the computational aspects of this work can be found in [17] and references therein)

Initial comparison of positive and negative triangularity AUG plasma

- used the workflow from the previous section
- mirrored the last closed flux surface geometric axis of the outermost flux surface
  - this flipped the triangularity from positive to negative
- ran the cases

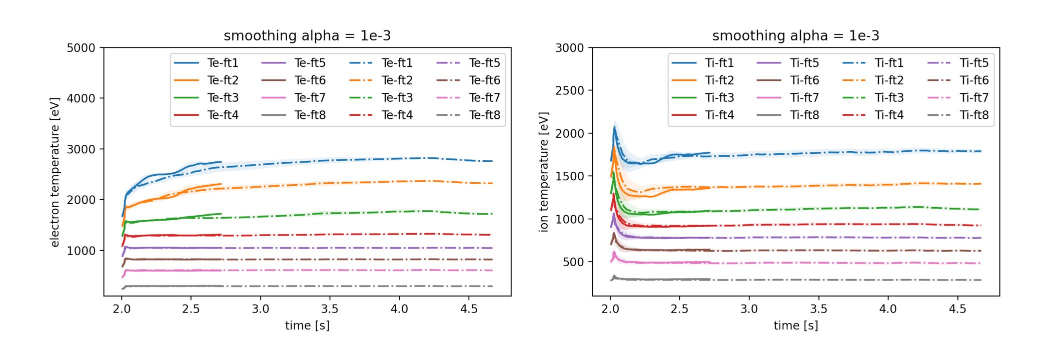

Figure: Time traces of the electron and ion temperatures for the positive (solid line) and negative (dashed-dotted) triangularity cases, for each of the eight flux-tubes.

- the electron and ion temperatures shown above have not yet completely saturated
- the positive triangularity is running with a smaller time-step (chosen automatically to limit the changes of temperatures and temperature gradients)
- but can see that the temperatures are very similar
- below we see that the temperature profiles are disappointingly similar!

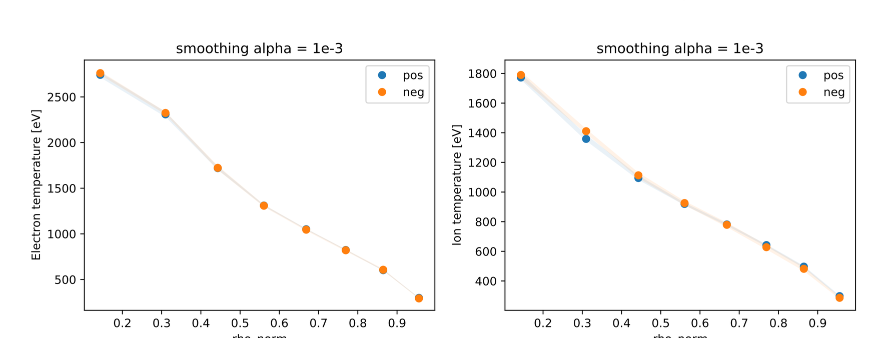

Figure: Electron and ion temperature profiles (with statistical uncertainty arising from the turbulence) for the positive and negative triangularity cases after the same number of cycles around the workflow.

Discussion

A few important aspects when running the Transport, Turbulence and Equilibrium Workflow that have not been covered in this paper but which are covered in other papers are

- ensuring that the transport code does not run for too long before the updated profiles are passed back to the turbulence code; as described in [8] limits are placed on the transport time-step as well as the allowed fractional changes in the value and spatial derivatives of the electron and ion temperature profiles
- deciding when a simulation has reached a quasi-steady-state; this is still a work in progress but, as described in [8], changes in the mean can be compared to an estimate of the standard deviation for time-bins and then criteria found for "stopping".

Outlook

- future challenge will be to bring the pieces of UQ together with the stochastic nature of the turbulence code to examine the feasibility of producing profile predictions incorporating a turbulence code together with uncertainty intervals
- work on this has started but is not yet available for the complete time-dependent workflow
- if this proves to be feasible with the used gyro-fluid turbulence code, the extension to gyro-kinetic turbulence codes will be an obvious next step
  - to do this, access to large compute resources will be needed!

Acknowledgments

This work was supported by the VECMA project, which has received funding from the European Union Horizon 2020 research and innovation programme under grant agreement No 800925. Calculations were performed in the Poznan Supercomputing and Networking Center, the CINECA MARCONI supercomputer, and computers of the Max Planck Computing and Data Facility and the Max Planck Institute for Plasma Physics.

References

[1] COSTER, D. P. et al., IEEE Transactions on Plasma Science **38** (2010) 2085.

[2] LÜTJENS, H. et al., Computer Physics Communications **97** (1996) 219.

[3] SCOTT, B. D., Physics of Plasmas **12** (2005) 102307.

[4] FALCHETTO, G. et al., Nuclear Fusion **54** (2014) 043018, 00005.

[5] BORGDORFF, J. et al., Journal of Computational Science **5** (2014) 719.

[6] IMBEAUX, F. et al., Computer Physics Communications **181** (2010) 987.

[7] HOENEN, O. et al., Designing and running turbulence transport simulations using a distributed multiscale computing approach, in *europhysics conference abstracts*, volume 37D, Espoo, Finland, 2013, 00005.

[8] LUK, O. et al., Computer Physics Communications **239** (2019) 126.

[9] LUK, O. et al., Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences **377** (2019).

[10] VEEN, L. et al., Easing multiscale model design and coupling with MUSCLE 3, in *Computational Science – ICCS 2020*, edited by KRZHIZHANOVSKAYA, V. V. et al., pages 425–438, Cham, 2020, Springer International Publishing.

[11] JANCAUSKAS, V. et al., EasyVVUQ: Verification, validation and uncertainty quantification for HPC simulations, 2021, https://github.com/UCL-CCS/EasyVVUQ.

[12] GUYER, J. et al., Computing in Science & Engineering **11** (2009) 6.

[13] STEFANIKOVA, E. et al., Review of Scientific Instruments **87** (2016) 11E536.

[14] GROEBNER, R. et al., Plasma Physics and Controlled Fusion **40** (1998) 673.

[15] WRIGHT, D. et al., Advanced Theory and Simulations **3** (2020) 1900246.

[16] LAKHLILI, J. et al., Uncertainty quantification for multiscale fusion plasma simulations with VECMA toolkit, in *Computational Science – ICCS 2020*, edited by KRZHIZHANOVSKAYA, V. V. et al., pages 719–730, Cham, 2020, Springer International Publishing.

[17] LUK, O. O. et al., Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences **379** (2021) rsta.2020.0074, 20200074.

* Coresponding author: David.Coster@ipp.mpg.de

This work was supported by the VECMA project, which has received funding from the European Union Horizon 2020 research and innovation programme under grant agreement No 800925. The views and opinions expressed herein do not necessarily reflect those of the European Commission.