Building a turbulence-transport workflow incorporating uncertainty quantification for predicting core profiles in a tokamak plasma

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

Max Planck Institute for Plasma Physics, Boltzmannstr 2, D-85748 Garching, Germany

E-mail: David.Coster@ipp.mpg.de

Received 2 June 2021, revised 12 October 2021
Accepted for publication 2 November 2021
Published 24 November 2021

Abstract
The impact of micro-scale turbulence on the macro-scale plasma profiles in a tokamak is a multi-scale problem (both in space and time) that is treated in this paper by the coupling of turbulence simulations of multiple flux-tubes to a core transport code, together with an equilibrium code. Work on quantifying the uncertainty in the predicted profiles, together with a comparison to experiment is also presented.

Keywords: multi-scale, turbulence-transport workflow, uncertainty quantification, tokamak transport

(Some figures may appear in colour only in the online journal)

1. Introduction
Magnetically confined plasmas hold out the hope for a nearly inexhaustible supply of energy. Amongst the many challenges in reaching the goal of a fusion power plant, is the understanding of energy transport in the core plasma. A very successful way of treating the core plasma numerically is as a set of 1 dimensional time-dependent partial differential equations describing the density, temperature and toroidal rotation as a function of a radial coordinate and time. The basic form of these is as a conservation equation involving a balance between the time rate of change of the quantity of interest (QOI), the radial divergence of the flux of the QOI and the sources of the QOI. The electron temperature, $T_e(\rho, t)$, is expressed in the form of an electron energy equation

$$\frac{3}{2} \frac{\partial}{\partial t} \langle n_e(\rho, t) T_e(\rho, t) \rangle = \nabla_{\rho} \left[ n_e(\rho, t) \chi_e(\rho, t) \nabla_{\rho} (T_e(\rho, t)) \right] + S_e(\rho, t), \quad (1)$$

where $n_e(\rho, t)$ is the electron density; $\chi_e(\rho, t)$ the thermal conductivity; and $S_e(\rho, t)$ the source of electron energy. Here the energy flux has been represented with a diffusive ansatz but, in general, a convective piece might also be needed.

Much of the challenge in treating this problem is in the determination of the transport coefficient, $\chi_e(\rho, t)$. A variety of models are available, but this work determines these transport coefficients by solving a turbulence problem using (in this case) the gyro-fluid flux-tube turbulence code GEM [1], running at eight positions across the plasma.

To address the disparity of time- and space-scales, a multi-scale approach has been implemented in a workflow as depicted in figure 1 which couples a transport code (in this case, the ETS [2]), a fixed boundary equilibrium code (here,
CHEASE [3]), a turbulence code (GEM [1]) and an utility code that converts fluxes to transport coefficients.

The genesis of this approach is described in Falchetto-2014 [4] where the workflow was implemented in the Kepler1 workflow engine [5] resulting in a workflow that allowed for the tight coupling of modular components with the unit of exchange between the codes the appropriate consistent physical object (CPO) [6]. Since the turbulence component required significant computing resources, UNICORE2 was used to schedule this component on a remote supercomputer (see [7, 8] for more information). At that time the supercomputer used was based in Juelich which supported UNICORE. Shortly after the work described in Falchetto-2014, the supercomputer in use switched to HELIOS in Japan, which did not support UNICORE. Since the porting of the complete Kepler environment used was very difficult at that time, a reduced version of the workflow was moved to using the MUSCLE2 coupling framework [9] which supported HPC use better, at the expense of requiring the entire workflow on the HPC. Development of the workflow is described in [10–12] and occurred in a number of European Commission funded projects: MAPPER3: ‘Multiscale Applications on European e-Infrastructures’, ComPat4: ‘Computing Patterns for High Performance Multiscale Computing’ and VECMA5: ‘Verified Exascale Computing for Multiscale Applications’. Work is ongoing to switch to the new MUSCLE36 [13] framework.

Starting from some initial profiles for the temperatures and densities, the equilibrium code calculates new metric coefficients (stored in the equilibrium CPO, green in figure 1). These, together with the temperature and density profiles (stored in the core profiles CPO, red in figure 1), are used by the turbulence code to advance the state of the turbulence simulation for a time window $\Delta t_{\text{turbulence}}$ that allows for some evolution of these fluxes. A running exponential mean is used to calculate flux-tube averaged fluxes (stored in the core transport CPO, light blue in figure 1) which are then converted to transport coefficients (also stored in the core transport CPO, blue in figure 1). These transport coefficients are used by the transport code to evolve the profiles over a time window $\Delta t_{\text{transport}}$. The cycle then repeats until a fixed number of iterations is performed, with the number of iterations determined by the HPC queue time limits. Typically a number (~5) of such jobs need to be performed where the final state of one job is the initial state of the next. Luk and co-authors in [11, 14] discuss some of the issues that arise in this coupling, including limiting the time-step used by the transport code based on ensuring that the temperatures and temperature gradients do not change too much during the transport code time-step (which can lead to a large change in the fluxes predicted by the turbulence code when it sees the new profiles)—this was used in the simulations described in this paper; the alternative is just to limit the time-step allowed for the transport code but this can slow down the whole simulation. Another issue addressed in the papers is on deciding whether the simulation has converged to a quasi steady-state: the inherent nature of turbulence means that a true steady state cannot be achieved but what is sought is a state that, if it were to be further simulated for any length of time, would remain bounded and not diverge from the so-far found quasi steady-state. The approach described in [11] bins the results based on time steps, calculates a mean and standard deviation for each bin, and then compares each bin with previous bins looking for a match based on the means and standard deviations [14]; also looks at binning based on time windows and the maxima and minima in these bins. While these convergence detection methods work, more effort is perhaps needed since early detection of convergence could save a significant amount of computer time, but a failed too early detection could result in a ‘wrong’ solution.

In order to investigate the uncertainty in the predicted profiles, the EasyVVUQ library [15], developed as part of the VECMA project7, is used to incorporate the calculation of uncertainty intervals into the workflow. As identified by the VECMA project, different approaches to implementing uncertainty quantification (UQ) are possible. The easiest to implement is to treat the application as a black-box and then to apply one of the standard UQ techniques to the inputs and outputs of the black-box. In this case the black-box is the entire workflow run to (quasi) steady-state.

Since the full workflow is expensive to run (of order 100,000 core-hours and taking approximately 5 days) with some remaining uncertainty about whether a true quasi steady-state has been achieved, two alternatives have been prepared. The first uses the same workflow, but replaces the turbulence code by a simple proxy (GEM0) that accepts the same inputs and returns the same outputs. For the second, as a development tool, and for tutorial purposes, an even simpler code solving equation (1) directly for the temperature with a prescribed density profile in a cylindrical approximation was developed as described in [16] (see also the VECMA EasyVVUQ tutorial website8).

We thus have three options for exploring UQ:

- **GEM-WF** The full workflow using the GEM turbulence code, which provides a relatively complete set of physics, is however too expensive to explore and develop UQ techniques, but is the ultimate target where we want to apply the UQ techniques.

- **GEM0-WF** The full workflow with the turbulence code replaced by a cheap proxy that does converge to a steady-state in about 30 seconds of CPU time (on one core), and uses the same infrastructure as GEM-WF so that techniques developed with GEM0-WF can later be directly applied to GEM-WF.

- **Fusion-tutorial** A tool that solves a similar problem that runs in about 20 ms, does not depend on the MUSCLE infrastructure required by GEM-WF and GEM0-WF, and that can be used to explore UQ techniques.

---

1. [https://kepler-project.org/](https://kepler-project.org/)
2. [https://unicore.eu](https://unicore.eu)
3. [https://cordis.europa.eu/project/id/261507](https://cordis.europa.eu/project/id/261507)
4. [https://cordis.europa.eu/project/id/671564](https://cordis.europa.eu/project/id/671564)
5. [https://cordis.europa.eu/project/id/800925](https://cordis.europa.eu/project/id/800925)
6. [https://github.com/multiscale/muscle3](https://github.com/multiscale/muscle3)
7. [https://vecma.eu](https://vecma.eu)
8. [https://easyvvuq.readthedocs.io/en/dev/fusion_tutorial.html](https://easyvvuq.readthedocs.io/en/dev/fusion_tutorial.html)
2. Results

UQ results are shown for the simple fusion tutorial example as well as from the transport, turbulence and equilibrium workflow using the GEM0 code in place of the GEM turbulence code. Then physics results are shown of the full transport, turbulence and equilibrium workflow together with a comparison with experimental data. Finally results comparing the usual positive triangularity with an artificially created negative triangularity AUG discharge are shown.

2.1. 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 [17], in about 20 ms allowing for fast investigations of UQ options.

Figure 2 shows a polynomial chaos expansion (PCE) ([18] and references therein) investigation using 10 varying parameters (as specified in table 1) with a polynomial of order 2 (resulting in \((2 + 1)^{10} = 59,049\) samples). (More details of the model can be found in section 3 of [16] or in the VECMA EasyVVUQ tutorial website).

The Sobol indices results shows that most of the variance in the temperature profile 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; and at the edge most of the variance comes from the boundary condition. Note that the impact of the edge temperature boundary condition dies out away from the edge.

Some caution is required in drawing physics conclusions from these results: the variances chosen for the input parameters were somewhat arbitrary and the model is much simpler than the real tokamak case.

But the fusion-tutorial application has allowed for an examination of scaling of the UQ analysis. Table 2 shows how long the actual analysis phase for PCE and stochastic collocation (SC) took for a case with 5 varying quantities, showing that the SC scaling is better than that for PCE.

Table 3 shows SC timings for 10 varying parameters. For order 3, 40 days were required to do the SC analysis. An analysis using a dimension adaptive method with sparse grids (as used in [19]) resulted in a similar accuracy as the order 3 case but in 0.25 days, and requiring only 1245 samples.

2.2. 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, the GEM0-WF 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 meant 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 a fraction of a minute rather than days—runs usually converge after about 10 cycles around the workflow, taking about 30 seconds.

As described in [20], using this variant, the workflow has been run with 4 uncertain parameters: 3 characterizing the width, position and magnitude of the electron heating profile and one characterising the boundary electron temperature. Here a normal distribution is used of \(\pm 20\%\). Results are shown in figure 3. Note that, similarly to the earlier fusion-tutorial case, the impact of the variances in the edge temperature boundary condition dies away as one moves away from the edge. The relative sizes of the contributions to the Sobol indices away from the edge is affected by choices made in the range of the varying parameters (which are different to those used in the fusion-tutorial case).

The main purpose of this model is to test whether lessons learned from the fusion-tutorial UQ analysis can be applied to the workflows. If the GEMO-WF UQ analysis can be done efficiently enough, the ultimate goal would be to apply this to the full GEM-WF case. Since the GEM0 variant requires roughly a factor of 10 million less resources than the GEM variant, it is
important to have this intermediate step for testing ideas. (Note that the GEM0-wf model is about 1500 times more expensive than the fusion-tutorial model.)

2.3. 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 and typically for 1500 cycles around the workflow and takes about 19 hours on the Marconi fusion supercomputer.

![Figure 2.](image)

**Figure 2.** (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. (Right) The profiles of the first Sobol indices for the electron temperature associated with the various sources of uncertainty.

**Table 1.** Quantities that were varied in the simple fusion example. In the third column, an uniform distribution is used to describe the distribution of the varying quantities, with the minimum and maximum specified as the arguments to ‘uniform’.

| Quantity | Description | Distribution |
|----------|-------------|--------------|
| $Q_{e_{tot}}$ | Total heating power (W) | Uniform(1.8×10^6, 2.2×10^6) |
| $H_0$ | Position of the Gaussian describing the heating profile (in $\rho_{i_0}$) | Uniform(0.0, 0.2) |
| $H_w$ | Width of the Gaussian describing the heating profile (in $\rho_{i_0}$) | Uniform(0.1, 0.5) |
| $\chi$ | Value of the thermal diffusivity (m² s⁻¹) | Uniform(0.8, 1.2) |
| $T_{e_{bc}}$ | Boundary condition for the temperature (eV) | Uniform(80.0, 120.0) |
| $b_{pos}$ | Position of the density pedestal (in $\rho_{i_0}$) | Uniform(0.95, 0.99) |
| $b_{height}$ | Height of the density pedestal (m⁻³) | Uniform(5×10⁹, 7×10⁹) |
| $b_{sol}$ | Value of the density in the SOL (m⁻³) | Uniform(1×10⁹, 3×10⁹) |
| $b_{width}$ | Width of the density pedestal (as a fraction in $\rho_{i_0}$) | Uniform(0.015, 0.025) |
| $b_{slope}$ | Slope of the density in the core (-) | Uniform(0.005, 0.020) |

**Table 2.** Comparison of time (in seconds) needed to perform PCE and SC analysis as a function of the polynomial order, for a case with five varying parameters.

| Order | Samples | PCE time | SC time |
|-------|---------|----------|---------|
| 1     | 32      | 1.75     | 0.41    |
| 2     | 243     | 3.13     | 2.76    |
| 3     | 1024    | 10.12    | 12.76   |
| 4     | 3125    | 47.77    | 45.95   |
| 5     | 7776    | 256.18   | 162.27  |
| 6     | 16 807  | 1242.13  | 541.31  |

**Table 3.** Comparison of time (in seconds) needed to perform SC analysis as a function of the polynomial order, for a case with ten varying parameters.

| Order | Samples | SC time |
|-------|---------|---------|
| 1     | 1024    | 14.97   |
| 2     | 59 049  | 10 180.64 |
| 3     | 10 48 576 | 3476 360.00 |

Multiple runs (≈5) are needed (with a new run starting from the output of the previous run) before a quasi-steady-state is achieved.

Figure 4 shows a comparison between the simulations and a few ASDEX Upgrade ‘standard H-mode’ shots. The initial simulation was set up for an earlier standard H-mode, but not all of the diagnostic signals are available for that shot; similar discharges (‘standard H-modes’) performed later were used as the basis of comparison. These shots were all 1 MA, −2.45 T, line average density of $8.5 \times 10^{19}$ m⁻³ and with heating between 8.2 and 9.5 MW. The heating sources were treated as Gaussian profiles so that sensitivity analysis could be done on the integral, position and width. The density profile was taken from the experiment and the
plasma was assumed to be pure deuterium (i.e. no impurities). The experimental uncertainties are given by the integrated data analysis used to combine different diagnostics. The uncertainty intervals indicated for GEM are those resulting from the turbulence—not those arising from parameter uncertainties—and are the mean and standard deviation calculated on a time window at the end of the simulation. The standard deviations are small because of the limits on changes in temperatures and their gradients (discussed earlier, with limits of 10% and 5% for the values and gradients, respectively).

One of the challenges with these complicated workflows is ensuring that the different components agree on radial coordinates, as well as sharing the same definition of fluxes and transport coefficients. As an aspect of verification, the fluxes can be compared to the integral of sources inside the position at which the flux is calculated. Figure 5 shows heavily (time) 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. 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 (for comparison, the energy confinement time is 0.068 s). These simulation results were taken from a long run that was used to try to calibrate various convergence criteria (see [11] and are almost certainly converged.

More details on the computational aspects of this work can be found in [14] and references therein, which also looks at applying quantitative measures for the validation of the computational results against the experimental results.

2.4. Initial comparison of positive and negative triangularity AUG plasma

In order to investigate the effect of negative triangularity on plasma confinement, the outer boundary of the plasma used in section 2.3 was mirrored around the geometric axis of the outermost flux surface, changing the sign of the upper and lower triangularity. Both cases have been started and, while not yet converged to a steady-state, initial observations are possible. Figure 6 shows the evolution of the electron and ion temperatures with time. Despite having performed the same number of cycles around the workflow, the elapsed physics time for the positive triangularity case is less than that for the negative triangularity case because of the time-step limiting in the transport code to prevent too large changes in profiles and gradients per time-step. Taking data from the end of the simulations and comparing the profiles, figure 7 shows very similar profiles for the two cases. It is to be noted, though, that the boundary conditions (applied at rho_nrm = 1) are the same for the two cases and so any impact of triangularity on the boundary temperatures would not be seen in these simulations.

3. Discussion

A hierarchy of models has been developed: the slowest but most accurate model takes about 100 000 core hours for a single (quasi) steady state solution; the intermediate model uses the same infrastructure as the expensive model, but is about 10 000 times faster and requires one core rather than 1024; the fastest model is another 1500 times faster and is much more portable, requiring none of the proprietary physics code or the coupling framework. This hierarchy allows one to look at different aspects of the problem at the same time—the accuracy of the full physics model as well as the scaling of the UQ methods to a large number of cases (100 000 for the most extreme case looked at so far).

The full workflow using the turbulence code (GEM-WF) shows a satisfactory match to the experiment suggesting that it would be interesting to explore the impact of varying the
heating profile and boundary conditions, but it is still too expensive to do this (except as a heroic one-off).

The full workflow using a simple proxy for the turbulence code (GEM0-WF) has shown that it is possible to do UQ on the workflow, treating it as a black-box, but that a significant number of runs of the code would be needed (hundreds to thousands).

The tutorial fusion application (fusion-tutorial) has allowed UQ aspects to be explored (PCE versus SC, the use of sparse dimension adaptive methods, etc) which will inform the application of UQ to the two workflow cases.

The investigation of positive and negative triangularity did not show a large difference: if differences are due to turbulence other than ion temperature gradient (such as trapped electron mode), then this would not be captured by the gyro-fluid turbulence code; if the differences are due to edge pedestal changes, then this too would be missed in these simulations since the boundary conditions were the same.
4. Outlook

The 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. The work on adaptive sparse grid UQ has indicated that the number of samples required can be substantially reduced, but it would still be too expensive to apply this directly to the GEM-WF. Work has started on exploring the application of UQ to the components of the workflow ([20] has done this for the GEM component and calculated the Sobol indices for the electron and ion heat fluxes as functions of the temperatures and their gradients), and the next step would be to explore propagating distribution functions around the workflow, rather than just the current profiles. Another approach under investigation is the replacement of the expensive model by a faster surrogate based on results from the expensive model.

If this proves to be feasible with the used gyro-fluid turbulence code, the extension to gyro-kinetic turbulence codes will be a conceptually easy subsequent step. However, 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.

ORCID iDs

D.P. Coster https://orcid.org/0000-0002-2470-9706
O. Hoenen https://orcid.org/0000-0001-6953-8800
J. Lakhlili https://orcid.org/0000-0003-2900-1026

Figure 6. 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.

Figure 7. 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.
References

[1] Scott B.D. 2005 Phys. Plasmas 12 102307
[2] Coster D.P., Basiuk V., Pereverzev G., Kalupin D., Zagorski R., Stankiewicz R., Huynh P. and Imbeaux F. 2010 IEEE Trans. Plasma Sci. 38 2085
[3] Lütjens H., Bondeson A. and Sauter O. 1996 Comput. Phys. Commun. 97 219
[4] Falchetto G.L. et al 2014 Nucl. Fusion 54 043018
[5] Altintas I., Berkley C., Jaeger E., Jones M., Ludascher B. and Mock S. 2004 Kepler: an extensible system for design and execution of scientific workflows 2004 Proc. 16th Int. Conf. Scientific and Statistical Database Management (23 June 2004 Santorini, Greece) pp 423–4 (https://ieeexplore.ieee.org/document/1311241)
[6] Imbeaux F. et al 2010 Comput. Phys. Commun. 181 987
[7] Strand P. et al 2010 A European infrastructure for fusion simulations 2010 18th Euromicro Conf. Parallel, Distributed and Network-Based Processing (Pisa, Italy, 17–19 February 2010) pp 460–7 (https://ieeexplore.ieee.org/document/5452432)
[8] Frauel Y., Coster D. and Guillerminet B., Imbeaux F., Jackson A., Konz C., Owsiak M., Plociennik M., Scott B. and Strand P. 2012 Easy use of high performance computers for fusion simulations Fusion Eng. Des. 87 2057–62
[9] Borgdorff J., Mamonski M., Bosak B., Kurowski K., Ben Belgacem M., Chopard B., Groen D., Coveyey P.V. and Hoekstra A.G. 2014 J. Comput. Sci. 5 719
[10] Hoenen O., Fazendeiro L., Scott B.D., Borgdorff J., Hoekstra A.G. and Strand P. 2013 Designing and running turbulence transport simulations using a distributed multiscale computing approach Europhysics Conference Abstracts vol 37D (1 – 5 July 2013, Espoo, Finland) p 00005 (http://ocs.ciemat.es/EPS2013PAP/html/)
[11] Luk O.O., Hoenen O., Bottino A., Scott B.D. and Coster D.P. 2019 Comput. Phys. Commun. 239 126
[12] Luk O.O. 2019 Phil. Trans. R. Soc. A 377 20180152
[13] Veen L.E. and Hoekstra A.G. 2020 Easing multiscale model design and coupling with MUSCLE 3 Computational Science–ICCS 2020 3 – 5 June 2020 Amsterdam, The Netherlands ed V.V. Krzhizhanovskaya, G. Závodszky, M.H. Lees, J.J. Dongarra, P.M.A. Sloot, S. Brissos and J. Teixeira (Cham: Springer) pp 425–38
[14] Luk O.O., Lakhili J., Hoenen O., von Toussaint U., Scott B.D. and Coster D.P. 2021 Phil. Trans. R. Soc. A 379 20200074
[15] Jancauskas V., Lakhili J., Richardson R. and Wright D. 2021 EasyVVUQ: verification, validation and uncertainty quantification for HPC simulations (https://github.com/UCL-CCS/EasyVVUQ)
[16] Sulaimanova D. et al 2021 J. Comput. Sci. 53 101402
[17] Guyer J.E., Wheeler D. and Warren J.A. 2009 Comput. Sci. Eng. 11 6
[18] Wright D.W. et al 2020 Adv. Theory Simul. 3 1900246
[19] Edeling W. et al 2021 Nat. Comput. Sci. 1 128
[20] Lakhili J., Hoenen O., Luk O. O. and Coster D. P. 2020 Uncertainty quantification for multiscale fusion plasma simulations with VECMA toolkit Computational Science–ICCS 2020 3 – 5 June 2020 Amsterdam ed V.V. Krzhizhanovskaya, G. Závodszky, M.H. Lees, J.J. Dongarra, P.M.A. Sloot, S. Brissos and J. Teixeira (Cham: Springer) pp 719–30