Excalibur-Neptune report 2047356-TN-15-2

Task 2.5 Performance considerations for non-intrusive uncertainty quantification and the influence of the boundary

Ben Dudson, Peter Hill, Ed Higgins, David Dickinson, and Steven Wright

University of York

David Moxey

KCL

October 28, 2022

Contents

1 Executive summary

In this report we summarise some performance considerations relevant to the use of non-intrusive uncertainty quantification (UQ) through tools such as the *easyvvuq* package [1, 2]. Minimising the number of simulations and effective use of workflow management tools are often the primary route to improving the time to solution and freeing up researcher time. The number of moments required for converged uncertainty measures near the boundary is also explored for a sample test case from the SD1D [3] test cases [4]. It is demonstrated that, depending on the boundary conditions selected and the uncertain inputs, convergence can require between 2^{nd} and 5^{th} order projections.

2 Introduction

In task 83-2.4 non-intrusive uncertainty quantification (UQ) for SD1D [3], a BOUT++ [5] based 1D fluid system, was explored. The corresponding report [6] introduced the approach to implementing a UQ workflow with *easyvvuq*, demonstrated some of the different possible workflows and made brief mention of possible performance considerations. Here we seek to build on this previous work in two ways. Firstly, in section 3 we will expand on some of the considerations relevant to optimising the time to solution (TTS) in a UQ workflow. The order of the UQ sampler used can determine the number of simulations required, potentially impacting TTS and in section 4 we will explore the influence of boundary conditions on the order of UQ sampler required for convergence.

3 Performance considerations

Approaches to UQ can be classified as either intrusive or non-intrusive. Intrusive approaches require the underlying computational model to be adapted in order to effectively run an ensemble of cases simultaneously, increasing the problem size of a single simulation. Non-intrusive approaches on the other hand require a large number of "standard" simulations to be performed for a range of inputs. These approaches have different performance considerations/characterisics. For example, suppose UQ increases the amount of computation required by a factor N compared to a single simulation which returns a result in time T. Here intrusive UQ increases the problem size by N and non-intrusive UQ requires N separate simulations to be performed. If the weak scaling of the code is not ideal, say due to collective communication, then simply increasing the number of processing units by N will not allow the intrusive approach to return a result in the same time T as previously and one must either wait NT to get the result or accept a degree of increased inefficiency. Conversely, the non-intrusive approach can split the problem into N trivially parallelisable jobs[1]. Provided there is sufficient computational resource to runs these jobs simulataneously it is possible to return a result in T. However, in the extreme limit where it is only possible to run one simulation at a time the non-intrusive approach may actually return a result slower than the intrusive approach run on the same number of processors, due to repetition of overheads. For example, suppose the simulation generates a large mesh as part of the initialisation of the simulation. Without code modification, time may then be wasted in repeatedly generating the same mesh within each simulation rather than reusing this. More generally, whilst the distribution of work into separate simulations used in non-intrusive methods can make the additional work trivially parallelisable this may typically make less efficient use of the available resources than intrusive methods due to the isolation of each simulation and hence the duplication of common operations. Furthermore, the isolation of separate simulations means it is not possible to take advantage of the expected similarity of the different cases. For example, suppose the simulation involves the iterative solution of a large sparse matrix problem for each case within the ensemble. In an intrusive approach one may envisage using the solution of one sample to provide a good first guess for subsequent samples, accelerating future solves and limiting the increase in required computation to less than the factor N. Such an approach is not trivially possible in non-intrusive UQ and certainly not without introducing some serialisation of the simulations. Where the balance lies strongly depends on both the simulation problem at hand as well as the properties of the computational systems available. Within the ExCALIBUR-Neptune project non-intrusive approaches are currently preferred and this report will focus on aspects relevant for these cases.

With non-intrusive UQ requiring a number of repeated simulation executions with different parameters, it is clearly important to ensure that individual simulations are setup as efficiently as possible. In doing so care must be taken, however, to consider the potential for variation across the runs required. For example, when manually running a single simulation one may be able to identify the resolutions required for the result to converge to a given degree. The required resolution may, however, vary with the inputs. As such, operating on the edge of satisfactory resolution for the nominal case may lead to severe under-resolution in other parts of the sample space. It may therefore be appropriate to build in a margin of safety to the resolutions used. This however, may degrade the overall efficiency of the problem by over-resolving some cases and using more resource than strictly necessary. Such problems may be avoided by the use of adapative algorithms which can adapt the order and resolution in order to keep errors to some fixed level. A common example which can be observed even in fixed-grid finite difference codes, such as BOUT++, is from time step adaptivity. Here the limiting time step may be strongly dependent on the uncertain input parameters and as such it is possible to find highly variable wall time requirements for simulations within a UQ study. This can have different consequences depending on the approach to running the independent simula-

[1] This is only true for non-adaptive sampling, which may not be the optimal approach for problems with a large number of uncertain inputs.

tions. Where it is possible to define and launch all simulations simultaneously, variable run time has the effect of slowing the TTS to that of the longest simulation. Where runs are executed sequentially the impact of variable run time may be more limited if one expects there to be similar numbers of simulations which go faster and slower than the nominal simulation. In the common case where a subset of the total number of simulations are executed in parallel the impact on TTS is less clear and depends upon the distribution of job run times across the jobs and the order of execution. Where it is possible to anticipate the impact on wall time requirements as a function of input parameters, it may be possible to attempt to load balance simulations across the available resource, such as allocating more processors to those problems with a smaller time step.

When running a single simulation the TTS can often be minimised by selecting an optimal number of processors. The optimal choice depends upon both the scaling efficiency of the simulation code as well as the impact on queue time. Queue time can become a significant factor in determining TTS in certain scenarios. This will depend the policies of the queue manager that have been set and will often become more significant when requiring a substantial fraction of the resource available. The impact of queue time can be compounded when attempting to run a large number of simulations and the user has several decisions to make and options to attempt to optimise TTS. For example, some machines may limit the number of separate jobs which can be submitted/running at any one time. In such instances it may be beneficial to bundle a number of simulations into single jobscripts. This may then increase the amount of time it takes for each job to start running, however, due to the increase in the scale of resource required. Another motivation for bundling jobs may be where the job manager policies are such that priority is given to larger jobs. Conversely, a large number of small jobs may be easier for the queue manager to schedule in the gaps around larger jobs (i.e. using small jobs for backfill). The optimal approach can therefore be highly sensitive to the specific details of the system at hand combined with the resource requirements and performance properties of the simulation code. Fortunately, these concerns are also not unique to any one code and there exist tools to assist the user in constructing appropriate workflows and approaches to execution such as FabSim3 [7] and QCG-PilotJob [8] integrated within the VECMA toolkit [9].

Of course, whilst optimising individual runs and the job strategy can be helpful, ultimately the number of simulations required plays a large part in determining the TTS and this is determined by the sampling strategy selected and the order of the sampler. For example, with a polynomial-chaos-expansion (PCE) sampler the number of simulations is given by $(O+1)^U$ where O is the order of the projection and U is the number of uncertain parameters. Minimising O therefore can significantly reduce the number of simulations required, however it may not be possible to determine the order required for convergence. Adaptive approaches, such as demonstrated with stochastic-collocation (SC) in task 2.4 [6], offer the potential to reduce the number of simulations further by selectively refining the sampling of parameter space in the inputs which provide the most information. Furthermore, these offer the ability to terminate a study early if results appear to be converged. These gains come at the cost of limiting the parallelisability of the simulations – there are fewer simulations to be performed at a time, introducing order dependencies. In some situations this may be of limited consequence, e.g. where it is only possible to run a small number of simulations at a time. It is important to note that developing sampling strategies with improved performance properties is still an area of active research, see for example [10].

4 The influence of the boundary

SD1D evolves the following plasma equations for plasma density n, pressure p and momentum $m_i n V_{||}$

$$\begin{aligned}
\frac{\partial n}{\partial t} &= -\nabla \cdot \left(\mathbf{b} V_{||} n\right) + S_n - S \\
\frac{\partial}{\partial t}\left(\frac{3}{2}p\right) &= -\nabla \cdot \mathbf{q} + V_{||}\partial_{||}p + S_p - E - R \\
\frac{\partial}{\partial t}\left(m_i n V_{||}\right) &= -\nabla \cdot \left(m_i n V_{||} \mathbf{b} V_{||}\right) - \partial_{||}p - F \\
j_{||} &= 0 \\
T_i &= T_e = \frac{1}{2}\frac{p}{en} \\
\mathbf{q} &= \frac{5}{2}p\mathbf{b}V_{||} - \kappa_{||e}\partial_{||}T_e
\end{aligned}$$

where the heat conduction coefficient $\kappa_{||e}$ is a nonlinear function of temperature T_e:

$$\kappa_{||e} = \kappa_0 T_e^{5/2}$$

and is given by the Spitzer conduction

$$\kappa_{||e} = 3.2 \frac{ne^2 T \tau_e}{m_e} \simeq 3.1 \times 10^4 \frac{T^{5/2}}{\ln \Lambda}$$

A similar set of equations can be included to describe neutrals [11].

In the 1D system studied by SD1D there are boundary conditions applied at either end of the domain. The left most boundary corresponds to the "upstream" representing where plasma enters the scrape-off-layer (SOL) from the core at the outboard mid-plane whilst the right most boundary corresponds to the "target" where the field line intersects the material surface and plasma can interact with the wall. In the presence of material surfaces plasmas form a "sheath" just in front of the surface which is a small region over which quasi-neutrality no longer holds and which holds a negative potential, accelerating ions and reflecting electrons from the material surface such that the net flux of ions and electrons to the wall is equal. In SD1D sheath boundary conditions are applied at the target and zero flow boundaries are applied upstream. It is also necessary to provide a plasma source upstream to compensate for the flow to the target. There are multiple options with which once can specify these sources, including the use of a feedback controller to keep the upstream density fixed. One expects the neutral density to be largest close to the target and hence for the plasma-neutral interactions to dominate here. There are many parameters within the model with which one can tune the behaviour near to the boundary such as the proportion of ions interacting with the wall which return to the system, known as the recycling fraction, the energy of neutrals coming from the wall, the sheath heat transmission coefficient etc. Imposing boundary conditions can constrain the solution and the choice of boundary treatment can sometimes lead to signifcant differences in the simulation. It can therefore be interesting to consider how the uncertainty due to boundary conditions behaves.

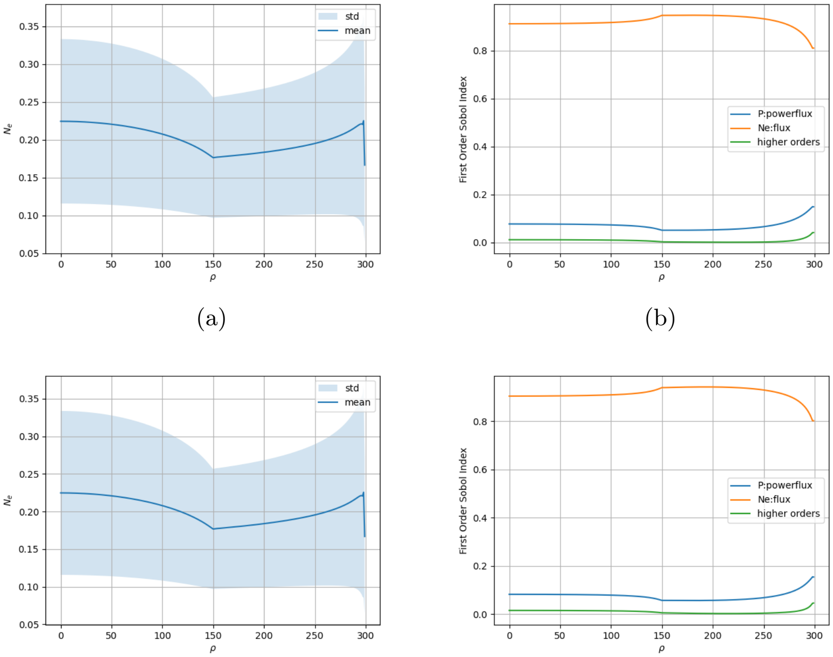
To explore this here we consider test case-04 of SD1D as used in task 2.4 [6]. This combines the full plasma system given above with that for the neutral gas and fixed upsteam sources, and can be captured by Hermes-3. Here we consider two scenarios. Firstly, we will treat the upstream density and pressure sources as uncertain. Secondly, we will consider uncertainty in the recycling fraction `frecycle` and the fraction of recycled ions redistributed over the domain, `fredistribute`[2].

For both of these scenarios we will also vary the density and pressure boundary conditions between free (constant gradient), zero gradient and constant flux. This provides six different cases to consider and for each of these we use PCE samples for all orders from one to six to explore the convergence of the results. We note that first order studies cannot capture higher order interactions in the Sobol indices so it is generally expected that at least second order will be required in most cases.

We begin by considering the uncertain upstream source cases. Figure 1 shows the output for 2^{nd} and 3^{rd} order studies with free boundary at the target. One can see good agreement in the uncertainty and the Sobol indices. It is interesting to note that the 5^{th} order results, shown in figure 2, show some deviation from these converged results. Despite this, the 6^{th} order study is also near identical to the 2^{nd} and 3^{rd} order, suggesting good convergence by 2^{nd} order. The zero gradient study shows near identical behaviour so results are not presented here. The constant flux boundary condition study is presented in figure 3. Firstly, one may clearly note how the structure of the solution has changed in this case. From the plot of the mean density it is not immediately apparent that there is any variation between the 2^{nd}, 4^{th} and 6^{th} order results. On close inspection, one can see that the uncertainty near the target broadens slightly as the order increases. This is a small relative change, but does indicate the results may not be fully converged by this point. Changing attention to the Sobol indices, the variation between the orders becomes more apparent. Whilst there is only a small change between 2^{nd} and 4^{th} order, there is a more significant change in the 6^{th} order result. In particular, one can see that the `P:powerflux` index has reduced somewhat across most of the domain, and particularly in the outer half of the domain. This is matched by a corresponding increase in the higher order indices, with the `Ne:flux` remaining relatively unchanged. This suggests that there are inter-dependencies that are relatively high order in nature. These cases serve to demonstrate some of the benefits and challenges for adaptive

[2] See section 6.2 of [11] for more details.

methods, such as those discussed in task 2.4. The free boundary results were well converged by 3^{rd} order whilst going to 6^{th} order required substantially more simulations (16 vs 49 in this case). A high order adaptive method may be able to stop early when convergence is observed. However, the case with constant flux boundaries demonstrates a challenge to this – results can appear broadly converged at low order but new variation can arise when continuing to yet higher order. One needs to take care in the stopping condition for adaptive approaches.



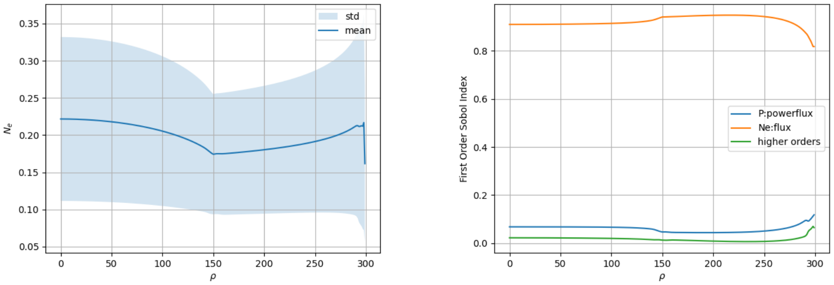
(a)

(b)

(c)

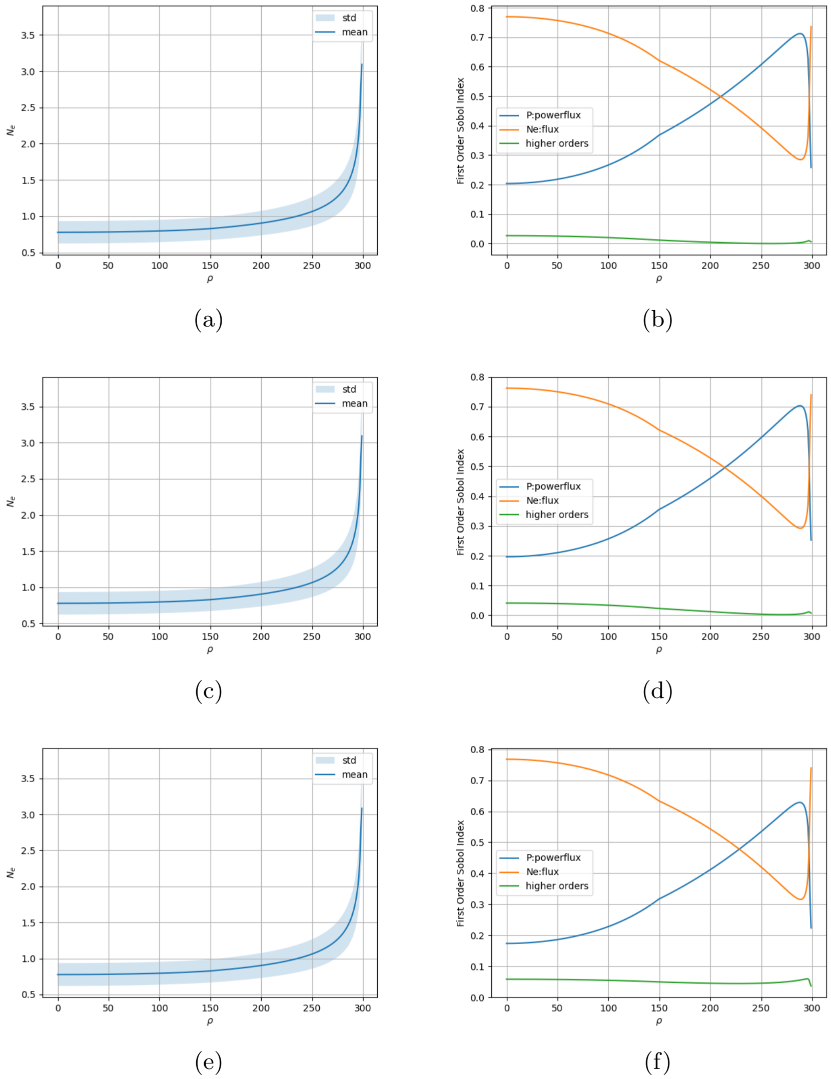
(d)

Figure 1: Plots of the mean normalised plasma density, N_e, as a function of parallel arc length along with the mean ± the standard deviation for a 2^{nd} (a) and 3^{rd} (c) order PCE sampler and the corresponding first order Sobol indices (b/d). Free boundary.



(a) (b)

Figure 2: Plots of the mean normalised plasma density, N_e, as a function of parallel arc length along with the mean ± the standard deviation (a) and the corresponding first order Sobol indices (b) for a 5^{th} order PCE sampler. Free boundary.



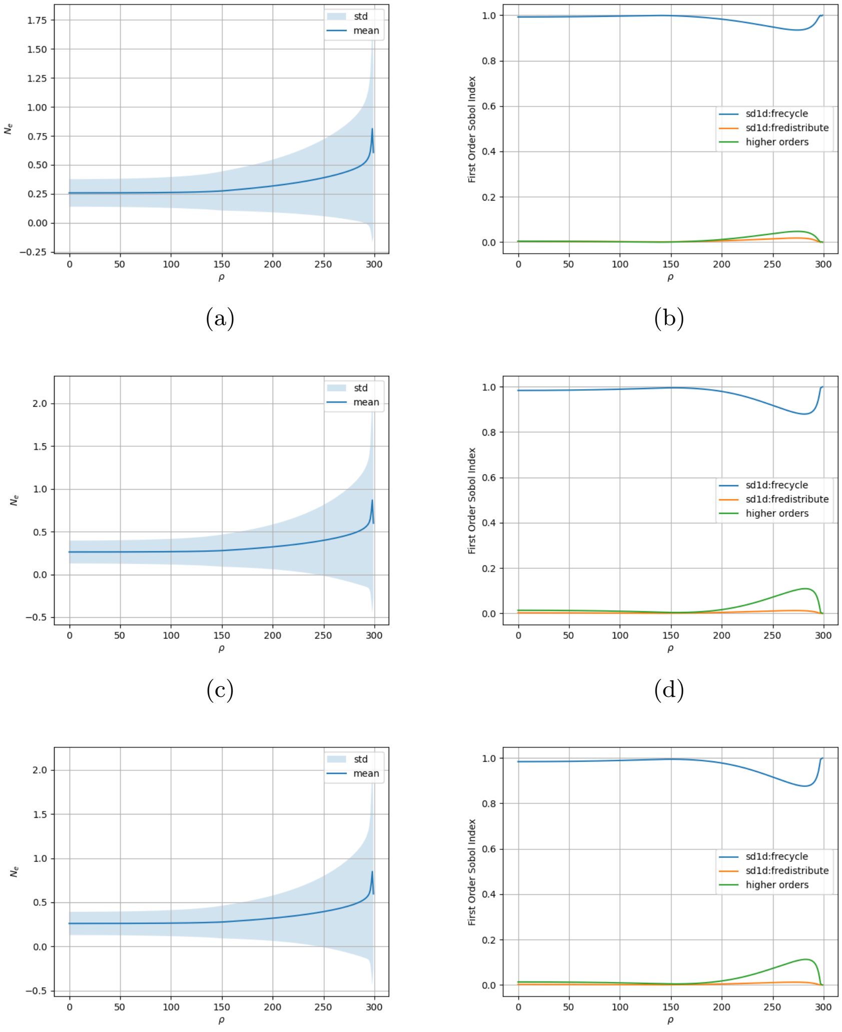
(a) (b)

(c) (d)

(e) (f)

Figure 3: Plots of the mean normalised plasma density, N_e, as a function of parallel arc length along with the mean ± the standard deviation for a 2^{nd} (a), 4^{th} (c) and 6^{th} (e) order PCE sampler and the corresponding first order Sobol indices (b/d/f). Constant flux boundary.

We now turn attention to the case where we treat the recycling fraction, `frecycle`, and the redistribution, `fredistribute`, as uncertain inputs. Here we adopt a uniform distribution with minimum of 0 and maximum of 1 for both parameters. Figure 4 shows the 2^{nd}, 4^{th} and 6^{th} order results for the free boundary case. This demonstrates reasonable convergence at 4^{th} order and the 5^{th} and 6^{th} order results are near identical. Again, the zero gradient results give very similar behaviour so are ommitted. The constant flux results are shown in figure 5. Although hard to see, this shows that there is variation in the uncertainty on the measured `Ne` between 4^{th} and 6^{th} order, particular near the target. This is, again, more visible in the Sobol indices, especially in the higher order and `sd1d:fredistribute` parameters. These results shown that at least 5^{th} order is required for convergence here. It is important to note that the confidence intervals shown here include unphysical negative densities which cannot have been found in any simulation. In such situations one may wish to work with logarithmic variables.

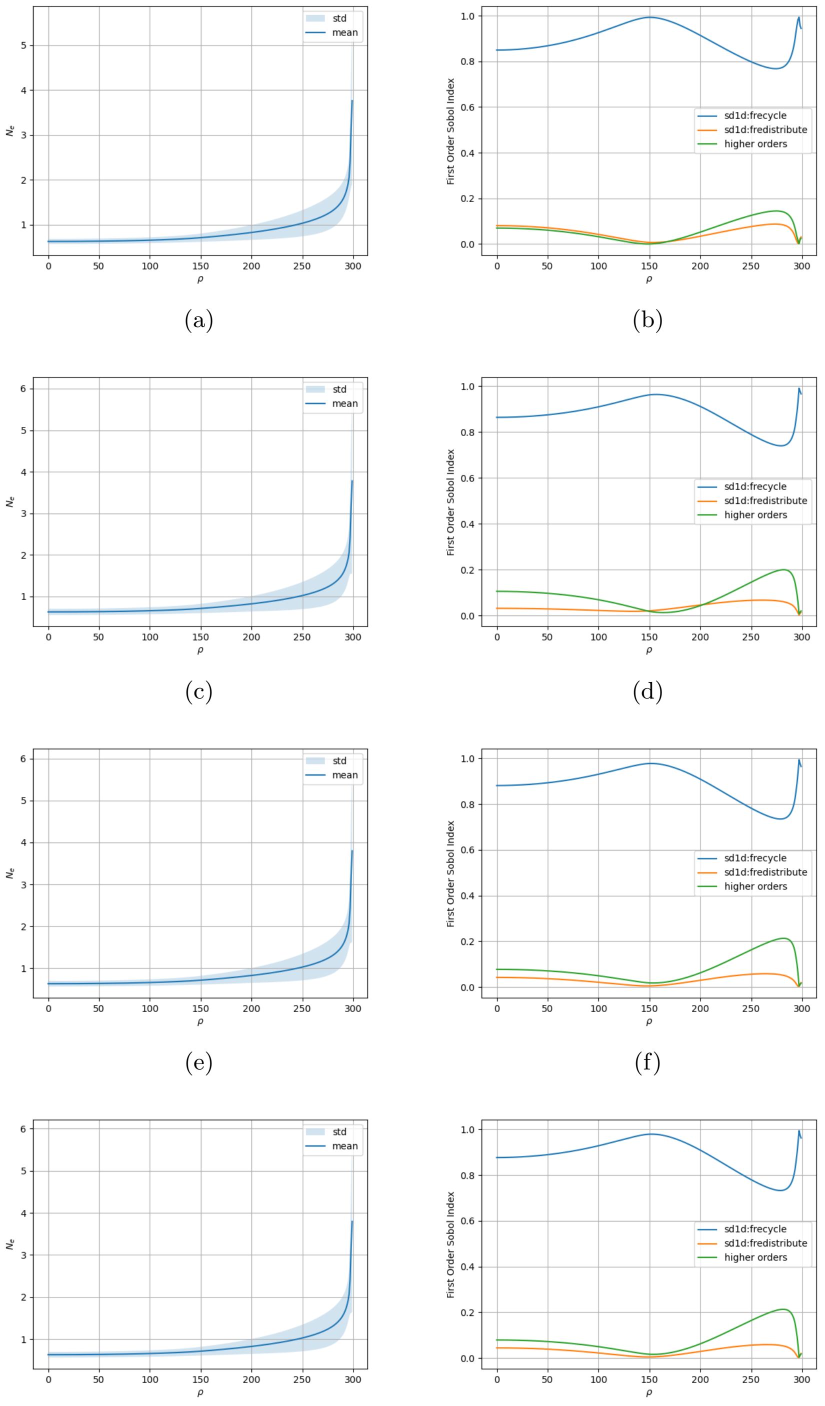


(a) (b)

(c) (d)

(e) (f)

Figure 4: Plots of the mean normalised plasma density, N_e, as a function of parallel arc length along with the mean ± the standard deviation for a 2^{nd} (a), 4^{th} (c) and 6^{th} (e) order PCE sampler and the corresponding first order Sobol indices (b/d/f). Free boundary.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Figure 5: Plots of the mean normalised plasma density, N_e, as a function of parallel arc length along with the mean ± the standard deviation for a 2^{nd} (a), 4^{th} (c), 5^{th} (e) and 6^{th} (g) order PCE sampler and the corresponding first order Sobol indices (b/d/f/h). Constant flux boundary.

5 Summary

In this report we have discussed some of the relevant performance considerations when performing UQ studies with a toolkit such as *easyvvuq*. Whilst there are many aspects that can influence the time to solution, two of the most significant aspects which impact the efficiency of such studies are minimising the number of simulations required and effective use of tools to help automate and manage complex workflows. The latter frees the user from having to micro-manage the simulations and analysis, allowing them to focus on other aspects of the work. In a crude sense, the number of simulations to be performed is controlled by the order of the projection and the number of uncertain inputs. However, adaptive schemes can optimise the approach to sampling such that simulations are targeted to refine in the directions returning the most useful information. This, combined with a method to test for convergence, can allow for a significant reduction in the number of simulations required. We demonstrated how Sobol indices and the population distribution estimates required different projection orders to converge depending on the uncertain parameters and boundary conditions used, ranging from 2^{nd} to 6^{th} order. This highlights one possible advantage of adaptive schemes, but also gave a demonstration of challenges to be faced in testing for convergence where Sobol indices were observed to roughly converge by 4^{th} order, only for significant change to be found at 5^{th} and 6^{th} order. For the simple test case considered here it was possible to perform high order PCE studies in under ten minutes using a single node of Archer2. More expensive systems would likely benefit from the use of FabSim3 [7].

6 Acknowledgements

We gratefully acknowledge compute time provided on CIRRUS and Archer2 through the Excalibur SEAVEA project [12].

7 References

[1] Wright, David W. and Richardson, Robin A. and Edeling, Wouter and Lakhlili, Jalal and Sinclair, Robert C. and Jancauskas, Vytautas and Suleimenova, Diana and Bosak, Bartosz and Kulczewski, Michal and Piontek, Tomasz and Kopta, Piotr and Chirca, Irina and Arabnejad, Hamid and Luk, Onnie O. and Hoenen, Olivier and Weglarz, Jan and Crommelin, Daan and Groen, Derek and Coveney, Peter V. Building confidence in simulation: Applications of easyvvuq. *Advanced Theory and Simulations*, 3(8):1900246, 2020, doi:https://doi.org/10.1002/adts.201900246.

[2] Richardson, Robin A. and Wright, David W. and Edeling, Wouter and Jancauskas, Vytautas and Lakhlili, Jalal and Coveney, Peter V. EasyVVUQ: A library for verification, validation and uncertainty quantification in high performance computing. *Journal of Open Research Software*, 8(1):1–8, 2020, doi:10.5334/JORS.303.

[3] Benjamin Dudson. SD1D: Sol and Divertor in 1D. `https://github.com/boutproject/SD1D`.

[4] Benjamin Dudson, Peter Hill, Ed Higgins, David Dickinson, Steven Wright and David Moxey. 1D fluid model tests. `https://github.com/ExCALIBUR-NEPTUNE/Documents/blob/main/reports/2047356/TN-04.pdf`.

[5] Benjamin Daniel Dudson, Peter Alec Hill, David Dickinson, Joseph Parker, Adam Dempsey, Andrew Allen, Arka Bokshi, Brendan Shanahan, Brett Friedman, Chenhao Ma, David Bold, Dmitry Meyerson, Eric Grinaker, George Breyiannis, Hasan Muhammed, Haruki Seto, Hong Zhang, Ilon Joseph, Jarrod Leddy, Jed Brown, Jens Madsen, John Omotani, Joshua Sauppe, Kevin Savage, Licheng Wang, Luke Easy, Marta Estarellas, Matt Thomas, Maxim Umansky, Michael Løiten, Minwoo Kim, M Leconte, Nicholas Walkden, Olivier Izacard, Pengwei Xi, Peter Naylor, Fabio Riva, Sanat Tiwari, Sean Farley, Simon Myers, Tianyang Xia, Tongnyeol Rhee, Xiang Liu, Xueqiao Xu, Zhanhui Wang, Sajidah Ahmed, and Toby James. BOUT++, 3 2022.

[6] Benjamin Dudson, Peter Hill, Ed Higgins, David Dickinson, Steven Wright and David Moxey. Non-intrusive UQ with BOUT++ 1D

fluid solver. https://github.com/ExCALIBUR-NEPTUNE/Documents/blob/main/reports/2047356/TN-08-1.pdf.

[7] Derek Groen, Agastya P. Bhati, James Suter, James Hetherington, Stefan J. Zasada, and Peter V. Coveney. FabSim: Facilitating computational research through automation on large-scale and distributed e-infrastructures. *Computer Physics Communications*, 207:375–385, 2016, doi:10.1016/j.cpc.2016.05.020.

[8] Tomasz Piontek Piotr Kopta and Bartosz Bosak. QCG-PilotJob. https://github.com/psnc-qcg/QCG-PilotJob.

[9] VECMA Team. Verified Exascale Computing for Multiscale Applications. https://www.vecma.eu.

[10] Linus Seelinger, Anne Reinarz, Leonhard Rannabauer, Michael Bader, Peter Bastian, and Robert Scheichl. High performance uncertainty quantification with parallelized multilevel markov chain monte carlo. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '21, New York, NY, USA, 2021. Association for Computing Machinery.

[11] Benjamin Dudson and others. SD1D Manual. https://github.com/boutproject/SD1D/tree/master/doc.

[12] SEAVEA Team. Software Environment for Actionable and VVUQ-evaluated Exascale Applications. https://excalibur.ac.uk/projects/seavea/.