

## Numerical error estimation in Random Noise coupled plasma edge simulations in nuclear fusion reactors

K. Ghoo<sup>s\*</sup>, G. Samaey<sup>†</sup>, M. Baelmans<sup>\*</sup>

<sup>\*</sup>KU Leuven, Department of Mechanical Engineering, Celestijnenlaan 300A, 3001 Leuven, Belgium

<sup>†</sup>KU Leuven, Department of Computer Science, Celestijnenlaan 200A, 3001 Leuven, Belgium

kristel.ghoos@kuleuven.be

**Abstract** - The plasma edge in nuclear fusion reactors is simulated using a coupled finite volume / Monte Carlo code. To unravel error contributions from such coupled simulation technique, a framework for error assessment has been developed. It was successfully applied to Random Noise simulations, where the trajectories in the Monte Carlo code are completely uncorrelated each iteration. However, difficulties were encountered in the estimation of the statistical error and the determination of the onset of steadiness. In this paper, several methods are examined to deal with these difficulties, leading to practical methods, which can be easily implemented.

### I. INTRODUCTION

In next generation nuclear fusion reactors such as ITER and DEMO, the divertor is designed for controlling particle and power exhaust from the reactor. In particular, this divertor has to withstand enormous heat power loads. Moreover, it has to ensure an efficient removal of the Helium ash. Simulations of the plasma edge play an essential role in the design of this component, as well as in the preparation and interpretation of experiments [1].

Plasma edge codes simulate the transport of plasma and neutral particles in the plasma edge, i.e. the outer reactor region where the plasma is in contact with the solid target plates and vessel and where neutral particles are removed via vacuum pumps. Plasma particles can be described accurately with fluid equations that are solved using finite volume methods (FV). Neutral particles, however, are governed by a kinetic transport equation that is solved using a Monte Carlo method (MC). To solve the full set of equations, the plasma and neutral codes are coupled and iteratively solved. A code used worldwide is B2-EIRENE [2]. Its latest version, SOLPS-ITER, is the principal tool for plasma edge simulations for ITER [3]. In high recycling and detachment regimes, such as targeted in ITER, plasma-neutral interactions become very strong. This causes convergence issues, increases the required computational time, and complicates the assessment of numerical errors.

Recently, a framework for error assessment has been developed [4] using a simplified 1D plasma edge model. Similar conclusions have been found using a simple slab case with synthetic noise [5]. Several coupling techniques have been examined and compared to each other with respect to speed and accuracy. It was found that making use of post-processing averaging decreases the computational time with an order of magnitude. Plasma edge simulations are usually run with the Random Noise coupling technique, where each MC iteration uses different seeds in the pseudorandom number generator. When particle and momentum balances remain stationary, the code is said to be converged and the final iteration is taken as the solution. To have a sufficiently low statistical error, many MC particles are used each iteration. By examining all error contributions, it has been concluded that significantly fewer MC particles are required per iteration if the average of the results of many stationary iterations is used. This new

simulation approach was recently tested with a B2-EIRENE ITER case [6]. The solution could be obtained in an order of magnitude less computational time without losing accuracy.

To determine the optimal numerical parameters to obtain a required accuracy in the smallest amount of computational time, it is important to have reliable methods to estimate all error contributions. This paper focuses on remaining difficulties in the understanding of the statistical noise that is present in Random Noise simulations, i.e. the assessment of the correlation time of simulation results during steady state iterations and the determination of the onset of steadiness.

The next section introduces briefly the used models and the implementation. In section 3, the error contributions that are present in a Random Noise simulation are discussed. Section 4 investigates methods to estimate the correlation time and presents several examples with variables from a B2-EIRENE ITER simulation. In section 5, methods to determine the transient are discussed.

### II. MODEL AND IMPLEMENTATION

This paper first examines error assessment methods with a simplified 1D model. This plasma model consists of a continuity and momentum equation, solved with FV correction equations. The model is mono-energetic, therefore, no energy equation is needed. False time stepping and relaxation factors are used to aid convergence. Both plasma equations contain source terms from neutral particles that have interacted with plasma particles in ionization or charge exchange events. These sources are computed with the neutral model, which consist of a simplified Boltzmann equation in 1D. Neutral particles are generated at the target, where plasma particles recombine to neutrals (the recycling process). While traveling, neutrals can interact with plasma particles through charge exchange, modeled as an elastic collision, and ionization, which means that the neutral is absorbed and becomes a plasma particle. For more information about this model, the reader is referred to [4]. Examples are also shown for a more complex and realistic ITER case with a partially detached plasma. The plasma equations include energy equations and are simulated in 2D with the B2-code. The neutral equation is simulated in 5D (2D space, 3D velocity) with the EIRENE-code. This test case simulates only Deuterium. More details can be found in

[6].

Each code iteration consists of one iteration of the FV code followed by a MC run using that plasma state as a background. It can be chosen to run the MC code with highly correlated trajectories between the iterations. This is called Correlated Sampling. The noise is said to be frozen and the residuals of the fluid code can decrease to machine accuracy to obtain a converged solution. Alternatively, with Random Noise, the MC code is run with different seeds each iteration such that particle trajectories are uncorrelated. The residuals cannot decrease but remain fluctuating around a constant level. To obtain the solution, an average is then taken over several iterations.

### III. ERRORS IN RANDOM NOISE SIMULATIONS

The total numerical error consists of several contributions. The discretization error is present due to the discretization of the plasma solution. The convergence error exists because the residual of the iterations is not negligible. Two error contributions are caused by the finite number of MC particles per iteration: a deterministic error, which is called the finite sampling bias, and a statistical error, which comes from sampling in the Monte Carlo procedure from a probability distribution.

#### 1. Deterministic errors

The discretization error, the finite sampling bias and the convergence error are all deterministic errors. Methods have been developed to estimate these contributions in Random Noise simulations [4]. The discretization error can be estimated using Richardson's extrapolation [7], as is typically done in FV codes. A similar technique, based on error reduction rates, can be used to estimate the finite sampling bias and the convergence error. These two error contributions cannot be separated from each other, but exhibit the same error reduction rates and can, therefore, be estimated together.

These estimation methods rely on comparing different solutions with each other. To determine the deterministic error associated with the MC noise, it is shown that it suffices to provoke the error using much fewer MC particles than typically used in plasma edge simulations [6]. The averaging should be continued long enough to guarantee that the statistical error is much smaller than the finite sampling bias. Similarly, to estimate the discretization error, grid refinement is needed. In that case, it is essential that the solutions that are being compared have a dominant discretization error. All other error contributions have to be made sufficiently small by choosing a sufficiently large number of iterations and MC particles. If the statistical error is still relatively high, the estimate itself will be dominated by statistical noise. On the other hand, if the statistical error is overestimated, a lot of computational time will be wasted on unnecessary iterations. Therefore, we conclude that a reliable estimate of the statistical error is extremely important for these applications.

#### 2. Statistical error

The statistical error  $\epsilon_s$ , originating from the finite amount of MC particles, stems from a probability distribution with mean 0 and standard deviation  $\sigma$ . This standard deviation  $\sigma$  is from the population of means of  $I$  iterations. The standard deviation of the iterations  $\sigma_1$  can be estimated easily as the sample standard deviation  $s$  of all iterations  $\phi_i$ :

$$s = \sqrt{\frac{1}{I-1} \sum_{i=1}^I (\phi_i - \bar{\phi})^2}, \quad (1)$$

with  $\bar{\phi}$  the average of the total number of iterations  $I$ . However, the standard deviation  $\sigma$  is harder to determine. Indeed, consecutive iterations are not independent of each other. Therefore, the iterations are correlated and the central limit theorem has to be adapted to take into account a correlation time  $T$ . With a known  $T$ , the statistical error can be easily calculated as

$$\epsilon_s \approx \sigma = \sigma_1 \sqrt{\frac{T}{I}}. \quad (2)$$

The correlation time  $T$  is a measure for the dependency between consecutive iterations and can be written as

$$T = 1 + 2 \sum_{\tau=1}^{\infty} \rho(\tau), \quad (3)$$

with  $\rho(\tau)$  the normalized autocorrelation with lag  $\tau$  [8, 9].

### IV. ESTIMATION OF THE CORRELATION TIME

To estimate  $T$ , we examine two approaches: one makes use of the variances of batch means (equation 2), the other is based on the autocorrelation (equation 3). First, the methods are introduced with an example of the simplified 1D model. Afterwards, three examples are given of the B2-EIRENE ITER case.

#### 1. Batching method

The first approach makes use of means of  $M$  consecutive iterations and equation 2. The computation of  $T$  becomes

$$T = \frac{\sigma_M^2}{\sigma_1^2} M, \quad (4)$$

where  $\sigma_1^2$  is the sample variance of the iterations, and  $\sigma_M^2$  is the sample variance calculated with batch means of  $M$  iterations (see equation 1). This equation is valid for  $M \rightarrow \infty$ .

The batches can be either non-overlapping, which means that each new batch starts where the previous one ends, or overlapping, which means that each new batch starts at a next iteration. For the same sample size and batch size, the bias on variance estimates ( $\sigma_M^2$ ) is approximately equal with both methods, however, with overlapping batch means, the variance on the estimated  $\sigma_M^2$  is only 2/3 of the variance estimated with non-overlapping batch means [10]. Figure 1 shows the obtained estimate of  $T$  for several values of  $M$  for the plasma

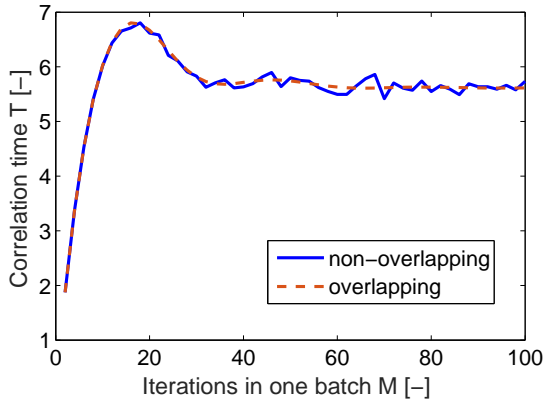


Fig. 1. Estimate of correlation time  $T$  in function of the number of means per batch  $M$  for the target flux in the simplified 1D test case.

density at the target in the simplified 1D model. It is important that the number of iterations per batch  $M$  is chosen sufficiently large. In this example, the estimated  $T$  remains equal when  $M \geq 5 T$ . If  $M$  is chosen too low, the batch means are still dependent on each other and the estimate of  $T$  will be inaccurate. On the other hand, when  $M$  is chosen very large, the number of batch means becomes low and the sample variance  $\sigma_M^2$  cannot be estimated reliably. In this example, many iterations ( $2 \cdot 10^5$ ) are available such that the statistical error on the estimated  $T$  is low.

However, in realistic simulations a relatively small number of iterations is available. Therefore, the estimate can be under- or overestimated with this method. An important advantage of this method is that not every variable of interest has to be stored for each iteration. During the run, batch means and standard deviations can be calculated and stored, as described in [6].

## 2. Autocorrelation method

The second approach to estimate  $T$  is based on equation 3. However, because only a finite number of iterations is available, the infinite sum is truncated at a specific  $\tau_i$ :

$$T = 1 + 2 \sum_{\tau=1}^{\tau_i} \rho(\tau), \quad (5)$$

where the autocorrelation  $\rho(\tau)$  is estimated using samples covariances  $\left(\frac{\text{cov}(I, I-\tau)}{\text{cov}(I, I)}\right)$ . Figure 2 shows the estimated autocorrelation function  $\rho(\tau)$  for the plasma density at the target in the simplified 1D test case (same case as figure 1). The blue line shows the autocorrelation calculated with many iterations ( $> 1000 T$ ), while the red line represents the autocorrelation calculated with few iterations ( $< 100 T$ ). When few iterations are available, as often the case in realistic simulations, the statistical error on  $\rho(\tau)$  can become large. The two dashed lines indicate the  $2\sigma$  confidence interval around  $\rho = 0$ . The correlation time  $T$  can be estimated accurately with the blue line ( $T = 5.7$ ). However, when  $< 100 T$  iterations are used,  $T$  is underestimated ( $T = 2.5$ ). Moreover, the estimate becomes

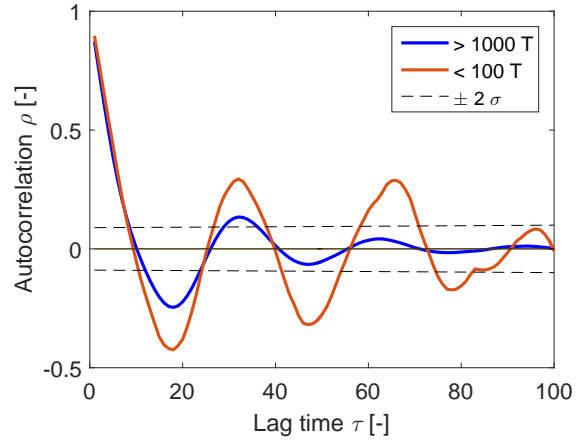


Fig. 2. Autocorrelation  $\rho$  in function of the lag time  $\tau$  for the target flux in the simplified 1D test case. To calculate the blue line, more than 1000  $T$  samples are available, while the red line is calculated with fewer than 100  $T$  samples. The dashed lines indicate the  $\pm 2\sigma$  confidence interval around  $\rho = 0$  belonging to the red line.

very sensitive to the truncation point. The higher  $\tau_i$ , the more influence of statistical noise in the calculation of  $\rho$ . On the contrary, if  $\tau_i$  is chosen too low,  $T$  may also be under- or overestimated. Therefore, this method should only be used when the number of iterations is many correlation times.

To alleviate the influence of statistical noise, a best fit can be constructed to approximate the autocorrelation  $\rho(\tau)$  with a specific function [11]. Typically, an exponentially decaying function is expected in an iterative process. Therefore, the values of  $\rho(\tau)$  that are not dominated by statistical noise can be used to fit an exponential function. The correlation time is subsequently estimated using the fitted function instead of the computed, noisy autocorrelation values. The quality of this estimate, of course, depends on how well the autocorrelation function is approximated. If more information is known about the shape of the autocorrelation function, this can be taken into account during the fit. This method can be used to obtain the correct order of magnitude when the available number of iterations is small, as will be shown in the next examples. In figure 2 the autocorrelation function is an exponentially decaying function multiplied with a periodic function. For this example, the examined variable has a clear periodic behavior in the iterations, which is also visible in the autocorrelation function. Because of this periodic behavior, a fitted exponential does not give a good approximation for this example.

## 3. Examples from a B2-EIRENE ITER case

This subsection gives examples of both methods for three different variables in a realistic simulation. First, attention is brought to the strategy for storing all variables of interest. This introduces a practical way to avoid excessive storage of data without losing too much information to estimate the correlation time.

### A. Storage strategy

With more dimensions, the number of computed variables grows quickly. If the values of all these variables have to be stored each iteration, a huge amount of memory is required. Therefore, instead of saving the values of each iteration, it is chosen to save the average and sample standard deviation of batches of 500 iterations, which can easily be calculated on the fly, as shown in [6]. This way, the number of stored variables is reduced 500 times. With these batches, the correlation time  $T$  can be calculated in two steps:

$$T = T_1 \cdot T_2, \quad (6)$$

where  $T_1$  is the correlation time of one batch, and  $T_2$  the correlation time of the series of batches. The correlation time  $T_1$  is easily calculated with the batching method:  $\sigma_1$  is saved during the run for each batch,  $\sigma_M$  (with  $M = 500$ ) can be calculated using the batch averages. If the batches are independent of each other,  $T_2$  will be 1. However, this is usually not the case, therefore,  $T_2$  should be estimated with either the batching or the autocorrelation method. In all examples,  $2.5 \cdot 10^5$  iterations (in stationary regime) have been executed. Therefore, 500 batches of 500 iterations each are available.

Before giving some examples, it should be mentioned that it is not necessary to have a highly accurate estimate of  $T$ . The statistical error scales only with  $\sqrt{T}$  (equation 2), therefore, a relative error of for example 10% will influence the estimate of the statistical error with only 5%. It is, however, necessary that the order of magnitude is estimated correctly.

It is noted that the correlation time is highly dependent on the simulation parameters. Especially the chosen time step and relaxation factors have a large influence. The correlation time is also extremely variable dependent. In the same simulation, different variables can have correlation times with different orders of magnitude, as is shown in the following examples.

### B. Example 1: ion density on the outer target

In the first example, the estimation of the correlation time of the ion density at a location on the outer target is shown. The estimate for  $T_1$  is calculated to be 148. To estimate  $T_2$ , the batching method and the autocorrelation method can be used and result in a satisfying estimate. The figures show results for total correlation time  $T$ .

Figure 3 shows the estimate of  $T$  with the batching method. The available batches (of 500 iterations each) are grouped together to form bigger batches, overlapping (full line) or non-overlapping (dashed line). When the number of non-overlapping batches is less than 30, the result remains reliable (indicated in dark blue and red). However, when the batches are larger, the estimates for both the overlapping and non-overlapping batches become increasingly more affected by the statistical noise (indicated in light blue and pink). Indeed, it can be observed that the estimate with non-overlapping and overlapping batches give radically different results, which indicates that the estimates are unreliable. In figure 3, it is observed that the value of the correlation time is still slightly increasing with  $M$  (for the reliable dark blue and red results). This increase is not steep anymore, therefore, we can conclude that the order of magnitude of the estimate  $T \approx 550$  will be

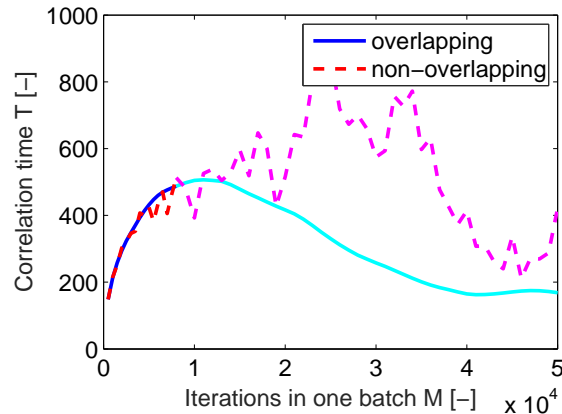


Fig. 3. Estimate of correlation time  $T$  in function of the number of iterations per batch  $M$  for the ion density at a location on the outer target in the B2-EIRENE ITER case. The full and dashed lines represent the results with respectively non-overlapping and overlapping batches. Dark colors (dark blue and red) represent statistically reliable results. When less than 30 non-overlapping batches are used, results are unreliable and presented with light colors (light blue and pink).

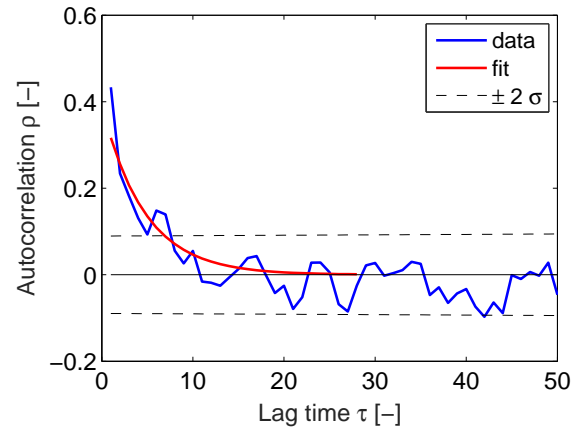


Fig. 4. Autocorrelation  $\rho$  for the batches (of each 500 iterations) in function of the lag time  $\tau$  for the ion density at a location on the outer target in the B2-EIRENE ITER case. The correlation time  $T$  is estimated to be 635 using the exponential fit.

approximately correct. Notice, however, that even at this point, where the number of iterations in one batch equals  $15 T$ , the resulting estimate does not remain constant yet.

Figure 4 shows the calculated values of the autocorrelation  $\rho$  (blue) of the batches (of each 500 iterations) with the  $2\sigma$  confidence intervals around 0 (dashed black). Clearly, only a few values of  $\rho$  contain useful information (outside the  $2\sigma$ -lines). The shown exponential fit (red) is calculated with these values of the autocorrelation and results in an estimated  $T$  of 635. This estimate has indeed the same order of magnitude as the estimate with the batching method.

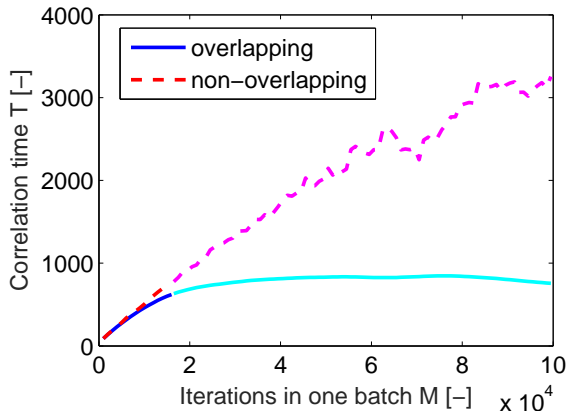


Fig. 5. Estimate of correlation time  $T$  in function of iterations per batch  $M$  for the ion density at a location on the inner target in the B2-EIRENE ITER case. The full and dashed lines represent the results with respectively non-overlapping and overlapping batches. Dark colors (dark blue and red) represent statistically reliable results. When less than 30 non-overlapping batches are used, results are unreliable and presented with light colors (light blue and pink). The (inaccurately) estimated correlation time  $T$  is 700.

#### C. Example 2: ion density on the inner target

Plasma variables on the inner target generally have a much larger correlation time. An example is discussed for the ion density. When the correlation time is very large, the batching method will fail, as figure 5 shows. The last reliable result provides an estimate of 700, however, the value of  $T$  is clearly still increasing. Also the dashed pink line of the estimate with non-overlapping batch means indicates that 700 may be a large underestimation.

Figure 6 shows the calculated values the autocorrelation  $\rho$  (blue) of the batches with the  $2\sigma$  confidence intervals around 0 (dashed black). Using the fitted exponential (red), the correlation time  $T$  is calculated to be 2780, which is more a trustworthy estimate.

#### D. Example 3: effective pumping speed

Previous examples analyzed plasma variables, which are calculated in the FV code. Neutral variables, calculated with the MC code, typically have a much lower correlation time. The MC code itself is linear and computes uncorrelated particle trajectories each iteration. Only the plasma background creates correlation. The example shown here is the effective pumping speed, computed at the pump which is not reached by plasma particles.

Figure 7 shows the estimate of  $T$  with the batching method. The value of  $T$  is very low and changes very slowly. This indicates that the value of the autocorrelation  $\rho$  will be very low but non-zero for a relatively large number of lag times  $\tau$ . Because low values of  $\rho$  are very sensitive to statistical errors, it is impossible to make an accurate estimate using the autocorrelation method, as shown in figure 8. It is concluded that the correlation time for this variable is approximately 2.5.

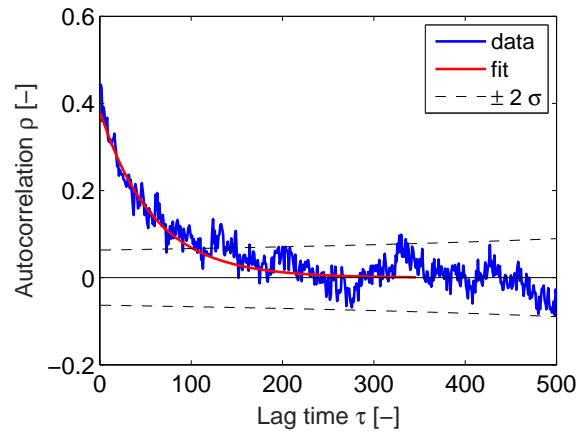


Fig. 6. Autocorrelation  $\rho$  for the batches (of each 500 iterations) in function of the lag time  $\tau$  for the ion density at a location on the inner target in the B2-EIRENE ITER case. The correlation time  $T$  is estimated to be 2780 using the exponential fit.

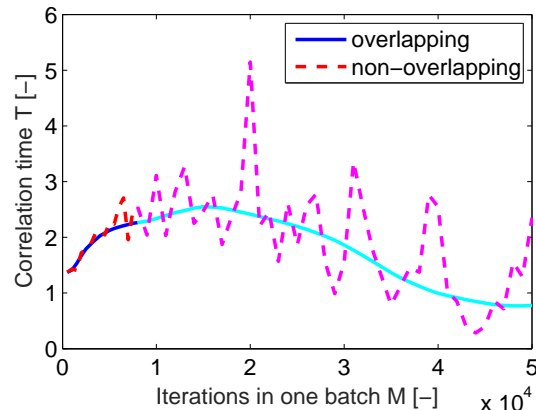


Fig. 7. Estimate of correlation time  $T$  in function of the number of iterations per batch  $M$  for the effective pumping speed in the B2-EIRENE ITER case. The full and dashed lines represent the results with respectively non-overlapping and overlapping batches. Dark colors (dark blue and red) represent statistically reliable results. When less than 30 non-overlapping batches are used, results are unreliable and presented with light colors (light blue and pink). The estimated  $T$  is 2.5.

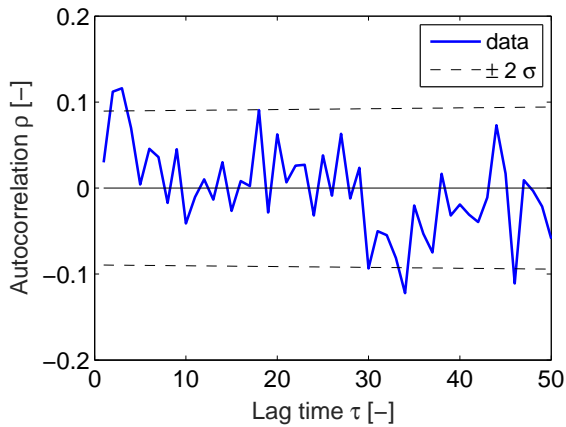


Fig. 8. Autocorrelation  $\rho$  for the batches (of each 500 iterations) in function of the lag time  $\tau$  for effective pumping speed in the B2-EIRENE ITER case. An accurate fit could not be constructed, therefore, the correlation time  $T$  cannot be estimated accurately.

## V. ONSET OF STEADINESS

All estimates assume that the iterations are in stationary regime while taking the average over the iterations. If a transient is still present, the error on the solution can be affected considerably. The additional error caused by the transient is called the initialization bias [12]. It is, therefore, important to discard the iterations of the transient. This number of iterations is usually determined by visually inspecting the progress of the value of variables during the iterations [8]. When the mean of this value appears constant, it is assumed that the stationary regime is reached.

To facilitate the visual examination of the transient, the statistical noise can be decreased by making use of averaging. Welch's procedure proposes two ways [8]: the first method makes use of independent replicas or runs, the second employs a moving average. Multiple replicas are very time-consuming to simulate and usually unnecessary. A moving average over several iterations, on the other hand, is a straight-forward post-processing step that can be extremely useful in decreasing the statistical error for the purpose of determining the end of the transient. Moreover, when only batch averages are stored, less memory is required. Figure 9 shows the batch averages of the plasma density in a cell at the outer divertor target from a B2-EIRENE ITER simulation. Because of the sufficiently small statistical errors of the batches, the transient can be clearly observed. After approximately 300 batches, the value appears to be stationary.

Although visual determination of the end of the transient may be inaccurate, it is generally sufficient [8, 11]. The deterministic error due to a small number of transient iterations, will not highly influence the result if the amount of iterations in the stationary regime is large. However, when many transient iterations are included, large errors can be made. A remaining transient is often indicated by a changing average, because the initialization bias appears. Moreover, the estimated standard deviation of the statistical error is often larger

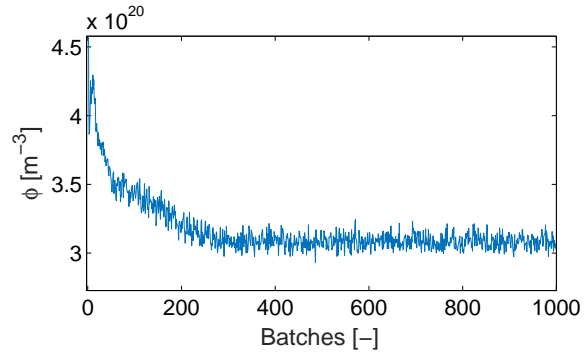


Fig. 9. Batch averages that are kept during the simulation, each consisting of 500 iterations. The transient disappears after approximately 300 batches (visual determination).

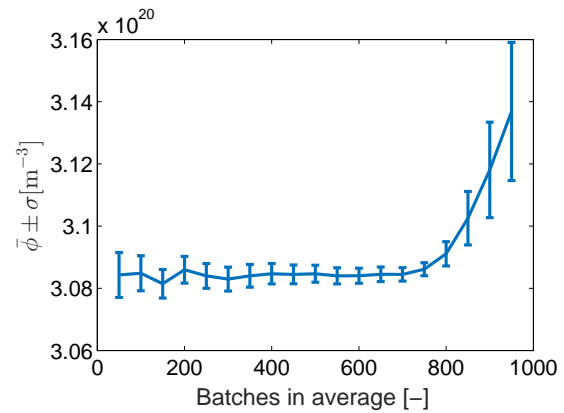


Fig. 10. The average in function of the number of batches used. The errorbars indicate the magnitude of the estimated standard deviation of the statistical error. With more than 800 batches (less than 200 discarded batches), the influence of the transient is clearly visible.

than during stationary iterations. This is mostly due to the estimated correlation time that can suddenly increase when a transient is still present. In figure 10, the average and the standard deviation are calculated with an increasing number of batches, starting from the last batch. When the iterations are in stationary regime, the average changes little and the statistical error decreases with the number of batches/iterations. However, when transient iterations are included, both the average and the standard deviation can change substantially as seen in figure 10 when 800 or more batches are included.

Another well-known method of determining the end of the transient is Schruben's test procedure for initialization bias [12]. For this method, visual inspection is not required. This test determines if the average changes during the run. The probability is estimated that a test statistic  $\hat{\alpha}$  is more unusual than the one that is observed if there is no initialization bias present. When  $\hat{\alpha}$  is smaller than a specified probability  $\alpha_{limit}$ , the hypothesis that no initialization bias is present is rejected. Figure 11 shows the values of  $\hat{\alpha}$  calculated for the example of figure 9: 250 batches should be discarded when  $\alpha_{limit}$  is set to 0.1. This method works well when the transient is clearly

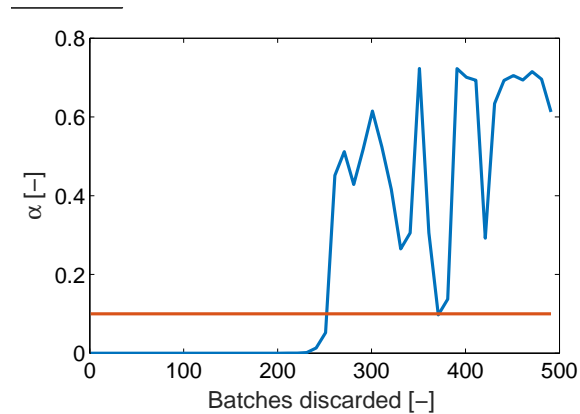


Fig. 11. Test statistic  $\hat{\alpha}$  of Schruben's test for initialization bias (blue) and the specified limit (red) in function of the number of discarded batches. At least 250 iterations should be discarded to avoid an initialization bias.

visible, like in this example. With a less clear transient, taking larger batches can bring improvement (just like with visual determination). However, with a large correlation time, this method sometimes fails to indicate the end of the transient.

While the correlation times can be very different for different variables in the same simulation, the onset of steadiness is almost equal for each variable. Therefore, it is sufficient to determine the end of the transient for one (or a few) variables, where it can be clearly determined and use that value for all variables.

## VI. CONCLUSIONS

This paper focused on two remaining difficulties for estimating errors in Random Noise simulations with coupled FV/MC plasma edge codes. First, the determination of the correlation time was discussed, which is important to obtain a reliable estimate of the statistical error. We examined and discussed advantages and drawbacks for two estimation methods. If the correlation time is low, the best results are provided with the batching method, which makes use of averages of groups of iterations. If the correlation time is high, a better estimate is often provided by the autocorrelation method, which is based on the computation of the autocorrelation of the variable. Second, several methods have been evaluated for the determination of the onset of steadiness. This is important to make reliable error estimates and to avoid an initialization bias. Generally, visual determination is sufficient. However, Schruben's test for initialization bias provides an adequate alternative.

## VII. ACKNOWLEDGMENTS

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No.633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission. This work is sponsored by Flanders Innovation and Entrepreneurship (IWT.141064).

## REFERENCES

1. A. S. KUKUSHKIN, H. D. PACHER, V. KOTOV, G. W. PACHER, and D. REITER, "Finalizing the ITER divertor design: The key role of SOLPS modeling," *Fusion Engineering and Design*, **86**, 2865–2873 (2011).
2. D. REITER, M. BAELMANS, and P. BÖRNER, "The EIRENE and B2-EIRENE codes," *Fusion Science and Technology*, **47**, 172–186 (2005).
3. S. WIESEN, D. REITER, V. KOTOV, M. BAELMANS, W. DEKEYSER, A. KUKUSHKIN, S. LISGO, R. PITTS, V. ROZHANSKY, G. SAIBENE, I. VESELOVA, and S. VOSKOBOYNIKOV, "The new SOLPS-ITER code package," *Journal of Nuclear Materials*, **463**, 480 – 484 (2015).
4. K. GHOOS, W. DEKEYSER, G. SAMAIEY, and M. BAELMANS, "Accuracy and convergence of coupled finite-volume / Monte-Carlo codes for plasma edge simulations of nuclear fusion reactors," *Journal of Computational Physics*, **322**, 162–182 (2016).
5. Y. MARANDET, H. BUFFERAND, G. CIRAOLO, P. GENESIO, P. MELIGA, J. ROSATO, E. SERRE, and P. TAMAIN, "Effect of statistical noise on simulation results with a plasma fluid code coupled to a Monte Carlo kinetic neutral code," *Contributions to Plasma Physics*, **56**, 604–609 (2016).
6. M. BAELMANS, P. BORNER, K. GHOOS, and G. SAMAIEY, "Efficient code simulation strategies for B2-EIRENE," *accepted for publication in Nuclear Materials and Energy* (2016).
7. P. J. ROACHE, "Perspective: A Method for Uniform Reporting of Grid Refinement Studies," *Journal of Fluids Engineering*, **116**, 405–413 (1994).
8. A. M. LAW and W. D. KELTON, *Simulation modeling and analysis*, McGraw-Hill, Inc. (1991).
9. J. FAN and Q. YAO, *Nonlinear Time Series: Nonparametric and Parametric Methods*, Springer Series in Statistics (2003).
10. C. ALEXOPOULOS, D. GOLDSMAN, and J. R. WILSON, "Overlapping batch means: Something more for nothing?" *Proceedings of the 2011 Winter Simulation Conference*, pp. 401–411.
11. M. E. J. NEWMAN and G. T. BARKEMA, *Monte Carlo methods in statistical physics*, Oxford university press (1999).
12. L. W. SCHRUBEN, "Detecting initialization bias in simulation output," *Operations Research*, **30**, 3, 172–186 (1982).