Numerical accuracy and convergence with EMC3-EIRENE

Kristel GHOOS¹, Heinke FRERICH², Wouter DEKEYSER¹, Giovanni SAMAEY³,
Martine BAELEMAN¹

¹ KU Leuven, Department of Mechanical Engineering, 3001 Leuven, Belgium

² University of Wisconsin Madison, Department of Engineering Physics, WI53706, USA

³ KU Leuven, Department of Computer Science, 3001 Leuven, Belgium

Some figures may appear in colour only in the online journal.

Abstract

The iterative Monte Carlo (MC) code EMC3-EIRENE is frequently used for plasma edge simulations in 3D applications. So far, a quantitative evaluation of the numerical quality of the code results remains an open issue. In this paper, we demonstrate a framework for the practical assessment of accuracy and convergence with EMC3-EIRENE. Moreover, we provide a first accuracy analysis with EMC3-EIRENE for a DIII-D divertor edge plasma case. First, we introduce post-processing averaging to efficiently reduce the variance of the statistical error. Then, we estimate the deterministic error contributions based on their theoretical reduction rates by comparing solutions with a different grid resolution, time step, or number of MC particles per iteration. Finally, using parameterized expressions for the error and the computational time, suitable numerical parameters are determined to achieve faster and/or more accurate results. We found that simulations can be more than twice as fast without losing accuracy by making use of post-processing averaging and choosing optimal parameters. In addition, we conclude that the discretization error is the dominant error contribution for the case selected in this paper, which demonstrates the importance of constructing an adequate mesh.

Keywords EMC3-EIRENE, accuracy, convergence, Monte Carlo

1 Introduction

The coupled Monte Carlo - Monte Carlo (MC-MC) code EMC3-EIRENE [1, 2] is developed to enable plasma edge simulations in complex, 3D geometries as present in stellarators such as W7-X and W7-AS. It has also been applied to study 3D phenomena in tokamaks, including ITER [3, 4].

EMC3 solves a simplified version of the Braginskii fluid equations by means of a Monte Carlo (MC) method [5]. Stochastic differential equations are derived to model the evolution of the particle properties. The domain is divided into a number of cells in which quantities of interest (density, velocity, temperatures) are estimated based on particle trajectories. MC particles are followed using small time steps with corresponding random walks, where convective and diffusive processes are taken into account [6]. The main advantage of this method is its high flexibility in the construction of the computational mesh. EMC3 is often coupled to the MC code EIRENE [2], which is used to solve a Boltzmann-type equation for neutral particles in 3D space.

The EMC3-EIRENE code system is solved for steady-state problems by iterating both codes alternately until the relative change in the solution is typically $\leq 1\%$ [7]. It is assumed that this value is a measure of the statistical error on the solution. It has been demonstrated, however, that this stopping criterium is insufficient as an error estimate since the relative change can give a misleading indication and neglects deterministic error contributions [8]. Deterministic errors arise from solving non-linear problems with a finite number of MC particles (bias), advancing fluid MC particles with a discrete time step (time integration error), and using discretized variables in space (discretization error).

Each deterministic error contribution can be estimated based on the theoretical error reduction rates by comparing solutions with a different grid resolution, time step, or number of MC particles per iteration. By varying the time step, the number of MC particles per iteration or the grid size, the magnitude can be estimated for respectively the time integration error, the bias and the discretization error. To obtain reliable estimates, it is important to have a sufficiently low variance on the solutions, which can be efficiently obtained by using post-processing averaging. This does not only enable accuracy analysis, but can also significantly reduce the required CPU-time. This error estimation strategy and simulation approach has been used previously to study

Figure 1: 2D representation of the field line along which the plasma field values will be examined.

the accuracy for B2-EIRENE simulation of a partially detached ITER divertor plasma [9]. In B2-EIRENE, only EIRENE is an MC code, while B2 is a deterministic, finite volume code.

In this paper, we analyze the different numerical error contributions for an EMC3-EIRENE simulation of an axisymmetric DIII-D divertor edge plasma. After a short case description in section 2, the simulation approach is explained, where post-processing averaging is introduced in section 3. Afterwards, we discuss the error reduction rates for the time integration error, bias and statistical error for simulations where only one module, the EMC3 energy module, is solved in section 4. Subsequently, we investigate the accuracy in a fully coupled EMC3-EIRENE simulation for a given grid resolution in section 5. We estimate the time integration errors, biases and statistical errors from all modules and discuss the optimal choice of numerical parameters. Finally, we perform a grid resolution study to analyze the discretization error and its influence on the optimal parameter choice in section 6.

2 Case description

Simulations are performed for an axisymmetric DIII-D divertor edge plasma. Only hydrogen is taken into account, without impurities. The parallel viscosity term in the momentum equation (as can be found in the model equations in [5]) is neglected as this is often done in practice. Anomalous cross-field diffusion coefficients for particle transport, electron energy and ion energy are respectively set to $0.3 \text{ m}^2\text{s}^{-1}$, $1 \text{ m}^2\text{s}^{-1}$ and $1 \text{ m}^2\text{s}^{-1}$. Upstream, the density is fixed at $1.2 \times 10^{13} \text{ cm}^{-3}$, and a particle, ion energy and electron energy source of respectively $1.1 \times 10^{20} \text{ s}^{-1}$, 0.22 MW and 0.22 MW enters the plasma edge from the core. Radial decay lengths at the outermost flux surface of the computational domain are set to 3.0 cm for density, energy and momentum transport. Neutrals are recycled from the Carbon divertor target tiles and the Carbon wall. They can interact with plasma particles through ionization and charge exchange.

The grid is constructed such that it is aligned with the magnetic field. The separatrix divides the domain in three zones: the core region, the scrape-off-layer region, and the private flux region, which respectively consist of 8×180 , 20×252 , 4×72 grid cells in the radial and poloidal direction. In the toroidal direction, 8 grid cells are present over a 40-degree sector. The poloidal distribution of grid nodes in the base mesh is along unperturbed flux surfaces. Cell size is increased empirically in the far scrape-off-layer and private flux region, accounting for the decreasing number of particles that travel in these regions. The 3D grid is generated from a 2D base mesh by tracing field lines from the grid node to subsequent R-Z planes. The large cells in the core domain and near the wall are used for neutral particles only. The plasma solver (EMC3) only operates in the regions with a higher resolution. The plasma grid is thus not extended to the first wall, but the neutral grid is.

In this paper, errors are examined for the plasma field variables along an open field line close to the separatrix. The evaluation along the field line is cut off near the midplane. The red line in figure 1 shows the projection of the chosen field line on a poloidal surface, while the black dots indicate the grid points at this surface. The line is specifically chosen close to the separatrix as plasma profiles typically have relatively large gradients there and we, therefore, expect relatively large numerical errors. The error will be expressed as the L1-norm over the field values ϕ_n as follows:

$$\epsilon = \frac{\sum_{n=1}^N |\phi_n - \phi_{n,\text{ref}}|}{\sum_{n=1}^N |\phi_{n,\text{ref}}|}, \quad (1)$$

Figure 2: Evolution of the electron temperature T_e in a cell close to the target in an EMC3-EIRENE simulation with two simulation approaches indicated (with and without post-processing averaging).

where the reference solution $\phi_{n,\text{ref}}$ is the most accurate, numerically obtained solution that is relevant for the error contribution(s) under investigation. This reference solution is, therefore, chosen differently depending on the examined error contribution. Numerical details are mentioned in later sections. The standard deviation of the iterations σ_1 , which is needed to evaluate the statistical error, is estimated using the sample standard deviation s :

$$\sigma_1 \approx s = \sqrt{\frac{\sum_{i=1}^I (\phi_i - \bar{\phi})^2}{I - 1}}, \quad (2)$$

with I the number of iterations and $\bar{\phi} = \frac{\sum_{i=1}^I \phi_i}{I}$ the average of ϕ_i over all iterations I after a statistically steady-state is reached (see next section).

3 Simulation approach with post-processing averaging

In typical plasma edge simulations with EMC3-EIRENE, three modules are iteratively solved: energy (EMC3), streaming (EMC3) and neutrals (EIRENE). In the energy module, the ion- and electron energy equations are solved for the ion and electron temperature, T_i and T_e . In the streaming module, the plasma continuity and momentum equations are solved for the plasma density n and the Mach number M . In EIRENE, neutral transport is simulated to estimate source terms for the plasma equations coming from plasma-neutral interactions. Starting from an initial condition, the modules are iteratively solved until the solution remains statistically stationary. To achieve convergence, it is often necessary to provide a relaxation (we take 0.6), which limits the change between iterations.

Convergence is usually measured by observing the development of the entire plasma during iterations. Suitable monitoring parameters are, for example, the total recycling flux and the average T_e , T_i and n_e at the targets (e.g. weighted by the local plasma pressure). Strongly depending on the plasma regime under investigation and on the relaxation technique, the evolution could be so slow that the iteration process has to take longer than necessary in order to ensure a converged solution, in which the change of the monitoring parameters remains statistically stationary. So far, the final solution is taken from the last iteration. In this paper, we introduce a post-processing averaging method to improve the statistics.

With averaging, we continue iterating even after the solution has reached a statistically stationary regime. From this point, we start calculating the post-processing average, where we update its value on the fly, as explained in Ref. [10]. This is graphically represented in figure 2, which shows the evolution of the electron temperature in a cell close to the target. We stress that the post-processing average does not interfere with the usual iteration process and is only used in post-processing.

Post-processing averaging is a well-known method to reduce the variance in iterative MC codes in many fields, like nuclear fission [11], statistical physics [12] and turbulent combustion [13]. Within plasma edge simulations, it has been recently introduced in B2-EIRENE, where it lead to an order of magnitude speed up in a partially detached ITER simulation [10]. Also for SOLEDGE2D-EIRENE, it has been indicated that a similar gain can be achieved using averaging [14]. Although, at first sight, it might seem to increase the computational effort, it is used in this work. Indeed, it can significantly speed up the simulation run time, as will be demonstrated in section 5.2.

4 Error reduction rates

The numerical error consist of several contributions: the discretization error, the time integration error, the bias, and the statistical error. In this section, we examine how the error contributions behave as a function of the time step Δt , the number of MC particles per iteration P , and the number of iterations I . The analysis in this section excludes the effect of the grid resolution on the error (i.e. the discretization error), which will be discussed later in section 6. To simplify the analysis, we examine the accuracy of the energy module separately with a fixed background (density n , Mach number M , sources) obtained from a fully coupled, converged EMC3-EIRENE simulation. The results of this analysis will be then applied to a fully coupled EMC3-EIRENE simulation in section 5. First, we discuss the time integration error, which is related to the time step Δt . Afterwards, the statistical error and the bias, which are present due to the finite number of MC particles, are elaborated.

4.1 Time integration error

MC particles in EMC3 are followed in small time steps Δt , with the corresponding random walk step determined by $\Delta r = \sqrt{2\beta_{\parallel}}\Delta t\chi_{\parallel} + \alpha_{\parallel}\Delta t + \sqrt{4\beta_{\perp}}\Delta t\chi_{\perp} + \alpha_{\perp}\Delta t$, with $\chi_{\parallel}$ and $\chi_{\perp}$ random unit vectors respectively parallel (1D) and perpendicular (2D) to the magnetic field, and with $\beta_{\parallel}$, $\beta_{\perp}$, $\alpha_{\parallel}$ and $\alpha_{\perp}$ respectively the diffusion (β) and advection (α) coefficients in the parallel ($\parallel$) and perpendicular ($\perp$) direction as determined by the model equations in Fokker-Planck form [5]. Because perpendicular transport is much smaller, it is customary to choose a larger time step in the perpendicular direction while managing spatial accuracy. The field values n , M , T_e and T_i are calculated from the particle trajectories using track length estimators in each plasma grid cell. These estimators accumulate contributions based on the time that a particle spent in the cell and the particle's weight, which represents a fraction of the total source [5].

This random walk diffusion process is referred to as the Euler-Maruyama discretization, where particle trajectories display $\mathcal{O}(\sqrt{\Delta t})$ convergence in the strong sense and $\mathcal{O}(\Delta t)$ convergence in the weak sense [6]. The tallies based on track length estimators are expected to exhibit a convergence order between these limits. The stochastic integration algorithm adopted in EMC3 has, therefore, an accuracy up to the first order. For the purpose of this work, it is sufficient to determine the order of convergence using only numerical experiments, without rigorous analysis.

The integration errors of the stochastic process arise from non-uniform transport coefficients, including curvature-related terms in non-uniform magnetic fields, and from the boundary conditions, which reflect the finite size scale of an actual system. A natural way to evaluate this error would make use of the ratio of the jump step Δr to the variation lengths of the transport coefficients. In this paper, we choose to characterize the error as a function of the time step Δt , which is the control parameter of the displacement Δr . While this choice limits a general, physical interpretation of this error component, it is easy to use in practice.

To determine the order of convergence experimentally, the solutions from simulations (energy module only) with varying time step are analyzed. First, the parallel time step is changed, keeping the perpendicular one constant ($\Delta t_{\perp} = 2 \times 10^{-7}$ s). Then, the perpendicular time step is varied, keeping the parallel time step constant ($\Delta t_{\parallel} = 10^{-8}$ s). From the electron temperature profile T_e along the selected field line (see section 2), the L1-errors ϵ_t are calculated using the most accurate solution with respectively $\Delta t_{\parallel} = 5 \times 10^{-9}$ s and $\Delta t_{\perp} = 10^{-8}$ s as the reference. As this reference point might be close to the set of points under investigation, care is taken to provide error bars that indicate the statistical uncertainty in the L1-norm coming from the statistical errors on both the investigated solution and the reference solution. All simulations are executed using $P = 10^5$ particles per iteration per processor on 20 processors. The solution is averaged over 100 statistically stationary iterations I .

For sufficiently small Δt , the observed order of convergence is approximately one, as can be seen from the green and red lines in figure 3 for respectively the parallel and perpendicular time step. Therefore, the time integration error (both parallel and perpendicular) can be approximated as

$$\epsilon_{t\parallel} = A_{t\parallel}\Delta t_{\parallel} \quad \text{and} \quad \epsilon_{t\perp} = A_{t\perp}\Delta t_{\perp}, \quad (3)$$

with $A_{t\parallel}$ and $A_{t\perp}$ constants, that can be calculated from the slopes in the figure. For large values of the parallel time step, the asymptotic regime is not reached yet.

Comparing the magnitude of the error with both time steps $\Delta t_{\parallel}$ (green) and $\Delta t_{\perp}$ (red), it is clear that the error related to the parallel time step is much larger for the same Δt : $A_{t\parallel} \approx 10A_{t\perp}$. This confirms that $\Delta t_{\perp}$ can be chosen an order of magnitude larger than $\Delta t_{\parallel}$ without becoming the dominant contribution and compromising accuracy. In the remainder of this paper, a perpendicular time step of 2×10^{-7} s in the energy module is selected for convenience, while the parallel time step can be varied and will be indicated as Δt , without the subscript $\parallel$.

The standard deviation σ_1 of the iterations is almost independent of the time step, as can be observed from the blue and orange lines in figure 3. This indicates that the magnitudes of the bias and the statistical error, which will be discussed next, are independent of the time step.

Figure 3: Time integration error ϵ_t and standard deviation σ_1 with a varying parallel (green, blue) and perpendicular (red, orange) time step Δt for the L1-norm of the electron temperature T_e along the field line.

4.2 Bias and statistical error

The stochastic character of the MC particle trajectories in an iterative non-linear MC code gives rise to a statistical error ϵ_s and a so-called bias ϵ_b [8]. The bias is the deterministic component of the error, which exists due to non-linearities, and decreases quickly with an increasing number of MC particles per iteration $\epsilon_b \propto \frac{1}{P}$. This bias includes the so-called convergence error, which is present due to a remaining non-zero relative change [8]. The statistical error has a probability distribution with mean 0 and a standard deviation σ .

The standard deviation σ characterizes the statistical error and is dependent on the number of MC particles per iteration P and the number of iterations I over which is averaged. The law of large numbers states that the standard deviation of an iteration scales with the inverse of the square root of the number of MC particles per iteration:

$$\sigma_1 = \frac{A_s}{\sqrt{P}}, \quad (4)$$

with A_s a constant. When an average over several iterations I is taken, the statistical error decreases further

$$\epsilon_s \approx \sigma = \frac{\sigma_1}{\sqrt{\frac{I}{T}}} = \frac{A_s \sqrt{T}}{\sqrt{PI}}, \quad (5)$$

with T the correlation time, which is a measure for the dependency between consecutive iterations. It is interpreted as the number of iterations after which a new iteration is statistically independent. The constant A_s is independent of P , I and Δt .

These errors are examined by analyzing the results of simulations obtained with several numbers of MC particles per iteration P , while keeping the time step constant ($\Delta t = 10^{-8}$ s). In figure 4, we show the estimated values of standard deviations σ_1 (blue) and the bias ϵ_b (green) for the L1-norm of the error on T_e for the chosen field line. The standard deviations σ_1 are estimated as sample standard deviations of the iterations. The bias is estimated from the averaged solutions ($I = 2 \times 10^4$) and using a reference solution with $P = 10^6$ ($I = 10^3$). The error bars indicate the statistical uncertainty due to the remaining statistical errors on the averaged solutions. The correlation time T , which is relatively small ($T \approx 5$), is estimated with the batching method (for more details, see [8, 16]). Because of the minor importance of this factor and for clarity of notation in the analysis, the correlation time will be included in the constant for the statistical error: $A'_s = A_s \sqrt{T}$.

It can be observed that the bias is relatively small compared to the standard deviation, even for a relatively small number of MC particles per iteration P . Therefore, the bias will not be of major importance in realistic simulations where a relatively high number of MC particles P is chosen. Even with small P , many iterations are required to reach a sufficiently small statistical error to evaluate the magnitude of the bias accurately. While the remaining statistical errors prevent the theoretical $\frac{1}{P}$ reduction rate from appearing distinctly in this example, the order of magnitude can be observed clearly. The $\frac{1}{\sqrt{P}}$ reduction rate for σ_1 , on the other hand, is clearly confirmed.

Because the number of particles specified in the input file specifies a maximum weight of a particle relative to the source strength, the number of actually followed MC particles in an EMC3 iteration is not generally equal to the specified number. The MC algorithm ensures that particles will be transported from all the sources.

Figure 4: Bias ϵ_b and standard deviation σ_1 as a function of the number of MC particles per iteration P for the L1-norm of the electron temperature T_e along the field line.

Instead of following exactly P particles, EMC3 continues to launch new particles until all sources are accounted for. This guarantees a minimal accuracy in all parts of the domain.

In summary, the total error originating from the limited number of MC particles per iteration is expressed as

$$\epsilon_b + \epsilon_s = \frac{A_b}{P} + \frac{A'_s}{\sqrt{PI}}, \quad (6)$$

where A_b and A'_s are constants, which are independent of P , I , and Δt .

5 Accuracy in EMC3-EIRENE

In a fully coupled EMC3-EIRENE simulation, the energy, streaming and neutral modules are solved sequentially in each iteration. To estimate the accuracy, error contributions from each module need to be taken into account. First, we discuss the parametrized expression for the total error, excluding the discretization error, which is the topic of the next section. Then, after explaining the parametrized model for the CPU-time, the optimal parameters are determined to obtain a minimal error for a given CPU-time.

5.1 Error contributions

The total error on a quantity of interest needs to take into account the error contributions from each module. This is expressed by summing all the contributions of the energy ϵ_{ene} , streaming ϵ_{str} and neutral ϵ_{EIR} module:

$$\epsilon = \epsilon_{ene} + \epsilon_{str} + \epsilon_{EIR}. \quad (7)$$

Like for the energy module, the error for the streaming module consists of a time integration error, a bias and a statistical error. In the neutral module (EIRENE), however, only a bias and a statistical error are present. No time integration error is present in EIRENE, because particle trajectories are followed kinetically from one interaction to the next one without an approximation in time. This results in the following equation:

$$\epsilon = A_{t,e}\Delta t_e + \frac{A_{b,e}}{P_e} + \frac{A'_{s,e}}{\sqrt{P_e I}} + A_{t,s}\Delta t_s + \frac{A_{b,s}}{P_s} + \frac{A'_{s,s}}{\sqrt{P_s I}} + \frac{A_{b,E}}{P_E} + \frac{A'_{s,E}}{\sqrt{P_E I}}, \quad (8)$$

where the constants and parameters related to the different modules are indicated with the additional subscripts $_e$ (energy), $_s$ (streaming) and $_E$ (EIRENE).

As mentioned in section 4.2, the statistical error can be estimated by calculating the sample standard deviation σ_1 and making use of the law of large numbers. More details can be found in [9, 15], where an identical approach is followed.

The constants for the deterministic errors can be estimated by comparing solutions with a different resolution. This method (often referred to as ‘Richardson extrapolation’) is typically used to determine the discretization error in finite volume codes [17], but can be applied to estimate any deterministic error with a consistent reduction rate. For parameter Y (which can be Δt or P for the errors ϵ_t and ϵ_b respectively), two solutions ϕ^Y and $\phi^{\alpha Y}$ with respectively resolution Y and αY are required, with α the scaling factor. Assuming

$\epsilon = A_Y Y^{p_Y}$, with p_Y a known order of convergence ($p_{\Delta t} = 1$ and $p_P = -1$), the error on solution ϕ^Y is estimated as

$$\epsilon^Y = \frac{\phi^{\alpha Y} - \phi^Y}{1 - \alpha^{p_Y}}. \quad (9)$$

From this, the constant can be easily obtained as $A_Y = \frac{\epsilon^Y}{Y^{p_Y}}$. Because the biases and time integration errors are relatively low, it is recommended to choose a lower resolution $\phi^{\alpha Y}$ than the standard choice ϕ^Y . This will not only save CPU-time, but also likely provide a more reliable estimate [15]. To obtain an accurate estimate, it is important to provoke the deterministic error that needs to be estimated, such that it is the dominating error in the difference $\phi^{\alpha Y} - \phi^Y$. Because equation 9 assumes that this difference is exclusively coming from ϵ^Y , remaining statistical errors undermine its accuracy. Post-processing averaging can, therefore, improve the accuracy by reducing the unwanted statistical errors. Another potential source of inaccuracy is the assumption that the asymptotic regime is reached where the expected error reduction rate is valid. A too low resolution should, therefore, be avoided. Although it is difficult to achieve a reliable estimate using only two solutions, it can give a reasonable evaluation of the order of magnitude of the error. More details can be found in Refs. [9, 15].

For the considered example, table 1 lists the approximate magnitude of the constants. Because these numbers represent the order of magnitude of the error rather than an accurate estimate, only one digit is given. For ease of interpretation, the typical parameter values are taken as a reference to scale the constants. This results in the following error equation:

$$\epsilon = A_{t,e}^* \frac{\Delta t_e}{\Delta t_e^*} + \frac{A_{b,e}^* P_e^*}{P_e} + \frac{A_{s,e}^* \sqrt{P_e^*}}{\sqrt{P_e I}} + A_{t,s}^* \frac{\Delta t_s}{\Delta t_s^*} + \frac{A_{b,s}^* P_s^*}{P_s} + \frac{A_{s,s}^* \sqrt{P_s^*}}{\sqrt{P_s I}} + \frac{A_{b,e}^* P_E^*}{P_E} + \frac{A_{s,E}^* \sqrt{P_E^*}}{\sqrt{P_E I}}, \quad (10)$$

with reference time steps $\Delta t_e^* = \Delta t_s^* = 5 \cdot 10^{-8}$ s, reference number of particles per iteration per processor $P_e^* = P_s^* = P_E^* = 10^5$ (20 processors are used in parallel). Because of this representation, the table gives the values of the error contributions for a typical choice of parameter values. We make the following observations:

- There is a clear difference in magnitude between the streaming and energy time integration error constants, $A_{t,s}^*$ and $A_{t,e}^*$, with the latter being an order of magnitude higher. This suggests that Δt_s can be chosen higher than Δt_e without compromising accuracy, thereby consuming less CPU-time in the streaming module.
- The several constants for the bias $A_{b,e}^*$, $A_{b,s}^*$ and $A_{b,E}^*$ are of the same order of magnitude. Since also the relevant statistical errors ($A_{s,s}^*$ for the error on n and $A_{s,e}^*$ for the error on T_e) have a similar order of magnitude, choosing the number of MC particles approximately equal in each code module results in similar values for these errors. Whether this is the optimal choice of parameters also depends on the CPU-time required in each module, which will be discussed further in the next subsection.
- The constant for the statistical error A_s^* is only relevant for the variables calculated in the module itself. The statistical error on the density, for example, is affected by the number of particles in the streaming module P_s , but not significantly by the number of MC particles in the energy module P_e or in EIRENE P_E . The latter (P_e and P_E) will only have an effect on the density error in an average sense, through the bias.
- The time integration error for the energy module and the statistical error are dominant and within $\approx 1\%$ for the typical parameter choice in this example, excluding the discretization error.

The constants in table 1 will be used in the next section to analyze the optimal choice of numerical parameters to achieve faster and/or more accurate simulation results. We stress that these constants are estimated for this specific case (see section 2). Different values for the constants will be obtained for different conditions and plasma parameters.

5.2 Optimal parameters

Using parametrized expressions for the error and the CPU-time, the optimal choice of the numerical parameters (Δt_e , Δt_s , P_e , P_s , P_E , I) can be determined to achieve a minimal error in a specified CPU-time. Before examining the optimal parameters, the model for the CPU-time is shortly discussed.

While the CPU-time required for each module scales linearly with the number of MC particles P and the number of iterations I , the dependency on the time step is less clear. In the considered practical range for the time steps, we fit a function of the form $a\Delta t^b$, with a and b the fitted constants, to the measured CPU-times for several time steps in each EMC3 module. The red and blue lines in figure 5 show the measured CPU-times (full lines) and the best fit (dashed lines) for respectively the energy and streaming module. Since EIRENE is independent of the time step, its CPU-time is constant w.r.t. the time step ($b_E = 0$). Moreover, its value is relatively small, as can be observed by comparing the green line (EIRENE) in figure 5 to the red (energy) and blue (streaming) lines for the usual time step $\Delta t \approx 5 \times 10^{-8}$ s. These CPU-times are calculated as an average of the measured CPU-times for several iterations with $P^* = 10^5$ on one processor.

Table 1

Constants for the error contributions,
according to equation 10.

	n	T_e
$A_{t,e}^*$	2×10^{-3}	2×10^{-3}
$A_{t,s}^*$	3×10^{-4}	1×10^{-4}
$A_{b,e}^*$	4×10^{-4}	2×10^{-4}
$A_{b,s}^*$	5×10^{-4}	6×10^{-4}
$A_{b,E}^*$	3×10^{-4}	1×10^{-4}
$A_{s,e}^*$	negligible	4×10^{-3}
$A_{s,s}^*$	7×10^{-3}	negligible
$A_{s,E}^*$	negligible	negligible

Figure 5: Measured CPU-times (full) and their fits (dashed) as a function of the time step Δt for each module for $P^* = 10^5$ on 1 processor.

Figure 6: Minimal error that can be obtained within a specified computational time for the non-optimal (red) and the optimal (black) choice of parameters.

The resulting parametrized expression for the CPU-time is the sum of the contributions of the energy, streaming and neutral modules:

$$\text{CPU-time [s]} = \left(a_e \Delta t_e^{b_e} \frac{P_e}{P^*} + a_s \Delta t_s^{b_s} \frac{P_s}{P^*} + a_E \frac{P_E}{P^*} \right) (I_{\text{tr}} + I), \quad (11)$$

with I_{tr} and I respectively the number of unsteady iterations in the transient, which have to be discarded, and the number of statistically stationary iterations, which contribute to the averaged solution. To take into account multiple parallel processors (20 here), it is sufficient for the purpose of this text to obtain the wall clock time by simply dividing the CPU-time by the number of processors, neglecting the overhead time. For this analysis, the following values have been used: $a_e = 0.001$, $b_e = -0.7$, $a_s = 0.0002$, $b_s = -0.8$, $a_E = 20$.

Using optimized parameters, a specified numerical error can be obtained more than twice as fast than with the initial, non-optimal parameter choice. Alternatively, for the same amount of computational time, approximately twice as accurate results can be reached with optimized parameters than with the non-optimal approach. This is illustrated in figure 6, where the minimal errors that can be reached with optimized parameters (black) are compared to the error obtained with the non-optimal parameter choice (red, $\Delta t_e = \Delta t_s = 5 \times 10^{-8} \text{s}$, $I = 1$, $P_e = P_s = P_E$). The selection of the optimal parameter choice is achieved by evaluating the total error for many parameter combinations in a large range and finally choosing the combination of parameters that provides the minimal error. Parameter scans have been performed over the six free numerical parameters: Δt_e , Δt_s , I , P_e , P_s , P_E . For this optimization, we use the constants for the error on n_i (see table 1, where $A_{s,e}$ is substituted by its value for the error on T_e in equation 10. Since all other constants (A_t , A_b) are of the same order of magnitude for n and T_e , the error will be similar for both the energy and streaming variables, and large statistical errors on the temperature are avoided. The number of iterations in the transient and the correlation time are respectively $I_{\text{tr}} = 50$ and $T = 5$.

To highlight the main differences between the non-optimal and the optimal approach, we discuss the minimal error that can be reached in 10^5s (≈ 1.2 days). By optimizing the numerical parameters, the total error is reduced by $\approx 50\%$. Details from all error contributions and the distribution of the CPU-time over the modules are illustrated in figure 7. Two aspects, which are discussed next, lead to the increased accuracy: the introduction of post-processing averaging which allows a lower choice of P , and the optimally chosen time steps. We stress that the simulation with the optimal parameter choice provides the solution in the same amount of computational time as the simulation with the original parameter choice.

Thanks to post-processing averaging, the total statistical errors $\epsilon_{s,e} + \epsilon_{s,s}$ has decreased by almost 50%, while the sum of the biases remains below 20% of the total error. To decrease the statistical error, both the number of MC particles per iteration P and the number of iterations I can be increased. With a lower choice of P (while keeping the time $\propto P(I_{\text{tr}} + I)$ constant), the time for the transient decreases, thus a higher share of the time contributes to the solution. Because the bias increases with a decrease in P , the optimal choice of P and I is a compromise between the statistical error and the bias. This optimum is typically wide and, thus, a large range of choices of P and I will lead to an increased accuracy [18, 19]. In this example (see figure 7), the number of iterations in the average was increased from 1 to 70, while the number of MC particles per iteration

Figure 7: Shares of the computational time for the modules (left), magnitude of the error contributions (middle), and parameter choice (right) for the usual (top) and optimal (bottom) selection of numerical parameters for a wall clock time of approximately 10^5 s (≈ 1.2 days).

was decreased from 5×10^5 in each code module to 3×10^5 in EIRENE and the EMC3 streaming module, and 10^5 in the EMC3 energy module.

The total time integration error $\epsilon_{t,e} + \epsilon_{t,s}$ has decreased drastically by exploiting the magnitude difference for the time integration error constants in the energy and streaming module. Time is saved by increasing the streaming time step Δt_s from $5 \times 10^{-8}s$ to $10^{-7}s$, with only a slight increase in $\epsilon_{t,s}$. This time is now used in the energy module, where a smaller time step Δt_e ($10^{-8}s$ instead of $5 \times 10^{-8}s$) significantly reduces the time integration error $\epsilon_{t,e}$. The magnitude difference between $A_{t,e}$ and $A_{t,s}$ is present because the parallel viscosity term in the momentum equation is neglected in this example. The parallel transport of the MC particles is, therefore, deterministic, such that the parallel time step limit is determined by the cross-field transport.

We note that the computational time required for EIRENE is relatively small in this low recycling case. With larger neutral densities in high recycling or detachment regimes, we expect this share to increase.

While different optimal parameters will be obtained for different case studies, we illustrated that the computational performance of EMC3-EIRENE simulations can be substantially improved by choosing the numerical control parameters appropriately. Knowledge of the numerical error contributions is, therefore, not only necessary to assess the numerical quality of the solution, but can also be used to optimize the code performance. We specifically point out the benefit of using post-processing averaging to increase the computational efficiency of EMC3-EIRENE simulations. The decrease in statistical error by taking an average over subsequent iterations enables the use of a smaller number of MC particles per iteration P . Since the bias is typically very small for a realistic choice of P , the accuracy remains high even with a smaller-than-usual P .

6 Effect of grid resolution

Additional to the time integration errors, bias and statistical error, the limited grid resolution gives rise to a discretization error. First, the magnitude of this error component is discussed by comparing simulations on three successively higher grid resolutions. Afterwards, it is explained how this error affects the optimal parameter choice.

6.1 Discretization error

The approximations due to the finite number of grid cells give rise to a so-called discretization error. Within each grid cell, the value of the scored variables is assumed to be constant. Moreover, during particle trajectories, transport coefficients and source terms are calculated based on the background densities, velocities and

Figure 8: Density (left) and electron temperature (right) profile for three successively finer grid resolutions for the chosen field line.

temperatures obtained in the previous iteration.

To examine the effect of grid resolution on the accuracy of the solution, simulations have been performed on three successively finer grid sizes. Two finer grids have been constructed with a two and four times higher resolution in each dimension compared to the original grid. Each grid is constructed separately such that field alignment is ensured. In practice, the grid cells of the original, low resolution grid are approximately split in 8 and 64. The runs have been performed with $\Delta t_e = 0.2 \Delta t_s = 10^{-8}$ s, and $P_e = P_s = 0.5 P_E = 10^5$ on 20 processors.

The solution significantly changes with grid resolution, as can be seen from figure 8, where respectively the density (left) and electron temperature (right) profiles along the selected field line are shown for the low (red), medium (blue) and high (black) resolution grid. Clearly, the density decreases and the electron temperature increases with higher resolution. Differences between the low and high resolution solutions are of the order of magnitude 10-15%.

In an MC code, second order convergence with grid resolution is expected ($p_X = -2$), because the variables are approximated as piece-wise linear profiles in the particle tracking algorithm [8]. The order of convergence p_X is estimated using the following equation [17]:

$$p_X = \frac{\log\left(\frac{\phi^{MR} - \phi^{HR}}{\phi^{LR} - \phi^{MR}}\right)}{\log(2)}, \quad (12)$$

where ϕ^{LR} , ϕ^{MR} and ϕ^{HR} are the solutions on the low, medium and high resolution grid, respectively. We evaluate p_X at each location along the profile on the low resolution grid, which results in an average of $p_X = -1.9$ for the density and $p_X = -0.50$ for the electron temperature. The predicted order of convergence is thus retrieved for the density, but not for the electron temperature. We anticipate that the asymptotic regime is not yet reached for these relatively low resolution grids and we expect it to appear for higher resolutions.

The discretization error on the original low resolution grid is calculated to be 8% and 16% for the density and electron temperature respectively. This L1-error is calculated with a reference solution extrapolated from the medium and high resolution grid, assuming second order convergence. Comparing these values to the errors in table 1, we conclude that the discretization error is the dominant error contribution for the usual parameter choice. For the medium and high resolution grid, the discretization errors are estimated to be between respectively 5-10% and 2-5%.

The dominant discretization error demonstrates the need for a good grid resolution. Globally refining the grid, as done in this paper, is easy, but not optimal in practice. Indeed the computational time as well as the memory requirements increase considerably with a higher number of grid cells. The solution is most sensitive to the discretization in regions with large gradients and thus strong variations in the parameters. Optimized grids can exploit this by increasing the resolution only in these regions of the domain, which can be identified by comparing the physical length scales to the cell width. This example showed very large variations in the density and Mach number in the first cells next to the target. It is, therefore, expected that the numerical accuracy can largely benefit from an increased resolution near the target.

6.2 Optimal parameters

Since the discretization is dominant, we expect a large influence of the grid resolution on the optimal parameter choice. In this section, we include grid dependency in the equations for the error and the computational time and discuss its influence on the optimal parameters.

Figure 9: Minimal error, including the discretization error, that can be obtained within a specified computational time for the low (red), medium (blue) and high (black) resolution grid. The full and dashed lines represent the results with and without post-processing averaging, respectively.

Apart from adding the discretization error to equation 8, the grid dependency of the other error contributions needs to be taken into account. Because grid cells are smaller, more MC particles will have to be followed to keep the same accuracy. The variance, therefore, increases with higher grid resolution. The following scaling has been approximately observed $\sigma_1^2 \propto X^{1.4}$, where X is the refinement in each direction (respectively 1, 2 and 4 for the low, medium and high resolution grid). This is also the expected scaling for the bias. Concerning the time integration error, it has been verified that its magnitude does not significantly change with grid size. This results in the following parametrized equation for the total numerical error:

$$\epsilon = \frac{A_d}{X^2} + A_{t,e}\Delta t_e + \frac{A_{b,e}X^{1.4}}{P_e} + \frac{A'_{s,e}X^{0.7}}{\sqrt{P_e I}} + A_{t,s}\Delta t_s + \frac{A_{b,s}X^{1.4}}{P_s} + \frac{A'_{s,s}X^{0.7}}{\sqrt{P_s I}} + \frac{A_{b,E}X^{1.4}}{P_E} + \frac{A'_{s,E}X^{0.7}}{\sqrt{P_E I}}, \quad (13)$$

with $A_d = 0.08$ and the other constants taken consistent with table 1.

For the computational time, equation 11 is used, where different constants a_e , b_e , a_s , b_s , a_E are calculated for each grid separately. Estimates for these constants are obtained using runs with two different values for Δt_e and Δt_s with $P = 10^5$ particles per processor on 20 parallel processors. This resulted in the following values: $a_e = 1.4$, $b_e = -0.3$, $a_s = 0.01$, $b_s = -0.6$ and $a_E = 34$ for the low resolution grid; $a_e = 7$, $b_e = -0.2$, $a_s = 0.03$, $b_s = -0.5$ and $a_E = 40$ for the medium resolution grid; $a_e = 0.5$, $b_e = -0.4$, $a_s = 50$, $b_s = -0.16$ and $a_E = 80$ for the high resolution grid. Although these estimates are rather rough because of the limited number of simulations, they give a sufficiently good representation of the computational times for the purpose of this text.

A high grid resolution is required to achieve a small total error, as can be seen from figure 9, where the minimal error is shown as a function of the wall clock time on the low (red), medium (blue) and high (black) resolution grid. Averaging is the best choice in all cases, as can be seen from the difference between the full and dashed lines, which indicate respectively the results with and without averaging. It is clear that the error with the low resolution grid cannot decrease with time because the discretization error is dominant. Even for very small computational times, a higher accuracy can be obtained with the medium grid. With more time available (from $\approx 9 \times 10^5$ s ≈ 1 day), the highest accuracy is obtained with the high resolution grid. The horizontal difference between the dashed (without averaging) and full (with averaging) lines indicates again a speed up of approximately factor 2. This speed up is thanks to the decrease of the statistical error with averaging. Of course, the advantages of averaging increase when the statistical error is dominant.

We conclude that the chosen grid resolution has a large influence on the numerical accuracy of the solution. For this example, even with a limited amount of computational time, the numerical error can be reduced from $\approx 8\%$ to $\approx 3\%$ by choosing a twice as high resolution in each direction than usual. This highlights the value of constructing more optimal grids for EMC3-EIRENE simulations.

7 Conclusion

In this paper, we demonstrated a framework to evaluate numerical accuracy and convergence in EMC3-EIRENE simulations. We also provided a first evaluation of the numerical accuracy with EMC3-EIRENE for the simulation of an axisymmetric DIII-D divertor edge plasma. This comprises the evaluation of all error contributions,

originating from approximations due to the chosen time steps, the numbers of MC particles, the number of iterations and the grid resolution. Using parametrized expressions for the error and the computational time, we calculated the optimal numerical parameters to achieve a minimal error for a specified computational time. In this analysis, we made use of post-processing averaging for variance reduction, which significantly speeds up the simulations as results could be obtained approximately twice as fast for the same accuracy. Furthermore, we demonstrated that the discretization error, originating from the limited grid resolution, is the dominant error contribution with a magnitude of 10-15% for the parameter choice as was originally used in this case.

With this analysis, we illustrated that simulations with EMC3-EIRENE can become significantly more efficient by applying post-processing averaging and choosing more adequate numerical parameters. We also demonstrated the substantial influence of the numerical grid on the accuracy, which stresses the importance of a good mesh construction and optimization.

Acknowledgements

The work of K. GhooS was sponsored by Flanders Innovation and Entrepreneurship (IWT.141064) and a travel grant (V4.128.18N) from Research Foundation - Flanders (FWO). Parts of the work are supported by the Research Foundation Flanders (FWO) under project grant G078316N. The computational resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government – department EWI.

References

- [1] Feng Y *et al* 2004 *Contr. Pl. Phys.* **44** 57
- [2] Reiter D *et al* 2005 *Fus. Sc. Techn.* **47** 172
- [3] Feng Y *et al* 2014 *Contr. Pl. Phys.* **54** 426
- [4] Schmitz O *et al* 2016 *Nuclear Fusion* **56** 066008
- [5] Feng Y *et al* 1999 *Journ. Nucl. Mat.* **266-269** 812
- [6] Van Kampen N G 1981, Stochastic Processes in Physics and Chemistry. *North-Holland: Amsterdam*
- [7] Kobayash M *et al* 2004 *Contr. Pl. Phys.* **44** 25
- [8] GhooS K *et al* 2018 *Contr. Pl. Phys.* **58** 652
- [9] GhooS K *et al* 2019 *Nuclear Fusion* **59** 026001
- [10] Baelmans M *et al* 2016 *Nucl. Mat. Ene.* **12** 858
- [11] Brown F B 2009, A review of Monte Carlo criticality calculations - convergence, bias, statistics, *International Conference on Mathematics, Computational Methods and Reactor Physics*, New York, American Nuclear Society
- [12] Newman M E J *et al* 1999, Monte Carlo methods in statistical physics. *Oxford university press*
- [13] Xu J *et al* 1999 *Journ. Comp. Phys.* **152** 192
- [14] Marandet Y *et al* 2016 *Contr. Pl. Phys.* **56** 604
- [15] GhooS K 2019, Accuracy-based simulation strategies for plasma edge simulations for nuclear fusion devices, *PhD thesis*, KU Leuven
- [16] GhooS K *et al* 2017, Numerical error estimation in Random Noise coupled plasma edge simulations in nuclear fusion reactors, *Proc. of International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, Korea, Korean Nuclear Society
- [17] Roache P J 1994 *Journ. Fl. Eng.* **116** 405
- [18] GhooS K *et al* 2016 *Journ. Comp. Phys.* **322** 162
- [19] Dekeyser W *et al* 2018 *Contr. Pl. Phys.* **58** 643