Multigroup-like MC resolution of generalised Polynomial Chaos reduced models of the uncertain linear Boltzmann equation (+discussion on hybrid intrusive/non-intrusive uncertainty propagation)

Gaël Poëtte

Multigroup-like MC resolution of generalised Polynomial Chaos reduced models of the uncertain linear Boltzmann equation

(+discussion on hybrid intrusive/non-intrusive uncertainty propagation)

Gaël Poëtte[a]

[a] *CEA DAM CESTA, F-33114 Le Barp, France*

Abstract

Monte Carlo-generalised Polynomial Chaos (MC-gPC) has already been thoroughly studied in the literature [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. MC-gPC both builds a gPC based reduced model of some kinetic equations of interest and solves it with an MC scheme in order to propagate uncertainties. In this paper, the kinetic equation of interest is the linear Boltzmann equation. MC-gPC allows important computational gains on this model on many applications [1, 4, 5, 6, 8, 9]: in a nutshell, the reasons for its success are spectral convergence [2] plus the fact that it is based on an MC resolution [1]. Furthermore, MC-gPC can be implemented thanks to *simple* modifications of an existing MC code [1, 9, 8]. But MC-gPC also presents some weaknesses: it is sensitive to the curse of dimensionality [1, 9] and is noisier than other strategies [10]. The aim of this paper is to present new MC schemes solving the same gPC based reduced model but attenuating the two previous drawbacks. They are based on multigroup-like resolution methods. The new MC schemes improve the run times of MC-gPC. The resolution scheme is intrusive: this means that modifications of an existing solver are necessary (even if people familiar with multigroup MC resolution will not be intimidated by them). The paper ends with a discussion about taking into account uncertainties at the early stages of the development of a simulation code together with some original and efficient hybrid intrusive/non-intrusive applications.

Keywords: Monte Carlo, generalised Polynomial Chaos, Uncertainty Quantification, Transport

1. Introduction

Monte Carlo-generalised Polynomial Chaos (MC-gPC) has already been thoroughly studied in the literature [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. MC-gPC both builds a gPC based reduced model of some kinetic equations of interest and solves it with an MC scheme in order to propagate uncertainties. In [3, 4], the kinetic equation is a flocking model; in [5, 6, 7] it corresponds to the quadratic Boltzmann equation; in [1, 2, 8, 9, 10], it is (or is closely related to) the linear Boltzmann equation. We also focus on the latter in this paper. It is given by

$$\begin{aligned}\partial_t u(\mathbf{x},t,\mathbf{v},\mathbf{X}) + \mathbf{v}\cdot\nabla_{\mathbf{x}} u(\mathbf{x},t,\mathbf{v},\mathbf{X}) = &- v\sigma_t(\mathbf{x},\mathbf{v},\mathbf{X})u(\mathbf{x},t,\mathbf{v},\mathbf{X}) \\ &+ v\int \sigma_s(\mathbf{x},\mathbf{v}\cdot\mathbf{v}',\mathbf{X})u(\mathbf{x},t,\mathbf{v}',\mathbf{X})\,\mathrm{d}\mathbf{v}',\end{aligned} \tag{1}$$

Email address: gael.poette@cea.fr (Gaël Poëtte)

together with the initial and boundary conditions

$$\begin{array}{l} u(\mathbf{x}, t=0, \mathbf{v}, \mathbf{X}) = u_0(\mathbf{x}, \mathbf{v}, \mathbf{X}), \quad \mathbf{x} \in \mathcal{D}(\mathbf{X}), \quad t \in [0,T] \quad \mathbf{v} \in \mathcal{V}, \quad \mathbf{X} \in \Omega, \\ u(\mathbf{x}, t, \mathbf{v}, \mathbf{X}) = u_b(t, \mathbf{v}, \mathbf{X}), \quad \mathbf{x} \in \partial\mathcal{D}(\mathbf{X}), \quad t \in [0,T] \quad \mathbf{v} \cdot n_s(x, \mathbf{X}) < 0, \quad \mathbf{X} \in \Omega, \end{array} \tag{2}$$

where n_s is the outward normal to Ω at $\mathbf{x}$. In the above expression, u is a density of particles. The variables $\mathbf{x} \in \mathcal{D} \subset \mathbb{R}^3$, $t \in [0,T] \subset \mathbb{R}^+$ and $\mathbf{v} \in \mathcal{V} \subset \mathbb{R}^3$ are respectively the space, time and velocity[1] variables. Variable $\mathbf{X} = (X_1, ..., X_Q)^t$ is a vector of Q independent[2] random variables of probability measure $\mathrm{d}\mathcal{P}_{\mathbf{X}} = \prod_{i=1}^{Q} \mathrm{d}\mathcal{P}_{X_i}$ modelling the uncertainties. Variables $(\mathbf{x}, t, \mathbf{v})$ are the *physical* variables in opposition to $\mathbf{X}$ which is refered to as the *uncertain* variable. The cross-sections $\sigma_t = \sigma_t(\mathbf{x}, \mathbf{v}, \mathbf{X})$, $\sigma_s = \sigma_s(\mathbf{x}, \mathbf{v} \cdot \mathbf{v}', \mathbf{X})$ are assumed to be given functions of $(\mathbf{x}, \mathbf{v}, \mathbf{X})$ and $(\mathbf{x}, \mathbf{v} \cdot \mathbf{v}', \mathbf{X})$ in this paper. They stand for the total and scattering cross-sections. The quantity σ_s defines how the velocities and angles are scattered when a reaction is encountered. Of course, the above notations are for macroscopic cross-sections, in the sense that many physical reactions are summed-up in the above notations, see [16, 17, 18]. System (1) together with boundary conditions (2) define the well-posed [19] mathematical problem we want to solve and in which we want to *be able to accurately take uncertainties*[3] *into account*. In other words, we are mainly interested in the statistics of $\mathbf{X} \to u(\mathbf{x}, t, \mathbf{v}, \mathbf{X})$ (i.e. mean, variance, histogram, sensitivity indices [20] etc.) at specified locations $\mathbf{x} \in \mathcal{D}$, times $t \in [0,T]$ and velocities $\mathbf{v} \in \mathcal{V}$. The uncertain transport equation is of importance in many physical domains such as neutronics [21, 22, 23, 9], photonics [24, 25, 26, 27, 28, 8, 29], biology [30], socio-economics [6, 31, 4], epidemiology [32] etc. In neutronics [9] or photonics [8] for example, equation (1) must be solved at each iteration/time step.

Of course, different values of $\mathbf{X}$ correspond to different fully decoupled deterministic equations: in principle, there is no difficulty in solving such uncertain problem. The main issue comes from the fact that exact propagation of uncertainties is very expensive from the computational point of view: equation (1) is often solved thanks to an MC scheme [33, 21, 34, 22, 23, 35, 36, 37]. This resolution method is known to be efficient for high $(3(\mathbf{x})+1(t)+3(\mathbf{v}) = 7)$ dimensional problems but costly. Running several deterministic MC computations for several values of $\mathbf{X}$ can consequently be prohibitive.

This is precisely where MC-gPC comes at play: MC-gPC builds the following reduced model[4]

$$\left\{ \begin{array}{ll} \partial_t u_0 + \mathbf{v} \cdot \nabla_{\mathbf{x}} u_0 & = -v \displaystyle\int \left(\sigma_t \sum_{k \le P} u_k \phi_k \right) \phi_0 \, \mathrm{d}\mathcal{P}_X + v \iint \left(\left(\sigma_s \sum_{k \le P} u_k \phi_k \right) \phi_0 \, \mathrm{d}\mathcal{P}_X \right) \mathrm{d}\mathbf{v}', \\ \dots & \dots \\ \partial_t u_P + \mathbf{v} \cdot \nabla_{\mathbf{x}} u_P & = -v \displaystyle\int \left(\sigma_t \sum_{k \le P} u_k \phi_k \right) \phi_P \, \mathrm{d}\mathcal{P}_X + v \iint \left(\left(\sigma_s \sum_{k \le P} u_k \phi_k \right) \phi_P \, \mathrm{d}\mathcal{P}_X \right) \mathrm{d}\mathbf{v}'. \end{array} \right. \tag{3}$$

System (3) is built from (1) by performing a Galerkin projection of (1) onto the $P-$truncated orthonormal gPC basis[5] $(\phi_k)_{k \in \{0,...P\}}$, see [1, 2]. This system is linear and is solved with an MC

[1] It may be decomposed into $\mathbf{v} = v\omega$ where $v = |\mathbf{v}| \in \mathbb{R}^+$ and $\omega = \frac{\mathbf{v}}{v} \in \mathbb{S}^2$.

[2] It is always possible to come back to such framework, at the cost of more or less tedious pretreatments leading to a controled approximation [11, 12, 13] and decorrelation [14, 15].

[3] geometrical, in the cross-sections, in the multiplicity, in the boundary conditions etc.

[4] We drop the dependencies, initial and boundary conditions for convenience.

[5] $(\phi_k)_{k \in \mathbb{N}}$ is orthonormal with respect to the scalar product defined by the measure $\mathrm{d}\mathcal{P}_{\mathbf{X}}$ of the input uncertain

scheme. MC-gPC allows important computational gains on this model[6] on many[7] applications. But MC-gPC also presents some weaknesses. Let us summarize the pros and cons of MC-gPC when applied to equation (1). The next points can be considered a summary of what can be found in [1, 2, 8, 9, 10]. The main benefits of MC-gPC for (1) are

- ($\checkmark_1$) spectral convergence: it corresponds to the fast convergence with respect to P (proved in [2]).
- ($\checkmark_2$) Only simple modifications of an existing MC code solving (1) are needed to solve (3), see [1].
- ($\checkmark_3$) The fact it is based on modifications of an existing MC code makes the MC-gPC solver inherit the parallel strategies implemented in the native MC code solving (1), see [1, 8, 9].
- ($\checkmark_4$) Finally, it is also efficient when applied to nonlinear physics [9, 8].

Now, the main drawbacks of MC-gPC when applied to (1) are

- ($*_1$) the curse of dimensionality: the size of the system $(P+1)$ grows exponentially fast with the number (Q) of uncertain parameters[8]. It is commonly accepted that gPC reduced models can handle $Q \sim 1-10$ but not much more (without additional *a priori* knowledge, see [38]).
- ($*_2$) This curse of dimensionality has an impact on different phases/characteristics of the MC-gPC computations:
  - (a) each MC particle must tally its contribution into an array of size $P+1$, see [1].
  - (b) The parallel reductions on these same arrays can consequently be costly too [1].
  - (c) Nonlinear physics needing small time steps/many iterations may not benefit from an MC-gPC implementation if the tallies/parallel reductions are too frequent, see [8, 9].
  - (d) It also has an impact on the memory consumption: each MC particles carries a field $\mathbf{X}_p$ of size Q in the MC schemes described in [1, 8, 9, 10].
- ($*_3$) Finally, it has recently been proved [10] that the MC schemes solving (3) and described in [1, 8, 9] are noisier than other strategies (even if the noise remains acceptable).

The aim of this paper is to present new MC schemes solving system (3) *and improving drawback* ($*_3$) *above, i.e. reducing the noise in the MC-gPC computations.* In practice, this is done in the following sections by suggesting new ways to discretise system (3) with MC schemes. The new MC schemes are multigroup-like resolution methods. They are much less noisy than the original MC-gPC implementation presented in [1, 2, 8, 9, 10]. Besides, by alleviating drawback ($*_3$), we also improve the behaviour of MC-gPC with respect to drawbacks ($*_2$)-a-b-c leading to faster run-times. But, on another hand, benefit ($\checkmark_2$) will not hold anymore: the modifications one must perform to an existing MC code in order to implement the new versions of MC-gPC are not that simple anymore even if people familiar with multigroup MC resolution should not be intimidated. Still, we consider these less simple modifications are worth it, the reasons why are emphasized in the next sections. The paper ends with a discussion on thinking about uncertainty propagation at early stage of the development of a code, intrusive methods and the possibility to perform some hybrid

---

parameters, i.e such that $\int \phi_k(\mathbf{X})\phi_l(\mathbf{X})\,\mathrm{d}\mathcal{P}_{\mathbf{X}} = \delta_{k,l}, \forall (k,l) \in \mathbb{N}^2$. Note that in practice, this basis is built once and for all once $\mathrm{d}\mathcal{P}_{\mathbf{X}}$ known.

[6] Ranging from $\times 7$ to $\times 50$ accelerations of the run-times with respect to classical non-intrusive methods.

[7] Linear [1, 4] or nonlinear [5, 6, 8, 9].

[8] The multivariate polynomial basis is built by tensorization of one-dimensional polynomial basis in every stochastic direction $(X_i)_{i\in\{1,\dots,Q\}}$. In each direction, the basis must be truncated up to certain orders $(p_i)_{i\in\{1,\dots,Q\}}$ which may depend on the directions $(X_i)_{i\in\{1,\dots,Q\}}$. Assume that $\forall i \in \{1,..,Q\}$, $p_i = p_{1\mathrm{D}}$, then the total number of polynomial coefficients, abusively called the polynomial order later on, is $P = P(p_{1\mathrm{D}}, Q) = (p_{1\mathrm{D}}+1)^Q$. It exhibits an exponential growth with respect to both $p_{1\mathrm{D}}$ and Q.

intrusive/non-intrusive uncertainty analysis from, for example but without loss of generality, the code developed for this paper.

The paper is organized as follows: section 2 recalls the main results of [10] concerning the fact that MC-gPC is noisier than other strategies. In particular, the factor responsible for the excess of variance of MC-gPC is identified. In section 3, we describe the new MC resolution strategies: they are based on multigroup semi-analog and non-analog MC resolutions of (3). We also insist in this section on how relying to this framework allows avoiding the aforementioned excess of variance. Section 4 is devoted to numerical examples with several uncertainty propagations, the resolution of an uncertain eigenvalue problem and some hydrid non-intrusive/intrusive computations. Section 5 is a concluding paragraph.

2. Forewords: the need to reduce the variance of MC-gPC

In this section, we recall few results from [1, 10] and deepen the analysis. The aim is to identify which terms are responsible for MC-gPC having lesser performances than ni-gPC[9] in terms of numerical noise and suggest corrections allowing to overcome drawback $(*_3)$ of section 1.

The gPC based reduced models need the numerical approximation of coefficients $(u_k)_{k\in\{0,...,P\}}$ appearing in the gPC development of u the solution of (1) onto the gPC basis $(\phi_k)_{k\in\{0,...,P\}}$. In other words u is approximated by the $P-$truncated expansion

$$u(\mathbf{x},t,\mathbf{v},\mathbf{X}) \approx u^P(\mathbf{x},t,\mathbf{v},\mathbf{X}) = \sum_{k=0}^{P} u_k(\mathbf{x},t,\mathbf{v})\phi_k(\mathbf{X}).$$

The above polynomial approximation bears some interesting convergence properties [39, 40, 41] as $P \to \infty$. Spectral (i.e. fast) convergence for the solution of equation (1) has even been proved in [2].

Now, there exists several ways to compute the previous coefficients. There are two main class of methods:

– non-intrusive gPC (ni-gPC) [42, 43, 38, 44, 45, 46, 47] consists in introducing the set of points/weights $(\mathbf{X}_i, w_i)_{i\in\{1,...,N\}}$ approximating random vector $\mathbf{X}$ and its probability measure $\mathrm{d}\mathcal{P}_{\mathbf{X}}$. The gPC coefficients of u are consequently recovered by numerical integration: $\forall k \in \{0, ..., P\}$

$$u_k(\mathbf{x},t,\mathbf{v}) \quad = \int u(\mathbf{x},t,\mathbf{v},\mathbf{X})\phi_k(\mathbf{X})\,\mathrm{d}\mathcal{P}_{\mathbf{X}} = \sum_{i=1}^{N} u(\mathbf{x},t,\mathbf{v},\mathbf{X}_i)\phi_k(\mathbf{X}_i)w_i + \mathcal{O}(N^{\beta}), \text{ with } \beta < 0. \quad (4)$$

The numerical integration error strongly depends on the choice of the experimental design $(\mathbf{X}_i, w_i)_{i\in\{1,...,N\}}$.

Independently of the choice of the experimental design, ni-gPC implies performing N runs of a code solving (1) in order to gather $(u(\mathbf{x},t,\mathbf{v},\mathbf{X}_i), w_i)_{i\in\{1,...,N\}}$ and post-treat them in order to build the gPC coefficients as in (4). In our MC resolution context, each run needs N_{MC}

[9] Which asymptotically presents the best performances in terms of numerical noise, see [10].

particles such that u is approximated as $u(\mathbf{x},t,\mathbf{v},\mathbf{X}_i) = u^{N_{MC}}(\mathbf{x},t,\mathbf{v},\mathbf{X}_i) + \mathcal{O}(N_{MC}^{-\frac{1}{2}})$, where $u^{N_{MC}}$ is the code output. This means that we have $\forall k \in \{0,...,P\}$

$$\begin{aligned} u_k(\mathbf{x},t,\mathbf{v}) &= \sum_{i=1}^{N} u^{N_{MC}}(\mathbf{x},t,\mathbf{v},\mathbf{X}_i)\phi_k(\mathbf{X}_i)w_i + \mathcal{O}(N^{\beta}) + \mathcal{O}(N_{MC}^{-\frac{1}{2}}), \\ &= \sum_{i=1}^{N} u^{N_{MC}}(\mathbf{x},t,\mathbf{v},\mathbf{X}_i)\phi_k(\mathbf{X}_i)w_i + \mathcal{O}(N^{\beta}) + \frac{\sigma_k(\mathbf{x},t,\mathbf{v})}{\sqrt{N_{MC}}}\mathcal{G}_k. \end{aligned} \tag{5}$$

In the above expression, $(\mathcal{G}_k)_{k\in\{0,...,P\}}$ are gaussian random variables of mean zero and variance one, see [10]. From the residual terms in (5), we can see that the experimental design for the uncertain variable $\mathbf{X}$ is tensorised with the one for the physical variables $(\mathbf{x},t,\mathbf{v})$ with an accuracy which is $\mathcal{O}(N^{\beta}) + \mathcal{O}(N_{MC}^{-\frac{1}{2}})$. Both N and N_{MC} must grow in order to converge. MC-gPC has been introduced in [1] mainly because a full MC experimental design on the whole set of variables $(\mathbf{x},t,\mathbf{v},\mathbf{X})$ can avoid this tensorization.

– MC-gPC allows integrating the gPC coefficients *on-the-fly* during MC resolution of system (3). The details are in [1] and briefly recalled in Appendix C. With MC-gPC, we have $\forall k \in \{0,...,P\}$

$$\begin{aligned} u_k(\mathbf{x},t,\mathbf{v}) &= u_k^{N_{MC}}(\mathbf{x},t,\mathbf{v}) + \mathcal{O}(N_{MC}^{-\frac{1}{2}}), \\ &= u_k^{N_{MC}}(\mathbf{x},t,\mathbf{v}) + \frac{\sigma_{k,\mathrm{MC}}(\mathbf{x},t,\mathbf{v})}{\sqrt{N_{MC}}}\mathcal{G}_k. \end{aligned} \tag{6}$$

Once again, $\mathcal{G}_k$ are gaussian random variables of mean zero and variance one, see [10]. In (6), the convergence only depends on N_{MC} which can be considered an advantage.

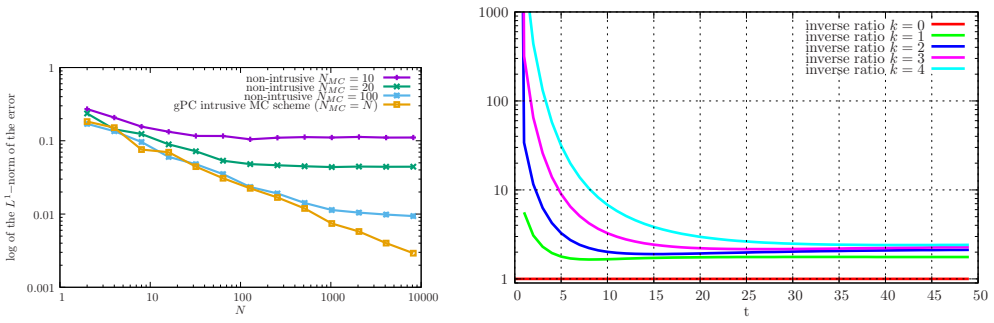

Figure 1: Left convergence studies with respect to N and N_{MC} for ni-gPC and MC-gPC on the variance of the number of physical particles. Right: evolution $\rightarrow \frac{\sigma^2_{k,\text{MC-sa}}(t)}{\sigma^2_{k,\text{sa}}(t)}$ of the ratio of variances between MC-gPC and ni-gPC.

Figure 1-left illustrates the behaviours summed-up in expressions (5) and (6). It presents some convergence curves in a very simple configuration (see Appendix A for all the details) for the two above numerical methods:

– the results obtained by ni-gPC use a deterministic black-box code solved by an MC scheme (the semi-analog one) of discretisation parameter N_{MC}. The uncertain counterpart is solved

with an MC sampling of $(X, \mathrm{d}\mathcal{P}_X)$ with N points. Three plots are displayed, corresponding to three convergence studies with respect to N for fixed values of $N_{MC} = 10, 20, 100$. Every curve obtained with ni-gPC presents, first, a converging behaviour with a slope which is characteristic of the numerical method used in order to integrate the uncertainties, i.e. here $\mathcal{O}(N^{-\frac{1}{2}})$. Then, the curves present a more or less pronounced kink: a change of slope, followed by a plateau, a stagnation of the accuracy. It corresponds to the point where the general accuracy becomes driven by the *coarser* numerical method, here $\mathcal{O}(N_{MC}^{-\frac{1}{2}})$. Increasing N (relative to the $x-$axis) does not allow any significant gain as $\mathcal{O}(N^{\beta}) = \mathcal{O}(N^{-\frac{1}{2}}) \ll 1$. In a sense, the locations of the kinks corresponds to optimal parameter choices (N_{MC}, N): increasing the accuracy in one direction without the other induces a loss of computational time. Finding these optimal parameters can be difficult in practice. But recent papers, aiming at balancing the different types of errors, are certainly promising in that direction, see [48].

– For MC-gPC, the behaviour is quite different and is described by (6). For MC-gPC $N = N_{MC}$, i.e. the experimental design is not anymore tensorized with the MC particles. The approximation obtained with the new MC scheme does not stagnate with the increasing number of samplings. The uncertainty is solved *on-the-fly* during the MC resolution and the convergence rate for the whole problem remains $\mathcal{O}(\frac{1}{\sqrt{N_{MC}}})$ avoiding the kinks in the curves obtained non-intrusively.

Note that on figure 1-left, the slope for MC-gPC and the slope for the finer ni-gPC approximation are the same, almost overimposed (at least for $N \in \{1, 100\}$) before the loss of convergence of ni-gPC (for $N > 100$). This typically means that their performances in terms of numerical error/variance are equivalent. But in [10], situations in which the variance of ni-gPC is way smaller than the one of MC-gPC have been identified: let us consider the simple (monokinetic and homogeneous) problem (already detailed in [10]) of Appendix B. For this problem, the asymptotical variances on the gPC coefficients of ni-gPC and MC-gPC have been analytically calculated. Let us consider the case in which the MC codes are based on a semi-analog (sa) MC scheme (see [10]). Then the asymptotical variances are given by:

– for ni-gPC [10], $\sigma^2_{k,\mathrm{sa}}$ as in (5) is given by $\forall k \in \{0, ..., P\}$

$$\begin{aligned} \sigma^2_{k,\mathrm{sa}} &= \int \sigma^2_{\mathrm{sa}}(t, \mathbf{X}) \phi_k(\mathbf{X}) \, \mathrm{d}\mathcal{P}_{\mathbf{X}}, \forall k \in \{0, ..., P\}, \\ &= \int U_0^2(\mathbf{X}) \left(e^{v \frac{\sigma_s^2(\mathbf{X}) - \sigma_t^2(\mathbf{X})}{\sigma_t(\mathbf{X})} t} - e^{2v(\sigma_s(\mathbf{X}) - \sigma_t(\mathbf{X}))t} \right) \phi_k(\mathbf{X}) \, \mathrm{d}\mathcal{P}_{\mathbf{X}}. \end{aligned} \tag{7}$$

– On another hand [10], with MC-gPC, $\sigma^2_{k,\text{MC-sa}}$ as in (6) is given by $\forall k \in \{0, ..., P\}$

$$\sigma^2_{k,\text{MC-sa}}(t) = \int U_0^2(\mathbf{X}) \phi_k^2(\mathbf{X}) e^{v \frac{\sigma_s^2(\mathbf{X}) - \sigma_t^2(\mathbf{X})}{\sigma_t(\mathbf{X})} t} \, \mathrm{d}\mathcal{P}_{\mathbf{X}} - \left[\int U_0(\mathbf{X}) e^{-v\sigma_a(\mathbf{X})t} \phi_k(\mathbf{X}) \, \mathrm{d}\mathcal{P}_{\mathbf{X}} \right]^2. \tag{8}$$

Figure 1-right displays $t \to \frac{\sigma^2_{k,\text{MC-sa}}(t)}{\sigma^2_{k,\mathrm{sa}}(t)}$ which is directly proportional to $\frac{N_{MC}^{\text{ni-gPC}}}{N_{MC}^{\text{MC-gPC}}}$. This ratio gives an idea of the number of MC particles needed by MC-gPC in order to reach the same amplitude of numerical noise as ni-gPC in the same conditions (but still assuming that $N \gg 1$). On figure 1-right, it is first possible to see that for $k = 0$, i.e. for the mean, the performances of MC-gPC

and ni-gPC are equivalent: the ratio is equal to 1. For all the other gPC coefficients, MC-gPC exhibits an excess of variance with respect to ni-gPC. The higher the order of the coefficient, the higher the excess of variance. Now, depending on the time of interest, this excess of variance is not necessarily dramatic: for late times, i.e for $t > 25$, the performances of MC-gPC and ni-gPC are almost equivalent. This corresponds to what is observed on figure 1-left: the convergence curves were computed in such a regime. But for early times in figure 1-right, we can see that for this problem, MC-gPC may need from $\times 10$ to more than $\times 1000$ more MC particles in order to reach the same accuracy as ni-gPC. For these early times, due to the configuration of interest described in (B.3), the solution is almost deterministic (i.e. not uncertain). In this paper, we aim at overcoming this problem: we build MC schemes for MC-gPC which lead to lesser or equivalent numerical errors on each gPC coefficients as ni-gPC (which corresponds to the less noisy strategy, see [10], even if not the less computationally intensive, see [1]).

For this, we need to have an idea of which terms in (8) is responsible for the excess of variance with respect to (7): in the example of figure 1-right, the earliest the time, the closer to a deterministic (i.e. without uncertainties) problem. Let us see what expression (7) and (8) become when the different quantities are independent of $\mathbf{X}$ (i.e. deterministic). We obtain for ni-gPC

$$\begin{aligned}
\sigma^2_{k,\text{sa}}(t) &= \int U_0^2(\mathbf{X}) \left(e^{v\frac{\sigma_s^2(\mathbf{X})-\sigma_t^2(\mathbf{X})}{\sigma_t(\mathbf{X})}t} - e^{2v(\sigma_s(\mathbf{X})-v\sigma_t(\mathbf{X}))t} \right) \phi_k(\mathbf{X})\, \mathrm{d}\mathcal{P}_{\mathbf{X}}, \forall k \in \{0,...,P\}, \\
&= U_0^2 \left(e^{v\frac{\sigma_s^2-\sigma_t^2}{\sigma_t}t} - e^{2v(\sigma_s-\sigma_t)t} \right) \int \phi_k(\mathbf{X})\, \mathrm{d}\mathcal{P}_{\mathbf{X}}, \forall k \in \{0,...,P\}, \\
&= U_0^2 \left(e^{v\frac{\sigma_s^2-\sigma_t^2}{\sigma_t}t} - e^{2v(\sigma_s-\sigma_t)t} \right) \delta_{k,0}, \forall k \in \{0,...,P\}.
\end{aligned} \tag{9}$$

With ni-gPC, in the deterministic case, the numerical error on the high order gPC coefficients is zero. For MC-gPC on another hand, in exactly the same conditions, we get

$$\begin{aligned}
\sigma^2_{k,\text{MC-sa}}(t) &= \int U_0^2(\mathbf{X}) \phi_k^2(\mathbf{X}) e^{v\frac{\sigma_s^2(\mathbf{X})-\sigma_t^2(\mathbf{X})}{\sigma_t(\mathbf{X})}t}\, \mathrm{d}\mathcal{P}_{\mathbf{X}} - \left[\int U_0(\mathbf{X}) e^{-v\sigma_a(\mathbf{X})t} \phi_k(\mathbf{X})\, \mathrm{d}\mathcal{P}_{\mathbf{X}} \right]^2, \\
&= U_0^2 e^{v\frac{\sigma_s^2-\sigma_t^2}{\sigma_t}t} \int \phi_k^2(\mathbf{X})\, \mathrm{d}\mathcal{P}_{\mathbf{X}} - \left[U_0 e^{-v\sigma_a t} \int \phi_k(\mathbf{X})\, \mathrm{d}\mathcal{P}_{\mathbf{X}} \right]^2, \\
&= U_0^2 \left(e^{v\frac{\sigma_s^2-\sigma_t^2}{\sigma_t}t} 1 - e^{2v(\sigma_s-\sigma_t)t} \delta_{k,0} \right).
\end{aligned} \tag{10}$$

In the above expression the ϕ_k^2 factor makes term 1 appear independently of the order k of the gPC coefficient. It leads to non-zero numerical errors on these quantities even in the deterministic case. The appearance of such factor is closely related to the explicit use, within the MC estimator (see the red terms in algorithm 6 of Appendix C), of the gPC basis $(\phi_k)_{k\in\{0,...,P\}}$ in order to tally the contributions of every MC particles.

From the previous analysis, we have an idea of what we must do in order to solve (3) with better performances in terms of noise: we need to get rid of the $(\phi_k)_{k\in\{0,...,P\}}$ term during the tallying phase (see the $+=$ operation in algorithm 6). Multigroup MC schemes are specifically designed to avoid this. In the next section, we present how we can solve (3) with a multigroup-like approach and verify it can deal with the previously identified spurious term.

3. New MC discretisations of reduced model (3)

As explained in the previous sections, MC-gPC discretises the following system

$$\left\{\begin{array}{ll} \partial_t u_0 + \mathbf{v}\cdot\nabla_{\mathbf{x}} u_0 & = -v\displaystyle\int\left(\sigma_t\sum_{k\leq P}u_k\phi_k\right)\phi_0\,\mathrm{d}\mathcal{P}_X + v\iint\left(\left(\sigma_s\sum_{k\leq P}u_k\phi_k\right)\phi_0\,\mathrm{d}\mathcal{P}_X\right)\,\mathrm{d}\mathbf{v}', \\ \dots & \dots \\ \partial_t u_P + \mathbf{v}\cdot\nabla_{\mathbf{x}} u_P & = -v\displaystyle\int\left(\sigma_t\sum_{k\leq P}u_k\phi_k\right)\phi_P\,\mathrm{d}\mathcal{P}_X + v\iint\left(\left(\sigma_s\sum_{k\leq P}u_k\phi_k\right)\phi_P\,\mathrm{d}\mathcal{P}_X\right)\,\mathrm{d}\mathbf{v}', \end{array}\right. \tag{11}$$

with an MC scheme. One MC scheme has been designed in [1], but it is not unique. One of the main objective of the MC resolution in [1] was to be able to implement MC-gPC with minimal modifications of an already existing MC code. As such, MC-gPC allows considerable gains on many problems [1, 8, 9]. But on another hand, this MC scheme leads to undesirable excess of variances on the high-order[10] gPC coefficients, especially in the deterministic case (see the need for more MC particles for early times in figure 1-right).

Spectral convergence with respect to P holds for system (11). We want to keep this property so we are going to keep trying to solve system (11). On another hand, we are going to try to improve its MC resolution[11]. We here build new MC schemes which, we will see later on, have better performances in terms of variances/errors $(\sigma^2_{k,\mathrm{MC}})_{k\in\{0,\dots,P\}}$ on the gPC coefficients $(u_k)_{k\in\{0,\dots,P\}}$ and even better run-times.

3.1. Building a multigroup structure for system (11)

In order to build new MC schemes for system (11), let us express it differently. Let us expand the collisional terms as

$$\left\{\begin{array}{ll} \partial_t u_0 + \mathbf{v}\cdot\nabla_{\mathbf{x}} u_0 & = -v\displaystyle\sum_{k\leq P}u_k\int\sigma_t\phi_k\phi_0\,\mathrm{d}\mathcal{P}_X + v\int\sum_{k\leq P}u_k\int\sigma_s\phi_k\phi_0\,\mathrm{d}\mathcal{P}_X\,\mathrm{d}\mathbf{v}', \\ \dots & \dots \\ \partial_t u_P + \mathbf{v}\cdot\nabla_{\mathbf{x}} u_P & = -v\displaystyle\sum_{k\leq P}u_k\int\sigma_t\phi_k\phi_P\,\mathrm{d}\mathcal{P}_X + v\int\sum_{k\leq P}u_k\int\sigma_s\phi_k\phi_P\,\mathrm{d}\mathcal{P}_X\,\mathrm{d}\mathbf{v}'. \end{array}\right. \tag{12}$$

Then we can rewrite (12) in a vectorial form as

$$\partial_t U(\mathbf{x},t,\mathbf{v}) + \mathbf{v}\cdot\nabla_{\mathbf{x}} U(\mathbf{x},t,\mathbf{v}) = -v\Sigma_t(\mathbf{x},\mathbf{v})U(\mathbf{x},t,\mathbf{v}) + v\int\Sigma_s(\mathbf{x},\mathbf{v}\cdot\mathbf{v}')U(\mathbf{x},t,\mathbf{v}')\,\mathrm{d}\mathbf{v}', \tag{13}$$

in which $U(\mathbf{x},t,\mathbf{v}) = (u_0(\mathbf{x},t,\mathbf{v}),...,u_P(\mathbf{x},t,\mathbf{v}))^t$. The new system (13) does look like a multigroup transport equation, see [16, 18, 33]. But it is not one: this is mainly due to the expression of the

[10] i.e. $k > 0$, see [10].

[11] Several MC schemes exists. For example, for the deterministic linear Boltzmann equation, several MC schemes exists: the semi-analog and the non-analog MC schemes are both intensively used in many applications (see [49, 27, 28, 21, 22, 23] for semi-analog based solvers and [50, 24, 25, 51, 52, 53, 29] for non-analog based ones).

cross-section Σ_t. Let us explicit the expressions of the Σ_t and Σ_s:

$$\Sigma_\alpha(\mathbf{x},\mathbf{v}) = \begin{pmatrix} ... & ... & ... \\ ... & \displaystyle\int \sigma_\alpha(\mathbf{x},\mathbf{v},\mathbf{X})\phi_k(\mathbf{X})\phi_l(\mathbf{X})\,\mathrm{d}\mathcal{P}_{\mathbf{X}} & ... \\ ... & ... & ... \end{pmatrix} \text{ for } \alpha \in \{s,t\}. \tag{14}$$

The previous matrices Σ_t, Σ_s are (potentially) full symetric matrices. For (13) to be rewritten as a multigroup transport system, Σ_t needs to be diagonal with positive coefficients. Let us introduce R such that

$$\Sigma_t(\mathbf{x},\mathbf{v}) = R(\mathbf{x},\mathbf{v})\Lambda_t(\mathbf{x},\mathbf{v})R^{-1}(\mathbf{x},\mathbf{v}),$$

with Λ_t a diagonal matrix built from the positive eigenvalues of Σ_t. Then (13) can be rewritten

$$\begin{aligned}\partial_t\left[R^{-1}(\mathbf{x},\mathbf{v})U(\mathbf{x},t,\mathbf{v})\right] + R^{-1}(\mathbf{x},\mathbf{v})\mathbf{v}\cdot\nabla_{\mathbf{x}}U(\mathbf{x},t,\mathbf{v}) = \quad & -v\Lambda_t(\mathbf{x},\mathbf{v})R^{-1}(\mathbf{x},\mathbf{v})U(\mathbf{x},t,\mathbf{v}) \\ & +vR^{-1}(\mathbf{x},\mathbf{v})\int \Sigma_s(\mathbf{x},\mathbf{v}\cdot\mathbf{v}')U(\mathbf{x},t,\mathbf{v}')\,\mathrm{d}\mathbf{v}'.\end{aligned} \tag{15}$$

By performing the change of variable $V(\mathbf{x},t,\mathbf{v}) = R^{-1}(\mathbf{x},\mathbf{v})U(\mathbf{x},t,\mathbf{v})$, we do obtain a vectorial term $\Lambda_t(\mathbf{x},t,\mathbf{v})$ in factor of $V(\mathbf{x},t,\mathbf{v})$ leading to:

$$\begin{aligned}\partial_t V(\mathbf{x},t,\mathbf{v}) + R^{-1}(\mathbf{x},\mathbf{v})\mathbf{v}\cdot\nabla_{\mathbf{x}}U(\mathbf{x},t,\mathbf{v}) = \quad & -v\Lambda_t(\mathbf{x},\mathbf{v})V(\mathbf{x},t,\mathbf{v}) \\ & +v\int \underbrace{R^{-1}(\mathbf{x},\mathbf{v})\Sigma_s(\mathbf{x},\mathbf{v}\cdot\mathbf{v}')R(\mathbf{x},\mathbf{v}')}_{\Lambda_s(\mathbf{x},\mathbf{v}\cdot\mathbf{v}')}V(\mathbf{x},t,\mathbf{v}')\,\mathrm{d}\mathbf{v}'.\end{aligned} \tag{16}$$

The next step concerns the term $R^{-1}(\mathbf{x},\mathbf{v})\mathbf{v}\cdot\nabla_{\mathbf{x}}U(\mathbf{x},t,\mathbf{v})$ as, at first glance, R^{-1} seems to need to be linearised for V to appear under the spatial operator $(\nabla_{\mathbf{x}})$. Assume that domain $\mathcal{D}$ is tesselated into $N_{\mathcal{D}}$ cells, i.e. we have $\mathcal{D} = \cup_{i=1}^{N_{\mathcal{D}}}\mathcal{D}_i$. Assume furthermore that the cross sections are constant per cell with respect to $\mathbf{x}$ so that we have

$$\Sigma_t(\mathbf{x},\mathbf{v}) = \sum_{i=1}^{N_{\mathcal{D}}} \Sigma_t^i(\mathbf{v})\mathbf{1}_{\mathcal{D}_i}(\mathbf{x}).$$

Then it is possible to diagonalise Λ_t^i, $\forall i \in \{1,...,N_{\mathcal{D}}\}$. This leads to having

$$\Sigma_t(\mathbf{x},\mathbf{v}) = \sum_{i=1}^{N_{\mathcal{D}}} R_i(\mathbf{v})\Lambda_t^i(\mathbf{v})R_i^{-1}(\mathbf{v})\mathbf{1}_{\mathcal{D}_i}(\mathbf{x}).$$

From this point on, two situations can occur:

1. either $(\Sigma_t^i(\mathbf{v}))_{i\in\{1,...,N_{\mathcal{D}}\}}$ are all diagonalisable in the same basis $\forall i \in \{1,...,N_{\mathcal{D}}\}$. This means that we have $R_i^{-1}(\mathbf{v}) = R^{-1}(\mathbf{v})$, $\forall i \in \{1,...,N_{\mathcal{D}}\}$. In this case, $R^{-1}(\mathbf{v})\mathbf{v}\cdot\nabla_{\mathbf{x}}U(\mathbf{x},t,\mathbf{v}) = \mathbf{v}\cdot\nabla_{\mathbf{x}}(R^{-1}(\mathbf{v})U(\mathbf{x},t,\mathbf{v})) = \mathbf{v}\cdot\nabla_{\mathbf{x}}V(\mathbf{x},t,\mathbf{v})$ which considerably eases the analysis. Unfortunately, it is easy finding configurations of interest in which the matrices $(\Sigma_t^i)_{i\in\{1,...,N_{\mathcal{D}}\}}$ are not diagonalisable in the same basis (see section 4.4).
2. Or $(\Sigma_t^i(\mathbf{v}))_{i\in\{1,...,N_{\mathcal{D}}\}}$ are not diagonalisable in the same basis and, if we introduce $R^{-1}(\mathbf{x},\mathbf{v}) = \sum_{i=1}^{N_{\mathcal{D}}} R_i^{-1}(\mathbf{v})\mathbf{1}_{\mathcal{D}_i}(\mathbf{x})$, we have

$$R^{-1}(\mathbf{x},\mathbf{v})\mathbf{v}\cdot\nabla_{\mathbf{x}}U(\mathbf{x},t,\mathbf{v}) \quad = \mathbf{v}\cdot\nabla_{\mathbf{x}}\left(R^{-1}(\mathbf{x},\mathbf{v})U(\mathbf{x},t,\mathbf{v})\right) + S(\mathbf{x},t,\mathbf{v}).$$

In the above expression, S can be expressed as

$$\begin{aligned} S(\mathbf{x},t,\mathbf{v}) &= -\mathbf{v}\cdot\nabla_{\mathbf{x}}R^{-1}(\mathbf{x},\mathbf{v})U(\mathbf{x},t,\mathbf{v}),\\ &= -\mathbf{v}\cdot\nabla_{\mathbf{x}}R^{-1}(\mathbf{x},\mathbf{v})R(\mathbf{x},\mathbf{v})V(\mathbf{x},t,\mathbf{v}), \end{aligned}$$

With the constant per cell hypothesis of R, R^{-1} and the presence of $\nabla_{\mathbf{x}}R^{-1}$ in S, such a term induces the appearance of jumps and of Dirac masses across cells. These jumps and Dirac masses can easily be dealt with in an MC resolution (see for example [54] even if a different strategy is applied in this paper). But still, they surely add complexity to the implementation and their treatment is the purpose of a whole paragraph of section 3 (see section 3.7).

Now, assume that the uncertain total cross-section matrix $\mathbf{x},\mathbf{v}\to\Sigma_t(\mathbf{x},\mathbf{v})$ is pre-assembled and diagonalised (per cell or per material) offline before any calculations. We consider that matrices $\mathbf{x},\mathbf{v}\to R(\mathbf{x},\mathbf{v}), \mathbf{x},\mathbf{v}\to R^{-1}(\mathbf{x},\mathbf{v})$ and $\mathbf{x},\mathbf{v}\to\nabla_{\mathbf{x}}R^{-1}(\mathbf{x},\mathbf{v})$ are known at the beginning of the study $\forall\mathbf{x}\in\mathcal{D},\forall\mathbf{v}\in\mathcal{V}$. This means we can rewrite (15) as *the multigroup-like* system of equations

$$\begin{aligned} \partial_t V(\mathbf{x},t,\mathbf{v})+\mathbf{v}\cdot\nabla_{\mathbf{x}}V(\mathbf{x},t,\mathbf{v}) = &+\left[\mathbf{v}\cdot\nabla_{\mathbf{x}}R^{-1}(\mathbf{x},\mathbf{v})R(\mathbf{x},\mathbf{v})\right]V(\mathbf{x},t,\mathbf{v})\\ &-v\Lambda_t(\mathbf{x},\mathbf{v})V(\mathbf{x},t,\mathbf{v})\\ &+v\int\Lambda_s(\mathbf{x},\mathbf{v}\cdot\mathbf{v}')V(\mathbf{x},t,\mathbf{v}')\,\mathrm{d}\mathbf{v}'. \end{aligned} \tag{17}$$

In the next sections, we build MC schemes solving system (17).

Remark 3.1. *Of course, if the cross-sections are constant per cell/material with respect to* $\mathbf{x}$*,* $\nabla_{\mathbf{x}}R^{-1}$ *is non-zero only at the interfaces between cells/materials and zero inside each cell/material. Introduce the notation* $\overline{\mathcal{D}}=\mathcal{D}\setminus\partial\mathcal{D}$ *in order to denote the interior of* $\mathcal{D}$*. Then we have*

- $\nabla_{\mathbf{x}}R^{-1}(\mathbf{x},\mathbf{v})=0,\forall\mathbf{x}\in\overline{\mathcal{D}}_i,\forall i\in\{1,...,N_{\mathcal{D}}\},\forall\mathbf{v}\in\mathcal{V}$*. The treatment within the interior of each cell is described in sections 3.4–3.5–3.6.*
- $\nabla_{\mathbf{x}}R^{-1}(\mathbf{x},\mathbf{v})\neq 0,\forall\mathbf{x}\in\mathcal{D}_i\cap\mathcal{D}_j,\forall(i,j)\in\{1,...,N_{\mathcal{D}}\}^2$ *for each interface between different materials. The treatment of this term at each interface between two (different) materials is fully detailed in section 3.7.*

The two above cases could be presented together but we think separating them can ease the description of the different MC schemes. There are many ways to deal with the non-zero term of the second above bullet. The one described in section 3.7 is different from the one of paper [54] and can be considered original.

Remark 3.2. *Note that under mild assumptions on* $\mathbf{X}$*,* $\Lambda_s(\mathbf{x},\mathbf{v},\mathbf{v}')=R^{-1}(\mathbf{x},\mathbf{v})\Sigma_s(\mathbf{x},\mathbf{v},\mathbf{v}')R(\mathbf{x},\mathbf{v})$ *may even be diagonal. Besides, the group scattering term related to* Λ_s *may have negative coefficients. Note also that for* k_{eff} *computations,* Λ_s *may change at each iteration, see section 4.6.*

Remark 3.3. *In the next sections, we assume that the cross-sections* Λ_s,Λ_t *are available. They can, for example, be pre-processed (with an accurate integration with respect to* $\mathrm{d}\mathcal{P}_{\mathbf{X}}$*) in a pretreatment code. Note that building physical data from pretreatment codes is common: in neutronics for example, Galilee [55], NJOY [56], CALENDF [57] or PREPRO [58] pre-process evaluated data [59] to make them usable in deterministic or MC resolution codes. From now on, we assume that the pretreated cross-sections* Λ_s,Λ_t *together with matrices* R,R^{-1} *and* $\mathbf{v}\cdot\nabla_{\mathbf{x}}R^{-1}$ *are available at the beginning of the computations.*

3.2. The multigroup-like MC schemes allowing to to solve (17)

Equation (17) has now exactly the same structure as a classical multigroup transport equation [16, 18, 33]. It can be solved applying the classical MC schemes, such as the semi-analog[12] MC scheme [33, 60, 49, 27, 28] or the non-analog one [33, 60, 50, 24, 25, 51, 52, 53, 29]. Readers familiar with such resolution technics can easily skip the next sections. Still, we include the next paragraphs despite the fact that this type of solvers are quite classical [16, 18, 17, 33] for three reasons. First, for the sake of reproducibility of the result of the paper. Second, because of the singular structure of the group scattering term Λ_s which can (legitimately) have negative coefficients and leads to treatments at the collision (see (22) and algorithm 4) which may be considered original. Third, because the special treatment at the interface between two different cells/materials induced by the term $\mathbf{v}\cdot\nabla_{\mathbf{x}}R^{-1}RV$ needs additional explainations (even if the treatment implied by this term is very similar to the one for Λ_s, see section 4.4).

All the next steps aim at applying the material of [33] and more precisely theorem 3.2.1 which ensures the MC schemes we build are unbiased converging ones.

3.3. Rewriting system (17) *as a scalar equation*

The first step in order to build an unbiased converging MC scheme for system (17) is to rewrite it as an equivalent scalar equation of unknown $f(\mathbf{x},t,\mathbf{v},g)$. Quantity f has one additional dependence with respect to the uncertainty related *group* g but is scalar. Of course, f also depends on V, unknown of (17). It is defined by

$$f(\mathbf{x},t,\mathbf{v},g)=\sum_{k=0}^{P}V_k(\mathbf{x},t,\mathbf{v})\delta_k(g). \tag{18}$$

The scalar equation satisfied by f is given by

$$\begin{aligned}\partial_t f(\mathbf{x},t,\mathbf{v},g)+\mathbf{v}\cdot\nabla_{\mathbf{x}}f(\mathbf{x},t,\mathbf{v},g)= \quad & +\left[\mathbf{v}\cdot\nabla_{\mathbf{x}}R^{-1}Rf\right](\mathbf{x},t,\mathbf{v},g)\\ & -v\Lambda_t(\mathbf{x},\mathbf{v},g)f(\mathbf{x},t,\mathbf{v},g)\\ & +v\iint\Lambda_s(\mathbf{x},\mathbf{v}\cdot\mathbf{v}',g\cdot g')f(\mathbf{x},t,\mathbf{v}',g')\,\mathrm{d}\mathbf{v}'\,\mathrm{d}g',\end{aligned} \tag{19}$$

where[13] $\left[\mathbf{v}\cdot\nabla_{\mathbf{x}}R^{-1}Rf\right](\mathbf{x},t,\mathbf{v},g)$, $\Lambda_t(\mathbf{x},\mathbf{v},g)$ and $\Lambda_s(\mathbf{x},\mathbf{v}\cdot\mathbf{v}',g\cdot g')$ are built from $R(\mathbf{x},\mathbf{v})$, $R^{-1}(\mathbf{x},\mathbf{v})$, $\mathbf{v}\cdot\nabla_{\mathbf{x}}R^{-1}(\mathbf{x},\mathbf{v})$, $\Lambda_t(\mathbf{x},\mathbf{v})$ and $\Lambda_s(\mathbf{x},\mathbf{v}\cdot\mathbf{v}')$. Those new cross-sections are built such that the k^{th} component of (17) is recovered when formally applying operator

$$L_k(h)=\int h\delta_k(g)\,\mathrm{d}g,\forall k\in\{0,...,P\},$$

[12] Also known as *implicit capture*.

[13] Note that we use what may seem to be abusive notations here but we take care of expliciting the dependences with respect to all variables. As a consequence, there are no ambiguities between, for example $\Lambda_t(\mathbf{x},\mathbf{v},g)$ involved in (19) and $\Lambda_t(\mathbf{x},\mathbf{v})$ involved in (17).

to every terms of equation (19). For example, introduce the notation $\left[\mathbf{v}\cdot\nabla_{\mathbf{x}}R^{-1}(\mathbf{x},\mathbf{v})R(\mathbf{x},\mathbf{v})\right]_{i,j} = \mathbf{R}_{i,j}(\mathbf{x},\mathbf{v})$, then we have

$$\begin{aligned}\mathbf{R}\left[f\right](\mathbf{x},t,\mathbf{v},g) &= \left[\mathbf{v}\cdot\nabla_{\mathbf{x}}R^{-1}Rf\right](\mathbf{x},t,\mathbf{v},g),\\ &= \int\sum_{i=0}^{P}\mathbf{R}_{i,k}(\mathbf{x},\mathbf{v})\delta_{i,k}(g',g)f(\mathbf{x},t,\mathbf{v},g')\,\mathrm{d}g'.\end{aligned}$$

Applying operator L_k to the above expression leads to

$$\begin{aligned}L_k\left(\left[\mathbf{v}\cdot\nabla_{\mathbf{x}}R^{-1}Rf\right](\mathbf{x},t,\mathbf{v},g)\right) &= \int\sum_{i=0}^{P}\mathbf{R}_{i,k}(\mathbf{x},\mathbf{v})\delta_{i}(g')f(\mathbf{x},t,\mathbf{v},g')\,\mathrm{d}g',\\ &= \sum_{i=0}^{P}\mathbf{R}_{i,k}(\mathbf{x},\mathbf{v})V_i(\mathbf{x},t,\mathbf{v}),\\ &= \left[\left[\mathbf{v}\cdot\nabla_{\mathbf{x}}R^{-1}R\right](\mathbf{x},\mathbf{v})V(\mathbf{x},t,\mathbf{v})\right]_k.\end{aligned}$$

Introduce the notation $\Lambda_t(\mathbf{x},\mathbf{v}) = \mathrm{diag}(\lambda_t^0(\mathbf{x},\mathbf{v}),...,\lambda_t^P(\mathbf{x},\mathbf{v}))$, then $\Lambda_t(\mathbf{x},\mathbf{v},g)$ is defined by

$$\Lambda_t(\mathbf{x},\mathbf{v},g) = \sum_{k=0}^{P}\lambda_t^k(\mathbf{x},\mathbf{v})\delta_k(g). \tag{20}$$

In a same manner, for the scattering counterpart, if we denote the general term of $\Lambda_s(\mathbf{x},\mathbf{v}\cdot\mathbf{v}')$ as $\lambda_s^{k,k'}(\mathbf{x},\mathbf{v}\cdot\mathbf{v}'),\forall(k,k')\in\{0,...,P\}^2$, we have

$$\Lambda_s(\mathbf{x},\mathbf{v}\cdot\mathbf{v}',g\cdot g') = \sum_{k'=0}^{P}\lambda_s^{k,k'}(\mathbf{x},\mathbf{v}\cdot\mathbf{v}')\delta_{k,k'}(g,g'). \tag{21}$$

Now, in the following paragraphs, we define few notations related to $\Lambda_s(\mathbf{x},\mathbf{v}\cdot\mathbf{v}',g\cdot g')$ which will ease the description of the MC schemes. Those quantities basically aim at building conditionnal probability distributions which will be used at each collision during the MC resolution. First, let us decompose $\Lambda_s(\mathbf{x},\mathbf{v}\cdot\mathbf{v}',g\cdot g')$ as

$$\Lambda_s(\mathbf{x},\mathbf{v}\cdot\mathbf{v}',g\cdot g') = \sum_{k'=0}^{P}\lambda_s^{k,k'}(\mathbf{x},\mathbf{v})\gamma^{k,k'}(\mathbf{x},\mathbf{v})P_{\lambda_s}^{k,k'}(\mathbf{x},\mathbf{v}\cdot\mathbf{v}')\delta_{k,k'}(g,g'), \tag{22}$$

where $\lambda_s^{k,k'}(\mathbf{x},\mathbf{v}) = |\int\lambda_s^{k,k'}(\mathbf{x},\mathbf{v}\cdot\mathbf{v}')\,\mathrm{d}\mathbf{v}'|$. The term $\gamma^{k,k'}(\mathbf{x},\mathbf{v})$ is introduced mainly because there may exists[14] some couples $(k,k')\in\{0,...,P\}^2$ such that $\lambda_s^{k,k'}(\mathbf{x},\mathbf{v})<0$. This term is equal to $\gamma^{k,k'}(\mathbf{x},\mathbf{v}) = \frac{\int\lambda_s^{k,k'}(\mathbf{x},\mathbf{v}\cdot\mathbf{v}')\,\mathrm{d}\mathbf{v}'}{|\int\lambda_s^{k,k'}(\mathbf{x},\mathbf{v}\cdot\mathbf{v}')\,\mathrm{d}\mathbf{v}'|}$, i.e. it corresponds to the sign of $\int\lambda_s^{k,k'}(\mathbf{x},\mathbf{v}\cdot\mathbf{v}')\,\mathrm{d}\mathbf{v}',\forall(k,k')\in\{0,...,P\}^2$. Such choice of decomposition is handy because we can then easily define $P_{\lambda_s}^{k,k'}(\mathbf{x},\mathbf{v}\cdot\mathbf{v}') = \frac{\Lambda_s^{k,k'}(\mathbf{x},\mathbf{v}\cdot\mathbf{v}')}{\lambda_s^{k,k'}(\mathbf{x},\mathbf{v})},\forall(k,k')\in\{0,...,P\}^2$ such that $P_{\lambda_s}^{k,k'}(\mathbf{x},\mathbf{v}\cdot\mathbf{v}')>0$ and such that $\int P_{\lambda_s}^{k,k'}(\mathbf{x},\mathbf{v}\cdot\mathbf{v}')\,\mathrm{d}\mathbf{v}'=1$ $\forall\mathbf{x}\in\mathcal{D},\mathbf{v}\in\mathcal{V},(k,k')\in\{0,...,P\}^2$. In other words, the quantities $P_{\lambda_s}^{k,k'}(\mathbf{x},\mathbf{v}\cdot\mathbf{v}')\,\mathrm{d}\mathbf{v}'$ are probability

[14] This is the case for example in the test-problem of section 4.5.

measures with respect to $\mathbf{v}'$, $\forall \mathbf{x} \in \mathcal{D}, \forall \mathbf{v} \in \mathcal{V}, \forall (k,k') \in \{0,...,P\}^2$. Quantities $(P_{\lambda_s}^{k,k'})_{(k,k')\in\{0,...,P\}^2}$ will be used to sample the velocity and angles of each MC particle conditionnally to changing of uncertain group from k' to k in a backward resolution or from k to k' in a forward one.

In the next sections 3.4–3.5–3.6, we consider we are within a cell $\overline{\mathcal{D}_i}$ (cf. remark 3.1). This means that in these sections, we can consider that the interface term $\left[\mathbf{v}\cdot\nabla_{\mathbf{x}}R^{-1}Rf\right](\mathbf{x},t,\mathbf{v},g) = 0$ without lack of generality. This will ease the next computations. The MC operations needed to treat the interface case $\left[\mathbf{v}\cdot\nabla_{\mathbf{x}}R^{-1}Rf\right](\mathbf{x},t,\mathbf{v},g) \neq 0$ are described in section 3.7.

3.4. The integral form of equation (19) *within a cell (i.e. for* $\mathbf{v}\cdot\nabla_{\mathbf{x}}R^{-1}RV = 0$*, cf. remark 3.1)*

In order to build an MC scheme for equation (19), we need to define consistent samplings. The first step for this consists in rewriting (19) in an integral form. The computations may appear tedious due to the fact that all dependences with respect to $(\mathbf{x},t,\mathbf{v},g)$ *must* be recalled: it is important in order to ensure the consistency/unbiasedness of the MC resolution (see theorem 3.2.1 of [33]). In order to rewrite (19) in integral form, we perform several successive changes of variable. We detail every of them in the following sections. The described methodology leads to the adjoint MC resolution of the equation (or backward Kolmogorov equation, see [61]). The direct counterpart (forward Kolmogorov equation) can easily be deduced from the next analysis (see remark 3.5).

As explained before, the methodology resumes to a succession of changes of variable. The first one consists in rewriting the transport equation (19) on a characteristic $\mathbf{x}+\mathbf{v}t$. Equation (19) rewritten along a characteristic $(\mathbf{x}+\mathbf{v}s,s,\mathbf{v},g)$ becomes

$$\begin{aligned}\partial_s f(\mathbf{x}+\mathbf{v}s,s,\mathbf{v},g) = \quad & -v\Lambda_t(\mathbf{x}+\mathbf{v}s,\mathbf{v},g)f(\mathbf{x}+\mathbf{v}s,s,\mathbf{v},g) \\ & + \iint v\Lambda_s(\mathbf{x}+\mathbf{v}s,\mathbf{v}\cdot\mathbf{v}',g\cdot g')f(\mathbf{x}+\mathbf{v}s,s,\mathbf{v}',g')\,\mathrm{d}\mathbf{v}'\,\mathrm{d}g'.\end{aligned} \tag{23}$$

Let us multiply each side of the equality by $\exp\left[\int_0^s v\Lambda_t(\mathbf{x}+\mathbf{v}\alpha,\mathbf{v},g)\,\mathrm{d}\alpha\right]$. We then get

$$\begin{aligned}&\partial_s\left[f(\mathbf{x}+\mathbf{v}s,s,\mathbf{v},g)e^{\int_0^s v\Lambda_t(\mathbf{x}+\mathbf{v}\alpha,\mathbf{v},g)\,\mathrm{d}\alpha}\right] = \\ &\qquad e^{\int_0^s v\Lambda_t(\mathbf{x}+\mathbf{v}\alpha,\mathbf{v},g)\,\mathrm{d}\alpha}\iint v\Lambda_s(\mathbf{x}+\mathbf{v}s,\mathbf{v}\cdot\mathbf{v}',g\cdot g')f(\mathbf{x}+\mathbf{v}s,s,\mathbf{v}',g')\,\mathrm{d}\mathbf{v}'\,\mathrm{d}g'.\end{aligned} \tag{24}$$

Integrating (24) in the time interval $[0,t]$ leads to

$$\begin{aligned}f(\mathbf{x}+\mathbf{v}t,t,\mathbf{v},g) = {} & f_0(\mathbf{x},\mathbf{v},g)e^{-\int_0^t v\Lambda_t(\mathbf{x}+\mathbf{v}\alpha,\mathbf{v},g)\,\mathrm{d}\alpha} \\ & + \int_0^t e^{-\int_s^t v\Lambda_t(\mathbf{x}+\mathbf{v}\alpha,\mathbf{v},g)\,\mathrm{d}\alpha}\iint v\Lambda_s(\mathbf{x}+\mathbf{v}s,\mathbf{v}\cdot\mathbf{v}',g\cdot g')f(\mathbf{x}+\mathbf{v}s,s,\mathbf{v}',g')\,\mathrm{d}\mathbf{v}'\,\mathrm{d}g'\,\mathrm{d}s.\end{aligned} \tag{25}$$

We have then

$$\begin{aligned}f(\mathbf{x},t,\mathbf{v},g) = {} & f_0(\mathbf{x}-\mathbf{v}t,\mathbf{v},g)e^{-\int_0^t v\Lambda_t(\mathbf{x}-\mathbf{v}(t-\alpha),\mathbf{v},g)\,\mathrm{d}\alpha} \\ & + \int_0^t e^{-\int_s^t v\Lambda_t(\mathbf{x}-\mathbf{v}(t-\alpha),\mathbf{v},g)\,\mathrm{d}\alpha}\iint v\Lambda_s(\mathbf{x}-\mathbf{v}(t-s),\mathbf{v}\cdot\mathbf{v}',g\cdot g')f(\mathbf{x}-\mathbf{v}(t-s),s,\mathbf{v}',g')\,\mathrm{d}\mathbf{v}'\,\mathrm{d}g'\,\mathrm{d}s.\end{aligned} \tag{26}$$

Equation (26) is an integral equation but still needs to be worked on: first, notice that

$$\begin{aligned}e^{-\int_0^t v\Lambda_t(\mathbf{x}-\mathbf{v}(t-\alpha),\mathbf{v},g)\,\mathrm{d}\alpha} &= e^{-\int_0^t v\Lambda_t(\mathbf{x}-\mathbf{v}\alpha,\mathbf{v},g)\,\mathrm{d}\alpha}, \\ &= \int_t^\infty v\Lambda_t(\mathbf{x}-\mathbf{v}s,\mathbf{v},g)e^{-\int_0^s v\Lambda_t(\mathbf{x}-\mathbf{v}\alpha,\mathbf{v},g)\,\mathrm{d}\alpha}\,\mathrm{d}s.\end{aligned}$$

Then, the integral counterpart of (19) is given by

$$\begin{aligned} &f(\mathbf{x},t,\mathbf{v},g) = \\ &+\int_t^\infty f_0(\mathbf{x}-\mathbf{v}t,\mathbf{v},g)v\Lambda_t(\mathbf{x}-\mathbf{v}s,\mathbf{v},g)e^{-\int_0^s v\Lambda_t(\mathbf{x}-\mathbf{v}\alpha,\mathbf{v},g)\,\mathrm{d}\alpha}\,\mathrm{d}s \\ &+\int_0^t e^{-\int_s^t v\Lambda_t(\mathbf{x}-\mathbf{v}(t-\alpha),\mathbf{v},g)\,\mathrm{d}\alpha}\iint v\Lambda_s(\mathbf{x}-\mathbf{v}(t-s),\mathbf{v}\cdot\mathbf{v}',g\cdot g')f(\mathbf{x}-\mathbf{v}(t-s),s,\mathbf{v}',g')\,\mathrm{d}\mathbf{v}'\,\mathrm{d}g'\,\mathrm{d}s. \end{aligned} \tag{27}$$

Building an MC scheme now implies introducing a set of random variables together with their probability measure in order to rewrite (27) as an expectation. The choice of the set of random variables is not unique and consequently leads to different MC schemes having different properties. In the following sections, we detail the construction of two MC schemes

- the semi-analog multigroup-like one (section 3.5),
- and the non-analog multigroup-like one (section 3.6).

3.5. The semi-analog multigroup (adjoint) MC scheme

The semi-analog MC scheme (also known as 'implicit capture' in the literature [17, 16]) starts from the integral form (27). Let us perform a change of variable ($\beta = t - s$ and β is immediately replaced by s) in the time integration in the scattering part. We obtain

$$\begin{aligned} &f(\mathbf{x},t,\mathbf{v},g) = \\ &+\int_t^\infty f_0(\mathbf{x}-\mathbf{v}t,\mathbf{v},g)v\Lambda_t(\mathbf{x}-\mathbf{v}s,\mathbf{v},g)e^{-\int_0^s v\Lambda_t(\mathbf{x}-\mathbf{v}\alpha,\mathbf{v},g)\,\mathrm{d}\alpha}\,\mathrm{d}s \\ &+\int_0^t e^{-\int_0^s v\Lambda_t(\mathbf{x}-\mathbf{v}\alpha,\mathbf{v},g)\,\mathrm{d}\alpha}\iint v\Lambda_s(\mathbf{x}-\mathbf{v}s,\mathbf{v}\cdot\mathbf{v}',g\cdot g')f(\mathbf{x}-\mathbf{v}s,t-s,\mathbf{v}',g')\,\mathrm{d}\mathbf{v}'\,\mathrm{d}g'\,\mathrm{d}s. \end{aligned} \tag{28}$$

It is then possible to factorize by

$$f_t(\mathbf{x},s,\mathbf{v},g)\,\mathrm{d}s = \mathbf{1}_{[0,\infty[}(s)v\Lambda_t(\mathbf{x}-\mathbf{v}s,\mathbf{v},g)e^{-\int_0^s v\Lambda_t(\mathbf{x}-\mathbf{v}\alpha,\mathbf{v},g)\,\mathrm{d}\alpha}\,\mathrm{d}s. \tag{29}$$

The above expression is a probability measure $\forall(\mathbf{x},s,\mathbf{v},g)\in\mathcal{D}\times[0,T]\times\mathbb{R}^3\times\{0,...,P\}$: indeed, it is positive and sums up to 1 $\forall(\mathbf{x},s,\mathbf{v},g)\in\mathcal{D}\times[0,T]\times\mathbb{R}^3\times\{0,...,P\}$. Using its expression in (28) leads to

$$f(\mathbf{x},t,\mathbf{v},g) = \iiint \begin{bmatrix} +\mathbf{1}_{[t,\infty[}(s) & f_0(\mathbf{x}-\mathbf{v}t,\mathbf{v},g) & \delta_{\mathbf{v}}(\mathbf{v}')\delta_g(g') \\ +\mathbf{1}_{[0,t]}(s) & f(\mathbf{x}-\mathbf{v}s,t-s,\mathbf{v}',g') & \dfrac{\Lambda_s(\mathbf{x}-\mathbf{v}s,\mathbf{v}\cdot\mathbf{v}',g\cdot g')}{\Lambda_t(\mathbf{x}-\mathbf{v}s,\mathbf{v},g)} \end{bmatrix} f_t(\mathbf{x},s,\mathbf{v},g)\,\mathrm{d}s\,\mathrm{d}\mathbf{v}'\,\mathrm{d}g'. \tag{30}$$

Let us now build the probability distributions related to the scattering part of the integral equation. With the notations introduced in section 3.3, we can rewrite Λ_s as

$$\Lambda_s(\mathbf{x}-\mathbf{v}s,\mathbf{v}\cdot\mathbf{v}',g\cdot g') = \sum_{k'=0}^{P}\lambda_s^{k,k'}(\mathbf{x}-\mathbf{v}s,\mathbf{v})\gamma^{k,k'}(\mathbf{x}-\mathbf{v}s,\mathbf{v})P_{\lambda_s}^{k,k'}(\mathbf{x}-\mathbf{v}s,\mathbf{v}\cdot\mathbf{v}')\delta_{k,k'}(g,g').$$

Let us integrate the above relation with respect to both g' and $\mathbf{v}'$: we obtain

$$\iint \Lambda_s(\mathbf{x}-\mathbf{v}s,\mathbf{v}\cdot\mathbf{v}',g\cdot g')\,\mathrm{d}\mathbf{v}'\,\mathrm{d}g' = \sum_{k'=0}^{P}\lambda_s^{g,k'}(\mathbf{x}-\mathbf{v}s,\mathbf{v})\gamma^{g,k'}(\mathbf{x}-\mathbf{v}s,\mathbf{v}).$$

$$\iint \Lambda_s(\mathbf{x}-\mathbf{v}s,\mathbf{v}\cdot\mathbf{v}',g\cdot g')\,\mathrm{d}\mathbf{v}'\,\mathrm{d}g' = \underbrace{\left[\sum_{k'=0}^{P}\lambda_s^{g,k'}(\mathbf{x}-\mathbf{v}s,\mathbf{v})\right]}_{\Lambda_s(\mathbf{x}-\mathbf{v}s,\mathbf{v},g)}\sum_{k'=0}^{P}\frac{\lambda_s^{g,k'}(\mathbf{x}-\mathbf{v}s,\mathbf{v})}{\Lambda_s(\mathbf{x}-\mathbf{v}s,\mathbf{v},g)}\gamma^{g,k'}(\mathbf{x}-\mathbf{v}s,\mathbf{v}).$$

Let us introduce

$$G_s(\mathbf{x}-\mathbf{v}s,\mathbf{v},g,g') = \sum_{k'=0}^{P}\frac{\lambda_s^{g,k'}(\mathbf{x}-\mathbf{v}s,\mathbf{v})}{\sum_{k'=0}^{P}\lambda_s^{g,k'}(\mathbf{x}-\mathbf{v}s,\mathbf{v})}\delta_{k'}(g'). \tag{31}$$

The quantity $G_s(\mathbf{x}-\mathbf{v}s,\mathbf{v},g,g')\,\mathrm{d}g'$ is positive and is summing up to 1. It is consequently a discrete probability measure $\forall(\mathbf{x},s,\mathbf{v},g)\in\mathcal{D}\times[0,T]\times\mathbb{R}^3\times\{0,...,P\}$. It defines the probability, incoming from the uncertain group g', to outcome, at a collision, in the uncertain group g. Furthermore, recall that $P_{\lambda_s}^{k,k'}(\mathbf{x}-\mathbf{v}s,\mathbf{v}\cdot\mathbf{v}')\,\mathrm{d}\mathbf{v}'$ is already positive and summing up to 1 $\forall(\mathbf{x},s,\mathbf{v},k,k')\in\mathcal{D}\times[0,T]\times\mathbb{R}^3\times\{0,...,P\}\times\{0,...,P\}$. Let us then introduce the following random variables associated to the previously identified probability measures

$$\left\{\begin{array}{lll} \tau & \text{with probability measure} & f_t(\mathbf{x},t,\mathbf{v},g)\,\mathrm{d}s,\\ \mathbf{g}' & \text{with probability measure} & G_s(\mathbf{x}-\mathbf{v}s,\mathbf{v},g,g')\,\mathrm{d}g',\\ \mathbf{V}' & \text{with probability measure} & P_{\lambda_s}^{g,\mathbf{g}'}(\mathbf{x}-\mathbf{v}s,\mathbf{v}\cdot\mathbf{v}')\,\mathrm{d}\mathbf{v}'. \end{array}\right. \tag{32}$$

Then (30) can be rewritten *in an adjoint recursive way* as an expectation over the above set of random variables (32)

$$f(\mathbf{x},t,\mathbf{v},g) = \mathbb{E}\left[\begin{array}{l} +\mathbf{1}_{[t,\infty[}(\tau)f_0(\mathbf{x}-\mathbf{v}t,\mathbf{v},g)\delta_{\mathbf{v}}(\mathbf{V}')\delta_g(\mathbf{g}')\\ +\mathbf{1}_{[0,t]}(\tau)\dfrac{\Lambda_s(\mathbf{x}-\mathbf{v}\tau,\mathbf{v},g)}{\Lambda_t(\mathbf{x}-\mathbf{v}\tau,\mathbf{v},g)}\gamma^{g,\mathbf{g}'}(\mathbf{x}-\mathbf{v}\tau,\mathbf{v})f(\mathbf{x}-\mathbf{v}\tau,t-\tau,\mathbf{V}',\mathbf{g}') \end{array}\right]. \tag{33}$$

Let us now consider a 'particle' solution $(f_p)_{p\in\{1,...,N_{MC}\}}$ of (33) having the particular form

$$f_p(\mathbf{x},t,\mathbf{v},g) = w_p(t)\delta_{\mathbf{x}}(\mathbf{x}_p(t))\delta_{\mathbf{v}}(\mathbf{v}_p(t))\delta_g(g_p(t)). \tag{34}$$

The MC scheme intensively uses the linearity of equation (33): if $(f_p)_{p\in\{1,...,N_{MC}\}}$ are independent solutions of (33) then $\sum_{p=1}^{N_{MC}}f_p$ is also solution of (33). Now, remains to identify the operations one has to apply in order to ensure each $(f_p)_{p\in\{1,...,N_{MC}\}}$ is effectively a MC solution of (33). To do so, one needs to plug f_p into (33) and solve a system of (compatible) equations with unknowns $w_p(t),\mathbf{x}_p(t),\mathbf{v}_p(t),g_p(t)$. Plugging (34) in (33) leads to

$$\begin{array}{l} w_p(t)\delta_{\mathbf{x}}(\mathbf{x}_p(t))\delta_{\mathbf{v}}(\mathbf{v}_p(t))\delta_g(g_p(t)) =\\ \quad\mathbf{1}_{[t,\infty[}(\tau)\quad w_p(0)\qquad\qquad\qquad\qquad\qquad\qquad \delta_{\mathbf{x}-\mathbf{v}t}(\mathbf{x}_p(0))\quad \delta_{\mathbf{v}}(\mathbf{v}_p(0))\quad \delta_g(g_p(0))\\ \quad\mathbf{1}_{[0,t]}(\tau)\quad \dfrac{\Lambda_s}{\Lambda_t}(\mathbf{x}-\mathbf{v}\tau,\mathbf{v})\gamma^{g,\mathbf{g}'}(\mathbf{x}-\mathbf{v}\tau,\mathbf{v})w_p(t-\tau)\quad \delta_{\mathbf{x}-\mathbf{v}\tau}(\mathbf{x}_p(t-\tau))\quad \delta_{\mathbf{V}'}(\mathbf{v}_p(t-\tau))\quad \delta_{\mathbf{g}'}(g_p(t-\tau)), \end{array}$$

so that the weight, the position, the velocity and the uncertain group satisfy

$$\left\{\begin{array}{lll} \mathbf{x}_p(t) = & \mathbf{1}_{[t,\infty[}(\tau)(\mathbf{x}_0+\mathbf{v}t) & +\mathbf{1}_{[0,t]}(\tau)(\mathbf{x}_{t-\tau}+\mathbf{v}\tau),\\ g_p(t) = & \mathbf{1}_{[t,\infty[}(\tau)g & +\mathbf{1}_{[0,t]}(\tau)\mathbf{g}'.\\ w_p(t) = & \mathbf{1}_{[t,\infty[}(\tau)w_p(0) & +\mathbf{1}_{[0,t]}(\tau)\dfrac{\Lambda_s}{\Lambda_t}(\mathbf{x}_p(t-\tau),\mathbf{v}_p(t-\tau))\gamma^{g,\mathbf{g}'}(\mathbf{x}_p(t-\tau),\mathbf{v}_p(t-\tau))w_p(t-\tau),\\ \mathbf{v}_p(t) = & \mathbf{1}_{[t,\infty[}(\tau)\mathbf{v} & +\mathbf{1}_{[0,t]}(\tau)\mathbf{V}'. \end{array}\right. \tag{35}$$

Practically, the above set of treatments means that in order to consistently solve system (19) in a backward way, with a converging unbiased MC scheme (see theorem 3.2.1 of [33]), for any given MC particle of fields $\mathbf{x}, t, \mathbf{v}, g$,

- a collisional time τ must be sampled according to f_t given by (29).
- if $\tau < t$ then a collision occurs:
  - the MC particle moves from its position to the position of the collision,
  - its outer/inner group is sampled from the discrete probability measure G_s given by (31),
  - its weight is multiplied by the ratio $\frac{\Lambda_s}{\Lambda_t}$ at the position of the collision. The weight may change of sign depending on the value of $\gamma^{g,\mathbf{g}'}$,
  - and finally, the velocity of the particle is sampled from $P_{\lambda_s}^{g,\mathbf{g}'}$ which is built from (22).
- if $\tau > t$, a collision does not occur and the particle simply moves in a straight line from its position to $\mathbf{x}_0$. The tracking of the particle ends with this event as it finishes the recursive treatment.

The above system of equation in terms of weight, position, velocity and group leads to the recursive numerical treatment/algorithm (remember we here detailed the adjoint formulation) summed up in algorithm 1.

3.6. The non-analog multigroup (adjoint) MC scheme

In the previous section, we presented the semi-analog MC scheme for solving the linear Boltzmann equation (19). Such an MC scheme is mostly used in neutronic applications [49, 27, 28, 21, 22, 23]. In this section, we describe the non-analog scheme, intensively applied in photonic ones [50, 24, 25, 51, 52, 53, 29]. As in the previous sections, we first rewrite the linear Boltzmann equation (19) in an integral form. The non-analog one is obtained from slightly different changes of variables which are detailed in the next section. We then present the set of random variables at the basis of the MC scheme. The scheme is also refered to as 'capture along the flight path' in the literature.

First, note that in this section, we intensively rely on the assumption that we have access to same quantities $\Lambda_s, \Lambda_t, R, R^{-1}$ as in the previous section 3.5 (see remark 3.3). The non-analog scheme needs the additional introduction of

$$\Lambda_a(\mathbf{x}, \mathbf{v}, g) = \Lambda_t(\mathbf{x}, \mathbf{v}, g) - \Lambda_s(\mathbf{x}, \mathbf{v}, g).$$

Let us decompose Λ_t into $\Lambda_a + \Lambda_s$ in (26). This allows keeping the term $e^{-\int_0^s v\Lambda_a(\mathbf{x}-\mathbf{v}(t-\alpha),\mathbf{v},g)\,\mathrm{d}\alpha}$ in factor of f_0 and f. Now using the fact that

$$e^{-\int_0^t v\Lambda_s(\mathbf{x}-\mathbf{v}(t-\alpha),\mathbf{v},g)\,\mathrm{d}\alpha} = e^{-\int_0^t v\Lambda_s(\mathbf{x}-\mathbf{v}\alpha,\mathbf{v},g)\,\mathrm{d}\alpha} = \int_t^\infty v\Lambda_s(\mathbf{x}-\mathbf{v}s,\mathbf{v},g)e^{-\int_0^s v\Lambda_s(\mathbf{x}-\mathbf{v}\alpha,\mathbf{v},g)\,\mathrm{d}\alpha}\,\mathrm{d}s,$$

equation (26) rewrites

$$\begin{aligned} f(\mathbf{x},t,\mathbf{v},g) = & \\ & + \int_t^\infty f_0(\mathbf{x}-\mathbf{v}t,\mathbf{v},g)e^{-\int_0^s v\Lambda_a(\mathbf{x}-\mathbf{v}\alpha,\mathbf{v},g)\,\mathrm{d}\alpha}v\Lambda_s(\mathbf{x}-\mathbf{v}s,\mathbf{v},g)e^{-\int_0^s v\Lambda_s(\mathbf{x}-\mathbf{v}\alpha,\mathbf{v},g)\,\mathrm{d}\alpha}\,\mathrm{d}s \\ & + \int_0^t \left[\begin{array}{l} v\Lambda_s(\mathbf{x}-\mathbf{v}s,\mathbf{v},g)e^{-\int_0^s v\Lambda_s(\mathbf{x}-\mathbf{v}\alpha,\mathbf{v},g)\,\mathrm{d}\alpha}e^{-\int_0^s v\Lambda_a(\mathbf{x}-\mathbf{v}\alpha,\mathbf{v},g)\,\mathrm{d}\alpha} \\ \times \iint P_s(\mathbf{x}-\mathbf{v}s,\mathbf{v}\cdot\mathbf{v}',g\cdot g')f(\mathbf{x}-\mathbf{v}s,t-s,\mathbf{v}',g')\,\mathrm{d}\mathbf{v}'\,\mathrm{d}g' \end{array} \right] \mathrm{d}s, \end{aligned} \tag{36}$$

where $P_s(\mathbf{x}-\mathbf{v}s, \mathbf{v}\cdot\mathbf{v}', g\cdot g')$ is defined by

$$P_s(\mathbf{x}-\mathbf{v}s, \mathbf{v}\cdot\mathbf{v}', g\cdot g') = \frac{\Lambda_s(\mathbf{x}-\mathbf{v}s, \mathbf{v}\cdot\mathbf{v}', g\cdot g')}{\Lambda_s(\mathbf{x}-\mathbf{v}s, \mathbf{v}, g)}.$$

It is then possible to factorize by

$$f_s(\mathbf{x}, s, \mathbf{v}, g)\,\mathrm{d}s = \mathbf{1}_{[0,\infty[}(s) v\Lambda_s(\mathbf{x}-\mathbf{v}s, \mathbf{v}, g) e^{-\int_0^s v\Lambda_s(\mathbf{x}-\mathbf{v}\alpha, \mathbf{v}, g)\,\mathrm{d}\alpha}\,\mathrm{d}s. \tag{37}$$

It is also a probability measure (with respect to Λ_s rather than Λ_t). We then rewrite the linear Boltzmann equation in another integral form given by

$$\begin{array}{l} f(\mathbf{x}, t, \mathbf{v}, g) = \\ \displaystyle\iint \left[\begin{array}{llll} \mathbf{1}_{[t,\infty[}(s) & f_0(\mathbf{x}-\mathbf{v}t, \mathbf{v}, g) & e^{-\int_0^s v\Lambda_a(\mathbf{x}-\mathbf{v}\alpha, \mathbf{v}, g)\,\mathrm{d}\alpha} & \delta_{\mathbf{v}}(\mathbf{v}')\delta_g(g') \\ \mathbf{1}_{[0,t]}(s) & f(\mathbf{x}-\mathbf{v}s, t-s, \mathbf{v}', g') & e^{-\int_0^s v\Lambda_a(\mathbf{x}-\mathbf{v}\alpha, \mathbf{v}, g)\,\mathrm{d}\alpha} & P_s(\mathbf{x}-\mathbf{v}s, \mathbf{v}\cdot\mathbf{v}', g\cdot g') \end{array}\right] \\ \quad \times f_s(\mathbf{x}, s, \mathbf{v}, g)\,\mathrm{d}\mathbf{v}'\,\mathrm{d}s\,\mathrm{d}g'. \end{array} \tag{38}$$

Integral form (38) obtained here is different from the one (27) used for the semi-analog MC scheme. It mainly differs due to the exponential term multiplying f_0 and f. Let us now introduce the random variables

$$\left\{\begin{array}{lll} \tau & \text{with probability measure} & f_s(\mathbf{x}, t, \mathbf{v}, g)\,\mathrm{d}s, \\ \mathbf{g}' & \text{with probability measure} & G_s(\mathbf{x}-\mathbf{v}s, \mathbf{v}, g, g')\,\mathrm{d}g', \\ \mathbf{V}' & \text{with probability measure} & P_{\lambda_s}^{g,\mathbf{g}'}(\mathbf{x}-\mathbf{v}s, \mathbf{v}\cdot\mathbf{v}')\,\mathrm{d}\mathbf{v}'. \end{array}\right. \tag{39}$$

Equation (38) can then be rewritten *in an adjoint recursive way* as an expectation over the above set of non-analog random variables (39)

$$f(\mathbf{x}, t, \mathbf{v}, g) = \mathbb{E}\left[\begin{array}{lll} +\mathbf{1}_{[t,\infty[}(\tau) & e^{-\int_0^t v\Lambda_a(\mathbf{x}-\mathbf{v}\alpha, \mathbf{v}, g)\,\mathrm{d}\alpha} & f_0(\mathbf{x}-\mathbf{v}t, \mathbf{v}, g)\delta_{\mathbf{v}}(\mathbf{v}')\delta_g(\mathbf{g}') \\ +\mathbf{1}_{[0,t]}(\tau) & e^{-\int_0^\tau v\Lambda_a(\mathbf{x}-\mathbf{v}\alpha, \mathbf{v}, g)\,\mathrm{d}\alpha} & \gamma^{g,\mathbf{g}'}(\mathbf{x}-\mathbf{v}\tau, \mathbf{v}) f(\mathbf{x}-\mathbf{v}\tau, t-\tau, \mathbf{V}', \mathbf{g}') \end{array}\right]. \tag{40}$$

In the next section we deduce the MC treatments to apply in order to solve (40).

The steps for the construction of the non-analog MC scheme are similar to the previous ones. Let us consider 'particle' solutions $(f_p)_{p\in\{1,\ldots,N_{MC}\}}$ of (40) having the particular form (34). Let us plug them into (40) in order to identify the operations to perform to make sure each $(f_p)_{p\in\{1,\ldots,N_{MC}\}}$ is solution of (40). This leads to

$$\begin{array}{l} w_p(t)\delta_{\mathbf{x}}(\mathbf{x}_p(t))\delta_{\mathbf{v}}(\mathbf{v}_p(t))\delta_g(g_p(t)) = \\ \quad \begin{array}{llllll} +\mathbf{1}_{[t,\infty[}(\tau) & w_p(0) & \delta_{\mathbf{x}-\mathbf{v}t}(\mathbf{x}_p(0)) & \delta_{\mathbf{v}}(\mathbf{v}_p(0)) & e^{-\int_0^t v\Lambda_a(\mathbf{x}-\mathbf{v}\alpha, \mathbf{v}, g)\,\mathrm{d}\alpha} & \delta_g(g_p(0)) \\ +\mathbf{1}_{[0,t]}(\tau) & w_p(t-\tau) & \delta_{\mathbf{x}-\mathbf{v}\tau}(\mathbf{x}_p(t-\tau)) & \delta_{\mathbf{V}'}(\mathbf{v}_p(t-\tau)) & e^{-\int_0^\tau v\Lambda_a(\mathbf{x}-\mathbf{v}\alpha, \mathbf{v}, g)\,\mathrm{d}\alpha} & \gamma^{g,\mathbf{g}'}(\mathbf{x}_p(t-\tau), \mathbf{v}_p(t-\tau)) \end{array} \end{array},$$

so that the weight, the position and the velocity satisfy

$$\left\{\begin{array}{lll} \mathbf{x}_p(t) = & \mathbf{1}_{[t,\infty[}(\tau)(\mathbf{x}_0+\mathbf{v}t) & +\mathbf{1}_{[0,t]}(\tau)(\mathbf{x}_{t-\tau}+\mathbf{v}\tau), \\ g_p(t) = & \mathbf{1}_{[t,\infty[}(\tau) g & +\mathbf{1}_{[0,t]}(\tau)\mathbf{g}'. \\ w_p(t) = & \mathbf{1}_{[t,\infty[}(\tau) w_p(0) e^{-\int_0^t v\Lambda_a(\mathbf{x}-\mathbf{v}\alpha, \mathbf{v}, g)\,\mathrm{d}\alpha} & +\mathbf{1}_{[0,t]}(\tau) e^{-\int_0^\tau v\Lambda_a(\mathbf{x}-\mathbf{v}\alpha, \mathbf{v}, g)\,\mathrm{d}\alpha}\gamma^{g,\mathbf{g}'}(\mathbf{x}_p(t-\tau), \mathbf{v}) w_p(t-\tau), \\ \mathbf{v}_p(t) = & \mathbf{1}_{[t,\infty[}(\tau)\mathbf{v} & +\mathbf{1}_{[0,t]}(\tau)\mathbf{V}'. \end{array}\right. \tag{41}$$

Practically, the above set of treatment means that in order to consistently solve system (19) in backward way, with a converging non-analog MC scheme (see theorem 3.2.1 of [33]), for any given MC particle of fields $\mathbf{x}, t, \mathbf{v}, g$,

– a collisional time τ must be sampled according to f_s given by (37).

– if $\tau < t$ then a collision occurs:

  – the MC particle moves from its position to the position of the collision,

  – its outer/inner[15] group is sampled from the discrete probability measure G_s given by (31),

  – its weight is multiplied by $e^{-\int_0^\tau v\Lambda_a(\mathbf{x}-\mathbf{v}\alpha,\mathbf{v},g)\,\mathrm{d}\alpha}$ corresponding to a change of weight along the flight path of the particle between times t and 0. If the cross-section remains constant along the flight path of the particle, then $e^{-\int_0^\tau v\Lambda_a(\mathbf{x}-\mathbf{v}\alpha,\mathbf{v},g)\,\mathrm{d}\alpha} = e^{-v\Lambda_a(\mathbf{v},g)\tau}$. The weight may change of sign depending on the value of $\gamma^{g,\mathbf{g}'}$.

  – Finally, the velocity of the particle is sampled from $P_{\lambda_s}^{g,\mathbf{g}'}$ which is built from (22).

– if $\tau > t$, a collision does not occur and the particle simply moves in a straight line from its position to $\mathbf{x}_0$. The tracking of the particle ends with this event as it finishes the recursive treatment.

The above system of equation in terms of weight, position, velocity and group leads to the recursive numerical treatment/algorithm (remember we here detailed the adjoint formulation) summed up in algorithm 1.

3.7. The integral form of equation (19) *at the interface (i.e. for* $\mathbf{v}\cdot\nabla_{\mathbf{x}}R^{-1}RV \neq 0$*, cf. remark 3.1)*

In the previous sections, we identified the samplings allowing to solve equation (19) within a cell (with two different MC schemes). In this section, we focus on what must be done when an MC particle crosses the interface between two cells with different total cross-sections Λ_t (in other words, when the time to reach the interface of a cell τ_{exit} is smaller than the sampled collision time τ_{inter}, see algorithm 1).

In this section, we focus on the treatment of term $\left[\mathbf{v}\cdot\nabla_{\mathbf{x}}R^{-1}R\right]V \neq 0$ of system (17). For this, we first notice that we have

$$\begin{aligned}\partial_s\left[\ln(R^{-1}(\mathbf{x}+\mathbf{v}s,\mathbf{v}))\right] &= \left[\partial_s(R^{-1}(\mathbf{x}+\mathbf{v}s,\mathbf{v}))R(\mathbf{x}+\mathbf{v}s,\mathbf{v})\right],\\ &= \left[\mathbf{v}\cdot\nabla_{\mathbf{x}}(R^{-1}(\mathbf{x}+\mathbf{v}s,\mathbf{v}))R(\mathbf{x}+\mathbf{v}s,\mathbf{v})\right].\end{aligned}$$

This means that on a characteristic and without considering the collision counterpart of (17) (which has been treated in sections 3.5–3.6), we have

$$\partial_s V(\mathbf{x}+\mathbf{v}s,s,\mathbf{v}) = \quad \partial_s\left[\ln(R^{-1}(\mathbf{x}+\mathbf{v}s,\mathbf{v}))\right]V(\mathbf{x}+\mathbf{v}s,s,\mathbf{v}). \tag{42}$$

By equivalently rewriting vector $V = (V_0,...,V_P)^t$ as matrix $V = \mathrm{diag}(V_0,...,V_P)$, the above expression can be recast as

$$\partial_s\left[\ln(V(\mathbf{x}+\mathbf{v}s,s,\mathbf{v}))\right] \quad = \partial_s\left[\ln(R^{-1}(\mathbf{x}+\mathbf{v}s,\mathbf{v}))\right]. \tag{43}$$

We are going to integrate the above expression with respect to time between $[0,t]$. But first, let us introduce $t_* \in [0,t]$ such that R^{-1} is constant for $s < t_*$ and $t_* < s$ but is discontinuous, i.e. such that

$$R_-^{-1}(\mathbf{v}) = R^{-1}(\mathbf{x}+\mathbf{v}t_*^-,\mathbf{v}) \neq R^{-1}(\mathbf{x}+\mathbf{v}t_*^+,\mathbf{v}) = R_+^{-1}(\mathbf{v}),$$

[15] Depending on performing a forward or backward resolution.

where t_*^- and t_*^+ are the superior and inferior limits toward t_*. By integrating (43) between $[0, t_*^-] \cup [t_*^-, t_*^+] \cup [t_*^+, t]$, we get

$$\begin{cases} V(\mathbf{x}+\mathbf{v}t, t, \mathbf{v}) & = V(\mathbf{x}+\mathbf{v}t_*^+, t_*^+\mathbf{v}), & \forall s \in [t_*^+, t], \\ V(\mathbf{x}+\mathbf{v}t_*^+, t_*^+\mathbf{v}) & = R_+^{-1}(\mathbf{v})R_-(\mathbf{v})V(\mathbf{x}+\mathbf{v}t_*^-, t_*^-\mathbf{v}), & \text{at } t_*, \\ V(\mathbf{x}+\mathbf{v}t_*^-, t_*^-\mathbf{v}) & = V(\mathbf{x}, 0, \mathbf{v}), & \forall s \in [0, t_*^-]. \end{cases}$$

Let us now focus on what happens at the interface, i.e. at t_*: if we rewrite $\mathbf{R}(\mathbf{v}) = R_+^{-1}(\mathbf{v})R_-(\mathbf{v})$ of general term $[\mathbf{R}(\mathbf{v})]_{i,j} = \mathbf{R}_{i,j}(\mathbf{v})$ for conciseness, we have

$$f(\mathbf{x}+\mathbf{v}t_*^+, t_*^+, \mathbf{v}, g) \quad = \int \sum_{k=0}^{P} \mathbf{R}_{g,k}(\mathbf{v})\delta_k(g')f(\mathbf{x}+\mathbf{v}t_*^-, t_*^-, \mathbf{v}, g')\,\mathrm{d}g'.$$

Let us now do the same trick as for the scattering cross-section Λ_s in the previous sections, i.e. we rewrite the above expression as

$$f(\mathbf{x}+\mathbf{v}t_*^+, t_*^+, \mathbf{v}, g) \quad = \int \sum_{k=0}^{P} |\mathbf{R}_{g,k}| \sum_{k=0}^{P} sgn(\mathbf{R}_{g,k}) \frac{|\mathbf{R}_{g,k}|}{\sum_{l=0}^{P} |\mathbf{R}_{g,l}|} \delta_k(g')f(\mathbf{x}+\mathbf{v}t_*^-, t_*^-, \mathbf{v}, g')\,\mathrm{d}g'. \quad (44)$$

By doing so, we just introduced a discrete probability measure assigning probability $\left(\frac{|\mathbf{R}_{g,k}(\mathbf{v})|}{\sum_{m=0}^{P}|\mathbf{R}_{g,m}(\mathbf{v})|}\right)_{k\in\{0,...,P\}k\in\{0,...,P\}}$ to group $k \in \{0, ..., P\}$. Introduce random variable $\mathbf{g}'$ sampled according to the discrete probability measure

$$\mathbf{g}' \sim \mathcal{B}\left(\{0, ..., P\}, \left\{\frac{|\mathbf{R}_{g,0}(\mathbf{v})|}{\sum_{k=0}^{P}|\mathbf{R}_{g,k}(\mathbf{v})|}, ..., \frac{|\mathbf{R}_{g,P}(\mathbf{v})|}{\sum_{k=0}^{P}|\mathbf{R}_{g,k}(\mathbf{v})|}\right\}\right). \quad (45)$$

Then integral equation (44) can be rewritten as an expection over realisations of random variable $\mathbf{g}'$ (backward formulation)

$$f(\mathbf{x}+\mathbf{v}t_*^+, t_*^+, \mathbf{v}, g) \quad = \sum_{k=0}^{P} |\mathbf{R}_{g,k}(\mathbf{v})| \times \mathbb{E}\left[sgn\left(\mathbf{R}_{g,\mathbf{g}'}(\mathbf{v})\right) f(\mathbf{x}+\mathbf{v}t_*^-, t_*^-, \mathbf{v}, \mathbf{g}')\right]. \quad (46)$$

Now, it only remains, just as in the previous section, to introduce an 'uncertain MC particle' solution of equation (46) in order to identify the treatments one must perform each time a particle crosses an interface with different uncertain cross-sections Λ_t on each side. The calculations are similar to the ones for $P_{\lambda_s}^{k,k'}(\mathbf{x}-\mathbf{v}s, \mathbf{v}\cdot\mathbf{v}')\,\mathrm{d}\mathbf{v}'$: a particle incoming with group $\mathbf{g}'$ outcomes with group g and sees its weight multiplied by $\sum_{k=0}^{P}|\mathbf{R}_{g,k}(\mathbf{v})|$.

Remark 3.4. *Note that when R is continuous through the cell interface, the above treatments degenerate toward the identity operator. In practice, in order to lighten the computations, the previous operations are triggered only when an interface between two different materials is crossed.*

In the next section, we sum-up the previous treatments/operations one must perform on an uncertain MC particle in order to consistently solve equation (19).

3.8. The algorithmic sketch of the multigroup-like MC-gPC resolution schemes

In this section, we sum-up the results of the previous sections concerning the semi-analog and non-analog MC schemes for the multigroup-like linear Boltzmann equation (19) in an algorithmic sketch and comment it. We also compare the next algorithms for multigroup-like MC-gPC to the ones implementing (the former version of) MC-gPC (which is recalled in Appendix C). The algorithmic sketches are for a *backward* resolution.

Remark 3.5. *The backward forms of the previous sections are convenient in order to identify the probability measures used for the treatment of each MC particles. The probability measures for a* direct *resolution can be built from* Λ_s *(see [61])* via *the construction of*

$$v\Lambda_S(\mathbf{x},t,\mathbf{v},g)P_S(\mathbf{x},t,\mathbf{v}\cdot\mathbf{v}',g\cdot g') = v'\Lambda_s(\mathbf{x},t,\mathbf{v}',g')P_s(\mathbf{x},t,\mathbf{v}\cdot\mathbf{v}',g\cdot g'),$$

and its use, instead of Λ_s*, in the different samplings described in the next algorithms.*

Function *backward_multigroup-like_MC-gPC_tracking()*

for $k \in \{0, ..., P\}$ **do**
set $v_k(\mathbf{x}, t, \mathbf{v}) = 0$
set $v_k^2(\mathbf{x}, t, \mathbf{v}) = 0$
end
for $p \in \{1, ..., N_{MC}\}$ **do**
set $s_p = t$ *#this will be the life time of particle p*
set $\mathbf{x}_p = \mathbf{x}$
set $\mathbf{v}_p = \mathbf{v}$
set $g_p = g$
set i_p=find_cell($\mathbf{x}_p$)
set $w_p(t) = \dfrac{1}{N_{MC}}$
while $s_p > 0$ **do**
if $\mathbf{x}_p \notin \mathcal{D}$ **then**
#here a general function for the application of arbitrary boundary conditions
apply_boundary_conditions($\mathbf{x}_p, s_p, \mathbf{v}_p, g_p$)
end
$\tau_{census} = s_p$
τ_{exit} = compute_time_until_cell_exit($\mathbf{x}_p, \mathbf{v}_p, g_p$)
τ_{inter} = sample_interaction_time($i_p, \mathbf{v}_p, g_p$)
$\tau = \min(\tau_{inter}, \tau_{exit}, \tau_{census})$
#set the life time of particle p to:
$s_p \longleftarrow s_p - \tau$
#move the particle p
$\mathbf{x}_p \longleftarrow \mathbf{x}_p + \mathbf{v}_p\tau$,
K = compute_weight_modif($\mathbf{x}_p$, $\mathbf{v}_p$, τ, τ_{census}, τ_{exit}, τ_{inter}, i_p, g_p)
$w_p \longleftarrow K \times w_p$
if $\tau == \tau_{census}$ **then**
#tally the contribution of particle p for the first and second moments
$v_{g_p}(\mathbf{x}, t, \mathbf{v}) += w_p \times v_0(\mathbf{x}_p, \mathbf{v}_p, g_p)$
$v^2_{g_p}(\mathbf{x}, t, \mathbf{v}) += w_p \times v_0^2(\mathbf{x}_p, \mathbf{v}_p, g_p)$
end
if $\tau == \tau_{exit}$ **then**
 # find the new cell in which particle p will evolve
$i_p^{\text{old}} = i_p$
$i_p \longleftarrow$ find_adjacent_cell(i_p^{old}),
#Multiply the weight of the particle according to g_p *and* i_p, i_p^{old}
$w_p \leftarrow w_p \times \sum_{k=0}^{P} \mathbf{R}_{g_p,k}^{i_p,i_p^{\text{old}}}(\mathbf{v}_p)$
#Sample the group $\mathbf{g}'$ *of particle p from* $\mathcal{B}$ *given by (45)*
$\mathbf{g}'$ =sample_uncertain_group($\mathbf{R}^{i_p,i_p^{\text{old}}}$, $\mathbf{v}_p$, g_p)
#Multiply the weight of the particle according to g_p *and* $\mathbf{g}'$
$w_p \leftarrow w_p \times sgn\left(\mathbf{R}_{g_p,\mathbf{g}'}^{i_p,i_p^{\text{old}}}(\mathbf{v}_p)\right)$
$g_p \longleftarrow \mathbf{g}'$
end
if $\tau == \tau_{inter}$ **then**
#Sample the group $\mathbf{g}'$ *of particle p from* $G_s(\mathbf{x}_p, \mathbf{v}_p, g_p, g')\,\mathrm{d}g'$
$\mathbf{g}'$ =sample_uncertain_group($\Lambda_s^{i_p}$, $\mathbf{x}_p$, $\mathbf{v}_p$, g_p)
#Multiply the weight of the particle according to g_p *and* $\mathbf{g}'$
$w_p \leftarrow w_p \times \gamma^{g_p,\mathbf{g}'}(\mathbf{x}_p, \mathbf{v}_p)$
#Sample the velocity $\mathbf{V}'$ *of particle p from* $P_{\Lambda_s}^{g_p,\mathbf{g}'}(\mathbf{x}_p, \mathbf{v}_p \cdot \mathbf{v}')\,\mathrm{d}\mathbf{v}'$
$\mathbf{V}'$ =sample_velocity($\mathbf{x}_p$, $\mathbf{v}_p$, g_p, $\mathbf{g}'$)
$g_p \leftarrow \mathbf{g}'$
$\mathbf{v}_p \leftarrow \mathbf{V}'$
end
end
end
 # remember we have $V = (v_0, ..., v_k)^t$
$U = \mathbf{R}^{-1}V$
 # remember we have $V^2 = (v_0^2, ..., v_k^2)^t$ *and* $\sigma^2_{MC} = (\sigma^2_{0,MC}, ..., \sigma^2_{P,MC})^t$
$\frac{1}{\sqrt{N_{MC}}}\sigma^2_{\text{MC}} = \mathbf{R}^{-1}V^2\mathbf{R} - U^2$
end

Algorithm 1: The semi-analog and non-analog multigroup-like MC-gPC schemes described in term of algorithmic operations in order to compute (adjoint) $V(\mathbf{x}, t, \mathbf{v})$ and its vector of asymptotical variances σ^2_{MC}.

The tracking phase allowing to solve (19) is described in algorithm 1. It describes the 'tracking' of an uncertain population of particles within the simulation domain $\mathcal{D}$. In order to present both implementations (of the semi-analog and non-analog multigroup-like MC schemes) in the same general framework/code, we encapsulated some key parts of the resolution in several functions: sample_interaction_time, compute_weight_modif, sample_uncertain_group, sample_velocity[16]. The four latter key functions are described in algorithms 2–3–4–5 but for the moment let us focus on the common canvas (i.e. algorithm 1).

Function *sample_interaction_time(int i, real* $\mathbf{v}$*, int g)*
  set τ = REAL_MAX
  U =sample_uniform_law()
  if $MC\_scheme == semi - analog$ **then**
    $\tau = -\dfrac{\ln(U)}{v\Lambda_t^i(\mathbf{v}, g)}$
  end
  if $MC\_scheme == non - analog$ **then**
    $\tau = -\dfrac{\ln(U)}{v\Lambda_s^i(\mathbf{v}, g)}$
  end
  return τ
end

Algorithm 2: The sampling of the interaction time function depending on the choice of the MC scheme. The cross-sections are assumed constant in cell i.

Function *sample_velocity(real* $\mathbf{x}$*, real* $\mathbf{v}$*, int g, int* g'*)*
  $\mathbf{V}'$ =sample_from_$P_{\lambda_s}^{g,g'}(\mathbf{x}, \mathbf{v}, g, g')$
  return $\mathbf{V}'$
end

Algorithm 3: Sampling of the velocity

In algorithm 1, we can see that each presented scheme relies on comparing three times, τ_{inter} the interaction time, τ_{exit} the time at which an MC particle p would get out of the cell i_p, τ_{census} the time before ending the time step. For each scheme, the particle moves along $\mathbf{v}_p\tau$ where τ is the minimum of the three above times. Its weight is modified or not (in compute_weight_modif) depending on the scheme. Furthermore, depending on the minimum of $\tau_{census}, \tau_{exit}, \tau_{inter}$, the particle sees its life time updated and finishes its treatment (*census*) or crosses the interface between two cells (*exit*) or encounters an interaction (*inter*). In the two latter cases, its group and weight may change. In the last one, its velocity may also change. All the samplings potentially depends on the uncertain field g_p carried out by the uncertain MC particle p. The first and second order moments of the gPC coefficients are computed during the MC resolution. The instrumentation of the tracking corresponds to the tallying phases (i.e. the $+=$ operations in algorithm 1).

Let us now focus on the encapsulated functions. First, note that they all only depend on particle fields $(\mathbf{x}_p, \mathbf{v}_p, i_p, g_p, ...)$. The first function, used to sample the interaction time, only needs the cell in which the particle evolves i_p, particle energy $\mathbf{v}_p$ and the uncertain one g_p and is detailed in

[16] We do not detail the functions compute_time_until_cell_exit, find_cell and find_adjacent_cell as they depend more on the type of grid (cartesian, structured, unstructured) than on the MC resolution scheme.

algorithm 2 (which is valid for a constant per cell cross-section). Depending on the chosen scheme, the interaction time is sampled from the total cross-section Λ_t (semi-analog) or from the scattering one Λ_s in the current cell i_p. Both are obtained inversing the cumulative density function of an exponential law.

The second encapsulated function corresponds to the modification of the weight of the particle, detailed in algorithm 5. For this function, the event the particle encounters explicitly appears in the treatment. The non-analog scheme is the only one having a treatment independent of the event. The weight of a particle remains unchanged for the semi-analog schemes for the *census* and *cell exit* events. It changes in the case of an interaction: for the semi-analog scheme, the weight is multiplied by the probability of being scattered $\frac{\Lambda_s}{\Lambda_t}$.

Function *sample_uncertain_group(Matrix* Λ*, real* $\mathbf{x}$*, real* $\mathbf{v}$*, int g)*

#Sample a uniform random variables in $[0,1]$

$U =$sample_uniform_law()

#Sample from G_s *given by* (31) *or* $\mathbf{R}$ *given by* (45) *depending on argument* Λ *:*

$$\mathbf{g}' = \min_{h\in\{0,\dots,G\}} \left\{ U < \frac{\sum_{g'=0}^{h} \lambda^{g,g'}(\mathbf{x},\mathbf{v})}{\sum_{g'=0}^{P} \lambda^{g,g'}(\mathbf{x},\mathbf{v})} \right\}$$

return $\mathbf{g}'$

end

Algorithm 4: The sampling of the uncertain group at the collision (G_s) or at the interface between two materials (in this case, G_s must be replaced by $\mathbf{R}$).

Function *compute_weight_modif(real* v*, real* $\tau_{\min}$*, real* τ_{census}*, real* τ_{exit}*, real* τ_{inter}*, int i, int g)*

set $K = 1$

if $MC\_scheme == semi-analog$ **then**

  if $\tau_{\min} == \tau_{exit}$ *or* $\tau_{\min} == \tau_{census}$ **then**

    $K = 1$

  end

  if $\tau_{\min} == \tau_{inter}$ **then**

    $$K = \frac{\Lambda_s^i(\mathbf{v},g)}{\Lambda_t^i(\mathbf{v},g)}$$

  end

end

if $MC\_scheme == non-analog$ **then**

  $K = e^{-v(\Lambda_t^i(\mathbf{v},g)-\Lambda_s^i(\mathbf{v},g))\tau_{\min}}$

end

return K

end

Algorithm 5: The weight modification depending on the MC scheme

At the interaction time, each scheme needs the sampling of the outer velocity $\mathbf{V}'$ and outer uncertain group $\mathbf{g}'$, summed up in algorithm 3 and 4. We do not spend time commenting algorithm 3, its details mainly depend on the format of the velocity scattering file/data and is classical in MC codes. The uncertain group scattering algorithm 4 is also classical in the sense that it

only corresponds to the sampling from a discrete probability measure. But remains singular as it corresponds to a change of *uncertain* group. Besides, this change of uncertain group may occur at the interface between two materials of different uncertain cross-sections. For both events (scattering or cell exit), the weight of the uncertain MC particle may change sign *via* the multiplication by $\gamma^{g_p,\mathbf{g}'}$ or $sgn(\mathbf{R}^{i_p,i_p^{\text{old}}}_{g_p,\mathbf{g}'})$ which, we recall are either equal to 1 or -1, see (22) and (46).

In algorithm 1, time steps are explicitly detailed but for the linear Boltzmann equation, time steps may coincide with the times of interest (MC methods are inconditionally stable for the linear Boltzmann equation). In other words, if one is only interested in time T, it is possible choosing $\Delta t = T$. This is not the case in [8, 9] in which the coupling with additional equations induces restrictions on the time step. Note that the above algorithm description still applies for the MC resolution of the linear Boltzmann equation coupled to other equations but may need additional instrumentations (track length estimator for example).

The algorithmic sketches for the multigroup-like MC-gPC solvers can now be compared to the ones of the classical MC-gPC ones (see [1, 10] for example). These are reminded in Appendix C. First, there are similarities between the multigroup-like MC-gPC (algorithms 1–2–3–4–5) and the MC-gPC ones (algorithms 6–7–8–9):

- for each MC-gPC versions, some arrays of size $P+1$ must be initialised and used to store the contributions of the uncertain MC particles. This is mainly because both versions of MC-gPC solve system (11).
- Besides, each time a call to the cross-sections is made, an additional field relative to the uncertain dimension is needed. For MC-gPC , it corresponds to $\mathbf{X}_p$, for the multigroup-like coounterpart, it corresponds to g_p.
- Furthermore, except from the fact that the calls depend on σ_s, σ_t for MC-gPC and on Λ_s, Λ_t for its multigroup counterpart, the operations are relatively similar (see the exponential samplings of algorithms 2 and 7 for example, or the computations in algorithms 5 and 9).

But there are also important differences (which are all highlighted in cyan in algorithm 1):

- The new multigroup-like MC-gPC solves the system of unknown $V = (v_0, ..., v_P)^t$ whereas MC-gPC solves the system of unknown $U = (u_0, ..., u_P)^t$. The new MC-gPC solver consequently needs a matrix multiplication $U = \mathbf{R}^{-1}V$ at the end of the time step $[0, t]$ in order to come back to unknown U.
- The new multigroup-like MC-gPC solver needs one additional sampling: the one for the change of uncertain group of algorithm 4 (when the particle encounters an interaction with matter or crosses an interface between two cells of different uncertain cross-sections).
- The weight may need an additional correction in case $\gamma^{g_p,\mathbf{g}'}$ or $sgn(\mathbf{R}^{i_p,i_p^{\text{old}}}_{g,\mathbf{g}'})$ is negative.
- In the new MC-gPC version, each uncertain MC particle p needs to keep its *uncertain group* g_p as a field whereas MC-gPC needs a vector $\mathbf{X}_p$ of sample of the uncertain parameters. The new version allows a gain in terms of memory consumption per MC particle, especially in high uncertain dimensions.

– In the tallying phase (see the lines with the + = operation), the term ϕ_k explicitly appears in the score of each uncertain MC particle for the MC-gPC algorithm 7. This term has been identified in section 1 as the one responsible for the excess of variance of MC-gPC : it does not appear for the multigroup-like one (see algorithm 2).

– The tallying phase does not anymore depend on a *for* loop: the uncertain MC particle only contributes to its own final group g_p. This considerably decreases the cost of this phase, especially in high uncertain dimensions (i.e. for $1 \ll P$).

The two last points are the ones which make the multigroup-like MC-gPC solver more efficient than MC-gPC. In the next section, we consider several benchmarks and verify the new solver ensures gains in terms of asymptotical numerical variance/error and of run-times.

4. Numerical results

In this section, we revisit several test-cases of papers [10, 1, 9] with the multigroup-like MC-gPC solvers described in the previous sections. In particular, we begin by some simple problems (sections 4.1–4.2) from [10] for which the excess of variance of the MC scheme of MC-gPC presented in [1] has been put forward. We verify that the new multigroup-like MC-gPC solver of section 3 does have a smaller asymptotically numerical variance (cf. the $\sigma_{k,\mathrm{MC}}$ term in expression (6)) than its previous versions and even compare it to some non-intrusive strategies. We also introduce new test-cases (see sections 4.3 and 4.4), specifically designed to easily verify the group scattering implementation (see algorithm 4) of the new solver and highlight the importance of the material of section 3.7. We then revisit (section 4.5) a test-problem from [1] for which MC-gPC already ensured gains. The new MC-gPC solver is even more efficient. We finally consider an eigenvalue/eigenvector problem (section 4.6) from [9]. This latter test-problem allows verifying the new MC-gPC solver can still efficiently handle k_{eff} computations (i.e. nonlinear/time dependences within the cross-sections).

Note that in the following test-cases, independent uniform random variables $\mathbf{X}$ are considered for the modeling of the input uncertain parameters (without loss of generality). The gPC framework implies using the orthonormal Legendre polynomials for the basis $(\phi_k(\mathbf{X}))_{k\in\mathbb{N}}$ in the next test-cases.

4.1. The collisional regime

In this section, we revisit one test-case from [10] on which the excess of variance of MC-gPC (with respect to ni-gPC) has been put forward. The configuration is homogeneous, monokinetic, the uncertainty affects the fission cross-section σ_f. The details of the test-problem are given in Appendix B. This regime corresponds to the case of an infinite medium with constant cross-sections with respect to time, space and energy. With these assumptions, the transport equation (1) resumes to $\forall k \in \{0, ..., P\}$

$$\begin{aligned}\partial_t \iint u(t,\omega,\mathbf{X})\phi_k(\mathbf{X})\,\mathrm{d}\omega\,\mathrm{d}\mathcal{P}_{\mathbf{X}} = \quad & -v\int \sigma_t(\mathbf{X})\int u(t,\omega,\mathbf{X})\,\mathrm{d}\omega\phi_k(\mathbf{X})\,\mathrm{d}\mathcal{P}_{\mathbf{X}} \\ & +v\int \sigma_s(\mathbf{X})\int P_s(\omega',\omega,\mathbf{X})u(t,\omega',\mathbf{X})\,\mathrm{d}\omega'\phi_k(\mathbf{X})\,\mathrm{d}\mathcal{P}_{\mathbf{X}}.\end{aligned} \tag{47}$$

From the definition of P_s ensuring[17] $\forall \omega \in \mathbb{S}^2, \forall \mathbf{X} \in \Omega, \int P_s(\omega', \omega, \mathbf{X})\, \mathrm{d}\omega' = 1$, it even simplifies to the classical ODE

$$\partial_t U(t) + v\Sigma_t U(t) = v\Sigma_s U(t), \tag{48}$$

where $U(t) = (U_0(t), ..., U_P(t))^T$ with $U_k(t) = \int u(t, \omega, \mathbf{X})\phi_k(\mathbf{X})\, \mathrm{d}\omega\, \mathrm{d}\mathcal{P}_{\mathbf{X}}, \forall k \in \{0, ..., P\}$ and Σ_t, Σ_s defined as in section 3. Now, if we apply the material of section 3 to the above equation and introduce the diagonal matrix Λ_t such that $\Lambda_t = R^{-1}\Sigma_t R$ and Λ_s such that $\Lambda_s = R^{-1}\Sigma_s R$, we get

$$\partial_t R^{-1}U(t) + v\Lambda_t R^{-1}U(t) = v\Lambda_s R^{-1}U(t).$$

The equation satisfied by $V(t) = R^{-1}U(t)$ is given by

$$\partial_t V(t) = -v\Lambda_t V(t) + v\Lambda_s V(t). \tag{49}$$

Now, assume that only the fission cross-section σ_f is uncertain: in neutronics for example, this means that $\sigma_t(\mathbf{X}) = \sigma_a + \sigma_s + \sigma_f(\mathbf{X})$ and that $\sigma_s(\mathbf{X}) = \sigma_s + \nu_f\sigma_f(\mathbf{X})$ where ν_f is the multiplicity of the fission reaction. In this case, both Λ_t and Λ_s can be diagonalised in the same basis. We can

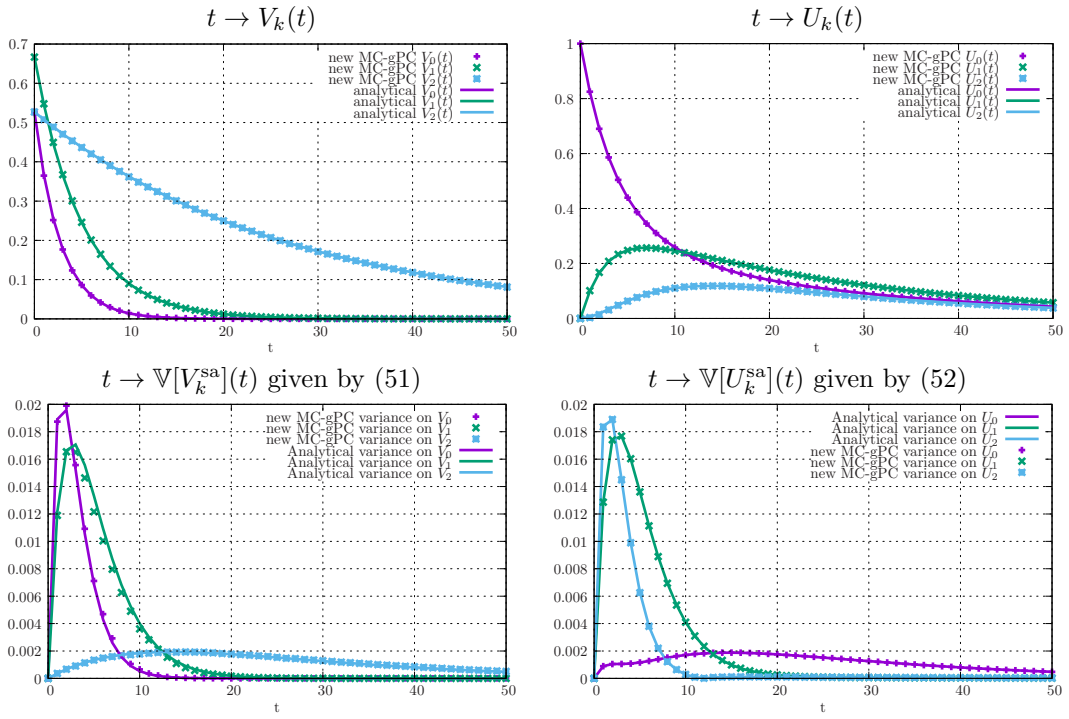

Figure 2: Time evolutions of $t \to V_k(t)$ (top-left), $t \to U_k(t)$ (top-right), $t \to \mathbb{V}[V](t)$ given by (51) (bottom-left) and $t \to \mathbb{V}[U](t)$ given by (52) (bottom-right) obtained with reference solutions and with the new MC-gPC strategy described in section 3.

[17] It only corresponds to a pretreatment of the cross-sections.

consequently write $\Lambda_\alpha = \text{diag}(\lambda_\alpha^0, ..., \lambda_\alpha^P), \forall \alpha \in \{s, t\}$. The solution $V = (V_0, ..., V_P)^t$ of (49) is then given by

$$V_k(t) = V_k^0 e^{-v(\lambda_t^k - \lambda_s^k)t}, \forall k \in \{0, ..., P\}, \tag{50}$$

together with $V_0 = (V_0^0, ..., V_P^0)^t = R^{-1}U_0$. Figure 2 (top-left) compares the analytical expression of $t \to V_k(t), k \in \{0, ..., P = 3\}$, see (50), to the one computed with a semi-analog MC scheme solving (17). The new MC-gPC solver solves the equation of unknown V and builds the solution U from V: the latter is given by $U(t) = RV(t)$. Figure 2 (top-right) compares the time evolutions of the gPC coefficients of U obtained with the new MC-gPC solver to the analytical expression of $U(t)$ obtained from (50). The results are in perfect agreement. Now, the material of paper [10] can be applied to the resolution of (49) in order to characterise the asymptotical variance of the semi-analog MC scheme for solving the equation of unknown V. It is given by (see [10] for more details)

$$\mathbb{V}[V_k^{\text{sa}}](t) = (V_k^0)^2 \left(e^{v \frac{(\lambda_s^k)^2 - (\lambda_t^k)^2}{\lambda_t^k} t} - e^{2v(\lambda_s^k - \lambda_t^k)t} \right), \forall k \in \{0, ..., P\}. \tag{51}$$

The asymptotical numerical variance $\mathbb{V}[U^{\text{sa}}]$ on the vector of gPC coefficients $U = (U_0, ..., U_P)^T$ can be deduced from the one $\mathbb{V}[V^{\text{sa}}]$ on the vector of gPC coefficients of V (whose components are given by (51)) using the classical probability result [62]

$$\mathbb{V}[U^{\text{sa}}](t) = \mathbb{V}[RV^{\text{sa}}](t) = R\mathbb{V}[V^{\text{sa}}](t)R^T. \tag{52}$$

Figure 2 (bottom-left) compares the results obtained with expression (51) and the variance computed by instrumenting the semi-analog MC solver on V (see the line about σ^2_{MC} in algorithm 1). Once again, the curves are in agreement. Finally, figure 2 (bottom-right) compares the analytical asymptotical variance on U to the one computed by instrumenting the MC-gPC resolution (i.e. with an estimator, see the second $+=$ term in algorithm 1). Once again, the curves are in agreement. With the results of figure 2, we consider the new semi-analog MC-gPC solver is verified (as in V&V for Verification & Validation, see [63]). This means we can now compare the asymptotical variances of the new MC-gPC solver with the one of MC-gPC and ni-gPC. But before, note that we can also solve the problem with a non-analog MC scheme: in this configuration, the numerical noise on V is zero (see [10]), $\mathbb{V}[V^{\text{na}}](t) = 0$, leading to a zero numerical noise also on U as $\mathbb{V}[U^{\text{na}}](t) = R\mathbb{V}[V^{\text{na}}]R^T = 0$. We do not display the verification plots for the non-analog solver but ensure that we recover a zero variance on this problem (as soon as there is one uncertain MC particle per group).

Now that our solvers are verified, let us focus on their different asymptotical variances with respect to time for the configuration of this section. Figure 3 compares the asymptotical variances for several resolution strategies

- semi-analog MC-gPC on U (as in [10]),
- non-analog MC-gPC on U (as in [10]),
- semi-analog ni-gPC (as in [10]),
- non-analog ni-gPC (as in [10]),

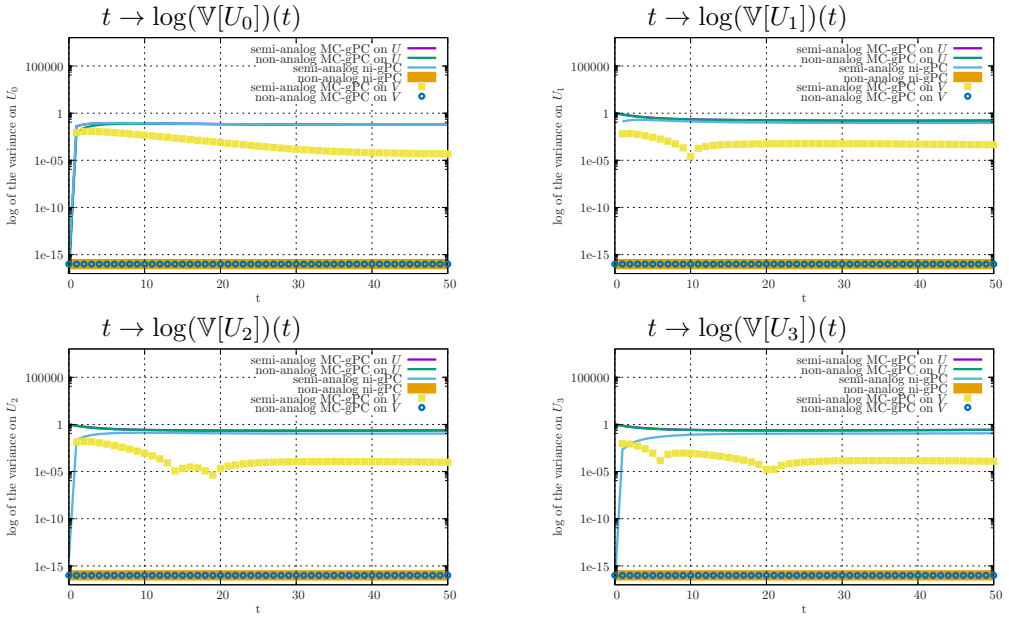

Figure 3: Time evolutions of the logarithm of $t \to \mathbb{V}[U_k](t)$ for $k \in \{0, 1, 2, 3\}$ for semi-analog MC-gPC on U, non-analog MC-gPC on U, semi-analog ni-gPC, non-analog ni-gPC, semi-analog MC-gPC on V, non-analog MC-gPC on V.

- semi-analog multigroup-like MC-gPC (on V, as in section 3),
- non-analog multigroup-like MC-gPC (on V, as in section 3),

on the gPC coefficients $(U_k)_{k\in\{0,...,P=3\}}$. With figure 3, we can see that:

- the MC-gPC strategies solving directly the equations on U are the noisiest ones (as already seen in [10]). In particular, we can see that at time $t = 0$ for which the problem is deterministic, the variance on the gPC coefficients is non zero as soon as $k > 0$. This is typically due to the ϕ_k^2 term, in red in expression (10) and in algorithm 6, appearing $\forall k \in \{1, ..., P\}$.
- Having a semi-analog or a non-analog MC-gPC solver on U does not lead to significant improvements (as already seen in [10]), the curves are almost overimposed with this logarithmic scale.
- The semi-analog ni-gPC solver only shows slightly better results than MC-gPC on this problem.
- The non-analog ni-gPC has the best performance (in terms of asymptotical variance but not in terms of run-times, see [10]) for this problem with a zero asymptotical error.
- The new MC-gPC strategies (i.e on V) are much less noisy than the classical MC-gPC ones. In particular, the error is zero at time $t = 0$ with the semi and non-analog multigroup-like MC schemes solving (17). Note that ϕ_k is not involved in the estimators (see the $+=$ terms in algorithm 1).

– The multigroup-like non-analog MC-gPC on V is much less noisy than the semi-analog MC-gPC one on V.

– The semi-analog multigroup-like MC-gPC on V is $\times 10^2$ to $\times 10^5$ much less noisy than the MC-gPC solvers and the semi-analog ni-gPC solver.

– The non-analog MC-gPC on V have the *same (optimal) performances* as the non-analog ni-gPC solver, with a zero asymptotical variance on this homogeneous configuration. On this problem, the non-analog MC-gPC solver on V is also 5 *times faster* than the ni-gPC one.

In this collisional regime, the new MC-gPC solvers, based on the resolution of the multigroup equation (17) of unknown V considerably improve the numerical noise. The solvers are now better in terms of noise than the less noisy strategies together with relying on only one run of an MC code. In particular, as can be seen on the pictures of figure 3, the new MC-gPC solvers do not anymore display an excess of variance for early times (i.e. in the deterministic case): the new MC-gPC solvers need as many (non-analog) or even less (semi-analog) MC particles as the ni-gPC resolutions. In other words, drawback $*_3$ emphasized in section 2 and figure 1 has been corrected (at least in this collisional regime). More detailed performance comparisons will be made in the next sections. In the next paragraph, we focus once more on a particular regime (free-flight).

4.2. The free-flight regime

In this section, we consider the free flight regime. It corresponds to the particular case where $\sigma_t = \sigma_s = 0$. In this regime, the non-analog and the semi-analog MC schemes are equivalent. In the uncertain free flight regime, (1) degenerates toward

$$\left\{ \begin{array}{l} \partial_t u(\mathbf{x}, t, \mathbf{v}, \mathbf{X}) + \mathbf{v}\partial_{\mathbf{x}} u(\mathbf{x}, t, \mathbf{v}, \mathbf{X}) = 0, \\ u(\mathbf{x}, 0, \mathbf{v}, \mathbf{X}) = u_0(\mathbf{x}, \mathbf{v}, \mathbf{X}), \\ u(\mathbf{x}, t, \mathbf{v}, \mathbf{X}) = u_b(t, \mathbf{v}, \mathbf{X}), \mathbf{x} \in \partial\mathcal{D}, \quad t \in [0, T], \quad \mathbf{v} \cdot n_s < 0, \quad \mathbf{X} \sim \mathrm{d}\mathcal{P}_{\mathbf{X}}. \end{array} \right. \tag{53}$$

In this case, the solution at time t is made of a contribution of the initial condition together with the contribution of the boundary one [16, 17, 19]. In order to ease the computation, let us consider an infinite medium so that the solution of (53) resumes to

$$u(\mathbf{x}, t, \mathbf{v}, \mathbf{X}) = u_0(\mathbf{x} - \mathbf{v}t, \mathbf{v}, \mathbf{X}), \forall \mathbf{x} \in \mathcal{D}, \mathbf{v} \in \mathcal{V}, t \in [0, T], \mathbf{X} \sim \mathrm{d}\mathcal{P}_{\mathbf{X}}. \tag{54}$$

Let us compute the gPC coefficients of the above solution with different MC strategies, for initial condition $u_0(x, X) = U_0(X)\mathbf{1}_{[-0.1,0.1]}(x)$ with $U_0(X) = \overline{U}_0 + \hat{U}_0 X$ with $\overline{U}_0 = 1$, $\hat{U}_0 = \frac{1}{2}$ and $X \sim \mathcal{U}([-1, 1])$.

Figure 4 presents the results in terms of spatial profiles obtained by different resolution strategies in the aforementioned configuration. Figure 4 (top-left) presents the spatial profiles $x \to U_k(x, t^*)$ for final time $t^* = 1$: every numerical strategies (MC-gPC, ni-gPC and the new MC-gPC one) allow capturing the gPC coefficients solution of the problem. Figure 4 (top-right) displays two different results obtained by two different ways to estimate the numerical noise on the gPC coefficients with the new MC-gPC solver:

– first, thanks to the MC estimator instrumenting every MC resolution (see the $+=$ operations in algorithms 1–6),

– Second, thanks to an estimation of the variance computed with $N_{\text{seed}} = 2000$ runs of $N_{MC} = 100$ particles, each initialised with a different seed.

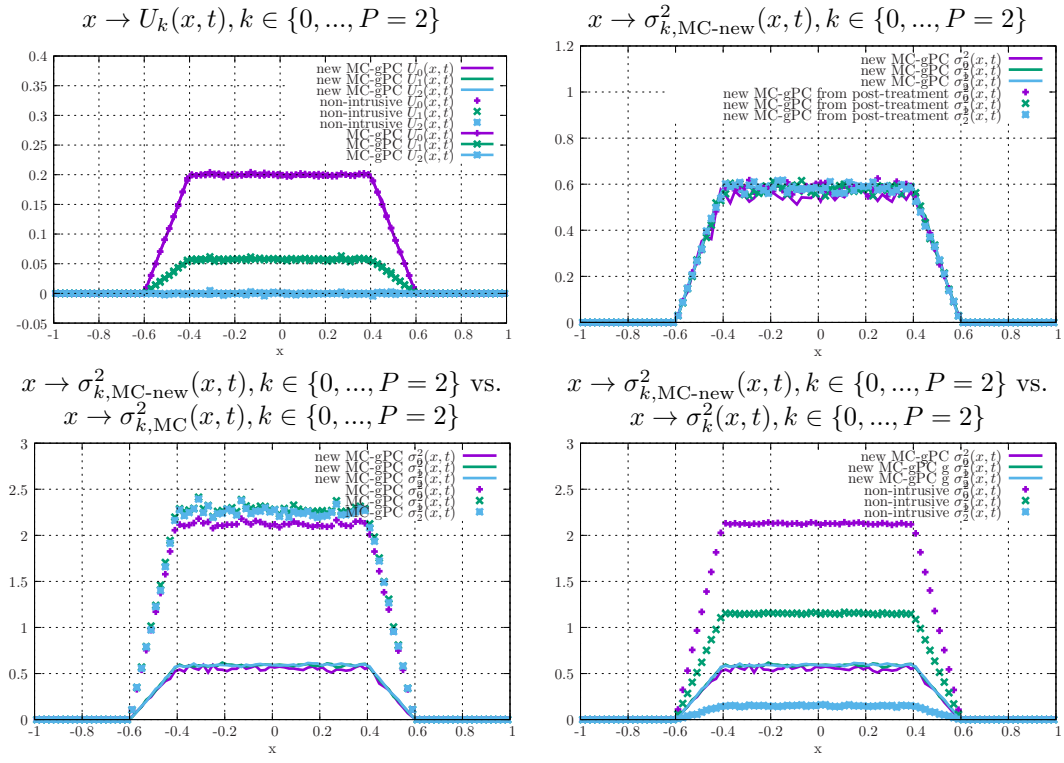

Figure 4: Spatial profiles $x \to \sigma^2_{k,\alpha}(t)$ for $k \in \{0,1,2,3\}$ for $\alpha \in \{$ semi-analog MC-gPC on U, non-analog MC-gPC on U, semi-analog ni-gPC , non-analog ni-gPC, semi-analog MC-gPC on V, non-analog MC-gPC on V $\}$.

The good agreement between the two estimations tends to verify our implementation.

Figure 4 (bottom-left) compares the numerical noise obtained with MC-gPC and the new multigroup-like MC-gPC solver: even in the free-flight regime, the new MC-gPC solver allows a gain with respect to MC-gPC on the numerical noise. The new MC-gPC is about 5 times less noisy than MC-gPC. Note that the two variants of MC-gPC have one common point: the numerical noise is the same on every gPC coefficient for this problem. Finally, figure 4 (bottom-right) compares the numerical noise of ni-gPC to the one of the new MC-gPC solver: the new MC-gPC solver is 5 times and 2 times less noisy than ni-gPC on, respectively, the two first gPC coefficients. But it is 4 times noisier on the third gPC coefficient, in this particular regime.

With the two previous sections, we focused on verifying that the new MC-gPC solvers ensure the expected gain in terms of asymptotical variances/errors. In the next sections, we show that additionally to being less noisy, the new MC-gPC solvers are less sensitive to the curse of dimensionality: the new MC-gPC are consequently even less computationally intensive than MC-gPC, which already presented important gains with respect to ni-gPC on many applications [1, 9, 8].

4.3. Uncertain scattering opacity: a verification (V&V) test case for (31)

The test-cases of the two previous sections allowed verifying our new MC-gPC solvers ensure a gain in terms of variance reduction. The test-case of this section has a more practical aim: we are

aware that any new implementation needs to be verified (as in verification & validation, see [63]). This test-case considerably eases the verification of the implementation of the group scattering term (31) of algorithm 4. Let us explain how: assume, as in section 4.1, that we are in an infinite medium of deterministic total opacity σ_t and unknown scattering one $\sigma_s(\mathbf{X})$. Furthermore, random variable $\mathbf{X} = X$ (1D) is uniformly distributed in $[-1, 1]$ so that our problem resumes to solving the uncertain ODE:

$$\begin{cases} \partial_t u(t, X) + v\sigma_t u(t, X) = v\sigma_s(X)u(t, X), \\ U(t = 0, X) = U_0 = 1. \end{cases} \tag{55}$$

In this case, we have $\Sigma_t = \Lambda_t = \sigma_t I$ where I is the identity matrix of size $P + 1$ $\forall P \in \mathbb{N}$. We consequently also have $R = R^{-1} = I$, independently of the order of the polynomial truncation. As a consequence, for this test-case,

$$\Lambda_s = R^{-1}\Sigma_s R = \Sigma_s = \left[\Sigma_s^{i,j}\right] = \left[\int \sigma_s(X)\phi_i(X)\phi_j(X)\,\mathrm{d}\mathcal{P}_X\right].$$

We also have $V = R^{-1}U = U$. For this test-case, the $P-$truncated gPC reduced model of (55)

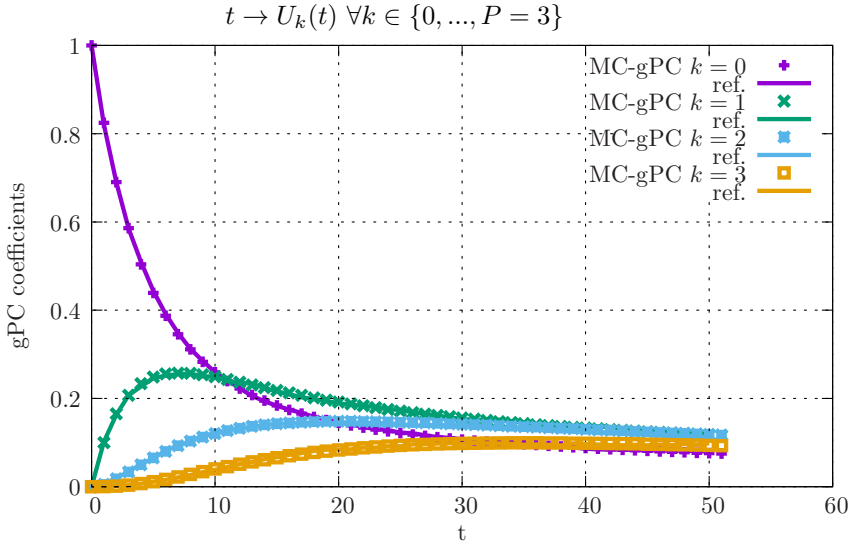

Figure 5: Time evolutions of the gPC coefficients $t \to u_k(t), k \in \{0, ..., P = 3\}$ for problem (55).

resumes to

$$\begin{cases} \partial_t u_k(t) + v\sigma_t u_k(t) = v\sum_{i=0}^{P} \Sigma_s^{i,k} u_k(t), \forall k \in \{0, ..., P\}, \\ u_k(t = 0) = \delta_{0,k}, \forall k \in \{0, ..., P\}. \end{cases} \tag{56}$$

The above system can easily be solved numerically with a non-intrusive application or even with a simple explicit Euler time discretisation scheme.

On another hand, this same equation (56) can also be solved applying the material described in the previous section, relative to term (31). Indeed, it can easily be rewritten

$$\begin{cases} \partial_t u_k(t) + v\sigma_t u_k(t) = v\lambda_s^k \sum_{i=0}^{P} \frac{|\Sigma_s^{i,k}|}{\lambda_s^k} sgn(\Sigma_s^{i,k}) u_k(t), \forall k \in \{0, ..., P\} \\ u_k(t=0) = \delta_{0,k}, \forall k \in \{0, ..., P\}, \end{cases} \tag{57}$$

where $\lambda_s^k = \sum_{i=0}^{P} |\Sigma_s^{i,k}|$ and where $sgn(x) = \frac{x}{|x|}$ $\forall x \in \mathbb{R} \setminus \{0\}$. It allows building the probability density measure of equation (31) whose sampling is described in algorithm 4. Indeed, as can be seen in the above expression, at time $t = 0$, $u_0(t = 0) \neq 0$ and $u_k(t = 0) = 0, \forall k \in \{1, ..., P\}$. This means that for $t > 0$, when we have $u_k(t) \neq 0$ for $k > 0$, this is due to a change of group/gPC coefficient from a scattering from group 0 to group $k > 0$.

Figure 5 presents the time evolutions of the gPC coefficients $t \to u_k(t), k \in \{0, ..., P = 3\}$ for $\sigma_t = 1$, $\sigma_s = \overline{\sigma}_s + \hat{\sigma}_s X$ with $\overline{\sigma}_s = 0.8$ and $\hat{\sigma}_s = 0.21$. The reference has been computed using ni-gPC with 10 Gauss-Legendre points. The computations have $N_{MC} = 10^4$ MC particles. Once again, both solvers present an excellent agreement.

4.4. Two materials with both uncertain total cross-section (the case where $\mathbf{v} \cdot \nabla_{\mathbf{x}} R^{-1} RV \neq 0$)

In this section, we present a test-case with two layers of two different uncertain materials. For this problem, we have $\mathbf{v} \cdot \nabla_{\mathbf{x}} R^{-1} RV \neq 0$ at the interface between the two materials. This test-case allows testing and verifying the resolution strategy of section 3.7 (we insist on the fact that without the material of section 3.7, the results would not be in agreement with the references).

Let us assume the particles are monokinetic, i.e. $\sigma_\alpha(\mathbf{x}, \mathbf{v}, \mathbf{X}) = \sigma_\alpha(\mathbf{x}, \mathbf{X})$, $\forall \alpha \in \{s, t\}, \forall \mathbf{X} \sim \mathrm{d}\mathcal{P}_{\mathbf{X}}, \forall \mathbf{x} \in \mathcal{D} = [0, 1]$, with $v = 1$, and that the scattering cross-sections are deterministic and isotropic for both materials, i.e. $P_s(\mathbf{x}, \mathbf{v}, \mathbf{X})\,\mathrm{d}\mathbf{v} = \mathbf{1}_{\mathbb{S}^2}(\omega)\,\mathrm{d}\omega$ $\forall \mathbf{x} \in \mathcal{D}$, $\forall \mathbf{v} = v\omega = \omega \in \mathcal{V} = \mathbb{S}^2$. The uncertain parameters $\mathbf{X} = (X_0, X_1)$ control the fluctuations of the total cross-sections and we have

$$\begin{cases} \sigma_t^0(X_0) & = \overline{\sigma}_t^0 + \hat{\sigma}_t^0 X_0 = 1 + 0.4X_0, \text{ with } X_0 \sim \mathcal{U}([-1, 1]), \\ \sigma_t^1(X_1) & = \overline{\sigma}_t^1 + \hat{\sigma}_t^1 X_1 = 1 + 0.4X_1, \text{ with } X_1 \sim \mathcal{U}([-1, 1]), \\ \sigma_s^0 & = \overline{\sigma}_s^0 = 1.3, \\ \sigma_s^1 & = \overline{\sigma}_s^1 = 0.9. \end{cases} \tag{58}$$

The geometry consists in domain $x \in [0, 1]$ with specular boundary condition at $x = 0$ and vaccum one at $x = 1$. Material 0 occupies the first half of the domain, i.e. $x \in [0, \frac{1}{2}]$ while material 1 occupies the remaining half $x \in [\frac{1}{2}, 1]$. In other words, we have

$$\sigma_\alpha(x, \mathbf{X}) = \sigma_\alpha^0(X_0)\mathbf{1}_{[0, \frac{1}{2}]}(x) + \sigma_\alpha^1(X_1)\mathbf{1}_{[\frac{1}{2}, 1]}(x), \forall \alpha \in \{s, t\}.$$

Both materials have a relatively high probability of being multiplicative as we have

$$\mathbb{P}(\sigma_s^0 > \sigma_t^0(X_0)) = 0.875 \text{ and } \mathbb{P}(\sigma_s^1 > \sigma_t^1(X_1)) = 0.375.$$

The initial conditions is given by $u(x, t = 0, \omega, X) = u_0(x) = 1$: it is deterministic, isotropic and spatially homogeneous. Figure 6 compares the results obtained with (old) MC-gPC (of [1]) and the new multigroup-like MC-gPC of this paper. The comparisons are made on the spatial profiles of the mean $x \to \mathbb{E}[U](x, t^*)$ (figure 6-left) and variance $x \to \mathbb{E}[U](x, t^*)$ (figure 6-right) of

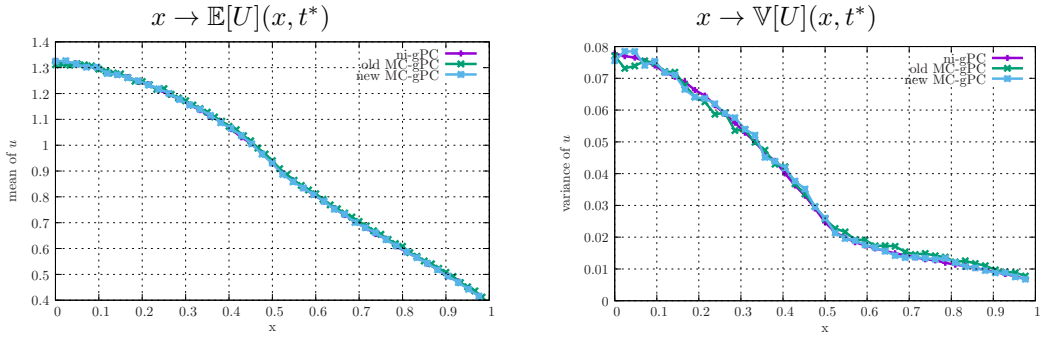

Figure 6: Left: spatial profiles of the mean $x \to \mathbb{E}[U](x,t^*)$ for $t^* = 1$ obtained from ni-gPC , (old) MC-gPC and from the new multigroup-like MC-gPC. Right: spatial profiles of the variance $x \to \mathbb{V}[U](x,t^*)$ for $t^* = 1$ obtained from ni-gPC , (old) MC-gPC and from the new multigroup-like MC-gPC.

$U(x,t^*,X) = \int u(x,t^*,\omega,X)\,\mathrm{d}\omega$ for $t^* = 1$. The computations have been carried out with $p_{1\mathrm{D}} = 2$ in each directions so that for every versions of gPC, i.e. we have $P = (2+1)^2 = 9$.

On figure 6-left, for the mean, we can see that the configuration, in the vicinity of $x = 0$ is globally mutiplicative: the mean density of particles is higher than its initial value of 1. On the second half of the domain $x \in [\frac{1}{2}, 1]$, the boundary condition tend to let particles get out of the domain. The variance (figure 6-right) is higher in the first half of the domain: this is closely related to the higher probability of being multiplicative within this part of the domain. The uncertainty considerably drops for $x \sim 1$.

Now, we can see on figure 6 that every gPC solvers are in agreement on both the mean and variance, which tends to verify the new solver. For this example, it is easy verifying that, for example for $p_{1\mathrm{D}} = 1$ in each direction, $\Sigma_t(x), x \in [0, \frac{1}{2}]$ and $\Sigma_t(x), x \in [\frac{1}{2}, 1]$ are not diagonalisable in the same basis. This also means that in order to obtain the results of figure 6, we had to consider discontinuous R^{-1} across the interface between the two material at $x = \frac{1}{2}$. In other words, for this test-case, the material of section 3.7 is activated each time an uncertain MC particle pass at $x = \frac{1}{2}$. We insist on the fact that without the material of section 3.7, the results would not be in agreement.

4.5. Uncertainty propagation and sensitivity analysis on a 3D uncertain problem

This example is a 3-dimensional stochastic (i.e. $Q = 3$) test-problem. The set-up (in terms of initial and boundary conditions) is the same as in [1] and is recalled below:

- monokinetic problem with $v = 1$, $x \in \mathcal{D} = [0,1]$, subdivided into $N_x = 20$ cells $\cup_{i=1}^{N_x} \mathcal{D}_i = \mathcal{D}$.
- Specular boundary condition on left (at $x = 0$) and vacuum one on the right hand side (at $x = 1$).
- Initially, the density of particles is homogeneous and deterministic, equal to 1, i.e. $u(x, t = 0, \omega, X) = u^0(x,\omega,X) = 1\ \forall x \in \mathcal{D}, \forall \omega \in \mathbb{S}^2$.
- The medium is homogeneous and considered uncertain. It depends on three parameters $\mathbf{X} = (X_1, X_2, X_3)$ affecting the microscopic total and scattering cross-sections and the material

density as

$$\begin{array}{lll} \sigma_t^m(x,t,\mathbf{X}) & = \sigma_t^m(X_1) & = \overline{\sigma}_t^m + \widehat{\sigma}_t^m X_1, \quad \forall x \in \mathcal{D}, t \in \mathbb{R}^+, \\ \sigma_s^m(x,t,\omega,\omega',\mathbf{X}) & = \sigma_s^m(X_2) & = \overline{\sigma}_s^m + \widehat{\sigma}_s^m X_2, \quad \forall x \in \mathcal{D}, t \in \mathbb{R}^+, \forall (\omega,\omega') \in \mathbb{S}^2, \\ \eta^m(x,t,\mathbf{X}) & = \eta^m(X_3) & = \overline{\eta}^m + \widehat{\eta}^m X_3, \quad \forall x \in \mathcal{D}, t \in \mathbb{R}^+, \end{array} \tag{59}$$

in which $\mathbf{X} = (X_1, X_2, X_3)$ are independent uniformly distributed random variables on $[-1,1]$, i.e. $\forall i \in \{1,2,3\}, X_i \sim \mathcal{U}([-1,1])$. The macroscopic cross-sections are defined, as in neutronics [18], by $\sigma_\alpha = \sum_{m=1}^M \sigma_\alpha^m \eta^m$ for M material. In this test-case, $M = 1$.

- For the next computations, the mean quantities are set to $\overline{\sigma}_t^m = 1.0$, $\overline{\sigma}_s^m = 0.9$, $\overline{\eta}^m = 1.0$ and the ones controling the variability to $\widehat{\sigma}_t^m = 0.4$, $\widehat{\sigma}_s^m = 0.4$, $\widehat{\eta}^m = 0.4$.
- We are interested in the mean $\mathbb{E}[U]$, variance $\mathbb{V}[U]$ profiles of $U(x,t,\mathbf{X}) = \int u(x,t,\omega,\mathbf{X})\,\mathrm{d}\omega$ at time $t = 1.0$.

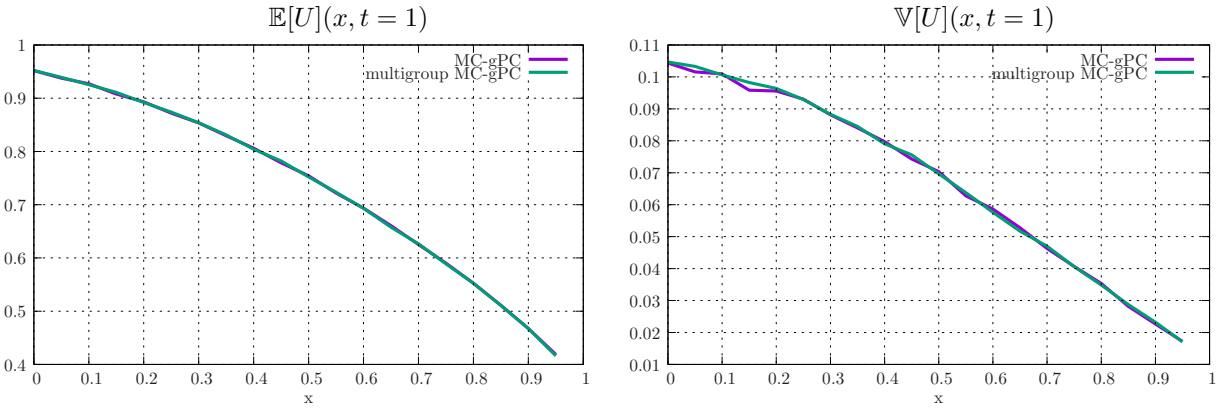

Figure 7: Comparison between MC-gPC and the new multigroup-like MC-gPC . Left: mean of $U(x,t=1,\mathbf{X})$ with respect to x. Right: variance of $U(x,t=1,\mathbf{X})$ with respect to x.

Figure 7 compares results obtained with MC-gPC and the new multigroup-like MC-gPC scheme for the aforementioned statistical outputs of interest. Before going through resolution strategy comparisons, let us present briefly the results: figure 7 (top) shows the mean of $U(x,t=1,\mathbf{X})$. Particles are globally absorbed in the vicinity of $x = 0$: indeed, we initially have $U(x,t=0,\mathbf{X}) = 1, \forall x \in \mathcal{D}$ and on figure 7 (top), we have $\mathbb{E}[U](x \sim 0, t=1) < \mathbb{E}[U](x \sim 0, t=0) = 1$. This averaged particle absorption occurs despite the probable multiplicative effect ($\sigma_s(\mathbf{X}) > \sigma_t(\mathbf{X})$ for some realisations of $\mathbf{X}$) of the medium. Particles are globally lost in the vicinity of $x = 1$, as $\mathbb{E}[U](x \sim 1, t=1) < \mathbb{E}[U](x \sim 1, t=0) = 1$, mainly due to the vacuum aboundary condition. Figure 7 (bottom-left) shows the variance of U. The uncertainty is more important in the vicinity of $x = 0$ and drops of a decade between $x = 0$ and $x = 1$.

Remark 4.1 (Few remarks).

- *Note that for this test-case, the group scattering matrix Λ_s is such that not all coefficients of the scattering matrix are positive. This means that the group scattering strategy presented in algorithm 4 is in a sense successfully verified with this test problem.*

– *Note also that the test-case is not exactly in the same conditions as in [1]: we here use less MC particles ($N_{MC} = 10^7$ instead of $N_{MC} = 32 \times 10^7$ and $N_x = 20$). These numerical parameters are chosen on purpose in order to highlight the gain in terms of MC noise for the new multigroup-like MC-gPC solver.*

– *Note also that the computations of this section have not been performed on the same computers as in [1], hence the small differences in run-times.*

On figure 7, the two resolutions are in agreement, even if they are not exactly of equivalent accuracies, whatever the statistical observable of interest (mean, variance). Each solution has been obtained with $N_{MC} = 10^7$ particles and $P = (p_{1\mathrm{D}} + 1)^Q = (2 + 1)^3 = 27$ coefficients. The new multigroup-like MC-gPC scheme solves the same set of equations as MC-gPC : we obtain accurate comparable solutions for low ($p_{1\mathrm{D}} = 2$, see [1] for more details on the fact the problem is already converged for such low polynomial order per direction) polynomial orders in every direction. The new MC-gPC scheme gives equivalent results with one run and $(P + 1)^Q = (2 + 1)^3 = 27$ gPC coefficients: it consequently also takes advantage of the fast convergence rate of gPC. The new solver is less noisy than the first MC-gPC version, this is especially observable on the variance (figure 7 right).

Second, let us discuss the average CPU times and costs of the two methods for (almost) equivalent accuracies:

– MC-gPC : cost = 1 × CPU time of 1 run $= 1 \times 6\,\mathrm{min}\,29s$.
– new MC-gPC : cost = 1 × CPU time of 1 run $= 1 \times 1\,\mathrm{min}\,25s$.
– ni-gPC : cost = 64 × CPU time of 1 run $= 64 \times 0\,\mathrm{min}\,54s = 58\,\mathrm{min}\,06s$.

Note that for the last line, for ni-gPC , the CPU time is averaged over the 64 non-intrusive runs. The above CPU times present a signicative differences deserving a careful study:

– first, one (old) MC-gPC run, for this particular set-up, costs about ×7 the average CPU time of the ni-gPC ones. This increase has been thoroughly studied in the previous MC-gPC papers [1, 8, 9] and has been explained by the number of tallies[18] one MC particle must perform. With MC-gPC, every MC particle must contribute to $P + 1$ cell-arrays. The main increase in computational time comes from the *tallying* and not the parallel reduction. This can take a considerable amount of time, especially if those tallies must be performed frequently (see [9, 8]).

– Now, the new multigroup-like MC-gPC of this paper does not need every MC particle to contribute to the P cell-arrays when tallying their respective scores. With the new solver, each MC particle only contributes to the cell-array of its group. As a consequence, the cost of the new multigroup-like MC-gPC solver is only ×1.5 with respect to one averaged ni-gPC run (for this test-case).

– Still, if we compare the total cost of an ni-gPC application, MC-gPC ensures a sequential gain of ×10 for this problem and the new multigroup-like MC-gPC solver ensures a sequential gain of ×40 (and a gain of ×4.5 with respect to MC-gPC).

– We finally recall that both MC-gPC solvers can benefit from the same parallel accelerations as classical MC codes (mainly domain replication, see [1, 35, 64, 65, 66, 37]).

[18] What we call a *tally* corresponds to the + = operations in algorithms 1–6.

With the three previous sections 4.1–4.2–4.3–4.4–4.5, we considered uncertain linear problems. With the next section, we want to emphasize the fact that the coupling of the new MC-gPC solver with nonlinear physics is still possible. We also insist on the differences in the implementation between MC-gPC and its new variant in this nonlinear context.

4.6. Eigenvalue/eigenvector problems: k_{eff} computations in neutronics

In this section, we revisit a test-case from [9] corresponding to an eigenvalue/eigenvector computation. Such kind of computation is of importance in epidemiology [32] or reactor physics [18] for a non-exhaustive list of applications. In such a problem, the iterative process involving the transport equation (1) and needed in order to estimate the eigenvalue/eigenvector leads to having a nonlinear dependence in the cross-section: for such problem, matrix Λ_s must be rebuilt in each material at each iteration.

4.6.1. Reminder of the material of [9] about MC-gPC and eigenproblems

The uncertain eigenvalue/eigenvector problem can be formalised as follows: find $k_{\text{eff}}(\mathbf{X})$ and $u(\mathbf{x}, t, \mathbf{v}, \mathbf{X})$ such that

$$\begin{aligned}\mathbf{v}\cdot\partial_{\mathbf{x}}u(\mathbf{x},\mathbf{v},\mathbf{X})+v\sigma_t(\mathbf{x},\mathbf{v},\mathbf{X})u(\mathbf{x},\mathbf{v},\mathbf{X}) = v\sigma_s(\mathbf{x},\mathbf{v},\mathbf{X})\int P(\mathbf{x},\mathbf{v}\cdot\mathbf{v}',\mathbf{X})u(\mathbf{x},\mathbf{v}',\mathbf{X})\,\mathrm{d}\mathbf{v}',\\ +\frac{v\nu_f(\mathbf{x},\mathbf{v},\mathbf{X})\sigma_f(\mathbf{x},\mathbf{v},\mathbf{X})}{k_{\text{eff}}(\mathbf{X})}\int P_f(\mathbf{x},\mathbf{v}\cdot\mathbf{v}',\mathbf{X})u(\mathbf{x},\mathbf{v}',\mathbf{X})\,\mathrm{d}\mathbf{v}',\end{aligned} \tag{60}$$

together with boundary conditions

$$u(\mathbf{x},\mathbf{v},\mathbf{X}) = u_b(\mathbf{v},\mathbf{X}),\quad \mathbf{x}\in\partial\mathcal{D}(\mathbf{X}),\quad \omega\cdot n_s(\mathbf{X})<0,\quad \mathbf{X}\sim\mathrm{d}\mathcal{P}_{\mathbf{X}}. \tag{61}$$

In the above problem, we are mainly interested in the statistics of $\mathbf{X}\to k_{\text{eff}}(\mathbf{X})$ and $\mathbf{X}\to u(\mathbf{x},\mathbf{v},\mathbf{X})$ (i.e. mean, variance, histogram, sensitivity indices [20] etc.) at specified locations $\mathbf{x}\in\mathcal{D}$ and velocities $\mathbf{v}\in\mathcal{V}$. In particular, we are interested in neutronics but many other physical applications could benefit the results of this paper (biology [30], socio-economics [6, 31, 4], epidemiology [32] etc.). Quantity k_{eff} is the first eigenvalue, and u, its corresponding eigenvector. The first eigenvalue k_{eff} physically corresponds to (for a non-exhaustive list of examples) the *effective multiplication factor* in neutronics [18]. It gives an estimation of the average number of neutrons coming out of a fission reaction. It is intensively used in reactor physics and in nuclear safety risk management [18]. In epidemiology, the same quantity is commonly called the *reproduction number* and is usually denoted by R_0, see [67]. It indicates how contagious an infectious disease is. It is an estimation of the average number of people who will contract a contagious disease from one person with that disease. Independently of the physics of interest, if $k_{\text{eff}}=1$, the environment (i.e. the spatial domain $\mathcal{D}$) is said *critical*. Nuclear reactors reproduce a critical environment to maintain stable chain reactions in order to produce energy. If $k_{\text{eff}}<1$, the environment is said *subcritical*. In this case, in epidemiology for example, each existing infection leads to less than one new one: the disease declines and dies out. In neutronics, care is taken to have subcritical environments for fuel transport or during storage. If $k_{\text{eff}}>1$, the environment is said *supercritical*. In this case, a disease spreads exponentially fast leading to an outbreak or an epidemic, a nuclear reactor may melt down. Being able to accurately estimate this eigenvalue (k_{eff} or R_0 etc. depending on the physics of interest) together with its uncertainties is consequently of primary importance.

The eigenproblem (60) is often solved thanks to a power iteration method [68, 69, 70, 71]. Let us begin by formally describing the general idea behind the algorithm: let us introduce the linear operators $L^{\mathbf{X}}$ and $F^{\mathbf{X}}$ defined by

$$\begin{cases} L^{\mathbf{X}}u = \mathbf{v}\cdot\nabla_{\mathbf{x}}u(\mathbf{x},\mathbf{v},\mathbf{X}) + v\sigma_t(\mathbf{x},\mathbf{v},\mathbf{X})u(\mathbf{x},\mathbf{v},\mathbf{X}) - v\sigma_s(\mathbf{x},\mathbf{v},\mathbf{X})\displaystyle\int P_s(\mathbf{x},\mathbf{v}\cdot\mathbf{v}',\mathbf{X})u(\mathbf{x},\mathbf{v}')\,\mathrm{d}\mathbf{v}', \\ F^{\mathbf{X}}u = \dfrac{v\nu_f(\mathbf{x},\mathbf{v},\mathbf{X})\sigma_f(\mathbf{x},\mathbf{v},\mathbf{X})}{k_{\text{eff}}(\mathbf{X})}\displaystyle\int P_f(\mathbf{x},\mathbf{v}\cdot\mathbf{v}',\mathbf{X})u(\mathbf{x},\mathbf{v}',\mathbf{X})\,\mathrm{d}\mathbf{v}', \end{cases}$$

and the boundary one defined by

$$B^X u = \left\{ u(\mathbf{x},\mathbf{v},\mathbf{X}) = u_b(\mathbf{v}\mathbf{X}), \quad \mathbf{x}\in\partial\mathcal{D}(\mathbf{X}), \quad \frac{\mathbf{v}}{v}\cdot n_s < 0, \text{ with } |\mathbf{v}| = v \right\}.$$

Then the power method consists in iterating on the fission part F of the linear operator and in looking for u the fixed point of

$$\begin{cases} L^{\mathbf{X}}u = \frac{1}{k_{\text{eff}}(\mathbf{X})}F^{\mathbf{X}}u, \\ B^{\mathbf{X}}u. \end{cases} \tag{62}$$

One possibility [9] consists in choosing the n^{th} iteration of the algorithm as the resolution of the instationary linear Boltzmann equation on a time step $[t^{n-1}, t^n = t^{n-1} + \Delta t]$:

$$\begin{cases} \partial_t u^n + L^{\mathbf{X}}u^n = \frac{1}{k_{\text{eff}}^{n-1}(\mathbf{X})}F^{\mathbf{X}}u^n, \\ u^0 = u^{n-1}, \\ B^{\mathbf{X}}u^n, \end{cases} \tag{63}$$

where

$$k_{\text{eff}}^n(\mathbf{X}) \quad = k_{\text{eff}}^{n-1}(\mathbf{X}) \times \frac{\int_{\mathcal{D}}\int_{\mathcal{V}} u^P(\mathbf{x},t^n,\mathbf{v},\mathbf{X})}{\int_{\mathcal{D}}\int_{\mathcal{V}} u^P(\mathbf{x},t^{n-1},\mathbf{v},\mathbf{X})}. \tag{64}$$

Then, asymptotically as $n\Delta t \to \infty$, the solution $u^n \approx u^{n-1} \approx u^\infty$ solves (62) $\forall \mathbf{X} \sim \mathrm{d}\mathcal{P}_{\mathbf{X}}$.

The stochastic power method designed in [9] and based on MC-gPC is an adaptation of the previously described algorithm. It can be summed up as, at each iteration/time step:

1. the resolution of (63) with MC-gPC (and in this paper, with the new multigroup-like solver) during time step $[t^{n-1}, t^n = t^{n-1} + \Delta t]$,
2. the projection of (64) onto the components of the gPC basis $(\phi_k)_{k\in\{0,\dots,P\}}$ as in [9]: for this, we define $\forall k \in \{0,...,P\}$

$$k_{\text{eff}}^{k,n} \quad = \int k_{\text{eff}}^{P,n-1}(\mathbf{X}) \times \frac{\int_{\mathcal{D}}\int_{\mathcal{V}} u^P(\mathbf{x},t^n,\mathbf{v},\mathbf{X})}{\int_{\mathcal{D}}\int_{\mathcal{V}} u^P(\mathbf{x},t^{n-1},\mathbf{v},\mathbf{X})}\phi_k(\mathbf{X})\,\mathrm{d}\mathcal{P}_{\mathbf{X}}, \forall k \in \{0,..,P\}, \tag{65}$$

   in which
   - $k_{\text{eff}}^{P,n-1}(\mathbf{X}) = \sum_{k=0}^P k_{\text{eff}}^{k,n-1}\phi_k(\mathbf{X})$ is an approximation of $k_{\text{eff}}^{n-1}(\mathbf{X})$ with $(k_{\text{eff}}^{k,n-1})_{k\in\{0,\dots,P\}}$ the gPC coefficients of the uncertain k_{eff} at the iteration $n-1$,
   - $u^P(\mathbf{x},t^{n-1},\mathbf{v},\mathbf{X})$ is the initial condition of a P-truncated MC-gPC resolution of (63),
   - and $u^P(\mathbf{x},t^n,\mathbf{v},\mathbf{X})$ is the $P-$truncated MC-gPC solution of (63) at time t^n.

Now, the main difference with the material described in [9] regarding the resolution of uncertain eigenproblems concerns the treatment of the fission term: $k_{\text{eff}}^{n-1}(\mathbf{X})$ appears in factor of $F^{\mathbf{X}}$ which means that Λ_s must be updated at each iterations. Note that it *does not imply a diagonalisation* at each iteration, only

– the reconstruction of Σ_s of general term

$$\Sigma_s^{i,j}(\mathbf{x}, \mathbf{v}\cdot\mathbf{v}') = \int \left[\sigma_s(\mathbf{x}, \mathbf{v}\cdot\mathbf{v}', \mathbf{X}) + \frac{\nu_f(\mathbf{x}, \mathbf{v}, \mathbf{X})}{\sum_{k=0}^{P} k_{\text{eff}}^{k,n-1}\phi_k(\mathbf{X})}\sigma_f(\mathbf{x}, \mathbf{v}\cdot\mathbf{v}', \mathbf{X})\right] \phi_i(\mathbf{X})\phi_j(\mathbf{X})\,\mathrm{d}\mathcal{P}_{\mathbf{X}}. \quad (66)$$

There are several ways to compute the above expression, see [72]. In practice, we use a Gauss quadrature rule with N points ($p_{1\text{D}}+1$ points in each direction in practice) such that $(X_i, w_i)_{i\in\{1,\dots,N\}} \sim (\mathbf{X}, \mathrm{d}\mathcal{P}_{\mathbf{X}})$ so that (66) is discretised as

$$\Sigma_s^{i,j}(\mathbf{x}, \mathbf{v}\cdot\mathbf{v}') \approx \sum_{l=1}^{N} w_l \left[\sigma_s(\mathbf{x}, \mathbf{v}\cdot\mathbf{v}', \mathbf{X}_l) + \frac{\nu_f(\mathbf{x}, \mathbf{v}, \mathbf{X}_l)}{\sum_{k=0}^{P} k_{\text{eff}}^{k,n-1}\phi_l(\mathbf{X}_l)}\sigma_f(\mathbf{x}, \mathbf{v}\cdot\mathbf{v}', \mathbf{X}_l)\right] \phi_i(\mathbf{X}_l)\phi_j(\mathbf{X}_l). \quad (67)$$

– Once Σ_s built, it only remains to build $\Lambda_s = R\Sigma_s R^{-1}$ (remember that R and R^{-1} have been computed once and for all at the beginning of the computation in each material). Note that this step could easily benefit from vectorial optimisations.

The general sketch for the resolution of uncertain eigenproblems with MC-gPC and the new multigroup-like MC-gPC solvers is recalled in Appendix D. We suggest finishing this paragraph with an application.

4.6.2. Application of the new multigroup-like MC-gPC solver to k_{eff} computations

Let us apply MC-gPC and its new variant to an analytical uncertain solution of (60). Let us assume the particles are monokinetic, i.e. $\sigma_\alpha(\mathbf{x}, \mathbf{v}, \mathbf{X}) = \sigma_\alpha(\mathbf{x}, \mathbf{X})$, $\forall\alpha \in \{s, t, f\}, \forall\mathbf{X} \sim \mathrm{d}\mathcal{P}_{\mathbf{X}}, \forall\mathbf{x} \in \mathcal{D}$, with $v = 1$, and that the scattering and fission reactions are deterministic, homogeneous and isotropic, i.e. $P_\alpha(\mathbf{x}, \mathbf{v}, \mathbf{X}))\,\mathrm{d}\mathbf{v} = \mathbf{1}_{\mathbb{S}^2}(\omega)\,\mathrm{d}\omega$ $\forall\mathbf{x} \in \mathcal{D}$, $\forall\mathbf{v} = v\omega = \omega \in \mathcal{V} = \mathbb{S}^2$, $\forall\alpha \in \{s, f\}$. Then equation (60) resumes to

$$\begin{aligned}\omega\cdot\nabla_{\mathbf{x}}u(\mathbf{x}, \omega, \mathbf{X}) + \sigma_t(\mathbf{x}, \mathbf{X})u(\mathbf{x}, \omega, \mathbf{X}) = \sigma_s(\mathbf{x}, \mathbf{X})\int u(\mathbf{x}, \omega', \mathbf{X})\,\mathrm{d}\omega',\\ + \frac{\nu_f(\mathbf{x}, \mathbf{X})\sigma_f(\mathbf{x}, \mathbf{X})}{k_{\text{eff}}(\mathbf{X})}\int u(\mathbf{x}, \omega', \mathbf{X})\,\mathrm{d}\omega'.\end{aligned} \quad (68)$$

Let us integrate the above equation with respect to $\mathbf{x} \in \mathcal{D}$ to get

$$\begin{aligned}\omega\cdot\int_{\partial\mathcal{D}} u(\mathbf{x}, \omega, \mathbf{X})\,\mathrm{d}\mathbf{x} + \int_{\mathcal{D}} \sigma_t(\mathbf{x}, \mathbf{X})u(\mathbf{x}, \omega, \mathbf{X})\,\mathrm{d}\mathbf{x} = \int_{\mathcal{D}} \sigma_s(\mathbf{x}, \mathbf{X})\int u(\mathbf{x}, \omega', \mathbf{X})\,\mathrm{d}\omega'\,\mathrm{d}\mathbf{x},\\ + \int_{\mathcal{D}} \frac{\nu_f(\mathbf{x}, \mathbf{X})\sigma_f(\mathbf{x}, \mathbf{X})}{k_{\text{eff}}(\mathbf{X})}\int u(\mathbf{x}, \omega', \mathbf{X})\,\mathrm{d}\omega'\,\mathrm{d}\mathbf{x}.\end{aligned} \quad (69)$$

Assume periodic boundary conditions (i.e. such that $\int_{\partial\mathcal{D}} u(\mathbf{x}, \mathbf{v}, \mathbf{X})\,\mathrm{d}\mathbf{x} = 0$) and homogeneous cross-sections (i.e. $\sigma_\alpha(\mathbf{x}, \mathbf{X}) = \sigma_\alpha(\mathbf{X})$), we obtain

$$\sigma_t(\mathbf{X}) = \sigma_s(\mathbf{X}) + \frac{\nu_f(\mathbf{X})\sigma_f(\mathbf{X})}{k_{\text{eff}}(\mathbf{X})}, \quad (70)$$

hence,

$$k_{\text{eff}}(\mathbf{X}) = \frac{\nu_f(\mathbf{X})\sigma_f(\mathbf{X})}{\sigma_t(\mathbf{X}) - \sigma_s(\mathbf{X})} = \frac{\nu_f(\mathbf{X})\sigma_f(\mathbf{X})}{\sigma_a(\mathbf{X}) + \sigma_f(\mathbf{X})}. \tag{71}$$

If the cross-sections and the multiplicity are deterministic, we recover the classical expression of a k_{eff} in an infinite medium. In the following, we intensively have resort to the conditions of test-case UD2O-1-0-IN[19] of [73], we are only going to assume that $\sigma_a = \overline{\sigma}_a + \hat{\sigma}_a \mathbf{X}$ is uncertain with $\overline{\sigma}_a = 0.027314$, $\hat{\sigma}_a = 0.01$ and $\mathbf{X} \sim \mathcal{U}([-1,1])$. Figure 8 (left) presents the results $(k^n_{\text{eff}})_{n\in\{1,\dots,40\}}$

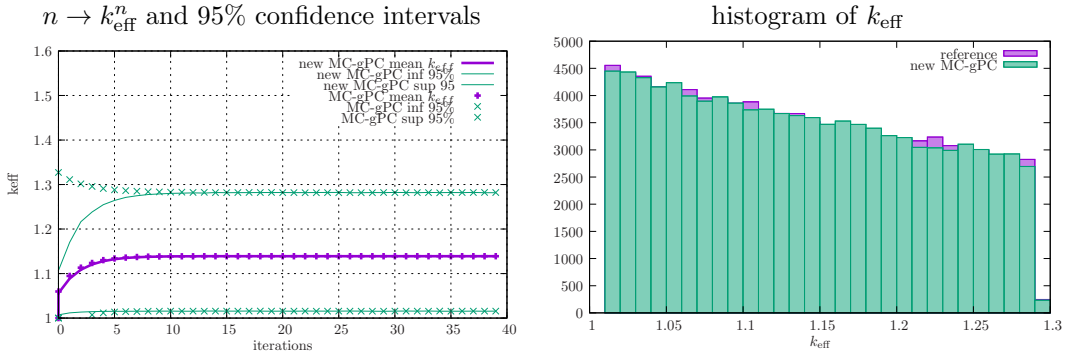

Figure 8: Left: evolution of the mean and 95% confidence interval of the k_{eff} through the iterations of the power method. Right: histograms of the k_{eff} obtained thanks to the reference solution (71) and the multigroup-like MC-gPC solver for $P = 7$ and $N_{MC} = 10^4$.

obtained with MC-gPC$_{P=7}$ and its new variant in the same conditions with respect to the number of (stochastic) power iterations n. Each time, care has been taken to make sure $n \times \Delta t$ is large enough to have $k^n_{\text{eff}} \approx k^\infty_{\text{eff}}$ as $n \times \Delta t = 200$. Figure 8 displays the reference solution obtained with MC-gPC (which already presented excellent agreements with the analytical solution, see [9]) in terms of mean solution and in terms of 2.5% and 97.5% quantiles[20]. The computation of the quantiles allows guarantying that the k_{eff} has a 95% probability of being between the $\times$ curves. Note that the quantile curves are not exactly equidistant from the mean: this is because the k_{eff} distribution is slightly skewed as testifies figure 8 (right). As a consequence, k_{eff} closer to 1 are more probable than higher ones. The two variants of MC-gPC are asymptotically (i.e. for $n \gg 1$) in perfect agreement on those statistical observables of the k_{eff} . The two variants only differ for the early iterations: they do not compute exactly the same transient regime. Figure 8 (right) displays the reference histogram obtained by sampling (10^6 samples) X through (71) and the new MC-gPC$_{P=7}$ one, obtained by sampling X through the gPC approximation $k^P_{\text{eff}}(X) = \sum_{k=0}^P k^k_{\text{eff}} \phi^X_k(X)$ for $P = 7$: the histogram have an excellent agreement, MC-gPC can efficiently approximate the histogram of $k_{\text{eff}}(\mathbf{X})$.

The previous studies were mainly qualitative. Let us consider more quantitative ones. Tables 1 and 2 present the run-times obtained with MC-gPC, ni-gPC and the new MC-gPC solver in the

[19] Infinite medium, $\overline{\nu}_f = 1.7$, $\overline{\sigma}_a = 0.027314$, $\overline{\sigma}_s = 0.464338$, $\overline{\sigma}_f = 0.054628$ where the $\overline{\ }$ notation stands for averaging.

[20] The quantiles have been evaluated with 10^6 samples of the MC-gPC$_{P=7}$ based approximations.

$N_{MC} = 5 \times 10^4$	$P = 1$	$P = 2$	$P = 3$	$P = 4$	$P = 5$
old MC-gPC 40 iter. $\Delta t = 5.0$	13.13s.	15.95s.	20.46s.	25.69s.	29.92s.
new MC-gPC 40 iter. $\Delta t = 5.0$	13.02s.	13.05s.	13.73s.	13.78s.	13.68s.
old MC-gPC 80 iter. $\Delta t = 2.5$	15.92s.	19.85s.	25.31s.	33.03s.	38.59s.
new MC-gPC 80 iter. $\Delta t = 2.5$	16.60s.	16.22s.	16.07s.	17.06s.	16.99s.
old MC-gPC 200 iter. $\Delta t = 1$	24.55s.	33.59s.	44.39s.	50.93s.	63.67s.
new MC-gPC 200 iter. $\Delta t = 1$	28.59s.	30.21s.	29.75s.	28.51s.	28.81s.

	Average time of ni-gPC runs
40 iter. $\Delta t = 5.0$	12.90s.
80 iter. $\Delta t = 2.5$	12.64s.
200 iter. $\Delta t = 1.0$	31.09s.

Table 1: run-times with respect to the polynomial order P and the number of power iterations n.

same conditions. Table 1 studies the dependence of the run-time with respect to P and to the number of iteration of the power iteration method. Table 2 studies the influence of the number of MC particles on the run-times of the three methods.

Let us begin by commenting on table 1: for this study, $N_{MC} = 5 \times 10^4$ and $n \times \Delta t = 200$ both remain constant. But the number of iterations n and the time step Δt do change, so that their product remains 200: in such conditions, the results, for this problem, are converged in terms of power iteration (just as in figure 8). First, as can be seen for the results of ni-gPC, even for the non-intrusive application, the run-time can be affected by the number of iterations: with $n \times \Delta t = 200 \times 1.0$, the average run-time of ni-gPC ($31.09s.$) is greater than for couples with smaller number of iterations n. The same increase can be observed for the two MC-gPC variants, see for example the first column ($P = 1$) of table 1: for $P = 1$, the run-times with respect to $n \times \Delta t$ of the MC-gPC solvers are even comparable to the ones of ni-gPC. On every line of table 1 for (the old version of) MC-gPC, the cost increases with respect to P. This is because with the MC scheme described in [1], each MC particle must contribute to every gPC coefficients (see the $+=$ operation in algorithm 6). On another hand, the cost on each line of table 1 for the new MC-gPC solver remains quite the same, even as P increases. This is because with the new MC-gPC solver, the tallying phase is much less costly as each MC particle only contribute to the gPC coefficient of its final group. Now, independently of P, as n grows, the cost of the MC-gPC solvers grow. For the old version of MC-gPC, the cost grows with both P and n as the tallying phase, which is the costly one, is more frequent with n. On another hand, with the new MC-gPC solver, the cost only grows with n, just as for ni-gPC. Furthermore, the cost with respect to n of the new MC-gPC is almost the same as the cost of one ni-gPC run.

40 iter. $\Delta t = 5.0$, $P = 5$	$N_{MC} = 10^3$	$N_{MC} = 10^4$	$N_{MC} = 10^5$	$N_{MC} = 10^6$
old MC-gPC	0.77s.	6.41s.	60.4s.	598.8s.
new MC-gPC	0.55s.	3.20s.	29.0s.	284.4s.
Average time of ni-gPC runs	0.25s.	2.9s.	20.3s.	211.0s.

Table 2: run-times with respect to N_{MC}.

Table 2 studies the influence of N_{MC} on the run-times of the three solvers. For each solver (i.e. each line of table 2), as the number of MC particles is multiplied by 10, the cost is multiplied by

10. This is typical of MC schemes. Now, the cost of the treatment of one MC particle is more important for the old version of MC-gPC than for the new one. The treatment of one MC particle with the new MC-gPC solver remains more important than for ni-gPC. Still, the gain with MC-gPC with respect to ni-gPC, in this configuration is about $\times\frac{211.0\times N}{598.8} = \frac{211.0\times 6}{598.8} = 2.11$ whereas the gain of its new variant with respect to ni-gPC is about $\times\frac{211.0\times N}{284.4} = \frac{211.0\times 6}{284.4} = 4.45$ with an even better numerical noise.

We consider this test-case demonstrates that the new version of MC-gPC can also handle (more) efficiently k_{eff} computations and also, indirectly, nonlinear problems for which the cross-sections may change with respect to time.

4.7. Hybrid intrusive/non-intrusive uncertainty propagation

In this section, we would like to take few lines to discuss about what intrusive uncertainty propagation codes (independently of the physics of interest) can bring:

- from the previous sections, we saw situations in which intrusiveness is worth it (from $\times 2$ to $\times 40$ computational gains).
- Still, intrusiveness can be more or less costly in terms of development and implementation: even if the modifications are simple, the verification always takes time.

Based on these two observations, we would like to show that hybrid non-intrusive/intrusive applications are at hand as soon as an intrusive code is available. Furthermore, these hybrid computations are competitive with respect to a full classical non-intrusive application.

Let us develop the idea: suppose that one can take into account intrusively the main sources of uncertainties. Take the example of section 4.5 and assume that the developments are ready in order to take into account the uncertainties on $\sigma_t(X_1), \sigma_s(X_2)$ but not yet on $\eta(X_3)$. Then we can quite easily run the intrusive code (here the MC-gPC solvers) in order to propagate the uncertainties with respect to X_1, X_2 several times for several values of $(X_3^i, w_i)_{i\in\{1,\dots,N\}} \sim (X_3,\, \mathrm{d}\mathcal{P}_{X_3})$ non-intrusively (here with ni-gPC). We run the MC-gPC solvers propagating the uncertainties with respect to (X_1, X_2) at several points $(X_3^i, w_i)_{i\in\{1,\dots,N\}}$ and gather the solutions of MC-gPC $(U(\mathbf{x}, t, \mathbf{v}, X_3^i), w_i)_{i\in\{1,\dots,N\}}$. We can then compute the remaining gPC coefficients for the remaining stochastic dimension X_3 non-intrusively by numerical integration on the gPC coefficients obtained intrusively *via* MC-gPC : the coefficients

$$U_k(\mathbf{x}, t, \mathbf{v}) = \int U(\mathbf{x}, t, \mathbf{v}, X_3)\phi_k^{X_3}(X_3)\,\mathrm{d}\mathcal{P}_{X_3} \approx \sum_{i=1}^{N} U(\mathbf{x}, t, \mathbf{v}, X_3^i) w_i \phi_k^{X_3}(X_3^i), \forall k \in \{0, ..., p_{1\mathrm{D}}^{X_3}\},$$

corresponds to the gPC coefficients obtained from a *classical* tensorisation of the gPC basis relative to X_1, X_2, X_3.

Figure 9 compares the results in terms of spatial profiles of the mean and variance obtained intrusively with MC-gPC and with the hybrid ni-gPC /MC-gPC strategy described above. The dimension $Q = 2$ relative to X_1, X_2 are treated intrusively thanks to MC-gPC with $P = (p_{1\mathrm{D}}+1)^Q = (2+1)^2 = 9$ and the dimension X_3 is treated non-intrusively (ni-gPC) thanks to 4 Gauss-Legendre points in direction X_3 and $p_{1\mathrm{D}}^{X_3} = 2$ resulting in a total of $(p_{1\mathrm{D}}+1)^2 \times (p_{1\mathrm{D}}^{X_3}+1) = 27$ hybrid gPC coefficients. On figure 9, the results are in agreement. Now, the costs of each numerical strategies are given by

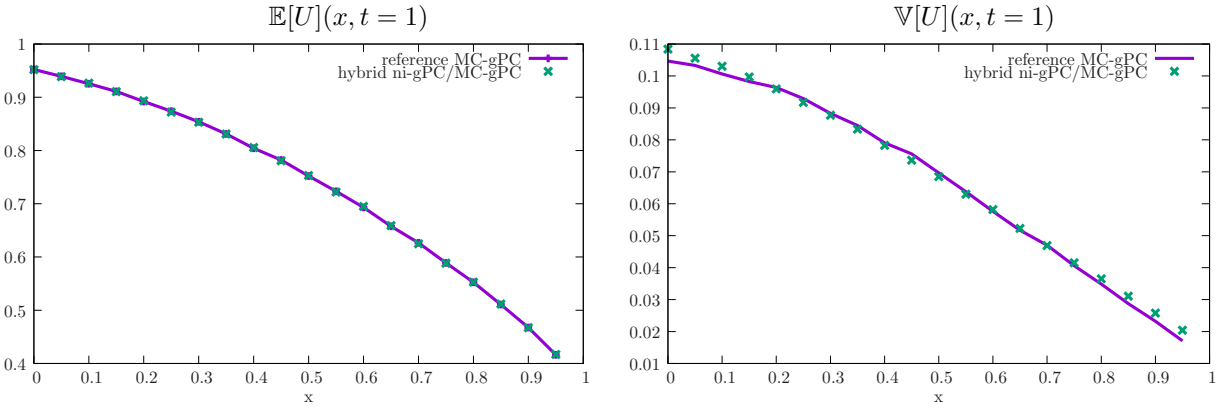

Figure 9: Comparison between the results from new multigroup-like MC-gPC in 3D stochastic dimension and some hybrid ni-gPC /MC-gPC results (for $Q = 2$ for MC-gPC and the last stochastic dimension taken into account by ni-gPC). Left: mean of $U(x, t = 1, X)$ with respect to x. Right: variance of $U(x, t = 1, X)$ with respect to x.

– new MC-gPC : cost = 1 × CPU time of 1 run $= 1 \times 1\,\text{min}\,25s$.
– ni-gPC : cost = 64 × CPU time of 1 run $= 64 \times 0\,\text{min}\,54s = 58\,\text{min}\,06s$.
– hybrid : cost = 4 × CPU time of 1 run $= 4 \times 0\,\text{min}\,58s = 3\,\text{min}\,52s$.

The above three first lines are the same as in section 4.5. The hybrid line corresponds to an ni-gPC /new MC-gPC hybridization (so we do not report the line concerning the old MC-gPC solver here). As can be seen, the hybrid computations cost more than the full intrusive MC-gPC strategy but still much less than the ni-gPC one. The gain is $\times 15$ instead of $\times 40$. Furthermore, with $Q = 2$, the cost of MC-gPC is closer to the cost of a non-intrusive application ($\approx 58s$) as less coefficients need to be computed. Furthermore, note that if we take $p_{1\text{D}} = 0$ in every stochastic direction, $P = 1$ and only one gPC coefficient (u_0) is computed: it corresponds to a deterministic run whose cost *is the same as a classical MC computation without uncertainties*. Concerning the ni-gPC computations: they have been made in 1D, only with respect to X_3. In this sense, the intrusive resolution of the two first stochastic dimensions (X_1, X_2) mitigates the curse of dimensionality on this problem (see point $*_1$ of section 1).

With the previous example, we wanted the reader to have in mind that developing an intrusive uncertainty propagation code *can be long but can be made progressive*: uncertain parameter per uncertain parameter, beginning by the ones which need to be systematically propagated (in order to identify them, it is common performing a sensitivity analysis for example, see [74]). This can be progressive and the code remains usable in a deterministic way ($p_{1\text{D}} = 0$). Finally, when a more exotic situation occurs, with respect to a new uncertain parameter, we can rely on the hybrid uncertainty propagation for punctual studies.

5. Conclusion

In this paper, we deepened the study and analysis of MC-gPC for the construction and the resolution of efficient gPC based reduced model of the uncertain linear Boltzmann equation. In particular, we focused on understanding, identifying and correcting the terms responsible for MC-gPC to have a higher numerical noise/error. The correction we suggest is based on rewriting astutely the gPC based reduced model as a multigroup-like system of equations and solve it with

classical multigroup MC schemes (semi-analog and non-analog mainly in this paper). The new MC-gPC solvers not only have better performances in terms of numerical noise but also presents improved run-times, considerably increasing the overall gain of MC-gPC. The aforementioned gains have been obtained on several benchmarks of the literature, for the resolution of the uncertain linear Boltzmann equation and in an uncertain eigenproblem context (computation of effective multiplication factor k_{eff} in neutronics). The paper also provides a discussion on implementing progressively intrusive uncertainty propagation codes and benefiting from their efficiency thanks to an intrusive/non-intrusive hybridization.

6. Acknowledgements

The author would like to thank Marc Sancandi for valuable discussions.

References

[1] G. Poëtte, A gPC-intrusive Monte Carlo scheme for the resolution of the uncertain linear Boltzmann equation, Journal of Computational Physics 385 (2019) 135 – 162. doi:https://doi.org/10.1016/j.jcp.2019.01.052.
URL http://www.sciencedirect.com/science/article/pii/S002199911930110X

[2] G. Poëtte, Spectral convergence of the generalized polynomial chaos reduced model obtained from the uncertain linear boltzmann equation, Mathematics and Computers in Simulation 177 (2020) 24–45. doi:https://doi.org/10.1016/j.matcom.2020.04.009.
URL https://www.sciencedirect.com/science/article/pii/S037847542030121X

[3] J. A. Carrillo, L. Pareschi, M. Zanella, Particle based gpc methods for mean-field models of swarming with uncertainty, Comm. Comp. Phys, 25 (2018) 508–531.

[4] J. A. Carrillo, M. Zanella, Monte Carlo gPC methods for diffusive kinetic flocking models with uncertainties Preprint (2019).

[5] L. Pareschi, An introduction to uncertainty quantification for kinetic equations and related problems (2020). arXiv:2004.05072.

[6] L. Pareschi, M. Zanella, Monte Carlo stochastic Galerkin methods for the Boltzmann equation with uncertainties: space-homogeneous case (03 2020). doi:10.13140/RG.2.2.28177.17760.

[7] L. Pareschi, M. Zanella, Monte carlo stochastic galerkin methods for the boltzmann equation with uncertainties: Space-homogeneous case, Journal of Computational Physics 423 (2020) 109822. doi:https://doi.org/10.1016/j.jcp.2020.109822.
URL https://www.sciencedirect.com/science/article/pii/S0021999120305969

[8] G. Poëtte, Efficient uncertainty propagation for photonics: Combining implicit semi-analog monte carlo (ismc) and monte carlo generalised polynomial chaos (mc-gpc), Journal of Computational Physics (2021) 110807doi:https://doi.org/10.1016/j.jcp.2021.110807.
URL https://www.sciencedirect.com/science/article/pii/S0021999121007026

[9] G. Poëtte, E. Brun, Efficient uncertain k eff computations with the Monte Carlo resolution of generalised Polynomial Chaos Based reduced models, working paper or preprint (Nov. 2020).
URL https://hal.archives-ouvertes.fr/hal-02996843

[10] G. Poëtte, Numerical analysis of the Monte-Carlo noise for the resolution of the deterministic and uncertain linear Boltzmann equation (comparison of non-intrusive gPC and MC-gPC), working paper or preprint (Nov. 2021).
URL https://hal.archives-ouvertes.fr/hal-03447798

[11] R. A. Todor, C. Schwab, Karhunen-Loève approximation of random fields by generalized fast multipole methods, J. Comp. Phys. 217 (1) (2006) 100–122.

[12] M. Meyer, H. Matthies, Efficient model reduction in non-linear dynamics using the Karhunen-Loève expansion and dual-weighted-residual methods, Comp. Meth. Appl. Mech. Eng. Informatikbericht 2003-08, TU Braunschweig, Germany (2004).

[13] J. Mercer, Functions of Positive and Negative Type and their Connection with the Theory of Integral Equations, Philos. Trans. Roy. Soc. 209 (1909).

[14] R. Lebrun, A. Dutfoy, A Generalization of the Nataf Transformation to Distributions with Elliptical Copula, Prob. Eng. Mech. 24,2 (2009) 172–178.

[15] R. Lebrun, A. Dutfoy, An Innovating Analysis of the Nataf Transformation from the Copula viewpoint, Prob. Eng. Mech. 24,3 (2009) 312–320.

[16] J. Spanier, E. M. Gelbard, Monte Carlo Principles and Neutron Transport Problems, Addison-Wesley, 1969.

[17] E. E. Lewis and W. F. Miller Jr., Computational Methods of Neutron Transport, John Wiley and Son New York, 1984.

[18] G. Bell, S. Gladstone, Nuclear Reactor Theory, Van Nostrand Reinhold Company, 1970.
URL https://books.google.fr/books?id=RNQmAQAAMAAJ

[19] F. Golse, G. Allaire, Transport et Diffusion, 2015, polycopié de cours.

[20] B. Iooss, P. Lemaître, A Review on Global Sensitivity Analysis Methods, Dellino, Gabriella and Meloni, Carlo, Springer US, Boston, MA, 2015, pp. 101–122. doi:10.1007/978-1-4899-7547-8_5.
URL http://dx.doi.org/10.1007/978-1-4899-7547-8_5

[21] E. Brun, S. Chauveau, F. Malvagi, Patmos: A prototype Monte Carlo transport code to test high performance architectures, In Proceedings of International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering, Jeju, Korea, 2017.

[22] E. Brun, F. Damian, C. Diop, E. Dumonteil, F. Hugot, C. Jouanne, Y. Lee, F. Malvagi, A. Mazzolo, O. Petit, J. Trama, T. Visonneau, A. Zoia, Tripoli-4®, cea, edf and areva reference monte carlo code, Annals of Nuclear Energy 82 (2015) 151 – 160, joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo 2013, SNA + MC 2013. Pluri- and Trans-disciplinarity, Towards New Modeling and Numerical Simulation Paradigms. doi:https://doi.org/10.1016/j.anucene.2014.07.053.
URL http://www.sciencedirect.com/science/article/pii/S0306454914003843

[23] T. Goorley, MCNP6.1.1-Beta Release Notes LA-UR-14-24680 (2014).

[24] J. A. Fleck, J. D. Cummings, An implicit monte-carlo scheme for calculating time and frequency dependent nonlinear radiation transport, Journal of Computational Physics (1971).

[25] R. P. Smedley-Stevenson, R. G. McClarren, Asymptotic diffusion limit of cell temperature discretisation schemes for thermal radiation transport, Journal of Computational Physics 286 (2015) 214 – 235. doi:https://doi.org/10.1016/j.jcp.2013.10.038.
URL http://www.sciencedirect.com/science/article/pii/S0021999113007146

[26] J.-F. Clouet, G. Samba, Asymptotic diffusion limit of the symbolic monte carlo method for the transport equation, Journal of Computational Physics 195 (1) (2004) 293 – 319. doi:https://doi.org/10.1016/j.jcp.2003.10.008.
URL http://www.sciencedirect.com/science/article/pii/S0021999103005333

[27] G. Poëtte, X. Valentin, A new implicit monte-carlo scheme for photonics (without teleportation error and without tilts), Journal of Computational Physics 412 (2020) 109405. doi:https://doi.org/10.1016/j.jcp.2020.109405.
URL http://www.sciencedirect.com/science/article/pii/S0021999120301790

[28] E. Steinberg, S. I. Heizler, Multi-frequency implicit semi-analog monte-carlo (ismc) radiative transfer solver in two-dimensions (without teleportation) (2021). arXiv:2108.02612.

[29] E. Steinberg, S. I. Heizler, Discrete implicit monte-carlo (dimc) scheme for simulating radiative transfer problems (2021). arXiv:2108.13453.

[30] B. Perthame, Transport Equations in Biology, Birkhauser Verlag, Basel Boston Berlin, 2000.

[31] L. Pareschi, P. Vellucci, M. Zanella, Kinetic models of collective decision-making in the presence of equality bias, Physica A: Statistical Mechanics and its Applications 467 (2017) 201 – 217. doi:https://doi.org/10.1016/j.physa.2016.10.003.
URL http://www.sciencedirect.com/science/article/pii/S0378437116306902

[32] B. M. Althouse, E. A. Wenger, J. C. Miller, S. V. Scarpino, A. Allard, L. Hébert-Dufresne, H. Hu, Stochasticity and heterogeneity in the transmission dynamics of sars-cov-2, 2020.

[33] B. Lapeyre, E. Pardoux, R. Sentis, Méthodes de Monte Carlo pour les équations de transport et de diffusion, no. 29 in Mathématiques & Applications, Springer-Verlag, 1998.

[34] M. Coste-Delclaux, C. DIOP, A. NICOLAS, B. BONIN, Neutronique, E-den, Une monographie de la Direction de l'énergie nucléaire, CEA Saclay; Groupe Moniteur, 2013.
URL https://hal-cea.archives-ouvertes.fr/cea-01152822

[35] D. Dureau, G. Poëtte, Hybrid Parallel Programming Models for AMR Neutron Monte Carlo Transport, in: Joint International Conference on Supercomputing in Nuclear Applications + Monte Carlo, no. 04202 in Parallelism and HPC, Monte Carlo, 2013.

[36] Martin, William R., et al., Monte Carlo photon transport on shared memory and distributed memory parallel processors, International Journal of High Performance Computing Applications 1.3 (1987) 57–74.

[37] Parallel performance study of monte carlo photon transport code on shared-, distributed-, and distributed-shared-memory architectures, Parallel and Distributed Processing Symposium, 2000.

[38] G. Blatman, Adaptive sparse polynomial chaos expansions for uncertainty propagation and sensitivity analysis, Thèse de doctorat, Université Blaise Pascal - Clermont II (2009).

[39] N. Wiener, The Homogeneous Chaos, Amer. J. Math. 60 (1938) 897–936.

[40] R. Cameron, W. Martin, The Orthogonal Development of Non-Linear Functionals in Series of Fourier-Hermite Functionals, Annals of Math. 48 (1947) 385–392.

[41] Ernst, Oliver G., Mugler, Antje, Starkloff, Hans-Jörg, Ullmann, Elisabeth, On the convergence of generalized polynomial chaos expansions, ESAIM: M2AN 46 (2) (2012) 317–339. doi:10.1051/m2an/2011045.
URL https://doi.org/10.1051/m2an/2011045

[42] F. Simon, P. Guillen, P. Sagaut, D. Lucor, A gPC based approach to uncertain transonic aerodynamics, CMAME 199 (2010) 1091–1099.

[43] D. Lucor, J. Meyers, P. Sagaut, Sensitivity Analysis of LES to Subgrid-Scale-Model Parametric Uncertainty using Polynomial Chaos, J. Fluid Mech. 585 (2007) 255–279.

[44] J.-M. Martinez, J. Cahen, A. Millard, D. Lucor, F. Huvelin, J. Ko, N. Poussineau, Modélisation des Incertitudes par Polynômes de Chaos – Étude d'un Écoulement en Milieux Poreux, Tech. Rep. Rapport DM2S/DIR/RT/06-006/A, CEA-CEMRACS (2006).

[45] G. Blatman, B. Sudret, Sparse Polynomial Chaos Expansions and Adaptive Stochastic Finite Elements using a Regression Approach, C. R. Méc. 336 (2008) 518–523. doi:10.1016/j.crme.2008.02.013.

[46] B. Sudret, Uncertainty Propagation and Sensitivity Analysis in Mechanical Models, Contribution to Structural Reliability and Stochastic Spectral Methods, Habilitation à Diriger des Recherches, Université Blaise Pascal - Clermont II (2007).

[47] M. Berveiller, B. Sudret, M. Lemaire, Stochastic Finite Element: a Non Intrusive Approach by Regression, Rev. Eur. Méc. Num. 15 (1-2-3) (2006) 81–92.

[48] G. Dimarco, L. Liu, L. Pareschi, X. Zhu, Multi-fidelity methods for uncertainty propagation in kinetic equations (2021). arXiv:2112.00932.

[49] C. Ahrens, E. Larsen, A semi-analog monte carlo method for grey radiative transfer problems, in: Proceedings of the ANS Topical Meeting: International Conference on Mathematical Methods to Nuclear Applications, 2001.

[50] J. A. F. Jr., The calculation of nonlinear radiation transport by a monte carlo method, Tech. rep., Lawrence Radiation Laboratory, University of California (1961).

[51] M. S. McKinley, E. D. B. III, A. Szoke, Comparison of implicit and symbolic implicit monte carlo line transport with frequency weight vector extension, Journal of Computational Physics (2003).

[52] A. G. Irvine, I. D. Boyd, N. A. Gentile, Reducing the spatial discretization error of thermal emission in implicit monte carlo simulations, Journal of Computational and Theoretical Transport 45 (1-2) (2016) 99–122. arXiv:https://doi.org/10.1080/23324309.2015.1060245, doi:10.1080/23324309.2015.1060245.
URL https://doi.org/10.1080/23324309.2015.1060245

[53] G. Poëtte, X. Valentin, A. Bernede, Canceling teleportation error in legacy imc code for photonics (without tilts, with simple minimal modifications), Journal of Computational and Theoretical Transport 49 (4) (2020) 162–194. arXiv:https://doi.org/10.1080/23324309.2020.1785893, doi:10.1080/23324309.2020.1785893.
URL https://doi.org/10.1080/23324309.2020.1785893

[54] A. Bernede and G. Poëtte, An Unsplit Monte-Carlo solver for the resolution of the linear Boltzmann equation coupled to (stiff) Bateman equations, Journal of Computational Physics 354 (2018) 211 – 241. doi:https://doi.org/10.1016/j.jcp.2017.10.027.
URL http://www.sciencedirect.com/science/article/pii/S0021999117307805

[55] M. Coste-Delclaux, C. Jouanne, F. Moreau, C. Mounier, Galilée-1: a validation and processing system for endf-6 and gnd evaluations, in: EPJ Web of Conferences, Vol. 111, EDP Sciences, 2016, p. 06005.

[56] R. Macfarlane, D. W. Muir, R. M. Boicourt, A. C. Kahler, III, J. L. Conlin, The njoy nuclear data processing system, version 2016 (1 2017). doi:10.2172/1338791.
URL https://www.osti.gov/biblio/1338791

[57] CALENDF-2010 : User Manual R-6277 (2011).

[58] N. M. Larson, Updated user's guide for sammy: Multilevel r-matrix fits to neutron data using bayes' equations (10 2008). doi:10.2172/941054.
URL https://www.osti.gov/biblio/941054

[59] A. Trkov, D. A. Brown, Endf-6 formats manual: Data formats and procedures for the evaluated nuclear data files (1 2018). doi:10.2172/1425114.
URL https://www.osti.gov/biblio/1425114

[60] G. Poëtte, Contribution to the mathematical and numerical analysis of uncertain systems of conservation laws and of the linear and nonlinear boltzmann equation, Habilitation à diriger des recherches, Université de Bordeaux 1 (Sep. 2019).
URL https://hal.archives-ouvertes.fr/tel-02288678

[61] G. C. Papanicolaou, Asymptotic Analysis of Transport Processes, Bulletin of the American Mathematical Society 81 (2) (1975).

[62] G. Saporta, Probabilités, Analyse de Données et Statistique, 2e édition, Technip, 2006.

[63] A. S. o. M. E. ASME V&V 20-2009, Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer, ASME (2009).

[64] F. Palais des Papes, Avignon (Ed.), Load balancing of parallel Monte Carlo transport calculations, 2005.

[65] B. T. A., P. S. Brantley, An efficient, robust, domain-decomposition algorithm for particle Monte Carlo, Journal of Computational Physics 228.10 (2009) 3882–3890.

[66] W. R. Martin, T.-C. Wan, T. S. Abdel-Rahman, T. N. Mudge, K. Miura, Monte carlo photon transport on shared memory and distributed memory parallel processors, The International Journal of Supercomputing Applications 1 (3) (1987) 57–74. arXiv:https://doi.org/10.1177/109434208700100306, doi:10.1177/109434208700100306.
URL https://doi.org/10.1177/109434208700100306

[67] D. P.L., S. E.J., L. T.F., Y. Y., J. K.H., Complexity of the Basic Reproduction Number (R0), Emerg Infect Dis. 25(1) (2019) 1–4, https://doi.org/10.3201/eid2501.171901.

[68] J. Lieberoth, Monte carlo technique to solve. the static eigenvalue problem of the boltzmann transport equation., Nukleonik, 11: 213-19(Sept. 1968). (1 1968).
URL https://www.osti.gov/biblio/4835730

[69] F. Sheben, Iterative Methods for Criticality Computations in Neutron Transport Theory (2011).

[70] S. Maire, D. Talay, On a Monte Carlo method for neutron transport criticality computations, IMA Journal of Numerical Analysis 26 (4) (2006) 657–685.
URL https://hal-amu.archives-ouvertes.fr/hal-01479840

[71] R. N. Blomquist, E. M. Gelbard, Alternative Implementations of the Monte Carlo Power Method, Nuclear Science and Engineering 141 (2) (2002) 85–100. arXiv:https://doi.org/10.13182/NSE01-30, doi:10.13182/NSE01-30.
URL https://doi.org/10.13182/NSE01-30

[72] B. J. Debusshere, H. N. Najm, P. P. Pébay, O. M. Knio, R. G. Ghanem, O. P. L. Maître, Numerical Challenges in the Use of Polynomial Chaos Representations for Stochastic Processes, J. Sci. Comp. 26 (2004) 698–719.

[73] Sood, Avneet, Forster, Arthur, Parsons, Kent, Analytical benchmark test set for criticality code verification, Progress in Nuclear Energy 42 (1) (2003) 55–106.

[74] S. da Veiga, F. Gamboa, B. Iooss, C. Prieur, S. for Industrial, A. Mathematics, Basics and Trends in Sensitivity Analysis: Theory and Practice in R, Society for Industrial and Applied Mathematics, 2021.
URL https://books.google.fr/books?id=Tqt9zgEACAAJ

Appendix A. A simple analytical uncertain solution

In this section we build an analytical solution in a simple uncertain configuration. It is used as a reference solution for the convergence studies of section 2 and figure 1-left. The configuration is monokinetic (i.e. $v = 1$) and homogeneous (i.e. $u(x, t, \mathbf{v}, X) = u(t, \omega, X)$). We assume the uncertainty, one-dimensional here for the sake of simplicity, affects the scattering cross-sections $\sigma_s = \overline{\sigma}_s + \hat{\sigma}_s X$, where $X \sim \mathcal{U}[-1, 1]$ and $\hat{\sigma}_s$ is closely related to the variance of the uncertain scattering cross-section. Let us introduce $U(t, X) = \iint u(x, t, \omega, X)\,\mathrm{d}x\,\mathrm{d}\omega$. In the previously described

configuration, the uncertain linear Boltzmann equation resumes to the following stochastic ordinary differential equation

$$\begin{cases} \partial_t U(t,X) + v\sigma_t U(t,X) = v\sigma_s(X)U(t,X), \\ U(0) = U_0, \end{cases} \tag{A.1}$$

satisfied by U. Introduce $\sigma_a = \sigma_t - \sigma_s$, then the solution is given by

$$U(t,X) = U_0 e^{-v\sigma_a(X)t} = U_0 e^{-v(\sigma_t - \overline{\sigma}_s - \hat{\sigma}_s X)t} = U_0 e^{-v(\overline{\sigma}_a - \hat{\sigma}_s X)t}. \tag{A.2}$$

The quantity $U(t,X)$ is a random variable indexed by time t, i.e. it is a stochastic process. In this case, mean and variance of the stochastic process (A.2) can be computed analytically and are given by

$$\begin{aligned} M_1^U(t) = \mathbb{E}[U(t,X)] &= \tfrac{1}{2} U_0 e^{-v\overline{\sigma}_a t} \frac{e^{v\hat{\sigma}_s t} - e^{-v\hat{\sigma}_s t}}{\hat{\sigma}_s t v}, \\ M_2^U(t) = \mathbb{E}[U^2(t,X)] &= \tfrac{1}{4} U_0^2 e^{-2v\overline{\sigma}_a t} \frac{e^{2v\hat{\sigma}_s t} - e^{-2v\hat{\sigma}_s t}}{\hat{\sigma}_s t v}, \\ \mathbb{V}[U](t) &= M_2^U(t) - (M_1^U(t))^2. \end{aligned} \tag{A.3}$$

Of course, higher order moments, probability of failure, complete characterisation of the probability density function of the stochastic process can be calculated but in figure 1, we focus on the variance $\mathbb{V}[U](t)$ to perform the convergence studies. Note that in practice, we take $v = 1, U_0 = 1, \sigma_t = \overline{\sigma}_s = 0.1, \hat{\sigma}_s = 0.1$. The L^1-norm of the error is computed at time $t = 10$. The curves of figure 1 implying an MC scheme are averaged over 128 computations with different seeds.

Appendix B. A simple test-case in the collisional regime

From [10], the variance of the error made $\forall X \in \mathrm{d}\mathcal{P}_X$ in a non-intrusive resolution is

$$\sigma_{\mathrm{sa}}^2(t,X) = U_0^2(X) \left(e^{\dfrac{(v\sigma_s(X))^2 - (v\sigma_t(X))^2}{(v\sigma_t(X))} t} - e^{2(v\sigma_s(X) - v\sigma_t(X))t} \right). \tag{B.1}$$

As a consequence, asymptotically with N_{MC}, according to the Central Limit Theorem, the error made on $U(t,X)$, $\forall X \in \mathrm{d}\mathcal{P}_{\mathbf{X}}$ is given by

$$U(t,X) - U^{N_{MC}}(t,X) \sim \mathcal{G}\left(0, \frac{\sigma_{\mathrm{sa}}(t,X)}{\sqrt{N_{MC}}}\right). \tag{B.2}$$

Figure 1 presents the results obtained on the uncertain configuration given by

$$\begin{cases} U_0 = 1.0, \\ X \sim \mathcal{U}([-1,1]), \\ \sigma_s(X) = \sigma_a + \sigma_s + \nu_f \sigma_f(X), \\ \sigma_a = 0.6, \sigma_s = 0.1, \nu_f \sigma_f(X) = 0.1 + 0.21X, \\ \sigma_t = \sigma_a + \sigma_s + \sigma_f = 1.0. \end{cases} \tag{B.3}$$

Note that as X is uniformly distributed in $[-1,1]$, the Legendre basis is used for $(\phi_k(X))_{k\in\{0,\dots,P\}}$ (orthonormal with respect to the scalar product defined by the probability measure of X).

Appendix C. The algorithms for the original MC-gPC schemes

In this section, we recall the sketch of the algorithm for the MC-gPC resolution described in [1, 8, 9, 10]. It is detailed, for a backward resolution, just as algorithm 1 for the new solver, in algorithm 6. This, to our opinion, eases the comparison of the two solvers. Note that paper [1] describes some algorithmic comparisons of MC-gPC and the native MC code from which it is based. Those comparisons are not recalled here.

Function *backward_MC-gPC_tracking()*

for $k \in \{0, ..., P\}$ **do**
set $u_k(\mathbf{x}, t, \mathbf{v}) = 0$
set $u_k^2(\mathbf{x}, t, \mathbf{v}) = 0$
end
for $p \in \{1, ..., N_{MC}\}$ **do**
set $s_p = t$ *#this will be the life time of particle p*
set $\mathbf{x}_p = \mathbf{x}$
set $\mathbf{v}_p = \mathbf{v}$
set $\mathbf{X}_p \sim \mathbf{X}$
set i_p=find_cell($\mathbf{x}_p$)
set $w_p(t) = \dfrac{1}{N_{MC}}$
while $s_p > 0$ **do**
if $\mathbf{x}_p \notin \mathcal{D}(\mathbf{X}_p)$ **then**
#here a general function for the application of arbitrary boundary conditions
apply_boundary_conditions($\mathbf{x}_p, s_p, \mathbf{v}_p, \mathbf{X}_p$)
end
$\tau_{census} = s_p$
τ_{exit} = compute_time_until_cell_exit($\mathbf{x}_p, \mathbf{v}_p, g_p$)
τ_{inter} = sample_interaction_time($i_p, \mathbf{v}_p, \mathbf{X}_p$)
$\tau = \min(\tau_{inter}, \tau_{exit}, \tau_{census})$
#set the life time of particle p to:
$s_p \longleftarrow s_p - \tau$
#move the particle p
$\mathbf{x}_p \longleftarrow \mathbf{x}_p + \mathbf{v}_p\tau$,
K = compute_weight_modif($\mathbf{x}_p$, $\mathbf{v}_p$, τ, τ_{census}, τ_{exit}, $\tau_{inter}, i_p, \mathbf{X}_p$)
$w_p \longleftarrow K \times w_p$
if $\tau == \tau_{census}$ **then**
#tally the contribution of particle p for the first and second moments
for $k \in \{0, ..., P\}$ **do**
$u_k(\mathbf{x}, t, \mathbf{v}) += w_p \times u_0(\mathbf{x}_p, \mathbf{v}_p, \mathbf{X}_p)\phi_k(\mathbf{X}_p)$
$u_k^2(\mathbf{x}, t, \mathbf{v}) += w_p \times [u_0(\mathbf{x}_p, \mathbf{v}_p, \mathbf{X}_p)\phi_k(\mathbf{X}_p)]^2$
end
end
if $\tau == \tau_{exit}$ **then**
 # find the new cell in which particle p will evolve
$i_p^{\text{old}} = i_p$
$i_p \longleftarrow$ find_adjacent_cell(i_p^{old}),
end
if $\tau == \tau_{inter}$ **then**
#Sample the velocity $\mathbf{V}'$ *of particle p from* $P^s_{\mathbf{V}'}(\mathbf{x}_p, \mathbf{v}_p \cdot \mathbf{v}', \mathbf{X}_p)\,\mathrm{d}\mathbf{v}'$
$\mathbf{v}'$ =sample_velocity($\mathbf{x}_p$, $\mathbf{v}_p$, $\mathbf{X}_p$)
$\mathbf{v}_p = V'$
end
end
end
end

Algorithm 6: The semi-analog and non-analog MC-gPC schemes described in term of algorithmic operations in order to compute (adjoint) $u(\mathbf{x}, t, \mathbf{v}, \mathbf{X})$.

The tracking phase allowing to solve (17) with (the old version of) MC-gPC is described in algorithm 6. It describes the 'tracking' of an uncertain population of particles within the simulation domain $\mathcal{D}$. In order to present both implementations (of the semi-analog and non-analog MC schemes) in the same general framework/code, we encapsulated some key parts of the resolution in several functions: sample_interaction_time, compute_weight_modif, sample_velocity[21]. The three latter key functions are described in algorithms 7–8–9 but for the moment let us focus on the common canvas (i.e. algorithm 6).

Function *sample_interaction_time(int i, real* $\mathbf{v}$*, real* $\mathbf{X}$*)*
  set τ = REAL_MAX
  U =sample_uniform_law()
  if $MC\_scheme == semi - analog$ **then**
    $\tau = -\dfrac{\ln(U)}{v\sigma_t^i(\mathbf{v}, \mathbf{X})}$
  end
  if $MC\_scheme == non - analog$ **then**
    $\tau = -\dfrac{\ln(U)}{v\sigma_s^i(\mathbf{v}, \mathbf{X})}$
  end
  return τ
end

Algorithm 7: The sampling of the interaction time function depending on the choice of the MC scheme. The cross-sections are assumed constant in cell i.

Function *sample_velocity(real* $\mathbf{x}$*, real* $\mathbf{v}$*, real* $\mathbf{X}$*)*
  $\mathbf{V}'$ =sample_from_$P_s(\mathbf{x}, \mathbf{v}, \mathbf{X})$
  return $\mathbf{V}'$
end

Algorithm 8: Sampling of the velocity

In algorithm 6, we can see that each presented scheme relies on comparing three times, τ_{inter} the interaction time, τ_{exit} the time at which an MC particle p would get out of the cell i_p, τ_{census} the time before ending the time step. For each scheme, the particle moves along $\mathbf{v}_p\tau$ where τ is the minimum of the three above times. Its weight is modified or not (in compute_weight_modif) depending on the scheme. Furthermore, depending on the minimum of $\tau_{census}, \tau_{exit}, \tau_{inter}$, the particle sees its life time updated and finishes its treatment (*census*) or crosses the interface between two cells (*exit*) or encounters an interaction (*inter*). In the latter case, its velocity change. All the samplings potentially depends on the uncertain field $\mathbf{X}_p$ carried out by the uncertain MC particle p. The first and second order moments of the gPC coefficients are computed during the MC resolution. The instrumentation of the tracking corresponds to the tallying phases (i.e. the $+=$ operations in algorithm 6).

Let us now focus on the encapsulated functions. First, note that they all only depend on particle fields $(\mathbf{x}_p, \mathbf{v}_p, i_p, \mathbf{X}_p, ...)$. The first one, to sample the interaction time, only needs the particle energy $\mathbf{v}_p$ and the uncertain one $\mathbf{X}_p$ and is detailed in algorithm 7. Depending on the chosen scheme, the

[21] We do not detail the functions compute_time_until_cell_exit, find_cell and find_adjacent_cell as they depend more on the type of grid (cartesian, structured, unstructured) than on the MC resolution scheme.

interaction time is sampled from the total cross-section σ_t (semi-analog) or from the scattering one σ_s in the current cell i_p. Both are obtained inversing the cumulative density function of an exponential law.

The second encapsulated function corresponds to the modification of the weight of the particle, detailed in algorithm 9. For this function, the event the particle encounters explicitly appears in the treatment. The non-analog scheme is the only one having a treatment independent of the event. The weight of a particle remains unchanged for the semi-analog schemes for the *census* and *cell exit* events. It changes in the case of an interaction: for the semi-analog scheme, the weight is multiplied by the probability of being scattered $\frac{\sigma_s}{\sigma_t}$.

Function *compute_weight_modif(real* $\mathbf{v}$*, real* $\tau_{\min}$*, real* τ_{census}*, real* τ_{exit}*, real* τ_{inter}*, integer* i*, real* $\mathbf{X}$*)*
  set $K = 1$
  if $MC\_scheme == semi - analog$ **then**
    if $\tau_{\min} == \tau_{exit}$ *or* $\tau_{\min} == \tau_{census}$ **then**
      $K = 1$
    end
    if $\tau_{\min} == \tau_{inter}$ **then**
      $K = \dfrac{\sigma_s^i(\mathbf{v}, \mathbf{X})}{\sigma_t^i(\mathbf{v}, \mathbf{X})}$
    end
  end
  if $MC\_scheme == non - analog$ **then**
    $K = e^{-v(\sigma_t^i(\mathbf{v},\mathbf{X}) - \sigma_s^i(\mathbf{v},\mathbf{X}))\tau_{\min}}$
  end
  return K
end

Algorithm 9: The weight modification depending on the MC scheme

At the interaction time, each scheme needs the sampling of the outer velocity $\mathbf{V}'$, summed up in algorithm 8. We do not spend time commenting 8, its details mainly depends on the format of the velocity scattering file/data and is classical in MC codes.

Appendix D. The resolution of uncertain eigenproblems with MC-gPC and its new variant

In this section, we recall the sketch of the algorithm for the MC-gPC resolution of uncertain eigenproblems described in [9]. The difference, in this case, is simple: only one line needs to be added, cf. the cyan line in algorithm 10, in order to recompute the fission cross-section which depends on $k_{\text{eff}}(\mathbf{X})$. Note that this must be done for every material.

Data: Δt, N_{MC}, P
Result: The gPC coefficients $(k_{\text{eff}}^k)_{k\in\{0,...,P\}}$ of k_{eff} and eigenvector which can be built from list_of_particles
begin
#initialisation of a population of particles
list_of_particles=sampleUncertainParticles(N_{MC})
#old gPC coefficients of the number of physical particles in the domain $\mathcal{D}\times\mathcal{V}$
set $U_{\text{old}}^0 = 1$
#new gPC coefficients of the number of physical particles in the domain $\mathcal{D}\times\mathcal{V}$
set $U_{\text{new}}^0 = 1$
#gPC coefficients of the estimated eigenvalue
set $k_{\text{eff}}^0 = 1$
for $k \in \{1, ..., P\}$ **do**
$U_{\text{old}}^k = 0$
$U_{\text{new}}^k = 0$
$k_{\text{eff}}^k = 1$
end
while *iter* $<$ *iter_max* **do**
for *mat* $\in$ *list_of_materials* **do**
compute_fission_cross_section_from_$k_{\text{eff}}(R_{\text{mat}}, R_{\text{mat}}^{-1}, k_{\text{eff}}^0, ..., k_{\text{eff}}^P)$
end
$(U_{\text{new}}^k)_{k\in\{0,..,P\}}$=trackUncertainParticles(list_of_particles, $\Delta t, k_{\text{eff}}^0, ..., k_{\text{eff}}^P$)
#build punctual uncertain values
$(U_{\text{new}}^P(\mathbf{X}_g))_{g\in\{1,..,N_G\}} = buildPunctualValues((\mathbf{X}_g)_{g\in\{1,..,N_G\}}, (U_{\text{new}}^k)_{k\in\{0,...P\}})$
$(U_{\text{old}}^P(\mathbf{X}_g))_{g\in\{1,..,N_G\}} = buildPunctualValues((\mathbf{X}_g)_{g\in\{1,..,N_G\}}, (U_{\text{old}}^k)_{k\in\{0,...P\}})$
$(k_{\text{eff}}^P(\mathbf{X}_g))_{g\in\{1,..,N_G\}} = buildPunctualValues((\mathbf{X}_g)_{g\in\{1,..,N_G\}}, (k_{\text{eff}}^k)_{k\in\{0,...P\}})$
#update the gPC coefficients of the eigenvalue
for $k \in \{0, ..., P\}$ **do**
$$k_{\text{eff}}^k \leftarrow \sum_{g=1}^{N_G} k_{\text{eff}}^P(\mathbf{X}_g) \times \frac{U_{\text{new}}^P(\mathbf{X}_g)}{U_{\text{old}}^P(\mathbf{X}_g)} \phi_k(\mathbf{X}_g) w_g$$
end
#update the old number of physical particles
for $k \in \{0, ..., P\}$ **do**
$U_{\text{old}}^k \leftarrow U_{\text{new}}^k$
end
#apply a population control algorithm
uncertainPopulationControl(list_of_particles, N_{MC})
iter++
end
end

Algorithm 10: General canvas of a stochastic/uncertain k_{eff} calculation Monte Carlo code

Function compute_fission_cross_from_k_{eff} is described in algorithm 11. It needs the integration of the scattering cross-section with respect to the uncertain parameter $\mathbf{X}$ and the current values of gPC coefficients of the k_{eff} . It builds Λ_s which is used in trackUncertainParticles which is nothing more than a call to algorithm 1, see [9].

Function *compute_fission_cross_section*$(R, R^{-1}, k_{\text{eff}}^0, ..., k_{\text{eff}}^P)$

for $l \in \{0, ..., P\}$ **do**

for $k \in \{0, ..., P\}$ **do**

$\Sigma_s^{l,k} \leftarrow 0$

#Use of N_G Gauss quadrature points in order to integrate Σ_s

$$\Sigma_s^{l,k} \leftarrow \sum_{g=1}^{N_G} \left(\sigma_s(\mathbf{X}_g) + \frac{\nu_f(\mathbf{X}_g)}{\sum_{k=0}^{P} k_{\text{eff}}^k \phi_k(\mathbf{X}_g)} \sigma_f(\mathbf{X}_g) \right) \times \phi_k(\mathbf{X}_g)\phi_l(\mathbf{X}_g) w_g$$

end

end

$\Lambda_s = R^{-1}\Sigma_s R$

return Λ_s

end

Algorithm 11: The update of the fission cross-section across the stochastic power iteration of algorithm 10.