# SCALING OF INTRUSIVE STOCHASTIC COLLOCATION AND STOCHASTIC GALERKIN METHODS FOR UNCERTAINTY QUANTIFICATION IN MONTE CARLO PARTICLE TRANSPORT

**Aaron J. Olson**[1], **Brian C. Franke**[2], **and Anil K. Prinja**[1]
[1]Department of Nuclear Engineering
University of New Mexico
Albuquerque, NM 87123
aolson1@unm.edu; prinja@unm.edu
[2]Sandia National Laboratories
Albuquerque, NM 87185
bcfrank@sandia.gov

## ABSTRACT

A Monte Carlo solution method for the system of deterministic equations arising in the application of stochastic collocation (SCM) and stochastic Galerkin (SGM) methods in radiation transport computations with uncertainty is presented for an arbitrary number of materials each containing two uncertain random cross sections. Moments of the resulting random flux are calculated using an intrusive and a non-intrusive Monte Carlo based SCM and two different SGM implementations each with two different truncation methods and compared to the brute force Monte Carlo sampling approach. For the intrusive SCM and SGM, a single set of particle histories is solved and weight adjustments are used to produce flux moments for the stochastic problem. Memory and runtime scaling of each method is compared for increased complexity in stochastic dimensionality and moment truncation. Results are also compared for efficiency in terms of a statistical figure-of-merit. The memory savings for the total-order truncation method prove significant over the full-tensor-product truncation. Scaling shows relatively constant cost per moment calculated of SCM and tensor-product SGM. Total-order truncation may be worthwhile despite poorer runtime scaling by achieving better accuracy at lower cost. The figure-of-merit results show that all of the intrusive methods can improve efficiency for calculating low-order moments, but the intrusive SCM approach is the most efficient for calculating high-order moments.

*Key Words*: Stochastic Galerkin, Stochastic Collocation, Monte Carlo, Uncertainty Quantification

## 1 INTRODUCTION

Random sampling or Monte Carlo is the approach of choice in uncertainty quantification (UQ) computations due to ease of implementation (it is a non-intrusive method) and relatively weak dependence of convergence rate in the stochastic dimension on the number of uncertain input variables. However, when statistical moments higher than the mean and variance as well as the pdf of the quantity of interest (QOI) are desired, it is frequently more efficient to employ stochastic spectral methods that represent the uncertainty explicitly in terms of random orthogonal polynomials. These so-called polynomial chaos expansions when used in conjunction with stochastic Galerkin (SGM) [1] and stochastic collocation (SCM) [2] projection methods reduce the bulk of the computational effort to solving a system of non-random equations for the expansion coefficients. In traditional applications to radiation transport, the equations for the expansion coefficients are solved using deterministic methods [3–5], typically by adapting or directly employing the numerical schemes used to solve the problem in the absence of uncertainty. Application of SGM yields a coupled system of equations for the expansion coefficients while SCM yields uncoupled equations, but the equations otherwise resemble transport equations in structure.

In recent preliminary investigations based on two random cross sections [6, 7], it was shown that the system of equations obtained using SGM and SCM could also be solved efficiently using Monte Carlo simulation by interpreting the expansion coefficients as state variables and simulating transitions between the different states using particle-weight adjustment schemes. Here we extend this approach to an arbitrary number of random input variables by allowing different materials and examine the scaling efficiency of the Monte Carlo simulation with respect to number of random inputs. We specifically investigate the memory requirements of growing weight arrays, the runtime scaling of generalized algorithms within these methods, and the resultant efficiency of these various UQ methods for the solution of moments of the QOI.

The scope of this paper is as follows. In Section 2 we present the SGM and SCM equations appropriate to the radiation transport problem of interest, followed in Section 3 by a discussion of the use of weight adjustments in weight arrays for the intrusive UQ approaches. In Section 4 we investigate memory and runtime scaling of the methods, while in Section 5 we compare distributions of the QOI, particle flux, for methods against the brute force Monte Carlo benchmark values and compare efficiency in terms of a statistical figure-of-merit. We wrap up the investigation in Section 6 with some concluding remarks.

## 2 TRANSPORT EQUATIONS

Without loss of generality, we consider steady state transport of single speed particles, and express the transport equation with uncertain total and scattering cross sections as:

$$\vec{\Omega}\cdot\nabla\psi(\vec{r},\vec{\Omega},\omega)+\Sigma_t(\vec{r},\omega)\psi(\vec{r},\vec{\Omega},\omega)=\frac{\Sigma_s(\vec{r},\omega)}{4\pi}\phi(\vec{r},\omega), \tag{1}$$

with a boundary source. The label $\omega$ indicates a particular stochastic realization and $\psi(\vec{r},\vec{\Omega},\omega)$ is the random angular flux. Otherwise standard notation is used. The uncertainty can be expressed for total or scattering cross sections succinctly as:

$$\Sigma_{t,s}(\vec{r},\omega)=\langle\Sigma_{t,s}(\vec{r})\rangle+\widehat{\Sigma}_{t,s}(\vec{r})\xi_{t,s}(\omega), \tag{2}$$

where $\langle\Sigma_{t,s}(\vec{r})\rangle$ are the mean cross sections, $\widehat{\Sigma}_{t,s}$ are proportional to standard deviations, and $\xi_{t,s}(\omega)$ are zero-mean independent and identically distributed random variables, for this investigation uniformly distributed over the range [-1,1]. The selection of this distribution of randomness leads to the Legendre polynomials as the natural basis for the polynomial chaos expansion. However, our approach applies to any choice of chaos polynomials, with only differences in the resulting coefficients and quadratures in the SGM and SCM implementations.

The randomness in the angular flux is represented through the polynomial chaos expansion in $2M$-dimensional random Legendre polynomials:

$$\psi(\vec{r},\vec{\Omega},\vec{\xi}(\omega)) = \psi(\vec{r},\vec{\Omega},\xi_{t_1}(\omega),\xi_{s_1}(\omega),\cdots\xi_{t_M}(\omega),\xi_{s_M}(\omega))$$
$$= \sum_{l_1=0}^{\infty}\sum_{m_1=0}^{\infty}...\sum_{l_M=0}^{\infty}\sum_{m_M=0}^{\infty}\psi_{l_1,m_1,...,l_M,m_M}(\vec{r},\vec{\Omega})P_{l_1}(\xi_{t_1}(\omega))P_{m_1}(\xi_{s_1}(\omega))...P_{l_M}(\xi_{t_M}(\omega))P_{m_M}(\xi_{s_M}(\omega)) \quad (3)$$

where $M$ is the number of materials in the system and the deterministic expansion coefficients $\psi_{l_1,m_1,...,l_M,m_M}(\vec{r},\vec{\Omega})$ are related to the angular flux by orthogonality of the Legendre polynomials with inner product defined over the appropriate probability space:

$$\psi_{l_1,m_1,...,l_M,m_M}(\vec{r},\vec{\Omega}) = a_{l_1m_1...l_Mm_M}\int_{-1}^{1}\int_{-1}^{1}...\int_{-1}^{1}\int_{-1}^{1}\psi(\vec{r},\vec{\Omega},\xi_{t_1},\xi_{s_1},...,\xi_{t_M},\xi_{s_M})$$
$$P_{l_1}(\xi_{t_1})P_{m_1}(\xi_{s_1})...P_{l_M}(\xi_{t_M})P_{m_M}(\xi_{s_M})d\xi_{t_1}d\xi_{s_1}...d\xi_{t_M}d\xi_{s_M}, \quad (4)$$

where $a_{l_1m_1...l_Mm_M} = \prod_{j=1}^{M}(2l_j+1)(2m_j+1)$. Note that the randomness in the angular flux arises from the random cross sections in Eq. (3).

## 2.1 SGM Transport Equations

To simplify notation in the following equations, we suppress the subscripts by defining $\psi(\vec{r},\vec{\Omega}) \equiv \psi_{l_1,m_1,...,l_M,m_M}(\vec{r},\vec{\Omega})$ and other flux expansion coefficients are defined in relative terms, such as $\psi_{m_1-1}(\vec{r},\vec{\Omega}) \equiv \psi_{l_1,m_1-1,...,l_M,m_M}(\vec{r},\vec{\Omega})$.

Projecting Eq. (1) over the chaos functions, using the recurrence relationship:

$$\xi P_m(\xi) = \frac{m+1}{2m+1}P_{m+1}(\xi) + \frac{m}{2m+1}P_{m-1}(\xi), \quad (5)$$

and noting the orthogonality of Legendre polynomials, the following coupled system of SGM transport equations are obtained:

$$\vec{\Omega}\cdot\nabla\psi(\vec{r},\vec{\Omega}) + \langle\Sigma_{t,j}\rangle\psi(\vec{r},\vec{\Omega}) + \widehat{\Sigma}_{t,j}\Big[\Big(\frac{l_j+1}{2l_j+3}\Big)\psi_{l_j+1}(\vec{r},\vec{\Omega}) + \Big(\frac{l_j}{2l_j-1}\Big)\psi_{l_j-1}(\vec{r},\vec{\Omega})\Big] =$$
$$\frac{\langle\Sigma_{s,j}\rangle}{4\pi}\phi(\vec{r}) + \frac{\widehat{\Sigma}_{s,j}}{4\pi}\Big[\Big(\frac{m_j+1}{2m_j+3}\Big)\phi_{m_j+1}(\vec{r}) + \Big(\frac{m_j}{2m_j-1}\Big)\phi_{m_j-1}(\vec{r})\Big],$$
$$l_1,m_1,...,l_M,m_M = 0,\infty \quad (6)$$

where j is the material at position $\vec{r}$. This infinite set of coupled equations is truncated in one of two ways. The first "total-order" truncation method (SG-1) truncates based upon the sum of the orders in all dimensions by setting $\psi_{l_1,m_1,...,l_M,m_M} = 0$ for all $\sum_{k=1}^{M} l_k + m_k \geq K$. The second "full-tensor-product" truncation method (SG-2) truncates based upon the order in each dimension independently by setting $\psi_{l_1,m_1,...,l_M,m_M} = 0$ for all $l_k \geq K$ and $m_k \geq K$. SG-2 is somewhat simpler to implement and calculates more moments, but as more random variables are considered, SG-1 produces significant memory savings and greater efficiency for moments solved, issues which are investigated later in this paper. SGM truncation requires truncation of the same type and order in each dimension.

## 2.2 SCM Transport Equations

For the stochastic collocation method, the multidimensional integral in Eq. (4) is approximated by a multidimensional quadrature rule of order K in each of $2M$ dimensions:

$$\psi_{l_1,m_1,...,l_M,m_M}(\vec{r},\vec{\Omega}) \approx a_{l_1 m_1 ... l_M m_M} \sum_{k_1=1}^{K} \sum_{n_1=1}^{K} ... \sum_{k_M=1}^{K} \sum_{n_M=1}^{K} w_{k_1} w_{n_1} ... w_{k_M} w_{n_M} \\ \psi_{k_1 n_1 ... k_M n_M}(\vec{r},\vec{\Omega}) P_{l_1}(\xi_{t_1}^{k_1}) P_{m_1}(\xi_{s_1}^{n_1}) ... P_{l_M}(\xi_{t_M}^{k_M}) P_{m_M}(\xi_{s_M}^{n_M}), \quad (7)$$

where $\psi_{k_1 n_1 ... k_M n_M}(\vec{r},\vec{\Omega}) \equiv \psi(\vec{r},\vec{\Omega},\xi_{t_1},\xi_{s_1},...,\xi_{t_M},\xi_{s_M})$ is the angular flux evaluated by fixing $\xi$ values at quadrature points and computing the cross section using Eq. (2) and $(\xi_t^k, w_k)$ are quadrature ordinates and weights. We limit considerations here to full tensor-product quadratures and defer to a future investigation the use of sparse-grid quadratures and other dimension reducing approaches that would yield greater computational efficiency.

Truncation with either the SGM or the SCM will cause error, though a larger K will cause less error. This error is investigated for one problem at two values for K later in this paper and for practical use a convergence study is likely the best way to determine the optimal choice for K.

# 3 MONTE CARLO SOLUTION METHODS

In this section we investigate how particle-weight adjustments are performed in each SGM method and in the intrusive SCM method. In each of these, the particle being simulated is replaced by a multi-dimensional matrix of particle weights, which the Monte Carlo tallies can be modified to accept. For the SGM approaches, each weight value in the matrix corresponds to one equation and flux moment $\psi_{l_1,m_1,...,l_M,m_M}$. Since this flux moment is the desired quantity for this research, the statistical uncertainty on the moments is calculated directly within the simulation. For the SCM intrusive approach each weight value in the matrix corresponds to a flux coefficient $\psi_{k_1,n_1,...,k_M,n_M}$. For efficiency, these coefficients are collected in batches and Eq. (7) is used to estimate the mean and variance of each moment.

Our non-intrusive SCM approach is the simplest alternative to a Monte Carlo approach. No modification is required of a standard Monte Carlo particle transport code. The code is simply executed for each combination of cross section values, which are dictated by the quadrature points applied to Eq. (2). This approach performs most efficiently if each simulation uses a correlated random number sequence, i.e. corresponding histories of each separate calculation are initiated with the same random number seed in the same generation sequence. This is a standard approach in most Monte Carlo codes. We designate this method as CR-SC, for correlated random-number stochastic collocation.

The details of SGM and SCM weight adjustments have been described in previous work [7]. In summary the SG-I (stochastic Galerkin interaction-based) weight adjustment method rearranges the coupled system in Eq. (6) to form a matrix inversion for weight adjustments at non-scattering events, associated with the $\widehat{\Sigma}_t$ terms, and another at scattering events, associated with the $\langle \Sigma_s \rangle$ and $\widehat{\Sigma}_s$ terms. The SG-S (stochastic Galerkin streaming-based) weight adjustment method treats $\langle \Sigma_t \rangle$ and $\widehat{\Sigma}_t$ terms together to arrive at weight adjustments for all particle streaming events through an eigenvalue solve involving the distance traveled. SG-S uses a nearly identical interaction-based weight-adjustment scheme for scattering events as the SG-I approach. Our intrusive SCM implementation is patterned after the correlated-sampling technique used in Monte Carlo biasing. We designate this approach CS-SC, for correlated-sampling stochastic collocation. There is no coupling of the terms in the SCM weight array, in which each element represents a tensor-product quadrature point. For each sampled streaming or interaction event, weight adjustments multiply by the ratio of the probability of the event with the quadrature-point cross sections to the probability of the event for the cross sections in the simulation.

For the SGM each weight adjustment is performed coupling two dimensions of the array which describes the relationship between moments of the two uncertain cross sections in the material that the particle is currently in. In order to evaluate these relationships, the moments which form rows along the two dimensions of interest are extracted from the full weight array, operated upon, and returned. Due to the multi-dimensionality of the flux moments, these values are usually not located contiguously in computer memory, requiring extra effort in collecting the values to be operated on. For each weight-adjustment event, each value in the weight array undergoes exactly one weight-adjustment operation.

After estimating the flux moments, flux probability distributions can be obtained by sampling the polynomial chaos expansion given in Eq. (3). Each combination of random samples creates one realization of the solution of the transport equation and through the compilation of samples, a probability distribution of the flux can be generated as a function of space and angle.

## 4 SCALING

### 4.1 Memory Scaling

Here the memory scaling of the two SGM truncation methods is investigated as a function of the selected SGM truncation order ($K$) and number of materials ($M$). The SG-1 truncation method creates a weight matrix that is the shape of a $2M$-dimensional simplex of length $K$ in each dimension. The SG-2 truncation method creates a weight matrix that is the shape of a $2M$-dimensional hypercube of length $K$ in each dimension. The SCM tensor-product quadrature used for the collocation grid in this paper is equivalent to the SG-2 polynomial-order truncation method, and will be examined in the following section. For SCM it is possible to achieve moment preservation similar to the SG-1 approach with comparable memory savings through the use of sparse-grid quadrature. Such alternative quadratures are not investigated here.

In order to accommodate an input-defined number of dimensions, the weight matrix is collapsed into a 1-dimensional array. The size of this weight array for the SG-1 method is calculated as the binomial coefficient $\binom{2M+K}{K}$, and for the SG-2 method as $K^{2M}$. The size of the weight array for the SG-1 and SG-2 truncation methods as a function of the number of materials (M) and truncation order (K) are recorded in Tables I and II. We note that the tensor-product SG-2 approach requires rapidly increasing amounts of memory with increasing truncation order or number of materials, and the method quickly becomes untenable within practical computer constraints. By contrast, the SG-1 approach enables the incorporation of some higher-order moments and permits evaluation of numerous material uncertainties. We have verified that, beyond the smallest weight array sizes, memory usage in our code scales appropriately and is approximately 7 times the array sizes listed in Tables I and II. The factor of 7 is due to additional weight-array-sized memory usage in both reflection and transmission flux tallies, and associated statistical-estimation arrays.

**Table I. SG-1 Size of Weight Array**

| | K=2 | K=3 | K=4 | K=5 | K=6 | K=7 |
|---|---|---|---|---|---|---|
| M=1 | 3 | 6 | 10 | 15 | 21 | 28 |
| M=2 | 5 | 15 | 35 | 70 | 126 | 210 |
| M=3 | 7 | 28 | 84 | 210 | 462 | 924 |
| M=4 | 9 | 45 | 165 | 495 | 1287 | 3003 |
| M=5 | 11 | 66 | 286 | 1001 | 3003 | 8008 |
| M=6 | 13 | 91 | 455 | 1820 | 6188 | $1.86 \cdot 10^4$ |

**Table II. SG-2 Size of Weight Array**

| | K=2 | K=3 | K=4 | K=5 | K=6 | K=7 |
|---|---|---|---|---|---|---|
| M=1 | 4 | 9 | 16 | 25 | 36 | 49 |
| M=2 | 16 | 81 | 256 | 625 | 1296 | 2401 |
| M=3 | 64 | 729 | 4096 | $1.56 \cdot 10^4$ | $4.67 \cdot 10^4$ | $1.18 \cdot 10^5$ |
| M=4 | 256 | 6561 | $6.55 \cdot 10^4$ | $3.91 \cdot 10^5$ | $1.68 \cdot 10^6$ | $5.76 \cdot 10^6$ |
| M=5 | 1024 | $5.90 \cdot 10^4$ | $1.05 \cdot 10^6$ | $9.77 \cdot 10^6$ | $6.05 \cdot 10^7$ | $2.82 \cdot 10^8$ |
| M=6 | 4096 | $5.31 \cdot 10^5$ | $1.68 \cdot 10^7$ | $2.44 \cdot 10^8$ | $2.18 \cdot 10^9$ | $1.38 \cdot 10^{10}$ |

### 4.2 Runtime Scaling

Runtime scaling of the various methods informs how useful a method will be as a problem increases in complexity. These problems can increase in complexity as either the number of materials (M) or the order of truncation (K) grows. Here a simple problem has been constructed to test the runtime scaling of the various methods as complexity is increased in these dimensions. Each problem is solved with a unit source normally incident on the left boundary of a slab of 1 cm with a vacuum condition on the right boundary. Particles are monoenergetic and scatter isotropically.

For the simplest case, this problem is defined as one material spread over 10 equally spaced material segments in a 1D slab. The next case defines a second material whose physical properties are identical to those of the original material and occupies half of the material segments in an alternating fashion such that roughly half of the weight adjustments are performed in this second material. The third case defines a third material which is physically identical to the first two, and the three materials are distributed in an alternating fashion. This process is carried out for a problem with up to five physically identical materials. The material arrangements are summarized in Table III. For each of these cases truncation values of 2 through 6 are tested. This problem suite allows the same physical problem to be solved with varying levels of methodological complexity isolating runtime differences that are due to such increased complexity.

**Table III. Material Arrangements for Runtime Scaling Problems**

| Num of Materials | Material Arrangements | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| M=1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| M=2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 |
| M=3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 |
| M=4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 |
| M=5 | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 |

The metric chosen for runtime comparison was the runtime per history divided by the number of moments solved by the method divided by the average runtime per history in the average-valued realization of the benchmark solution. It thus compares the runtime per history per moment to the runtime per history for an average independent realization. Results of these scaling demonstrations for each SGM and SCM implementation are plotted in Figure 1. The reader should keep in mind that these runtime scaling results do not directly convey that fewer moments are being calculated per order with the SG-1 methods.

As expected, the non-intrusive CR-SC method produces nearly constant values. The relative runtime for CR-SC is greater than 1.0 because it performs extra computation in cycling through grid points and setting random number seeds. The intrusive CS-SC implementation is faster than the CR-SC method and also remains mostly constant with increased problem complexity, where each new point to the weight array for this problem simply means one more weight adjustment to perform per interaction.

The SGM implementations behave more interestingly. The SG-2S implementation has a hypercube shaped weight grid which requires the solution of only one eigenvalue problem for each streaming event and thus appears to scale in direct proportion with the number of flux moments represented in the weight array. The SG-1S implementation has a simplex shaped weight grid causing cuts over which the weight adjustments are performed to be of different sizes and requiring the solution of $K$ eigenvalue problems for each streaming event. To its benefit some applications of the solution of these eigenvalue problems are not needed as entire rows of the array have been eliminated by truncation. With increasing complexity, the increased number of required eigenvalue solves begins to dominate scaling behavior. The SG-2I implementation does not maintain the directly proportionate scaling of its streaming-based counterpart since each weight adjustment requires a matrix inversion. When the matrices are small this cost is not overwhelming, but cost increases with increased matrix size. The SG-1I implementation solves the same number of matrix inversions as the SG-2I method, but most of the inversions involve smaller matrices due to truncation.

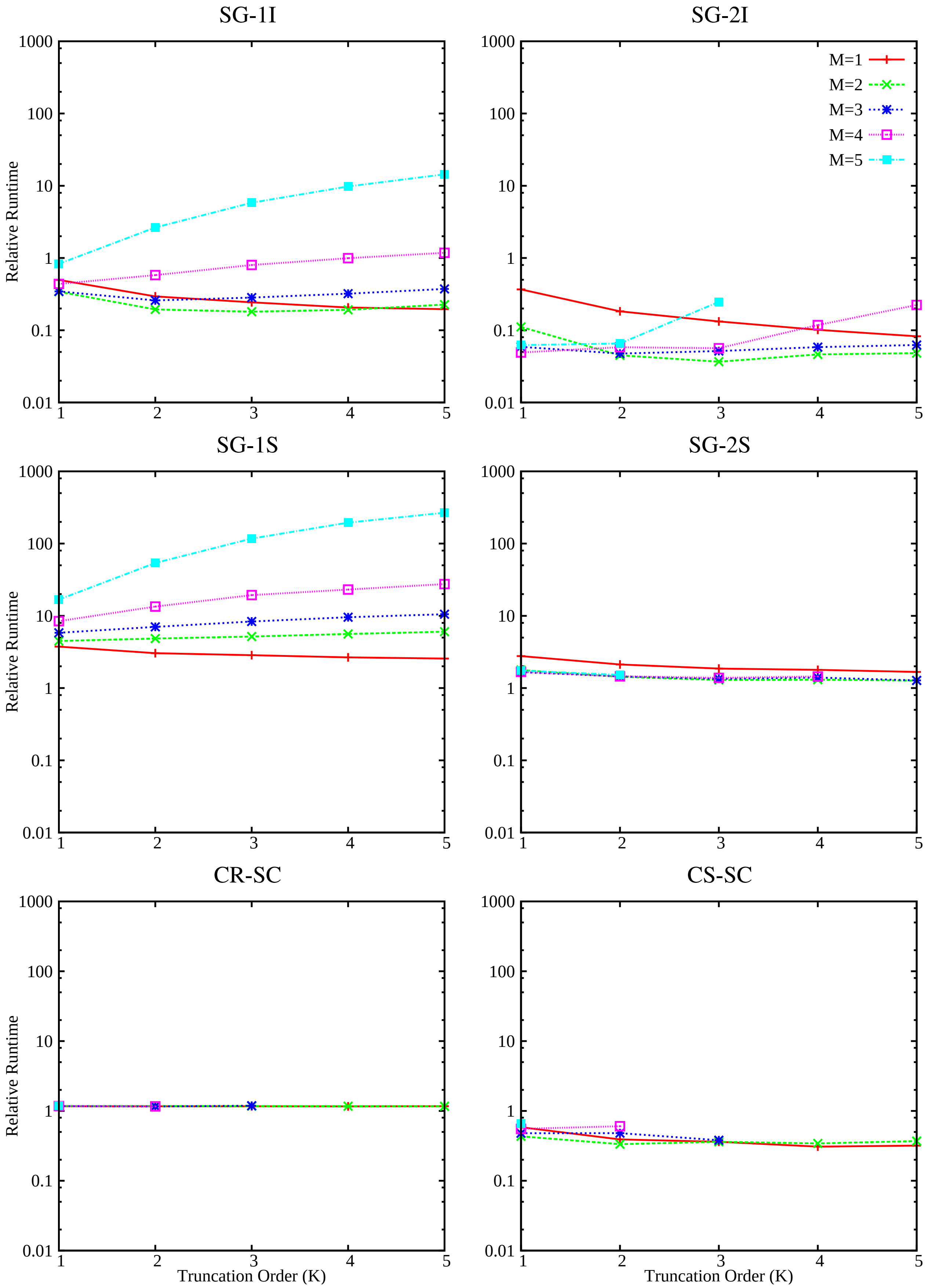

**Figure 1. Runtime Scaling for Stochastic Methods**

It should be noted that there is some increased cost for the SG-1 methods over the SG-2 methods in extracting and returning cuts of the weight array on which to operate since the math for performing such a referencing is not as convenient and in this implementation includes a binary search. While the $K = 5, M = 5$ case for SG-1S is about 100 times slower per moment than SG-2S, it can be seen from Tables I and II that the SG-2S implementation requires solving almost $10^4$ more moments.

## 5 RESULTS

In this section, the accuracy and efficiency of the methods are examined through a test problem evaluated at two different truncation orders $K$. Particles are monoenergetic and scatter isotropically. The boundary conditions are a normally incident source on the left side and vacuum on the right side of the slab geometry. The slab has a thickness of 1 cm, and is broken into four equally sized segments which hold the specified material. A description of the two materials and their arrangement in the slab is seen in Table IV.

**Table IV. Material Arrangement and Definitions for Test Problem**

| Material Arrangement | | | |
|---|---|---|---|
| 1 | 2 | 2 | 1 |

| Material Definitions | | | | |
|---|---|---|---|---|
| Material<br>Number | $\langle \Sigma_t \rangle$<br>(cm$^{-1}$) | $\langle \Sigma_s \rangle$<br>(cm$^{-1}$) | $\widehat{\Sigma}_t$<br>(cm$^{-1}$) | $\widehat{\Sigma}_s$<br>(cm$^{-1}$) |
| 1 | 5.0 | 3.5 | 1.0 | 0.4 |
| 2 | 5.0 | 1.0 | 0.5 | 0.4 |

Probability density functions of transmitted flux are created using the new methods (SG-1S and SG-2S) at various truncation orders and are compared against a Monte Carlo benchmark in Figure 2. The SGM solutions were generated by simulating $1 \times 10^8$ particle histories. Benchmark results were generated by randomly sampling $1 \times 10^7$ realizations and simulating each of them with $1 \times 10^6$ particle histories. With increasing truncation order, the pdfs from the SGM methods appear to be converging to the benchmark solution. For this particular problem a truncation order of $K = 2$ gives only a crude approximation of the actual pdf. Truncation at $K = 3$ does considerably better, but still fails to capture the correct peak of the distribution. At $K = 4$ and higher, the SGM methods appear to accurately represent the pdf.

We can also compare the pdfs of the two truncation methods in Figure 2. For $K = 2$ the SG-2 method more accurately calculates the lowest and highest flux values, while both have an inaccurate plateau shape. For $K = 3$, the SG-2 more accurately represents the peak and slopes of the pdf. This more accurate solution for the SG-2 truncation than the SG-1 truncation can be attributed to the contributions of the higher-order cross-moments. Nevertheless, both truncation schemes appear to be converging to the correct result.

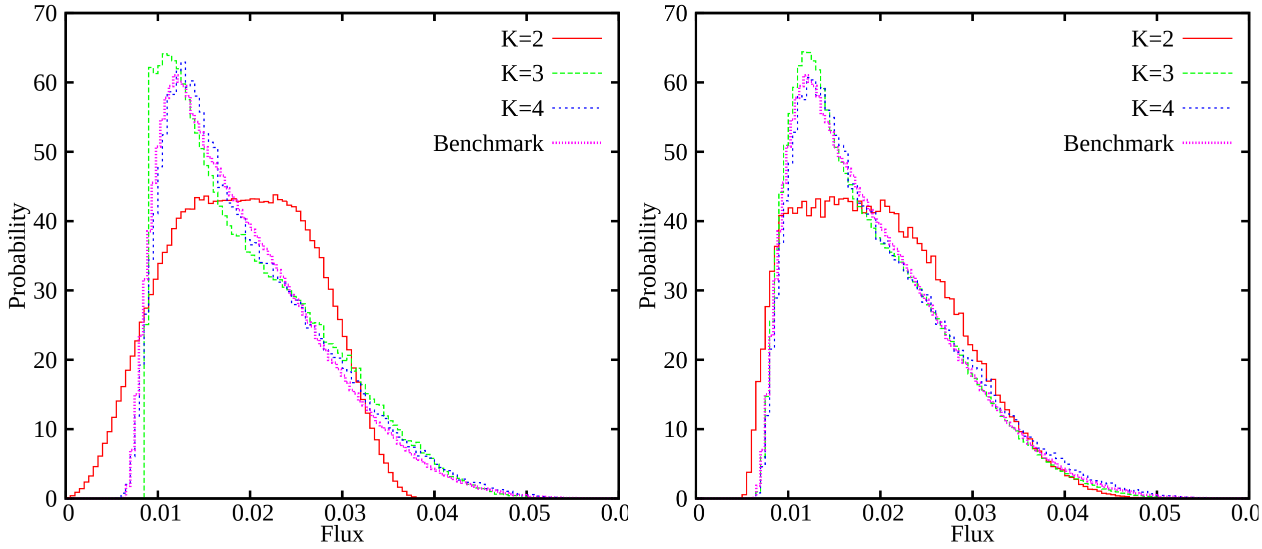

**Figure 2. Transmission probability distributions for SG-1S (left) and SG-2S (right)**

Another metric for comparing these methods is the efficiency of solving various moments. Statistical uncertainty and runtime are compared through a figure-of-merit:

$$FOM = \frac{1}{R^2 T}, \tag{8}$$

where $T$ is computational time and $R$ is the relative error calculated as the standard deviation of the flux moment divided by its mean.

The FOM values of seven methods are compared in calculating selected flux moments. The first method is the benchmark using brute force Monte Carlo (MC). The second method evaluates each quadrature point with correlated random number seeds (CR-SC) by setting the seed for each history at each collocation point to be the same random number seed as corresponding histories at the other points. The third is the intrusive correlated-sampling SCM implementation (CS-SC). The mean flux moment values from this method are those reported in Tables V and VI. Fourth through seventh are SGM implementations using all four combinations of truncation methods and the interaction-based versus streaming-based moment-coupling methods. Each of these methods are solved for truncation order $K = 3$ in Table V and $K = 4$ in Table VI. The benchmark was generated from $1 \times 10^7$ realizations each solved with $1 \times 10^6$ histories. The SCM and SGM implementations were solved using $1 \times 10^8$ histories each. FOMs are not included for moments that did not achieve at least a 10% uncertainty, because the FOMs are not expected to be reliable estimates. Also FOMs are not included for the SG-1 truncation for the higher-order moments which the scheme does not calculate.

**Table V. Efficiency Figures-of-Merit for K=3**

| Flux Moment | | Figure of Merit | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | SCM | | SGM | | | |
| $l_1$,$m_1$,$l_2$,$m_2$ | $\psi_{l_1,m_1,l_2,m_2}$ | MC | CR-SC | CS-SC | SG-1I | SG-2I | SG-1S | SG-2S |
| 0,0,0,0 | 1.9E-2 | 1.9E+0 | 8.4E+1 | 1.7E+2 | 6.8E+2 | 4.9E+2 | 2.4E+2 | 1.4E+2 |
| 1,0,0,0 | -1.2E-2 | 3.6E-2 | 2.5E+1 | 1.3E+2 | 2.4E+2 | 1.8E+2 | 1.5E+2 | 9.0E+1 |
| 2,0,0,0 | 3.0E-3 | 1.2E-3 | 1.3E+0 | 6.4E+1 | 2.4E+1 | 1.7E+1 | 5.8E+1 | 3.4E+1 |
| 1,1,0,0 | -2.7E-3 | 4.9E-4 | 1.7E+1 | 7.7E+1 | 2.7E+1 | 1.8E+1 | 2.4E+1 | 1.3E+1 |
| 2,2,0,0 | 1.3E-4 | $a$ | 2.5E+0 | 2.5E+1 | $b$ | 5.3E-1 | $b$ | 1.7E+0 |
| 1,1,1,1 | 1.0E-4 | $a$ | 1.3E+1 | 7.0E+1 | $b$ | $a$ | $b$ | 9.6E-1 |
| 2,2,2,2 | 1.3E-7 | $a$ | 2.4E-1 | 4.2E+1 | $b$ | $a$ | $b$ | 2.1E-2 |
| Time (s): | | 2.9E+6 | 21437 | 7759 | 600 | 843 | 3649 | 5941 |

$a$ FOM values omitted due to poor statistical convergence

$b$ Moment not calculated with this truncation method

**Table VI. Efficiency Figures-of-Merit for K=4**

| Flux Moment | | Figure of Merit | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | SCM | | SGM | | | |
| $l_1$,$m_1$,$l_2$,$m_2$ | $\psi_{l_1,m_1,l_2,m_2}$ | MC | CR-SC | CS-SC | SG-1I | SG-2I | SG-1S | SG-2S |
| 0,0,0,0 | 1.9E-2 | 1.9E+0 | 3.4E+1 | 5.2E+1 | 2.7E+2 | 1.6E+2 | 9.1E+1 | 5.0E+1 |
| 1,0,0,0 | -1.2E-2 | 3.6E-2 | 1.3E+1 | 3.9E+1 | 9.6E+1 | 5.9E+1 | 5.6E+1 | 3.1E+1 |
| 2,0,0,0 | 3.0E-3 | 1.2E-3 | 6.3E-1 | 1.9E+1 | 8.9E+0 | 5.3E+0 | 2.0E+1 | 1.1E+1 |
| 3,0,0,0 | -5.4E-4 | 3.1E-5 | 1.7E-2 | 7.1E+0 | 1.5E+0 | 9.1E-1 | 5.9E+0 | 3.1E+0 |
| 1,1,0,0 | -2.7E-3 | 4.9E-4 | 8.3E+0 | 2.4E+1 | 1.0E+1 | 6.2E+0 | 9.9E+0 | 4.6E+0 |
| 2,2,0,0 | 1.4E-4 | $a$ | 9.1E-1 | 7.1E+0 | $b$ | 1.5E-1 | $b$ | 5.2E-1 |
| 3,3,0,0 | -5.7E-6 | $a$ | 1.1E-1 | 2.2E+0 | $b$ | 4.6E-2 | $b$ | 7.6E-2 |
| 1,1,1,1 | 1.1E-4 | $a$ | 4.8E+0 | 2.1E+1 | $b$ | $a$ | $b$ | 3.3E-1 |
| 2,2,2,2 | 1.4E-7 | $a$ | 1.3E-1 | 1.0E+1 | $b$ | $a$ | $b$ | $a$ |
| 3,3,3,3 | 1.3E-10 | $a$ | 1.2E-3 | 3.3E+1 | $b$ | $a$ | $b$ | $a$ |
| Time (s): | | 2.9E+6 | 66363 | 25318 | 1526 | 2558 | 8994 | 17295 |

$a$ FOM values omitted due to poor statistical convergence

$b$ Moment not calculated with this truncation method

As expected, all the SGM and SCM results outperform the brute-force Monte Carlo approach. The MC approach is not penalized by increased dimensionality like these other methods, but is slow at solving higher moments. The CR-SC implementation, though it barely converged the solution for the higher moments with this problem, still converged each of the reported moments with better statistics on mutually converged moments than the MC approach ranging from about 44 to 35,000 times more efficient for the $K = 3$ case and from about 18 to 17,000 times more efficient for the $K = 4$ case. The CS-SC implementation improves from the CR-SC method in figure-of-merit ranging in the $K = 3$ case from about 2 to about 175 times more efficient, and in the $K = 4$ case from about 1.5 to about 27,500 times more efficient. Whereas the CR-SC improves some on a naive implementation of the SCM by correlating the random number seeds for each history, the CS-SC method also ensures correlated particle tracks and uses weight adjustments to require only one particle simulation. The CS-SC method would be expected to handle more complicated physics better than the CR-SC and MC methods, as the proportionate cost of particle simulations to weight adjustments would be expected to increase, though the amount of dedicated intrusive code also grows with more complicated physics for the CS-SC method, and not the others. Further savings for the SCM methods could likely be introduced by using sparse and/or adaptive collocation grids.

In all cases the SG-1 methods produced about 1.5 to 2 times better figures-of-merit than their SG-2 counterparts, although the SG-1 truncation calculates less moments. The SG-I implementations range from about 3 times more efficient than the SG-S implementations in low-order moments to about 3 times less efficient at higher order moments. This shows that the SG-S streaming calculations favor the higher-order moments compared to the SG-I approach which takes longer to populate them through matrix inversions. Among the SGM implementations, SG-1I is the most efficient at producing the lowest-order moments. At slightly higher-order moments, the SG-1S is the most efficient. Beyond the coverage of the SG-1 truncation, SG-2S produces statistically meaningful results to higher moment combinations and with greater efficiency than SG-2I.

The SGM implementations perform in most cases better than the SCM implementations for the lowest order moments, but for higher moments fall behind both SCM implementations and converge fewer moments for this problem. The CR-SC method is a considerable improvement on the MC approach for this problem while maintaining non-intrusive implementation, though the highest order moments almost did not converge, where the CS-SC implementation converged all of these moments with some margin and greater efficiency. For problems involving greater physical complexity than these, the CS-SC approach would be expected to continue to outperform the CR-SC method by a larger margin and be more difficult to implement by a larger margin, though it is unclear which of the six implementations would scale best with increased complexity.

## 6 CONCLUSIONS

We have implemented a variety of Monte Carlo approaches for the stochastic Galerkin method (SGM) and the stochastic collocation method (SCM) for quantifying the uncertainty in particle flux due to uncertainty in cross sections and examined the memory and runtime scaling and efficiency of the methods. Different methods yielded different advantages and disadvantages in implementation, memory, runtime, and solution efficiency.

We calculated memory scaling requirements and examined runtime scaling compared to a benchmark runtime for increased complexity in truncation order and number of materials. The total-order truncation method used in our SGM implementations scaled in memory as binomial coefficients. The tensor-product truncation method used in our other SGM and our SCM implementations scaled exponentially with the number of materials. The first method thus shows only minor memory savings for problems with few materials, but for problems with many materials it may be the only feasible truncation choice. The SCM methods maintained a somewhat consistent runtime per particle per collocation point. The SG-2S implementation also maintained a somewhat consistent runtime per particle per flux moment, though the other SGM methods showed increase in this metric, especially with the increase in number of materials.

We solved one monoenergetic, isotropically scattering test problem at two different truncation orders and examined computational efficiency. All SGM and SCM approaches outperform the MC benchmark approach. The CS-SC method shows better figures-of-merit in both cases than the CR-SC method, though both solve high-order cross moments at least as efficiently as the SGM approaches. The SGM and CS-SC implementations are increasingly less efficient at higher-order moments. The total-order truncation method produces better efficiency for moments solved, but does not solve as many moments. The interaction-based SGM solves lower-order moments more efficiently, whereas the streaming-based SGM solves higher-order moments more efficiently.

These methods could be applied to uncertainty distributions which vary in a way other than uniformly by simply choosing a polynomial expansion appropriate to the distributions. The introduction of sparse or adaptive grids would likely increase the efficiency of the SCM implementations. Dimensional Reduction methods such as HDMR may allow application to problems with many more uncertain variables. More complex problems in geometry, particle physics, or solving with more materials may be of interest and may produce different relative efficiencies among the methods.

## 7 ACKNOWLEDGMENTS

The first author gratefully acknowledges the support of the Department of Energy under the Nuclear Energy University Programs Graduate Fellowship. Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

## 8 REFERENCES

[1] R. G. Ghanem and P. D. Spanos, *Stochastic finite elements: a spectral approach*, Dover (2003).

[2] L. Mathelin, M. Y. Hussaini, and T. A. Zang, "Stochastic Approaches to Uncertainty Quantification in CFD simulations," *Numer. Algorithms*, **38**, pp. 209 (2005).

[3] M. M. R. Williams, "Polynomial Chaos Functions and Neutron Diffusion," *Nucl. Sci. Eng.*, **155**, pp. 109 (2007).

[4] E. D. Fichtl and A. Prinja, "The Stochastic Collocation Method for Radiation Transport in Random Media," *Journal of Quantitative Spectroscopy and Radiative Transfer*, **112**, pp. 646 (2011).

[5] E. D. Fichtl, *Stochastic Methods for Uncertainty Quantification in Radiation Transport*, The University of New Mexico (2009).

[6] B. C. Franke and A. K. Prinja, "Efficient Monte Carlo Solution for Uncertainty Propagation with Stochastic Collocation," *Trans. Am. Nucl. Soc.*, **106**, pp. 346 (2012).

[7] B. C. Franke and A. K. Prinja, "Monte Carlo Solution for Uncertainty Propagation in Particle Transport with a Stochastic Galerkin Method," *International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering (M&C 2013)*, Sun Valley, ID, May, 2013, American Nuclear Society.