Tests for UQ Framework

Paul Talbot

March 6, 2014

1 Introduction

This work outlines the development of an uncertainty quantification (UQ) framework using generalized polynomial chaos expansions and stochastic collocation (PCESC), verified using Monte Carlo (MC) sampling. The intended use is as a "black-box wrapper," agnostic of the algorithm whose uncertaintly is quantified. To verify the several stages this framework undergoes in development and its independence from any deterministic solver, we present here several test codes of increasing complexity that the UQ framework will act on. The four test codes solve four problems: a polynomial expression; 1D mono-energetic neutron transport in a semi-infinite medium with uniform source and single material; 1D k-eigenvalue neutron diffusion transport with two energy groups and a single material; and a 2D, two energy group k-eigenvalue neutron diffusion transport quarter-core benchmark.

1.1 Algorithm

Each problem-solving code is treated as a black box that reads in an input file and produces a result readable from an output file. The problem-solving code can be represented as a function U of certain input parameters θ in deterministic parameter space Θ and uncertain parameters $Y(\omega)$ in uncertainty space Γ, where Y could be a single parameter or a vector of uncertain parameters and ω is a single realization in the uncertainty space Γ. We expand $U(\theta, Y)$ in basis polynomials characteristic of the uncertain parameters:

$$U(\theta; Y) \approx U_P(\theta; Y) \equiv \sum_{p=0}^{P} u_p(\theta)\psi_p(Y), \tag{1}$$

Generally, we omit the dependency θ when considering stochastic space ($U(\theta; Y) = U(\theta)$). $u_p(\theta)$ are polynomial expansion coefficients, $\psi_p(Y)$ are orthonormal basis polynomials, and the sum is necessarily truncated at finite order P. In the limit as P approaches infinity (or if $U(Y)$ can be expressed exactly as a polynomial of order P), there is no approximation. Ideally the expansion converges after a reasonably small number of terms.

We make use of the orthonormal nature of the polynomial basis to calculate the coefficients c_i,

$$u_p(\theta) = \int_\Gamma U(Y)\psi_p(Y)dY. \tag{2}$$

With the right choice of polynomials, we can apply quadrature to solve the integral,

$$u_p = \sum_{\ell=0}^{L} w_\ell U(Y_\ell)\psi_p(Y_\ell). \tag{3}$$

In this case we are applying Gaussian quadrature, where an expansion of order L can exactly integrate a polynomial of order $2L-1$. While the order of the polynomial $\psi_p(Y_\ell)$ is p, the equivalent polynomial order of $U(Y_\ell)$ is unknown and must be determined or approximated. If $U(Y)$ is scalar, L need only be $(p+1)/2$; this is the low bound for quadrature order. Coefficient convergence as a function of quadrature order is further explored for some of the cases in this report (see §4.1).

Once the coefficients are calculated, they in combination with the basis polynomials create a reduced-order model that can be sampled like the original function, but ideally at much less computational expense. The measure of success for the PCESC algorithm is its ability to preserve the mean and variance of the original function, as well as produce a virtually identical probability density function (pdf) for the solution quantity of interest, $U(\theta; Y)$. The mean, variance, and pdf are confirmed using brute-force Monte Carlo sampling of the original code.

2 Deterministic Solvers

In this section we identify the four deterministic solvers on which uncertainty propagation will be applied.

2.1 Polynomial Solver

We include this test case because of the analytic solution, mean, and variance. Since we treat a solver as a black box accepting inputs and returning outputs, evaluating a polynomial is a very simple test case. The solver performs one of the two function evaluations, depending on the number of uncertain variables:

$$f(x) = 1 + 2x, \tag{4}$$

$$f(x, y) = x(1 + y). \tag{5}$$

The analytic first and second moments of the functions $f(x)$ and $f(x, y)$ are shown in Table 1.

Function	$\mathbb{E}[f]$	$\mathbb{E}[f^2]$
$f(x)$	$1 + 2\mathbb{E}[x]$	$1 + 4\mathbb{E}[x] + 4\mathbb{E}[x^2]$
$f(x, y)$	$\mathbb{E}[x](1 + \mathbb{E}[x])$	$\mathbb{E}[x^2]\big(1 + 2\mathbb{E}[y] + \mathbb{E}[y^2]\big)$

Table 1: First and Second Moments of Polynomial Solvers

2.2 Source Solver

Like polynomial solvers, this solver evaluates a simple expression; however, there are several potential uncertain terms. Because the solution U varies nonlinearly with some of the terms, a polynomial expansion of finite degree cannot exactly represent the solution. This solver is the analytic solution to an isotropic monoenergetic neutron source spread homogeneously throughout a purely absorbing one-dimensional semi-infinite medium. The governing PDE is

$$-D\frac{d^2\phi}{d^2x} + \Sigma_a\phi = S. \tag{6}$$

The unknown is the scalar flux ϕ. D is the diffusion coefficient, Σ_a is the macroscopic absorbing cross section, and the forcing term S is the homogenous source strength. Boundary conditions include no-traction on the left and TODO on the right. The solution at any particular point x is given by

$$\phi = \frac{S}{\Sigma_a}\left(1 - e^{-x/L}\right), \tag{7}$$

$$L^2 = \frac{D}{\Sigma_a}. \tag{8}$$

We can summarize this solver as

$$\phi = \phi(S, D, x, \Sigma_a), \quad \phi \in \mathbb{R}. \tag{9}$$

2.3 1D Diffusion Solver

This problem is a simple version of a k-eigenvalue criticality problem using 1D, two-energy diffusion for neutron transport. While this problem is 1D, we use a 2D mesh to solve it by imposing reflecting boundary conditions on the top and bottom, with vacuum (no-traction) boundaries on the right and left. We also consider only one homogeneous material for the entire mesh. The governing PDE for this equation is

$$-\frac{d}{dx}D_g\frac{d\phi_g}{dx}+(\Sigma_{g,a}+\Sigma_{g,s})\phi_g=\sum_{g'}\sigma_s^{g'\to g}\phi_{g'}+\frac{\chi_g}{k(\phi)}\sum_{g'}\nu_{g'}\sigma_{f,g'}\phi_{g'},\qquad g\in[1,2], \tag{10}$$

$$\Sigma_{g,a}=\Sigma_{g,c}+\Sigma_{g,f}, \tag{11}$$

The unknown is the two-vector angular flux $\phi=(\phi_1,\phi_2)$ and the quantity of interest is criticality eigenvalue $k(\phi)$. Group index g denotes the energy group; D is the group diffusion cross section; x is the location within the problem; $\Sigma_a,\Sigma_s,\Sigma_f,\Sigma_c$ are the macroscopic absorption, scattering, fission, and capture cross sections respectively; and χ is the fraction of neutrons born into an energy group. In this case, we consider only downscattering, and fission neutrons are only born into the high energy group ($\Sigma_s^{2\to1}=\chi_2=0$).

The input parameters for this solver include all the material properties $(D_g,\Sigma_{g,c},\Sigma_{g,s},\Sigma_{g,f},\nu_g)$, $g=1,2$. The output parameter is k. We can summarize this solver as

$$k=k\big(D_g,\Sigma_{g,c},\Sigma_{g'\to g,s},\Sigma_{g,f},\nu_g\big),\quad g\in(1,2),\quad k\in\mathbb{R}. \tag{12}$$

2.4 Quarter Core Solver

The last solver acts on the PDEs most often used in approximating nuclear reactor behavior. In particular, we consider a two-dimensional k-eigenvalue steady state problem with two energy groups. The domain is square as

$$D=[0,200\text{ cm}]^2, \tag{13}$$

and is shown in Fig. 1. We assign the labels top, bottom, left, and right to corresponding locations in Fig. 1.

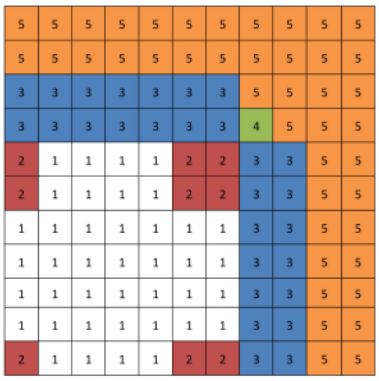


Figure 1: Problem Domain

We consider all fission neutrons to be initialized with fast energies, and disregard up-scattering in energy. Our coupled PDEs are

$$-\boldsymbol{\nabla} D_1^{(R)} \boldsymbol{\nabla} \phi_1(\bar{x}) + (\Sigma_{1,a}^{(R)} + \Sigma_{1\to 2,s}^{(R)})\phi_1(\bar{x}) = \frac{1}{k(\phi)} \sum_{g'=1}^{2} \nu_{g'}^{(R)} \Sigma_{g',f}^{(R)} \phi_{g'}(\bar{x}), \tag{14}$$

$$-\boldsymbol{\nabla} D_2^{(R)} \boldsymbol{\nabla} \phi_2(\bar{x}) + \Sigma_{2,a}^{(R)} \phi_2(\bar{x}) = \Sigma_{1\to 2,s}^{(R)} \phi_1(\bar{x}). \tag{15}$$

We impose reflecting boundary conditions on the left and bottom, and vacuum (or no-traction) boundaries on the top and right, defined by

$$\text{Vacuum:} \quad -D \left.\frac{\partial \phi}{\partial x_1}\right|_{x_1=0} = 0, \tag{16}$$

$$-D \left.\frac{\partial \phi}{\partial x_2}\right|_{x_2=0} = 0, \tag{17}$$

$$\text{Reflective:} \quad \left.\frac{\partial \phi}{\partial x_1}\right|_{x_1=0} = 0, \tag{18}$$

$$\left.\frac{\partial \phi}{\partial x_2}\right|_{x_2=0} = 0. \tag{19}$$

The unknown is the two-vector $\phi = (\phi_1, \phi_2)^T$ and the quantity of interest k is the criticality eigenvalue, given by

$$k(\phi) = \sum_{g=1}^{2} \iint_D \nu \Sigma_f^{(g)} \phi(x_1, x_2) dx dy, \tag{20}$$

Group index $g \in (1, 2)$ denotes the energy group of the property; material index $R(\bar{x}) \in (1, 2, 3, 4, 5)$ is the material of the property; D is the neutron diffusion cross section; $\phi(\bar{x})$ is the scalar neutron flux measured in neutrons per area per time; ν is the average number of neutrons produced per fission; $\Sigma_{g'\to g,s}^{(R)}$ is the macroscopic neutron interaction cross section for neutrons with initial energy in group g' transitioning to energy group g in material R; $\Sigma_{g,f}^{(R)}$ is the macroscopic neutron fission cross section for neutrons in energy group g for material R; and $\Sigma_{g,a}^{(R)}$ is the total neutron absorption interaction cross section for group g in material R, which is further given by the macroscopic capture and fission cross sections as

$$\Sigma_{g,a}^{(R)} \equiv \Sigma_{g,c}^{(R)} + \Sigma_{g,f}^{(R)}, \quad g = 1, 2. \tag{21}$$

We can summarize this solver as

$$k = k\Big(D_g^{(R)}, \Sigma_{g,c}^{(R)}, \Sigma_{g'\to g,s}^{(R)}, \Sigma_{g,f}^{(R)}, \nu_g^{(R)}\Big), \quad g \in (1, 2), R \in (1, 2, 3, 4, 5), \quad k \in \mathbb{R}. \tag{22}$$

3 Univariate UQ

In this section we limit ourselves to the analysis of the UQ algorithm as it applies to univariate versions of the deterministic solvers. This gives a baseline before approaching multivariate inputs in their variety.

3.1 Polynomial Solver: $f(x)$

The purpose of the polynomial solver is to demonstrate convergence on an analytic mean, variance, and pdf for both Monte Carlo and PCESC methods. We consider two cases. In the first, x is distributed uniformly as $x \sim \mathcal{U}(3,7)$. In the second, x is distributed normally as $x \sim \mathcal{N}(5,4)$. The expected analytic moments are given in Table 2.

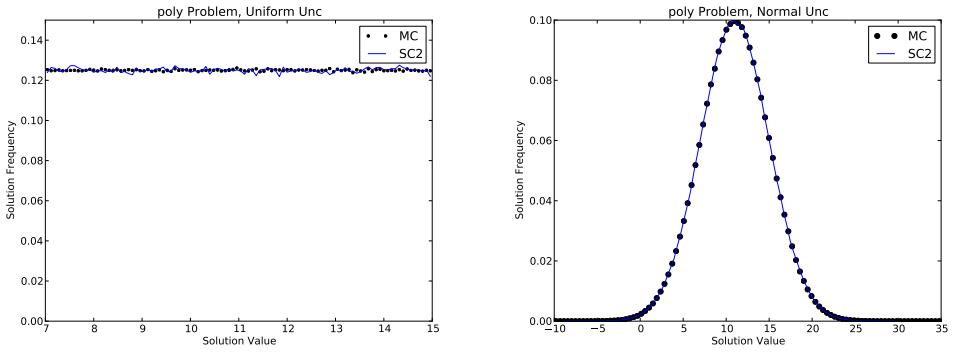
Distribution	m_1	var
$\mathcal{U}(3,7)$	11	79/3
$\mathcal{N}(5,4)$	11	16

Table 2: Analytic Moments for Univariate Polynomial Solver

Because this case expands a first-order polynomial in polynomials, it can be exactly replicated with a first-order expansion. Table 3 shows the Monte Carlo and PCESC statistics. The solution PDFs are in Fig. 2.

Distr.	UQ	runs|order	mean	variance
Uniform	MC	1×10^7	10.999555712	5.3374535046
	SC	2	11.0	5.33333333333
Normal	MC	1×10^7	11.0000448162	15.9892244775
	SC	2	11	16

Table 3: Polynomial Solver, Univariate Statistics



(a) Uniform (b) Normal

Figure 2: Univariate Polynomial Solver PDFs

3.2 Source Solver: $\phi = \phi(S, D, x, \Sigma_a)$

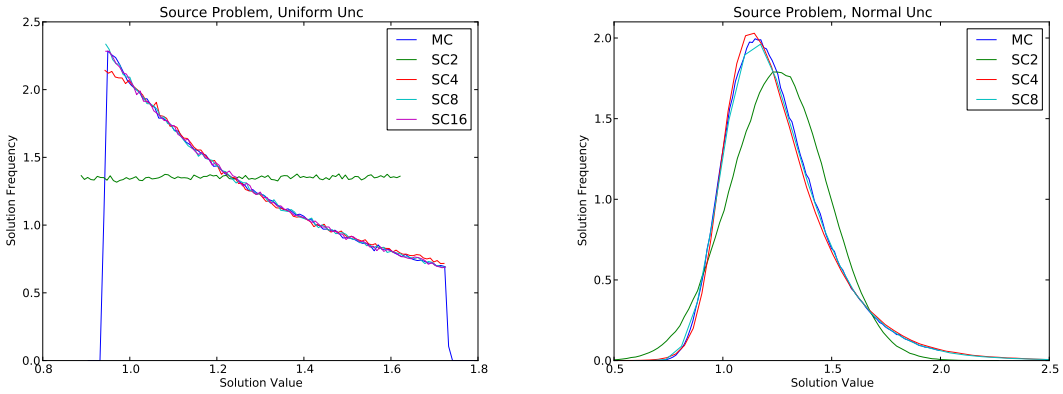
For the univariate case, we introduce uncertainty in the absorption cross section Σ_a. We consider two cases: first, when $\Sigma_a \sim \mathcal{U}(0.5, 1)$; second, when $\Sigma_a \sim \mathcal{N}(0.75, 0.0225)$. The other parameters are

$$S = 1.0 \text{ n/cm}^2\text{/s}, \tag{23}$$
$$D = 0.5 \text{ /cm}, \tag{24}$$
$$x = 2.0 \text{ cm}. \tag{25}$$

We consider several increasing levels of polynomial expansion and the convergence onto the Monte Carlo solution. The results are in Table 4. The solution PDFs are in Fig. 3.

Distr.	UQ	runs|order	mean	variance
Uniform	MC	1×10^6	1.26069628111	0.0632432419713
	SC	2	1.25774207229	0.0495341371244
	SC	4	1.26064320417	0.0604388749588
	SC	8	1.26108375978	0.0637370898233
	SC	16	1.26112339681	0.0639754882641
Normal	MC	1×10^6	1.24922240195	0.0488719424418
	SC	2	1.2547221522	0
	SC	4	1.25569029702	0.049198975952
	SC	8	1.25569096924	0.0492316191443
	SC	16	1.25569096924	0.0492316191611

Table 4: Source Solver, Univariate Statistics



(a) Uniform (b) Normal

Figure 3: Univariate Source Solver PDFs

3.3 1D Diffusion Solver: $k = k(D_g, \Sigma_{g,c}, \Sigma_{g' \to g,s}, \Sigma_{g,f}, \nu_g)$

For the univariate case, we introduce uncertainty in the neutron capture cross section of the second energy group $\Sigma_{2,c}$. We consider three cases, $\Sigma_{2,c}$ distributed normally in each. We choose a different mean for each distribution. Each mean corresponds to a different reactivity state: subcritical, critical, and supercritical ($k = (0.9, 1.0, 1.1)$). The three distributions are respectively $\Sigma_{2,c} \sim (\mathcal{N}(0.055969, 0.1), \mathcal{N}(0.04438, 0.1), \mathcal{N}(0.035181, 0.1))$. We also restrict Monte Carlo sampling of the uncertain parameter to three standard deviations around the mean, effectively creating a truncated normal distribution. For this reason, the mean and variance of the Monte Carlo sampling do not converge on the analytic values. The results are shown in Table 5. The solution PDFs are shown in Fig. 4.

Type	UQ	runs|order	mean	variance
Subcritical	MC	1×10^6	0.907699673282	0.00632790358771
	SC	2	0.907653521565	0.00595095987773
	SC	4	0.907813174929	0.00633526057266
	SC	8	0.907813389468	0.00633649118503
	SC	16	0.907813389471	0.00633649126226
Critical	MC	1×10^6	1.01144778738	0.0105716289269
	SC	2	1.01107577584	0.00974910361412
	SC	4	1.0113755955	0.0105739848126
	SC	8	1.01137628238	0.0105785232446
	SC	16	1.01137628243	0.0105785241745
Supercritical	MC	1×10^6	1.11593980088	0.0168503796482
	SC	2	1.11537873568	0.0151539223359
	SC	4	1.11590698755	0.016795821961
	SC	8	1.11590896034	0.0168106966474
	SC	16	1.11590896088	0.0168107061554

Table 5: 1D Solver, Univariate Statistics

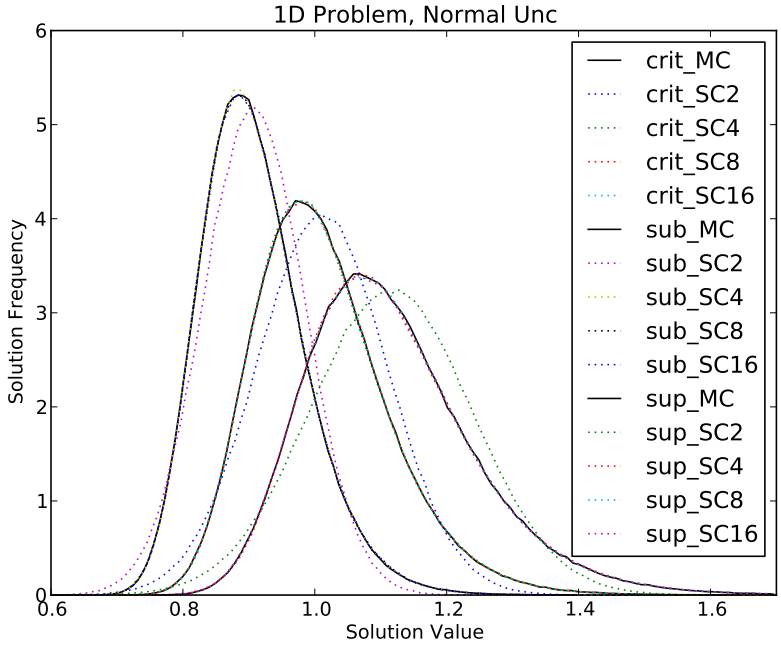


Figure 4: Univariate 1D Solver PDFs

3.4 Quarter Core Solver: $k = k\left(D_g^{(R)}, \Sigma_{g,c}^{(R)}, \Sigma_{g'\to g,s}^{(R)}, \Sigma_{g,f}^{(R)}, \nu_g^{(R)}\right)$

The original benchmark parameters before introducing uncertainty are in Table 6.

5	5	5	5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	5	5	5	5
3	3	3	3	3	3	3	5	5	5	5
3	3	3	3	3	3	3	4	5	5	5
2	1	1	1	1	2	2	3	3	5	5
2	1	1	1	1	2	2	3	3	5	5
1	1	1	1	1	1	1	3	3	5	5
1	1	1	1	1	1	1	3	3	5	5
1	1	1	1	1	1	1	3	3	5	5
1	1	1	1	1	1	1	3	3	5	5
2	1	1	1	1	2	2	3	3	5	5

Figure 5: Quarter Core Map

Material (R)	Group	$D_g^{(R)}$	$\Sigma_{g,c}^{(R)}$	$\nu\Sigma_{g,f}^{(R)}$	$\Sigma_{1\to 2,s}^{(R)}$
1	1	1.255	4.602e-3	4.602e-3	2.533e-2
	2	2.11e-1	5.540e-2	1.091e-1	
2	1	1.268	4.609e-3	4.609e-3	2.767e-2
	2	1.902e-1	8.675e-2	8.675e-2	
3	1	1.259	6.083e-3	4.663e-3	2.617e-2
	2	2.091e-1	4.142e-2	1.021e-1	
4	1	1.259	4.663e-3	4.663e-3	2.617e-2
	2	2.091e-1	3.131e-2	1.021e-1	
5	1	1.257	6.034e-4	0	4.754e-2
	2	1.592e-1	1.911e-2	0	

Table 6: Material Properties for Core

Similar to the one-dimensional problem, we introduce uncertainty in the low-energy capture cross section, in this case only in Material 1 $\left(\Sigma_{2,c}^{(1)}\right)$, which makes up the majority of the core. We consider two distributions, $\Sigma_{2,c}^{(1)} \sim \left(\mathcal{U}(0.0454, 0.0654), \mathcal{N}(0.0554, 0.01^2)\right)$. Because of the nonlinear dependence of the solution on the input parameter, we use order 32 Gauss quadrature to obtain all the polynomial expansion coefficients, instead of scaling them with the total expansion order. The Monte Carlo sampling algorithm rejects samples outside three standard deviations, causing discrepancy in the mean and variance.

Type	UQ	runs|order	mean	variance
Uniform	MC	1×10^6	1.00406413634	0.000446173081079
	SC	2	1.00416405471	0.000375112851817
	SC	4	1.00416405471	0.000390962150246
	SC	8	1.00416405471	0.000406864600682
	SC	16	1.00416405471	0.000421349517322
	SC	32	1.00416405471	0.000425027572716
Normal	MC	1×10^6	1.01333702129	0.00160652595587
	SC	2	1.01643813464	0.00138703446968
	SC	4	1.01643813464	0.00184314998697
	SC	8	1.01643813464	0.00184690058216
	SC	16	1.01643813464	0.00184724103523
	SC	32	1.01643813464	0.00184726152781

Table 7: Quarter Core Solver, Univariate Statistics

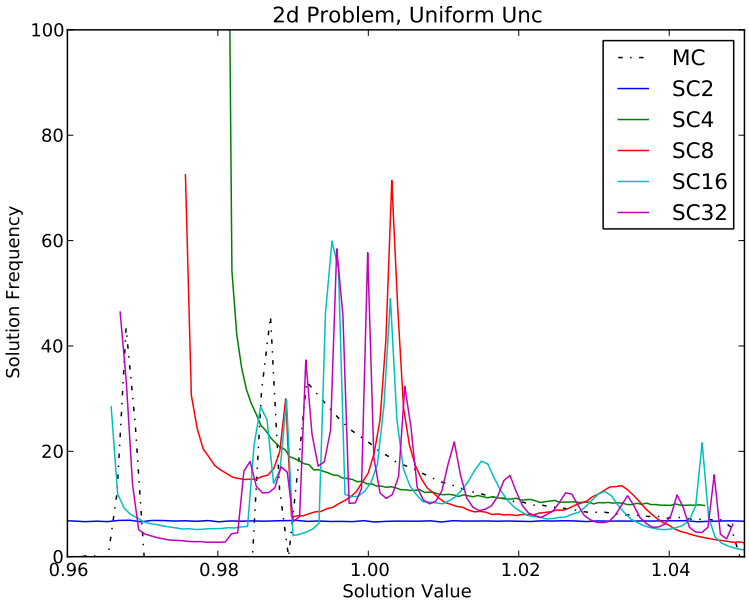


Figure 6: Uniform Univariate Quarter Core Solver PDFs

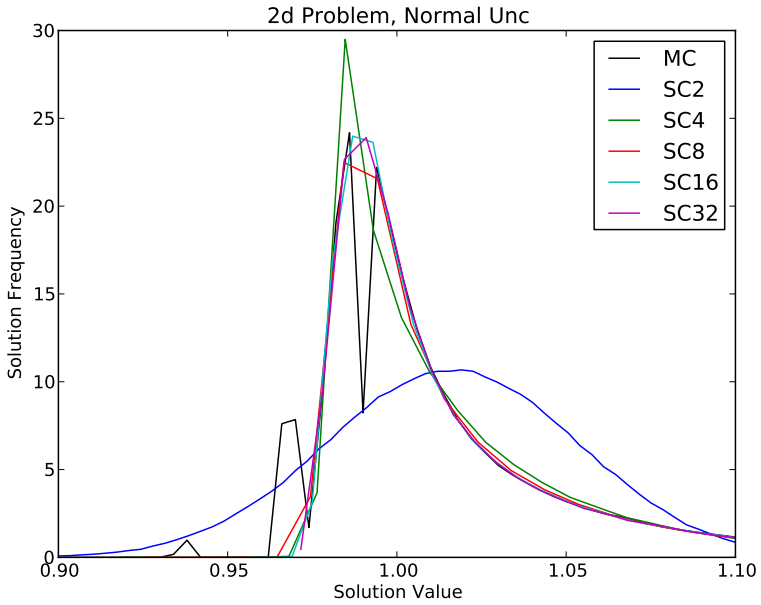


Figure 7: Normal Univariate Quarter Core Solver PDFs

4 Convergence

There are several factors that contribute to the successful convergence of the polynomial chaos expansion sum with coefficients calculated by quadrature. The coefficients are calculated as

$$u_p \approx \sum_{\ell=0}^{L} w_\ell U(Y_\ell)\psi_p(Y_\ell). \tag{26}$$

Two significant sources of error appear in this calculation: first, the quadrature truncation error from using only L terms in the quadrature set; second, the inherent error in the solver $U(Y)$, which we denote spatial discretization error. While this assumes the error in the solver is chiefly from the division of the domain into a mesh of finite refinement (for example, using finite element or finite volume methods), it could similarly apply to other solution method errors. We consider the impact of both quadrature truncation error and spatial discretization error on the convergence of the expansion coefficients.

4.1 Quadrature

One concern with using stochastic collocation to build the polynomial chaos moments is the appropriate order of quadrature to use. With Gaussian quadrature, a sum with n terms can exactly integrate a polynomial of order $2n-1$. The chaos moment expression we integrate is

$$u_p = \int_\Omega U(Y)\rho(Y)\psi_p(Y)dY, \tag{27}$$

$$\approx \sum_{\ell=0}^{L} w_\ell U(Y_\ell)\psi_p(Y_\ell), \tag{28}$$

where ρ is the probability distribution of Y and ψ_p is the $p-th$ order basis polynomial. In the simplest case when $U(Y)$ is a scalar quantity, the order of the expression under the integral is determined solely by the basis polynomial. Thus a quadrature order of $(i+1)/2$ is the minimum quadrature order necessary to integrate the chaos moments. In the case $U(Y)$ is highly nonlinear and ill-approximated by even high-order polynomials, the necessary quadrature order required to accurately determine moments is much higher.

As an example of chaos moment convergence as a function of quadrature order, we show the PCESC moments of the 1D diffusion solver critical case for 7th-order quadrature in Table 8, as determined by increasing orders of quadrature from the minimum to 12-th order, which assumes $U(Y)$ alone is well-approximated by a 7th-order polynomial. As can be seen, the lower moments require smaller quadrature orders, and at least order 8 quadrature is necessary to see reasonable convergence for higher moments. While the mean converges with low-order quadrature, the highest expansion order doesn't converge to discretization error until 10-th order quadrature is used.

Table 9 shows expansion moments calculated using quadrature order equal to expansion order for orders up through 31.

Quadrature Order	u_0	u_1	u_2	u_3	u_4	u_5	u_6	u_7
5	1.34648094	-0.13502447	0.02226791	-0.0045477	0.00000000	0.00101863	-0.00092988	0.00313821
6	1.34648100	-0.13502501	0.02227125	-0.00456451	0.0010908	-0.00027317	0.00000000	0.00025291
8	1.34648101	-0.13502506	0.02227158	-0.00456614	0.0010978	-0.00029964	9.01120742e-05	-2.67010906e-05
10	1.34648101	-0.13502506	0.02227158	-0.00456616	0.0010979	-0.00030000	9.13419949e-05	-3.05173759e-05
12	1.34648101	-0.13502506	0.02227158	-0.00456616	0.0010979	-0.00030001	9.13732306e-05	-3.06142832e-05

Table 8: Chaos Moment Convergence with Increasing Quadrature Order

Moment	SC1	SC3	SC7	SC15	SC31
0	1.34608094	1.3464801	1.34648101	1.34648101	1.34648101
1	-0.13145279	-0.1350169	-0.13502506	-0.13502506	-0.13502506
2	0	0.02222067	0.02227158	0.02227158	0.02227158
3	0	-0.00431037	-0.00456614	-0.00456616	-0.00456616
4	0	0	0.0010978	0.0010979	0.0010979
5	0	0	-0.00029964	-0.00030001	-0.00030001
6	0	0	9.01120742e-05	9.13747084e-05	9.13746830e-05
7	0	0	-2.67010906e-05	-3.06188612e-05	-3.06187650e-05
8	0	0	0	1.11868181e-05	1.11865704e-05
9	0	0	0	-4.42800671e-06	-4.42735312e-06
10	0	0	0	1.89021407e-06	1.88853134e-06
11	0	0	0	-8.66874281e-07	-8.62852707e-07
12	0	0	0	4.24658199e-07	4.15693998e-07
13	0	0	0	-2.18790217e-07	-1.99976426e-07
14	0	0	0	1.12938059e-07	7.56801496e-08
15	0	0	0	-4.90000878e-08	2.05894301e-08
16	0	0	0	0	-1.22580841e-07
17	0	0	0	0	2.51097599e-07
18	0	0	0	0	-4.18278488e-07
19	0	0	0	0	6.25910592e-07
20	0	0	0	0	-8.61060458e-07
21	0	0	0	0	1.09209939e-06
22	0	0	0	0	-1.26866179e-06
23	0	0	0	0	1.32892404e-06
24	0	0	0	0	-1.21597330e-06
25	0	0	0	0	9.01437926e-07
26	0	0	0	0	-4.09433059e-07
27	0	0	0	0	-1.70599526e-07
28	0	0	0	0	6.94034499e-07
29	0	0	0	0	-9.99892578e-07
30	0	0	0	0	9.71463606e-07
31	0	0	0	0	-5.97051529e-07

Table 9: Chaos Moments for 1D Critical Case

4.2 Spatial Discritization

To see the effect of spatial discretization on coefficient convergence, we consider the quarter core benchmark solver. We consider five univariate cases. In each, we vary a different parameter chosen from $\Sigma_{2,c}^{(1)}, \Sigma_{2,f}^{(1)}, \Sigma_{2,c}^{(4)}, \Sigma_{2,f}^{(4)}, D_2^{(5)}$. For each parameter, we use the coarsest mesh (11x11) possible in the quarter-core benchmark and plot the expansion coefficients up to order 256, shown in Fig. 8. While the Material 4 and Material 5 properties converge quickly to acceptable levels, the Material 1 properties fail to converge past a magnitude of 10^{-4}.

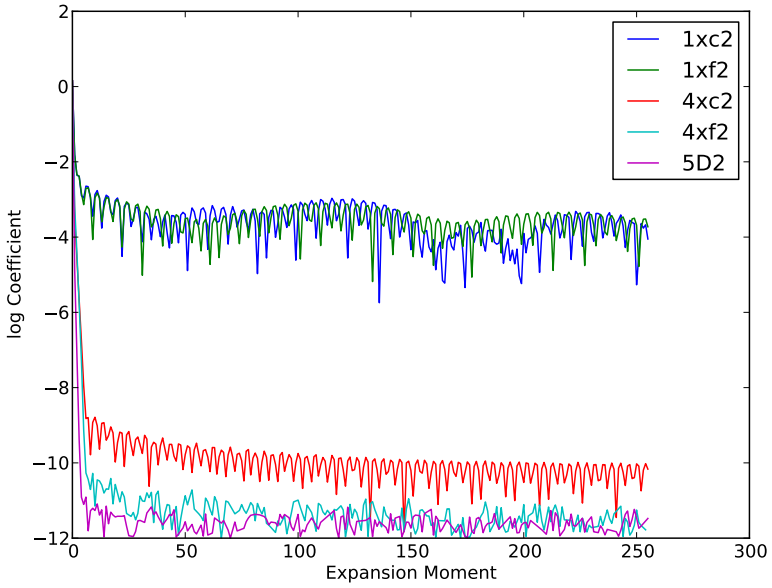


Figure 8: Quarter Core Solver Coarse Mesh Coefficient Decay

To demonstrate the effect of increasing mesh refinement, we consider only the low-energy capture cross section for the first region (1xc2, $\Sigma_{2,c}^{(1)}$). In each successive refinement $n = (1, 2, 3, 4, 5)$ we subdivide each of the 121 regions of the problem into n^2 cells, so that for $n = 2$ the global mesh is 22x22, and so on. The results are shown in Fig. 9. With only one additional refinement level, the coefficients converge to acceptable levels.

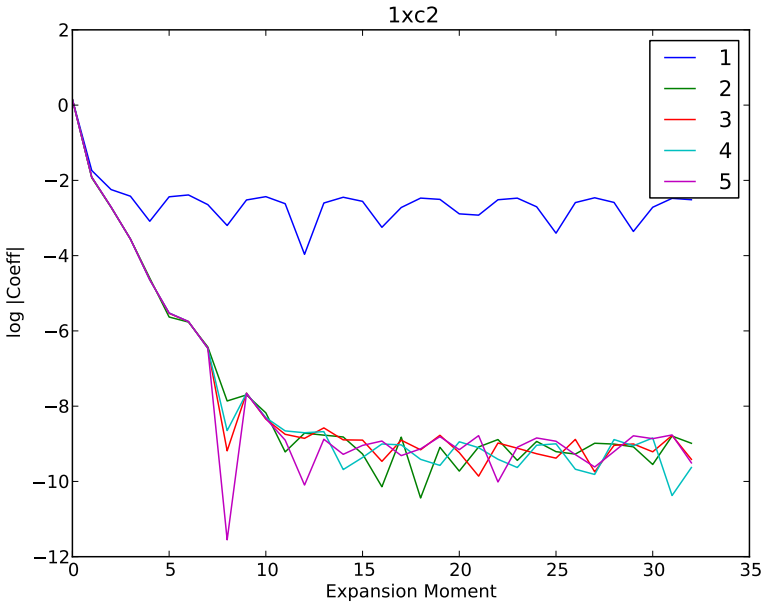


Figure 9: 2G2D: Coefficients over Mesh Refinement

We repeat the original exercise, this time making use of the $n = 2$ refined mesh, and see all coefficients converging to an acceptable level after several terms. The results are in Fig. 10.

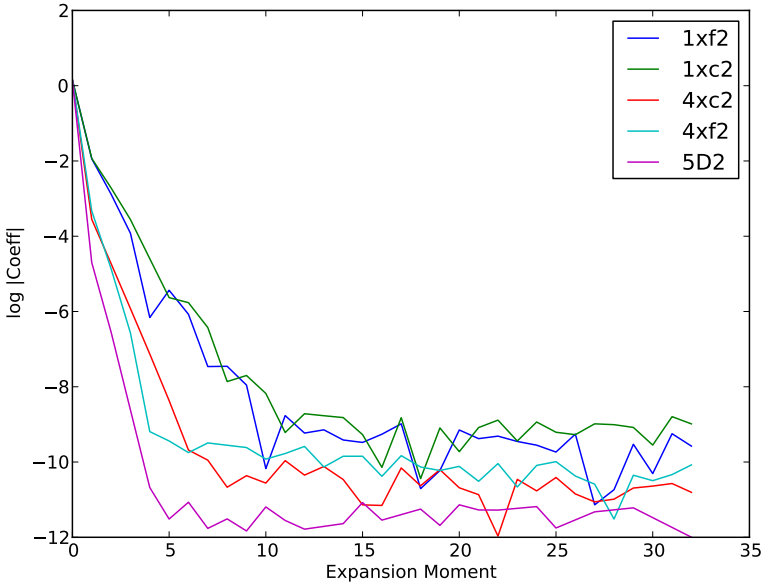


Figure 10: 2G2D: Coarse Mesh Coefficient Decay

5 Multivariate UQ

We now extend the univariate uncertainty quantification to multivariate solvers. For now, we limit ourselves to the tensor product collocation space, which we acknowledge as sufficient but quite inefficient. Improving on this approach is the significant focus for future work. We consider uncertain vector $Y = [Y_1, Y_2, \ldots, Y_N]$ as uncertain input parameters for the solution $U(Y)$. We expand as before, but considering N uncorrelated input parameters:

$$U(\theta; Y) \approx U_P(\theta; Y) = \sum_{p_1=0}^{P_1} \sum_{p_2=0}^{P_2} \cdots \sum_{p_N=0}^{P_N} u_{\bar{p}}(\theta) \prod_{n=1}^{N} \psi_{p_n}(Y), \tag{29}$$

where now multi-index $\bar{p} = [p_1, p_2, \ldots, p_N]$ denotes a single set of expansion orders, one for each variable. The multi-index is taken from the multi-index set $\Lambda(L)$, which includes all desired multi-indices. In the case of tensor product space,

$$\Lambda_{\mathrm{TP}}(L) = \left\{ \bar{p} = (p_1, \ldots, p_N) : \max_{1 \le n \le N} P_n \le L \right\}. \tag{30}$$

This tensor product space quickly becomes unmanageably large, as the size of the index set scales as

$$|\Lambda_{\mathrm{TP}}(L)| = (L+1)^N. \tag{31}$$

However, this index set will serve to demonstrate the UQ algorithm and its capacity to treat multivariate UQ for deterministic solvers.

5.1 Polynomial Solver: $f(x, y)$

We once again include this case for its analytic mean and variance. We introduce uniform uncertainty to both parameters as $x \sim \mathcal{U}(3, 7), y \sim \mathcal{U}(1, 6)$. The analytic expected statistics along with the Monte Carlo and PCESC statistics are shown in Table 10, and the PDFs in Fig. 11

type	runs/order	mean	variance
Analytic	-	45/2	81.8611111111
MC	1×10^7	22.4962950638	81.8754664265
SC	(1,1)	22.5	81.8611111111

Table 10: Polynomial Solver, Bivariate Uniform Distribution Statistics

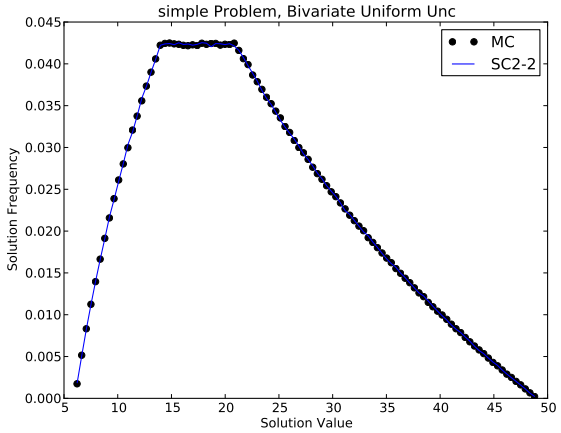


Figure 11: Polynomial Solver, Bivariate Uniform PDFs

5.2 Source Solver: $\phi = \phi(S, D, x, \Sigma_a)$

In addition to the absorption cross section, we introduce uncertainty in the location at which the flux is measured. This occasion might arise when the exact absorption properties of a medium are unknown and a point detector is placed with some uncertainty. We allow the input parameters to be uncertain as $\Sigma_a \sim \mathcal{U}(0.5, 1), x \sim \mathcal{U}(1.5, 2.5)$. The other two parameters remain constant as

$$S = 1.0 \text{ n/cm}^2\text{/s}, \tag{32}$$

$$D = 0.5 \text{ /cm}. \tag{33}$$

The statistical results are in Table 11 and the PDFs in Fig. 12.

type	order(Σ_a, x)	mean	variance
MC	1×10^6	1.24791828682	0.0508287413676
SC	(2,2)	1.24804231569	0.0506451101763
SC	(2,4)	1.24804212351	0.0506466208388
SC	(4,2)	1.24806746049	0.0507934845282
SC	(4,4)	1.24806726831	0.0507949951904

Table 11: Statistics for Source Solver with Bivariate Uniform Uncertainty

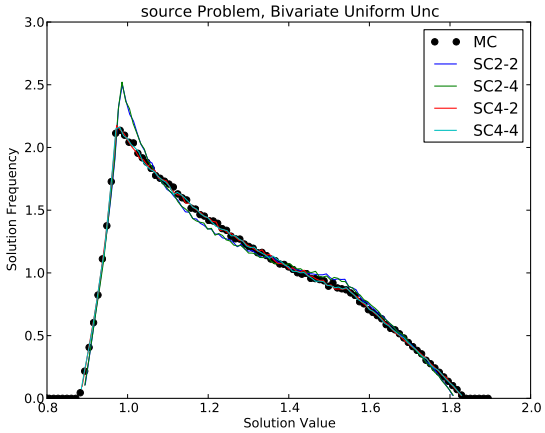


Figure 12: Bivariate Source Solver Solution Distributions

5.3 Quarter Core Solver: $k = k\left(D_g^{(R)}, \Sigma_{g,c}^{(R)}, \Sigma_{g'\to g,s}^{(R)}, \Sigma_{g,f}^{(R)}, \nu_g^{(R)}\right)$

In this case we consider five uncertain parameters simultaneously, as in Table 12. Each was given approximately 10% uncertainty from its mean in the benchmark problem.

Region	Energy Group	Parameter	Uncertainty
1	2	Σ_c	$\mathcal{U}(0.050, 0.061)$
1	2	Σ_f	$\mathcal{U}(0.098, 0.120)$
4	2	Σ_c	$\mathcal{U}(0.037, 0.046)$
4	2	Σ_f	$\mathcal{U}(0.092, 0.112)$
5	2	D	$\mathcal{U}(0.143, 0.175)$

Table 12: Quarter Core Multivariate Uncertainty Space

We also can attempt to make informed guesses as to the appropriate expansion order of each input parameter by performing independent convergence studies for each. Using the results from §4.2, we use expansion orders based on convergence tolerance. The results are in Table 13.

type	tol	$\mathcal{P}(\Sigma_{2,c}^1, \Sigma_{2,f}^1, \Sigma_{2,c}^4, \Sigma_{2,f}^4, D_2^5)$	mean	variance
MC	-	-	0.999064586714	0.0262019588485
SC	1e-4	(5, 4, 3, 3, 1)	1.00191085676	0.000202816510815
SC	1e-5	(6, 5, 4, 3, 1)	1.00112487809	0.000554497293737
SC	1e-6	(8, 7, 5, 4, 2)	1.00018130781	0.00245280240592
SC	1e-8	(10, 9, 6, 4, 3)	1.00014901416	0.00207327774315

Table 13: Statistics for Multivariate Quarter Core Solver

The mean is converged quite swiftly. The variance improves quickly with decreased tolerance,

but there is still an order of magnitude difference between the Monte Carlo-calculated variance and the PCESC-calculated variance. This suggests that considering only independent convergence of parameters is insufficient, and future studies will include consideration for convergence by pairwise input parameters.

6 Index Sets

In the previous section, we use coefficients for the multivariate polynomial expansion obtained from a tensor product space; that is, we consider all possible combinations of expansion orders for all the input parameters. In this section we compare and contrast this *tensor product* (TP) index set with the *total degree* (TD) and *hyperbolic cross* (HC). We use the following notation:

$$\bar{p} = (p_1, p_2, \ldots, p_N) : \text{multi-index for expansion order of each uncertain input parameter,} \tag{34}$$

$$\psi_{\bar{p}} = \prod_{n=1}^{N} \psi_{p_n}(Y_n) : \text{product of expansion polynomials for each uncertain input parameter,} \tag{35}$$

$$\Lambda_{\mathrm{T}}(L) : \text{full index set of type T with maximum index } L \text{ in each dimension,} \tag{36}$$

$$\mathbb{P}_{\Lambda_T(L)} : \text{polynomial space spanned by polynomial products } \{\psi_{\bar{p}}\}, \bar{p} \in \Lambda_T(L). \tag{37}$$

For clarity, we consider a particular example where $\bar{Y} = (Y_1, Y_2) \in [-1, 1]^2$ with polynomial expansion truncated at order $L = 6$ in both dimensions.

6.1 Tensor Product TP

The tensor product space considers all possible combinations of expansion order in all dimensions. The index set is described as

$$\Lambda_{\mathrm{TP}}(L) = \left\{ \bar{p} = (p_1, p_2, \ldots, p_N) : \max_{1 \le n \le N} P_n \le L \right\}, \tag{38}$$

and the polynomial space is

$$\mathbb{P}_{\Lambda_{\mathrm{TP}}(L)}(\Gamma) = \mathbb{P}_{\Lambda(L)}(\Gamma_1) \otimes \cdots \otimes \mathbb{P}_{\Lambda(L)}(\Gamma_N). \tag{39}$$

In our example, this includes all $\bar{p} = (p_1, p_2)$,

(0,0)	(0,1)	(0,2)	(0,3)	(0,4)	(0,5)	(0,6)
(1,0)	(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(0,6)
(2,0)	(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	(0,6)
(3,0)	(3,1)	(3,2)	(3,3)	(3,4)	(3,5)	(0,6) ,
(4,0)	(4,1)	(4,2)	(4,3)	(4,4)	(4,5)	(0,6)
(5,0)	(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(0,6)
(6,0)	(6,1)	(6,2)	(6,3)	(6,4)	(6,5)	(6,6)

for a total of 49 combined expansion coefficients. The size of this index set scales exponentially with dimension as

$$|\Lambda_{\mathrm{TP}}(L)| = (L+1)^N, \tag{40}$$

and becomes very large as order and dimension are increased. While the effectiveness of the method depends on the regularity of $U(\bar{Y})$, for $U(\bar{Y})$ with k derivatives TODO(have to explain this hilbert space), the error goes as

$$||U(\bar{Y}) - U_{\mathrm{TP}(L)}|| \sim M^{-k/N}, \tag{41}$$

where the M is the number of coefficients or the size of the input space,

$$M = |\Lambda_{\mathrm{TP}}| = (L+1)^N. \tag{42}$$

The tensor product index set suffers from the curse of dimensionality, where convergence will suffer with dimensionality.

6.2 Total Degree

7 Conclusions

In conclusion, we have shown that the PCESC uncertainty quantification algorithm shows good agreement with both analytic and Monte Carlo results for both univariate and multivariate uncertainty spaces. The algorithm handles deterministic solvers in a "black box" sense, meaning it is agnostic of the complexity of the solver itself, as shown by four solvers ranging from simple polynomial evaluation to nonlinear multigroup diffusion. In addition, we have shown that convergence for the PCESC method requires careful consideration of both quadrature order for each input parameter as well as the spatial discretization error of the deterministic solver.

We now turn our attention to the applicability of this algorithm. Because of the exponential growth in expense for multivariate uncertainty spaces using the tensor product polynomial set, the efficiency of PCESC is lost when compared to Monte Carlo with the inclusion of many uncertain variables or a high degree polynomial expansion.

There are, however, many more efficient methods for constructing the uncertainty space than using a full tensor product system. We will explore sparse grid quadrature and analysis-of-variance style methods, as well as HDMR methods to reduce the necessary sample space and recovery the efficiency of the PCESC method.