Deterministic Uncertainty Quantification with Raven

Paul Talbot

talbotp@unm.edu

May 29, 2013

1

1 Introduction

RAVEN (Reactor Analysis and Virtual control ENvironment) is a software framework that acts as

the control logic driver for thermal hydraulic code RELAP-7. Central to the purpose of RELAP-7

is determining safety margins in accident-type scenarios for light water nuclear reactors.

Because the inputs to RELAP-7 are likely to have some level of uncertainty in them, RAVEN

allows for the capability to use brute-force Monte Carlo to quantify output uncertainty in terms of

input uncertainty. This makes it valuable as a PRA (probability risk assessment) code, and allows

users to more clearly understand margins calculated with RELAP.

We propose a method of stochastic collocation methods along with generalized polynomial

chaos to sample from the uncertainty space of input variables RELAP in an intelligent way and

propagate those uncertainties through the code, leveraging RAVEN's interface. This avoids the

need to introduce stochastic noise from Monte Carlo calculations and, for a limited number of

uncertain inputs, o  
ers signicant speedup over brute force Monte Carlo for the same degree of

precision. Stochastic collocation may be slower than Monte Carlo as the number of uncertain

variables grows, but much of this loss can be gained by employing sparse grid methods to reduce

the number of necessary samples. The accuracy cost in stochastic collocation and generalized

polynomial chaos originates in truncating innite sums to a small number of terms; the accuracy

of the method generally increases with increasing terms.

We intend eventually to extend the uncertainty quantification tools in RAVEN to propagate

uncertainty from inputs of one code through other coupled mutliphysics models in the MOOSE

(multiphysics object-oriented simulation environment) system. Of particular interest is BISON, a

fuels performance code, which could in turn provide inputs for RAVEN.

2 gPC: Generalized Polynomial Chaos

In general, stochastic processes can be represented eciently by a basis consisting of an orthogonal

set of polynomials, especially if chosen correctly. While homogeneous chaos only makes use of Hermite

polynomials, a more generalized polynomial chaos (gPC) intelligently selects basis polynomials

based on weighting functions.

Polynomial Random Distr. Weighting Span

Legendre Uniform 1/2 [-1,1]

Hermite Normal exp(􀀀x2)=

p

2 (􀀀1;1)

Laguerre Gamma xk􀀀1 exp(􀀀x)=􀀀(k) [0;1)

Consider an uncertain (and therefore treated as stochastic) process U(p; ) that is a function of its

\certain" input parameters and phase space p as well as uncertain parameters . In general, may

be the combination of many (1; 2; :::; n; :::; N) if U depends on many uncertain parameters. We

wish to expand U in terms of one of the polynomial bases in order to quantify its uncertainty. The

polynomial basis is chosen based on the form and span of the uncertainty, as shown in the table.

For any case, U is expanded as

U(p; )

XI

i=1

ciBi(); (1)

where the approximation is because of term truncation at Pt < 1, ci are polynomial coecients,

and Bi is the polynomial of order i that best ts the uncertainty in U. Since the polynomials are

known, we can solve for the unknown coecients using the orthogonality of the basis polynomials

as

ci =

(U(p; );Bi())

(Bi()2)

; (2)

using () as inner product notation

􀀀

f(x); g(x)

Z

S

f(x)g(x)dx; (3)

where S is the support of x.

3 SCM: Stochastic Collocation Method

The stochastic collocation method (SCM) makes use of quadrature sets to sample from the random

space generated by uncertainty. We can make use of quadrature sets consisting of roots of the same

polynomials used as basis functions in order to calculate the inner product for the gPC coefficients,

(U;Bi)

Z

U()Bi()d; (4)

=

Z

d1

Z

d2:::

Z

dN

U(p; 1; 2; :::; N)Bi(1; 2; :::; N); (5)

XM1

m1=1

wm1

XM2

m2=1

wm2 :::

XMN

mN=1

wmN

!

U(p; m1 ; m2 ; :::; mN )Bi(m1 ; m2 ; :::; mN ); (6)

3

where wmn are weights obtained from quadrature sets corresponding to the polynomial basis chosen.

The quadrature set may or may not have the same level of truncation as the polynomial expansion;

that is, Mn need not be the same as M1 or I.

We can further modify the inner product calculation by finding the coefficient term at node

m (1;m1 ; :::; n;mn; :::; N;mN ) so that

ci =

XM1

m1=1

wm1

XM2

m2=1

wm2 :::

XMN

mN=1

wmN

!

ci(1;m1 ; :::; n;mn; :::; N;mN ); (7)

=

XM1

m1=1

wm1

XM2

m2=1

wm2 :::

XMN

mN=1

wmN

!

ci;m; (8)

ci;m U(p; m)Bi(m); (9)

where ci;m is the coefficient to the i-th order basis polynomial corresponding to a single sample

realization m of U(p; ). Furthermore, we bring weights inward and multiply them to obtain weights

that also correspond to a single realization m of U, so that

wm =

NY

h=1

wmh; (10)

ci =

XM1

m1=1

:::

XMn

mn=1

:::

XMN

mN=1

!

wmci;m: (11)

3.1 Constructing Multidimensional Bases

We now give examples of expanding a multivariate function in multiple bases. In future sections

we explore alternate distributions and polynomials, as well as mapping uncertain spaces onto the

[0,1] normalized shifted Legendre polynomial space; for now, we assume all random variables n

are already expressed as uncertain variables with values 2 [0; 1].

3.1.1 Polynomials and Distributions

We consider a set of eight typical uncertainty distributions and their corresponding polynomials and

quadrature. We summarize them in the table below, taken from TODO CITE Xiu and Karniadakis.

Unc. Distribution Basis Polynomials Support

Continuous Normal Hermite (􀀀1;1)

Gamma Laguerre [0;1)

Beta Jacobi [a; b]

Uniform Legendre [a; b]

Discrete Poisson Charlier f0,1,2,...g

Binomial Krawtchouk f0,1,...,Ng

Negative Binomial Meixner f0,1,2,...g

Hypergeometric Hahn f0,1,...,Ng

FIXME get rid of the discontinuous ones? They're not in scipy.

Definitions and examples of these distributions are included in the appendix.

4

3.1.2 Example: Single-Dimensional expansion

Starting with the simplest case, we consider a function of single uniform-uncertainty variable = 1,

f() = a + b; 2 [0; 1]; (12)

where a and b are arbitrary scalars. We expand f() in normalized shifted Legendre polynomials,

f() =

1X

i=0

fi ~ Pi(); (13)

=

X1

i=0

fi ~ Pi(): (14)

We can truncate the sum at 1 term because we know a priori f() is order 1 in , so it can be exactly

represented by Legendre polynomials of up to order 1; in general, this is not known and perfect

accuracy can only be guaranteed with innite terms. Using the orthogonality of the normalized

shifted Legendre polynomials, we find the coefficients fi given by

fi =

Z 1

0

f() ~ Pi()d: (15)

We can approximate the integral with shifted Gauss-Legendre quadrature,

fi =

1X

`=0

w`f(`) ~ Pi(`); (16)

=

X1

`=0

w`f(`) ~ Pi(`); (17)

where once again, because we know the shifted Legendre polynomial order is no greater than 1 and

f() is order 1, the integral has maximum order 3 and shifted Legendre quadrature can exactly

integrate polynomials of order 2n 􀀀 1. It is straightforward to insert the values from the shifted

Legendre quadrature set and see that the coefficients obtained are

f0 = a +

b

2

; f1 =

b

p

3

6

; fi>1 = 0: (18)

If we reconstruct f() using these coefficients and the first three normalized Legendre polynomials,

we obtain our original function a + b.

3.1.3 Example: Multivariate Expansion

We now consider multidimensional function of 1; 2,

f() f(1; 2) = (a 􀀀 b1)(c 􀀀 d2); (19)

where (a; b; c; d) are arbitrary scalars. We expand each dimension in normalized shifted Legendre

polynomials,

f() =

1X

i1=0

1X

i2=0

fi1;i2

~ Pi1(1) ~ Pi2(2); (20)

5

where fi1;i2 is the combined coefficient for the multivariate polynomial term. The coefficients can

be obtained in the same manner as the single dimension expansion,

fi1;i2 =

Z 1

0

Z 1

0

f() ~ Pi1(1) ~ Pi2(2)d1d2; (21)

and approximated with Legendre quadrature

fi1;i2 =

1X

`1=0

w`1

1X

`2=0

w`2f(1;`1 ; 1;`2) ~ Pi1(1;`1) ~ Pi2(2;`2); (22)

=

1X

`1=0

1X

`2=0

w`1w`2f(1;`1 ; 1;`2) ~ Pi1(1;`1) ~ Pi2(2;`2): (23)

Using the first two terms from each sum, we obtain the coefficients

f0;0 =

(2a + b)(2c + d)

4

; (24)

f0;1 =

d

p

3

12

(2a + b); (25)

f1;0 =

b

p

3

12

(2c + d) (26)

f1;1 =

bd

12

; (27)

f(x; y) = f0;0 ~ P0(1) ~ P0(2) + f0;1 ~ P0(1) ~ P1(2) + f1;0 ~ P1(1) ~ P0(2) + f1;1 ~ P1(1) ~ P1(2); (28)

= (a + b1)(c + d2): (29)

3.1.4 General Multivariate Expansion

From the two examples above, it is straightforward to extrapolate the general formulation for an expansion

in an unknown number of dimensions. We consider a function of (1; 2; ; n; ; N)

f() f(1; ; n; ; N): (30)

We expand it in N dimensions in normalized shifted Legendre polynomials,

f() =

1X

i1

1X

i2

1X

iN

fi1;i2; ;iN

NY

n=1

~ Pin(n); (31)

=

1X

i1

1X

iN

fi

NY

n=1

~ Pin(n); (32)

where for simplicity we have defined fi as the coefficient for the full set of polynomials at a particular

set in the sum i = (i1; ; iN). As before, the coefficients fi are determined using orthogonality,

fi =

Z 1

􀀀1

Z 1

􀀀1

"

f()

NY

n=1

~ Pin(n)

#

d1 dN; (33)

6

which is approximated with Legendre quadrature as

fi =

1X

`1=0

1X

`N=0

NY

n=1

w`n

!

f(`)

NY

n=1

~ Pin(n;`n); (34)

=

1X

`1=0

1X

`N=0

NY

n=1

w`n

~ Pin(n;`n)

!

f(`); (35)

where for convenience we define

f(`) f(1;`1 ; ; n;`n; N;`N ): (36)

In practice, it is computationally e  
effctive to store a tensor of coefficients fi for each abscissa

of each quadrature. This coefficient tensor has dimensionality equal to the number of uncertain

parameters N, and each dimension has length equal to the number of quadrature abscissa used for

that uncertain parameter. In this case, for a three-variable function, coeff[i,j,k] corresponds to

fi;j;k.

3.2 Alternative Uncertainties

3.2.1 Arbitrary Uncertainties

While the probability distributions and polynomial chaos above are useful in describing uncertainties,

there exist many other possible uncertainty distributions. However, it is possible to project

arbitrary uncertainties into uniform [0,1] space and treat them with shifted Legendre polynomials.

In fact, we require only the percent point (or percentile) distribution of an uncertain variable to

create a mapping between its natural domain and the [0,1] domain. The drawback to this method

is that shifted Legendre polynomials may not efficiently describe the distribution, and many terms

may be necessary to develop an accurate representation.

We follow here the pattern outlined by TODO CITE Xiu and Kerniadakis. Consider an uncertain

parameter with arbitrary probability distribution function f(). We can expand this

parameter in basis polynomials that describe the desired [0,1] space; namely, (normalized) shifted

Legendre polynomials ~ Pi,

=

1X

i=0

i ~ Pi: (37)

As before, we find the coefficients using the orthogonality of, and the inner product in the Hilbert

space spanned by, the polynomial basis,

i =

Z

S

Pi()g()d; (38)

where g() is the uniform probability distribution of 2 [0; 1]. We note that Eq. (38) is mathematically

nonsensical, in that we assume to be dependent on and their supports are not guaranteed

to be the same; that is, they are likely to belong to diff  
erent probability spaces - if not, then there

is no need to perform the mapping. To correlate the two, we introduce a new uncertain variable

u 2 [0; 1]. Recalling the probability distribution functions f() and g(), we transform probability

space to show

du = f()d = dF(); du = g() = dG(); (39)

7

where F;G are the cumulative distribution function (cdf)'s for f; g,

F() =

Z

􀀀1

f(s)ds; G() =

Z

􀀀1

g(s)ds: (40)

We require both and to be mapped to the domain of u, and show

= F􀀀1(u); = G􀀀1(u); (41)

where F􀀀1;G􀀀1 are the inverse of the cdf, or percent point function (ppf). Using these transformations,

we return to the expansion of and write

=

1X

i=0

iPi; (42)

i =

Z 1

0

F􀀀1(u)Pi

􀀀

G􀀀1(u)

du; (43)

=

1X

n=0

wnF􀀀1(un)Pi

􀀀

G􀀀1(un)

; (44)

where we have applied shifted Gauss-Legendre quadrature to evaluate the integral. We note that

the only requirement for mapping any arbitrary uncertainty onto a common space is the ability to

evaluate the ppf of an uncertainty distribution at quadrature points (un). Also, this procedure is

general for any pdf g() to map onto the domain of ; for our purposes, 2 [0; 1] is the most

beneficial.

8

A Polynomials and Distributions

For reference we include the polynomial, distribution, and quadrature definitions for the continuous

distributions used in this document. To describe polynomials, we make use of the Pachhammer

symbol (a)n

(a)n = a(a + 1)(a + 2):::(a + n 􀀀 1); n = 1; 2; 3; ::: (45)

with (a)n = 1. The generalized hypergeometric series rFs is given by

rFs(a1; :::; ar; b1; :::; bs; z) =

1X

k=0

(a1)k (ar)k

(b1)k (bs)k

zk

k!

: (46)

0.0 4 2 0 2 4

0.1

0.2

0.3

0.4

0.5

f(x)

Normal

(0,1)

(1,1)

(0,2)

0.000 2 4 6 8 10

0.05

0.10

0.15

0.20

0.25

Gamma

(3)

(4)

(5)

4 3 2 1 0 1 2 3 4

x

0.0

0.2

0.4

0.6

0.8

1.0

f(x)

Uniform

(-2,1)

(-1,2)

(-3,5)

0.0 0.2 0.4 0.6 0.8 1.0

x

0.0

0.5

1.0

1.5

2.0

2.5 Beta

(2,2)

(3,2)

(2,5)

Figure 1: Several Distributions

A.1 Standard Distributions

There are several standard distributions for which quadratures with corresponding polynomials are

well-known, making them efficiently represented with small quadratures. We present four here:

normal, Gamma, uniform, and Beta.

9

A.1.1 Normal and Hermite Hen

The normal or Gaussian distribution has support from 􀀀1 to 1 and is characterized by Hermite

polynomials, with the associated Gauss-Hermite quadrature. The pdf of the normal distribution

has the form

(x; ; 2) =

1

p

22

exp

􀀀

(x 􀀀 )2

22

; x 2 (􀀀1;1); (47)

where ; 2 are the mean and variance respectively. Two di  
fferent kinds of Hermite polynomials

exist: one the \probabilist" Hermite polynomial Hen(x), and the more often seen \physicist"

Hermite polynomial Hn(x). The two are essentially the same with the important exception

Hn(x=

p

2) = Hen(x).

Hen = (􀀀1)nex2=2 dn

dxn e􀀀x2=2; (48)

Hn = (􀀀1)nex2 dn

dxn e􀀀x2

: (49)

We make use of the probabilist here because of its conformity with the Gaussian distribution. The

Hermites are orthogonal,

Z 1

􀀀1

Hem(x)Hen(x)e􀀀x2=2dx =

p

2n!nm: (50)

Hermite quadrature integrates exactly functions of the kind

Z 1

􀀀1

f(x)e􀀀x2=2dx =

XL

`=0

w`f(x`): (51)

The abscissas of the quadrature are given by roots of the Hen polynomial and weights are given by

w` =

L!

p

2

n2[Hen􀀀1(x`)]2 : (52)

A normal distribution is shown with = 0; 2 = 1 in Fig. 1.

10

A.1.2 Gamma and Laguerre L  
n

The Gamma distribution has support from 0 to 1 and is characterized by Laguerre polynomials

with the associated Gauss-Laguerre quadrature. The pdf of the Gamma distribution has the form

(x;   
; ) =

x  
e􀀀x=

+1􀀀(  
 + 1)

;   
 > 􀀀1; > 0; x 2 (0;1); (53)

􀀀(  
)

Z 1

0

t  
e􀀀t dt

t

; (54)

􀀀(  
 + 1) =   
􀀀(  
); (55)

where   
; are shape and scale constants, respectively. The (generalized) Laguerre polynomials

L(  
)

n are the solutions to the second order PDE

xy00 + (  
 + 1 􀀀 x)y0 + ny = 0; (56)

and are given by

L(  
)

n (x) =

x􀀀  
ex

n!

dn

dxn

􀀀

e􀀀xxn+

; (57)

=

(  
 + 1)n

n! 1F1(􀀀n;   
 + 1; x); (58)

Z 1

0

exx  
L(  
)

m (x)L(  
)

n (x)dx =

􀀀(n +   
 + 1)

n!

mn;   
 > 􀀀1: (59)

General Laguerre quadrature exactly integrates functions of the kind

Z 1

0

f(x)e􀀀xx  
dx =

XN

`=0

w(  
)

` f(x(  
)

` ): (60)

The abscissas of the quadrature are the roots of the polynomial L(  
)

n , and the weights are given by

w(  
)

` =

1

x(  
)

`

d

dx

L(  
)

N (x(  
)

` )

􀀀1

: (61)

A Gamma distribution with shape   
 = 3 and scale = 1 is shown in Fig. 1.

11

A.1.3 Uniform and Legendre Pn

The uniform distribution has support from a to b, but is typically defined over the domain [-1,1],

and is characterized by Legendre polynomials with the associated Gauss-Legendre quadrature. The

pdf of the uniform distribution is at between a and b and zero everywhere else,

(x; a; b) =

1

b 􀀀 a

; x 2 [a; b]; (62)

where a; b are the maximum and minimum value, respectively. The Legendre polynomials Pn(x)

are solutions to the PDF

d

dx

(1 􀀀 x2)

d

dx

Pn(x)

+ n(n + 1)Pn(x) = 0; (63)

and are given by

Pn(x) =

1

2nn!

dn

dxn

(x2 􀀀 1)2

; (64)

Z 1

􀀀1

Pm(x)Pn(x)dx =

2

2n + 1

mn: (65)

It should be noted that shifting Pn(x); x 2 [􀀀1; 1] to Pn(z); z 2 [a; b] is performed by the transformation

Pn(z) =

b 􀀀 a

2

Pn

b 􀀀 a

2

x +

a + b

2

; x 2 [􀀀1; 1]; z 2 [a; b]: (66)

Legendre quadrature exactly integrates functions of the kind

Z 1

􀀀1

f(x)dx =

XL

`=0

w`f(x`): (67)

The abscissas of the quadrature are the roots of the polynomial Pn, and the weights are given by

w` =

2

(1 􀀀 x2`)

d

dxPn(x`)

2 : (68)

A uniform distribution with minimum -1 and double-range 2 is shown in Fig. 1.

12

A.1.4 Beta and Jacobi P(  
;)

n

The Beta distribution has the same support as the uniform distribution, a to b, but is often defined

over the domain [0,1], and is characterized by Jacobi polynomials with associated Jacobi quadrature.

The Legendre polynomials are a particular type of the Jacobi polynomials with   
 = = 0. The

pdf of the beta distribution is given by

(x;   
; ) =

x  
􀀀1(1 􀀀 x)􀀀1

B(  
; )

; x 2 [0; 1]; (69)

B(  
; ) =

Z 1

0

t  
􀀀1(1 􀀀 t)􀀀1dt; (70)

where   
; are shape parameters. The Jacobi polynomials are given by

P(  
;)

n (x) =

(􀀀1)n

2nn!

(1 􀀀 x)􀀀  
(1 + x)􀀀 dn

dxn

h

(1 􀀀 x)􀀀  
(1 + x)(1 􀀀 x2)n

i

; (71)

=

(  
 + 1)n

n! 2F1

􀀀n; 1 +   
 + + n;   
 + 1;

1 􀀀 x

2

; (72)

Z 1

􀀀1

(1 􀀀 x)  
(1 + x)P(  
;)

m (x)P(  
;)

n (x)dx =

2  
++1

2n +   
 + + 1

􀀀(n +   
 + 1)􀀀(n + + 1)

􀀀(n +   
 + + 1)n!

mn: (73)

Jacobi quadrature exactly integrates functions of the kind

Z 1

􀀀1

f(x)(1 􀀀 x)  
(1 + x)dx =

XL

`=0

w`f(x`): (74)

The abscissas of the quadrature are the roots of the polynomial P(  
;)

n , and the weights are given

by

w` = 􀀀

(2n +   
 + + 2)

(n +   
 + + 1)

􀀀(n +   
 + 1)􀀀(n + + 1)

􀀀(n +   
 + + 1)(n + 1)!

2  
+

Pn+1(x`) d

dxPn(x`)

: (75)

A beta distribution with   
 = 2; = 2 is shown in Fig. 1.

13

A.2 Non-Standard Distributions

There are many other distributions commonly used in uncertainty, but without a convenient set of

polynomials and quadrature to t them. Because of the widespread use of these distributions, we

present some here with approaches to representation by quadrature and polynomials.

0.00 1 2 3 4 5 6

0.2

0.4

0.6

0.8

1.0

f(x)

Exponential

(1)

(2)

(3)

0.000 2 4 6 8 10

0.05

0.10

0.15

0.20

0.25

0.30

Log Normal

(1,2)

(2,2)

(1,5)

0 1 2 3 4 5

x

0.0

0.2

0.4

0.6

0.8

1.0 Triangular

(1/2,1,4)

(2/3,0,4)

(1/3,0,3)

0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0

x

0.0

0.2

0.4

0.6

0.8

1.0

1.2

f(x)

Weibull

(1)

(3/2)

(3)

Figure 2: Alternate Distributions

A.3 Exponential

The exponential distribution ranges from 0 to 1 and has the form

(x;   
 =   
e􀀀  
x; x 2 [0;1); (76)

where   
 is a rate scaling factor. TODO finish.

A.3.1 Lognormal

The log normal is descriptively the log of the normal distribution. It ranges from 0 to 1 and has

the form

(x; ; 2) =

1

x

p

22

exp

􀀀

(ln x 􀀀 )2

22

; x 2 (0;1): (77)

14

where ; 2 are the mean and variance, respectively. TODO finish.

A.3.2 Triangular

The triangular distribution ranges from a to b and rises linearly from a to a point, after which it

falls linearly to b. The pdf is given by

(x; a; b; c) =

8>>>><

>>>>:

0; x < a;

2(x􀀀a)

(b􀀀a)(c􀀀a) ; a x c;

2(b􀀀x)

(b􀀀a)(b􀀀c) ; c < x b;

0; b < x;

(78)

where a; b; c are the minimum, maximum, and location of the highest point, respectively. TODO

finish.

A.3.3 Weibull

The Weibull distribution ranges from 0 to 1 and has the form

(x; ; k) =

k

x

k􀀀1

e􀀀(x=)k

; (79)

where ; k are the scale and shape parameters, respectively. Often, = 1 and k is the only shaping

parameter. TODO finish.

A.3.4 Arbitrary

Many other distributions may arise in characterizing the uncertainty of input parameters. In the

event none of the above distributions are close enough, using the distribution's ppf to represent it

using shifted Legendre polynomials is recommended, with care for the number of terms used.

15