

Introduction to Sampler

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

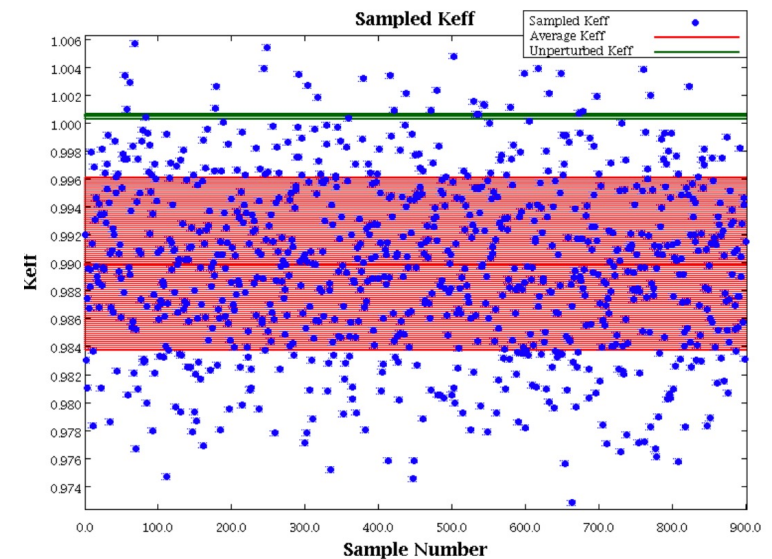
Learning goals

- Understand the stochastic sampling approach to uncertainty quantification in Sampler
- Understand the data sampling capabilities available in SCALE 6.3.1
- Understand how to generate a Sampler data sampling input from a working CSAS input
- Be able to process the results of calculations to compare with TSUNAMI-generated estimates of data-induced k_{eff} uncertainties

Sampler: Statistical uncertainty analysis with SCALE

Calculates uncertainty in *any* result of a SCALE computation due to two basic types of uncertainty.

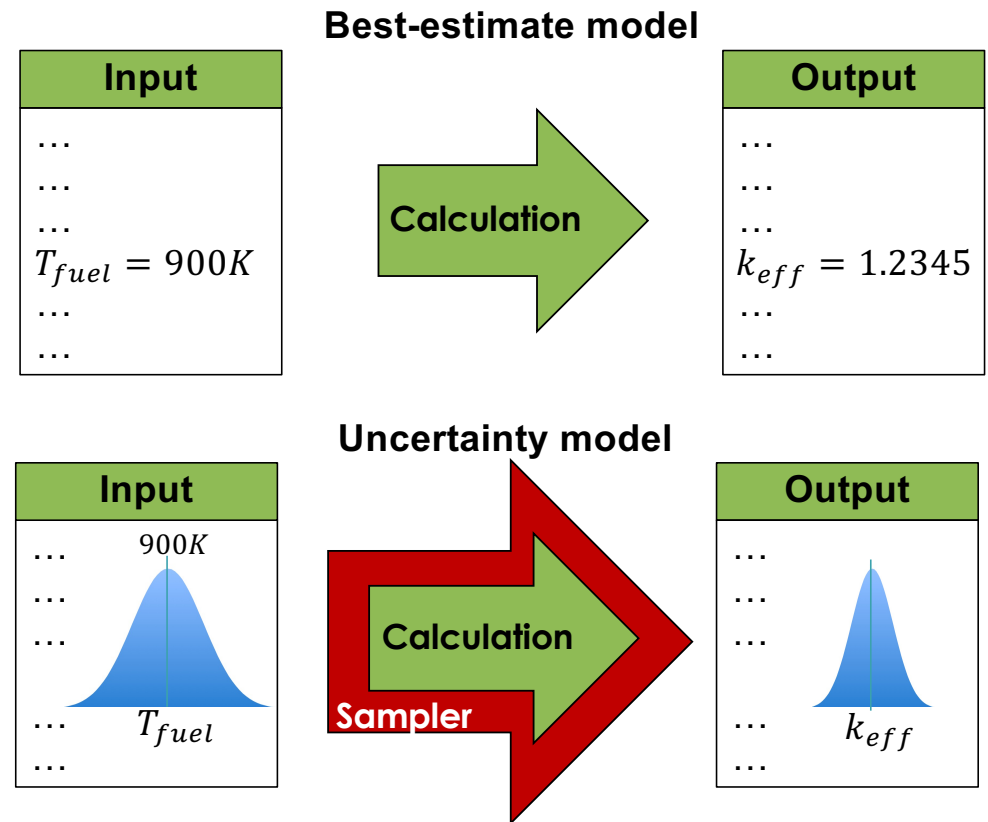
- **nuclear data uncertainty**
 - cross sections
 - fission yield
 - decay
 - recoverable energy
 - kinetics
- input parameter uncertainty
 - geometry
 - composition
 - and anything else that appears in an input file



Goal: Complete UQ in SCALE simulations

Complete means every **number** in the model can have **uncertainty** that is propagated to uncertainty in outputs.

- known as the best-estimate plus uncertainty (BEPU) methodology
- must be applicable to
 - **every input**
cross section, clad thickness, fuel temperature, decay constant, composition, etc.
 - **every SCALE sequence**
TRITON, Polaris, ORIGEN, MAVRIC, etc.
- time-dependent problems may introduce non-linearity, $\Delta f \approx \frac{\partial f}{\partial x} \Delta x$, which first-order perturbation theory methods (TSUNAMI) cannot resolve



Ref: W. Wieselquist, "Uncertainty in Anything: The Future with SCALE/Sampler", SCALE Users' Group Workshop (2017) . <https://www.ornl.gov/scale/scale/2017-scale-users-group-workshop>

Sources of Sampler nuclear data uncertainty

- Cross section covariances:

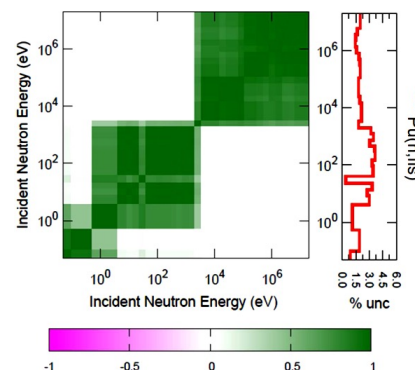
SCALE ENDF/B-VII.1,VIII.0 supplemented by other sources

- Fission product (FP) yield:

SCALE fission standard deviations from ENDF/B-VII; correlations generated by combining independent and cumulative yields

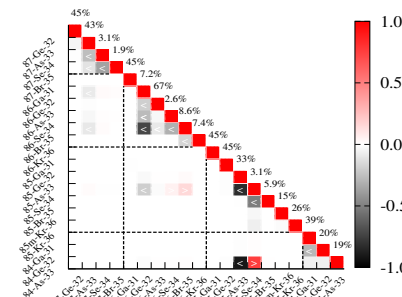
- Decay data:

ENDF-VII.1 modified to include branching correlations due to the constraint that branch sum=1.0

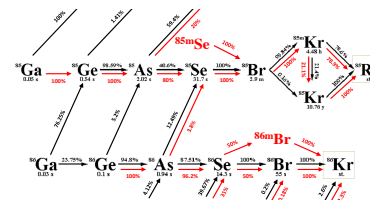


Pu^{239} fission covariance

Fission yield covariance

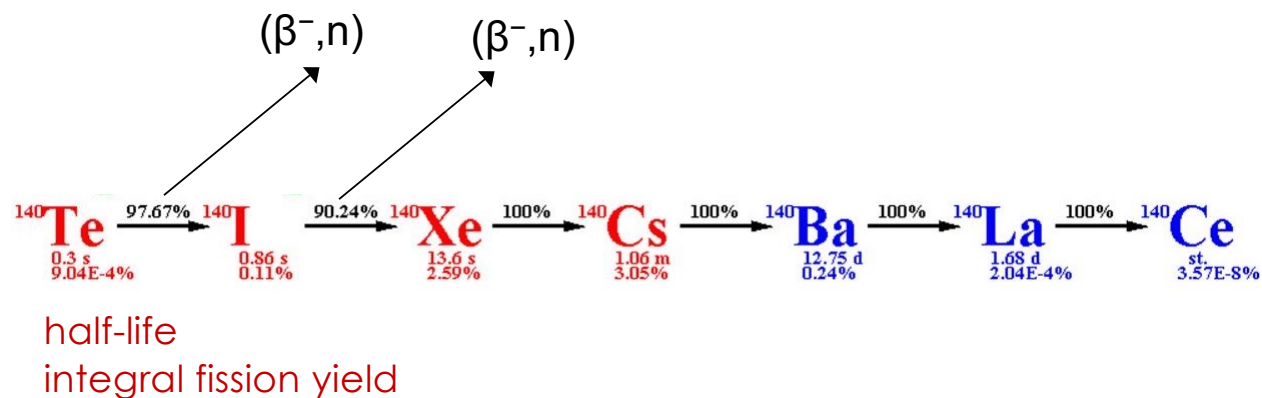


Mass A=85-87 yields



Decay data and FP yield uncertainties

- Independent FP yields are mainly from model calculations...
→ have large uncertainties
- Cumulative FP yields are based largely on measurements...
→ have smaller uncertainties
- Beta-delayed neutron branching ratios
→ usually have large uncertainties



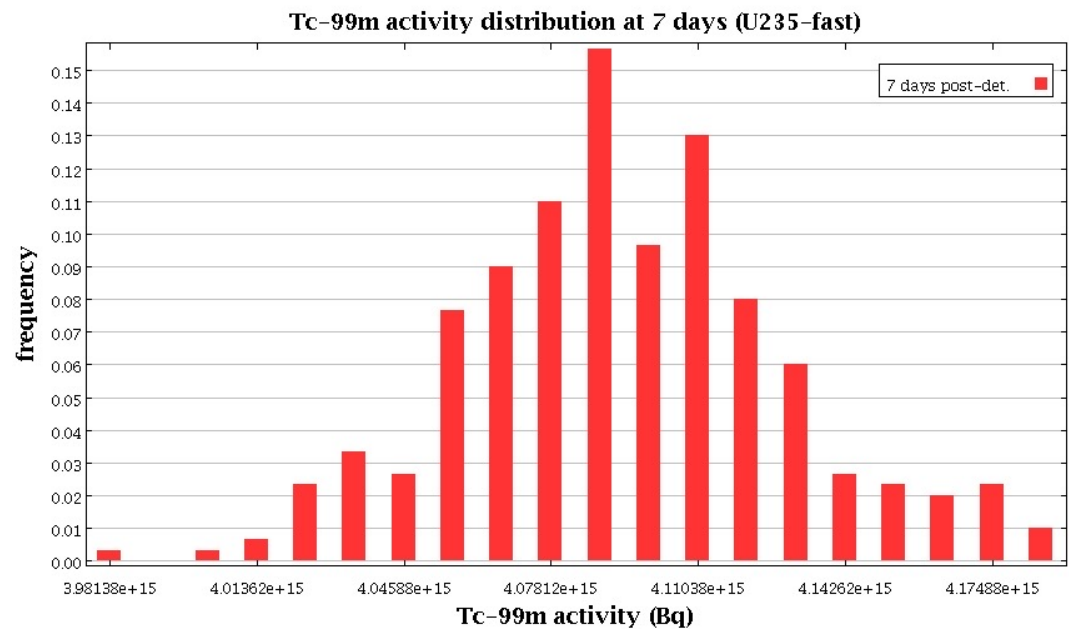
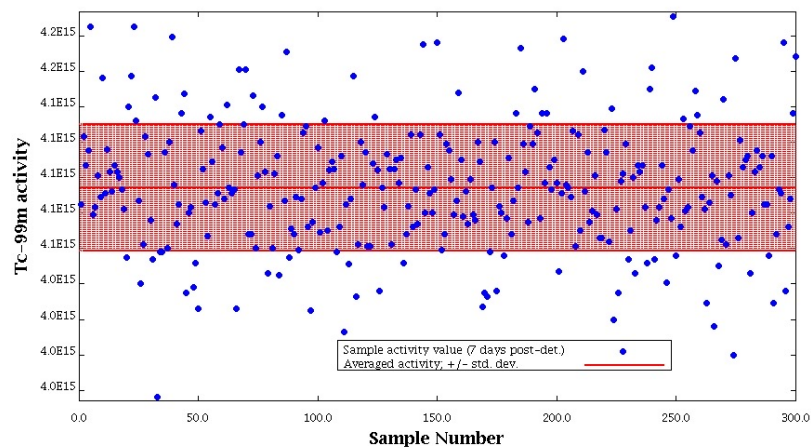
Nuclear data uncertainty

- Sampler can propagate uncertainty assumed in model inputs
 - User has control over **input parameter** uncertainty used
 - User has NO control over **nuclear data** uncertainty used
- Nuclear data uncertainty is a current hot topic
 - How should nuclear data evaluators map uncertainty in measurements to uncertainty in evaluated cross sections?
 - What are the appropriate distributions? Normal? Log-normal?
 - How is cross-correlation handled? Can two reactions be correlated? Can two different nuclides have correlated reactions?
 - SCALE's nuclear data uncertainty in SCALE 6.3 is based on ENDF/B-VII.1 and ENDF/B-VIII.0, supplemented by other sources to fill in gaps

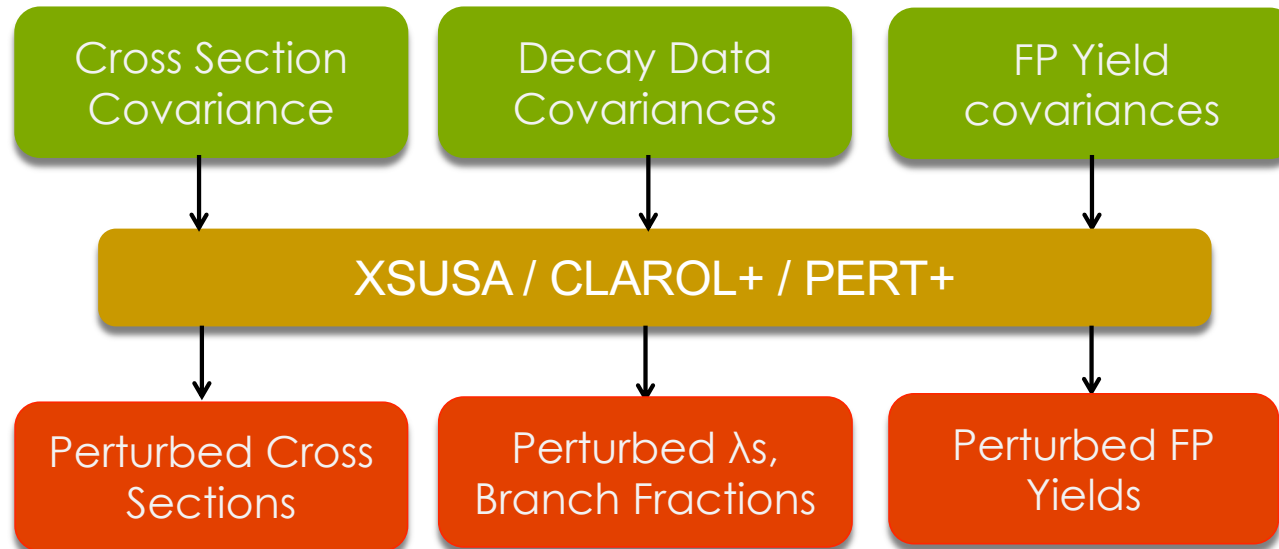
How does Sampler work?

- SCALE provides covariance data for neutron interaction cross sections, FP yields, and decay data
- For a given sample
 - nuclear data are perturbed based on a Gaussian probability distribution characterized by the covariance data
 - perturbed libraries (XS, FP yields, decay) are created for each random sampling
 - SCALE input (1 or more) is run with the perturbed libraries
 - user-requested output data and perturbed files are collected and archived
- After the calculation, Sampler can post-process SCALE output to calculate means, standard deviations, and correlation coefficients
- Sampler can also perturb parameters in the input files based on a user-specified probability distribution

Example: impact of FP yield and decay data uncertainties for ^{235}U fission burst



Generation of perturbed nuclear data in SCALE



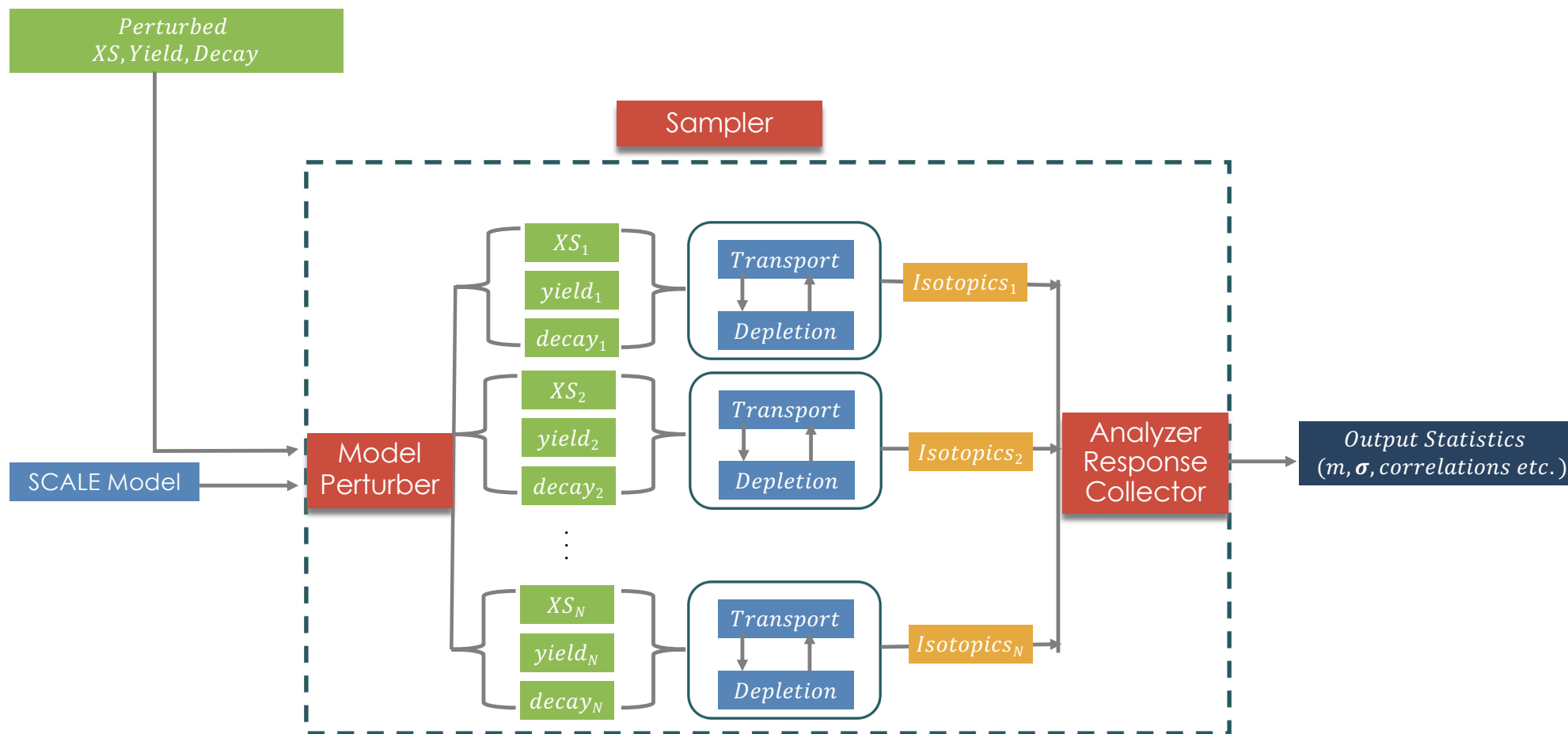
A STATISTICAL SAMPLING METHOD FOR UNCERTAINTY ANALYSIS WITH SCALE AND XSUSA

M. L. WILLIAMS,^{a,*} G. ILAS,^a M. A. JESSEE,^a B. T. REARDEN,^a
D. WIARDA,^a W. ZWERMANN,^b L. GALLNER,^b M. KLEIN,^b
B. KRZYKACZ-HAUSMANN,^b and A. PAUTZ^b

^aOak Ridge National Laboratory, Oak Ridge, Tennessee 37830

^bGesellschaft für Anlagen- und Reaktorsicherheit (GRS)

UQ calculations with coupled transport/depletion



Sampler input is easy

```
=sampler
read parameters
...
end parameters

read case [myCase]**
  sequence=triton/origen/opus
  ...
  end sequence
end case

read variable [myVariable]
...
end variable

read response [myResponse]
...
end response

end
```

Sampler run-time parameters: what to perturb, how many samples, libraries etc.

Case : a set of SCALE sequences (any) for your analysis (e.g., KENO model)

Variable : a variable in your SCALE simulation that is allowed to be modified by Sampler (e.g., tank radius)

Response : quantity of interest that is requested to post-process with Sampler (e.g., k_{eff})

****alternative:**

```
read case [myCase]
  import=/path/to/file
end case
```

Nuclear data perturbations

```
=sampler
read parameters
  n_samples=300
  perturb_xs=yes
  run_cases=no
  library="v7.1-252n"
end parameters
read case[LCT_033_045]
sequence=CSAS5
  LEU-COMP-THERM-033-045
v7.1-252n
  read comp
    h-poly      6 0 6.0586E-02 300.0 END
    f           6 0 1.2302E-02 300.0 END
    c           6 0 2.9128E-02 300.0 END
    u-235       6 0 6.2282E-05 300.0 END
    u-238       6 0 3.0126E-03 300.0 END
    u-234       6 0 6.2548E-07 300.0 END
  end comp
  read parm
    npg=10000 gen=10050 nsk=50 sig=0.00020 htm=no
  end parm
  read geometry
  global unit 1
    cuboid 6 1 40.725 -40.725 43.350 -43.350 44.110 -44.110
  end geometry
  end data
  end sequence
end case
read case[LCT_042_001]
import="/home/wm4/LEU-COMP-THERM-042-001.inp"
end case
```

```
read parameters
  n_samples      = 1000
  perturb_xs     = yes
  perturb_decay  = yes
  perturb_yields = yes
end parameters
```

Note

- **1000** pre-calculated nuclear data perturbations available in SCALE

Perturbation of compositions and geometry

Achieved by using Sampler variables

- Quantify uncertainties or calculate correlation coefficients
- Choose from several parametrized distributions for sampling composition and geometry inputs (**uniform, normal, beta**)
- Use simple mathematical expressions to calculate perturbed inputs ($L = \sqrt{A}/\sin(W)$ where $W = N(\mu, \sigma)$)
- Assign common perturbations to specified cases allowing identical realizations for shared characteristics
- Specify uncertain input parameter by using
 - **Sampler variables**
 - SCALE Input Retrieval Engine (SIREN) to point to its location in the SCALE input

Variables

Embedded Input for Declaring Uncertain Input Parameters

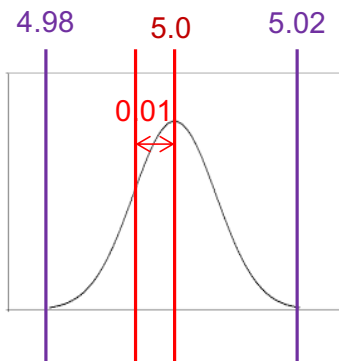
Substitute number in input with #{variable_name} and define variable

SCALE Input Composition

```
read composition
  uo2 1 den=10.4 900
      92235 5.0
      92238 95.0 end
end composition
```

With Placeholder Variables

```
read composition
  uo2 1 den=10.4 900
      92235 #{u235_wo}
      92238 #{u238_wo} end
end composition
```



```
read variable[u235_wo]
  distribution=normal
  value=5.00
  stddev=0.01
  min=4.98
  max=5.02
end variable
```

```
read variable[u238_wo]
  distribution=expression
  expression="100.0 - u235_wo"
end variable
```

Responses

- Collect quantities of interest from the perturbed SCALE cases

```
read response[NAME]  
  type=TYPE  
  [type-specific-params]  
end variable
```

- Type

- **grep** – regular expression from the text output file
- **origen_nuclides** – concentrations from the F71 file
- **stdcmp** – standard composition files
- **opus_plt** – data from an OPUS-generated PLT file
- **triton** – TRITON homogenized cross sections (xfile016)
- **variables** – the geometry perturbation sampled values

grep example

```
read response[keff]  
  type = grep  
  regexp="keff = [[:digit:]]\.[[:digit:]]*"   
  regexp="[[:digit:]]\.[[:digit:]]*"   
end response
```

Regular expression shortcuts for SCALE sequences

regexp = ":kenova.keff:"	k-eff from KENO5 / CSAS5 sequence
regexp = ":kenova.ealf:"	EALF from KENO5 / CSAS5 sequence
regexp = ":kenova.nubar:"	nu-bar from KENO5 / CSAS5 sequence
regexp = ":kenovi.keff:"	keff from KENO6 / CSAS6 sequence
regexp = ":xsdn.lambda:"	lambda from XSDRN / CSAS1 sequence
regexp = ":scale.number:"	matches any number (e.g. "1", "1.0", "1.23e-7", "-0.3")

Response type: **origen_nuclides**

	origen_nuclides
units	fixed (moles)
nuclides	user chooses at post-processing time (do not need to rerun sampler to look at different nuclides!)
filename	*.samplerfiles/ \${SUBDIR}/ \${OUTBASENAME}.f71
format	binary
precision	16 digits

- Multiple names for the file that contains nuclide concentration
 - ORIGIN (binary) concentrations file (F71)
- Traditionally has the file name "ft71f001" in the SCALE temporary directory \${TMPDIR}
- Sampler has its own rule to copy

\${TMPDIR}/ft71f001 →
*.samplerfiles/my_pert_c00001/ my_pert_c00001.f71
- Sampler must find this file
 - If not, it redoes the calculation!
- The extra file copies Sampler sets up must succeed
 - If not, it will not post-process

	origen_nuclides
units	fixed (moles)
nuclides	user chooses at post-processing time (do not need to rerun sampler to look at different nuclides!)
filename	*.samplerfiles/ \${SUBDIR}/ \${OUTBASENAME}.f71
format	binary
precision	16 digits

Response type: **origen_nuclides**

- User can ask for any of the 2237 nuclides tracked by ORIGIN

```

read param
    force_run=no
end param
...
read response[moles]
    type=origen_nuclides
    mixture=1
    nuclides= rn-200 rn-221 rn-222
              rn-223 rn-224 rn-225
              rn-226 end
end response

```

- ft71f001 is a "stack" of nuclide vectors
 - when read (e.g., by ORIGIN or OPUS), a printout of contents is shown
- User can access specific ranges in the ft71f001 using either
 - step_from=X step_to=Y**
 - mixture=C** where C is a case number (ORIGIN/Polaris) or mixture id (TRITON)

Response type: **opus_plt**

- Each “=opus” call creates a numbered *.plt file from an ORIGIN concentration file

*'first opus creates *0.plt file*

=opus

```
data="ft71f001"  
library="end7dec"  
units=curies  
nrank=40  
symnuc=rn-222 end
```

end

- Sampler response accesses the file by the dataset number (depletion step)

*'access *0.plt contents*

read response[mycuries]

```
type=opus_plt  
ndataset=0  
nuclides=rn-222 total end
```

end response

	opus_plt
units	user chooses
nuclides	user chooses at runtime via powerful ranking or explicit selection
filename	*.samplerfiles/ \${OUTBASENAME}.<n>.plt (where <n> is a very long integer starting at 0)
format	text
precision	5 digits

Response type: **opus_plt**

- By default, OPUS gives top 40 nuclides contributing to the units of interest
- In Sampler, only available nuclides can be output—includes powerful total and subtotal
- OPUS .plt format is simple

```
time (d)
Ci
nuclide
6 24
0.000e+00 3.653e+03 3.653e+05 3.653e+07 3.653e+09 3.653e+11
u238 8.001e-05 8.001e-05 8.001e-05 8.001e-05 7.989e-05 6.852e-05
th234 0.000e+00 8.001e-05 8.001e-05 8.001e-05 7.989e-05 6.852e-05
pa234m 0.000e+00 8.001e-05 8.001e-05 8.001e-05 7.989e-05 6.852e-05
u234 0.000e+00 2.238e-09 2.256e-07 1.968e-05 7.989e-05 6.852e-05
pb210 0.000e+00 1.089e-17 1.347e-10 7.086e-06 7.989e-05 6.852e-05
bi210 0.000e+00 1.089e-17 1.347e-10 7.086e-06 7.989e-05 6.852e-05
po210 0.000e+00 1.089e-17 1.347e-10 7.086e-06 7.989e-05 6.852e-05
po218 0.000e+00 1.484e-16 1.347e-10 7.086e-06 7.989e-05 6.852e-05
rn222 0.000e+00 1.484e-16 1.347e-10 7.086e-06 7.989e-05 6.852e-05
ra226 0.000e+00 1.484e-16 1.347e-10 7.086e-06 7.989e-05 6.852e-05
th230 0.000e+00 1.029e-13 1.035e-09 7.086e-06 7.989e-05 6.852e-05
bi214 0.000e+00 1.484e-16 1.347e-10 7.086e-06 7.989e-05 6.852e-05
pb214 0.000e+00 1.484e-16 1.346e-10 7.085e-06 7.987e-05 6.850e-05
po214 0.000e+00 1.484e-16 1.346e-10 7.084e-06 7.987e-05 6.850e-05
pa234 0.000e+00 1.280e-07 1.280e-07 1.280e-07 1.278e-07 1.096e-07
tl210 0.000e+00 3.117e-20 2.828e-14 1.488e-09 1.678e-08 1.439e-08
at218 0.000e+00 2.969e-20 2.693e-14 1.417e-09 1.598e-08 1.370e-08
tl206 0.000e+00 1.458e-23 1.803e-16 9.488e-12 1.070e-10 9.174e-11
rn218 0.000e+00 2.969e-23 2.693e-17 1.417e-12 1.598e-11 1.370e-11
hg206 0.000e+00 2.068e-25 2.559e-18 1.346e-13 1.518e-12 1.302e-12
pb209 0.000e+00 2.650e-24 2.404e-18 1.265e-13 1.426e-12 1.223e-12
bi209 0.000e+00 0.000e+00 5.210e-47 4.726e-41 2.487e-36 3.053e-35
subtotal 8.001e-05 2.402e-04 2.404e-04 3.307e-04 1.119e-03 9.593e-04
total 8.001e-05 2.402e-04 2.404e-04 3.307e-04 1.119e-03 9.593e-04
```

	opus_plt
units	user chooses
nuclides	user chooses at runtime via powerful ranking or explicit selection
filename	*.samplerfiles/ \${OUTBASENAME}.<n>.plt (where <n> is a very long integer starting at 0)
format	text
precision	5 digits

Comparison: **opus_plt** and **origen_nuclides**

	opus_plt	<u>origen_nuclides</u>
units	user chooses	fixed (moles)
nuclides	user chooses at runtime via powerful ranking or explicit selection	user chooses at post-processing time (do not need to rerun sampler to look at different nuclides!)
filename	*.samplerfiles/ \${OUTBASENAME}.<n>.plt	*.samplerfiles/ \${SUBDIR}/ \${OUTBASENAME}.f71
format	text	binary
precision	5 digits	16 digits

Grep example

- within .out file

```
best estimate system k-eff      0.99352 + or - 0.00019
```

- grep to extract

```
read response[mykeff]  
  type=grep  
  regexp="best estimate system k-eff[[:space:]]+[[:digit:]]+\.[[:digit:]]+"  
  regexp="[[:digit:]]\.[[:digit:]]+"  
end response
```

- alternative using ":kenova.keff:"

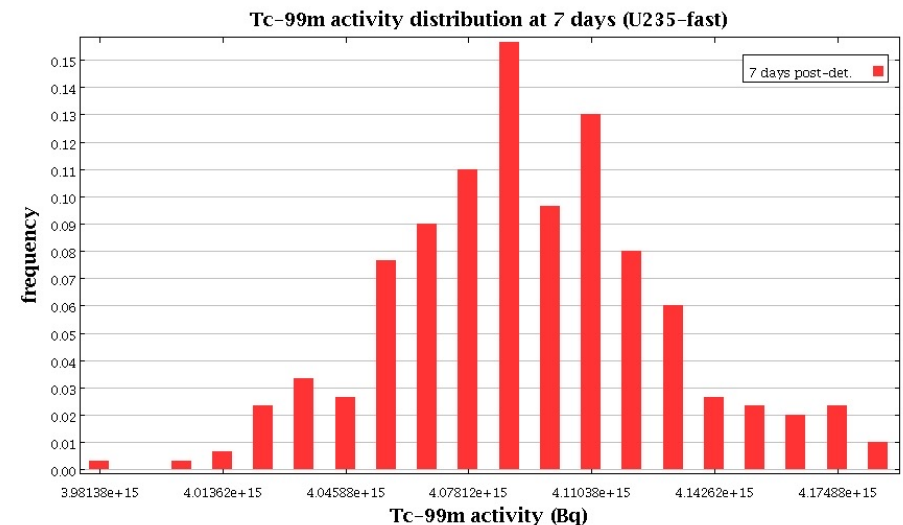
```
read response[mykeff]  
  type=grep  
  regexp="keff = :scale.number:"  
  regexp= ":kenova.keff:"  
  eregexp=":kenova.keff:"  
end response
```

Notes:

- 1) "regexp" gets the k_{eff} value.
- 2) "eregexp" gets the uncertainty (error) in the k_{eff} value.
- 3) In most cases, the uncertainty of interest is the standard deviation of the k_{eff} values themselves, and the uncertainty in each case k_{eff} is more or less neglected.

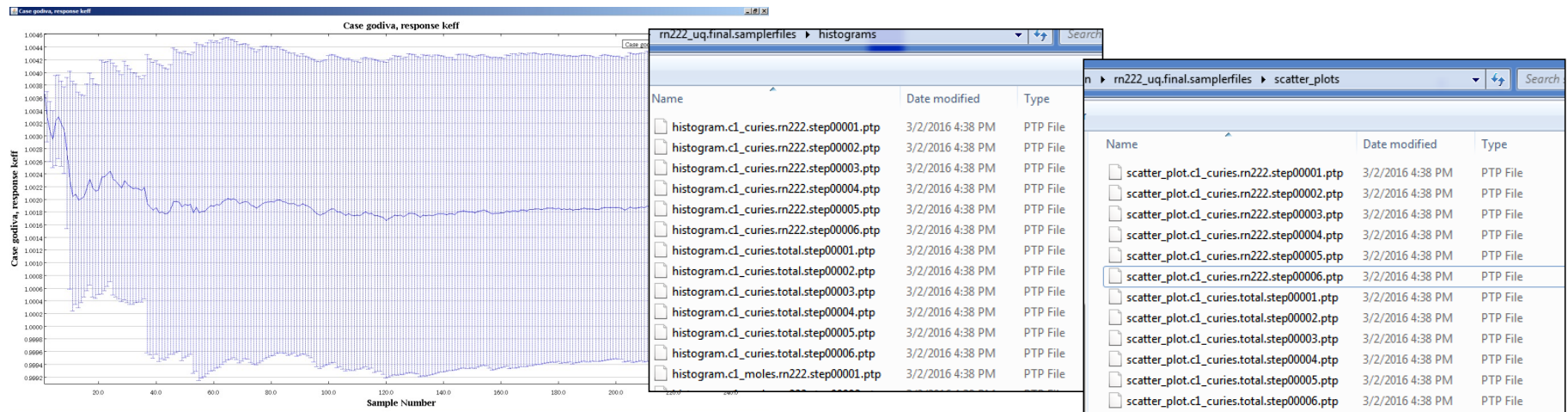
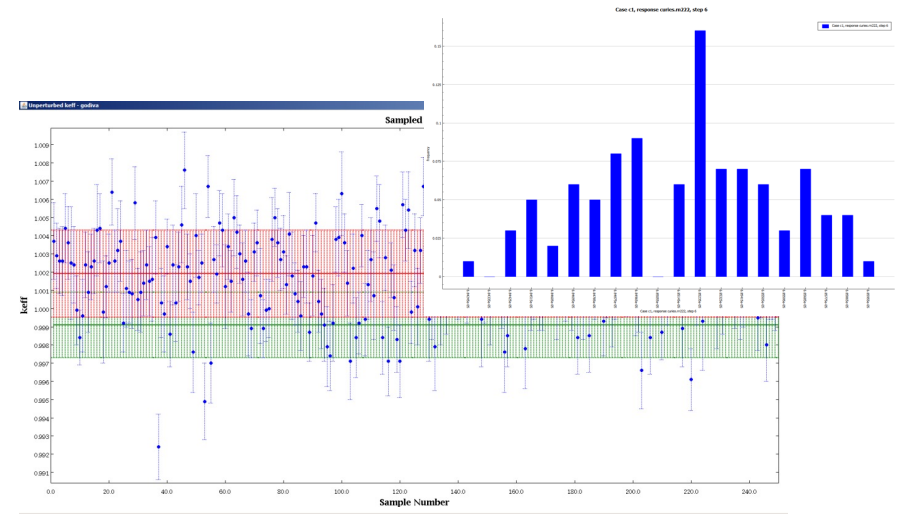
Post-processing

- Creates response tables as requested
 - average values, correlations, covariances, etc.
- Correlations created for multiple cases
 - e.g., critical experimental correlations
- PTP plot files
 - running averages, histograms, scatter plots



Post-processing (cont.)

- PTP plot files
 - One for each time value of the response
 - Scatter plots showing data points
 - average vs. unperturbed
 - Histograms
 - Running averages



SCALE 6.3.1 | Last Updated: March 2024

Example for k_{eff} : input

- LEU-COMP-THERM-033-045 case:
homogenous mix of UF_4 and paraffin
- 300 realizations (samples)
- v7.1-252n library

```
=sampler
read parameters
  n_samples=300
  perturb_xs=yes
  run_cases=no
  library="v7.1-252n"
end parameters
read case[LCT_033_045]
sequence=CSAS5
  LEU-COMP-THERM-033-045
v7.1-252n
  read comp
    h-poly      6  0   6.0586E-02   300.0 END
    f           6  0   1.2302E-02   300.0 END
    c           6  0   2.9128E-02   300.0 END
    u-235       6  0   6.2282E-05   300.0 END
    u-238       6  0   3.0126E-03   300.0 END
    u-234       6  0   6.2548E-07   300.0 END
  end comp
  read parm
    npg=10000 gen=10050 nsk=50 sig=0.00020 htm=no
  end parm
  read geometry
    global unit 1
    cuboid 6 1 40.725 -40.725 43.350 -43.350 44.110 -44.110
  end geometry
  end data
  end sequence
end case

read response[keff]
  type = grep  regexp = ":kenova.keff:"  eregexp = ":kenova.keff:"
end response
end
```

SCALE 6.3.1| Last Updated: March 2024

Example for k_{eff} : results

- Nominal k_{eff} : 0.99419 ± 0.00019
- Average of 300 realizations: 0.99424
 - Good agreement with nominal
- Standard deviations of 300 realizations: $0.00626 \Delta k$
- TSUNAMI-3D estimate of data-induced uncertainty: $0.00619 \Delta k$
- Good agreement between sensitivity-based and sampling-based k_{eff} uncertainty

Questions?

