Los Alamos
NATIONAL LABORATORY
EST. 1943

LA-UR-16-27623
Approved for public release; distribution is unlimited.

Title: ROMUSE 2.0 User Manual

Author(s): Khuwaileh, Bassam
Turinsky, Paul
Williams, Brian J.

Intended for: Report

Issued: 2016-10-04

Disclaimer:
Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

ROMUSE 2.0 User Manual

By Bassam Khuwaileh, Paul Turinsky**, and Brian Williams**

**Los Alamos National Laboratory, Los Alamos, NM, U.S.A*

***North Carolina State University, Raleigh, NC, U.S.A*

Abstract

ROMUSE (**R**educed **O**rder **M**odeling **B**ased **U**ncertainty/**S**ensitivity **E**stimator) is an effort within the Consortium for Advanced Simulation of Light water reactors (CASL) to provide an analysis tool to be used in conjunction with reactor core simulators, especially the Virtual Environment for Reactor Applications (VERA). ROMUSE is written in C++ and is currently capable of performing various types of parameters perturbations, uncertainty quantification, surrogate models construction and subspace analysis. Version 2.0 has the capability to interface with DAKOTA which gives ROMUSE access to the various algorithms implemented within DAKOTA.

ROMUSE is mainly designed to interface with VERA and the Comprehensive Modeling and Simulation Suite for Nuclear Safety Analysis and Design (SCALE) [1,2,3], however, ROMUSE can interface with any general model (e.g. python and matlab) with Input/Output (I/O) format that follows the Hierarchical Data Format 5 (HDF5). In this brief user manual, the use of ROMUSE will be overviewed and example problems will be presented and briefly discussed. The algorithms provided here range from algorithms inspired by those discussed in Ref.[4] to nuclear-specific algorithms discussed in Ref. [3].

Contents

Introduction to ROMUSE

ROMUSE (**R**educed **O**rder **M**odeling **B**ased **U**ncertainty/**S**ensitivity **E**stimator), is a C++ based analysis code that is designed mainly to be used in conjunction with reactor analysis codes (e.g. reactor core simulators) to perform different forms of mathematical analysis on the simulator of interest (e.g. uncertainty quantification, surrogate model construction, and subspace analysis). ROMUSE interfaces with the I/O of the simulator of interest such that the I/O data are wrapped, modified and then used in ROMUSE modules to make conclusions about the problem of interest (refer to FIGURE 1).

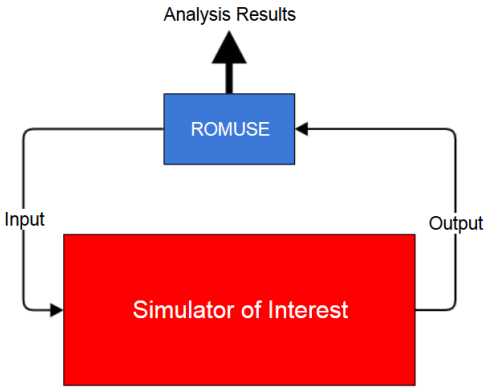

FIGURE 1.ROMUSE SCHEME.

Currently, ROMUSE is compatible with SCALE6.1, VERA-CS and any code with HDF5 I/O format. Depending on the simulator of interest, the ROMUSE input card requires various types of information. Table 1 summarizes the main input card parameters. ROMUSE can read, manipulate and write input parameters. Moreover, response level analysis is possible by executing the simulator of interest and then interpreting and analyzing the responses of interest.

In order to provide flexibility in using ROMUSE, it can be used at different levels (refer to Figure 2); for example, it can be used to generate perturbed cases only, without running the simulator of interest (refer to the next section). Furthermore, the user can command ROMUSE to further run the simulator and collect the corresponding responses of interest. Finally, ROMUSE can be interfaced with DAKOTA 6.4 (6.1+) and have access to the algorithms therein. In the latter case ROMUSE will compute the quantities needed for the problem of interest and provide them to DAKOTA.

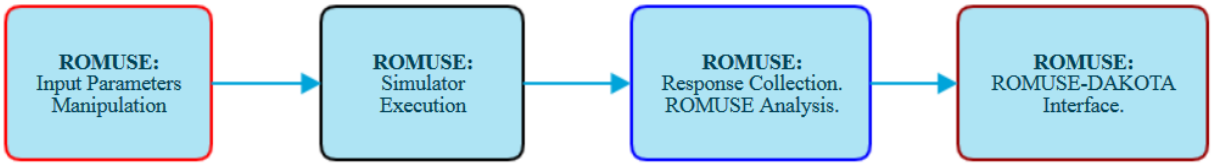

FIGURE 2. ROMUSE EXECUTION LEVELS.

ROMUSE provides various sequences, each have different parameters and perform different algorithm or execution flow. Table 1 summarizes the main input card parameters with a brief description. However, in the coming sections these parameters will be mentioned in the corresponding sections.

Comprehensive Uncertainty Quantification (UQ) studies can be performed via ROMUSE. Different UQ algorithms available are:

1- Brute force Monte Carlo UQ.
2- Karhunen-Loeve (KL) expansion based (might appear as the Multi-Physics Efficient Range Finding Algorithm – **MP-EUQ** in the case of coupled multi-physics codes).
3- Surrogate Based UQ (SBUQ).
4- DAKOTA based UQ and DA studies.

For more information about the first 3 options, refer to Ref. [3]. Options 2-4 requires performing dimensionality reduction which can be computed via ROMUSE, which implements efficient algorithms for single physics and multi-physics dimensionality reduction which are explained in details in Ref. [3]. These algorithms are then used to perform dimensionality reduction on the uncertainty source space (i.e. revealing the active or important DoFs). Once the active DoFs are determined they can be used to perform linear KL-based uncertainty quantification, surrogate construction, or can be communicated with DAKOTA. Note that the material (nuclei) and reaction ids used throughout ROMUSE follow the notation used by the SCALE covariance and cross-sections library [1].

TABLE 1. ROMUSE INPUT CARD MAIN PARAMETERS.

Parameter	**Description**
SEQUENCE=	The sequence name: **X-ROMUSE-UQ** ;*X={GMODEL,SCALE,VERA}* **X-ROMUSE-RFA** ;*X={GMODEL,SCALE,VERA}* **ROMUSE-EXE;** **ROMUSE-RESPONSES;** **ROMUSE-DAKOTA;**
Ref:	Directory of reference case run
Command:	The command used to execute the simulator of interest.
Queue Names:	Queues to be used for executing the commands and jobs submission
Max Model Runs:	The number of model snapshots used for the analysis
Perturbation Distribution:	Type of sampling distribution: **Uniform** **Normal**
Perturbation Range:	The range of perturbation (relative standard deviation in the case of normal distribution)

Number of Parameters:	The number of parameters to be perturbed (in the case of GMODEL)
Number of Responses:	The number of responses to be analyzed (always required if ROMUSE is used for more than input perturbation)
END	A flag to end the sequence information
PoI:	Parameters of Interest: two lines come after this, the first is the parameters file name and the second is the database path to the parameters (required in the case of GMODEL)
SCALE-PoI:	Parameters of Interest: If all parameters are to be perturbed "**ALL**" must be placed in the next line, otherwise, "**Materials:**" and "**Reactions:**" parameters must be used. Used with SCALE.
VERA-PoI:	Parameters of Interest: If all parameters are to be perturbed "**ALL**" must be placed in the next line, otherwise, "**Materials:**" and "**Reactions:**" parameters must be used. Used with VERA.
RoI:	Response of Interest: two lines come after this, the first is the response file name and the second is the database path to the response (required in the case of GMODEL)
Given Perts File:	The name of the file containing the given perturbations.
Library Name:	The name of the formatted library (required in the case of SCALE,VERA)
Covariance Directory:	The directory containing the unformatted covariance data
S-SPACE=	Output the basis (default = ON) {ON,OFF}
EUQ=	Perform KL-Based UQ or MP-EUQ (default=OFF) {ON,OFF}
SMCUQ=	Perform Surrogate Based MC UQ (default=OFF) {ON,OFF}
MCUQ=	Perform brute force MC UQ (default=ON) {ON,OFF}

System Requirements and Specifications

ROMUSE is a C++ code compiled and tested via gcc version 4.8.3 on linux Red Hat 4.8.3-9 (64 bit). All the tests are performed on CentOS Linux release 7.1.1503 available on NCSU HPC. ROMUSE package comes with the required boost, Eigen, and HDF5 libraries. Moreover, ROMUSE comes with the

required covariance library, which is based on SCALE covariance library "*44groupcov*" formatted in a binary file and a text formatted library [1].

Once in ROMUSE root directory, the user can compile the code by first accessing the MakeFile and modifying the path and compiler parameters according to the system's structure and then typing "*make –B ROMUSE*".

The structure of the ROMUSE problem must be such that all input files, tagged input files, reference Libraries, and/or reference cases are in the same directory. The ROMUSE input file must be named "*romuse.in*" and the simulator input file must be named "*input.txt*". The rest can be customized in ROMUSE input file as will be shown latter in this manual.

Perturbing Input Parameters

Tags: Perturb, parameters, uncertainty quantification, Subspace Finding, RFA, validation, verification

Introduction

Perturbing input parameters is a vital feature for any analysis code as many mathematical and statistical analyses such as Uncertainty Quantification and Data Assimilation require manipulating the parameters set. However, in order to build an analysis code that can work with a wide variety of simulators, ROMUSE perturbation module is designed such that it allows the user to manipulate the input parameters of any given model as long as the parameters are formatted in Hierarchical Data Format 5 (HDF5) format or SCALE cross-sections library binary format [1] or VERA cross-section library format [2, 6]. The ability to perturb any parameters in HDF5 formatting allows ROMUSE to work with any simulator with such I/O format.

ROMUSE has been built to serve nuclear reactor simulators, therefore special attention has been given to nuclear reactor physics problems, mainly simulated via SCALE [7] and VERA-CS [2]. ROMUSE user can determine the type of perturbations depending on the subsequent application. ROMUSE offers three types of possible perturbations:

1- **Covariance Based Perturbations**: this type of perturbation requires a covariance library to be provided so that the perturbations can be generated along the given covariance library. This perturbation type can be used to generate Monte Carlo samples for Uncertainty Quantification studies.
2- **Given Perturbations**: the user can provide his/her own perturbations to ROMUSE. This type of perturbation is very useful for validation/verification studies where users are to study the behavior of the model of interest over an interval in the parameter's space.
3- **Random Perturbations**: ROMUSE can introduce random perturbations into the nuclear data cross-sections. The user must provide a range of the perturbations and choose a distribution to drag perturbations from. This type of perturbation is important for applications such as the subspace analysis, surrogate modeling and other important analysis types.

The three types mentioned above can be applied to any type of parameters (e.g. cross-sections). The only requirement is to have the parameters in SCALE, VERA, or HDF5 input format.

TABLE 2. PARAMETERS PERTURBATIONS INPUT CARD ENTRIES.

Parameter	Description
SEQUENCE=	**Y-ROMUSE-X** *{Y=SCALE,VERA,GMODEL} |{X=UQ, RFA,GP}*
Ref:	**Path to the reference case**
Max Model Runs:	**Number of perturbed cases to be generated**
Library Name:	**Name of binary cross-section library**
SCALE-PoI:	**The parameters of interest. "ALL" to perturb all parameters or use "Materials:" and "Reactions:" Parameters for specific nuclei, reaction pairs. Required with Y=SCALE.**
VERA-PoI:	**The parameters of interest. "ALL" to perturb all parameters or use "Materials:" and "Reactions:" Parameters for specific nuclei, reaction pairs. Required with Y=VERA**
PoI:	**HDF5 Formatted Parameters. Example:** *PoI:* */Path/to/Database* *Parameter Database address*
Covariance Directory:	**Path to the Covariance Library(this library comes with ROMUSE installation)**
Perturbation Range:	**The range of perturbation (required if SCALE-ROMUSE-RFA)**
Perturbation Distribution:	**Uniform or Normal**
Materials:	**The Materials or Nuclei to be perturbed. Example:** *Materials:* *N # Number of materials or nuclei* *For 1=1,N* *Nuclei ID # Nucleus or Material ID.*
Reactions:	**The reactions be perturbed. Example:** *Reactions:* *M # Number of Reactions* *For 1=1,M* *Reaction ID # Reaction ID.*
END	**Flag that ends the sequence.**

SCALE Cross-Sections Library Perturbation

SCALE cross-section library "*44groupcov*" is formatted in a binary file which can be read via ROMUSE. In order to generate perturbed libraries for a SCALE case, ROMUSE requires the following information (refer to Table 2):

1- Reference library path.
2- Type of perturbation: depending on the perturbation type further requirements might be needed.
   a- Covariance Based Perturbation: covariance library path and the number of Monte Carlo samples.
   b- Given Perturbation: the given perturbations file path.
   c- Random Perturbation: the relative range of perturbations (e.g. ±10%) and the number of samples.

Example Input Card (1)

In this example case (refer to Figure 3), ROMUSE will generate 100 different cross-section libraries based on a covariance library stored in "**/Path/To/Covariance/Files/**". Note that ROMUSE assumes that the user has already run a reference case which is located at "**/Path/To/Reference/Case/**". ROMUSE will perturb all cross-sections with covariance data in the provided path.

After running the card below with the correct reference case and library, a new directory called "**SNAPS**" with subfolders labeled "**snap_0,…,snap_99**" will be created. Each of these subfolders will have a full case imitating what is inside the reference case provided with a perturbed library.

Corresponding Example: Manual/Examples/SCALE/UQ

```
SEQUENCE=SCALE-ROMUSE-UQ

Ref:
Ref-Case

Max Model Runs:
100
SCALE-Pol:
ALL

Covariance Directory:
/Path/to/Covariance/Library

Library Name:
ft04f001

END
```

FIGURE 3. SCALE UQ PERTURBATION INPUT CARD.

Example Input Card (2)

In this example case (refer to Figure 4), ROMUSE will generate 100 different perturbed cases. The perturbations are generated via a uniform distribution with ±10% of the reference cases. Note that ROMUSE assumes that the user has already run a reference case which is located at "**/Path/To/Reference/Case/**". ROMUSE will perturb the specified cross-sections corresponding to 2 nuclei (U^{235} and U^{238}) and 3 reactions (total cross section, fission cross-section and elastic scattering [ID: 1,18,2 respectively]).

After running the card below with the right reference case and library, a new directory called "**SNAPS**" with subfolders labeled "**snap_0,…,snap_99**" will be created. Each of these subfolders will have a full case imitating what is inside the reference case provided with a perturbed library.

Corresponding Example: Manual/Examples/SCALE/RFA

```
SEQUENCE=SCALE-ROMUSE-RFA
Ref:
Ref-Case
Max Model Runs:
100
Perturbation Range:
0.1
Perturbation Distribution:
Uniform
SCALE-Pol:
Materials:
2
92235
92238
Reactions:
3
1
18
2
Library Name:
ft04f001
END
```

FIGURE 4.SCALE-RFA EXAMPEL INPUT CARD.

VERA Cross-Sections Library Perturbation

VERA cross-section library is actually MPACT covariance library [6]. ROMUSE is able to read this library into a vector and then manipulate it in different manners. As mention in the previous section, the information required depends on the type of perturbation.

Currently ROMUSE is compatible with the MPACT 47 groups VERA library (refer to **Table 3**). This library includes 295 different isotopes with different reaction cross-sections data. But currently ROMUSE has only access to the 44 groups covariance library [1], therefore, ROMUSE has a mapping

capability that is able to map perturbations from one group structure to another. Mapping the perturbations is performed via linear interpolation based on the assumption of constant lethargy intervals.

The perturbed libraries are generated via ROMUSE. ROMUSE reads the "44GroupCov" covariance library available within SCALE6.1 package and perturbs the 47 group MPACT cross-section library version 4. ROMUSE decomposes the 44-group matrix, creates 44-group perturbation factors, projects the 44-group perturbation factors to 47-group perturbation factors, then perturbs the 47-group cross-sections.

Based on the concept of a constant lethargy flux as the weight function, if the group structures do not align, ROMUSE determines the fractional lethargy width of the un-aligned boundaries and uses these terms to map the cross-section perturbations from one group structure to another [9, 3].

The following equation exemplifies how to map a perturbation from the *A* group structure to the *B* group structure:

$$\Delta\sigma_i^B = \Delta\sigma_{j-1}^A + \frac{\Delta\sigma_j^A - \Delta\sigma_{j-1}^A}{\left|L\left(E_j^A\right) - L\left(E_{j-1}^A\right)\right|} \times \left|L\left(E_i^B\right) - L\left(E_{j-1}^A\right)\right|.$$

where $\Delta\sigma_i^B$. is the perturbation associated with the i^{th} group in the *B* group structure and $\Delta\sigma_j^A$ is the computed perturbation associated with the j^{th} group in the *A* group structure. In this case $i \in \left(j-1, j\right)$. *or* $E_i^A \in \left(E_j^B, E_{j-1}^B\right)$. $L\left(E_j^A\right)$. is the lethargy term for E_j^A. Note that when groups align the perturbations are not altered.

TABLE 3. 47 GROUP STRUCTURE.

g	*Energy Boundary*	*g*	*Energy Boundary*	*g*	*Energy Boundary*	*g*	*Energy Boundary*
1	20 MeV	13	78.9 eV	25	2.3824 eV	37	0.5032 eV
2	6.0653 MeV	14	47.8512 eV	26	1.8554 eV	38	0.35767 eV
3	3.6788 MeV	15	29.023 eV	27	1.4574 eV	39	0.2705 eV
4	2.2313 MeV	16	13.71 eV	28	1.2351 eV	40	0.18443 eV
5	1.3534 MeV	17	12.099 eV	29	1.1664 eV	41	0.14572 eV
6	0.8208 MeV	18	8.3153 eV	30	1.1254 eV	42	0.11157 eV
7	4.9787 MeV	19	7.33822 eV	31	1.0722 eV	43	0.08197 eV
8	0.1832 MeV	20	6.47602 eV	32	1.0137 eV	44	0.0569 eV
9	67.38 KeV	21	5.715 eV	33	0.97100 eV	45	0.0428 eV
10	9.119 KeV	22	5.04348 eV	34	0.9099 eV	46	0.0306 eV
11	2.0347 KeV	23	4.4509 eV	35	0.7821 eV	47	0.0124 eV
12	0.13 KeV	24	3.9279 eV	36	0.62506 eV	-	-

Example Input Card (1):

In this example case, ROMUSE will generate 100 different cross-section libraries based on the covariance library. Note that ROMUSE assumes that the user has already run a reference case which is located at "**/Path/To/Reference/Case/**". ROMUSE will perturb all cross-sections available in the library and has covariance data in the covariance library.

ROMUSE will generate a new directory called "**SNAPS**" with subfolders labeled "**snap_0,…,snap_99**". Each of these subfolders will have a full case imitating what is inside the reference case provided with a perturbed library.

Corresponding Example: Manual/Examples/VERA/UQ

```
SEQUENCE=VERA-ROMUSE-UQ

Ref:
REF-Case
Max Model Runs:
100
VERA-PoI:
ALL
Library Name:
mpact47g_70s_v4.0_11032014.fmt
Covariance Directory:
/Path/to/Covariance/Library

END
```

FIGURE 5. VERA-UQ INPUT CARD.

General Parameter Perturbation

In addition to the perturbation modes mentioned above, ROMUSE can manipulate the parameters of any simulator with HDF5 input format. In this case the user should provide ROMUSE with the database address of the I/O data.

Example Input Card (1)

Assuming that we have a PYTHON, MATLAB based simulator or any other simulator with HDF5 formatted I/O, then the following ROMUSE input card will be able to manipulate the input parameters.

Running the card below (refer to Figure 6) will generate 100 different perturbed Input cases based on a normal distribution with a ±10% standard deviation of the parameter value. Note that the user can use **SEQUENCE=GMODEL-ROMUSE-RFA** and then provide a covariance library. The samples will be Monte Carlo samples that can be used for Uncertainty Quantification. Moreover, note that ROMUSE assumes that the user has already run a reference case which is located at "**/Path/To/Reference/Case/**". The parameters HDF5 file name is **Params.h5** and the address of the parameters of interest is "**/Params**"

ROMUSE will generate a new directory called "**SNAPS**" with subfolders labeled "**snap_0,…,snap_99**". Each of these subfolders will have a full case imitating what is inside the reference case provided with a perturbed input file (in HDF5 format)

Corresponding Example: Manual/Examples/GMODEL/RFA

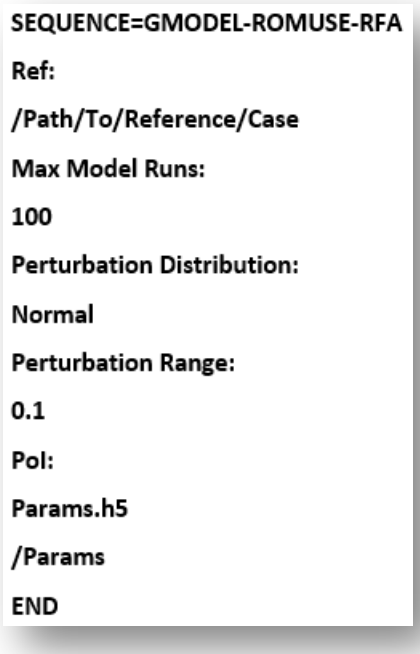

FIGURE 6.GMODEL-RFA INPUT CARD.

Complex Sequences

Obviously, perturbing the input parameters must be followed by various steps depending on the goal of the parameter perturbation performed in the first place. Therefore, complex sequences are required. In the following subsection, a new type of ROMUSE sequences is introduced. The sequences introduced herein are referred to as "*Complex Sequences*" due to their nature. These sequences consist of a combination of different types of sequences to perform certain tasks. The following subsections present a summary of two important types of complex sequences.

Execution and Response Collection: ROMUSE-EXE and ROMUSE-RESPONSES

ROMUSE-EXE sequence comes after any parameters perturbation case. Providing this sequence in ROMUSE input file will execute the perturbed cases. Two key pieces of information must be provided; the command to be used for the execution and the job submission details (refer to Table 4). Note that for this sequence to work as expected, a pre-requisite perturbation sequence must exist such that the "**SNAPS**" folder is already there.

ROMUSE-RESPONSES sequence provides information about the responses of interest to be collected and analyzed. It is always preferred to have the responses in a database format (i.e. HDF5 VERA-CS format). However, in many cases such as SCALE and DAKOTA, the output is dumped onto text files.

TABLE 4. ROMUSE-EXE AND ROMUSE-RESPONSES INPUT CARD PARAMETERS.

SEQUENCE=	ROMUSE-EXE or ROMUSE-RESPONSES
Command:	/path/to/command
Queue Names:	Job submission information. Example: N # *number of queues* M # *number of cores per job* For i=1:N Queue Name # *C: capacity of this queue*
Flags:	Single Physics=YES/NO (Default NO) Multi Physics=YES/NO (Default NO) (required for ROMUSE-RESPONSES)
Simulator Out:	Output-File-Name Response Tag (the tag to search for in a text formatted output files)
END	A flag to end the sequence information

Example Input Card (1)

Running the card below (Figure 7) will generate 3 different perturbed Input cases based on the gradient free Range Finding Algorithm (RFA), uniform distribution ±20% of the reference value of U^{235} and U^{238} total for fission and elastic scattering cross-sections. Note that the user can use **SEQUENCE=SCALE-ROMUSE-UQ** and then provide a covariance library (refer to Figure 8). In this case, the samples will be Monte Carlo samples that can be used for Uncertainty Quantification. Moreover, note that ROMUSE assumes that the user has already run a reference case which is located at Ref-CASE="**/Path/To/Reference/Case.**

ROMUSE will generate a new directory called "**SNAPS**" with subfolders labeled "**snap_0,…,snap_2**". Each of these subfolders will have a full case imitating what is inside the reference case provided with a perturbed input file (in HDF5 format)

Corresponding Example: Manual/Examples/SCALE/RFA-EXE and Manual/Examples/SCALE/UQ-EXE

```
SEQUENCE=SCALE-ROMUSE-RFA
Ref:
Ref-Case
Max Model Runs:
3
Perturbation Range:
0.2
SCALE-PoI:
Materials:
2
92235
92238
Reactions:
3
1
18
2
Library Name:
ft04f001
END

SEQUENCE=ROMUSE-EXE
Command:
/Path/to/SCALE/ batch6.1
Queue Names:
1
1
zeus 64
END

SEQUENCE=ROMUSE-RESPONSES
Simulator Out:
input.out
k-eff =
Flags:
Single Physics=YES
END
```

FIGURE 7. ROMUSE-EXE AND ROMUSE-RESPONSES EXAMPLE INPUT CARD (RFA).

```
SEQUENCE=SCALE-ROMUSE-UQ
Ref:
Ref-Case
Max Model Runs:
50
SCALE-Pol:
Materials:
2
92235
92238
Reactions:
3
1
18
2
Covariance Directory:
/Path/to/Covariance
Library Name:
ft04f001
END

SEQUENCE=ROMUSE-EXE
Command:
/Path/To/SCALE/batch6.1
Queue Names:
1
1
gnep 92
END

SEQUENCE=ROMUSE-RESPONSES
Simulator Out:
input.out
k-eff =
Flags:
Single Physics=YES
END
```

FIGURE 8. ROMUSE-EXE AND ROMUSE-RESPONSES EXAMPLE INPUT CARD (UQ).

ROMUSE-DAKOTA

This sequence requires the presence of the two previous sequences. This sequence offers an interface between ROMUSE and any simulator (General Model, SCALE and VERA-CS). In the following subsection a full SCALE6.1 based example will be introduced; later versions of this manual will present further models resembling using ROMUSE with VERA-CS (refer to Figure 9).

This interface is tricky, due to the fact that the different required parameters depend on the DAKOTA method of interest. For more information about DAKOTA please refer to Ref. [5]. In order to solve this problem ROMUSE will adapt a flexible technique that is almost independent of the DAKOTA method used.

ROMUSE can compute a lot of interesting quantities related to the simulator of interest (SCALE, VERA-CS, and General Models) which are summarized in Table 5, for example: parameter's standard deviation, parameter's mean**,** response's mean, response's standard deviation, parameter and responses perturbations, sensitivity coefficients and many more. The premise of this interface is to make any quantity read, computed or generated via ROMUSE available to DAKOTA such that the construction of any DAKOTA case will be easier through the ROMUSE input card parameters summarized in Table 6. Therefore, in order to develop a general interface between ROMUSE and DAKOTA depends on a Dakota input file provided by the user and placed in the same directory with "*romuse.in"* file. However, this DAKOTA input must have special tags telling ROMUSE the structure of the DAKOTA input file.

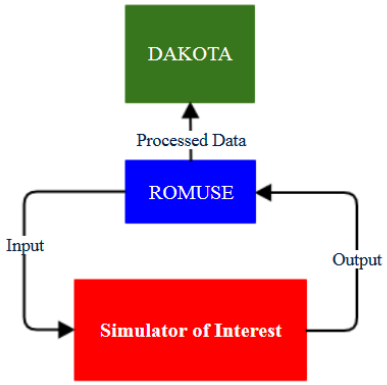

FIGURE 9.ROMUSE-DAKOTA INTERFACE SCHEME.

TABLE 5. ROMUSE-DAKOTA TAGS.

Tag	Associated Quantity
$np	Number of Parameters
$m	Parameter's Mean
$std	Parameter's Standard Deviation (Absolute)
$p	Parameter's Descriptors
$tnr	Number of Response Functions
$ip	Parameter initial points
$lp	Parameter lower bound
$up	Parameter upper bound

TABLE 6.ROMUSE-DAKOTA INTERFACE INPUT PARAMETERS.

SEQUENCE=	ROMUSE-DAKOTA
Surrogate Type:	e.g. P-01, P-02, P-03, Gaussian-surfpack,Gaussian-dakota
Number of Validation Points:	Number of Points for Validation
Max Number of Active Subspace Construction Points:	Maximum number of snapshots used for subspace construction.
Subspace Approximation Error Tolerance:	Error tolerance for the algorithm of interest.
Recycle Samples=Yes	Use same samples for building the subspace and for surrogate construction (not recommended).
Covariance Directory:	/path/to/covariance/files
END	End tag for the sequence.
Input File:	Name of DAKOTA input file with the tags
Surrogate Data File:	The name of the file to store the surrogate construction data in it (this will be generated via ROMUSE).
Surrogate Challenge Data File:	The name of the file to store the surrogate challenge data in it (this will be generated via ROMUSE).
Observations File:	The name of the file to store the measurements and their uncertainties in it (in case the users wants to generate these measurements via ROMUSE).
Dakota Command:	The Dakota command

Example Input Card (1):

This example builds a surrogate to parameterize SCALE6.1 neutronics model (NEWT) [8] in terms of the fission cross section of U^{235}. The surrogate is then used to perform uncertainty quantification via Monte Carlo sampling and compared to the performance of the linearly estimated uncertainty (via SCALE6.1 SAMS module) [8]. *Corresponding Example: Manual/Examples/SCALE/DAKOTA/SURR-UQ.*

In this example ROMUSE will be used to prepare a full surrogate based uncertainty quantification study for a nuclear fuel pin cell problem simulated via SCALE6.1. First ROMUSE will generate 100 samples of the input parameters to be used in later sequences. Each perturbed case is then run via SEQUENCE=ROMUSE-EXE sequence. Once the cases are run, the responses are collected via the SEQUENCE=ROMUSE-RESPONSES sequence. Finally, SEQUENCE=ROMUSE-DAKOTA will construct the full DAKOTA case. A polynomial surrogate is used in the form represented by Eq.(1). DAKOTA estimates the coefficients (c_i) based on the given surrogate construction data and the challenge data used to validate the surrogate model.

$$\hat{f}(\mathbf{x}) = c_0 + \sum_{i=1}^{n} c_i x_i + \sum_{i=1}^{n}\sum_{j\geq i}^{n} c_{ij} x_i x_j + \sum_{i=1}^{n}\sum_{j\geq i}^{n}\sum_{k\geq j}^{n} c_{ij} x_i x_j x_k \,. \tag{1}$$

Different orders of the polynomial surrogates require different minimum number of construction data points:

$$n_{c_{linear}} = n + 1. \tag{2}$$

$$n_{c_{quad}} = \frac{(n+1)(n+2)}{2} \tag{3}$$

$$n_{c_{cubic}} = \frac{(n^3 + 6n^2 + 11n + 6)}{6} \tag{4}$$

where n is the number of parameters and n_c is the number of required points. n_c can grow dramatically as n and the order of the polynomial increase. For more information on the available surrogate types refer to Ref.[5]. Since ROMUSE has the capability to perform dimensionality reduction, this example utilizes the gradient free range finding algorithm to find the basis of the active subspace in the uncertainty sources space so as to satisfy a provided error criteria. Figure 11 shows that for the error upper bound for this example.

This example uses a nuclear fuel pin cell model simulated via SCALE and the goal is to determine the uncertainty in the multiplication factor (k_{eff}) due to the fission cross-section of Uranium - U^{235} isotope. Figure 10 shows a representation of the 2-dimensional geometry of the fuel pin cell. At the center (or the red region) is the fuel bullet which contains the uncertainty source in this example surrounded by a Helium gap (He) which is region green region. The blue region is the zirc4 alloy cladding, and finally the yellow region represents the borated water (H_2O and B) the water serves as a coolant to extract the heat and as a moderator that slows down the neutrons such that they have a high probability causing the fission of U235 nuclei. Boron is added to the coolant such that it absorb excess neutrons that might lead to super-criticality ($k_{eff} > 1{:}0$) in the core system. In this example, a single fuel pin cell is extracted from the reactor core model therefore the reference keff is 1.172437. Since this example uses the 44 group library and considers the uncertainty due to one cross-section type of one isotope then the total number of parameters is 1x1x44 = 44 energy dependent parameters.

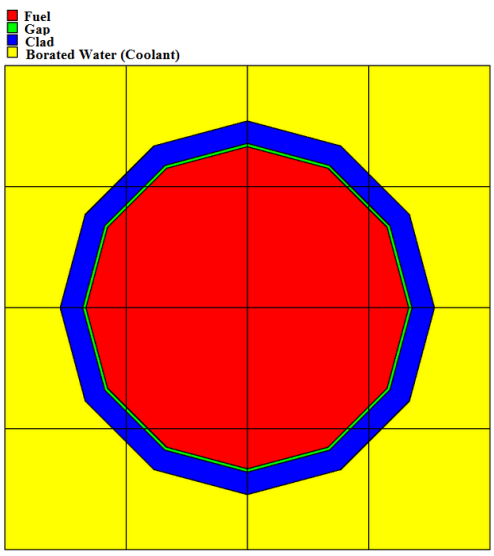

FIGURE 10. NUCLEAR FUEL PIN CELL MODEL.

First, samples are generated (100 samples), ROMUSE-EXE sequence will run the corresponding cases, and ROMUSE-RESPONSES will collect the multiplication factor estimated via the perturbed cases. Finally, ROMUSE-DAKOTA will send the proper commands to build the subspace and pick the proper subspace for the provided error upper bound tolerance (in this case 10^{-2} or 1% of error upper bound). Figure 11 shows that for the error upper bound for this example. The term of the error upper bound and how it ROMUSE calculates it can be found in Ref. [3]. Figure 12 shows the ROMUSE input (*romuse.in*). The tagged DAKOTA input file must be provided in the same directory as that of *romuse.in.* This file will instruct ROMUSE on the DAKOTA method and how to construct the DAKOTA case (refer to Figure 13). After running this case ROMUSE will generate a folder named **DAKOTA-CASE,** which will have a full DAKOTA case ready to run by the user or the user can add another ROMUSE-EXE sequence to run the DAKOTA case directly from inside ROMUSE; however, it is recommended to check the case before running it.

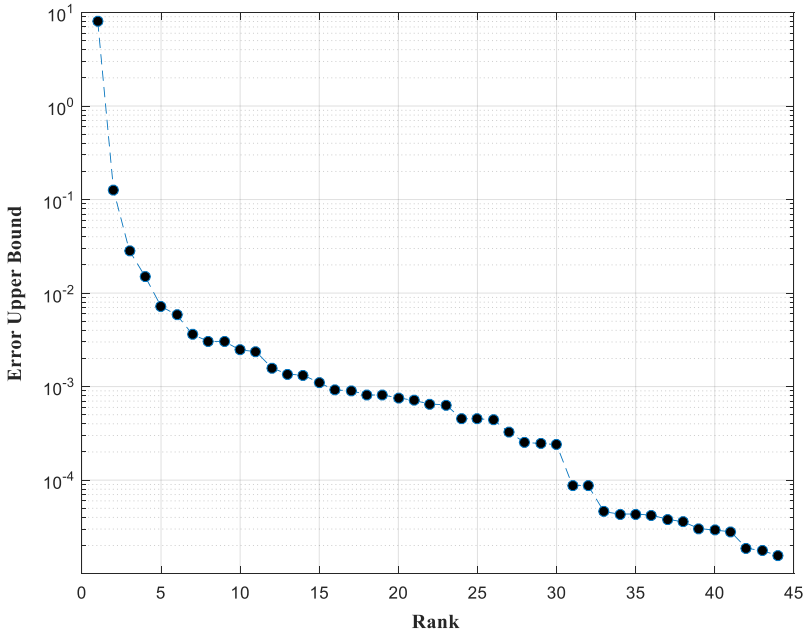

FIGURE 11. ERROR UPPER BOUND VS. RANK.

```
SEQUENCE=SCALE-ROMUSE-UQ
Ref:
Ref-Case
Max Model Runs:
100
SCALE-PoI:
Materials:
1
92235
Reactions:
1
18
Library Name:
ft04f001
Covariance Directory:
/Path/to/Covariance
END
SEQUENCE=ROMUSE-EXE
Command:
/Path/to/SCALE/Command/batch6.1
Queue Names:
1
1
gnep 92
END
SEQUENCE=ROMUSE-RESPONSES
Simulator Out:
input.out
k-eff =
Flags:
Single Physics=YES
END
SEQUENCE=ROMUSE-DAKOTA
Input File:
dakota_romuse_scale.in
Surrogate Data File:
surrogate_build_pts.dat
Surrogate Challenge Data File:
surrogate_build_pts_ff.dat
Surrogate Type:
P-01
Number of Validation Points:
10
Max Number of Active Subspace Construction Points:
44
Subspace Approximation Error Tolerance:
0.01
Recycle Samples=Yes
Covariance Directory:
/Path/to/Covariance
END
```

FIGURE 12. ROMUSE-DAKOTA EXAMPLE INPUT CARD.

```
environment
      tabular_graphics_data
      method_pointer = 'UQ'
method,
    id_method = 'UQ'
    model_pointer = 'SURR'
    output verbose
    sampling
    sample_type lhs
    samples = 500
    seed = 5034
model,
    id_model = 'SURR'
    surrogate global,
    polynomial quadratic
    import_points_file = 'surrogate_build_pts.dat' freeform
    challenge_points_file = 'surrogate_build_pts_ff.dat' freeform
    metrics 'rsquared'  'root_mean_squared'
      Press
variables,
  normal_uncertain = $np
    means = $m
    std_deviations = $std
    descriptors = $p
responses,
    response_functions = $tnr
    no_gradients
    no_hessians
```

FIGURE 13. TAGGED DAKOTA INPUT FILE.

As shown in Figure 11, rank 5 achieves the error upper bound required. Once the basis are calculated they can be used to construct different surrogate types: linear (P-01), quadratic (P-02), cubic (P-03), and Gaussian process surfpack. Table 7 compares the performance of the different surrogates with the linearly estimated uncertainties (sandwich equation) [3]. Table 8 summarizes the error analysis results as reported by DAKOTA output file for the 2nd order polynomial and the Gaussian process surrogate. The leave-one-out cross-validation RMSPE and validation RMSPE (calculated on the challenge samples) are discussed in Ref. [5]. The RMSPE values can be compared to the observed range in calculated k_{eff} values, which is approximately 1240 pcm.

TABLE 7. COMPARISON OF UNCERTAINTY ESTIMATED VIA DIFFERENT SURROGATE TYPES.

Method	**Standard Deviation (*pcm*)**
ROMUSE – linear	149
ROMUSE – non-linear	155
ROMUSE-Gaussian Process	166
Sandwich Equation	133

TABLE 8. SURROGATE ERROR ANALYSIS AS REPORTED BY DAKOTA.

Surrogate Type	Cross-Validation RMSPE (pcm)	R-Squared Goodness of Fit	Validation RMSPE (pcm)
P-02	141	92%	17
Gaussian Process Surfpack	290	88%	56

Acknowledgment

The authors would like to express their deep thanks and gratitude towards North Carolina State High Performance Computing (NCSU-HPC) for providing the required computer allocation. Moreover, the authors would like to deeply thank the Consortium of Advanced Simulation of Light Water Reactors (CASL) for supporting this effort.

Special thanks goes to Los Alamos National Lab (CCS-6) group for their endless support and although an author the contribution and consultant of Dr. Paul J. Turinsky has a vital effect on this work which is hardly expressed by authorship only.

References

1. Williams, M. L., Wiarda, D., Arbanas, G., & Broadhead, B. L. (2009). Scale nuclear data covariance library. ORNL/TM-2005/39, Version, 6.
2. Turner, J. A. (2013). *Virtual Environment for Reactor Applications (VERA): Snapshot 3.1*. CASLU-2013-0164-000, Oak Ridge National Laboratory.
3. Khuwaileh, B. A. A. (2015). Scalable Methods for Uncertainty Quantification, Data Assimilation and Target Accuracy Assessment for Multi-Physics Advanced Simulation of Light Water Reactors. PhD Dissertation, North Carolina State University.
4. Smith, R. C. (2013). *Uncertainty Quantification: Theory, Implementation, and Applications* (Vol. 12). SIAM.
5. Adams, B. M., Bohnhoff, W. J., Dalbey, K. R., Eddy, J. P., Eldred, M. S., Gay, D. M., ... & Swiler, L. P. (2009). DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimization,

Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 5.0 User's Manual. *Sandia National Laboratories, Tech. Rep. SAND2010-2183*.

6. Kochunas, B., Collins, B., Jabaay, D., Downar, T. J., & Martin, W. R. (2013). *Overview of development and design of MPACT: Michigan parallel characteristics transport code*. American Nuclear Society, 555 North Kensington Avenue, La Grange Park, IL 60526 (United States).
7. Marshall, W. J., & Rearden, B. T. (2011). Criticality Safety Validation of Scale 6.1. *ORNL/TM-2011/450*.
8. Scale (2011): A Comprehensive Modeling and Simulation Suite for Nuclear Safety Analysis and Design, ORNL/ TM-2005/39, Version 6.1, Oak Ridge National Laboratory, Oak Ridge, Tennessee. Available from Scale: A Comprehensive Modeling and Simulation Suite for Nuclear Safety Analysis.
9. Jessee, M. A. (2008). Cross-section adjustment techniques for BWR adaptive simulation. ProQuest.
10. Koranne, S. (2011). Hierarchical data format 5: HDF5. In *Handbook of Open Source Tools* (pp. 191-200). Springer US.