NUREG/IA-0407

International Agreement Report

Proposal for the Development and Implementation of an Uncertainty and Sensitivity Analysis Module in SNAP

Prepared by:
Prof. Dr. Rafael Macián-Juan, Dr. Wolfgang Tietsch*, Dr. Felix Sassen*

Lehrstuhl für Nukleartechnik
Technische Universität München
Boltzmannstr. 15
85747 Garching, Germany

*Westinghouse Electric Germany GmbH
Dudenstrasse 44
68167 Mannheim, Germany

A. Calvo, NRC Project Manager

Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555-0001

Manuscript Completed: September 2011
Date Published: April 2012

Prepared as part of
The Agreement on Research Participation and Technical Exchange
Under the Thermal-Hydraulic Code Applications and Maintenance Program (CAMP)

Published by
U.S. Nuclear Regulatory Commission

AVAILABILITY OF REFERENCE MATERIALS IN NRC PUBLICATIONS

NRC Reference Material

As of November 1999, you may electronically access NUREG-series publications and other NRC records at NRC's Public Electronic Reading Room at http://www.nrc.gov/reading-rm.html. Publicly released records include, to name a few, NUREG-series publications; *Federal Register* notices; applicant, licensee, and vendor documents and correspondence; NRC correspondence and internal memoranda; bulletins and information notices; inspection and investigative reports; licensee event reports; and Commission papers and their attachments.

NRC publications in the NUREG series, NRC regulations, and *Title 10, Energy*, in the Code of *Federal Regulations* may also be purchased from one of these two sources.

1. The Superintendent of Documents
   U.S. Government Printing Office
   Mail Stop SSOP
   Washington, DC 20402–0001
   Internet: bookstore.gpo.gov
   Telephone: 202-512-1800
   Fax: 202-512-2250
2. The National Technical Information Service
   Springfield, VA 22161–0002
   www.ntis.gov
   1–800–553–6847 or, locally, 703–605–6000

A single copy of each NRC draft report for comment is available free, to the extent of supply, upon written request as follows:

Address: U.S. Nuclear Regulatory Commission
Office of Administration
Publications Branch
Washington, DC 20555-0001
E-mail: DISTRIBUTION.RESOURCE@NRC.GOV
Facsimile: 301–415–2289

Some publications in the NUREG series that are posted at NRC's Web site address http://www.nrc.gov/reading-rm/doc-collections/nuregs are updated periodically and may differ from the last printed version. Although references to material found on a Web site bear the date the material was accessed, the material available on the date cited may subsequently be removed from the site.

Non-NRC Reference Material

Documents available from public and special technical libraries include all open literature items, such as books, journal articles, and transactions, *Federal Register* notices, Federal and State legislation, and congressional reports. Such documents as theses, dissertations, foreign reports and translations, and non-NRC conference proceedings may be purchased from their sponsoring organization.

Copies of industry codes and standards used in a substantive manner in the NRC regulatory process are maintained at—

The NRC Technical Library
Two White Flint North
11545 Rockville Pike
Rockville, MD 20852–2738

These standards are available in the library for reference use by the public. Codes and standards are usually copyrighted and may be purchased from the originating organization or, if they are American National Standards, from—

American National Standards Institute
11 West 42nd Street
New York, NY 10036–8002
www.ansi.org
212–642–4900

Legally binding regulatory requirements are stated only in laws; NRC regulations; licenses, including technical specifications; or orders, not in NUREG-series publications. The views expressed in contractor-prepared publications in this series are not necessarily those of the NRC.

The NUREG series comprises (1) technical and administrative reports and books prepared by the staff (NUREG–XXXX) or agency contractors (NUREG/CR–XXXX), (2) proceedings of conferences (NUREG/CP–XXXX), (3) reports resulting from international agreements (NUREG/IA–XXXX), (4) brochures (NUREG/BR–XXXX), and (5) compilations of legal decisions and orders of the Commission and Atomic and Safety Licensing Boards and of Directors' decisions under Section 2.206 of NRC's regulations (NUREG–0750).

DISCLAIMER: This report was prepared under an international cooperative agreement for the exchange of technical information. Neither the U.S. Government nor any agency thereof, nor any employee, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for any third party's use, or the results of such use, of any information, apparatus, product or process disclosed in this publication, or represents that its use by such third party would not infringe privately owned rights.

NUREG/IA-0407

International Agreement Report

Proposal for the Development and Implementation of an Uncertainty and Sensitivity Analysis Module in SNAP

Prepared by:
Prof. Dr. Rafael Macián-Juan, Dr. Wolfgang Tietsch*, Dr. Felix Sassen*

Lehrstuhl für Nukleartechnik
Technische Universität München
Boltzmannstr. 15
85747 Garching, Germany

*Westinghouse Electric Germany GmbH
Dudenstrasse 44
68167 Mannheim, Germany

A. Calvo, NRC Project Manager

Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555-0001

Manuscript Completed: September 2011
Date Published: April 2012

Prepared as part of
The Agreement on Research Participation and Technical Exchange
Under the Thermal-Hydraulic Code Applications and Maintenance Program (CAMP)

Published by
U.S. Nuclear Regulatory Commission

ABSTRACT

This report presents a proposal for the implementation of a module in the SNAP program that permits uncertainty propagation in the codes which it supports. In particular, as a first application, it suggests that the module be developed and tested with TRACE. Once the functionality and applicability have been tested, an extension to the other codes could be carried out with relatively small additional effort. The module in SNAP should allow for both propagating uncertainty from model input variables and physics models implemented in the code. Furthermore it should be capable to support sensitivity analysis with respect to model input parameters.

The report is structured in several sections that give an overview of the most commonly used uncertainty propagation methodologies today. The review performed leads to the conclusion that the most promising methodology for implementation in SNAP is, in the opinion of the authors, the one based on a statistical computational framework. For this reason, the report is focused on the proposal of an uncertainty module for SNAP that can successfully and efficiently carry out uncertainty propagation for the codes that it supports based on the statistical methodology developed by LANL and GRS.

CONTENTS

FIGURES

1. HISTORY

Considerable progress has been made in recent years with the aim of improving the quality and depth of the safety analysis of nuclear systems. This progress has been reflected in the development of increasingly more sophisticated analytical tools for the simulation of the behavior of nuclear systems, which has led to a better understanding of the physical phenomena that govern this behavior and their interactions. The result has been a better definition of the safety limits applied to nuclear power plant (NPP) operation.

There has been, in particular, a move away from over-conservatism in safety analysis towards the application of so-called best estimate methodologies. The traditional conservative approaches are based on excessively severe assumptions with respect to the models used and to the applied system´s boundary and initial conditions. Furthermore, the results obtained are interpreted with additional safety margins, which commonly yield design solutions far removed from the actual safety limits and, hence, resulting in reduced plant operation performance. Best estimate calculations employ more accurate physical models, numerical solution procedures and close to actual system boundary conditions and, as a result, offer more realistic plant behavior descriptions to decision makers.

The application of best estimate safety analysis has opened new prospects but also raised new problems that need to be addressed. Thus, considering more realistic evaluations makes it crucial to assess their degree of reliability, especially when, for instance, they have to be compared against safety limits that must not be crossed. For this reason, it is necessary to identify and quantify all the sources of possible uncertainties that could affect the reliability of the results, and then apply mathematically sound and justifiable methodologies that can propagate them through the nuclear safety analysis procedure to the predictions of the systems variables of interest (the so-called outputs of the analysis).

A series of such uncertainty propagation methodologies have been developed and tested in the last 20 years, and most of them have proved their value by meeting some of the requirements that any methodology must fulfill. However, there is not yet a consensus as to which methodology is the best to satisfy all the needs that a complete uncertainty propagation analysis demands. Nevertheless, two main approaches have been adopted by a large number of researchers and users of these methodologies, namely, one based on statistical propagation and processing, and another based on the application, when computationally feasible, of *variational* methods that quantify analytical sensitivities, usually first order, of the physical-numerical model of a nuclear system, and combine them with the uncertainty measures of models and boundary conditions to render the uncertainty of the output variables of interest.

2. DEFINITIONS

Uncertainty:Lack of certainty. A state of having limited knowledge, in which it is not possible to exactly describe an existing state or future outcome.

Index dependent uncertainty: uncertainty quantification of a variable represented as a function of another variable, referred to as the index, e.g. Pressure vs. time (index), temperature vs. pressure (index). These variables may or may not have an actual dependency on each other.

Scalar uncertainty: uncertainty of a variable that is not dependent on another one. This can also be applied to the case in which the variable is a function of another, but the uncertainty is calculated only for a specific value of the index variable. In the actual application of the methodology described in this report, an index-dependent uncertainty quantification is in reality a series of scalar uncertainty determinations for a consecutive set of values of the index variable.

Sensitivity: the quantification of the dependence of a variable on another. In the context of uncertainty and sensitivity analysis, sensitivity is understood as the quantification of the degree of dependency of an output variable of interest on a given input variable. Several measures are possible for this quantification, e.g. first or higher order derivatives (Jacobian Matrices), statistical correlation coefficients, etc.

Code: a code is a computer program used for the simulation of nuclear system behavior. In the context of the uncertainty propagation methodology, the code is also considered as a deterministic function, which input with stochastic information, will produce a stochastic output. The stochastic input information is represented by the sample on input and model values, and the stochastic output information by the sample of code outcomes resulting from executing the code simulation a number of times equal to the size of the sample.

PDF: Probabilistic Density Function. A function that describes the density of probability associated to the distribution of a stochastic variable. In the context of uncertainty analysis, the PDFs are one of the possible measures of the uncertainty of input and models variables, as well as of the code´s output variables of interest, when the sample is large enough to yield a statistically significant empirical PDF (based on the statistical processing of the sample data).

Dependency: the influence that the value taken by a given variable has on the value taken by the dependent variable. Dependency is measured by the quantification of sensitivity.

3. REVIEW OF THE INTERNATIONAL STATE OF THE ART

3.1 Uncertainties in Best Estimate Analysis

The application of Best Estimate codes to determine safety margins for reactor operation and accident conditions and to develop Emergency Response Guidelines for accident management prompted the development of several methodologies to quantify the uncertainty associated with the code predictions (Wickett 1998).

Before a Best Estimate code can be confidently applied to safety analysis, three important issues should be addressed (Boyack, 1990). First, it is necessary to demonstrate the capability of the code models and solution procedure to simulate the physical processes expected during the scenario of interest. Second, the scalability of the results produced by physical models, usually developed to predict phenomena in separate effect test and scaled integral facilities, to a full size nuclear power plant (NPP) should be determined. And third, the role of the uncertainty of the parameters that describe the NPP on the uncertainty of the code results, ie boundary and initial conditions, should be evaluated.

However, an additional fourth step is also needed. The predictive capability of the code physical models and experimental correlations is neither perfect nor valid for all applications. For this reason, validation and assessment of the code models can provide useful information about their performance under a wide range of physical conditions. Such information can then be quantified as an uncertainty associated with the code physical models, and then propagated to the code results by means of appropriate techniques.

Several sources of uncertainty affecting Best Estimate results are related to different aspects of the solution procedure. One important source of uncertainty is the experimental correlations at the heart of many of the code models. They are usually adapted to best reproduce the sets of data on which their derivation is based. However, they may not perform so well when required to predict other data from different experiments, although they may be investigating similar physical phenomena but with different boundary conditions. Moreover, the correlations and models may be used to predict phenomena outside the range of conditions for which the correlation or model was originally developed. In such situations, a degree of uncertainty should be expected in the predictions.

Another source of code uncertainty is related to the fact that many of the code physical models and correlations have been assessed and validated against integral and separate effect test facilities. Many of the integral and most of the separate effect test facilities are scaled down with respect to full size NPPs. Thus, a good performance by the models in these assessments is no guarantee that they will do as well when applied to full scale NPPs, since different physical phenomena may dominate the transients' development at different scales. For example, one-dimensional flow assumption may be appropriate at small scale, but not at full scale.

Finally, simplifications, assumptions and nodalization schemes may affect the behavior of the code physical models and the solution procedures. For instance, a coarse geometric representation of a NPP may miss local important local effects that could control a transient's behavior. In addition, the initial state of the NPP may not be fully known, and the value of some important parameters or boundary conditions may be known only approximately. The results calculated by the code under such conditions necessarily reflect this uncertainty.

3.2 Sources of uncertainty

Uncertainty is the lack of certainty owing to inadequate or lack of relevant information. Two types of basic uncertainty have been identified: the so-called stochastic or statistical, and the epistemic or systematic. Both types of uncertainty may manifest themselves as uncertainty in input parameters or as uncertainty associated with physical models in the code.

When referring to a model used for the simulation of the behavior of a real system, which is the purpose of nuclear safety analysis, statistical uncertainty is a result of the random variability of the values of the variables that describe the system and its boundary conditions, also known as input variables to the model, and of the lack of precision of the code´s physical models. If we assume that they may have a stochastic nature, then statistical functions can be used to quantify this uncertainty. Such an assumption, which can be easily justified in the case of input variables obtained from measured data, is more difficult to justify for code models. In this situation, the assumption of adequacy of the model must also be made, that is, that it represents the best approach possible to describe a given physical process, and the "errors" when its predictions are compared to experimental results are produced by stochastic variability of the experimental data and by the effect that this variability in the data used to develop the model had in the determination of the important parameters of the model.

The epistemic uncertainty is due to lack of knowledge, whether realized or not by the analyst's two sources. For instance, the real degree of variability of the input variables is not completely known, and an approximation is necessary, or that the physical models lack the description of some process or mechanism with more or less relevance in the quality of the predictions of the model. Epistemic uncertainty is difficult to quantify, precisely because of the lack of realization of what are its sources. For this reason, in practice, it tends to be included as an unknown factor in the statistical uncertainty, and the statistical measures employed to quantify the total uncertainty are applied without making any distinction between them.

However, since in real life applications both types of uncertainties are present in the systems analyzed, there is always a trend towards trying to reduce epistemic uncertainty as much as possible and to convert it to statistical one. This can usually be achieved by attempting to gain better knowledge of the system, process or mechanisms, to improve the predictive capability of the models, or by improved analysis of measured data from a wider range of sources that can provide more accurate information about the true ranges of variability of important system variables. Better experimental data, or more relevant for the physical processes described by the models will improve our knowledge and reduce epistemic uncertainty. In general an increase of information will reduce this kind of uncertainty and turn it into the more easily quantifiable statistical type.

In practical terms, when the uncertainty has been considered as being quantifiable by statistical measures, then the sources affecting the results of the analysis of a system can be identified as:

1. The physical model and mathematical structure, i.e., how accurately does a mathematical and physical model describe the true system for the transient of interest.

2. The numerical approximation, i.e., how appropriately a numerical method can solve the physical-mathematical model of the system: dimensionality and numerical accuracy play here a very important role.

3. The initial / boundary conditions, i.e., how precise is the data / information for initial and / or boundary conditions,

4. The data used to describe the geometry and structure of the system, the so-called input variables or parameters.

5. The degree of skill and experience of the analysis that will determine the accuracy, degree of detail and adequacy of the system model developed for the analysis, as well as the choice of relevant boundary conditions, and for some codes, the selection of the most appropriate physical modes or model parameters.

Sources 1 and 2 are difficult to quantify in a typical uncertainty propagation analysis, although assessment of experimental data with different types of physical models and mathematical descriptions can shed some light on this problem. It is however, much more difficult to ascertain than 3 and 4. For this reason one of the initial steps in the CSAU methodology (Boyak 1990), which has been also adopted by many others, is the determination of the adequacy of a given code to simulate the behavior of interest. Once this is done, basically out of experience and expert opinion, one does not include this source of uncertainty in the process of its quantification. Sources 2 and 3 are the basis for all the methodologies proposed (except for UMAE, see Section 5) and for this reason they are the main subject of this report.

Finally Source 5 is also quite difficult to quantify, and only with the repetition of the simulation by other analyses or by comparison of the results of benchmarks with the same codes, where the differences in the results can sometimes be blamed on the different user approaches to model the same transient with the same code (for similar mathematical, numerical and major physical models). This kind of uncertainty is not amenable to being included in an uncertainty propagation methodology like the one described in this report, although if an estimate can be made, its effects could be introduced as a bias in the relevant output variables.

3.3 Review of available Methodologies

3.3.1 AEA Technology Methodology

The AEA Methodology was developed by AEA Technology in the UK (Wickett 1998). Its main purpose was to quantify the uncertainty associated with Best Estimate calculations by developing a systematic sensitivity study. Success in this endeavor requires the use of a Best Estimate code robust enough to carry out the many calculations needed for the different combinations of input and code model parameters required to obtain an estimate of the code uncertainty.

The scope of this methodology encompasses uncertainty in plant initial conditions, in modeling parameters used as input for the code, in boundary conditions and in the code models and correlations. Once the type of transient scenario to which the methodology is to be applied is defined, expert judgment is used to determine its physical characteristics and to assess the code capability to simulate the transient. If such an evaluation is successful, the calculation of uncertainties can be carried out.

The methodology recognizes two well defined phases from which uncertainty information can be gathered to be used during the analysis. First, during the code assessment phase, code developers usually adjust the code to fit the experimental data from a series of separate and integral test facilities. The ‘fit’ may be considered good when the code predictions, for a range of calculations that cover the known or expected uncertainty in code models and data, bound the experimental results. The ranges of variation of the code models and other relevant variables are usually set based on expert judgment.

The analyst, following his own experience and with information gathered from previous code assessment and validation studies, determines the intervals of variation for all the parameters involved in the uncertainty analysis. These ranges constitute uncertainty statements for the variables and should be in the form of ‘reasonable uncertainty ranges’. They are defined by the methodology as the smallest range of values that a given variable can take, which includes all the values for which there is a reasonable certainty that they are consistent with all the available evidence. In other words, the expert, based on what he know about the system, can determine a reasonable minimum range of variation for it. Reasonable certainty means that justifications can be given to demonstrate the consistency or inconsistency of the choice of the minimum variation range. The important concept is the ‘available evidence’, i.e. what the expert knows. Two cases are then possible:

1. Data-based cases: if the variable whose uncertainty is to be determined can be directly experimentally measured, then statistical confidence limits for its range of variation can be determined, and a reasonable certainty exists that the variable will not have values outside these limits. For example, if the expert knows, from previous assessments or experimental measurements, that a specific input temperature

cannot be less than 100 K or larger than 150 K with a 95% probability, then the statistical confidence limits would be 100 and 150 K, with a 95% confidence level.

2. Data-free cases: many times in thermal-hydraulic research, the variable of interest has not been directly measured in the relevant conditions. In such cases the range has to be subjectively set and reasonably justified. For instance, if the expert of the example above, despite not performing any measurement or assessment can from previous experience, determine that the input temperature may lie between 100 and 150 K.

If the value of a certain variable depends on those of other variables, e.g. through a mathematical formula, combinations of the ranges of variation of these variables may provide the range of variation for the value.

The most important idea is that the ranges of variation of the input variables should only exclude those values inconsistent with all available evidence or experience. If further measurements of the variables lie outside the specified ranges, then the reasoning to support these ranges was wrong, and new ranges of variation should be established.

The second phase takes place after the code's development and assessment phase. The code should then be assessed against an independent data base of experiments to make sure that the code physical models are able to successfully predict the data with uncertainty ranges as narrow as those used during the development phase. The independent data base must be representative of the physical phenomena expected to occur during the transients of interest. If this phase is successful, the code and the information gathered to perform the uncertainty analysis can then be used for plant calculations. If not, then further code development and assessment work is necessary.

The mechanics of the uncertainty analysis entails the execution of the code for the transient and system of interest (e.g. a NPP, a test facility, etc.) a number of times. The analysts must decide this number based, once more, on personal experience according to the number of input variables and code physical models whose uncertainty he/she wants to include. By selecting maximum and minimum values within the ranges of variation of these variables and models, a series of code executions combining the selected values are then performed. The output information can then be analyzed and the uncertainty in the code results quantified in the form of ranges of variation for the output parameters of interest, which should be small enough to support safety decisions.

3.3.2 GRS Methodology

The GRS methodology is based on the use of statistical techniques to calculate sensitivity and uncertainty information from the results of a computer simulation. Input variables, code models, boundary conditions and code solution algorithms are all considered as sources of uncertainty, which is then propagated to the code results.

The statistical techniques and the philosophy of the GRS methodology are based on well-established concepts and tools from probability and statistics (Wickett 1998). The initial steps of the GRS methodology are also shared by most of the other methodologies described in this Section. They involve the selection of a transient scenario for a NPP and a Best Estimate code to carry out the analysis.

The next step identifies the physical processes controlling the evolution of the transient in order to determine whether the code selected for the analysis is able to simulate them. This is done by reviewing the code documentation and manuals and by assessing the capability of the code to simulate transient scenarios of a similar type based on existing code validation and assessment studies.

After the applicability of the code has been demonstrated, the next step entails the assignment of the so-called Subjective Probability Density Functions (SPDF) to quantify the uncertainty of the input variables and the code models. The selection of these functions is the most subjective part of the GRS methodology. The SPDFs reflect how well the uncertainty in input parameters and code models is known. This knowledge may range from detailed data on mean values and other statistical parameters to a simple notion of the expected range of variation of a certain parameter, i.e. maximum and minimum values. In addition, dependencies between parameters and models must also be accounted for during this phase, by means of appropriate correlations or joint probability distributions, provided they are known or can be reasonably derived.

The rest of the development of the methodology is conditioned by its ultimate goal of quantifying the uncertainty in the results of the code simulations of NPP transients. To this effect, the GRS methodology makes use of experimental data from one (or several) integral test experiments similar to the NPP transient of interest. Uncertainty information from the code results is obtained by applying statistical techniques that yield statistically based uncertainty limits. A series of acceptance criteria determine whether the code can describe the type of scenario analyzed and the SPDFs give reasonable uncertainty information for the output. If this is the case, the uncertainty analysis methodology is ready to be applied to the analysis of a similar scenario in an actual NPP.

The statistical part of the GRS methodology is objective and based on probability theory and statistical methods. The process starts with the generation of a random sample of the input parameters and of the parameters that quantify the uncertainty in the code models, according to the SPDFs assigned to them. After executing the code for all the elements of the sample, relevant statistically based uncertainty and sensitivity information can be extracted from the code results. The minimum number of code calculations is given by the Wilks' formula (Wilks 1962) according to the degree of precision desired for the uncertainty measures. Thus, the number of required calculations does not depend on the number of input and code model parameters or on any assumption about the probability distribution of the results (Glaeser 1994).

During the assessment phase, the uncertainty information computed from the analysis of the code results is usually in the form of uncertainty limits, which are then compared with the measured integral test data. The purpose is to see whether the data falls into the calculated intervals. If this is the case, the uncertainty information for input and code model parameters (SPDFs) and the statistical procedure (number of code runs, sensitivity measures, etc.) can be used for plant calculations. If the experimental data are not bound by the uncertainty intervals, the uncertainties of the input and code model parameters need to be revised. The use of full scale separate effect tests and relevant plant data when available, can be used to refine the SPDFs and to account for scaling effects and plant specific uncertainties.

The statistical techniques and procedures used by the GRS methodology have also been employed in the methodologies developed at Empresa Nacional del Uranio (ENUSA) and at Institut de Protection et Sureté Nuclearire (IPSN) (Wickett 1998).

3.3.3 Methods based on Sensitivity Analysis. Adjoint Methods

Sensitivity Analysis studies the relationship between the response of a mathematical model or computer code and the variables that determine this response. Conceptually, sensitivity is a measure of the effect that a certain code input or core physical model has on the code's response. It can be defined locally, i.e. for a specific value of the variable, or globally, i.e. for a range of variation of the variable. Methods based on local sensitivity analysis define sensitivity as the first order derivative of the code's response with respect the variable of interest. The derivative is calculated at selected points within the range of variation of the variable, e.g. at its nominal value, and gives a measure of the intensity of the relationship between the response and the variable. Mathematically this translates to how much the response will change for a unit change in the value of the variable. By definition, sensitivity is a linear measure, and if the code's response is significantly non-linear, then higher order sensitivities (higher order derivatives) may be used.

The calculation of uncertainties from local sensitivity data is carried out by using the covariance-variance matrix of the input variables. This matrix contains the uncertainties for each variable in the form of statistical variances (diagonal elements of the matrix), and the joint uncertainties for pairs of variables in the form of statistical covariances (non-diagonal terms of the matrix). The covariances are a measure of joint uncertainty for two variables which are dependent on each other. If they are independent, then their covariances are zero, since changes in the values of one of them do not affect the values the other variable. For example the steam generator level and the time it takes for the pressurizer relief valve to open after receiving the opening signal have clearly a covariance of zero, since their values are in no way related.

The application of local sensitivity methods has several advantages. For instance, it can determine a quantitative value of the importance of a given variable with respect to the code's response, and it can do this in a simple and systematic way. Moreover, any changes in the uncertainties of the input variables and code models are easily transformed into new uncertainty estimations of the code's response once the covariance-variance matrix has been obtained.

The analytical nature of the methodology does not require multiple code calculations, which is one of the disadvantages of the methodologies based on a statistical approach, e.g. GRS and CSAU.

However, this technique also has limitations. In general, it is difficult to develop the equations needed to obtain the sensitivities, especially for complex non-linear physical models and numerical solution procedures, such as those used in modern thermal-hydraulic system codes. One approach to compute the sensitivities is to do it 'numerically', that is, by calculating changes in the code's response due to changes in the input variables and code models. If the number of variables is large, however, other analytical techniques, such as the Adjoint and Forward methods (Cacuci 1988) must then be applied. In particular, for analyses that involve a large number of input variables and code models, Adjoint methods are usually very efficient. These methods compute the adjoint matrix of the system of differential equations that describe the flow, and apply it to the variance-covariance matrix of the selected variables.

In general, the application of sensitivity analysis to non-linear models is limited. Only the Adjoint approach can, in certain cases, be extended to non-linear problems. A variety of Adjoint methods have been developed as part of perturbation and variational theory for reactor calculations, and they have been successfully applied to sensitivity analysis of several important linear problems of reactor theory (Cacuci 1988). Although the Adjoint methods have been extended to deal with non-linear models, there are still questions about the applicability of this approach to the sensitivity analysis of complex thermal-hydraulic models (Ronen 1988). In particular, thermal-hydraulic problems involving discontinuous functions of state are especially challenging. In an effort to overcome these limitations, Cacuci (Cacuci 1988) has developed a sensitivity theory for non-linear systems. The application of this theory identifies the maximum, minimum and 'saddle points' (unstable equilibrium point) within the code's response state space (defined by the values the code's response can take) for different combinations of input variables and code models. So far, only a very limited application to a system code, CATHARE, has been reported (Barre 1995).

Another important limitation is the need to develop a complete variance-covariance matrix of the input variables and code models. While the uncertainties of the variables may be generally known with reasonable effort, the covariances, which represent the joint uncertainty between two given variables when their values are dependent on each other, are not easily available.

3.3.4 CSAU Methodology (Code Scaling, Applicability and Uncertainty)

This methodology was developed by the US-NRC and its contractors with the aim of evaluating uncertainties in the results of the Best Estimate codes used to analyze transient scenarios in NPPs (Boyack et al. 1990). The methodology creates a unified and systematic procedure to assess the applicability and scaling capability of the Best Estimate codes to perform safety calculations for NPPs. It also aims at qualifying the code results with the uncertainty derived from input variables, noding schemes used to model the NPPs, and other contributors such as biases in the code models and in the assumptions of the studies.

The main objective of the methodology is to develop a procedure for combining quantitative analyses and expert opinion in order to quantify the uncertainty associated with the code results.

The method is auditable, traceable, and practical, and provides a systematic and comprehensive approach to:

- defining the transient scenario and determining the important physical phenomena,
- evaluating code applicability based on its ability to model the physical phenomena,
- assessing code scale-up capabilities, and
- quantifying code uncertainties resulting from code accuracy, code scalability, and plant state and operating conditions.

The CSAU evaluation methodology consists of three primary elements, namely the identification of physical processes important for the transient scenario of interest and the determination of the code's ability to simulate them, the identification, ranking and determination of the range of variation of code input and code model parameters, and, finally, sensitivity and uncertainty calculations.

The first step begins with the selection of the NPP and scenario of interest. Then, the physical processes expected to occur must be identified in order to determine the suitability of a frozen version of the code (no modifications allowed) to model them.

The characteristics of the second and third steps are determined by the scope of the methodology and the technique used to calculate the uncertainty in the code results. The strategy followed by CSAU aims at qualifying the code for its generic application to a certain type of accident scenarios in nuclear power plants (NPPs). In this context, the code accuracy and scale-up capability need to be assessed against the results of integral tests addressing scenarios similar to the one under study. The comparison of the code predictions against the results of these tests can provide a measure of accuracy and biases for the prediction of the important parameters that define the scenario, e.g., maximum temperatures, maximum pressure, etc., can be established. The determination of the code accuracy and scaling capability yields specific numerical values for the code results, e.g. a temperature increment or decrement that can then be added to the calculated code's uncertainties when applied to the scenario of interest in a NPP. For instance, for maximum Peak Clad Temperature (PCT) best estimate value T_{PCT}, and an uncertainty of $\pm T_u$ degrees, the additional bias, T_{bias}, would give a total maximum PCT of $T_{max\_PCT} = T_{PCT} + T_u + T_{bias}$.

The second step involves the assessment of code uncertainties and biases and the ranking of the input variables and code models selected in the first step by their importance with respect to the main physical processes controlling the transient. In addition, during the second phase of the methodology a computer model of the facility under study is developed and the noding scheme established. This scheme should have enough detail to capture all the important phenomena expected and should be as close as possible to that used for the code assessment against the database of integral tests mentioned above.

From all the information gathered in previous steps, the analyst determines the range of variation of the main parameters selected in step 1 and will calculate the biases to account for the code accuracy and scaling capabilities. In a general sense, the basic premise of this approach rests on the need to qualify the code in an 'integral sense', that is, as an intertwined set of physical models, solution procedures and input and boundary conditions interacting to yield predictions for the transient of interest. The technique for the quantification of the code overall uncertainty and biases is left to the analyst to decide.

The third and last step of the CSAU methodology deals with the calculation of the uncertainty in the code results resulting from the uncertainties in the input variables and code models, and then with the combination of this information with the biases and uncertainties obtained from the previous step.

CSAU does not specify which methods should be used by the analyst to carry out steps two and three. For the test application of the methodology, the development team chose to use the technique based on response surfaces sampled by a MonteCarlo method. This calculates the output uncertainty information based on a large number of calculations with the response surface instead of the actual code (Lellouche 1990). A combination of randomly selected values for the input variables and code models constitutes one trial calculation. After many trials, uncertainty information can be extracted from a statistical analysis of the results.

Finally, the quantification of biases based on the determination of the code accuracy and expert judgment introduced in step two (Wulff 1990) is performed by the simple addition of the biases to the results obtained from the application of the surface response methods in step three (see example above for the PCT.)

3.3.5 UMAE Methodology (Uncertainty Methodology based on Accuracy Extrapolation)

The philosophy behind the UMAE methodology, developed at the University of Pisa, Italy, is somewhat different to that of the previously discussed methods. It assesses the uncertainty in the results of a Best Estimate code by following a so-called 'integral' extrapolation approach. This is in contrast with the statistically and sensitivity based methods presented above which draw uncertainty information from a more deductive approach by propagating the statistical uncertainties in input variables and code models to the code results.

UMAE is based on the notion that the uncertainties in the code results can be quantified by comparing the code predictions against a series of experimental test results; usually from integral test facilities, which gives the method its 'integral' character. The accuracy of such predictions is measured by the use of the Fast Fourier Transform (FFT). When the code is then applied to study a certain transient in a NPP, the measure of the uncertainty associated with the code results is calculated by extrapolation of this accuracy information obtained from validation against integral test facilities that simulate a similar scenario. The extrapolation is carried out by means of certain statistical techniques that provide confidence intervals bounding the results of the analysis.

The uncertainty information the extrapolation process yields refers to the modeling of physical processes by the code during the transient, rather than to specific code models and input variables. The processes are called Relevant Thermal-hydraulic Aspects (RTA). Each RTA, e.g. subcooled blowdown, pump behavior, etc., can be described by a series of single valued parameters, integral parameters, non-dimensional parameters, and parameters related to the time sequence of events. These parameters don't represent code models or input variables, but rather variables or models that define the transient behavior of the RTA in which they play a significant role. For instance, in (Wickett 1998) the authors of UMAE offer a table with the uncertainty values computed for several important parameters and events during a small break LOCA analysis: for the RTA Saturated Blowdown, the relevant variables are the time when primary pressure is 10% of nominal value, and the integral break flow at different times during the saturated blowdown phase.

For this reason, the UMAEA methodology requires the selection of these RTA defining variables of interest before the start of the calculations, since the process of generation of uncertainty information revolves around them. The minimum number of parameters that need to be considered is claimed to be 40 rather arbitrarily. If a 'time-dependent' study of uncertainty is desired, similar to the one offered by the statistically based methodologies, further work is then necessary. For instance, to obtain a time dependent tolerance interval for, say, clad temperature, the whole extrapolation procedure must be made at certain times during the transient in order to provide time varying uncertainty estimation. The authors are not very clear on how this can be done.

Obviously, it is then mandatory that the set of integral test experiments selected for the extrapolation be relevant to the scenario or transient under investigation, whether this takes place in a NPP or in another test facility. They should cover the phenomena and physical processes expected to occur in the transient under study. Therefore, the application of UMAEA requires the use of an extensive, qualified experimental data base from a number of integral experimental facilities. Use of the proposed thermal-hydraulic test matrices by the Committee for the Safety of Nuclear Installations (CSNI) is strongly encouraged, in order to be able to determine those experiments that closely model the scenarios to which UMAE is applied. However, the unavailability of experimental data, because of lack of access or because of the non-existence of relevant experiments, may thwart the application of UMAE. According to the authors, the minimum number of experimental tests necessary for each application can be fixed at three; a number proposed without justification. This is a serious drawback, since it is quite difficult to defend statistically sound uncertainty measures based on a sample of size three.

When the application of UMAE to a full size NPP is the ultimate goal, the problem of scaling can be addressed by including in the experimental database facilities with different scaling ratios. The spread of the comparisons should then provide a certain measure of the uncertainty due to scaling that can be carried to the full size NPP. The main problem with this approach is the possibility of different transient behaviors, because of scaling effects, determining the most influential thermal-hydraulic processes occurring in the different test facilities. It is not clear how the UMAE methodology can handle this problem, whicht would seriously compromise the validity of the extrapolation procedure, since the integral transient results could be determined by different physical mechanisms.

Finally, another important requirement is that the code must be frozen, that is, no modifications to the models are allowed, and a fixed reference noding scheme selected and validated for the study. How this should be done is not clearly stated in [Wickett 1998]. However, it is in principle difficult to agree in a single noding diagram that can be used for a number or test facilities and a full scale NPP model. Smaller facilities may require more refined noding to resolve some of the physical processes during the transients, and such a fine noding may be not be feasible or suitable for much larger size NPPs.

The user of this methodology must be qualified and experienced, since he must select the relevant experimental tests that simulate the transient of interest. To do this, the user must understand completely the transient, and qualitatively and quantitatively evaluate the code results for each comparison with experimental data, since, in many cases, this will not be as good as desired. Moreover, the user has to assess both the effect of scaling and the effect of the differences in the nodalization schemes employed for the computer models of the integral test facilities and for that of the nuclear system under investigation. This is, nevertheless, a general requirement common to all methodologies.

In summary, the main premise behind the UMAE methodology, similarly to the original purpose of the CSAU methodology, is to obtain the uncertainty associated with the code prediction of specific parameters of interest, e.g., Peak Clad Temperature (PCT), time for core uncovery, etc., for specific transients, e.g. SBLOCA, LBLOCA, ATWS, etc. from the analysis of relevant integral tests, and then use this information, usually in the form of confidence bands, to qualify the results of simulations of similar transients in the analysis of actual NPPs.

3.4 Comparison of the Methodologies

In the discussion and description presented above, a common goal of all the methodologies is to provide the information needed to generate quantitative statements about the quality of the code predictions. This is achieved in all cases by assigning to the code results a measure of their uncertainty.

All the methodologies reviewed in this section begin the uncertainty analysis process by selecting a transient scenario in a certain NPP, and a Best Estimate code to simulate the transient. The qualification of the code to successfully perform the analysis is determined by means of a critical review of the code documentation and of the previous code validation relevant to the type of transient scenario selected. The ability of the code physical models and solution procedure to capture most (at least the most important) of the physical processes expected to occur during the transient constitutes the measure to decide on the code suitability.

Once these preliminary, but important, steps have been completed, the next phase of the uncertainty analysis is the one in which most methodologies differ.

Direct sensitivity methodologies, such as the Adjoint Method introduced in Section 2.2.3, are, in principle, more accurate, but they suffer from important limitations for an efficient application. For instance, the difficulty of deriving the Adjoint equations for complex non-linear mathematical models. Moreover, the derivation of uncertainty information from the sensitivities demands knowledge of the variance-covariance matrix of the input and code model parameters (see Section 2.2.3) which is not always available.

Another limitation is their local character, which restricts accuracy to small variations in the vicinity of the nominal values of the variables if the linearity of the methodology is to be maintained. This limits the use of the sensitivity measures to scenarios in which the change in system variables is mildly non-linear. Although a non-linear theory has been developed, its use is still limited to locate specific values (maximum, minimum, saddle points, etc.) for selected code output variable.

The analytical character of this approach may suggest that once the Adjoint equations are derived, the method does not require either user or experimental input. But if the calculation of uncertainty is the ultimate goal, the user and experimental input is contained in the variance-covariance matrix. All the issues faced by the other methodologies regarding code scalability, model adequacy, uncertainties in input variables and code models and how to obtain them as objectively as possible also influence this methodology based on analytical sensitivity analysis.

GRS and AEA methodologies seem to be similar in their approach to calculating uncertainty based on a sample of code results. However, the procedure to generate the sample is crucially different. Whereas the GRS methodology is supported by a statistical background, AEA is heavily dependent on expert opinion and analyst’s choice.

AEA and GRS methodologies consider the uncertainty in input parameters and code models as the most important elements in the analysis. The accurate quantification of this uncertainty is essential to justify the calculated uncertainty in the code results. This quantification is also the most subjective step in the analysis.

The main difference between the AEA and GRS methodologies is the manner in which the information about the uncertainty in input variables and code models is used and propagated to the code results. AEA methodology relies heavily on the analyst’s judgment to decide how to combine the ranges of variation of the input variables and code models, and on the number of necessary code calculations to provide a reliable measure of the uncertainty of the code’s results. GRS methodology, however, combines the uncertainty information in input variables and models according to a statistically sound technique, and decides the number of necessary code calculations also on statistical grounds. The lack of statistical rigor makes it very difficult to defend the objectivity and applicability of the uncertainty statements derived using the AEA methodology.

Scaling and bias issues in these two methodologies should be addressed in the selection of appropriate uncertainty information for the code models based on comparisons to experimental data for a variety of separate effects tests at full and reduced scales. However, the techniques needed for these tasks are still a matter of research.

The use of experimental data from integral tests by the AEA and GRS methodologies is merely to benchmark the quality of the uncertainty information assigned to the input variables and to the code models. If this benchmark is successful, then a degree of confidence is gained in the use of the methodology for a similar scenario in a NPP.

In a general sense, the basic philosophy of the Adjoint, GRS and AEA methodologies is very similar: the uncertainty in input variables and code models is propagated to the code results. It is in the specific techniques used for the propagation where they differ. AEA makes use of expert judgment, GRS uses statistical sampling and the Adjoint method is based on an analytical first order approximation.

CSAU can also be considered as a statistically based methodology. The major limitation of the statistical procedure proposed by CSAU is the use of the response surface technique. Replacing the actual code by a functional fit introduces several serious limitations into the analysis. The use of this technique conditions some of the initial steps of the CSAU methodology, since it forces the selection and ranking of input variables and code models based on a prior expert judgment. The range of variation of these variables is also determined by expert opinion supported by code validation studies.

The non-linear behavior of many of the transient scenarios of interest in NPPs also imposes certain requirements on any uncertainty analysis method. The response surface is usually a linear fit for a specific code result, e.g. Peak Clad Temperature, although second order terms may also be included to account for non-linear behavior. Nevertheless, only for mildly non-linear transients, in which the values of the important system parameters do not depart from the ranges assumed to build the response surface, could the results obtained with this technique be strongly defended.

As discussed above, several of the steps of CSAU rely to a great extent on input from expert opinions. The more objective part of CSAU is the generation of the response surface and the MonteCarlo sampling procedure performed on it. However, this objectivity is affected by the very selection of those parameters that define the surface and by their ranges of variation, which are expressed simply as maximum and minimum values with no measure of how probable the intermediate values are.

Another limitation is related to the application of MonteCarlo sampling to a response surface in the manner proposed by CSAU. It is not suitable to calculate uncertainty for time dependent code results. The response surface is a fit for a selected single value, and time is absent from the derivation of the response surface equation. Moreover, if the uncertainty of a different code output variable is desired, then a new response surface must be developed and sampled. This may require the selection of different parameters for the fit depending on the output variable and the physical phenomena that determine its value during the transient.

Finally, CSAU includes the comparison of code predictions to integral test experimental data as a means to determine biases in the code models and to account for scaling effects. The biases are simply added to the uncertainty measures calculated from the MonteCarlo sampling, e.g. an additional temperature increase to the PCT, which is also somewhat questionable from a statistical point of view.

The last methodology reviewed, UMAE, is inspired by CSAU but without the surface response technique to account for the uncertainty in input variables and code models. The uncertainty information which is propagated to the code results is obtained in the form of biases and uncertainty limits based on the extrapolation of the code accuracy. Accuracy is measured by certain techniques applied to the comparison of the code predictions to a series of integral test experiments. The 'integral' character of the method, which assesses all code models at the same time, makes it difficult to identify the contribution of single models and input variables to the final uncertainty in the code results. This limits the flexibility in its application, since the integral facilities must represent very closely the scenario of interest in the actual NPP. The number of available experimental data bases for a given scenario is usually small, and this affects the statistical application of the techniques and the statistical reliability of the results obtained with very small size samples. Besides, the authors do not make clear how uncertainties in plant state and boundary conditions can be introduced into the calculation process.

From the comparison and ideas presented above, the GRS methodology appears to offer the highest degree of flexibility in its use and the soundest statistical basis. There still remains the problem of obtaining accurate objective uncertainty information for input variables and code models, but, as noted above, this is common to all methodologies except for UMAE.

Compared to Adjoint methods, the GRS methodology is not as analytical and local in its scope, but their philosophies are similar, i.e. the propagation of uncertainties in input variables and code models. Yet GRS offers a more flexible alternative, and its implementation is easier to a wide variety of computer programs without major code modifications.

The use of the actual code in the computation process makes it superior to the statistical approach of CSAU, which is based on response surfaces, and releases it from the need to rank and select important parameters a priori.

Compared to AEA, their similarities end in the procedure followed to propagate the uncertainty in input variables and code models to the code results. GRS's statistical rigor is replaced by expert judgment in AEA methodology. Thus, the uncertainty statements based on the GRS methodology results are statistically meaningful, whereas those produced by AEA method cannot be easily defended on statistical grounds.

Finally, it is important to indicate that the GRS methodology does not directly address the problem of introducing into the uncertainty calculation the influence of scaling, that is the extrapolation of the uncertainty information obtained from code validation and assessment based on scaled integral and separate effect tests to the analysis of a full scale plant. UMAE methodology offers the most rigorous approach to this issue. Nevertheless, the small number of facilities that represent the same or similar transients and phenomena at different scales, makes it difficult to address this issue in a rigorous statistical way. Unfortunately, in the end, the inclusion of this uncertainty effect requires for all the methodologies an important input from the experts, primarily in the form of biases added to the code's uncertainty in the way suggested by CSAU.

Other issues that need to be addressed involve:

- the different nodalilzation used in plant studies, when compared to that used to assess and validate the code predictions;

- the uncertainties introduced in the code's results by the use of one-dimensional models to simulate three-dimensional effects (often, part of the problem of scaling is related to the fact that small facilities may show 1D behavior, when 3D flow may be important in a larger scale NPPs);

- the uncertainty introduced by the assessment and validation of the code models with separate effect tests that measure local phenomena, when they are used to describe the integral behavior of the plant analyzed.

These are still open questions and the subject of active research.

4. PROPOSED METHODOLOGY FOR THE IMPLEMENTATION IN SNAP

The use of uncertainty propagation methodologies based on statistical methods has special advantages with respect to the other methodologies described on Section 5. Without claiming that they could be the best application for all possible application, it is well accepted that their flexibility and sound mathematical basis on statistical theory make them a very good candidate to quantify a statistical quantity such as uncertainty. We have summarized in this section the most relevant arguments that support our selection of a statistical methodology such as that developed by GRS-LANL as a firm candidate to be implemented in SNAP.

Sound statistical basis

The methods employed for the propagation of uncertainty in the GRS-LANL methodology are rooted in Non-parametric Statistical Theory (Conover 1980). The justification that the output values of the computer model is a stochastic variable IF the input parameters are also random variables, is a basic concept in probability theory: a function, deterministic or random, of random variables is also a random variable. Therefore, the application of statistical techniques to the sample of output values obtained after application of the computer model to a sample of input variables is clearly justified.

Most of these techniques have been developed for their application to the analysis of sampled data, and are routinely used by statisticians and scientists who need to extract meaningful, statistically sound information form collections of experimental results or from the surveying and polling of populations (Conover, 1980; Mandel, 1984; Smith, 1994; Crow, 1960).

The statistical character of the methods is especially well suited for uncertainty analysis, because it reflects the probabilistic nature of the lack, or inadequacy, of knowledge of the input and code model parameters that will determine the uncertainty of the computer model’s response. In addition, uncertainty statements about the code performance should be made in probabilistic terms, since this is the very essence of uncertainty, and the methodology allows this need to be fulfilled naturally.

The input from expert judgment is reduced to a minimum

Most of the proposed uncertainty methodologies presented in Section 5 require a great deal of input from expert judgment to be effective. While the need for a knowledgeable analyst is clear, it is also important to reduce the need for subjective input. A methodology that relies heavily on expert judgment may not be applied in a systematic manner with the guarantee that the results will not be ‘user’ dependent, even if the same analysis were to be repeated by different analysts.

Compared to the methodologies currently in use, the GRS-LANL methodology has the potential to reduce expert judgment to a minimum, while maintaining it if no other practical alternative is available. In this respect, the area still requiring subjective input is the quantification of the uncertainties of the input and code model parameters. Unfortunately, as can be inferred from the description of the methodology offered above, this is one of the crucial steps for the application of the methodology. In any case, this limitation appears also in all the other methodologies, except, because of its completely different approach, in UMAE.

Despite this drawback, the selected methodology is basically objective once the initial quantification of input uncertainties is completed. Only the size of the samples and the selection of sensitivity measures need input from the analyst, of whom a certain familiarity with the basics of the statistical techniques used is expected. This is also necessary for the correct interpretation of the results.

Unlimited number of input and code model parameters

Save for limitations related to the lack of knowledge about the uncertainty of input and code model variables, which can preclude the selection of a given input variable when no justification can be given for a particular range or probability function, the number of inputs is not limited. Even the effort required to carry out the analysis is not dependent on the number of input and code model parameters considered, except for the effort to gather the necessary data to describe their uncertainties.

This is a very useful property when compared to other uncertainty methodologies, especially those making use of “surface-response” approximations to the computer model. Therefore, it is not necessary, in principle, the a-priori selection and ranking of input and code model variables to discard those that, based on expert opinion, are not relevant for the computer model simulation under investigation.

The reduction in expert opinion is clearly a useful consequence of this property of the methodology, since all input and code model variables can theoretically be considered for the analysis, and their importance assessed once the results are obtained. In practice, the lack of uncertainty quantification for all the possible variables and the effort required to gather it, make it necessary the ranking and selection of variables prior to the application of the methodology.

The actual computer code is used during the calculation process

Although this point may appear obvious, it is not so in the case of methodologies which use the “surface-response” method, such as the first application of CSAU. In these methodologies the actual computer code is replaced by a functional fit (linear or non-linear) for a certain code result. The procedure for surface response generation is based on the variation over the expected range of some input variables deemed to be important a priori. Then, a least squares approximation is used to generate the coefficients of the functional fit. However, the new code replacement is only strictly used in a very limited region of the input sample space, and even in this case, assuming that the actual code will behave ‘smoothly’, i.e. non-linear effects will not be determinant in the code response. In addition, the need to create a manageable response surface, forces the analyst to make choices regarding the importance of the different input parameters prior to the generation of the functional fit. For these, and other statistical reasons (e.g., lack of ‘randomness’ in the code results) many authors have expressed their reservation and doubts about the use of such a technique in uncertainty analysis (Hofer 1990; Yadigaroglu 1990; Downing 1985).

The method is very well suited for time dependent computer models

The execution of the actual computer code for the sample of the input variables makes it possible to obtain a time dependent output sample if the code is simulating a transient. Thus, averages, means, standard deviations and tolerance intervals can be also calculated as time dependent functions (index-dependent uncertainty). These then offer a dynamic picture of the uncertainty in the code results. Moreover, sensitivity parameters and importance measures for the input variables can be also calculated as time dependent functions. These are very useful when studying the influence of the input variables at different moments during the computer simulation.

A relatively large number of code calculations are needed

This characteristic may be a disadvantage the methodology proposed. Its statistical character requires a certain minimum number of code executions to provide enough sample information to achieve a high degree of precision in the statistical measures used to quantify the uncertainty in the code results. For models that are large and computer intensive, this may be a drawback. However, even for the case of “response surface” methods, a number of code executions are also needed and its number is dependent on the number of input parameters. If this number is relatively large, the number of code calculations may be close to that needed for the methodology selected, thus off-setting the computational advantage of using a surface response. In particular, if cross terms are needed to account for non-linearities in the physical scenario simulated, the combinatorial character of the procedure can enormously increase the number of calculations needed to determine a surface response.

5. SENSITIVITY AND UNCERTAINTY ANALYSIS USING A STATISTICAL SAMPLE OF INPUT VALUES

The sensitivity and uncertainty analysis using a statistical sample of input values is described in the following chapters. The description is structured by first providing an overview over the fundamental mathematical concepts, followed by an implementation guide and closing with a description of sensitivity analysis results.

5.1 Fundamental Concepts

This section introduces the fundamental concepts supporting the statistically based uncertainty propagation methodology proposed to be implemented as a module in SNAP.

It is important to note that in the following subsections the word input variables will also represent variables that determine or are the outcome of a given code´s physical models.

5.1.1 Purpose

The statistical content of the GRS-LANL methodology is based on the work of McKay et al. (McKay 1988). This research was initiated as a means of quantifying the uncertainty associated with the results of computer codes developed to track the migration of radioactive effluents in the biosphere. Its conceptual basis lies on the statistical methods applied to the analysis of experimental and observational data. By applying these techniques to the results of computer codes it is then possible, in principle, to quantify the uncertainty in their results and to determine their sensitivity with respect to input variables and code models.

Computer codes are deterministic by nature, that is, for a given set of input values, they yield a unique output. However, if the input parameters are treated as random variables, then the stochastic nature of the inputs induces a stochastic nature in the outputs. The justification of the use of statistical methods to computer codes is based on such a premise.

Randomness in the input variables can be due to various factors, e.g., statistical variation, unknown values, etc. Uncertainty is related to the random nature of lack of precise information of the exact values of these variables and is commonly, although not necessarily, quantified by means of Probability Density Functions (PDFs). This uncertainty is propagated through a deterministic computer model to the results. The uncertainty in the code results can then be measured by appropriate statistical techniques and presented in the form of tolerance bands, standard deviations, or even an empirical PDF if statistically possible.

The goal of a methodology based on the use of a statistical sample of the values that input variables can attain as a means to represent their uncertainty, is to make use of a set of statistical techniques capable of quantifying the uncertainty in the code´s results. The methodology can also be used to determine the sensitivity of these results to the input variables based on the calculation of correlation coefficients and the application of regression analysis.

5.1.2 Basic Mathematical Concepts

In order to define a random variable it is necessary to introduce the concept of *sample space*, commonly known as *population*. A random variable is then the collection of all possible different outcomes of a statistical experiment (e.g. rolling a die, polling people, measuring a physical quantity, etc.) Each one of the possible outcomes is known as a point in the sample space. For example, for the die case, each number 1, 2, 3, 4, 5 or 6 is a point of the sample space $\{1,2,3,4,5,6\}$, which contains all possible outcomes when the die is cast.

A random variable can then be considered, in general, as a function that assigns a real number to each point of a sample space. It can be, for instance, the value of a measurement, the number obtained when rolling a die, a number assigned to each possible answer to a survey, or simply assigning 1 and 0 to the two different outcomes of tossing a coin: heads and tails (Conover 1980).

A random variable can be completely described by its Probability Density Function, PDF, $f(x)$, which gives the probability that the variable X takes a value equal to x, one of the points in its sample space, if the variable can adopt discrete values, e.g. integer numbers; or the probability that X lies within $(x\text{-}dx,\ x+dx)$, in the case that X is continuous. That is,

$$f(x) = \begin{cases} \text{Probability}\{X = x\} \text{ if } X \text{ is a discrete random variable} \\ \text{Probability } \{x - dx < X < x + dx\} \text{ if } X \text{ is a continuous random variable} \end{cases} .$$

(1)

From the PDF, the Cumulative Probability Function, CPF, can be expressed as

$$F(x) = \begin{cases} \text{Probability } \{X \le x\} = \sum_{z \le x} f(z) \text{ for a discrete variable} \\ \text{Probability } \{X \le x\} = \int_{-\infty}^{x} f(z)\,dz \text{ for a continuous variable} \end{cases}$$

(2)

A PDF can also describe a set of random variables. In this case, it is known as a joint PDF and it expresses the combined probability density for the whole set. For instance, for two discrete random variables, X_1 and X_2, the function $f_{12}\,(x_1,x_2)$ represents the Probability $\{X_1 = x_1 \text{ and } X_2 = x_2\}$. And from the joint PDF, the joint CPF can be expressed as

$$F_{12}(x_1, x_2) = \text{Probability}\{X_1 \le x_1 \text{ and } X_2 \le x_2\} .$$

(3)

The concept of joint probability is the basis to define the independence of two random variables, whether discrete or continuous, as those whose joint PDF can be expressed as

$$f_{12}(x_1,x_2) = f_1(x_1)\cdot f_2(x_2)\,,$$

(4)

and, in general, for n independent random variables:

$$f_{1\ldots n}(x_1,\ldots,x_n) = \prod_{i=1\ldots n} f_i(x_i)\,,$$

(5)

where the individual $f_i(x_i)$ are the individual marginal PDFs for each random variable X_i. Each of these marginal PDFs can be obtained from the joint distribution as:

$$f_k(x_k) = \begin{cases} \displaystyle\sum_{\substack{All\,Sample\,Space \\ for\,all\,x_i\,,\ i\neq k}} f_{1\ldots n}(x_1,\ldots,x_n) \text{ for discrete random variables} \\ \\ \displaystyle\int_{-\infty}^{+\infty} f_{1\ldots n}(x_1,\ldots,x_n)\, \underbrace{dx_1\ldots dx_n}_{\substack{i=1\ldots n \\ i\neq k}} \text{ for continuos random variables} \end{cases},$$

(6)

which represents the stochastic variation of X_k averaged over all possible values of the rest of the variables $X_{i=1\ldots,k-1,k+1,\ldots n}$.

When the random variables are not independent, conditional PDFs define the stochastic variation of one variable, say X_i, for a fixed value, x_k, of another random variable, X_k. In the case of two variables, X_1 and X_2, this can be represented as:

$$f_{1|2}(x_1 \mid x_2) = \frac{f_{12}(x_1,x_2)}{f_2(x_2)}\,.$$

(7)

From the PDF of a random variable, useful information can be obtained about its sample space. Functions of the random variable are defined to calculate this information. The three most used functions are the population mean, also known as expected value $E[X]$, the population variance, $var[X]$, and the population standard deviation, σ. They are defined in terms of a random variable's PDF as:

Expected Value of X : $E[X]=\int_{-\infty}^{+\infty} x\,f(x)\,dx = \mu$

Variance of X: $\mathrm{var}[X]=E[(X-\mu)^2] = \int_{-\infty}^{+\infty} (x-\mu)^2\ f(x)\ dx = \sigma^2$

Standard Deviation of X : $\sigma = \sqrt{\text{var}[X]}$

(8)

It is important to note that *E[X]* is not necessarily the most probable value of *X*, and that *var[X]* measures the 'spread' of the random variable about its expected value, or mean, μ.

As noted above, the application of statistical methods to quantify the uncertainty of computer codes is based on the notion that the input variables and the predictions of the code models can be considered as random variables. For a series of *k* input variables and code models, the code's output random variable *Y* is then a function of a vector of random variables $(X_1,...,X_k)$. Each one of these random variables X_i corresponds to one input variable or code physical model. The code, in general a non-linear deterministic function, ***Code()***, transforms the vector of input random variables into the output, and in the process, the stochastic nature of the input variables and code model predictions, $(X_i,\ i=1,k)$, is propagated to the output, thus making *Y* also a random variable. Mathematically this can be expressed as:

$$\boldsymbol{X} = (X_1, X_2, \ldots, X_k)$$
$$Y = Code(\ \boldsymbol{X}\ )\ .$$

(9)

The vector of input random variables, ***X***, is called a multivariate, since it is a multidimensional random variable which contains several variates or single random variables.

One code execution requires a sample of size ***1*** of the multivariate ***X***. The sampling process yields a set of *k* values, $\{x_i,\ i=1,k\}$, each one of them a point of the sample space of each of the *k* random variables that form the multivariate $\boldsymbol{X} = (X_1, ..., X_k)$. This sample is one single *k*-dimensional vector of values, with each element X_i representing a possible value for one of the *k* input variables and code models considered. After the code's execution, a single value of the output random variable ***Y***, *y*, is obtained as a function of the $\{x_i,\ i=1,k\}$

$$y = Code(\ (x_1, \ldots, x_k)\ )\,.$$

(10)

In general, a *sample* of size *n*, is a *collection* of *n random variables (represented as* $()_n$), in which each of the *n* variables can be a single variate, e.g. $(Y)_n$, or a multivariate, e.g. $(\boldsymbol{X})_n$. Thus, for instance, a *sample* of size *N* of the set of *k* input variables $\boldsymbol{X} = (X_1,...,X_k)$ yields the sequence $((\boldsymbol{X})_1,...,(\boldsymbol{X})_N)$. Each one of the $(\boldsymbol{X})_i$ is a random multivariate composed of *k* input *random variables* $\{X_i, j=1..k\}$. The sampling process yields the values $(\mathbf{x})_i = (x_1,..., x_k)_i$, which are the actual values of the input variables input to the function *Code()*, i.e. the computer code, to obtain the output value y_i. This process is illustrated in Figure 1.

By means of the sampling process, *k* values x_i are obtained; one for each of the *k* input random variables X_i. The code is then executed with each *k*-vector of values $(\mathbf{x})_i = (x_1,..., x_k)_I$ from the sampling process *i*, and one code output variable, *$(y)_1$* is then obtained. This process is repeated *N* times, and a sample of size *N, $(Y)_{i=1,\ldots N}$* , is then obtained for the output random variable *Y* (note that it is assumed that there is only one output variable in the example; in general *Y* could be a vector of *m* output variables $\boldsymbol{Y}$ = *$(Y_1,\ldots,Y_m)$*).

INPUT SAMPLE of Size N

OUTPUT SAMPLE of Size N

$$\begin{pmatrix}(\boldsymbol{X})_1\\(\boldsymbol{X})_2\\\vdots\\(\boldsymbol{X})_N\end{pmatrix} \equiv \begin{pmatrix}(X_1,\ldots,X_k)_1\\(X_1,\ldots,X_k)_2\\\vdots\\(X_1,\ldots,X_k)_N\end{pmatrix} \xrightarrow{\text{Assign Values to input}} \begin{pmatrix}(x_1,\ldots,x_k)_1\\(x_1,\ldots,x_k)_2\\\vdots\\(x_1,\ldots,x_k)_N\end{pmatrix} \longrightarrow \begin{pmatrix}Code((x_1,\ldots,x_k)_1)\\Code((x_1,\ldots,x_k)_2)\\\vdots\\Code((x_1,\ldots,x_k)_N)\end{pmatrix} \equiv \begin{pmatrix}(y)_1\\(y)_2\\\vdots\\(y)_N\end{pmatrix} \equiv \begin{pmatrix}Code((\boldsymbol{X})_1)\\Code((\boldsymbol{X})_2)\\\vdots\\Code((\boldsymbol{X})_N)\end{pmatrix} \equiv \begin{pmatrix}(Y)_1\\(Y)_2\\\vdots\\(Y)_N\end{pmatrix}$$

Input Multivariates

Input Variables for N runs

Output Variables for N runs

$(\boldsymbol{X})_i$: represents sample number *i* of the *k* input variables.
Code $((\boldsymbol{X})_i)$: represent the application of the Code to sample *$(\boldsymbol{X})_i$* .

Figure 1. Sampling Process Scheme. k Input variables; Sample of Size N.

The calculation process, ***Code*$((\boldsymbol{X})_i)$**, effectively samples the space of ***all*** the possible code output values, and generates a set of *N* values for the output variable of interest $\boldsymbol{Y}$, *$\{(Y)_i, i=1 \ldots N\}$*. It is important to note that each of these values can also be a function of time, not just a single value, i.e. *$y_i(t)$*, if the code investigated is a transient code, and that $\boldsymbol{Y}$ itself may be a vector-valued function, e.g. if more than one output variable is considered of interest.

Statistical techniques can then be applied to extract information from these code output values. For example, two important functions of the values of a sample *$(y)_i$* are the *estimators* of their expected value (mean) *$E[(y)_i]$* and of their variance *$var[(y)_i]$* (see above). They are known as

sample mean : $m_y = \frac{1}{N} \sum_{i=1,N} y_i$,

(11)

sample variance : $s_y^2 = \frac{1}{N-1} \sum_{i=1,N} (y_n - m_y)^2$,

(12)

and *sample standard deviation* : $s_y = \sqrt{s_y^2}$.

(13)

These are *unbiased estimators*, that is, the expected value of the *sample mean* *$E[m_y]$* is equal to the expected value (mean) of the random variable *$E[Y]$*, and the expected value of the sample variance *$E[s_y^2]$* is the same as the variance of the random variable *$var[\boldsymbol{Y}]$*, computed according to Eq. 8 (note that *Y* represents the actual *population* or *random variable*, that is ***all*** possible values that $\boldsymbol{Y}$ can take, not just those obtained during the sampling process).

$$\begin{aligned} E[m_y] &= E[Y] \\ E[s_y^2] &= \mathrm{var}[Y] \end{aligned} .$$

(14)

This means that by analyzing a sample of the output values, one can extract statistical information that describes the entire population of possible code output results. That is, given an output variable of interest, the mean of the output sample values (that is, the mean of the code output values obtained for this variable after N code executions) will give an unbiased estimate of the mean of ***all*** the possible code output results for this variable. The variance of the output sample values will also be an unbiased estimate of the variance (uncertainty) of all the possible code results for the output variable selected. In summary, the calculation of the mean and the variance of a series of output values for a given output variable can yield information about the mean and variance of ***all*** possible output results for this variable that could be obtained by ***randomly*** varying the input variables and core model predictions within their allowed ranges of variation.

If the output variable is a function of time $\boldsymbol{Y}(t)$, then these estimators can be computed at any point in time. This provides a time dependent picture of the evolution of the mean and the variance (uncertainty) of this variable as the transient progresses , i.e. $m_y(t)$ and $s^2{}_y(t)$. For time τ, then

$$m_y(\tau) = \frac{1}{N} \sum_{i=1,N} (y(\tau))_i \text{ and,}$$

$$s_y^2(\tau) = \frac{1}{N-1} \sum_{i=1,N} ((y(\tau))_n - m_y(\tau))^2 \,. \tag{15}$$

There are different sampling plans in statistics and their characteristics determine the statistical information that can be obtained from the analysis of the collection of sampled values they produce. The most fundamental sampling plan is the *simple random sample*. Let us assume a random variable $\boldsymbol{X}$, with a sample space S_X. S_X contains all possible values that X can take. In a *random sample* of size n a single value for X is extracted from S_X n times. Each one of these ‘extractions’ is a *random variable* itself, and can take any value in S_X with a certain probability. After n ‘extractions’ a set of n *random variables* (one per ‘extraction’) is obtained $\{(X)_1,..., (X)_n\}$ which constitutes a sample of $\boldsymbol{X}$ of size n.

If the $(X)_{i=1..n}$ are *independent* and they all have *the same Probability Density Functions (PDF)*, then the sample procedure is known as a *Simple Random Sample*. It is important to note that, in the case of a multivariate *random variable*, e.g. $\boldsymbol{X} = (X_1,...,X_k)$ its individual components (variates), $X_{i=1,...,k}$, which are also *random variables*, may not be independent and may not have the same PDFs, but the sampling procedure will still be a *simple random sample* if the conditions are satisfied by their **joint** PDF, that is, the PDF of $(\boldsymbol{X})_i = ((X_1, ..., X_k))_i$ for n samples, $i=1,..., n$ (Conover 1980.)

To illustrate the application of these concepts to the sampling of input variables of a computer code, let us assume a sample of size n, $(\boldsymbol{X})_i = ((X_1, ..., X_k))_i$, $i=1,... n$. Since more than one input variable is considered (there are k), then a ***joint*** PDF can be assigned to the multivariate $\boldsymbol{X}=(X_1, ..., X_k)$, which is a function of the individual PDFs of the input variables $X_{1...k}$. When a sampling procedure is carried out, a set of values $(\boldsymbol{x})_i = ((x_1,...,x_k))_i$ is obtained, one for each $X_{1...k}$. If the process is repeated n times, then a sample of size n is generated (there will be n values for each of the k input variables). The sample thus obtained can then be considered a *simple random sample* if one can assume that the ***joint*** PDF of the entire set of input variables, $(X)_i$ has not changed during the sampling process, and if there is ***no relation*** between one set of k sampled values (one for each input variable) and any other set. The individual PDFs of the input parameters, however, may change in the process, and some of the k input variables may be dependent from each other. The independence requirement applies to an entire set in relation with all other $n-1$ sets generated during the sampling process, not to the individual k components of the set.

Another class of important sampling plans is the so-called *stratified sampling plans*. They essentially divide the range of variation of a random variable, its sample space, into intervals (strata) and each of them is then randomly sampled. They produce a more even coverage of the entire sample space resulting in better estimators, i.e. with smaller variances, than the *simple random sample* technique. The most commonly used of this plans is the Latin Hypercube Sampling (LHS).

5.1.3 Quantification of Uncertainties

Once the sample of code output results, $(\mathbf{Y})_N$, has been obtained by executing the code, ***Code( )***, for the sample of input values $(\mathbf{X})_N$, there remains the task of extracting statistically meaningful information from the sample of outputs, $(\mathbf{Y})_N$. This is accomplished by applying appropriate techniques that use this information to generate uncertainty estimations for the code results.

As previously discussed, a result of describing the uncertainties in the input variables with probability distribution functions is that the code output results are also random variables. The PDFs of the code results contain all the information needed to compute their uncertainty. The problem is that such functions are usually unknown. Therefore, in order to quantify exactly the uncertainty one should generate the PDFs from the sampled output values. But this is not always feasible, especially if an analytical expression for ***Code( )***, the code, is not known. In such a case, it is impossible to obtain an analytical expression for the probability distribution of the output variable (McKay 1988). Thus, the only remaining alternative is to obtain as much information as possible about the PDFs properties and main parameters from *empirical* distribution functions and estimators.

One of the most useful estimators are the *quantiles*, A quantile is a point in the sample space of a random variable such that the probability of the variable being less than or equal to the quantile is a given value $p < 1$. In mathematical terms, a $p = 0.95$ quantile for a random variable X is expressed as

$$P(X \leq x_{.95}) = 0.95 , \tag{16}$$

and, in general, the p quantile, x_p is

$$P(X \leq x_p) = p . \tag{17}$$

When the probabilities are expressed in percentages, the quantiles are known as percentiles. In the case of the 95% percentile, (equal to the 0.95 quantil) there is a 95% probability that the value of the random variable X will be less than or equal to $x_{95\%}$.

If X is a time dependent random variable $X(t)$, e.g. the output of a transient code, then the quantiles vary with time and can be used to quantify uncertainty for the duration of the transient.

Another important concept is the *tolerance interval*, which is a more qualitative and conservative measure of uncertainty (McKay 1988). In order to understand its application, it is necessary to make the distinction between confidence and probability content of a random variable. Let us assume that the interval (L,U) - L lower *tolerance limit* and U upper *tolerance limit* - is a tolerance interval with a confidence of 95% and a coverage of 90% of the probability of the variable X. This means that if a series of samples of size n were to be made of X, for 95% of them - ***confidence*** - 90% of the n sampled values from each sample -***probability coverage*** - would lie within (L,U).

Thus, a tolerance interval (L,U) is an estimate of a random variable that contains a specified fraction of the variable's probability, p, with a prescribed level of confidence, γ (Crow 1960.) For the above case, $p = 0.9$ and $\gamma = 0.95$. Tolerance intervals are constructed from sampled data so as to enclose $p\%$ of the population of a random variable X with a given confidence γ. They show where most of the population of the random variable X can be expected to lie.

Tolerance intervals can be better understood by answering the following questions: *How probable is it that if we sample the random variable X the values obtained will be within (L,U)*? Answer: p. *How sure are we of this*? Answer: γ. In the case of a computer code, the sampling of the variable is the calculation of a value for a given code output variable of interest with a set of values sampled from the code input variables.

It is important to distinguish between a *confidence interval* and a *tolerance interval*. A confidence interval is an interval about the **estimate of a population measure** of a random variable X, that, with a specified confidence, contains the population measure (Crow 1960). For instance, if the population measure of interest is the mean μ, then an estimate of μ can be the **mean of a sample** of X, x_{mean}. Based on the sampling distribution of x_{mean}, one can define an interval about x_{mean} that will contain the value of μ with a certain degree of confidence. Thus, for a 95% confidence interval, if one drew many samples and computed x_{mean} for each of them, in about 95 cases out of 100, the confidence intervals would contain μ. They are a measure of the 'closeness' of the sample estimate to the true value of the population measure. The extreme points of a confidence interval are known as **confidence limits**. The confidence is measured by the **confidence coefficient,** e.g. 95%. Confidence coefficient, interval length and sample size are mutually dependent. As an example, if the standard deviation of the sample mean $\sigma_X = (s^2_x)^{1/2}$, being s^2_x the variance, is known, then a 95% confidence interval can be constructed about the sample mean x_{mean} as $x_{mean} \mp 2\ \sigma_X$. This means that the population mean μ will lie within the interval with a 95% confidence.

In uncertainty analysis the main goal is to quantify the variability of the code output due to the variability in the inputs. A confidence interval for the *expected (mean)* value of the code output, a single number, cannot correctly represent this variability. As an example, the variance of the sample mean, s^2_x, which is used to establish confidence intervals (see above), is proportional to σ^2/N for many distributions, with σ^2 the variance of the population of the *random variable*. This means that, if the sample of output values were large enough, N large, then the confidence interval would tend to zero width, which obviously cannot represent the uncertainty of the output as defined earlier (McKay 1988.) Thus, confidence intervals **should not be used** to quantify the uncertainty in code results.

If a random sample of output values, $((Y)_1,..., (Y)_n)$, has a *normal* PDF, it is possible to compute *tolerance intervals* from the sample mean, m_y, and sample standard deviation, s_y as :

$$(L,U) = (m_y - K\,s_y \;,\; m_y + K\,s_y) \tag{18}$$

where K is the so-called *tolerance factor,* whose values depend on the sample size, probability coverage, p, and confidence level, γ. For example, for a sample of $N = 100$, a coverage of 90% probability and a confidence of 95%, K is equal to 1.874 (McKay 1988). The values for K are tabulated for different p, N, and γ in standard statistical tables (Crow 1960, Conover 1980.)

It is not easy to guarantee, though, that the sample of the output values is *normally* distributed. Nevertheless, if the sample is a random one, statistical tests for normality can then be used to quantify how well the hypothesis of normality fits the sampled data. Three of these tests are the W-statistic (Shapiro 1968), the Lilliefors test (Lilliefors 1967) and the Kolmogorov's normality test (Kolmogorov 1933). These tests can provide also a measure of the 'non-normality 'of the sample, and help make decisions as to the suitability of applying the techniques based on the assumption of normality.

5.1.4 Quantification of Sensitivity

One of the advantages of the methodology described in this Section is that it is not necessary to perform an a priori selection of important input variables and code models. From the information contained in the sample of the input variables and in the sample of the code output results, statistical measures of correlation and importance can be computed.

The purpose of a sensitivity analysis is to quantify the influence of the input variables on the code results. Sensitivity measures can assign a numerical value to this influence, and thus, be useful for an *a posteriori* ranking of the importance of each of the input variables with respect to the output variable of interest.

The most detailed sensitivity measures are local in nature, that is, they are calculated for variations in the neighborhood of a point in the sample space of the input variable. In actual applications, more than one input variable is included, and the sample space corresponds to all of them. Thus, a point is usually a vector containing a series of values, one for each of the input variables. Usually, the point of interest for the sensitivity is the nominal value, $\boldsymbol{x_0}$, (nominal input values) and the local sensitivity measure is a vector, $\boldsymbol{s}$, of partial first order derivatives of the code output variable of interest Y with respect to each of the code input variables X_i

$$s(x_0) = \left[\frac{\partial Y}{\partial X_1}, \ldots, \frac{\partial Y}{\partial X_k} \right]_{x_0},$$

(19)

whose value is a function of the vector of input variables x_0. This is the concept used in the Adjoint uncertainty methodology described in Section 5.

Global sensitivity measures *statistically* quantify the variability of the code results with respect to the ***entire*** sample space of the input variables. They are less precise than the local measures, because the influence of a given input variable is quantified by a *single* number covering the entire range of variation of the variable. However, they are much less computer intensive and provide valuable information for the ranking of input variables by importance. They are especially efficient when the effect of the variation of an input variable on the code results is not dramatically different at two distinct locations within the interval of variation of the variable.

Most of the global sensitivity measures are related to *regression analysis*. Some of them are useful to detect linear relationships, and some others, like the so-called *Rank Correlations* are useful to quantify relations between variables that behave monotonically with respect to each other (e.g., smooth variations of one variable correspond to smooth variations of the other one.) Comparison between these two types of measures applied to the same set of data can detect non-linearity in the behavior of the computer code.

Examples of linear measures are the Simple Correlation Coefficient SCC or *Pearson's moment product*, and the Partial Correlation Coefficient (PCC). Their rigorous definition can be found in (Conover 1980) and (McKay, 1988). The most important advantage of the PCC is that it eliminates the linear influence of the remaining input variables on the output, leaving only that of the input variable whose sensitivity is being calculated.

In order to deal with models which are **not** clearly linear, Simple (SRCC) or Partial (PRCC) Rank Correlation Coefficients can be used. To calculate these two measures, the sample values of the input and output whose relationship is to be determined are separately 'ranked', i.e. forming two separate, ordered lists in decreasing or increasing order, and a *rank* (ordinal) assigned to each value. If the two 'un-ranked' original series of values are related monotonously (see above), then the *ordered* series are linearly related. This is true even if the relationship between the unordered series is not linear. Thus the absolute values of SRCC and PRCC will quantify the degree of relationship between the given input and the output of interest. The closer the values of these coefficients to one, the more influence the input will have on the behavior of the output. Several authors have proposed formulas to compute the SRCC and PRCC from the sampled values. The two more used in sensitivity analysis are the Spearman's and Kendall's coefficients (Conover 1980).

Finally, another kind of sensitivity measures is related to "response-surface" analysis, also known as *regression analysis*. They are called *Regression Coefficients*, and their calculation is based on the approximation of the computer model by a linear (or non-linear function if needed) for a reduced number of important input variables. The selection of such variables can be based on a previous application of the global sensitivity measures described above. Response-surface techniques yield regression coefficients as parameters that multiply the input variables in the mathematical function replacing the actual computer code. Further discussion on these techniques can be found in the literature (Draper 1981; Myers 1971; Downing 1985).

In (McKay, 1988) a comparison between the interpretation of the meaning of a regression coefficient and a correlation coefficient sheds light on their application to sensitivity analysis. While they are mathematically related, a correlation coefficient measures the *degree of strength* of the relation between the input and the output. The normalization of these coefficients by the sample standard deviations makes them non-dimensional. On the other hand, the regression coefficients measure the *intensity of this relation*, i.e. how much the output changes *per unit* of change in the input. As an example, in a time dependent case, input and output variables can be 'strongly' correlated during the whole calculation, but the intensity of this relation may vary with time, e.g. take the linear model $y = a + b(t)x$, there is a complete linear relationship (|*correlation coeff.*| = 1) between y and x, but the intensity of this relationship varies with time as given by the regression coefficient $b(t)$.

From the discussion above, it is clear that the selection of the sensitivity measure depends on the information desired from the analysis. The following considerations can serve as a guide:

If the ranking of input variables according to their global importance for a given output variable is sought, then a correlation coefficient should be selected.

When detailed knowledge of the influence of an input variable on a given output for the entire range of variation of the parameter is desired, then regression coefficients are needed.

Finally, if local, detailed sensitivity information in a neighborhood about the nominal values of the input variables is required, a local measure based on partial derivatives about the nominal point in the input space should then be computed.

5.2 Main Steps for the Implementation of the Methodology

This sub-section describes the most important steps in the procedure to carry out an uncertainty propagation based on the statistically based methodology proposed for implementation in SNAP. Figure 1 shows an scheme of the sub-modules proposed as a basis for the development of an uncertainty propagation module in SNAP. They are described in detail below.

5.2.1 Quantification of Uncertainties in Input and code model parameters

The application of an uncertainty methodology follows a series of steps that culminate in a statement about the uncertainty in the code results. Most of the methodologies described in section 5 provide a series of guidelines devised to help in the determination of the relevant input variables and code physical models according to the characteristics of the simulation. Some of them also need the a-priori ranking and selection of a set of these variables based on their perceived importance (expert opinion) in determining the outcome of the simulation. Thus the module should provide enough freedom to support suitable selection of uncertain input parameters corresponding to different methodologies. The module may support the user in the context of the workflow by providing an analysis dependent suggestion for the process identification and ranking table i.e. the selection of relevant uncertain input parameters.

As has also been noted before, the advantage of a methodology based on a statistical sample of input variables and code models is that there is no need to make an *a priori* selection of which of the input variables are more important for the simulation. Methods for sensitivity analysis (see above) can provide a more objective *a posteriori* assessment of the importance of the input parameters.

After the relevant input variables and code models have been identified, the analyst can proceed to the quantification of their uncertainties. The variables are assumed to be *random variables* with respect to taking values within their ranges of variation (uncertainty). The range of statistical variation of all these variables needs to be determined with the help of experimentally observed ranges or from previous experience. According to this information, Probability Density Functions (PDFs) should be assigned to the input variables before the sampling process can take place. PDFs quantify the likelihood of these variables taking specific values (or being less than or larger than a given value in case they are continuous variables) within their range of variation (their sample space).

This initial phase of the analysis is the most subjective of the entire process. The determination of PDFs is not a simple task, and for many variables the actual functions are not known. However, some guidance can be obtained from the fact that when available, observations or experimental results can suggest specific probability distributions. If experimental data is available which was used to develop some of the models in the code or to determine important parameters used by the models, statistical information about range and probability distributions can be obtained. Another source of information is the validation of models against a wide range of experiments. This approach may provide statistical information for the uncertainty in the code models by applying appropriate techniques to the results of the validations (Vinai, 2007).

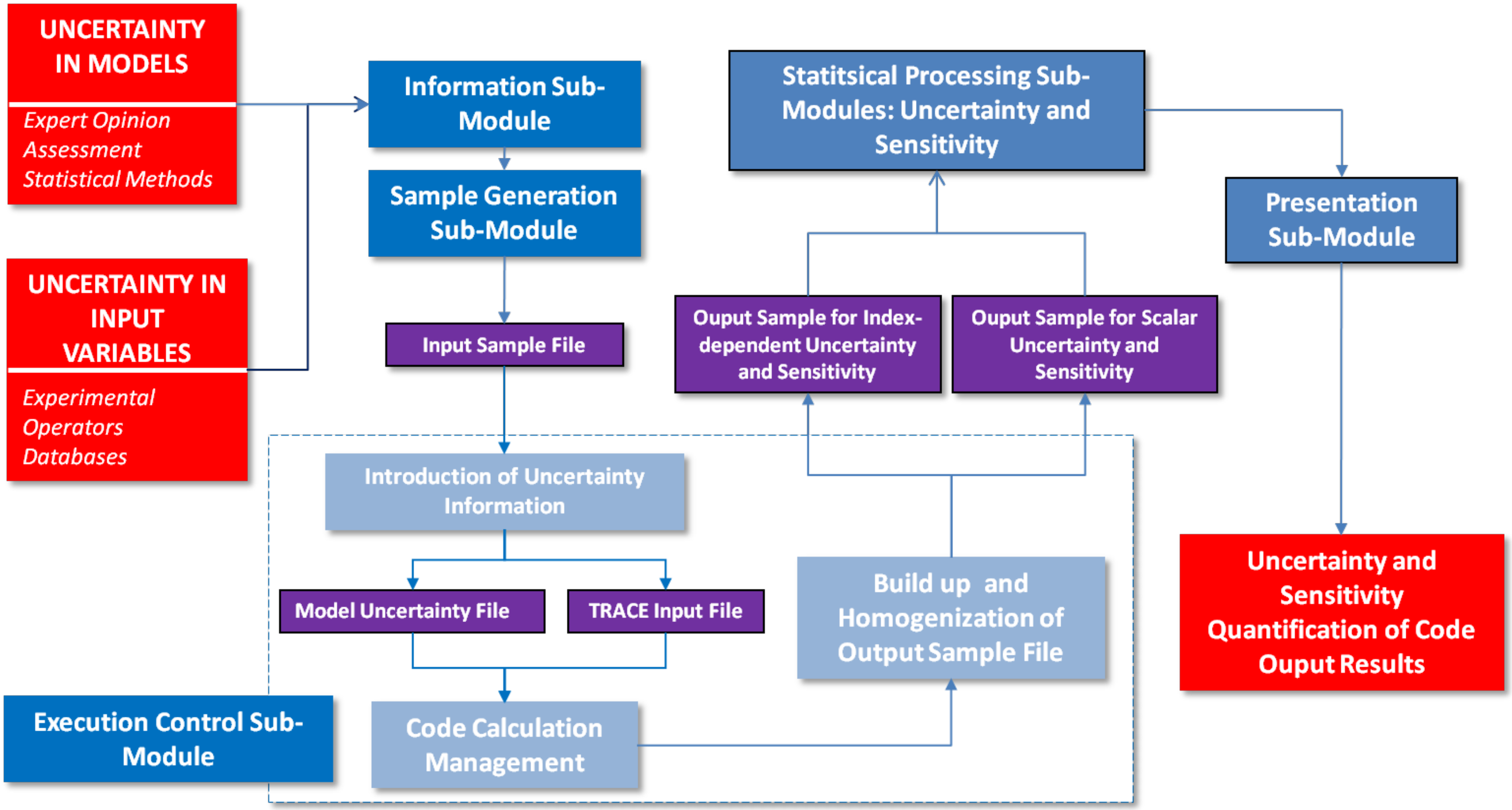
UNCERTAINTY IN MODELS
Expert Opinion
Assessment
Statistical Methods
UNCERTAINTY IN INPUT VARIABLES
Experimental
Operators
Databases
Information Sub-Module
Sample Generation Sub-Module
Input Sample File
Introduction of Uncertainty Information
Model Uncertainty File
TRACE Input File
Code Calculation Management
Execution Control Sub-Module
Build up and Homogenization of Output Sample File
Ouput Sample for Index-dependent Uncertainty and Sensitivity
Ouput Sample for Scalar Uncertainty and Sensitivity
Statitsical Processing Sub-Modules: Uncertainty and Sensitivity
Presentation Sub-Module
Uncertainty and Sensitivity Quantification of Code Ouput Results

Figure 2. Scheme of the Sub-Modules Proposed for a SNAP Uncertainty Propagation Module

When no data is available, the only recourse is to assign Subjective PDFs (SPDF) based on experience or subjective judgment.

Of the most frequently used PDFs, the one that maximizes the lack of knowledge is the *Uniform Distribution*, since, based on the range of variation of a variable, it assigns equal probability to each value within its sample space. The *Normal* or *Lognormal* distributions are usually employed to describe experimental measurements and other natural variations, although a wide variety of distribution types are available. The PDFs can also be truncated to account for the fact that some parameters may have their range of variation limited by physical constraints, e.g. pressures too low or two high. In general, the assignation of the appropriate PDFs should be made according to the information available for each variable.

As is clear, the choice of PDFs for the input variables and code model parameters will influence the results of the analysis, since the stochastic characteristics of the input PDFs are propagated through the deterministic computer model to the output results. Therefore, special care should be taken when assigning uncertainty information to input variables and code model parameters.

A final aspect of this procedure needs to be considered. When some input variables are not mutually independent, their dependency must be explicitly taken into account during the assignment of the uncertainty measures, these being PDFs, variation intervals, etc. Following (McKay 1988) several situations can arise given two variables:

If the range of variation of one variable ***is not*** affected by the range of variation of the other variable, as a first approximation, their dependency can be ignored. The dependency might also be expressed by a joint probability distribution, or by the pairing of values according to the dependency characteristics of the values sampled from each of the dependent variables, e.g. a function that relates both variables $F(X_1, X_2) = 0$.

If, however, the range of variation of one of the parameters is limited by the range of variation of the other, this limitation must be included in their joint PDF, and taken into consideration during the sampling process.

5.2.2 The Sampling Process

Once the PDFs and ranges of variation have been assigned to the input variables and code models, the space of these random variables has to be sampled. The module should have the freedom to support different sampling processes corresponding to the fundamental mathematical concepts described in the previous chapters.

The important decision at this stage is to determine the size of the sample. This decision has to be a balance between the desired precision of the analysis and the computing resources required to obtain the sample of output values, i.e. the number of code calculations needed. However, a precise determination of the sample size to achieve a specific degree of precision is not usually possible, except for very restrictive assumptions if no other assumption is made with respect to the type of uncertainty measure used for the output variable. In (McKay 1988) some general considerations to help in deciding the sample size are offered. It is important to note, that the precision of the results obtained with statistically based uncertainty propagation methodologies is not dependent on the number of input parameters, but, among other factors, on the *sample size* and *randomness* of the sampling procedure. The latter condition ensures the randomness of the sample of output values.

Thus, for a random sample of code output values, if the sample size is 20, the largest and smallest values in the sample represent 75% of the probability content with a 95% probability, that is they constitute a tolerance interval with a 75% probability and a 95% certainty (see above discussion on tolerance intervals). A minimum sample size of 20 introduces a dispersion of output values that contain *most of the uncertainty* of the output results for random variations of the input variables and code model predictions. Of course, this is just a minimum, and larger sample sizes will contribute to increase the probability content of the tolerance limits with a higher degree of certainty.

If a sensitivity analysis of the input variables is also desired, the minimum sample size has to permit the calculation of derivatives (local sensitivity) or correlation measures (global sensitivity). In both cases, samples with a minimum of size $k+1$, being k the number of input variables and code models, are then necessary. Therefore, it is clear that the number of input variables and code models considered in the analysis does play a role in deciding the sample size if a sensitivity study is also needed.

Therefore, the minimum sample size can be taken as the largest of 20 and $k+1$. But, as a rule offered in (McKay 1988), a good sample size should be as a minimum about twice the anticipated number of input variables and code models included in the analysis.

A more rigorous approach to determine the minimum sample size is provided in (Conover 1980) and was introduced to the statistical methodology by GRS. The general formula was obtained by Noether (Noether 1967) as

$$\alpha \geq \sum_{i=0}^{r+m-1} \binom{n}{i} (1-q)^i \, q^{n-i} \,, \tag{20}$$

where $(1\text{-}\alpha)$ is the confidence of the *tolerance interval*, q is the proportion of the population within the interval, n is the size of the sample (the unknown), and r and m are numbers related to the position in the ordered sample of the values that from the upper and lower limits of a *tolerance interval*. The most used values are $r+m=1$ (one-side tolerance limits) and $r+m=2$, two-sided tolerance limits. A value of $r=0$ represents a one side upper tolerance limit (the lower side is -∞) and an $m=0$, represents a one side lower tolerance limit (the upper side is +∞)

The solution of this equation for n, when q and $1\text{-}\alpha$ are known, yields the ***minimum*** sample size. In (Conover 1980) tables are given with the solution for various $(1\text{-}\alpha, q)$ pairs for the cases $r+m=1$ and $r+m=2$. For $r+m=2$, Eq. (20) is known as the Wilks Formula (Wilks 1962): $\alpha \geq (1\text{-}n)\ q^n + n\ q^{n\text{-}1}$. For instance, the ***two-sided*** tolerance limits for 95% of the population ($q=0.95$), with 95% confidence $(1-\alpha = 0.95)$ the minimum sample size, n, is ***93***.

5.2.3 Useful considerations for the Sampling Process

The generation of the sample needed to apply the statistical uncertainty propagation methodology should take into account some important requirements to in order not to affect the statistical significance of the uncertainty measures obtained from the propagation procedure.

- Sampling Method

  On a mathematically rigorous basis, *Simple Random Sampling* should be the technique used to generate the sample of values needed to modify the input variables and code physical models. The reason for this is that, according to (Conover 1989), only this type of sampling procedure makes the application of the Wilks' (Noether´s) formula (Eq. (20)) and the statistical information derived from it statistically significant to the level desired (e.g. β=95% γ=95%) with the sample sizes obtained from the application of the formula.

  Other methods of sampling, such as Latin Hypercube Sampling (LHS), commonly employed in Monte Carlo techniques in order to reduce the variance of the final result by producing a uniform covering of the sampling space, is not proven to satisfy the mathematical requirements of the formula.

- Truncation of Sampling Space

  The sampling intervals of the variables and code physical model predictions must account for the physically or operationally meaningful variation. For instance, in case of uncertainty quantified by normal PDFs the sampling interval can usually be truncated at $\mu \pm 3\sigma$, with μ being the average value of a variable or the 'bias' of a given code physical model's prediction. The reason for choosing 3σ as the cut-off value is to cover as much variation as possible, while, at the same time, limiting the maximum and minimum values, so that the samples are not distorted by extreme values, which, in many cases, would simply not be physically or operationally feasible to reach.

- Dependence between input variables

  The dependences between system variables and between code physical models must be considered.

  - Physical dependencies can be included in the form of functions or experimentally obtained values for the dependent variables, from which a form of correlation can be derived.
  - If some variables considered during the identification phase of the methodology ar e found to be strongly dependent (e.g. correlation coefficients above 0.9), they can be merged in a single

variable obtained by the composition of those mutually dependent, or their influence in the code´s results represented by just one of them.

- Effect of the Size Sample in the Results of the Application of the Wilks' Formula

When the sample generated is larger than that needed for the statistical significance of the tolerance interval (e.g. a sample size 150, larger than the 93 needed for a β= 95%, γ= 95%), several options are available according to the application of the formula. One can obtain a higher confidence level γ, interval coverage β, or maintaining $\gamma$and β, select the maximum and minimum limits of the tolerance interval with a lower variance. In this latter case, instead of the maximum and minimum of the ranked output variable values, one can select the second maximum or minimum, or the third, and so on, depending on the sample size according to the Wilks´s (Noether´s) formula. It is mathematically proven that these "internal" values suffer form a lower variance in their prediction of the β-quantiles of the actual *pdf* defining the stochastic behavior of the output variable (see 7.2.6).

The reason for choosing a larger sample size is twofold. On one hand, one can refine the uncertainty quantification after the propagation process has been carried out if needed. On the other hand, one can still have a large enough sample if some of the code runs should fail. The problem of what to do with failed runs in the final determination of the uncertainty is still a matter of discussion amongst analysts.

The issue of considering failed runs has been addressed from the point of view that, for complete uncertainty evaluations, completed samples should be used. However, it is necessary:

- To make sure that the discarded runs produced values for those variables that were able to predict, which fall within the tolerance bands computed with the completed runs. If this is not the case, then those samples outside the bands should be included in the determination of new, wider tolerance bands.

- To identify, if feasible, the causes for the failure, and to determine which parameters caused the problems encountered by the code, i.e. the sampled values used in the failed run.

- For those parameters that cannot be computed by the code, because it failed before they were reached, e.g. time to complete the quench in case of a LOCA, it is not possible to include them in the sample of output values of such variable, and they must be then discarded to compute uncertainty and sensitivity measures, since the code provided NO information that can be used about the output.

It is then that the results of all code executions, as well as the input files and sample files created, must be stored for later review if necessary. An uncertainty propagation module must carry out such an auditing procedure automatically.

Finally, a larger sample size usually results in more statistically accurate sensitivity analyses, because the increased number of output-input value pairs is increased, thus reducing the possibility of spurious correlations.

5.2.4 Calculation Process

With the n sets of sampled values of input variables and code models, the computer code is executed n times, each one with a different randomly chosen set of sampled values, and the results recorded to form a sample of size n of the code results.

During the execution of the code, 'unusual' combinations of input values, or too wide variation ranges for some of them, may result in code failure. These calculations must then be discarded, and the sample size is consequently reduced. This is another reason why the size of the sample should be as large as possible to allow for some loss of output information caused by code failure. However, all code calculations that complete successfully should be included in the statistical process to obtain the tolerance intervals, regardless of the output values. This would maintain the random character of the sample of output variables, which would be hampered were these calculations discarded. They represent possible combinations of values of input and model variables that a user may choose for the analysis, and as such are possible outcomes of the code calculations. However, a failed code calculation would not be used in the analysis, since it would produce no results. This somehow justifies that failed code calculations are discarded from the statistical treatment of the output results.

The calculation process must be automated to be computationally efficient, that is, the code must be able to retrieve the values from the sample of input variables in the order in which they were generated $(x_1, x_2,\ldots, x_k)_1$, $(x_1, x_2,\ldots, x_k)_2,\ldots, (x_1, x_2,\ldots, x_k)_n$, and execute the simulation n times (the sample size), each one of them with a set $(x_1, x_2,\ldots, x_k)_i$. This will generate a sample of size n for l output variables of interest that can then be processed statistically: $(y_1(t), y_2(t),\ldots, y_l(t))_{i,i=1,n}$. In general the variables can be (or not) a function of time t or any other variable that one takes as index. The collection of the values of the output variables must also been extracted automatically and preserved after each code execution to be used in the assembly of the sample of output variables.

5.2.5 Processing of the Sample of Code Results

It has been already mentioned that the stochastic nature of the uncertainty in the input variables is propagated to the output results of the calculations and makes them behave as random variables. The study of the uncertainty information contained in the sample of code results can be carried out by means of the techniques described above. They can yield *tolerance limits* of the desired confidence and probability content that permit quantifying statements about the uncertainty associated with the code output results. In addition, visual inspection and analysis of the graphs containing the results for a given output variable $Y_m(t)$ of all the code executions, e.g. $(y_m(t))_1,\ldots, (y_m(t))_n$, for $1 \leq m \leq l$, may also provide information regarding the influence certain input variables or code models on this output variable, e.g. system pressure, clad temperature, etc.

In this respect, two very useful statistics (functions of an output random variable) that can summarize the uncertainty information of an output variable are the mean (for the case the output is a function of time),

$$m_{y_m}(t) = \frac{1}{n} \sum_{i=1,n} (y_m(t))_i \ , \tag{21}$$

and the standard deviation,

$$s_{y_m}(t) = \left(\frac{1}{n+1} \sum_{i=1,n} \ ((y_m(t))_i \ - \ m_{y_m}(t))^2 \right)^{1/2} . \tag{22}$$

Together with the **tolerance limits** and the **nominal case**, $(y_m(t))_0$, i.e. the case with the nominal values of the input parameters, these are the basic measures used to quantify the uncertainty in the code results.

5.2.6 Summary of Information and Statements on Uncertainty

The final phase of the methodology involves the interpretation and the presentation of the code results, qualified by the measures of uncertainty and sensitivity calculated in the previous phases. The statistical basis of the techniques employed in the analysis provides confidence in the statistical interpretation of the results if the assumptions and conditions needed for the application of these techniques are met, e.g. random sampling, correct treatment of input dependencies, low correlation among the input variables, etc. Most of these conditions have been introduced and discussed in the previous sections. The module should be able to provide a graphical summary of the analysis results.

Two types of uncertainty evaluations are feasible in case of a transient simulation, namely a scalar determination (also known as "index independent") and a determination depending on the variation of a given variable (the "index") as the transient progresses. Time is usually, but not necessarily, such a variable.

Time dependent and scalar uncertainty make use of similar uncertainty measures based, for instance, on Tolerance Intervals. They are defined for a given sample of a random variable $\boldsymbol{X}$ of size N, $X^{(i)}$ $i=1...N$, as that interval in which there is probability at least $1-\gamma$ that at least a fraction β of the population lays between $X^{(r)}$ and $X^{(N+1-m)}$ inclusive, that is

$$\mathrm{P}(X^{(r)} \leq \textit{at least a fraction } \beta \textit{ of the population} \leq X^{(N+1-m)}) \geq 1 - \gamma.$$

From this definition it is clear that the limits of the Tolerance Interval are actually estimates of the 1-β (minimum) and β (maximum) quantil of the actual *pdf* of $\boldsymbol{X}$.

When $X^{(0)}$ and $X^{(N+1)}$ are considered as $-\infty$, $+\infty$ respectively, then $r = 0$ corresponds to the upper one-sided tolerance limit, and $m = 0$ corresponds to the lower one-sided tolerance limit.

Application of the Noether´s formula (Conover 1980) which is applicable to both two-sided and one-sided tolerance intervals,

$$\gamma \geq \sum_{i=0}^{r+m-1} \binom{N}{i} (1-\beta)^i \, \beta^{N-i} \ .$$

(20)

If in Eq. (20) neither of the values of r and m is zero, its solution determines the minimum sample size N for two-sided tolerance intervals of the form $[X^{(r)}, X^{(N+1-m)}]$ given β and γ. If $r = 0$, the solution of the equation gives the sample size for the upper tolerance interval $[-\infty, X^{(N+1-m)}]$, and, finally, if $m = 0$, one obtains the size of the sample for the lower tolerance limit $[X^{(r)}, +\infty]$.

This equation also determines the values of m or r (depending of the type of interval selected) for a given sample size; that is, from the ordered output sample, it selects the values of the limits of the one-sided tolerance intervals that satisfy β and γ.

5.3 Sensitivity Analysis of the Output Variables

In addition to uncertainty information, the application of the methods presented in Section 7 offers useful information about the sensitivity of the code results with respect to the input variables and code physical models. These techniques compute several measures of sensitivity and correlation that can also be used to rank the input parameters according to their importance with a given output variable of interest.

When the computer model is a time dependent simulation, the sensitivity of a certain code result to the input and code model parameters may vary with time. This is clear if one thinks of the different physical processes that usually dominate during different parts of the simulation of complex phenomena, e.g. a nuclear plant transient simulation. The possibility of computing sensitivity measures based on random samples of input variables and code models and on code results at regular time intervals allows us to study these variations in sensitivity and gain a more clear understanding of the computer model and the role that each parameter plays in a given computer simulation.

Finally, the ranking of the input variables and code models according to the values of their sensitivity measures may help to identify those with little relevance to the specific scenario, thus focusing resources on those input variables and code models that play a leading role. These studies could, for instance, identify those input variables and code models for which better quantification of uncertainties are needed, e.g. more accurate ranges of variations and PDFs.

5.3.1 Statistical Sensitivity Measures

The utilization of statistical sensitivity measures makes it feasible to rank input variables and code models by their importance with respect to the uncertainty in the desired code outputs, while being much less computationally intensive than local measures.

Similarly to the determination of uncertainty, in the calculation of sensitivity measures, one can also calculate them as index-dependent and as scalar. In the first case, the output variable is assumed as varying with the index variable, e.g. time, and in the scalar case the output variable is takes at a given value of the index, e.g., at a given time, or it represents a maximum or a minimum value for the entire simulation, e.g. maximum clad temperature. Index-dependent sensitivities can provide valuable information about the influence of specific input variable or physical models in the simulation of events at different stages in a transient, e.g. during blowdown, reflood in a LOCA, etc.

For both index-dependent and scalar sensitivity analyses, two global measures of correlation can be employed as global sensitivity measures, namely the Pearson's Ordinary Product Moment correlation coefficient, and the Partial Correlation Coefficient (PCC) of the Pearson's Ordinary Product Moments.

These measures are most effective in determining the relationship between an input variable and a given code output when the relation is ***approximately linear***. If the relation is monotonic but not necessarily linear, a rank based correlation coefficient such as Spearman's would be more effective.

Pearson's Ordinary Product Moment Correlation

This correlation provides a measure of ***linear*** association between two variables based on the statistical information from samples of the input and output variables.

This measure can also be applied to the sample of input variables and code model parameters in order to identify spurious or real correlations between them resulting from the sampling process, to the pair input variable-output, or even to a pair of output variables with samples generated during the uncertainty calculations. The correlation between two inputs x_i and x_j from a given sample of size N is expressed as

$$c_{ij} = \frac{\sum_{n=1}^{N} (x_{in} - m_i)(x_{jn} - m_j)}{(\sum_{n=1}^{N} (x_{in} - m_i)^2 \sum_{n=1}^{N} (x_{jn} - m_j)^2)^{1/2}} \quad , \tag{23}$$

where m_i and m_j are the sample means for x_i and x_j respectively.

The correlation coefficients between an input x_j and an output y_k of a sample of size N can be computed by

$$c_{ik} = \frac{\sum_{n=1}^{N} (x_{in} - m_i)(y_{kn} - m_k)}{(\sum_{n=1}^{N} (x_{in} - m_i)^2 \sum_{n=1}^{N} (y_{kn} - m_k)^2)^{1/2}} \quad . \tag{24}$$

The correlation coefficient c_{ik} takes values in the interval $[-1, 1]$. A perfect correlation will be given by $|c_{ik}| = 1$, whereas a complete lack of correlation is indicated by $c_{ik} = 0$. The sign of the coefficient indicates an increasing (+) or decreasing (-) relation.

When there is no correlations among the inputs, the Pearson's Correlation Coefficients between the output y_k and the input x_i, denoted as c_i, is rigorously equal to the Standardized Regression Coefficient b_i^*, deduced from the coefficients of the multi-linear regression b_i by the relationship: $b_i^* = b_i \dfrac{\sigma(x_i)}{\sigma(y_k)}$. In this relationship, b_i^* is dimensionless (like c_i) whereas the dimension of b_i is the same as that of $\dfrac{\partial y_k}{\partial x_i}$.

The correlation between two input variables depends ***only*** on the points selected from the sample space. If the correlations are not related to any real connection between the variables, they are called 'spurious' and are an artifact of the sample technique. Thus, the use of Latin Hypercube Sampling (LHS) yields samples which can present large 'spurious' correlations between input variables (± 0.8) and it introduces biases in the uncertainty propagation which must be avoided. In general, sample sizes larger than 25 obtained with Simple Random Sampling, or larger than the number of variables considered, will not present large spurious correlations which could render these samples unfit for their application to uncertainty propagation analyses.

The correlation between an input variable and a code output depends mainly on the sample points of the variable, and on the code ***Code(·)*** which produces the output. In a real application the value of an output variable will be influenced by many input variables, and this influence will affect the 'resolution' of the global sensitivity measure, that is, the correlation coefficient. In order to 'isolate' the individual effect of an input variable on a given code output, the so-called Partial Correlation (PCC) can be employed.

Partial Correlation of the Pearson's Ordinary Product Moments

The Partial Correlation Coefficients (PCC) provides a better measure of linear relationship between two variables, which belong to a set of variables, that is, their values can be influenced by any variable in the set. For instance, in the case of an uncertainty propagation study, the PCC can yield a measure of the linear relation between a given input variable and a code output by statistically ´eliminating´ the influence of all the other input variables. The method relies on the evaluation of the correlation coefficient based on the residuals obtained from the two least-square regressions of the variables being correlated (e.g. two inputs or an input and an output) on the remaining input variables. Such an adjustment eliminates the linear trends in the output variable of interest associated with the other input variables. For a set of K input variables and one output variable of interest, the complete correlation matrix is a $K+1$ x $K+1$ matrix of elements c_{ij} (correlations between input parameters) and the elements $c_{i,k+1}$ for the correlation between each one of the input variables ($i = 1...K$) and the selected output. The partial correlation coefficient between variables i and j, r_{ij}, is calculated with the elements c^{ij} of the inverse of the correlation matrix in the following way:

$$r_{ij} = -\frac{c^{ij}}{(c^{ii}c^{jj})^{1/2}} .$$

(25)

The partial correlation coefficient for the relation between an input variable X_i and an output is given then by:

$$r_{iK+1} = -\frac{c^{iK+1}}{(c^{ii}c^{K+1K+1})^{1/2}} ,$$

(26)

for a sample of K input variables. Note that, while the elements c_{ii}, autocorrelation is 1.0, in general, the same diagonal elements of the inverse, c^{ii}, **are not**.

5.3.2 Multi-linear Regression Sensitivity

The sensitivity measures based on Multi-linear regression are calculated by expressing the 'model', in this case the code, in the form

$$y = \boldsymbol{M}\,\beta + \varepsilon ,$$

(27)

where y is an *N-by-1* vector of observations (output values) for a specific variable of interest, $\boldsymbol{M}$ is an *N-by-P* matrix of regressors (values of the P independent variables, $\boldsymbol{\beta}$ is a *P-by-1* vector of coefficients, and ε is an *N-by-1* vector of random disturbances. This equation is exact, and the solution of the problem is to find an estimate vector $\boldsymbol{b}$ of β. The least squares solution to Eq. (7) is;

$$b = \hat{\beta} = (\boldsymbol{M}^T\boldsymbol{M})^{-1}\,\boldsymbol{M}^T\;y .$$

(28)

Although this equation is used for the development of additional statistical formulas, it has very poor numerical properties, and the application of the multiple regression technique should be performed by using special regression algorithms available in the literature or in mathematical calculation programs, e.g. the `regress` function in MATLAB.

The vector y contains the n output values for the variable of interest and the matrix $\boldsymbol{M}$ is a *K-by-N* matrix containing the N sample values for each of the K independent variables considered. The final result is a K element vector b of regression coefficients, which satisfy Eq. (27) with mean value ε equal to 0.

The function `regress` returns a least-square fit of y on $\boldsymbol{M}$ by solving the linear model of Eq. (27) and the error is distributed according to a standard normal ε=$N(0,\sigma^2)$, using a QR decomposition of $\boldsymbol{M}$ and a least-squares based matrix inversion to solve for b. The matrix $\boldsymbol{R}$ is used to compute confidence intervals.

After b has been obtained, an estimate for y can be computed as

$$\begin{gathered}\hat{y} = \boldsymbol{M}b = \boldsymbol{H}y \\ \boldsymbol{H} = \boldsymbol{M}(\boldsymbol{M}^T\boldsymbol{M})^{-1}\boldsymbol{M}^T\text{,}\end{gathered} \tag{29}$$

and then we can obtain the residuals r, that is, the differences between the actual values of y and those computed from Eq. (29).

$$r = y - \hat{y} = (\boldsymbol{I} - \boldsymbol{H})\,y\,. \tag{30}$$

The residuals correspond to the random errors ε in Eq. (27), and are helpful to assess the adequacies of the model assumptions. These residuals are correlated and have variances which depend on the location of the data points ($\boldsymbol{M}$ matrix values).

These residuals are 'scaled' so that they have all the same variance by turning them into a t-Student variable with $v = N\text{-}P\text{-}1$ degrees of freedom.

$$t_i = \frac{r_i}{\hat{\sigma}^2_{(i)}\sqrt{1-h_i}}\,, \tag{31}$$

and the estimate for the variance of the residuals as

$$\hat{\sigma}^2_{(i)} = \frac{\|r\|^2}{N-P-1} - \frac{r_i^2}{(N-P-1)\,(1-h_i)}\,, \tag{32}$$

with t_i is the scaled residual for the $i\text{-}th$ data point, r_i is the raw residual for the $i\text{-}th$ data point, N is the sample size, P is the number of input variables and model parameters of the model, and h_i is the $i\text{-}th$ diagonal element of $\boldsymbol{H}$. If the model is correct, the residuals should have the same variance, and a hypothesis testing comparing the 'scaled' residuals t_i to the values of the t-distribution can spot those residuals which correspond to model values y_i which are outliers with respect to the rest of the observations that form the sample vector y. In this sense, the calculation of the confidence intervals c_i for the mean error of the residual (according to the assumptions of normality of ε) can also be used to identify outliers when the confidence interval for the $i\text{-}th$ data point does not include zero with a significance probability of α.

$$c_i = r_i \pm t_{\left(1-\frac{\alpha}{2}, \nu\right)} \, \hat{\sigma}_{(i)} \sqrt{1-h_i} \, .$$

(33)

The sensitivity coefficients computed for each of the K variables considered in the uncertainty analysis are the elements of the vector $\boldsymbol{b}$, which have the units of y for normalized input and model variables. This normalization makes all the variables to have a nominal value of one following the expression

$$\overline{x}_i = \frac{x_{i,nominal} + x_i}{x_{i,nominal}} \, .$$

(34)

Finally, those variables which are expressed in % about a mean of *0*, have also been normalized by making the nominal value of $\overline{x}_i$ equal to one

$$\overline{x}_i = 0.01 x_i + 1 \, .$$

(35)

The normalization process makes it possible to use the regression coefficients as sensitivity measures for each variable that can be compared to one another. However, one should also be careful when using this information since it must be assessed together with the R^2 statistic calculated in the regression process, equivalent to the correlation coefficient in single-variable linear regression, which gives a measure of the adequacy of using a multiple linear model to account for the variation of y. In this sense, R^2 values give a measure of the 'percentage' of the variability of y attributable to the linear model selected (Eq. (27)). Values close to *1* mean a strong validity of the model, whereas values close to 0 signal a poor linear representation of the variability of y.

The sensitivity coefficients elements of the vector $\boldsymbol{b}$ are basically the linear derivatives of the output variable with respect to each of the input and code model variables x_i, $\frac{\partial y}{\partial x_i}$, and have the units of y when the x_i are normalized as described above. Their validity as sensitivity coefficients depends strongly on the linear character of the relationship between y and b_i. They are not directly comparable to the statistical correlation coefficients, because these take into account the variation range of the variables in terms of their variances (or other measure of 'spread'). By using the information contained in the standard deviations of y and x_i, $\sigma(y)$ and $\sigma(x_i)$, the linear coefficients b_i can be 'normalized' and made non-dimensional, which results in the standardized regression coefficients (SRC) computed as

$$b_i^* = b_i \, \frac{\sigma(x_i)}{\sigma(y)} \, .$$

(36)

The SRCs are comparable to the Pearson's correlation coefficients, and, depending on the linear character of the relationship, to the Spearman's correlation coefficients.

5.3.3 Spearman's Rank Correlation Coefficients

Spearman's rank correlation coefficients are non-parametric measures of correlation that can assess how well a monotonic function can describe the relationship between two variables, without any assumptions on the type of frequency distribution of the variables. The calculation of the coefficients does not necessitate a linear relationship between the variables or that they are measured on interval scales.

The Spearman's correlation is a special case of the Pearson's product-moment coefficients, but instead of using the actual values of the variables, the sampled data for a given variable $X_i=\{x_{i,n}\}_{n=1...N}$ and the output variable $Y_j=\{y_{j,n}\}_{n=1...N}$ are converted to ranks. Thus, with two ordered samples, one for each variable to be correlated, the differences between the ranks of *corresponding* pair of values of the variable X_i and the output variable Y_j, $(x_{i,n}, y_{j,n})_{n=1,..,N}$, for a sample of size N, are calculated as

$$D_{i,n}=Rank(x_{i,n})-Rank(y_{j,n}),$$

(37)

and the result used to compute the correlation coefficient for the pair (X_i, Y_j), ρ_i for a sample size n

$$\rho_i=1-\frac{6\sum_{n=1}^{N}D_{i,n}^2}{N(N^2-1)} \qquad (18)$$

Partial Spearman's rank correlation coefficients can also be calculated following a similar procedure as that explained above for the Pearson's coefficients.

6. WORK STATEMENT

The purpose of the development of the module is to provide the NRC-codes integrated in the SNAP platform with the capability of performing automated uncertainty and sensitivity analysis within the framework of SNAP.

We propose that this can be achieved by developing a computational and data management module that can be integrated as a part of SNAP, whose functions are accessible transparently for the user as new options are added to the menu of SNAP.

The development of such a tool will require:

- Development and implementation of an efficient software system for data management input processing and presentation integrated in the GUI capabilities of SNAP. Use of intermediate working files may help in the fulfillment of this task, together with standard available procedures for the handling of information stored in data files whether binary or ASCII.
- Development and implementation of the algorithms needed to apply the different statistical functions and calculation procedures described in previous sections, and any new ones proposed by the users of the US-NRC codes that are of interest for the special kind of analysis they may be interested in. Standard numerical algorithms and procedures in FORTRAN or C are available in the literature for many of these tasks, while others will have to be developed or adapted to the peculiarities of the uncertainty analysis methodology proposed.
- Development of an efficient computational flow for the automatic execution of the code in single processor or multiple parallel environment that can keep track of all aspects of the execution process (restarts, failed runs, etc., storage of input and output files, extraction of output data of interest and assembly of the sample out file) and report them for audit purposes.

The complete procedure has already been developed with the support of specific statistical packages for several system codes, and has been applied with success to carry out uncertainty analyses of NPP transients running in both Linux and Windows environments. Thus, the viability of such a proposed system does not need to be proved. However, such applications are not fully integrated in a specific computational environment and the process has not been fully automated. What is needed is the integration of such a capability in the SNAP computational environment according to its particular characteristics, make it fully automatic and transparent to the user, and test the developed product.

7. REQUIREMENTS

This section presents the requirements that a module to perform uncertainty propagation calculations in SNAP should provide.

It is based on the application of a statistically based methodology as the one described in previous sections and follows closely the developments that GRS has made in their own uncertainty calculation tool SUSA.

This section has been structured in a logical process of uncertainty calculation after the steps described in Section 7.1. It is also assumed that the analyst, before setting out to use the SNAP module, will have completed the preliminary, and important, steps outlined in the CSAU methodology and adopted by many analysts. A summary of them:

- Establish that the code is capable of simulating, physically and numerically the transient of interest.
- Establish that the documentation of the code is complete and up-to-date.
- Determination of the most important input variables and code physical models in accordance with the best information available and their own knowledge and consultations with experienced experts (see Figure 2).

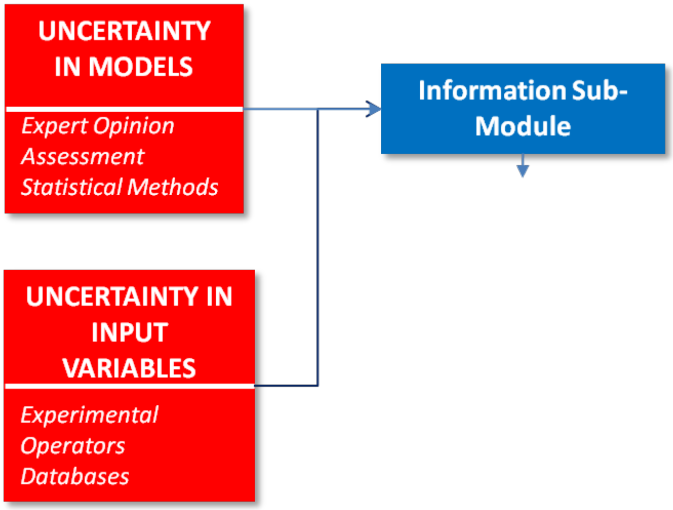

Figure 3. Scheme of the sub-modules with possible expert input.

Quantification of the uncertainty in the input variables and code physical models selected as most relevant. For this step, a considerable effort is usually required, which involves literature research, own assessment studies and also input from experienced users. This process should ideally be a labor of a long term plan for the qualification of the code and its models based in programs for systematic code validation and assessment studies. The results of such studies, with the appropriate techniques (see as an example Vinai, 2007) can be used to create databases of uncertainty quantification for code models whose information can be employed to carry out uncertainty analyses for a wide variety of transients. The flexible character of the statistical methodology proposed can treat each individual model as an independent source of uncertainty. This makes it possible to refine the reliability and accuracy of the uncertainty determinations of the code simulations by progressively improving the databases with the addition of uncertainty information obtained from more assessment studies and from the application of more advanced or refined techniques to extract the uncertainty information from the results of the assessments.

Therefore, after the analyst has completed the previous steps to the best of their capabilities and availability of data, they will be in the position to start the actual uncertainty propagation with the help of the proposed module in SNAP.

7.1 Sub-Module for Information Management

The credibility of the results of an uncertainty propagation analysis relies on the methodology employed and on the quality of the data used to quantify the uncertainty in input variables and code physical models. To guarantee the adequacy of the data used, it is necessary to create a data base that stores this information, processes it, and makes it available to be used in the uncertainty calculation process.

The function of the Sub-module for the management of information aims at fulfilling the requirements of traceability and quality control.

The Sub-Module will have to perform the following actions

- Identification of the variables with a unique name.
- Recording of the uncertainty data sources.
- Differentiate between input variables, as those whose value can be changed in the input file, and code models, require a special procedure to take into account the uncertainty information.
- Assignation of the uncertainty measures:
  - Type of measures.
  - Parameters which define the measures: e.g. mean values, variances, medians, quantiles, etc. The sub-module should also accept the input of empirical PDFs as measures of uncertainty.
  - Range of Variation: maximum and minimum values allowed.
- Identification of dependencies between variables and determination of the statistical (correlation coeff.) or analytical (functions) measures of dependence.

The outcome of this sub-module is an internal database that contains all the information necessary to quantify the uncertainty of input variables and physical models. It should be arranged in data structures similar to records, which can be edited at any time through the SNAP GUI, as new, more precise information becomes available. The database will be accessed by the sample generation sub-module to retrieve the uncertainty measures for each variable together with any dependency, so that the statistical design of the sample can be carried out.

7.2 Sub-Module for Random Sample Generator

After the data has been entered in to the database, the next task is the generation of a sample of the values that the input variables can take according to the uncertainty information. Therefore a sub-module will have to be dedicated to the generation of the sample. Several types of sampling procedures are available, however only two of them have been used for the application of statistical uncertainty propagation methodologies, namely

- **Simple Random Sample**: this is the only sampling procedure that rigorously guarantees the applicability of Noether´s formula and the probability coverage and confidence level of the tolerance intervals obtained. For this reason it is the method that should be used and, therefore its implementation in the sub-module is required.

- **Latin Hypercube Sample**: the applicability of the Noether´s formula and the validity of the tolerance limits obtained are not rigorously proven. However, it has also been used due to the uniform probability coverage that it provides. The sample space is uniformly sampled so that every interval in which one divides the sample space has the same probability of being accessed by the sampling algorithm.

The sampling process will require the use of a good pseudo-random number generator, of which a wide range exists in the statistical literature.

After the sample has been generated, it should be stored in a sample file that can be binary for faster access or in ASCII coding if the file is to be used outside of the SNAP environment. The binary sample file will also be used for sensitivity analysis, since it contains the variation of all the input and model variables, needed for the calculation of the statistical sensitivity measures described in Section 7.

Finally, the generated sample should be analyzed for correlations between the input variables. For this purpose, it is necessary that the sub-module can calculate the Correlation Matrix of the sampled values based on several statistical correlation measures, some of which, Pearson´s and Spearman´s, have already been introduced in Section 7; but additional ones can complement these two.

7.3 Code Execution Sub-module

This sub-module will have to manage the automatic execution of the code as many times as the size of the sample generated by the Random Sample Generation Module.

The automation procedure must be able to:

- Modify the code input file to introduce the modified value (with respect to the nominal value) of the input variables for each sample ´element´.
- Modify the code physical models by modifying the values of certain model´s parameters, for which the uncertainty has been quantified, or the by directly modifying the values yielded by the model when the uncertainty information corresponds to the complete model´s predictions. This should be done during run-time, without the need to re-compile the code every time a new execution is required.
- Manage the execution process so that it is able to handle restarts of the code with changing uncertainty information if some variables important in the restarted simulation were not initially included in the simulation previous to the restart, e.g. a steady state, followed by a restart with a transient that required a reconfiguration of the original components.
- Manage the execution procedure so that successful runs (completed until a certain minimum time) are identified and unsuccessful runs are stored for further inspection, together with the sampled values that initiated them.
- Manage the execution in a sequential or parallel computational environment. In the second case, the module should have the capability of initiating a multi-threaded execution procedure with each thread carrying out an execution for a sample ´element´.
- Extract the output variables of interest from the appropriate output file (xtv in the case of TRACE) and store this data, so that it can be assembled in a file containing the sample of code output variables of interest.

Finally, the module should be able to store the relevant input and output files for each run, and keep a record of the execution process for audit and post-evaluation purposes.

The output of this sub-.module should be:

- The output sample file, which contains the uncertainty information of all the output variables of interest.
- A compressed file containing input and output files for every run, ideally one with successful runs and another with failed runs.
- A file with information that tracks the execution process for audit purposes.

A procedure has to be developed that can eliminate those elements from the input sample file created by the random sampling sub-module, i.e. all the values used for the modification of the input variables and physical models for failed code executions, so that it is eliminated from the subsequent uncertainty and sensitivity calculation process. This is usually done by removing the line in the sample file that contains the information for the failed code execution and shifting the rest upwards. The order in which the sample elements are arranged in the file **MUST** be preserved, after the elimination. This is not important for uncertainty estimation, but it is fundamental for sensitivity calculations.

7.4 Sub-Module for the determination of uncertainty

The calculation of the uncertainty in the selected code results is performed by this sub-module. The main tasks are then:

- Transformation of the output sample file (the values for the output variables will usually be given at different values of the index variables, e.g., different times) for each code execution:
  - Homogenization of the index for all the code executions so that the values for the output variables of interest are all given at the same values of the index.
  - Process of the file for scalar uncertainty: extraction of values for a given output variable at the same values of the index variable, or localization of maximum or minimum values for the variable during the entire transient analyzed.
- Processing of the homogenized output sample file:
  - Calculation of the Tolerance Intervals depending on the number of successful code executions (sample size) by applying Noether´s formula (or the simplified Wilks formula) and solving it for given values of r and m (double or single sided tolerance interval, first or higher order Wilks methods) and for probability content β and confidence level γ.
  - The tolerance intervals can be calculated for:
    - ✓ A series of index values, so that they can show the uncertainty of the output variable of interest as the index changes, e.g. as the transient progresses if the index variable is time.
    - ✓ As scalar uncertainty tolerance intervals.
- Additional uncertainty measures can also be computed based on the processed output sample files:
  - Assume normality in the output variable and calculate mean and standard deviation. To correctly interpret this uncertainty measure a Statistical Normality Test is Required so that the degree of statistical significance to assign normality to the distribution of the uncertainty can be considered as normal.
  - If the number of code executions (sample size) is large enough, an empirical PDF can be extracted from the sampled data by making use of several techniques:
    - ✓ Bayesian Inference. This method requires that the sample fulfills special statistical requirements, and usually assumes a type of distribution in the sampled values.
    - ✓ Non-parametric estimator development that require no a-priori assumptions about the type of distribution and can extract reliable information from relatively low sized samples (Vinai 2007).

The outcome of this sub-module should be:

- A processed output sample file with a homogenized index variable that can be used for the calculation of index-dependent tolerance intervals, or any other uncertainty measure selected by the user. This file should be also offered in ASCII format so that the user can use it with other statistical tools not offered in the SNAP module if necessary.
- A processed output sample file ready for scalar uncertainty analysis according to the user's selection of which type of scalar uncertainty is to be calculated, e.g. Maximum PCT, time for quenching, maximum system pressure, etc. This file should be also offered in ASCII format so that the user can use it with other statistical tools not offered in the SNAP module if necessary.
- Files containing the information about the tolerance intervals calculated for each variable of interest, whether index dependent or scalar. This information should be given in a manner that allows it to be used with other additional statistical tools as given in the form of ASCII files or in a format ready to be plotted by the next sub-module.
- Files containing information on other alternative uncertainty measures, such as estimated PDFs through Bayesian inference or non-parametric estimation and means and variances together with the results of the normality tests if the normality assumption is used.

7.5 Sub-Module for the Processing of Sensitivity Measures

The sub-module for sensitivity analysis will make use of the input sample file, modified to eliminate the entries (lines) for the failed code execution, which was created by the random sampling sub-module (see 9.2) and modified, if needed, after the code executions have been completed (see 9.39). The order in which the samples are arranged in this file is fundamental for the correct determination of sensitivity analysis. Therefore, as mentioned above, it should be preserved.

The output sample file, which will lack those failed executions, will be used together with the input sample to calculate the sensitivity measures presented in Section 7. Additional measures could also be implemented in accordance with the needs of future users.

The procedure for the calculation of the sensitivity measures is similar to that followed for the uncertainty determination. Index-dependent and scalar sensitivity measures can be calculated based on the information contained in the homogenized output sample file.

The outcome of this sub-module should be:

- Files containing the information about the statistical sensitivity measures calculated for each variable of interest, whether index dependent or scalar. This information should be given in a manner that allows it to be used with other additional statistical tools as given in the form of ASCII files or in a format ready to be plotted by the next sub-module.
- A file containing in ASCII the final input sample (with the failed executions eliminated), a file containing the index-homogenized sample of output variables of interest, and a file

containing the index-independent sample of the variables of interest for scalar analysis of sensitivity. This data can then be used to compute other sensitivity measures that the user may need with the help of other statistical packages.

7.6 Sub-Module for Plotting of Uncertainty and Sensitivity

The sub-module for plotting should be able to generate plots for the output variable of interest that show:

- Index-dependent uncertainty: in the form of index dependent tolerance intervals, together with maximum, mean, median, nominal and minimum values.
- Scalar uncertainty:
  - Tolerance Intervals with maximum, mean, median, nominal and minimum values.
  - Variation intervals and results of normality test in the case where the uncertainty in the variable is assumed to follow a normal distribution.
  - Empirical PDFs obtained with Bayesian inference: plot and ASCII data.
  - Empirical PDFs obtained with non-parametric estimators: plot and ASCII data.
- Plots of all the results for the code executions for the output variable of interest.
- Plots of the PDFs assigned to quantify the uncertainty in input variables and code physical models.
- Scatter plots of the sample values for each input variables and code physical models.
- Plots of the correlation matrices selected to check the input sample files for spurious or real (enforced) correlations between the input variables and code physical models.

All these plots can easily be generated based on the information contained in the working files that must be generated by each sub-module described above. It is then essential to keep a detailed book-keeping and an efficient management of the information that is being produced at each stage of the uncertainty and sensitivity calculation process.

All these files should be stored for each anaylsis for future access by the same SNAP module. It is then advisable to treat each calculation as a `Project´, which can be managed and accessed as a whole by the uncertainty module in SNAP to continue the post-processing if needed.

8. REFERENCES

Barré, F., de Crécy, A., Perret, C., 1995, 'The Sensitivity Analysis by Adjoint Method for the Uncertainty Evaluation of the CATHARE Code', Proceedings of NURETH-7, Vol. 3, pp. 2045-2065, New York, USA.

Boyak B.E., et al., 1990, 'Quantifying Reactor Safety Margins, Part : 1 An Overview of the Code Scaling, Applicability, and Uncertainty Evaluation Methodology', *Nuclear Engineering and Design*, **119**, 1-15.

Cacuci, D.G., 1988, Uncertainty Analysis, Ch. 3, Ronen, Y. Editor, CRC Press, Florida, USA.

Conover, W.J., 1980, Practical Nonparametric Statistics, 2nd Ed., Wiley Ed., New York.

Crow, E.L., Davis, F.A., Maxfield, M.W., 1960, Statistics Manual, Dover, New York.

D'Auria, F., 1998 in Uncertainty Methods Study for Advanced Best Estimate Thermal-hydraulic Code Applications, Vol. 2, Report from the Uncertainty Analysis Group, CSNI.

Downing, D.J., Gardner, R.H. and Hoffman, F.O., 1985, 'An Examination of Response-Surface Methodologies for Uncertainty Analysis in Assessment Models', Technometrics, **27**, 151.

Draper, N.R. and Smith, H., 1981, Applied Regresion Analysis, 2nd Ed., Wiley, New York.

Glaeser, H., et al., 1998, in Uncertainty Methods Study for Advanced Best Estimate Thermal-hydraulic Code Applications, Vol. 2, Report from the Uncertainty Analysis Group, CSNI.

Glaeser, H., Hofer, M., Kloos, K. and Skorek, T., 1994, Unsicherheits- und Sesitivitätsuntersuchungen von Analysen mit Thermohydraulik-Recehenprogramme, GRS Abschlussbericht, GRS-A-2177.

Hofer, E., Karwat, H. and Werner, W., 1992, 'Comments on Quantifying Reactor Safety Margins', *Nuclear Engineering and Design*, **132**, 411-414.

Hölmström, H., 1992, 'Quantification of Code Uncertainties', *Proc. Of CSNI Specialists Meeting on Transient Two-Phase Flow*, Aix en Provence, France.

Kolmogorov, A.N., 1933, 'Sulla Determinazione Empirica di una Legge di Distribuzione', *Giornalle dell' Instituto Italiano degli Attuari*, **4**, 83-91.

Lellouche, G.S, and Levy, S., 1990, 'Quantifying Reactor Safety Margins, Part : 4 Uncertainty Evaluation of LBLOCA Analysis based on TRAC-PF1/MOD1', *Nuclear Engineering and Design*, **119**, 67-95.

Lillefors, H.W., 1967, 'On the Kolmogorov-Smirnov Test for Normality with Mean and Variance Unknown', *Journal of the American Statistical Association*, **62**, 399-402.

Mandel, J., 1984, The Statistical Analysis of Experimental Data, Dover Ed., New York.

McKay, M.D., 1988, Uncertainty Analysis, Ch. 4, Ronen, Y. Editor, CRC Press, Florida, USA.

Myers, R.H., 1971, Response Surface Methodology, Allyn and Bacon, Boston.

Noether,, G.E., 1967, Elements of Nonparametric Statistics, Wiley Ed., New York, (2.4, 3.3, 5.1, 5.7, 5.8, 5.9).

Ronen, Y., 1988, Uncertainty Analysis, Ch. 1, Ronen, Y. Editor, CRC Press, Florida, USA.

Shapiro, S.S., Wilk, M.B. and Chen H.J., 1968, 'A Comparative Study of Various Tests for Normality', Journal of the American Statistical Association, **63**, 1343-1372.

Smith, D.L., 1994, Probability, Satistics and Data Uncertainties in Nuclear Science and Technology, OCDE-American Nuclear Society.

Vinai, P., 2007, Development and Application of Objective Uncertainty Measures for Nuclear Power Plant Transient Analysis, Thesis No. 3897, EPFL, Switzerland.

Wickett, T. (Ed.), 1998, Uncertainty Methods Study for Advanced Best Estimate Thermal-hydraulic Code Applications, Vol. 1, Report from the Uncertainty Analysis Group, CSNI.

Wilks, S.S., 1962, Mathematical Statistics, Wiley, New York, (3.3).

Wulff, W., 1990, 'Quantifying Reactor Safety Margins, Part : 3 Assessment and Ranging of Parameters', *Nuclear Engineering and Design*, **119**, 33-65.

Yadigaroglu, G., 1992, 'Comments on Quantifying Reactor Safety Margins', *Nuclear Engineering and Design*, **132**, 425-429.

NRC FORM 335
(9-2004)
NRCMD 3.7

U.S. NUCLEAR REGULATORY COMMISSION

BIBLIOGRAPHIC DATA SHEET

(See instructions on the reverse)

1. REPORT NUMBER
(Assigned by NRC, Add Vol., Supp., Rev., and Addendum Numbers, if any.)
NUREG/IA-0407

2. TITLE AND SUBTITLE
Proposal for the development and implementation of an uncertainty and sensitivity analysis module in SNAP

3. DATE REPORT PUBLISHED

MONTH	YEAR
April	2012

4. FIN OR GRANT NUMBER

5. AUTHOR(S)
Prof. Dr. Rafael Macián-Juan, Dr. Wolfgang Tietsch*, Dr. Felix Sassen*

6. TYPE OF REPORT
Technical

7. PERIOD COVERED *(Inclusive Dates)*

8. PERFORMING ORGANIZATION - NAME AND ADDRESS *(If NRC, provide Division, Office or Region, U.S. Nuclear Regulatory Commission, and mailing address; if contractor, provide name and mailing address.)*

Lehrstuhl für Nukleartechnik
Technische Universität München
Boltzmannstr. 15
85747 Garching, Germany

*Westinghouse Electric Germany GmbH
Dudenstrasse 44
68167 Mannheim, Germany

9. SPONSORING ORGANIZATION - NAME AND ADDRESS *(If NRC, type "Same as above"; if contractor, provide NRC Division, Office or Region, U.S. Nuclear Regulatory Commission, and mailing address.)*

Division of Systems Analysis
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, D.C. 20555-0001

10. SUPPLEMENTARY NOTES
A. Calvo, NRC Project Manager

11. ABSTRACT *(200 words or less)*

This report presents a proposal for the implementation of a module in the SNAP program that permits uncertainty propagation in the codes which it supports. In particular, as a first application, it suggests that the module be developed and tested with TRACE. Once the functionality and applicability have been tested, an extension to the other codes could be carried out with relatively small additional effort. The module in SNAP should allow for both propagating uncertainty from model input variables and physics models implemented in the code. Furthermore it should be capable to support sensitivity analysis with respect to model input parameters.

The report is structured in several sections that give an overview of the most commonly used uncertainty propagation methodologies today. The review performed leads to the conclusion that the most promising methodology for implementation in SNAP is, in the opinion of the authors, the one based on a statistical computational framework. For this reason, the report is focused on the proposal of an uncertainty module for SNAP that can successfully and efficiently carry out uncertainty propagation for the codes that it supports based on the statistical methodology developed by LANL and GRS.

12. KEY WORDS/DESCRIPTORS *(List words or phrases that will assist researchers in locating the report.)*

RELAP5
SNAP
uncertainty
sensitivity
CAMP (Code Application Maintenance Program)
TRACE
Germany
LANL
GRS
Germany

13. AVAILABILITY STATEMENT
unlimited

14. SECURITY CLASSIFICATION
(This Page)
unclassified
(This Report)
unclassified

15. NUMBER OF PAGES

16. PRICE

Printed
on recycled
paper

Federal Recycling Program

UNITED STATES
NUCLEAR REGULATORY COMMISSION
WASHINGTON, DC 20555-0001

OFFICIAL BUSINESS

NUREG/IA-0407

Proposal for the Development and Implementation of an Uncertainty and Sensitivity Analysis Module in SNAP

April 2012