MelSUA – An Open-Source MATLAB Toolbox for Sensitivity and Uncertainty Analysis with MELCOR Code

Piotr Darnowski, Piotr Mazgaj
Institute of Heat Engineering, Warsaw University of Technology,
Nowowiejska 21/25 Room 113, 00-665, Warsaw, Poland
piotr.darnowski@pw.edu.pl, piotr.mazgaj@pw.edu.pl

ABSTRACT

The MelSUA (MELCOR Sensitivity and Uncertainty Analysis) is an open-source tool dedicated to sensitivity and uncertainty (S&U) analysis with MELCOR computer code. The tool was implemented as a toolbox for the MATLAB computing environment.

This paper presents the methodology, development process, implementation, and code structure. Additionally, basic usage, guidelines, and example input and example results are presented. The basic aim of the software is to support MELCOR analysts familiar with MATLAB in performing sensitivity and uncertainty analysis for severe accident studies in Nuclear Power Plants. The main advantage of the MelSUA is the use of a popular MATLAB environment with powerful statistical toolboxes, which allows data analysis in an easy manner with matrix-based mathematical functionality and advanced post-processing. At the current stage of development, the tool allows performing Monte Carlo or Wilks-based uncertainty analysis using Simple Random Sampling or Latin Hyper Cube sampling for parameters with various continuous and discrete probability distributions. The tool is equipped with routines for sensitivity analysis with linear regression and simple correlations like Spearman or Pearson. It works with XML or MAT input files, automatically processes input decks, and generates ready-to-run execution scripts for Microsoft PowerShell.

Currently, the early version of the tool is available on the GitLab repository (gitlab.com/darczu/x-core/-/tree/master/Modules/MLC_package/SenstivityUncertanityAnalysis).

The tool was developed as an additional contribution to the NARSIS (New Approach to Reactor Safety Improvements) Horizon 2020 research project.

Keywords: *MELCOR, open-source, sensitivity, uncertainty, severe accident, MATLAB*

1 INTRODUCTION

Sensitivity and uncertainty analysis (S&UA) is a common approach used in reactor safety studies. It is especially the case for a design basis type accidents or transients, investigated with thermal-hydraulic system codes. Uncertainty evaluation is a standard part of state-of-the-art deterministic safety analysis like the BEPU approach [1]–[3]. In the field of severe accident analysis, it is less common, as integral tools like MELCOR or ASTEC are more computationally expensive. In spite of that, recently, it has become an active research topic with several research reports applying various quantification methods [4]–[11]. Uncertainty quantification for severe accident modeling can be considered to be even more important than for design basis analysis because uncertainties are much greater. Basically, the phenomena are more complex, with multiple time scales, and substantially less understood phenomenology. In parallel, sensitivity analysis is desired as it allows understanding which modeling parameters or phenomena are important and is commonly performed as an addition to uncertainty study.

This work is a part of the wave of activities related to the use of severe-accident integral codes with S&U methods. The MelSUA tool described in this paper was developed as an alternative to various tools available for performing S&UA with MELCOR computer code [12], [13]. A very popular is DAKOTA, developed by Sandia National Laboratories, which is used by many researchers as it is coupled with SNAP environment for U.S. NRC codes [14], and it covers not only MELCOR but also TRACE, RELAP5, PARCS, and others. For the ASTEC code, there is a dedicated SUNSET module developed by IRSN [15]. Recently, RAVEN was coupled with MELCOR code, applying Python environment [7]. In the past, a dedicated software, MELCOR Uncertainty Engine, was available [16], but it is currently no more available for code users, and it has some important limitations like no distribution truncation, a limited number of distributions, and others. There are other tools dedicated for more general use (non-nuclear), which can be used for S&UA in the nuclear field; an example is a popular UQLab Toolbox for MATLAB [17].

In spite of the availability of the aforementioned tools, the Authors decided to develop their own software dedicated to MELCOR. It allowed us to fully control the process, learn methods by development, use powerful capabilities of statistical toolboxes and customize input, output, and post-processing. The motivation was to automate the process from the initial development of the input deck up to the batch of calculations. The following paper describes the software development status, its capabilities, example application, and plans for the future. The practical application of the software for the analysis of in-vessel hydrogen production is presented in [8] and in the last chapter.

2 MELSUA – METHODOLOGY

2.1 General Methodology

The MelSUA code general workflow is presented in Figure 1. It is also the realization of the S&UA methodology for a severe accident analysis applied in the NARSIS project [8]. The approach is similar to the SNL methodology [16], which utilizes the MELCOR uncertainty tool or RAVEN-MELCOR code coupling methodology [7]. The most important difference is that it uses a MATLAB environment with its toolboxes, also XML or M-file input files, and Power-Shell scripts for batch running.

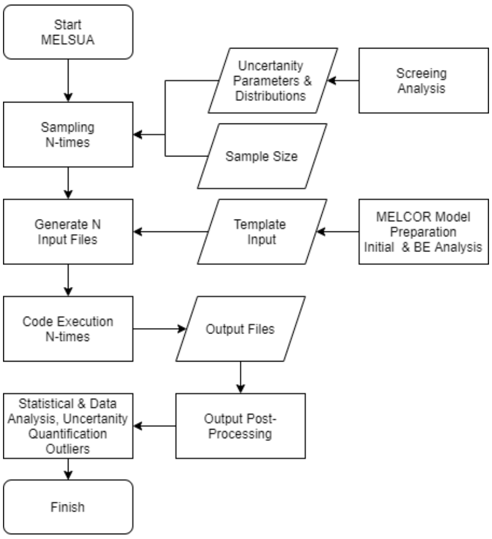


Figure 1: MelSUA code workflow and S&UA methodology. Based on [8].

The code currently allows Monte-Carlo or Wilks type uncertainty approaches, likely the two most simple and popular in nuclear safety applications. For sensitivity analysis, it uses simple statistical correlation indicators Pearson, Spearman, and Kendall coefficients, where the first two are very common in practical applications due to simplicity [18].

The framework is divided into three main steps (modules), which are executed separately: Step 1 – Pre-processing module, which covers template input processing, uncertainty parameters sampling, and input deck generation; Step 2 – Calculations module, which covers the generation of folders and ready to run Power-Shell scripts; and Step 3 – Post-processing module, which allows results presentation and analysis. In general, each module is divided into several sub-steps, which will be discussed in the following sub-chapters.

2.2 Pre-processing Module (Step-1)

The pre-processing module prepares a master directory with sub-folders for the set of computational cases with sampled values of uncertainty parameters. The structure of the module is presented in Figure 2. First of all, the user has to provide code setup and flags, which includes case name, folder names, locations, and uncertainty input type (XML or M-files); the user can switch on/off plotting, saving, and folder copying.

The user has to select the sampling method, seed for the random number generator, and its type. In order to obtain desired statistical significance, the user has to select the proper size of the sample. For Monte Carlo, it should be a large number (e.g., N=500), and for Wilks type analysis, it has to follow proper sample size selection rules (e.g., N=93 for two-sided first order 95%/95% probability/coverage) [18]–[20].

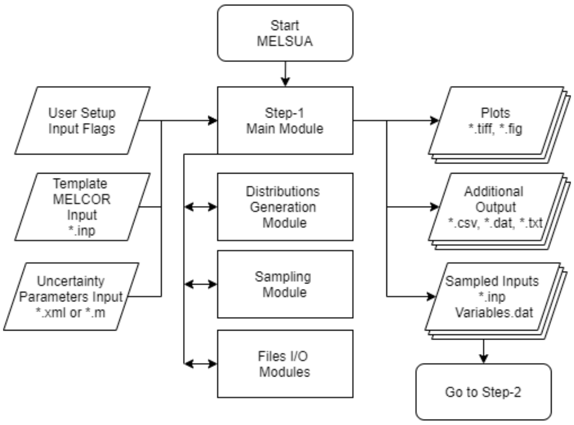
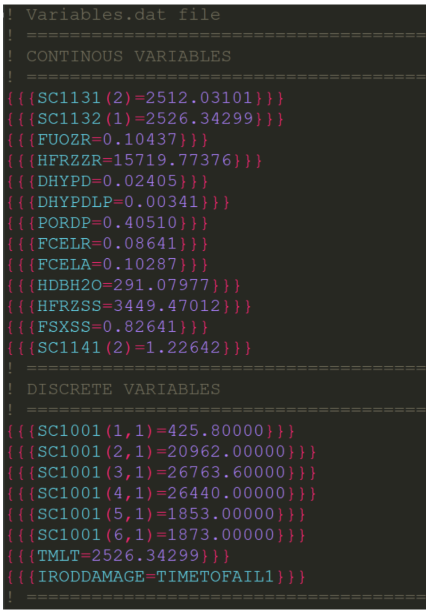


Figure 2: Step-1 Pre-Processing module workflow logic.

Moreover, the user provides the file path and file name of the template MELCOR input deck, which will be copied and used in the generation of the calculation batch. What is important, the MELCOR template input deck must be prepared before analysis, and it will be considered as a reference model.

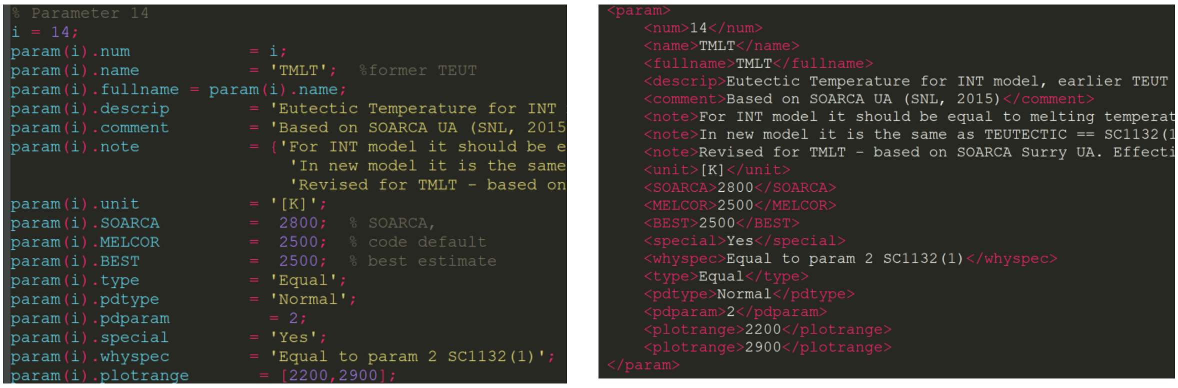
The current version of the MelSUA uses MELCOR variables file card to define the file containing a list of uncertainty parameters (MELCOR field: DefineVariablesFile Variables.dat – see Figure 3). Files are generated by the code for each calculation case. The variable file and the input, use triple brackets (e.g. {{{TMLT=2500.0}}}) to indicate which fields (e.g. field TMLT) should be changed and also to provide default variable values (e.g. 2500). In the template model, the user should indicate all selected variables, as this file is copied and modified by the code (example card: MP_PRC 5600.00 {{{TMLT=2500.0}}} 7.07E5). What is important, in the current version, the default values provided in the template input deck are not used in the final analysis. The best-estimate model (special case - case0) variable file is generated based on the best estimate values of uncertainty parameters provided in the uncertainty input.



```
! Variables.dat file
! ====================================
! CONTINOUS VARIABLES
! ====================================
{{{SC1131(2)=2512.03101}}}
{{{SC1132(1)=2526.34299}}}
{{{FUOZR=0.10437}}}
{{{HFRZZR=15719.77376}}}
{{{DHYPD=0.02405}}}
{{{DHYPDLP=0.00341}}}
{{{PORDP=0.40510}}}
{{{FCELR=0.08641}}}
{{{FCELA=0.10287}}}
{{{HDBH2O=291.07977}}}
{{{HFRZSS=3449.47012}}}
{{{FSXSS=0.82641}}}
{{{SC1141(2)=1.22642}}}
! ====================================
! DISCRETE VARIABLES
! ====================================
{{{SC1001(1,1)=425.80000}}}
{{{SC1001(2,1)=20962.00000}}}
{{{SC1001(3,1)=26763.60000}}}
{{{SC1001(4,1)=26440.00000}}}
{{{SC1001(5,1)=1853.00000}}}
{{{SC1001(6,1)=1873.00000}}}
{{{TMLT=2526.34299}}}
{{{IRODDAMAGE=TIMETOFAIL1}}}
! ====================================
```

Figure 3: Example of the input variables file with sampled parameters. In this example, SC1001 was not sampled.

The user has to provide input data with uncertainty parameters, and there are two options. The first is to use a template M-file with a structure object (INPUT_SUA_Distributions_v2.m). For each parameter, it creates a structure with fields with properties like variability range, distribution, type (discrete or continuous), truncation, and best-estimate value. The template/prototype of the structure is present in the struct_prototype.m file. The second approach is to use an XML file, which has a very similar structure to M-file but is more readable. Example input files are presented in Figure 4.



```
% Parameter 14
i = 14;
param(i).num              = i;
param(i).name             = 'TMLT';  %former TEUT
param(i).fullname = param(i).name;
param(i).descrip          = 'Eutectic Temperature for INT
param(i).comment          = 'Based on SOARCA UA (SNL, 2015
param(i).note             = {'For INT model it should be e
                             'In new model it is the same
                             'Revised for TMLT - based on
param(i).unit             = '[K]';
param(i).SOARCA           =  2800;  % SOARCA,
param(i).MELCOR           =  2500;  % code default
param(i).BEST             =  2500;  % best estimate
param(i).type             = 'Equal';
param(i).pdtype           = 'Normal';
param(i).pdparam            = 2;
param(i).special          = 'Yes';
param(i).whyspec          = 'Equal to param 2 SC1132(1)';
param(i).plotrange         = [2200,2900];
```

```
<param>
    <num>14</num>
    <name>TMLT</name>
    <fullname>TMLT</fullname>
    <descrip>Eutectic Temperature for INT model, earlier TEUT
    <comment>Based on SOARCA UA (SNL, 2015)</comment>
    <note>For INT model it should be equal to melting temperat
    <note>In new model it is the same as TEUTECTIC == SC1132(1
    <note>Revised for TMLT - based on SOARCA Surry UA. Effecti
    <unit>[K]</unit>
    <SOARCA>2800</SOARCA>
    <MELCOR>2500</MELCOR>
    <BEST>2500</BEST>
    <special>Yes</special>
    <whyspec>Equal to param 2 SC1132(1)</whyspec>
    <type>Equal</type>
    <pdtype>Normal</pdtype>
    <pdparam>2</pdparam>
    <plotrange>2200</plotrange>
    <plotrange>2900</plotrange>
</param>
```

Figure 4: Example of the input with uncertainty parameters M-file (Left) and XML (Right).

The selection of the uncertainty parameters with properties (type of probability distribution function (PDF), distribution descriptors) and justification is a huge task itself, and the user has to perform it by proper methodology. The formal approach is to use the PIRT approach; alternatively, for less obvious situations, the user can study literature and use engineering judgment. Also, preliminary sensitivity analysis can be performed for screening analysis (see Figure 1). Users can use any distribution type considered by MATLAB Statistics and Machine Learning Toolbox™; hence several dozen distributions are possible. What is important, in the current version, a separate approach exists for discrete distributions, and it uses piecewise linear distribution, which allows the modeling of uniform and non-uniform discrete distributions.

All collected input data is processed by the main module (Step1_Main_SUA_v3.m, see Figure 2). It calls internal routines for distribution generation (it uses MATLAB probability distribution type objects), sampling routine, and routines for file processing – copying and creating input files, folder structure, and folders themselves.

The sampling routine allows Simple Random Sampling (SRS) or Latin Hyper-Cube Sampling (LHS). Where in general, the user is recommended to use SRS for the Wilks approach and SRS for Monte Carlo if failed runs are expected [8], [21].

This module produces the principal folder with sub-folders for all cases with copies of the master input deck and variables file. Best-estimate case folder (case0) is also created. This module also generates figures with PDFs, Cumulative Distribution Function (CDFs), plotted distributions and samples, and histograms of samples. Examples are presented in Section 3. It creates input echo and data files (CSV and DAT) with sampled variables in various representations, which can simplify eventual analysis.

2.3 Calculations Module (Step-2)

The Step-2 MelSUA execution can be performed separately from Step-1. The structure of the module is presented in Figure 5. The main goal is to generate folders with ready-to-run structure and Power-Shell scripts for batch running. It allows running several PowerShell terminals in parallel and each with a series of runs. Basically, in the input setup to this module (currently Step2_Main_SUA_v2.m), the user provides the desired number of terminals and a total number of cases. Thanks to that user can run multiple cases in parallel, as typically single MELCOR run demands single CPU thread. The user does not have to run all cases generated in Step-1, but can preselect specific cases or sub-batches of cases. It allows splitting analysis into more than one calculation batch and eventually running it on separate workstations.

The user has to provide additional setup, especially the Step-1 results directory and Step-2 directory. The user has to provide a calculations directory (or path) where MELCOR calculations will be performed and the names of a folder for the analysis. Also, the names of MELGEN and MELCOR executables are necessary, and they have to be located in the execution directory.

Then, the Step-2 M-file can be executed. As a result, in the Step-2 working directory, a package (folder) for calculations is generated. It contains copies of sub-folders with input decks and Power-Shell scripts. Then the user can execute MELCOR computations by running the script (melcor_execute_<name of the case>.ps1).

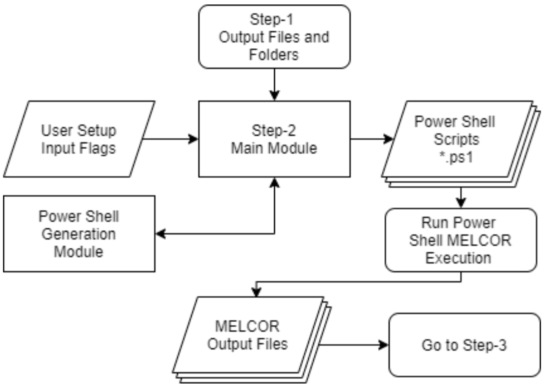


Figure 5: Step-2 Calculations module workflow logic.

2.4 Post-Processing Module (Step-3)

The last step covers the post-processing of results, and the logic of the module is presented in Figure 3. It covers the main module, which reads results generated in Step-2. The module generates plots in the folder with Step-2 results, which contain Figure-of-Merit (FoM) for each run, FoM as a function of time with an indication of failed cases, best-estimate results, mean and median data. It generates box-chart plots with outliers; also empirical CDFs, and swarm plots and allows data normalization. Additionally, it generates horse-tail plots with mean, median, min and max cases;

also, it prepares interpolated data and shaded band plots with 50% and 95% regions. The code allows the comparison of various cases with combined data visualization.

For sensitivity analysis, it generates scatter plots with regression lines and bar plots with Pearson, Spearman, and Kendall correlation coefficients with rho-values and p-values for all considered parameters. Additionally, the module has a procedure to detect failed cases and failure statistics is provided. The module generates results data files (CSV and DAT) with comparisons of sampled parameters and FoM.

Nevertheless, this step is the least developed and demands a larger user experience. However, the reader can use it as a template to develop their own solutions.

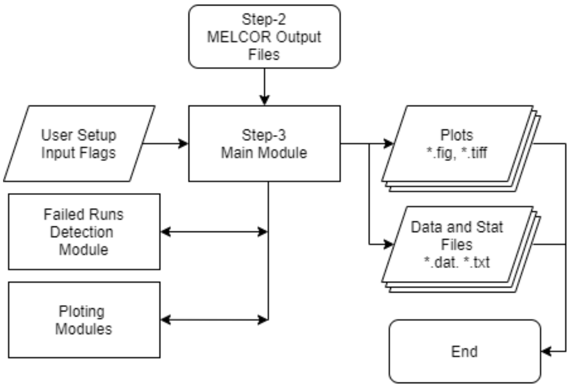


Figure 6: Step-3 Post-Processing module workflow logic.

What is important, at the current stage of development, the code demands the application of the MELCOR EDF package (External Data File). Hence, the template input deck prepared in Step-1 has to contain EDF input cards for FoM.

3 EXAMPLE APPLICATION

Selected results for Wilks type uncertainty calculation are presented in this chapter to indicate MelSUA capabilities. A large Pressurized Water Reactor with unmitigated LB-LOCA with core meltdown was considered with a focus on in-vessel H_2 production. Calculations were prepared using SRS for 120 samples, which corresponds to at least 95%/95% two-sided interval (demands at least 93 runs). Uncertainty parameters are enlisted in Table 1, and PDFs are presented in Figure 7. For details of the model, distributions, and others, the reader is referred to the publication [8] and NARSIS project D4.5 report (waiting for publication).

Table 1: List of uncertainty parameters used in the example analysis.

No	MELCOR Field	Description
1	SC1131(2)	Zircalloy Melt Breakout Temperature
2	SC1132(1)	Fuel Rod Collapse Temperature
3	FUOZR	Fractional Dissolution of Uranium in Molten Zirconium
4	HFRZZR	Candling/Refreezing HTC for Zirconium
5	DHYPD	Debris Diameter in Core Region
6	DHYPDLP	Debris Diameter in Lower Plenum
7	PORDP	Debris Porosity
8	FCELR	Radiation Exchange Factors – Radial
9	FCELA	Radiation Exchange Factors – Axial
10	HDBH2O	In-Vessel Falling Debris HTC
11	SC1001	Zircaloy–Steam Oxidation Correlation
12	IRODDAMAGE	Time-at-Temperature Model
13	TMLT	Interactive Model Melting/Eutectic Temperature
14	HFRZSS	Candling/Refreezing HTC for Steel
15	FSXSS	Fractional Dissolution of Steel Oxide in Molten Stainless Steel
16	SC1141(2)	Maximum Melt Flow Rate after Breakthrough

Results of the uncertainty calculations are presented in Figure 8. Figure 8 (Left) presents FoM vs. time for all results with best estimate results, some indicators like mean and magenta lines showing min/max corresponding to proper Wilks two-sided limits. Figure 8 (Right) presents generated empirical CDF for the FoM with indicated min/max and additional results of Monte Carlo type analysis with N=400 runs. Interestingly for this case, we can observe that Wilks limits are conservative, as expected in comparison to Monte Carlo derived 5% and 95% limits.

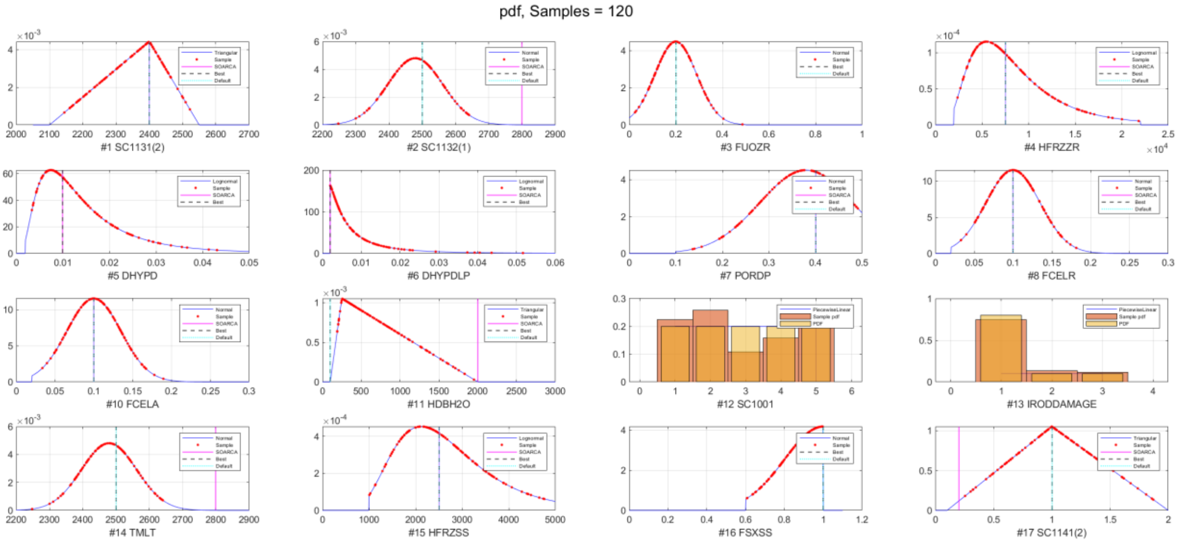


Figure 7: PDFs for studied uncertainty parameters. Includes distribution curves, samples, code default, and best-estimate SORACA values.

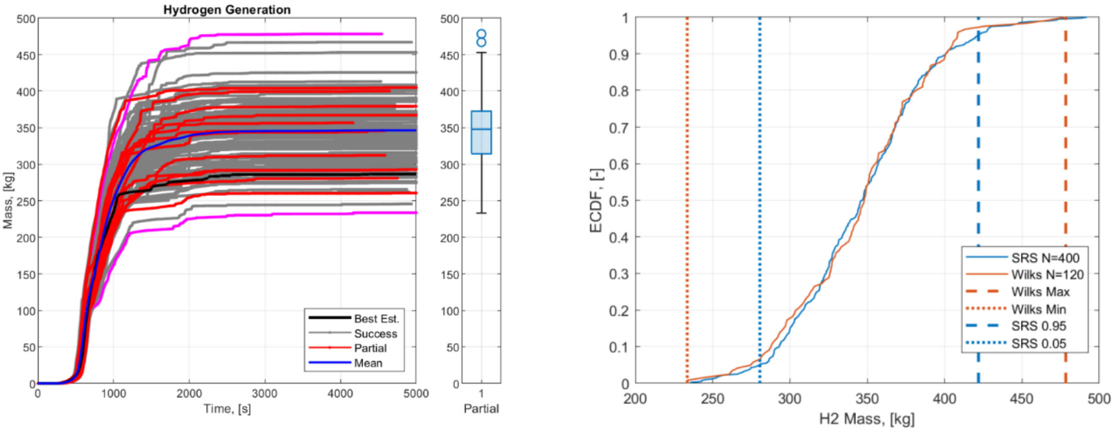


Figure 8: Left – H_2 (FoM) mass vs. time for the example with N=120 runs. Fully successful cases (grey); partially successful cases (red) - converged after the time limit 2500 s; max/min cases (magenta); mean curve (blue); best estimate case (black). Box-chart presents median, outliers, upper, lower quartiles boxes, and non-outlier min/max. Right – Empirical CDFs for Wilks (red) and Monte Carlo with N=400 cases (blue). Min/Max vertical lines (red) for Wilks with 95%/95% interval and 5% and 95% percentiles curves for Monte Carlo (blue).

Figure 9 is an example representation of correlation coefficients for studied continuous uncertainty parameters. For this particular case with ρ-value and p-value, where $|\rho|>0.2$ with $p<0.05$ indicates correlation. For this case, correlation can be indicated for parameters #1, #4, #5, and #7, which were considered in Table 1.

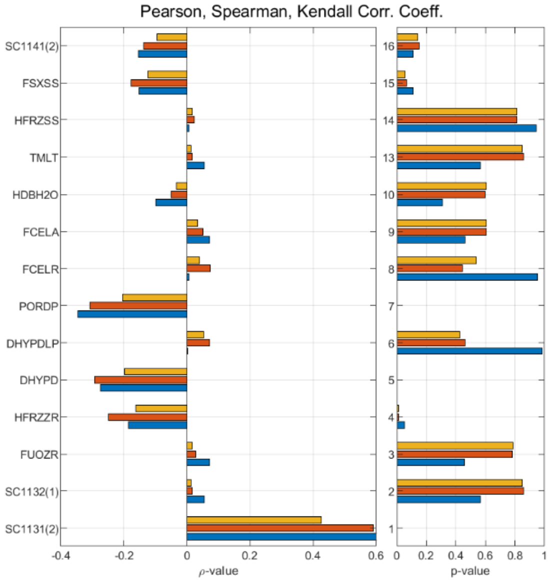


Figure 9: Correlation coefficients, for example, results. Pearson (yellow), Spearman (red), and Kendall (blue) correlation coefficients for MELCOR parameters.

4 CONCLUSIONS AND SUMMARY

The MelSUA software was briefly presented. It is still under development. The source code is open, and it is available at the public GitLab repository: gitlab.com/darczu/x-core/ /tree/master/Modules/MLC_package/SenstivityUncertanityAnalysis

The code's manual was not prepared, but it is considered to be available in the near future. Hence, from the current perspective, it is still internal code, and potential users are encouraged to establish a contact in case of problems.

There are several possible paths for future code development and enhancement. First of all, it should be re-coded with an object-oriented approach with classes to make it more accessible to the user and less prone to mistakes. Additionally, the tool cooperates with Power-Shell, but it can be easily extended to Bash or Batch environments. MELCOR output data is currently processed only for predefined EDF files, and it is an important drawback. It should be extended to allow post-processing of the binary files with AptPlot scripts. Currently, Step-3 post-processing demands more development to make it more automatic. There are several possibilities to develop code beyond standard Monte Carlo or Wilks type uncertainty analysis functionality. Also, the applied statistical analysis approach is one of the simplest, and it can be extended to other approaches thanks to MATLAB functionality. The code has the potential to be also used with different computer codes, and it is likely that in the future, it will be extended to other codes.

The tool's main advantage is the use of a popular MATLAB environment and powerful statistical toolboxes, and it is specifically dedicated to MELCOR. It allows to use of SRS or LHS, truncated distributions, dozens of different distributions, XML input files, and several other features.

ACKNOWLEDGMENTS

The MelSUA code was developed as an additional contribution to the NARSIS research project. This research project has received funding from the Euratom research and training programme 2014-2018 under Grant Agreement No. 755439—project Horizon 2020 NARSIS (New Approach to Reactor Safety Improvements).

REFERENCES

[1] F. D’Auria, ‘Best Estimate Plus Uncertainty (BEPU): Status and perspectives’, *Nuclear Engineering and Design*, vol. 352, p. 110190, Oct. 2019, doi: 10.1016/j.nucengdes.2019.110190.

[2] C. Queral, J. Montero-Mayorga, J. Gonzalez-Cadelo, and G. Jimenez, ‘AP1000® Large-Break LOCA BEPU analysis with TRACE code’, *Annals of Nuclear Energy*, vol. 85, pp. 576–589, Nov. 2015, doi: 10.1016/j.anucene.2015.06.011.

[3] U. S. Rohatgi and J. S. Kaizer, ‘Historical perspectives of BEPU research in US’, *Nuclear Engineering and Design*, vol. 358, p. 110430, Mar. 2020, doi: 10.1016/j.nucengdes.2019.110430.

[4] L. I. Albright *et al.*, ‘Material Interactions in Severe Accidents – Benchmarking the MELCOR V2.2 Eutectics Model for a BWR-3 MARK-I Station Blackout: Part I – Single Case Analysis’, *Nuclear Engineering and Design*, vol. 382, p. 111292, Oct. 2021, doi: 10.1016/j.nucengdes.2021.111292.

[5] S. Galushin and P. Kudinov, ‘Sensitivity and uncertainty analysis of the vessel lower head failure mode and melt release conditions in Nordic BWR using MELCOR code’, *Annals of Nuclear Energy*, vol. 135, p. 106976, Jan. 2020, doi: 10.1016/j.anucene.2019.106976.

[6] Z. Guo *et al.*, ‘Uncertainty analysis of ATF Cr-coated-Zircaloy on BWR in-vessel accident progression during a station blackout’, *Reliability Engineering & System Safety*, vol. 213, p. 107770, Sep. 2021, doi: 10.1016/j.ress.2021.107770.

[7] M. D’Onorio, A. Giampaolo, F. Giannetti, F. Mascari, and G. Caruso, ‘Severe accident sensitivity and uncertainty estimation using MELCOR and RAVEN’, *J. Phys.: Conf. Ser.*, vol. 2177, no. 1, p. 012021, Apr. 2022, doi: 10.1088/1742-6596/2177/1/012021.

[8] P. Darnowski, P. Mazgaj, and M. Włostowski, ‘Uncertainty and Sensitivity Analysis of the In-Vessel Hydrogen Generation for Gen-III PWR and Phebus FPT-1 with MELCOR 2.2’, *Energies*, vol. 14, no. 16, p. 4884, Aug. 2021, doi: 10.3390/en14164884.

[9] M. Spirzewski, P. Domitr, and P. Darnowski, ‘Global uncertainty and sensitivity analysis of MELCOR and TRACE critical flow models against MARVIKEN tests’, *Nuclear Engineering and Design*, vol. 378, p. 111150, Jul. 2021, doi: 10.1016/j.nucengdes.2021.111150.

[10] M. Włostowski, P. Domitr, and P. Darnowski, ‘A Sensitivity Study of Critical Flow Modeling with MELCOR 2.2 Code Based on the Marviken CFT-21 Experiment’, *Energies*, vol. 14, no. 16, p. 4985, Aug. 2021, doi: 10.3390/en14164985.

[11] N. E. Bixler *et al.*, ‘SOARCA uncertainty analysis of a short-term station blackout accident at the Sequoyah nuclear power plant’, *Annals of Nuclear Energy*, vol. 145, p. 107495, Sep. 2020, doi: 10.1016/j.anucene.2020.107495.

[12] Sandia National Laboratories, ‘MELCOR Computer Code Manuals Vol. 1: Primer and Users’ Guide Version 2.2.19018’. SAND2021-0252 O, 2021.

[13] Sandia National Laboratories, ‘MELCOR Computer Code Manuals Vol. 2: Reference Manual Version 2.2.18019’. SAND2021-0241 O, 2021.

[14] E. K. Boafo and E. Numapau Gyamfi, ‘Uncertainty Quantification in Support of Severe Accident Analysis Code User Confidence Using MELCOR-DAKOTA’, *Journal of Nuclear Engineering and Radiation Science*, vol. 8, no. 3, p. 031703, Jul. 2022, doi: 10.1115/1.4053050.

[15] K. Chevalier-Jabet, F. Cousin, L. Cantrel, and C. Séropian, ‘Source term assessment with ASTEC and associated uncertainty analysis using SUNSET tool’, *Nuclear Engineering and Design*, vol. 272, pp. 207–218, Jun. 2014, doi: 10.1016/j.nucengdes.2013.06.042.

[16] R. Gauntt, N. Bixler, and K. Wagner, ‘An uncertainty analysis of the hydrogen source term for a station blackout accident in Sequoyah using MELCOR 1.8.5’, SAND2014-2210, 1200657, 505260, Mar. 2014. doi: 10.2172/1200657.

[17] S. Marelli and B. Sudret, ‘UQLab: A Framework for Uncertainty Quantification in Matlab’, in *Vulnerability, Uncertainty, and Risk*, Liverpool, UK, Jun. 2014, pp. 2554–2563. doi: 10.1061/9780784413609.257.

[18] H. Glaeser, ‘GRS Method for Uncertainty and Sensitivity Evaluation of Code Results and Applications’, *Science and Technology of Nuclear Installations*, vol. 2008, pp. 1–7, 2008, doi: 10.1155/2008/798901.

[19] N. W. Porter, ‘Wilks’ formula applied to computational tools: A practical discussion and verification’, *Annals of Nuclear Energy*, vol. 133, pp. 129–137, Nov. 2019, doi: 10.1016/j.anucene.2019.05.012.

[20] S. W. Lee, B. D. Chung, Y.-S. Bang, and S. W. Bae, ‘ANALYSIS OF UNCERTAINTY QUANTIFICATION METHOD BY COMPARING MONTE-CARLO METHOD AND WILKS’ FORMULA’, *NUCLEAR ENGINEERING AND TECHNOLOGY*, p. 8, 2014.

[21] M. Perez *et al.*, ‘Uncertainty and sensitivity analysis of a LBLOCA in a PWR Nuclear Power Plant: Results of the Phase V of the BEMUSE programme’, *Nuclear Engineering and Design*, vol. 241, no. 10, pp. 4206–4222, Oct. 2011, doi: 10.1016/j.nucengdes.2011.08.019.