OVERVIEW OF THE SAPHIRE PROBABILISTIC RISK ANALYSIS SOFTWARE

Curtis L. Smith[*], Ted Wood, James Knudsen, Zhegang Ma

[*] *Idaho National Laboratory, PO Box 1625 MS 3850, Idaho Falls, ID 83415 curtis.smith@inl.gov*

The Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE) is a software application developed for performing a complete probabilistic risk assessment (PRA) using a personal computer (PC) running the Microsoft Windows operating system. SAPHIRE Version 8 is funded by the U.S. Nuclear Regulatory Commission (NRC) and developed by the Idaho National Laboratory (INL). INL's primary role in this project is that of software developer and tester. However, INL also plays an important role in technology transfer by interfacing and supporting SAPHIRE users, who constitute a wide range of PRA practitioners from the NRC, national laboratories, the private sector, and foreign countries. In this paper, we provide an overview of the current technical capabilities found in SAPHIRE Version 8, including the user interface and enhanced solving algorithms.

I.   BACKGROUND

The Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE) is a software application developed for performing a complete probabilistic risk assessment (PRA) using a personal computer (PC) running the Microsoft Windows operating system. SAPHIRE Version 8 is funded by the U.S. Nuclear Regulatory Commission (NRC) and developed by the Idaho National Laboratory (INL). INL's primary role in this project is that of software developer and tester. However, INL also plays an important role in technology transfer by interfacing and supporting SAPHIRE users, who constitute a wide range of PRA practitioners from the NRC, national laboratories, the private sector, and foreign countries.

SAPHIRE can be used to model a complex system's response to initiating events and quantify associated consequential outcome frequencies (or probabilities). Specifically, for nuclear power plant applications, SAPHIRE 8 can identify important contributors to core damage (Level 1 PRA) and containment failure during a severe accident which leads to releases (Level 2 PRA). It can be used for a PRA where the reactor is at full power, low power, or at shutdown conditions. Furthermore, it can be used to analyse both internal and external initiating events and has special features for managing models such as flooding and fire. It can also be used in a limited manner to quantify risk, using PRA techniques, in terms of release consequences to the public and environment (Level 3 PRA).

In SAPHIRE 8, the act of creating a model has been separated from the analysis of that model in order to improve the quality of both the model (e.g., by avoiding inadvertent changes) and the analysis. Consequently, in SAPHIRE 8, the analysis of models is performed by using what are called Workspaces. Currently, there are Workspaces for three types of analyses: (1) Events and Condition Assessment (ECA); (2) Significance Determination Process (SDP); and (3) a General Analysis (GA) workspace. Workspaces are independent of each other and modifications or calculations made within one workspace will not affect another. In addition, each workspace has a user interface and reports tailored for their intended uses.

II.   TECHNICAL CAPABILITIES

The INL began development of a PRA software application on a PC in the mid-1980s. The initial version, Integrated Risk and Reliability Analysis System (IRRAS), was released by the Idaho National Engineering Laboratory (now INL) in February 1987. The development of IRRAS continued over the following years, leading SAPHIRE to become a powerful PRA tool. SAPHIRE 8, building upon earlier versions, provides the functions required for performing a PRA. Users can supply basic event data, create and solve fault trees and event trees, perform uncertainty analyses, and generate reports.

Extending this analysis, Workspaces are user-friendly interface that streamlines and automates select SAPHIRE analysis processes.

SAPHIRE 8 contains improved editors for creating event trees and fault trees, defining accident sequences and basic event failure data, solving system fault trees and accident sequence event trees, quantifying cut sets, performing sensitivity and uncertainty analyses, documenting the results, and generating reports. SAPHIRE capabilities for performing a PRA are summarized below.

- Graphical fault tree construction
- Graphical event tree construction
- Rule-based fault tree linking
- Fast cut set generation
- Fault tree flag sets
- Failure data
- Uncertainty analysis
- Cut set editor, slice, display, and recovery analysis tools
- Cut set path tracing
- Cut set comparison
- Cut set post-processing rules
- Cut set end state partitioning
- End state analysis
- User-defined model types

While the capabilities listed above provide for a full-feature analysis tool, the remainder of this paper will focus on recent technical advancements found in SAPHIRE 8. For additional information on SAPHIRE, we refer the reader to the series of NUREG reports (NUREG/CR-7039) found on the NRC's website ((US Nuclear Regulatory Commission, 2011). The discussion of the technical capabilities will cover the topics of:

- Importance measure uncertainty
- Improved upper bound quantification estimator
- Integrated model solving
- Convolution Adjustments
- Common-cause Failure Adjustment Calculations
- Event Tree Sequence Phases
- Decomposition Event Trees (DETs)

Importance Measure Uncertainty
The importance measures in SAPHIRE Version 8 are largely the same as in SAPHIRE Version 7. However, two improvements to importance measures have been implemented. First, the exact expression for the Fussell-Vesely (FV) is used [$FV = F(i) / F(x)$] as opposed to the older approximation [$FV = 1 - F(0)/F(x)$]. Note that $F(i)$ is the minimal cut set upper bound for the group of cut sets containing the event while $F(x)$ is the original minimal cut set upper bound. The second change is that uncertainty analysis can be performed on importance measures. SAPHIRE Version 8 can perform the importance measure calculation, but can now sample from the basic event uncertainty and feed this into the calculation. The results of the analysis provide importance measure metrics such as mean, standard deviation, and percentiles. In addition to these metrics, a graphical plot is available, for example as shown in Figure 1.
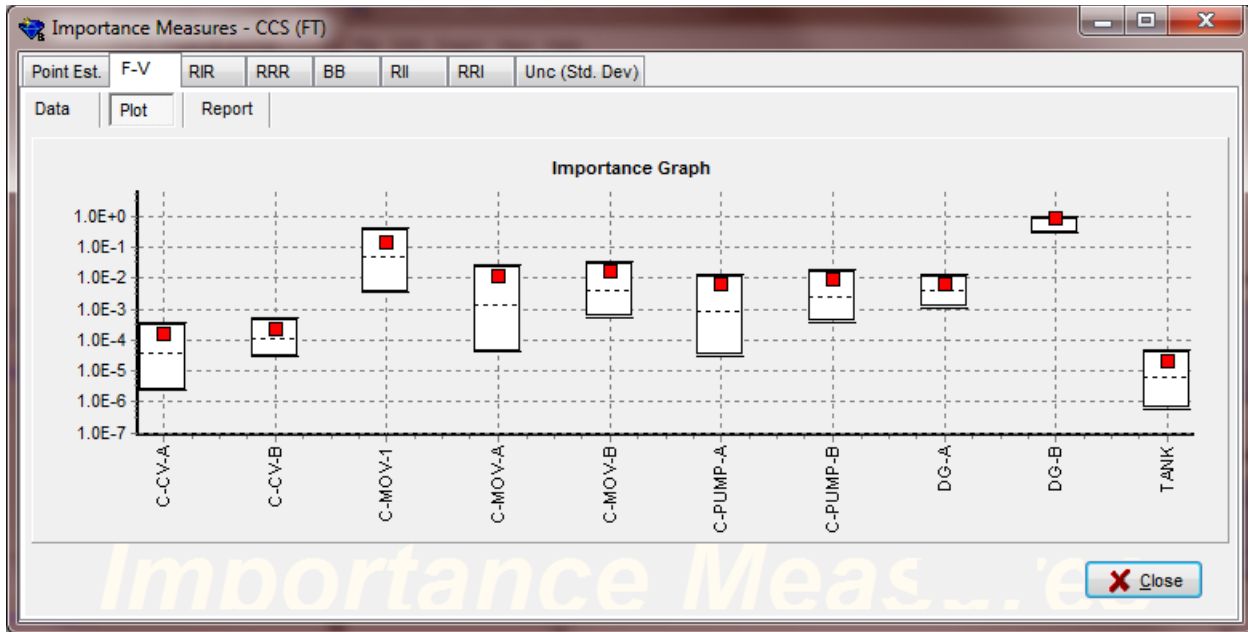
Figure 1. Plotting of importance measure uncertainty

Improved Upper Bound Quantification Estimator

In May of 2007, the INL SAPHIRE development team created an idea for a hybrid approach to quantifying cut sets produced from logic models, using a technique wherein cut sets are used with a binary decision diagram (BDD) approach. (Google, 2007) Later, Dr. Woo Sik Jung proposed a similar approach and developed a tool called ACUBE (Advanced Cut Set Upper Bound Estimator). (Jung, 2015) The ACUBE approach is to reduce the conservatism in top event probability quantification by separating the cut sets into two groups defined as *major* cut sets and *minor* cut sets. Major cut sets are defined as cut sets with higher probabilities (as a group) and minor cut sets have lower probabilities. The major cut sets are converted to a BDD, which then provides an exact answer, P(Smaj), like in the INL approach. The minor cut sets answer, P(Smin), is obtained using the traditional minimal cut set upper bound methodology. The two answers are combined as in the equation: P(S) ≈ 1-(1-P(Smaj))*(1-P(Smin)).

Recently, SAPHIRE has been modified to implement a similar quantification approach, which we call SCUBE (SAPHIRE Cut Set Upper Bound Estimator). The existing SAPHIRE BDD solver is used in addition to the minimal cut set upper bound implementation. To separate the cut set list, SCUBE uses a percent contribution resulting in two groups of cut sets, the major and minor cut sets. When using the SCUBE feature, an analysis report will be produced containing (as shown in Figure 2):

- Name – Name of fault tree, event tree, or end state
- Description - Description of fault tree, event tree, or end state
- BDD Cntrib. % - Contribution value to divide the major cut sets from the minor cut sets
- BDD Cntrib. C S Cnt – The number of cut sets in the major cut set list
- MC Cntrib. – The Minimal Cut Set Upper Bound for the minor cut set list - P(Smin)
- BDD Answer – The exact answer for the major cut set list - P(Smaj)
- SCUBE Answer – The combined answer - 1-(1-P(Smaj))*(1-P(Smin))
- Min Cut U. B. – The original Minimal Cut Set Upper Bound for all cut sets
- Delta (MC - SCUBE) – The difference between the MCUB and SCUBE estimate
- Overestimation % - $(P_{MCUB} - P_{SCUBE}/ P_{SCUBE}) * 100$ [based upon the paper by Dr. Jung]

	A	B	C	D	E	F	G	H	I	J	K
1	End State	SCUBE									
2	Name	Description	BDD	BDD	MC	BDD	SCUBE	Min Cut	Delta	Overesti	Cut Set
3	CD		50%	18	3.18E-6	3.15E-6	6.33E-6	6.35E-6	1.91E-8	0.3%	6204
4			60%	37	2.55E-6	3.75E-6	6.30E-6	6.35E-6	4.73E-8	0.8%	6204
5			70%	75	1.92E-6	4.34E-6	6.26E-6	6.35E-6	9.03E-8	1.4%	6204
6			80%	155	1.29E-6	4.90E-6	6.19E-6	6.35E-6	1.52E-7	2.5%	6204
7			85%	241	9.78E-7	5.19E-6	6.17E-6	6.35E-6	1.77E-7	2.9%	6204
8	CD-EQ	CORE DAMAGE FROM SEISMIC	50%	5	4.66E-5	1.84E-5	6.50E-5	5.03E-5	-1.47E-5	-22.6%	4318
9			60%	7	4.62E-5	1.95E-5	6.56E-5	5.03E-5	-1.54E-5	-23.4%	4318
10			70%	9	4.57E-5	2.00E-5	6.57E-5	5.03E-5	-1.54E-5	-23.5%	4318
11			80%	11	4.40E-5	2.33E-5	6.74E-5	5.03E-5	-1.71E-5	-25.3%	4318
12			85%	12	4.38E-5	2.33E-5	6.71E-5	5.03E-5	-1.68E-5	-25.1%	4318
13	CD-FLI	Core damage due to internal	50%	6	2.02E-10	2.30E-10	4.32E-10	4.33E-10	4.28E-13	0.1%	17
14			60%	8	1.52E-10	2.81E-10	4.32E-10	4.33E-10	4.28E-13	0.1%	17
15			70%	9	1.29E-10	3.03E-10	4.32E-10	4.33E-10	8.75E-13	0.2%	17
16			80%	12	7.42E-11	3.52E-10	4.26E-10	4.33E-10	6.79E-12	1.6%	17
17			85%	13	5.74E-11	3.68E-10	4.26E-10	4.33E-10	7.12E-12	1.7%	17
18	CD-FRI	Core damage due to internal fire	50%	15	1.97E-4	1.99E-4	3.96E-4	3.95E-4	-7.81E-7	-0.2%	18674
19			60%	24	1.61E-4	2.36E-4	3.97E-4	3.95E-4	-1.05E-6	-0.3%	18674
20			70%	42	1.22E-4	2.74E-4	3.95E-4	3.95E-4	6.41E-8	0.0%	18674
21			80%	86	8.31E-5	3.11E-4	3.94E-4	3.95E-4	1.06E-6	0.3%	18674
22			85%	131	6.39E-5	3.30E-4	3.93E-4	3.95E-4	2.11E-6	0.5%	18674

Figure 2. Example SCUBE analysis results for a generic PRA model

Integrated Model Solving

Some PRAs require modeling failures due to causes other than general stochastic failure, such as fires, floods, or seismic events. Historically, model types other than "Random" were created using external events models in SAPHIRE and were run as separate models. Version 8 uses a "Model Type" definition to create models other than "Random." In Version 8, the basic events have been extended to allow the user to specify a basic event's model type(s) as well as different reliability models for each applicable Model Type, and will display this information in one over-arching "virtual" basic event (a "virtual" basic event is one basic event that has multiple failure mode attributes such as fails due to a fire cause, fails due to a flooding cause, fails due to a random cause, etc.). Version 8 allows a combination of model types to be evaluated and integrated at the same time, and presents cut sets which can contain basic events of different model types (e.g., Fire + Random).

Integrated models in SAPHIRE may also be created where the event trees for different initiating events are created, and flag sets are used to specify which components are failed due to the initiating event, or conditions resulting from the initiating event (for example, if a fire in a room disables specific components in that room). Further, the analyses for different parts of the model can be run using the built-in macro system. Further, buttons that run macros for portions of the model, such as "Internal CDF Only," "Internal and External CDF," etc., can be created by the user to enable single-click analysis options (see Figure 3).

Figure 3. Example user-created macro buttons for integrated models

If the Version 8 "Model Type" feature is used to build integrated models, then basic events following different initiators can have different reliability models assigned. For example, a fire event tree which transfers to a loss of offsite power event tree, then system failures in the loss of offsite power event tree could be due to fire-related only, random, or fire-related + random events. The random failure of a component could be given one type of reliability model while the fire-related failure could be assigned a different reliability model.

4

Convolution Adjustments

In SAPHIRE, a convolution event is used determine a more accurate result for a group of "time related" events in a cut set. In this approach, a convolution model is used to calculate the accurate result, and the value for the convolution basic event is the adjustment factor needed to correct the cut set result. To apply the convolution events to a group of cut sets, the user must add a code statement to the Post Processing Rules and solve when applying these rules during cut set generation.

Within the basic events, a new Failure Model has been added to SAPHIRE as a "Convolution (O)" type. This event allows a user to select items that need to be convolved with an optional recovery item. Steps to create a Convolution Event are: (1) create a new Basic Event; (2) select "Convolution (O)" under Failure Model; (3) enter a Mission Time; (4) add desired events by clicking the "Add Event" Button (these are other failure events that result in the need to perform the convolution); and (5) set the remaining options such as failure criteria and associated recovery events. Once the convolution event is created, the "correction factor" for the convolution factor is show in the "probability" field (see Figure 4) – this is the correction factor needed to adjust the matching cut set to make it more accurate.
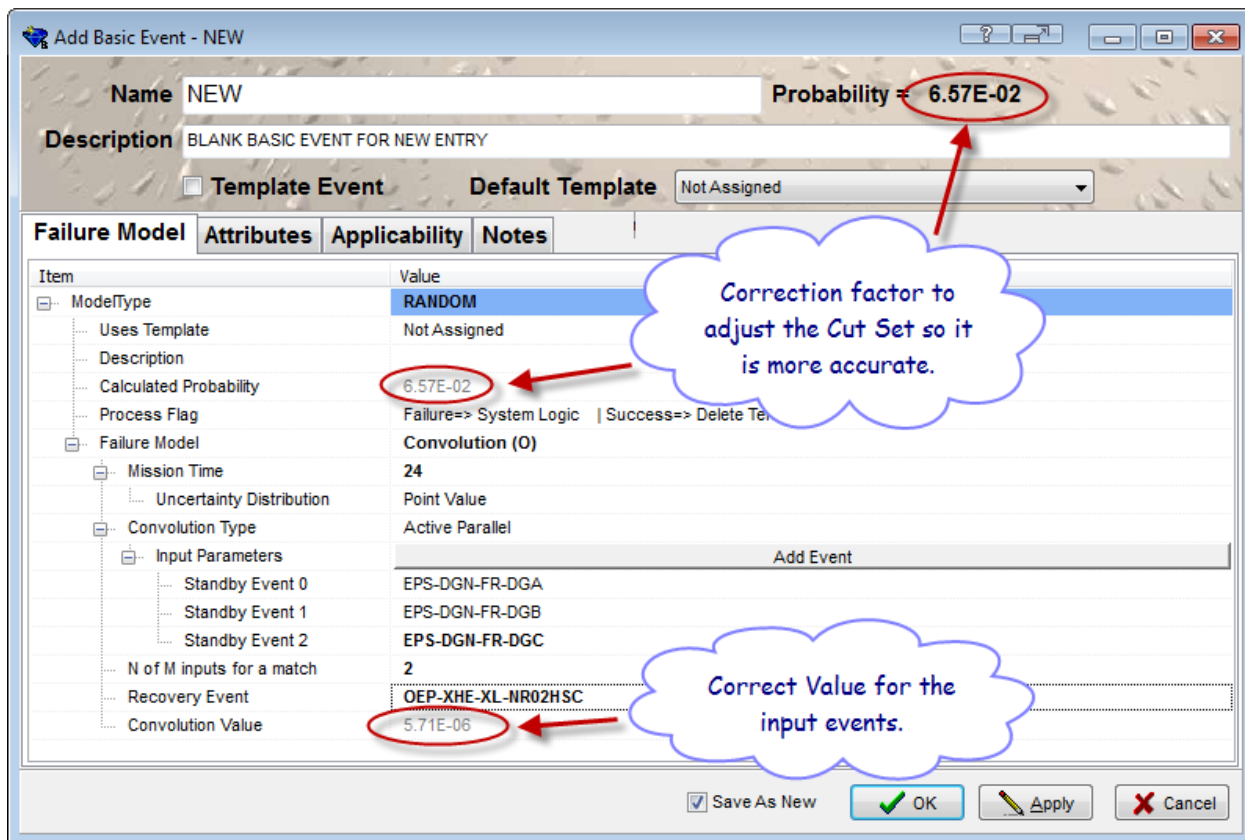
Figure 4. Example of the convolution basic event editing screen

Common-cause Failure Adjustment Calculations

During analyses where a component is failed, we need to consider its associated common-cause failure (CCF) basic event (when present in the model). SAPHIRE has been designed to automatically adjust a basic event's associated CCF event under different conditions. To better understand the conditional CCF calculation, we describe a simple example.

Assume that the probability of a system "S" (with two components, A and B) failing was found to be P(S) = 7.02E-6. We now want to determine the calculation for P(S | Train A "Fails"). When something fails, we condition on the total failure probability Qt (assuming potential for shared cause). The definition of conditional probability is, in this case:

P(S | Train A "Fails") = Pr(S AND At)/Pr(At)

We now need to consider what is the term "S AND At"? This requires Boolean algebra:

$S = Ai * Bi + CCFab$
$At = Ai + CCFab$
where *=AND operator and +=OR operator

S AND At = Ai * Bi * Ai + Ai * Bi * CCFab + CCFab * Ai + CCFab*CCFab
= Ai * Bi + CCFab (after Boolean algebra)

We can now determine the associated probabilities. To do that, we need to return to the alpha-factor model equations and expand the events in terms of alpha factors. For example, the term $Ai=Bi=α1Qt$.

$Pr(S \; AND \; At) = α1*Qt*α1*Qt + α2*Qt$
$Pr(At) = Qt$

Finally,

$P(S | Train \; A \; "Fails") = Pr(S \; AND \; At)/Pr(At) = (α1*Qt*α1*Qt + α2*Qt) /Qt$
$= α1*Qt*α1 + α2$
= Independent Term(s) + CCF Term(s)

Consequently, when SAPHIRE is asked to determine the CCF probability when component A fails, SAPHIRE will automatically provide:

$Pr(CCFab) = α2$

If the CCF term is not adjusted automatically, it would remain at its original value ($α2*Qt$) which would misrepresent the conditional system failure probability.

Event Tree Sequence Phases
    New in SAPHIRE is the ability to depict a "transition point" in an event tree sequence using a "phase" designator. For example, the phase can denote a transition in operation or time at key pinch points in a sequence model. The Phase feature in SAPHIRE 8 can be applied to a Level 2 PRA model in order to subdivide accident sequences and perform either Level 1 or Level 2 analysis by choosing various phases.

    For example, it is possible to assign all Level 1 event trees as pertaining to the core damage (CD) phase. Figure 5 shows an example for the Large LOCA event tree where the core damage sequences of the Level 1 event trees are transferred to the Level 1/Level 2 bridge tree called "L2-CSTET." The L2-CSTET event tree includes two phases: CD and PDS (not shown). The first top event (PHASE_CD_PDS) in the L2-CSTET is used as a "switch" between the CD and PDS phases. This switch allows the core damage sequences transferred from Level 1 event trees go to the up branch to the CD end state or, for the Level 2 results, the core damage sequences go to the down branch and are evaluated with the remaining top events in the L2-CSTET event tree for containment system status. The up branch of the PHASE_CD_PDS is defined as the **CD phase**. The down branch of the PHASE_CD_PDS and the remaining top events of the event tree are defined as the **PDS phase**. For the analyst to evaluate either Level 1 or Level 2 PRA, the option during cut set generation simply needs to be selected for the applicable phase.

Figure 5. Example event tree using the Phase option in SAPHIRE 8

Decomposition Event Trees (DETs)

One other addition to the Level 2 modeling in SAPHIRE was the addition of event tree "sub-trees," also called decomposition event trees (DETs). With this feature, the analysts can use a shortcut modeling approach for representing a number of top events into a separate supporting event tree (sort of an event tree within and event tree). SAPHIRE 8 treats the DET logic as if they are inserted into the "parent" event tree in place of the top event it represents.

III. CONCLUSIONS

SAPHIRE 8 can be used to model a complex system's response to initiating events and quantify associated consequential outcome frequencies (or probabilities) at the component, system, subsystem, sequence, event tree, and end state levels. Specifically, for nuclear power plant applications, SAPHIRE 8 can identify important contributors to core damage (Level 1 PRA) and containment failure during a severe accident which leads to releases (Level 2 PRA). It can be used for a PRA where the reactor is at full power, low power, or at shutdown conditions. Furthermore, it can be used to analyze both internal and external initiating events and has special features for managing models such as flooding and fire.

In SAPHIRE 8, multiple enhancements have been made to the code, both on the user interface and analysis engine in order to support modern risk analysis needs. In this paper, we have described only some of the recent modifications, including: the ability to determine the epistemic uncertainty for all the importance measures; a way to determine a more accurate quantification for groups of cuts sets; the ability to create and efficiently use complex models containing many different hazards such as internal and external events; an approach to adjust cut set values in order to properly account for situations where time-dependence within a cut set calls for a convolution calculation; how SAPHIRE adjusts CCF events automatically in order to treat changing conditions in the baseline model; the use of phase-based pinch points in the event tree logic model; and the idea of event tree "sub-trees" through the use of decomposition event trees. These attributes, and other features, results in SAPHIRE 8 being one of the most powerful cut set based PRA tools in existence.

IV. REFERENCES

Google. (2007). Hybrid assessment tool, and systems and methods of quantifying risk. Retrieved from https://www.google.ie/patents/WO2008005637A3

Jung, W. S. (2015). "A method to improve cutset probability calculation in probabilistic safety assessment of nuclear power plants." *Reliability Engineering and System Safety*, 134, 134-142.

US Nuclear Regulatory Commission. (2011). Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE) Version 8 (NUREG/CR-7039). Retrieved from http://www.nrc.gov/reading-rm/doc-collections/nuregs/contract/cr7039/

DISCLAIMER

This information was prepared as an account of work sponsored by an agency of the U.S. Government. Neither the U.S. Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness, of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof.