Report on the modelling strategy

Workpackage 2

Author(s)	Verificator(s)	Approvers(s)
Name: Jure Brence	Name: Saso Dzeroski	Name: Bastien Poubeau
Date: 20.5.2024	Date: 20.5.2024	Date:
Signature: Jure Brence Digitally signed by Jure Brence DN: cn=Jure Brence, o=Jozef Stefan Institute, ou, email=jure.brence@ijs.si, c=SI Date: 2024.05.20 16:33:02 +02'00'	Signature: Digitally signed by SAŠO DŽEROSKI DN: c=SI, st=Slovenija, o=IJS, 2.5.4.97=VATSI-55560822, givenName=SAŠO, sn=DŽEROSKI, serialNumber=2464246716058, cn=SAŠO DŽEROSKI Date: 2024.05.20 23:53:41 +02'00'	Signature: VALIDATION ÉLECTRONIQUE CERTIFIÉE IRSN Bastien POUBEAU 2024.05.21 11:44:42 +02'00'

Version	Date	Description of modifications
1	17/05/2024	Initial version

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission-Euratom. Neither the European Union nor the granting authority can be held responsible for them.

	Diffusion:
■	ASSAS Partners
■	End-user group and advisory board
■	European commission
■	Public
☐	Other:

	Publication:
■	ASSAS website
■	European commission continuous reporting portal
■	Project Share Point
☐	Other:

SUMMARY

Abbreviations 5

Introduction 6

Specifications of the simulator 7

Severe accident codes 8

Surrogate modelling 11

Formulation of modelling problems 12
Global models 12
Local models 13
Phenomenological division 14

Findings 15
CESAR 15
ICARE and the reactor pressure vessel 15
Vessel rupture 16
Containment 16

Selection of problems 17

Surrogate model validation 17
Root-mean square error (RMS) 18
Fast Fourier Transform Based Method (FFTBM) 18
Wasserstein metric 19

Surrogate modelling problem details 20

CESAR in the RCS 20
The general approach 20
Output variables 21
Input variables 23
Data 24
Data analysis 25
Surrogate models 26
Data generation 28

Solver initialization for CESAR 30
Problem definition 30
Model outputs 32
Model inputs 33
Data generation 35

CESAR and ICARE in the VESSEL 36
Introduction 36
General approach 36
Output variables necessary for other modules 38
Output data required by the simulator interface 39
Input variables from other ASTEC modules 39
Input variables from operator's actions 40
Additional internal state variables 40
Data generation 41
Surrogate Modelling 41

Post-vessel rupture containment **44**
General approach 44
Data 44
Output variables 44
Input variables 47
Operator actions 48
Data generation 49

Global surrogate of MELCOR **50**
Problem definition 50
The general approach 50
Data 51
Input variables 51
Output variables 53
Operator actions 53
Data generation 54

Conclusions ***54***

References ***54***

Abbreviations

- **ASTEC:** Accident Source Term Evaluation Code
- **AFW:** Auxiliary feedwater
- **CET:** core exit temperature
- **CL:** cold leg
- **CSS:** Containment Spray System
- **CVCS:** Chemical and Volume Control System
- **FP:** Fission products
- **FW:** Feedwater
- **HL:** Hot leg
- **HMI:** Human-Machine Interface
- **HPI:** High pressure injection
- **LOCA:** Loss of coolant accident
- **LPI:** Low pressure injection
- **LSTM:** Long-short term memory (neural network)
- **MCR:** Main Control Room
- **ML:** Machine learning
- **NPP:** Nuclear Power Plant
- **PES:** Portable Equipment Systems
- **PSA:** Probabilistic Safety Analysis
- **PWR:** Pressurized Water Reactor
- **RCS:** Reactor coolant system (excluding the primary vessel in the document)
- **RHRS:** Residual Heat Removal System
- **RNN:** Recurrent neural network
- **RPV**: Reactor pressure vessel
- **SAMG:** Severe Accident Management Guideline
- **SBO:** Station Blackout
- **SG:** Steam generator
- **SI:** Safety Injection
- **SIS:** Safety injection system

Introduction

The project ASSAS aims to develop methods for designing and implementing surrogate models to speed up computer simulations of severe accidents in nuclear reactors. In ASSAS, we focus on two widely adopted programs for simulating severe accidents in nuclear reactors: ASTEC and MELCOR, with the majority of attention devoted to ASTEC.

Surrogate models are computationally efficient approximations of computationally expensive calculations and can range from simple mathematical models to complex neural networks. Modern approaches typically make use of deep learning methods to handle the high dimensionality of the problem. Machine learning techniques train a mathematical model by providing many examples of inputs and outputs, referred to as the training data.

Surrogate models are typically used to replace an entire calculation or simulation. However, when speeding up very complicated computer simulations, such as simulations of severe accidents, surrogate models can be developed to replace only specific parts of the simulation software - those that represent a computational bottleneck.

To achieve the goals of the project, the first important step is to study the simulation software and its computational bottlenecks, identify and characterize the most promising surrogate modelling problems, and design a methodology for addressing them. This is a necessary step that informs the next, crucial step in the project – generating the training data. Once the surrogate modelling problems and their requirements are well understood, batch calculations can be launched to generate and store a large amount of training data that covers the needs of all identified modelling problems. In the final step, the methodology and training data enable the development, training and evaluation of surrogate models for severe accident simulations.

In this document, we report on the findings of the first step as part of Work Package 2 – Methodology. We begin by summarizing the specifications of the physical simulator, which inform the simulation scenarios to cover and the required outputs of the simulation software and consequently its surrogate models.

Next, we overview the simulation software ASTEC and its various modules, with a focus on information that is important for developing the surrogate modelling strategy.

We proceed by introducing the concepts of surrogate modelling and categorize different formulations in respect to the simulation software.

After that, we evaluate the many possible options of surrogate modelling problem formulations for ASTEC, highlighting the benefits and challenges of each formulation, and conclude the section by presenting the final selection of modelling problems to be addressed in the project.

The main part of this report presents the details of each of the selected modelling problems. This includes the description and definition of the problem, the identification of the required outputs of the surrogate model, a list of possible inputs to the surrogate model, an analysis of a small data sample, recommendations and requirements for generating data, machine learning method recommendations and preliminary surrogate modelling attempts.

A report on the methodology used for surrogate modelling will be issued at the end of the project, to evaluate the pros and cons of the different explored strategies. Recommendations will be issued for the continuation of the work performed. Especially, the final report will assess the resources that would be necessary to develop surrogate models for industrial simulators, which

have to cover far more scenarios than those considered during the project. This will give insights about the possibility to scale-up the machine-learning approach for severe accident simulation.

Specifications of the simulator

The functional specifications of the basic-principles severe accident simulator developed in the ASSAS project are defined in [1]. The simulator features a simplified four-loop 1,300 MWe Western-type Pressurized Water Reactor (PWR). Therefore, all geometrical characteristics of the reactor, including the core, are fixed once for all. The Human-Machine Interface (HMI) is based on Westinghouse's platform TEAM_SUITE®, which controls the execution of the severe accident code ASTEC (Accident Source Term Evaluation Code). The simulator will be designed to run two possible severe accident scenarios:

- a Large Break Loss-of-Coolant-Accident (LB-LOCA) with Safety Injection (SI) and Containment Spray System (CSS) failure,
- a Station Blackout (SBO) with Auxiliary Feedwater (AFW) failure.

The systems connected to the reactor will be simplified to focus on the most critical equipment to manage the SA. The Residual Heat Removal System (RHRS), the Chemical and Volume Control System (CVCS), and the balance-of-plant will not be considered for the accident management. Portable Equipment Systems (PES) can be deployed to the reactor in less than 24 hours to mitigate the accident. The selection of the plant model and the scenarios are a compromise between competing objectives:

- Causing and displaying most of the physical phenomena involved in a severe accident,
- Concentrating the resources of the project on the riskiest scientific challenges linked to the development of a severe accident simulator,
- Limiting the complexity of the development of data-driven surrogate models linked to the number of degrees of freedom of the reactor model.

Hence, the number of possible actions in the simulator is limited to approximately a dozen, including the operator actions and the malfunctions managed by the training instructor. It is also assumed that all accidents will start with the same initial conditions, namely a reactor operating at its nominal power.

The simulator will have different displays to show the results of calculations. The so-called 2D displays will show synthetic information about the accident progression. It includes all sensors available in the Nuclear Power Plant (NPP) Main Control Room (MCR) and additional visuals to show the progression of the accident, for pedagogical purpose. Only a tiny fraction of ASTEC's internal variables will be displayed, focusing on the most important figures of merit to understand the effect of mitigation actions and the consequences of the accident. So, the 2D dashboards display mostly aggregated data summarising the information contained in several control volumes of the calculation code. They will be complemented by a 3D-display (Virtual Reality module), that will feature the most important physical parameters on a 3D view of the reactor. In this case, the local values of the parameters in each control volume of the calculation code will be necessary.

The priority for machine-learning models is to calculate the data necessary for the 2D displays.

In ASSAS, machine-learning models are used to speed up the execution of the simulator, while keeping a good level of accuracy. Therefore, the targeted acceleration factor of the simulator (the ratio between the simulated time and the computation time) is a central criterion of success. For a realistic training, this ratio must be greater than one. However, the critical phase of severe accidents spans over several days. Usually, severe accident mitigation is considered successful if the containment resists for 5 days. Condensing a 5-day scenario in a single day of work would

be a great advantage, especially for the ex-vessel phase of the accident that evolves relatively slowly. This corresponds to an acceleration factor of 15. Higher acceleration factors open the way to new applications of the simulator that could be a continuation of the project.

Severe accident codes

ASTEC and MELCOR are the severe accident codes developed respectively by IRSN and under the guidance of the American Nuclear Regulatory Commission (US-NRC) and considered in the frame of the project. These system codes model the different physical phenomena involved in a severe accident and calculate the global behaviour of the NPP. They are used to assess Severe Accident Management Guidelines (SAMG), to support level 2 Probabilistic Safety Analysis (PSA) and to evaluate the radioactive source term released to the environment during an accident. They are designed to be relatively fast, to give the possibility to evaluate several hundreds of scenarios with reasonable resources (approximately 4 days for 300 calculations up to the vessel rupture on a 64-core CPU with the simplified input deck used in the project). Their models rely on first principle physical equations and empirical correlations, that are validated against experimental results and high-fidelity calculations.

The ASSAS consortium does not have access to the source code of MELCOR. Therefore, the only possibility to develop surrogate models is to adopt a black-box approach, replacing MELCOR completely for a specific phase of the severe accident.

ASTEC's source code is owned by IRSN, the coordinator of the ASSAS project. Hence, it is possible to envisage different hybrid methods to couple data-driven models with physical models. An overview of ASTEC can be found in [2]. Figure 1 shows the different physical modules of ASTEC. The main modules of interest for the ASSAS project are:

- CESAR is dedicated to thermal-hydraulics in the primary vessel, the primary circuit and the secondary circuit.
- ICARE is dedicated to the degradation of the core.
- SOPHAEROS is dedicated to the deposition, transport and speciation (only in the containment) of the fission products in the primary circuit (without the vessel) and the containment building.
- CPA is dedicated to thermal-hydraulics in the containment.
- MEDICIS is dedicated to the corium-concrete interaction in the reactor cavity during the ex-vessel phase of the severe accident.
- SYSINT is dedicated to the management of safety systems.
- ISODOP calculates the decay of radioactive products.

ASTEC is not able to model the following phenomena:

- Steam explosion resulting from the interaction of the corium with liquid water,
- Mechanical damage of the containment building,
- Neutronics,
- Dispersion of radionuclides into the environment.

Figure 2 shows the computational structure of ASTEC. All modules, except for ICARE and CESAR, are called sequentially at each macro-timestep. They read and write in a central database managed by the ODESSA tool. Each module may need several internal iterations, called micro-timesteps, to calculate the state of concerned physical variables at the end of the macro-timestep. The numerical coupling of CESAR and ICARE is more complex. CESAR is called at each micro-timestep of ICARE to compute the thermal-hydraulic conditions in the reactor. CESAR

usually requires several sub-cycles (CESAR's micro-timestep) to predict variables at the end of the ICARE micro-timestep, which results in 3 levels of intricated timesteps. Each timestep is managed dynamically, to ensure the best performance, stability and reproducibility of calculations (including the limitation of numerical noise propagation). At the end of its micro-timestep, ICARE will correct the prediction calculated at the beginning of its micro-timestep thanks to the updated thermal-hydraulic conditions calculated by CESAR, especially concerning the redox conditions. This numerical scheme has been developed to account for the tight coupling linking CESAR and ICARE in the reactor core, in the first place the positive feedback effect associated with the exothermic and temperature activated oxidation of the fuel cladding.

Figure 1: Physical models of ASTEC

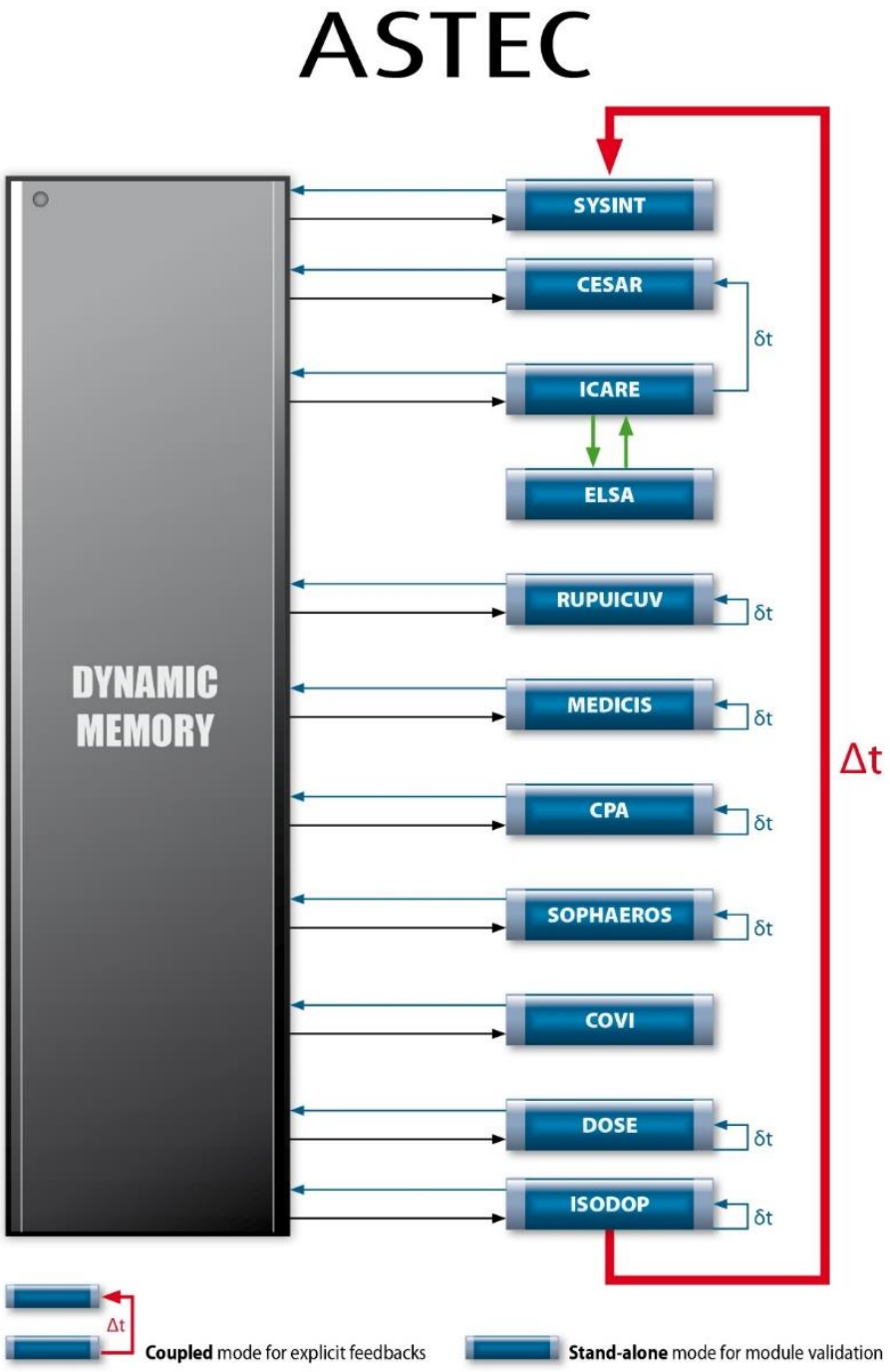


Figure 2: Computational structure of ASTEC

ASTEC has been developed since the 1990s and it is optimised to exploit the single core CPUs available at that time. The sequential structure of the code also contributes to its numerical stability. Therefore, ASTEC is relatively slow to calculate a single severe accident sequence, even on performant multi-core CPUs. This limitation has been overcome in the context of safety analysis by launching numerous sequences simultaneously on a large number of CPUs. However, this strategy is not relevant for a simulator application, which requires a real-time execution of the code for a single sequence.

The characteristics of the reactor model used by ASTEC for the simulator have been agreed upon at the beginning of the project. The discretization of the input deck and some physical models have been simplified to increase the performances of the simulator and limit the volume of training data. In the end, the reactor model is divided in 4 domains: the vessel, the primary circuit, the secondary circuit, and the containment, which includes a zone to model the environment. The description of the vessel is two-dimensional and axisymmetric (5 concentric rings divided in 15 axial elevations, plus a volume for the lower plenum of the vessel). The primary and secondary circuits feature a 1d-description with 158 control volumes altogether. Like the primary and secondary circuits, a zonal description is used for the containment. The reactor building is thus divided in 18 zones, connected by 54 junctions. The containment model is completed by 62 walls.

The simulator will give a deterministic answer to the user, like the instrumentation of a real power plant would give to an operator. In addition, only default modelling options and physical parameters will be used for the simulator, to limit the complexity of the developed surrogate models. Indeed, each additional degree of freedom for the simulation would drastically increase the resources required to train models machine-learning models, a phenomenon known as the "curse of dimensionality". It is usually observed that the amount of necessary training data grows exponentially with the number of independent degrees of freedom of the system to model. Hence, ASSAS will not consider uncertainties coming from physical parameters, which is suitable for a basic-principles simulator. Being able to capture the effects of the different operator actions, automatic actions and malfunctions is already a major scientific challenge. The return from experience of the project will give valuable feedback to evaluate the feasibility to develop more general models with additional inputs.

The most time-consuming model of ASTEC is CESAR, followed by ICARE and SOPHAEROS. The respective contribution of the modules depends on the type of scenario and the phase of the accident. Most of the CPU time consumed by CESAR is dedicated to the resolution of a large system of coupled non-linear partial differential equations. On the contrary, ICARE calls several models sequentially that all account for a small fraction of the computation time. SOPHAEROS calculates the chemo-physical state of each control volume sequentially, following the flow direction between the different control volumes. These various characteristics will have consequences on the surrogate modelling strategies to consider.

Severe accident modelling is subject to large uncertainties caused by different reasons. Positive feedback effects and cliff-edge effects lead to a chaotic evolution of the accident. In addition, many modelling options are available in the code, for example concerning empirical correlations. Validation data is extremely scarce: it is limited to the two severe accidents that have affected water-cooled reactors (Three Mile Island and Fukushima) and to separate effect or semi-integral validation experiments (for example the QUENCH and the PHEBUS PF programmes). Hence, the user effect plays a significant influence on calculation results, which is even greater if different severe accident calculation codes are used. In addition to these uncertainties linked to physical models, ASTEC is also subject to numerical noise propagation, which is increased by the use of a simplified discretization of the reactor model and semi-implicit time scheme. This numerical sensitivity must be considered during the training and evaluation of surrogate models. For instance, the evaluation of the performances of the simulator must be assessed statistically, to ensure that the observed differences are not only due to numerical noise propagation.

Surrogate modelling

Machine Learning (ML) refers to the process by which algorithms learn from data without being explicitly programmed. It involves developing models that can make predictions or decisions

based on input data, often through supervised or unsupervised learning techniques. ML has a wide range of applications in various fields such as natural language processing, computer vision, robotics, and financial forecasting. In the context of our project, we will be using ML to develop surrogate models that can predict the behavior of selected parts of ASTEC under different conditions.

Surrogate Modelling is a type of ML technique used in predicting the behavior of complex systems such as ASTEC under different conditions. The goal of SM is to create an accurate and efficient approximation of the target model's output without having access to its internal workings, which is particularly useful when the original calculation is computationally very expensive.

Surrogate modelling has several advantages, including its ability to handle high-dimensional data and nonlinear relationships between inputs and outputs, making it an ideal choice for predicting complex systems like ASTEC. However, surrogate models also have some limitations, such as the potential for overfitting if the surrogate model is too complex or if there are insufficient training examples. They usually require a large amount of data, which grows exponentially with respect to the number of degrees of freedom of the modelled system.

Formulation of modelling problems

Designing surrogate models for severe accident simulators is challenging, since the simulators are composed of many different modules, each of which is complex on its own. In order to develop precise, efficient and useful surrogate models, it is crucial to carefully formulate the surrogate modelling problem.
During the first year of work on Workpackage 2, a lot of effort has been invested into exploring the many possible formulations of surrogate modelling problems.
In this section, we classify the considered formulations along several axes and discuss the advantages and disadvantages of each category.

Global models

One formulation of surrogate modelling replaces the entire simulation software (i.e. ASTEC or MELCOR) for the whole timeframe of the scenario, or only for a particular phase of the accident. The lone global surrogate model can take as its **input**:

- The type of scenario (LOCA or SBO),
- The **initial conditions** at the beginning of the simulation (for surrogate models that do not start at the time of the initiating event, since initial conditions are fixed for all scenarios)
- The parameters defining **operator and instructor actions**

The **outputs** of a global surrogate model are the time evolution of the variables of interest, displayed by the simulator (such as thermal-hydraulic variables, core degradation variables, etc.).

An advantage of global surrogate models is that they do not rely on input from any other simulator modules. Since they do not require integration into existing software, their deployment is significantly simplified. Furthermore, the predictions of global surrogate models are expected to be faster than the alternatives.

On the other hand, global surrogate models are expected to be significantly more difficult to train. Since they replace the entire simulator, they must learn all the complex dynamics covered

by the simulator and cannot rely on other simulation modules to cover any deficiencies. As such, we expect them to require more training data than the alternative of local models.

Local models

The second promising formulation of surrogate modelling entails surrogate models that replace only a part of the simulator software. To deploy such a surrogate model, it must be integrated into the rest of the simulator software. The local surrogate model must be carefully connected to the other parts of the code to ensure a functional simulator. To achieve a strong speeding up, the model should replace code that is computationally intensive and represents a bottleneck in the computation. There are several options that a local surrogate model can replace:

- Simulator **modules**, which typically specialize in subfields of physics and chemistry (modules such as CESAR, which simulates thermal hydraulics, or SOPHAEROS, which handles chemical and physical reactions).
- Simulator code that governs a particular **physical system** (such as the reactor vessel, the containment, steam generators, etc.).
- A particular, expensive **computation step** during the calculations of a module (such as the solving of a system of differential equations).
- A **combination** of the above (for example, two tightly-coupled modules can be jointly replaced by a single surrogate model in one physical system, leaving the modules intact everywhere else).

A local surrogate model necessarily works in steps, synchronized with the global timestep of the simulator. At every timestep, it generates its outputs for the next timestep based on the information, available at the current timestep. For this reason, this type of surrogate model is often called a **time stepping** (or time stepper) model. The inputs of a local surrogate model can be:

- **Input coupling variables** – variables computed by other parts of the simulation software (for example, a steam generator surrogate model might take as input the thermal-hydraulic variables in the connections between a steam generator and the primary circuit).
- **Operator/instructor actions** can be included explicity as input to a surrogate model.
- **Internal variables** – variables that are not required by the user or other parts of the code, but can help the surrogate model to better capture the state of the system.

Since system and coupling variables are time dependent, at time t_i, the surrogate model can access their values at times $t_0, t_1, \dots, t_i$. The values at time t_0 are initial conditions, defined for the accident scenario, whereas the values at times $t_1, \dots, t_i$ have been predicted by the local model itself or other parts of the simulator at previous timesteps. Typically, a time window is used to select a certain number of past time steps as input. For instance, the values of system and coupling variables from the last 10 steps can be used.

The outputs of a local model are defined by the requirements of other parts of the code and the desired simulation outputs, and can include:

- **Output coupling variables** – variables computed by the replaced code, that are used by the other parts of the simulator as input (for example, the module CESAR computes thermal-hydraulic variables in the vessel, which are required as input by the core degradation module ICARE).
- **Variables, displayed to the user**– variables that are used as information displayed to operators or to summarize the accident.

An important advantage of local surrogate models, compared to their global counterparts, is that they require much less training data. Namely, for a time stepper, every time point represents

an example. In contrast, for a global model, one example can be one entire simulation. Furthermore, since a local model specializes in a physical system or physical subfield, the dimensionality of its inputs and outputs is lower, and the range of possible simulated dynamics is less complicated. This potentially allows the surrogate model to predict the variables more accurately.

On the other hand, compared to global surrogate models, local models are expected to perform slower and offer a lesser speeding up in simulation time. This is in part because they replace only a part of the simulator (albeit a bottleneck part), but also because they must make predictions for every timestep of the simulation. Furthermore, an important downside of time steppers is **error accumulation** – since the input of a time stepper includes its own predictions from a previous timestep, any inaccuracies build up over time.

Phenomenological division

A surrogate model can replace a part of (or the entire) simulator only for some phases of the accident. If the considered phase features dynamics very different from the other phases, this can allow the surrogate model to specialize in those dynamics, reducing the amount of training data required and potentially improving its accuracy.

Such temporal locality is easier to design for local models, provided that they perfectly replace a part of the simulation code and replicate its outputs. This is because all the information, required to engage the original code, is computed by the surrogate model and the rest of the simulator. On the other hand, the practical implementation of switching between surrogate models and direct simulation can be a tough engineering challenge.

In contrast, a global surrogate model does typically not compute all the system variables required to run the direct simulation. As such, once a global surrogate model is engaged, it is not possible to switch back to the direct simulation.

Findings

CESAR

The ASTEC module CESAR governs the simulation of thermal hydraulics in the reactor cooling system and the vessel. It is also one of the most computationally intensive modules of ASTEC. The studies performed during WP2 indicate that:

- The **RCS** is composed of several physical systems (primary circuit, secondary circuit, steam generators), which are well isolated with few coupling dimensions to other physical systems. Furthermore, CESAR is not coupled strongly to other modules in the RCS. Therefore, the simulations of thermal hydraulics in the physical systems of the RCS are a **good candidate** for replacement by **local time-stepping** surrogate models. The preliminary surrogate modelling attempt for steam generators supports this thesis. There are still several possible formulations to consider: a single surrogate model can replace CESAR in the entire RCS, or several surrogate models can specialize in individual physical systems.
- In the **vessel**, CESAR and ICARE are coupled very strongly: in contrast to the rest of the simulator, they exchange information not only every global timestep, but following a smaller, adaptive internal timestep. Replacing CESAR in the vessel would represent a significant challenge due to its tight coupling to ICARE. As such, we conclude that a surrogate model for CESAR in the vessel is out the scope of ASSAS.
- CESAR's calculation process includes the application of Newton's method for solving equations. This step is computationally very demanding. However, **Newton's method** requires an **initial guess** for the solution, which it then refines. In the simplest version, the values at the previous timestep are used. This represents an opportunity for the application of **machine learning** models to improve the initial guess, based on the inputs to the CESAR module. Improving the initial guess can result in a significant reduction in the number of iterations required to solve the equation, speeding up the computation. Since the integration of such a machine learning model into CESAR is feasible, we conclude that this is a promising task to tackle within ASSAS.

ICARE and the reactor pressure vessel

ICARE is an ASTEC module that simulates core degradation and is composed of several sub-modules. It is a computationally very demanding part of the simulator. We find that:

- Due to the reasons elaborated in the CESAR section, it is very difficult to decouple CESAR and ICARE in the vessel. As such, developing a **surrogate model for ICARE is not feasible.**
- Since the calculations governing the dynamics in the vessel are a crucial and computationally intensive part of the simulation, a surrogate model for the vessel is still required. Fortunately, the vessel is a well-isolated physical system – the main coupling point to other systems is at the inflow and outflow of the RCS and the fission products inventories to be managed by SOPHAEROS. As such, a surrogate model that **jointly replaces CESAR and ICARE in the vessel** is the most **promising solution**. Such a model can ignore the intricacies of the internal coupling between CESAR and ICARE, and focus only on modeling the couplings to external modules and on predicting the crucial core degradation quantities displayed to the simulator user.
- ICARE has several submodules, such as ELSA and MOVEMAG. The studies performed during WP2 indicate that **separate surrogate models for ICARE submodules are not promising candidates** for development in ASSAS. Firstly, such surrogate models are not easy to develop due to the intricacies of the internal timestep and the CESAR coupling.

Secondly, such surrogate models are not likely to yield significant speed-up, since the sub-modules are much less computationally intensive than CESAR and only contribute to a small fraction of the ICARE calculation time. As such, submodule outputs should be included in the joint surrogate model for CESAR and ICARE, or directly simulated (as is done currently), whichever is simpler to implement.

Vessel rupture

The phenomenon of vessel rupture marks a significant change in requirements and priorities for the simulation of a severe accident. The phase before vessel rupture can last hours and the detailed simulation of dynamics in the vessel and RCS is crucial, with the timesteps of the simulation kept short. Although certain modules, such as SOPHAEROS, are computationally demanding, CESAR and ICARE are by far the most important computational bottlenecks during this period.

In the simplified input deck, we assume that the corium relocates entirely out of the vessel. After that point, there are no more relevant physics in the vessel to simulate, and the dynamics in the RCS become less interesting – the focus after vessel rupture is on the containment. At this time, CESAR and ICARE can be turned off, which dramatically speeds up the computation. However, in the post-rupture phase, the time-scale changes from hours to days of simulation. Thus, although the simulation is faster, it must also simulate much longer time periods, and speed-ups are still needed. In this phase, the main computational bottleneck is SOPHAEROS.

Due to these reasons, a promising approach is to address the development of surrogate models **separately for the phases of the accident before and after vessel rupture**. Before vessel rupture, the surrogate modelling formulations discussed above are to be considered, while the post-rupture phase requires a focus on the containment.

Containment

The dynamics in the containment are primarily simulated by the thermal-hydraulic module CPA, the chemistry – by the module SOPHAEROS, and the core-concrete interaction – by the module MEDICIS. Before vessel rupture, the modules dealing with the containment are not computationally demanding compared to CESAR and ICARE. However, in the post-rupture phase, they are the only relevant parts of the code running. As such, developing surrogate models for the containment in the post-rupture phase is of great interest.

In the post-rupture phase, **both local time-stepping** surrogate models for the slowest modules, and a **global model** for the entire ASTEC code (excluding CESAR and ICARE, since they are turned off after vessel rupture) are **viable options.**

In the post-rupture phase, the desired speed-up factor is higher than in the early accident phase. Since the simulation of days of dynamics are required, this currently takes a long time to simulate using direct simulation. As such, a **global model is the preferred solution**, since it offers a higher potential for speed-up by predicting the future dynamics in one shot, instead of timestep by timestep.

Selection of problems

As evident from the previous section, there are many possible formulations of surrogate modelling problems for speeding up a severe accident simulator. The investigation and development of even a single surrogate model from among the many options is a challenging and time-consuming task. It is therefore necessary to carefully select a small number of surrogate modelling problems that offer both the highest chance of success, as well as the highest potential impact on the computational efficiency of the simulation software. Having reviewed the possibilities and their advantages and disadvantages, we focus on the following five surrogate modelling problems.

Partners	**To replace**	**Physical system**	**Model type**
PSI, Phimeca	CESAR	RCS	Local time stepping
IRSN, BelV, ENEA	CESAR: Newton's method initialization	RCS	Local time stepping
KIT, CIEMAT	CESAR + ICARE	VESSEL	Local time stepping
JSI, Energorisk	CPA, SOPHAEROS, MEDICIS	Containment	Global model
KTH	MELCOR	All	Global model

The details of each selected surrogate modelling problem are provided in the next chapter.

Surrogate model validation

An essential issue is the validation of developed surrogate models developed during the project, that is, to be able to decide whether the surrogate models replicate well enough the models in the system code (ASTEC or MELCOR). Specifically, this means comparing the time-dependent variables, calculated with the system code and with the surrogate model, and deciding whether they are sufficiently similar.

Basically, two aspects of the simulation results, representing the time-dependent quantities, will have to be compared:

- The shapes of the curves.
- In some cases, whether some of the characteristic events (for instance, the rupture of the reactor pressure vessel, to name the most crucial event in the accident progression) have been simulated with the surrogate model as well. This validation will be done implicitly, based on the comparison of suitable figures of merit.

Usually, when comparing experimental and theoretical results, curves are compared visually. However, in the present case, there are two drawbacks:

- The huge number of comparisons that will have to be performed necessitates an automated computational procedure (this is a quantitative constraint).
- Related to that, the difficulty of keeping the same criteria over the huge number of comparisons (this is a qualitative constraint).

Thus, it is necessary to establish an automated and formal procedure for the validation of surrogate models.

Several methods will be used to evaluate the performance of the surrogate models to take into account the large uncertainties in the field of severe accident described above. As usual in the field of machine-learning, performances will be evaluated on a test dataset that has not been

used during the training of the algorithm. If surrogate models are included in the simulator as planned, additional tests will be carried out following the classical acceptance procedures of simulators.

Validation will be carried out in two steps:

1. Model validation: each model will be individually validated according to established criteria.
2. Global validation: the overall response of the simulator will be validated.

However, the range of discrepancy allowed in quantitative comparison for each variable will become clearer when results obtained with surrogate models start to be compared with results from the system code. Thus, at present, no precise quantitative criteria are defined yet.

The use of the following methods is foreseen:

- Root-mean square error (RMS)
- Fast Fourier Transform Based Method (FFTBM)
- Wasserstein metric

Root-mean square error (RMS)

The machine-learning community usually uses the root-mean square error, also known as the L_2 norm, to train and evaluate models. Non-regression tests performed along with the development of ASTEC also rely on this simple metric. However, this method has the following shortcomings:

- It gives too much importance to high-frequency oscillations of variables, which have less physical meaning.
- Some essential difference between simulations by the system code and the surrogate model, which reflects an important physical difference, might go unnoticed because the differences in other parts of the curves might be small.

For this reason, the use of more advanced methods is foreseen with the purpose of focusing on the most important characteristics of the response (that is, the time-dependent behavior of physical quantities as a response to initial and boundary conditions). Nevertheless, the RMS method will still be used, as it provides a kind of basic reference value of the difference between calculated (in two different ways) time-dependent quantities.

Fast Fourier Transform Based Method (FFTBM)

The FFTBM was developed in the nuclear community with the purpose of quantifying differences between experimental and theoretical results [5]. Its use has not been noticed outside nuclear and even in that field, it has not been universally accepted yet. Nevertheless, as in the past two decades it has been used by a number of research organizations (in the nuclear field), it seems to be suitable for the kind of time-dependent quantities that appear in simulations of nuclear accident scenarios. Although the FFTBM was mostly used for validation of design-basis accident simulations, it has been used recently in the validation of results of simulations of a severe accident with the ASTEC code [6]. Thus, the FFTBM method is foreseen as the main method to be used for validation of surrogate models.

The basic idea of the FFTBM is to compare two time-dependent quantities through their Fast Fourier Transforms (so that discrete values are obtained). From both FFT, an average (over the

considered frequencies) amplitude difference AA and a characteristic weighted frequency WF are calculated:

$$\mathrm{AA} = \frac{\sum_{n=0}^{2^m} |\tilde{\Delta}F(f_n)|}{\sum_{n=0}^{2^m} |\tilde{F}_{\mathrm{exp}}(f_n)|}$$

$$\mathrm{WF} = \frac{\sum_{n=0}^{2^m} |\tilde{\Delta}F(f_n)| f_n}{\sum_{n=0}^{2^m} |\tilde{\Delta}F(f_n)|}$$

where summations are performed over the considered frequencies, *ΔF* denotes the difference between the amplitudes at given frequencies, and the index *exp* refers to the reference function. The agreement between quantities is considered good if AA is low at high WF.

Wasserstein metric

The Wasserstein metric is based on transport theory [7]. That is, the distance between two curves is quantified as if the curves were piles of sand, and the minimum amount of transport of sand necessary to transform one curve into another is calculated. Although the Wasserstein metric is supposed to be used for comparison of probability density functions, it has been accepted (quite) universally for comparison of functions with complex shapes.

The Wasserstein distance between two (probability density) functions *P* and *Q* is calculated as:

$$W_p(P,Q) = \left(\inf_{J \in \mathcal{J}(P,Q)} \int ||x-y||^p dJ(x,y) \right)^{1/p}$$

where the minimum is calculated over all joint distributions *J* (*x,y)* that have marginals *P* and *Q* and *p* is a characteristic parameter.

Although it is not foreseen to be the main method to be used for validation of surrogate models, the Wasserstein metric might be used occasionally if it appears that the FFTBM method has some serious shortcomings in some specific cases.

Surrogate modelling problem details

CESAR in the RCS

We are developing surrogate models for the thermal-hydraulic part of the reactor cooling system (RCS) during the transient state, i.e., from the initiating event until RPV failure. CESAR and ICARE will be deactivated after RPV failure. For a given initiating event (SBO, LOCA, …) the surrogate models should be able to reproduce the thermal-hydraulics of the RCS, as well as the effects of operator's actions on the evolution of the transient, that may be applied during the simulation. The relevant ASTEC modules in this stage are CESAR (which addresses the vessel, the primary and secondary circuits), ICARE, SOPHAEROS and ISODOP.

Our task addresses CESAR out of the RPV. Therefore, the model is not directly coupled with ICARE, even if correlations probably exist with variables calculated by ICARE. The fission product inventory determined by SOPHAEROS doesn't directly impact CESAR, but their power release determined by ISODOP adds extra heat to CESAR, since part of alpha, beta and gamma rays is absorbed by the walls.
The principal requirements for this surrogate model are:

- the ability to predict the time evolution of most relevant quantities, listed later, depending also on the interactions with operators during the simulation.
- simulation should be at least as fast as real time.

It is worth noting that the RCS can experience different flow conditions: fluids can enter (exit) the vessel from (to) the hot and cold legs, depending on the phase of the accident. For example, when the large break of the cold leg opens, water will flow from the vessel to the cold leg. When a large debit of cold water is injected into the RCS with portable equipment, water may flow from the hot leg to the vessel.

The general approach

The operators should be able to interact with the calculation at any time during the transient, to observe the impact of their decisions. Therefore, our main focus will be on developing a time-stepping model that makes predictions only one timestep at a time, displaying outputs with respect to simulator specifications.
The main issue is to decide which parts of the module to replace:

- One surrogate model covering the whole RCS. One surrogate model for each system or parts of the RCS, e.g. steam generator (SG), hot and cold legs, secondary circuit.
- A mix between surrogate models and physical model

A first attempt of surrogate modelling was made to:

- Test our ability to replace CESAR in the four SGs by a neural network (following the work of IRSN's intern Louis Proffit).
- Identify relevant input and output variables.
- Define new challenges and ways to improve the surrogate model.

This first approach encompasses the whole primary circuit (see picture below). Only variables at vessel outlets and inlets are tracked. The vessel itself is not encompassed but features such as the total decay heat must be considered as input. Other aspects such as the inventory loss through the breaks and valves, heat loss in the steam generator, and of course mass and enthalpic flows from/into the vessel must be considered also as input and output.

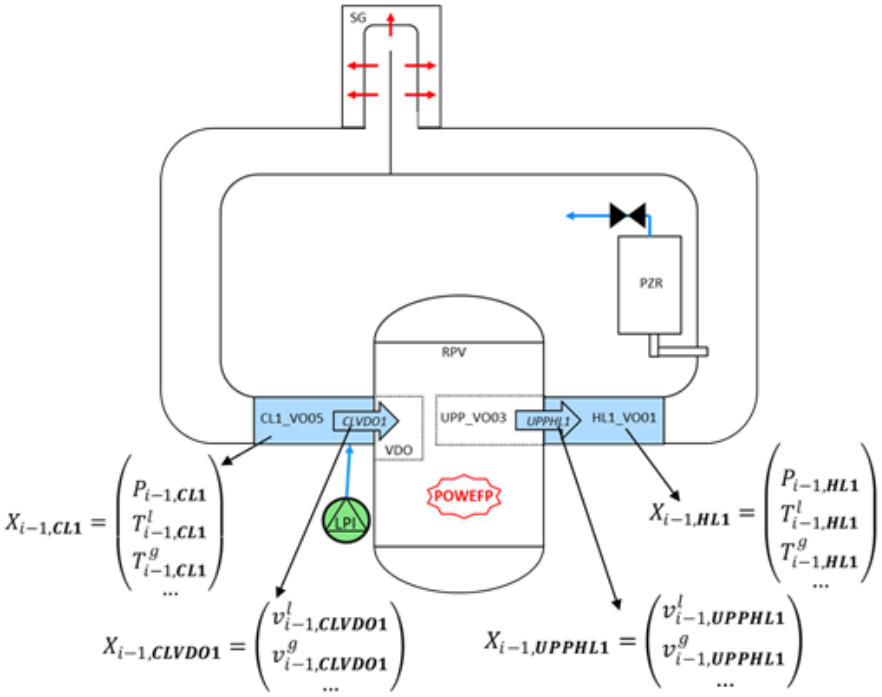


Figure 3: variables in the primary circuit model

However, we cannot exclude more global approaches (e.g. using decision-tree or clustering methods, attention mechanism etc.), which could probably help us to identify parameters or events that are the most decisive for the outcome predictions. To some extent, they could support the time-stepping approach, by dividing the model in several surrogate models depending on the values of some key-parameters or criteria .
The next step will consist in investigating the modelling limitations of the first approach, as this involves a surrogate model encompassing the whole RCS as one part.
An alternative would be to develop a meta-model for each system, similarly to L. Proffit's work [4].

Output variables

As we aim at simulating whole sequences until RPV failure, outputs will essentially be state variables at macro-time step t+Δt, given these same state variables at the current macro-time step t as inputs.

Regarding the current approach, which focusses on vessel inlets and outlets, the model should predict state variables in hot legs and cold legs; more precisely, in the volumes directly connected to vessel channels. Furthermore, parameters quantifying loss/gain of mass and heat in the RCS should also be accounted. Additionally, we provisionally also consider parameters from the vessel giving some more context. Eventually, such information will be provided by the metamodel dedicated to CESAR-ICARE inside the vessel. The table below (Table 1) lists the parameters that are considered as inputs and outputs in the frame of our preliminary approach. The final list of variables, taking into account the specifications of the simulator and the characteristics of the final input deck, are detailed in

	System state variables considered as inputs (value at t) and outputs (value at t+Δt), to be calculated in each control volume of the systems replaced by the surrogate model	**Inputs from other modules / portions of the circuits**
Primary variables	- Gas and liquid temperature - Void fraction - Partial pressures of steam and hydrogen - Gas and liquid velocities - Wall temperature - Reactor cooling pumps velocity - Pressurizer level - % of opening and mass flow rates of valves (SGs & Pressurizer safety & relief valves) - Accumulators mass flow rate - Accumulators pressure - Accumulators level - RCS hydrogen concentration / mass - Flow conditions at breaches	- Power generated by the deposited FPs on the Walls (computed by SOPHAEROS) - Containment counterpressure - Thermal-hydraulic state variables in the control volumes adjacent to the systems replaced by the surrogate model - Water injected by safety systems (SG feedwater, PES)
Suggestion of additional features calculated from primary variables to be considered	- RCS mass flow rates - RCS and Pressurizer pressure - RCS average temperatures - RCS subcooling - SG pressures - SG steam mass flow rates - SG water level	- Total decay heat - % of degraded fuel - Core exit temperature - Water level in RPV

Table 2. In this preliminary analysis, we chose figures of merit different from the ones required by the simulator, because the discussions on the specificities of the user's expectations were still in progress at that time.

Variables in the current approach	Variables providing more context at RCS boundaries
Gas and liquid temperature Void fraction Pressure Partial pressures Gas and liquid velocities **SIS mass flow rates (input only)** Heat exchange in SG Opening section of PORVs <u>Additional features calculated from the above-mentioned primary variables:</u> H2, N2, CO2, CO, BHO2 mass flows Liquid mass flow Steam mass flow Mass flow through PORVs	Core exit temperature TRIC Water level in RPV **Total decay heat (input only)**

Mass flow through the breaks Total mass of water in primary Total mass of steam in primary Total mass of water in secondary Total mass of steam in secondary Total mass of hydrogen	

Table 1. parameters used in the preliminary ML approach for RCS

It must be mentioned here that N_2, CO_2, CO, BHO_2 are removed in the latest version of the PWR input deck. Only steam and H2 will be considered in the future. Some other variables will probably also not be included, such as the total mass of water in the primary for instance; though they could be of use for ML. Fluid velocities in RCS will be used as inputs for SOPHAEROS[1], which simulates FP transport in the primary circuits. As FP transport will have to be computed in the whole primary circuit, it will be necessary to compute state variables for all primary circuit volumes, and not only HL and CL.
All variables listed below won't be necessarily predicted directly by the metamodel (i.e, as metamodel's outputs), but rather explicitly calculated from variables predicted by the metamodel using SENSORs.

	System state variables considered as inputs (value at t) and outputs (value at t+Δt), to be calculated in each control volume of the systems replaced by the surrogate model	**Inputs from other modules / portions of the circuits**
Primary variables	- Gas and liquid temperature - Void fraction - Partial pressures of steam and hydrogen - Gas and liquid velocities - Wall temperature - Reactor cooling pumps velocity - Pressurizer level - % of opening and mass flow rates of valves (SGs & Pressurizer safety & relief valves) - Accumulators mass flow rate - Accumulators pressure - Accumulators level - RCS hydrogen concentration / mass - Flow conditions at breaches	- Power generated by the deposited FPs on the Walls (computed by SOPHAEROS) - Containment counterpressure - Thermal-hydraulic state variables in the control volumes adjacent to the systems replaced by the surrogate model - Water injected by safety systems (SG feedwater, PES)
Suggestion of additional features calculated from primary	- RCS mass flow rates - RCS and Pressurizer pressure - RCS average temperatures - RCS subcooling - SG pressures	- Total decay heat - % of degraded fuel - Core exit temperature - Water level in RPV

[1] The RCSMESH function is activated for the considered ASTEC input deck, which enables the use of a coarser discretization for SOPHAEROS than for CESAR.

variables to be considered	- SG steam mass flow rates - SG water level	

Table 2. variables to be considered for the metamodel to be interfaced with the simulator

From primary CESAR state variables variables calculated by the surrogate model, ASTEC will evaluate secondary variables like:

- Enthalpy or volume mass of the fluids,
- Undamaging of circuits possibly leading to an induced breach,
- The reactor cooling pump torque.

It has been estimated that these calculation steps are already fast enough (reading in the thermodynamic database or evaluating a correlation), whereas it would be difficult for the surrogate model to learn the complete thermodynamic properties of interest. The same is true for the break mass flow rates (in case of Break), which is computed explicitly from a correlation: it seems therefore preferable to exclude it from the scope of the surrogate model.

Input variables

As mentioned above, most inputs are simply the same variables as outputs but at the previous timestep. The flow conditions at the boundaries of the domain and the containment counterpressure must be considered as additional inputs. Moreover, decay heat from fission products deposited on the walls must be considered as pure heat source that is only dependent on time, and thus, doesn't need to be calculated. Operator's action, instructor loaded malfunctions and induced circuit breaches must be also accounted as an input. In the scenario of SBO considered in this preliminary test[2], only LPI is re-activated. This can be represented in our surrogate model by a binary value (ON/OFF) or by the LPI mass flow. The latter makes maybe more sense, as the mass flow is dependent on the system pressure. But this won't cover the case where the operator activates the injection while the pressure is still too high; in other words, when the status of SIS is ON, and mass flow still near 0 kg/s. Therefore, probably both (status and mass flow) should be accounted as inputs for the calculation, while only SIS status might be "playable" by the operator.

Data

Data were generated by simulating typical severe accident sequences using the PWR1300MW-simplified input deck provided by IRSN within the ASTEC v3.1 package.

a. Base case scenario

We defined as base case scenario a Station Blackout (SBO). This scenario includes the remote (forced) opening of SEBIM valves when the mean fluid temperature at the top of the core is above 800°C. However, no other active system is operational; that is, except passive systems such as safety valves in the primary and secondary circuits or discharge of the hydro-accumulators, every kind of water injection is lost: no FW and AFW (turbine-driven pump is assumed to be lost) in the secondary side, no SI etc.

b. Sampling of operator's action

[2] The preliminary test is disconnected from the scenarios considered in the simulator. It is only intended to test the methodology.

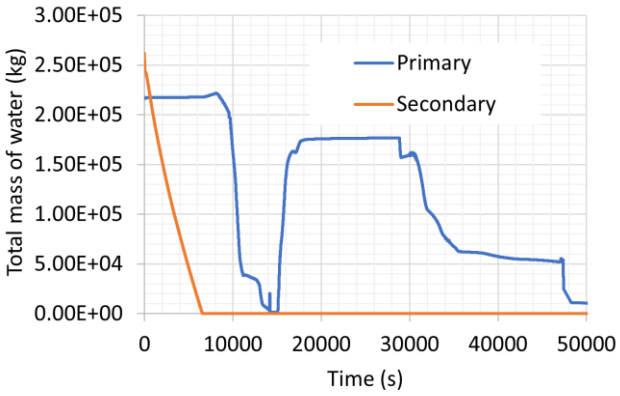
Taking this base case as reference, we varied these parameters:

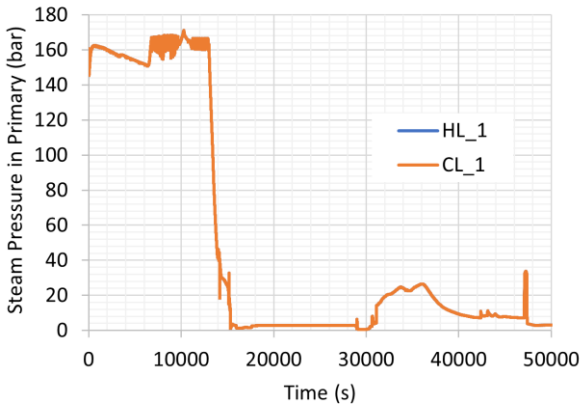
- the number of LPIs to be activated (1 or 2) in the cold legs,
- the recovery time of LPIs.

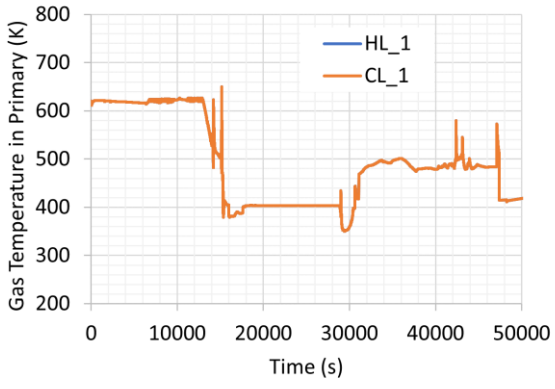
Defining any LPI before the remote (forced) opening of SEBIM valves is useless, because the pressure is then too high. So, after running the base case and checking when the pressure drop occurs (subsequent to SEBIM remote opening), as well as when the RPV fails, respectively around 13000 and 24000 s, we launched batch calculations by sampling different recovery times within this time range, as well as the number of LPI pumps. The following sample matrix was launched:

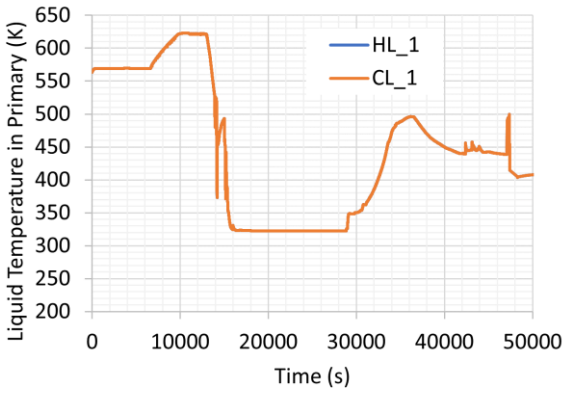
Number of LPIs (-)	1, 2
LPI recovery time (s)	1000000 (no injection during this simulation: base case scenario) 13000, 14000, 15000, 16000, 17000, 18000, 19000, 20000, 21000, 22000, 23000

These 23 simulations (base-case + 22 runs with LPI) run until RPV failure successfully. The database was saved at each time step, providing a convenient dataset for RNN. In order to reduce the size of output files, the filter provided by IRSN was used, retaining only essential variables. The evolution of some important variables is depicted below, such as the inventory of water in the primary and secondary circuits, pressure in hot and cold legs, as well as fluid temperatures, and of course water injection rate through LPI injection (in this example, LPI got recovered after 15000 s) and steam loss through SEBIM valves.









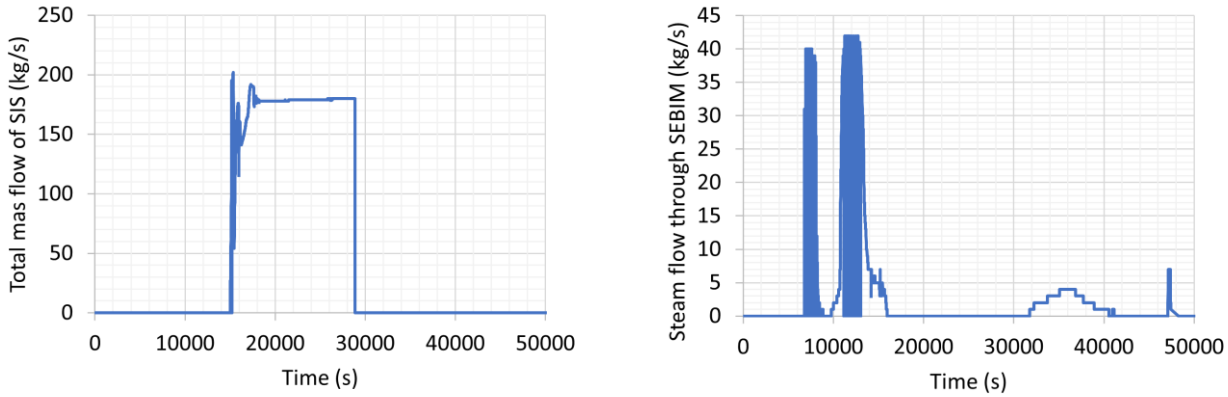


Figure 3. Sample of SBO with recovery of 2 LPIs in cold legs after 15000 s; RPV failure occurs at 47382 s.

Data analysis

The data analyzed in this study are inherently high-dimensional, comprising 23 simulations, each consisting of 2000 to 10000 timesteps and encompassing 153 physical variables.
Figure 4 As a preliminary step, we aimed to identify redundant variables that were fully correlated with others. For this, we employed Pearson correlation coefficients, calculated for each pair of variables. This analysis allowed us to identify pairs of variables exhibiting perfect correlations, with coefficients of 1 or -1. Variables with a correlation of 1 are perfectly positively correlated, indicating a direct linear relationship, while those with a correlation coefficient of -1 are perfectly negatively correlated, reflecting an inverse linear relationship. Upon identifying these relationships, we removed one variable from each perfectly correlated pair from our dataset (Figure 4).

This makes it possible to reduce the dataset to 89 variables.
This approach provides a straightforward method of initial data simplification; however, in the subsequent phases of our study, we are likely to employ more sophisticated dimensionality reduction techniques that are better suited to the specificities of our problem.

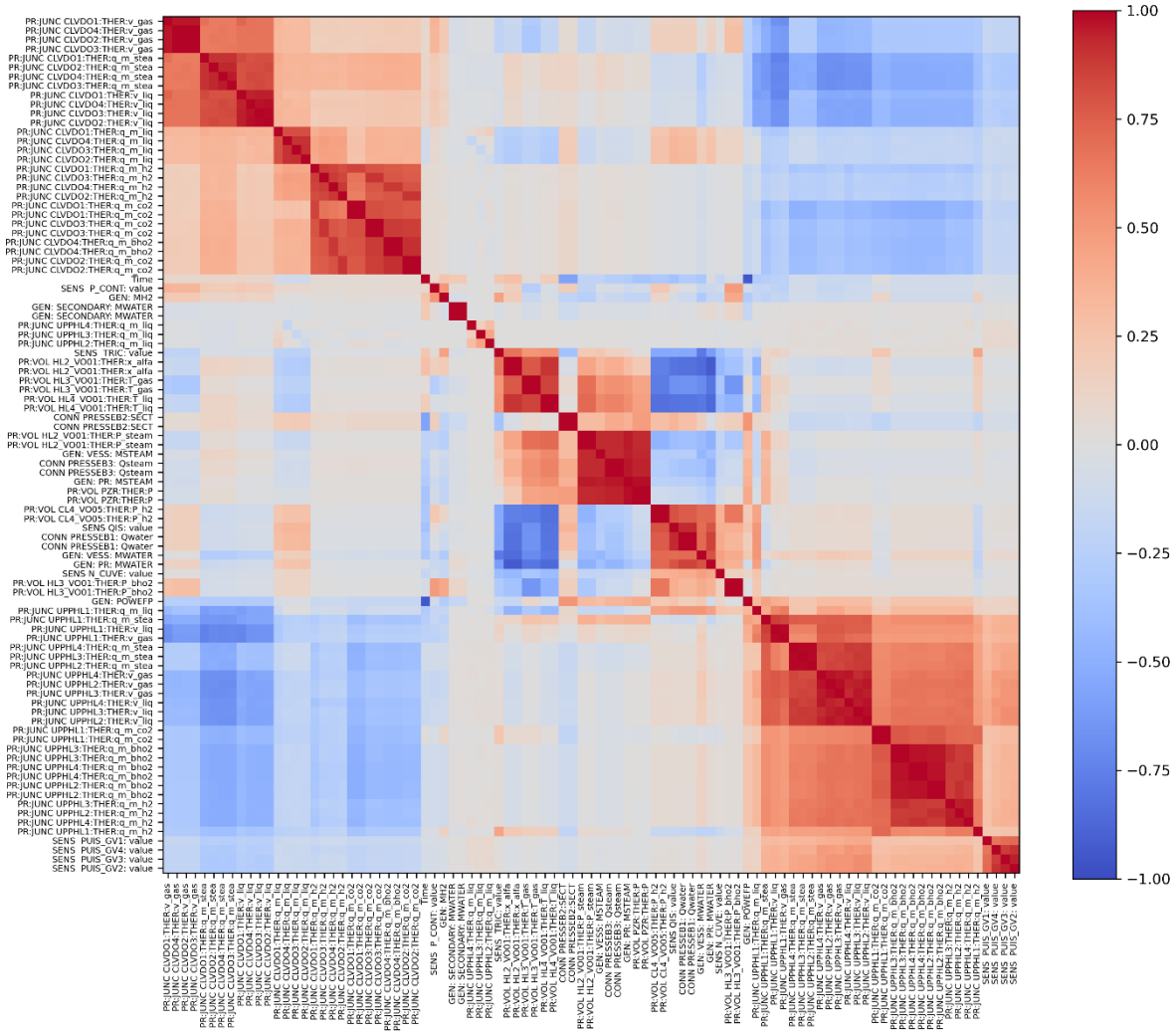
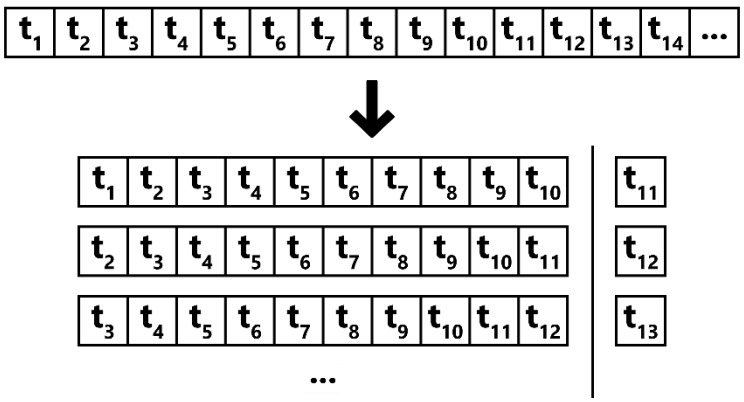


Figure 4. Correlation matrix of the 89 physical variables in the data set after the filtering with the Pearson correlation.

For the rest of the study, these 89 quantities were normalized so that they only varied between 0 and 1, in order to facilitate the work of the surrogate models by avoiding them having to manage several different orders of magnitude for the different variables.

Surrogate models

The aim of our surrogate model is to predict the data in this dataset from the first time steps of the data. To this end, we have divided all our simulations into small sequences of 10 time steps, and we will try to train a model to predict the 11th timestep of this same sequence. As the initial data had different time steps during the simulation, we interpolated the data to maintain a uniform timestep throughout. We obtained 216326 sequences of 10 time steps this way.



This set of short sequences is divided in two: 80% of them are used for learning the model (randomly chosen), while the remaining 20% will be used as test samples to evaluate the model thus constructed.
Inspired by the work of Louis Proffit, our first attempt at building a surrogate model involves the use of recurrent neural networks (RNN), and one type of RNN in particular: the Long-Short Term Memory networks (LSTMs). This type of neural network is widely used to predict time series.
To start with, we've built our meta-model with 3 layers (2 LSTMs layers at 100 units each and a dense layer) to test the method. To improve the results later on, we will consider adding more layers to increase the depth of the network and/or increasing the number of units in existing layers.

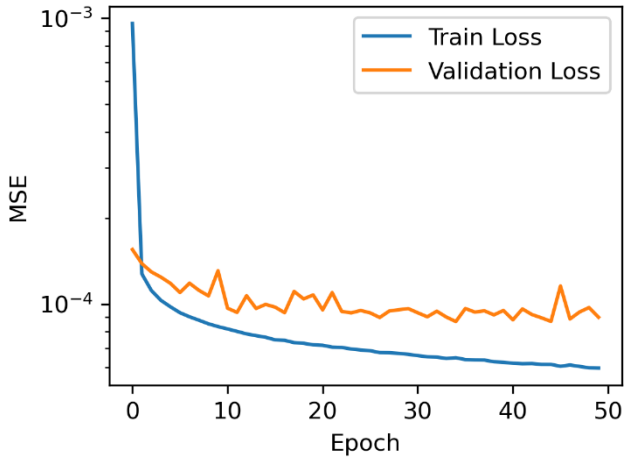


Figure 5. Training and Validation Loss Curves for the LSTM Network used

To check whether the neural network is working properly and not overfitting, we need to look at the value of the loss function for the training data (train loss) and the value of the cost function for the cross-validation data (validation loss).
Figure 3 shows that train loss decreases rapidly at the start of training, then continues to decrease more gradually through epoch (the different iterations of our method). This suggests that the model is learning well from the training data. Validation loss also decreases, but more slowly than train loss. This is a good sign, indicating that the model is generalizing and learning from relevant patterns and not simply memorizing the training data.

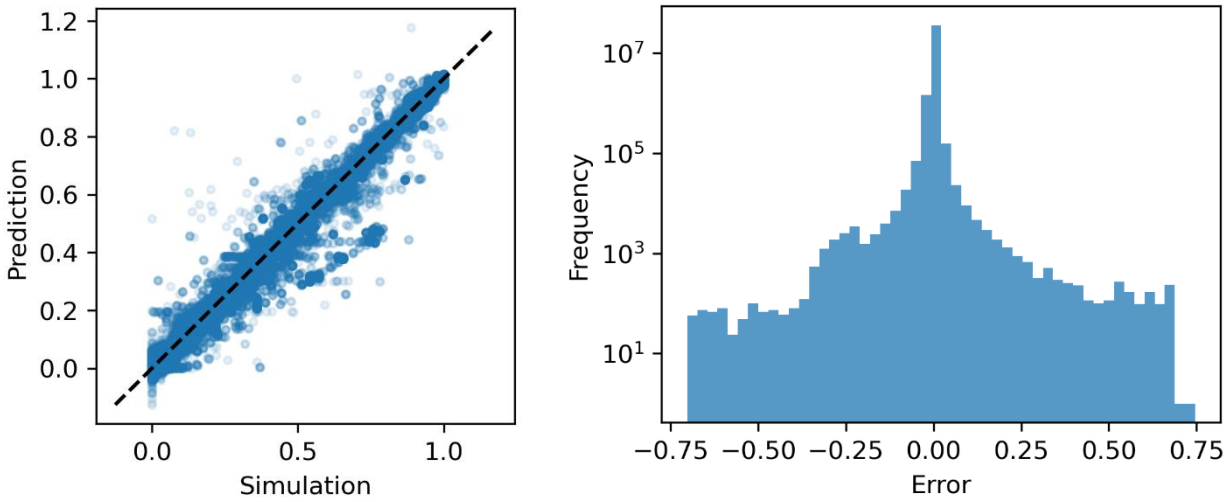


Figure 6. Compare predicted normalized values with simulations and plot the histograms of the errors.

We can look at the predictions of the 11th time step made by the constructed meta-model and observe how far they deviate from the values given by the simulations in figure 4. Overall, the points on this graph are gathered around the $y = x$ line. The error is unbiased but also with a large variance. The mean squared error is arround 8.2×10^{-5}, which seems far too large to be able to predict a complete time series with an error that accumulates over timesteps.

To enhance these results, an initial step could involve data cleansing. The presence of outliers in the valves, which cause peaks in the data, significantly disrupts the learning process. Special attention should be paid to investigate these peculiar phases of the accident. Additionally, oscillations in variables related to velocity prove challenging to predict. Implementing a data smoothing technique could be an effective method to improve the overall quality and predictability of the dataset.
A second step would be to increase the number of layers in our LSTM, and/or the number of units in the LSTM layers. This may require more simulations to work. We could also consider adding a Feature Selection beforehand, taking more physical knowledge into account. This work is still in progress.

Data generation

Timesteps

As it was decided to work on developing a time-stepping surrogate model, it is more reasonable to save results at each macro-time steps during the simulation. Working with a fixed timestep of 1.0 s would be also more convenient for methods such as RNN, but the time step affects the accuracy of the calculations, and besides, we must keep in mind that each ASTEC module runs with its own micro-timestep; that is why, it is preferable to keep the timestep adaptative. The dataset obtained by calculating with an adaptative timestep can be turned by interpolation into a new dataset with a fixed timestep later on.
On the other hand, saving the database at a certain frequency only will result in much lighter data, and thus, allows us to run more samples (a finer variation of the LPI recovery time for example). This may be interesting to better capture the general impact of operator's measure on the sequence evolution.However, such results shouldn't be directly used to develop time-steppers, relying on a fixed (and comparatively smaller) timestep, as some fast transients may be lost. Another interesting idea would be to save short series of timesteps at a given frequency; like 20 timesteps in a row every 500 timesteps for instance. It could help to increase the diversity of data. But, here as well, we should be careful not to miss some relevant transients.

Filter

The size of the results repository can easily exceed 1 TB for around 20 runs if no saving filter is applied. It is then necessary to apply the filter provided by IRSN. . By focusing on CESAR, and excluding most variables addressing CPA, as well as some variables from VESSEL, 20 runs may yield around some 100 GB. However, we will eventually have to save also CPA and VESSEL variables, as the dataset will be also used by the partners, who deal with CPA, and CESAR-ICARE in the vessel. Exchanging, uploading and downloading such heavy data is still an open issue.

Extraction of the database

Extracting the whole database at each timestep is extremely time-consuming (it can take days). At this stage it is more convenient to identify the most interesting variables and to extract these variables only. PSI provided Python-scripts that extract some selected variables at each timestep.
Moreover, and especially for our SBO cases where variations can be applied only after a certain time (that is, after the pressure drop due to SEBIM forced opening, then allowing for LPI), all runs will be strictly identical in the early phase of the sequence, until reaching the time at which we start sampling operator's actions. Therefore, data extraction can start after this point, to prevent the generation of the same datapoints over and over, bringing no valuable information for machine learning. However, it would be more efficient to generate samples by restarting calculations at each operator's action time directly from the base case calculation, instead of simulating every sample from the beginning.

Solver initialization for CESAR

Problem definition

CESAR, the thermal-hydraulic module of ASTEC, is the first contributor to the calculation time of ASTEC. It solves an implicit non-linear system of equations thanks to a Newton-Raphson algorithm (also known as gradient descent), which accounts for most of its CPU time. In the native version of ASTEC, a single system of equations is solved at each internal timestep of CESAR, which comprises all the main circuits of the reactor, so all the volumes in the DOMAINs PRIMARY, SECONDARY and meshes in the VESSEL. At each timestep, the solver is initialised with the converged solution of the previous timestep. The purpose of the task described here is to improve the solver initialisation thanks to machine-learning models like in [3].

Short description of the Newton Raphson algorithm:
Let $\underline{X}$ be the vector of CESAR unknowns, and F the function representing the set of equations to solve. A solution of the system of equations $\underline{X}_{sol}$ satisfies:

$$F(\underline{X}_{sol}) = \underline{0}$$

Practically, the numerical solver will create a series of approximate solutions $\underline{X}_k, k \in \mathbb{N}$, satisfying $\underline{X}_0 = \underline{X}_{init}$ and $\lim_{k \to \infty} \underline{X}_k = \underline{X}_{sol}$. In the standard version of CESAR, $\underline{X}_{init}$ is the converged solution of the previous timestep. The solutions $\underline{X}_k$ are calculated iteratively thanks to a gradient descent:

$$\underline{X}_{k+1} = \underline{X}_k - J^{-1}(\underline{X}_k) \cdot F(\underline{X}_k)$$

With J the Jacobian matrix of the function F.

Practically, the algorithm is stopped at the kth iteration when the following condition is respected for each jth component of the vector of unknows:

$$|x_{j,k+1} - x_{j,k}| < \epsilon_j$$

With ϵ_j the convergence criteria for each primary variable, which can be expressed in Pa, m/s, K….

This strategy has several advantages. The most important is that it guarantees rigorously the same calculation accuracy as the reference models because the convergence criteria of the solver will not be modified. So, errors will not accumulate. ASTEC's functionalities will also remain untouched. In addition, integration will be relatively easy, because the function used to initialise the solver is well-identified, with a precise list of output variables. It can call an external function on demand. ASTEC will automatically calculate all secondary variables (enthalpies, volume masses…) from the primary variables provided by the initialiser.

Newton-Raphson algorithms converge quadratically when they are close to the solution. However, the number of iterations increase drastically if the algorithm is initialised far from it. Therefore, CESAR adopts an adaptative timestep management, leading to the increase of its internal timestep if the solution is found in less than a predefined number of iterations. On the contrary, the timestep is decreased if the number of iterations exceeds a certain threshold. Therefore, the internal timestep of CESAR can vary significantly during a calculation, typically between 10 ms and 10 s, with most points between 0.1 s and 1 s. The reduction of CESAR's timestep is difficult to predict. It is more common during the most energetic phases of the accident (usually, the core degradation) and sudden and large changes in boundary conditions (actuation of safety systems, like the discharge of accumulators when the primary pressure reaches their design pressure), but there is no general rule. These are the most challenging phases of the accident for the simulator to run in real time.

A better initialisation of the solver will reduce the number of gradient-descent operations, so the number of linear systems to assemble and solve. If successful, the strategy will lead to the increase of the internal timestep of CESAR, with even more positive effects on performances. Reference[1] shows that the main advantage of this method for the considered case was to avoid the dramatic increase of solver iterations when the system evolved rapidly. The benefits were less important for more linear phases of the calculation. Hence, the number of gradient descent iterations necessary to converge to the solution of the system is smoothed during the calculation. Such properties would fit well with the specifications of a simulator since they would prevent the simulator to "freeze" during the most demanding phases of the calculation. Reducing the calculation time when the local acceleration factor is lower than 1 is sufficient for a simulator, even if the impact is limited in other phases. On the contrary, such an approach cannot reach as high acceleration factors as global surrogate models. Hence, it is less interesting when very high global acceleration factors are looked after, for example for uncertainty propagation (which is not the purpose of ASTEC). It also requires access to the source code, which explains why it is less popular in the literature.

Reducing the timestep of CESAR may have a positive impact on ICARE as well. In fact, ASTEC uses three hierarchical levels of timesteps. First, ASTEC has its own macro-timestep, at which all modules exchange data (except for ICARE/CESAR). Second, during a macro-timestep, ICARE typical needs several internal timesteps to achieve its convergence criteria. Third, CESAR timesteps must be a fraction of ICARE's internal cycles, to ensure the proper coupling of the two modules, which occurs at the end of each ICARE timestep. However, the stability of the numerical scheme of ASTEC requires that the ratio of the ICARE timesteps to the CESAR timesteps does not exceed a given threshold. If the limit is reached, the ICARE timestep will automatically decrease. Therefore, the increase of CESAR's timestep can also lead to the increase of ICARE's timestep.

As a benchmark, the machine-learning models should be compared to more classical solver initialisation techniques, like polynomial extrapolations of previous timesteps, using successive derivation orders. Such methods are expected to behave poorly when the phenomenology or the boundary conditions change suddenly because they just extrapolate from the past. Machine-learning methods should enable to anticipate the future better, while taking advantage of the past. A hybrid approach may also be fruitful, for example by using a classifier that will decide about the initialisation strategy (linear extrapolation of previous timesteps, machine-learning algorithm, initialisation with the converged solution of the previous timestep). The classifier could also recommend reducing the timestep right away, because it anticipates that the maximal number of iterations will probably be reached, resulting in a restart of CESAR with a shorter

timestep[3]. Besides more classical training methods, reinforcement learning may be used for two reasons. First, data is relatively easy to generate. Second, the reward function is easy to compute: it can correspond to the number of iterations necessary to converge to the solution, or to the residue of the system of equations after initialisation.

First attempts to generate surrogate models should focus on simplified problems. To start, the initialiser could be trained for a specific system and a specific event. For example, it could predict the thermal-hydraulic conditions in the pipe where a breach has just occurred, while all other variables are initialised with the converged solution of the previous timestep. If this step is successful, new systems and news types of events can be added to the model. The integration of the initialiser into CESAR will be straightforward, whatever the scope of the considered variables is.

If the strategy proves to be efficient, it will be possible to decline it (when relevant) to other modules and models in ASTEC using a Newton-Raphson solver, provided the source code is adapted to save the additional necessary data.

Model outputs

The outputs of the model correspond to the **primary variables** of CESAR (BASE:CESAR_IO:OUTPUTS:STRU VARPRIM in the ODESSA database, provided that the necessary option has been activated) in all zones[4], walls, and junctions of the circuit, as they have been calculated by the solver:

- partial pressure of each gas (steam and hydrogen actually),
- void fraction,
- liquid temperature,
- gas temperature,
- liquid velocity,
- gas velocity,
- external and internal temperature of walls,
- pump velocity.

For swollen water level volumes, an additional variable is added: the void fraction of the liquid phase. The list of non-condensable gases is fixed at the beginning of the calculation. The outputs of one timestep are a fraction of the inputs of the next timestep (so the details about the output variable can be found in the next section dedicated to inputs).

The current version of the input deck has 234 zones, 309 junctions, and 217 walls. It results in a total of 2233 output variables. These variables are spatially correlated. They probably do not all have the same importance for the initialisation of the solver, so the machine-learning models may concentrate on the most significant ones (like the flow conditions in the zones close to a circuit breach). ZONES can vary in shape, volume, and type of junction with other ZONES. Therefore, it might prove difficult to treat them like an image. A representation as a graph may be more adapted.

[3] This situation happens 30% of the time with the current implementation of CESAR, leading to many failed attempts to find a solution. The database will include information about the failure or success of the convergence, that can be used to train the classifier.
[4] A ZONE is usually called a MESH in the VESSEL and a VOLUME in the circuits.

Model inputs

The objective of the method is to predict the new values of all CESAR state variables as precisely as possible, based on all data available at the beginning of the timestep. The ASTEC developers' team has launched a systematic identification of all variables having an influence on CESAR's solution in the current calculation scheme. So, it used a deterministic approach, without considering statistical correlations (all variables having an impact on the results have been kept, even if this influence is negligible). The variables of interest are the following:

- CESAR internal timestep,
- state variables,
- data from systems connected to the circuit (SYSINT),
- data from other DOMAINs obtained through CONNECTIs.
-

Variable name	Type of variable	Dimensionality
CESAR timestep	Scalar	1
Pressure, temperatures (liquid / solid), gas partial pressures, void fraction	1-D scalar field in DOMAINs PRIMARY and SECONDARY 2-D axisymmetric scalar fields in DOMAIN VESSEL	Number of ZONEs (234) x number of variables
Fluid velocities (liquid / solid)	1-D scalar field in DOMAINs PRIMARY and SECONDARY 2-D axisymmetric scalar fields in DOMAIN VESSEL	Number of JUNCTIONs (309) x 2
Wall temperatures (internal / external)	1-D scalar field in DOMAINs PRIMARY and SECONDARY 2-D axisymmetric scalar fields in DOMAIN VESSEL	Number of WALLs (217) x 2
Volume of the mesh available for fluid circulation, blockage, hydraulic diameter and porosity	2-D axisymmetric scalar fields in DOMAIN VESSEL	Number of meshes * 4
Residual power in the circuit (FP POWER calculated by SOPHAEROS and ISODOP)	1-D scalar field in DOMAINs PRIMARY and SECONDARY	Number of WALLs (217)
Heat flux between the fluid and components, and its derivative with respect to the fluid properties (temperature, partial pressures, void fraction…)	2-D axisymmetric scalar fields in DOMAIN VESSEL	Number of components[5] in the core exchanging energy with the fluid (fixed list generated at the start of calculation with a STRU CONV): 66* 6
Mass fluxes for steam consumption / hydrogen	2-D axisymmetric scalar fields in DOMAIN VESSEL	Number of components in the core reacting chemically with the fluid (subset of the

[5] COMPONENTS are managed directly by ICARE.

creation their derivatives with respect to temperature		components exchanging energy with the fluid) * 4
BREACHes:	Boolean	3 localisations (primary breach for the LOCA scenario, and induced primary breach or steam-generator tube rupture for all scenarios)
Containment counterpressure (for primary breach)	Scalar	1
Primary pumps velocity	Scalar	4
Relief and safety valves	Scalar	2 variables (section and counterpressure) 3 groups of systems (intact steam generators, steam generator with broken tube and pressurizer) →Dimension is 6.
Safety injection by accumulators	Scalar	1 variable (debit) 4 systems
Injection in the RCS with portable equipment	Scalar The pumps are not regulated (on/off only): the debit from the single injection point will be divided between the 8 entry points (4 hot and cold legs) of the safety injection system.	1 variable (debit)
Steam generator feedwater with portable equipment	Scalar	1 variable (debit) 2 injection lines feeding 4 systems

It must be noted that many degrees of freedom in ASTEC will be fixed for the project, for example the temperature of the water injected by portable equipment. Many malfunctions are also assumed.

A machine-learning approach may identify additional variables of interest used by other modules, like dimensionless quantities (Nusselt, Reynolds numbers…), which provide information about the fluid regime. Usually, considering more than one timestep in the past can improve the performance of the surrogate models.On the contrary, some of the identified variables may be neglected because of their weak correlation with results. A statistical analysis or dimension reduction techniques may be used to identify the most important variables.

CESAR only considers data from the current timestep to calculate the new converged solution. However, machine-learning algorithms may use data from previous CESAR iterations. Practically, the ODESSA database will only contain data from the current macro-timestep. So, at the beginning of the macro-timestep, no data about previous CESAR iterations will be available in

the main memory. The same will be true for the training database. Specific provisions must be considered by the machine-learning algorithm to save historical data explicitly or implicitly (for example in the internal memory of a recursive neural network).

Data generation

For the needs of ASSAS, an option has been introduced in ASTEC to save the necessary additional data. For each saving of the ODESSA database, the input and output variables identified above will be saved. Since the ODESSA database is saved only at each macro-timestep, several couples of vectors (input, outputs) will be saved, corresponding to the number of CESAR timesteps during the macro-timestep. Therefore, each macro-timestep will consist of many training datapoints. When the solver does not converge before the maximal number of iterations is reached, the input data is still saved with a tag informing about the non-convergence.

Several cases can lead to numerical issues. No guarantee exists concerning the existence and unicity of the solution of the non-linear system of equations. The absence of solution, a local change in concavity, or the existence of multiple solutions can prevent the solver convergence. In this case, the solver will restart with a shorter timestep. If the iterative reduction of the timestep does not allow the convergence of the solver, the calculation crashes. The existence of several solutions may also lead to numerical oscillations. These can easily be distinguished from physical oscillations, which have a longer periodicity and are more regular (usually a sinusoid, whereas numerical oscillations have a triangular shape).

Saving data for the internal iterations of CESAR does not impact the size of the ODESSA database significantly. To increase data diversity while limiting disk usage, it may be interesting to save data only for interesting snapshots (in the sense of phases of the scenario that strongly influence subsequent developments). Each snapshot should be constituted of two successive macro-timestep (or more) to have enough historical data to initialise the algorithm.

$$Exhaustive\ data\ saving\ approach: (t_i)_{1 \leq i \leq N}$$
$$Saving\ limited\ to\ some\ snapshots, with\ triplets\ of\ timesteps\ for\ example: (t_i, t_{i+1}, t_{i+2})_{i \in S}$$

With t_i the ith macro-timestep of the calculation, N the total number of macro-timesteps, $S \subset [\![1; N]\!]$ the list of snapshots. The criteria to define S can be static (every nth iteration), dynamic (snapshots with slower convergence are saved more frequently), or adaptative (snapshots with a bad performance of the machine-learning model are saved).

The adaptative timestep management in CESAR requires specific provisions. If machine-learning algorithms allow a better initialisation of CESAR's solver, the module will automatically increase its timestep. At the next iteration, the initialiser will be called with a longer timestep (so a longer time horizon for its prediction) than used in similar conditions by the native CESAR solver. Therefore, specific provisions must be observed to train the initialiser with timesteps that will be representative of the situations in which it will be called, when interfaced with ASTEC. Two strategies can be proposed:

- Train the initialiser to predict the converged solution at $t + n\Delta t_{CESAR}$ from the data available at t (with Δt_{CESAR} the internal timestep of CESAR and n an integer).
- Train the initialiser on a small data sample; generate new data with a longer CESAR timestep allowed by the use of the initialiser in ASTEC; train the initialiser on the new data; repeat the process iteratively.
- Generate training data with specific rules for the adaptative timestep management, leading to a larger number of iterations at each timestep.

Such strategies can be deployed if offline tests are satisfactory.

Remark: it should be tested if learning how to predict the difference between the true solution and the solution at the previous time step, normally used by ASTEC to initialize Newton algorithm, is more efficient that predicting the true solution directly.

CESAR and ICARE in the VESSEL

Introduction

The surrogate model will estimate the behaviour of the thermal-hydraulic, core degradation, and fission product release in the **VESSEL domain of ASTEC**.
The ASTEC modules involved in the developed models are CESAR and ICARE.

The models must cover the evolution of the vessel from the nominal operating conditions at the initiating event time up to the failure of the lower head. Therefore, models should handle different states inside the vessel, covering among others:

- Intact core with inadequate cooling.
- Partially degraded fuel rods.
- Largely degraded core with the formation of a molten pool and/or debris.
- High fraction of steam and hydrogen in the thermal-hydraulic channel.
- Fission products heat and release.
- Molten material movements.
- Changes in core channel geometry.

Several questions are still open about the scope of the surrogated model:

- The lower plenum can be considered within the domain of interest of the surrogate model or left to be estimated by the native ASTEC model (this second option may complexify the interfacing of the surrogate model substantially).
- The degradation of the vessel wall and the failure of the lower head can be computed by the ICARE devoted model or by the surrogate model.

General approach

Two approaches were identified at the beginning of the project for the development of surrogate models for the core degradation:

- focusing on specific physical models, remaining close to the separation between the different modules of ASTEC;
- focusing on a specific region of the modelled domain (namely, the primary vessel) and including the complete phenomenology.

Of these two main alternatives, only the second one is considered for the vessel to successfully achieve the aimed results for core degradation and fission product release. The main reasons for this selection are:

- The tight coupling between CESAR and ICARE requires the interaction of both modules at every CESAR Newton Raphson internal iteration.
- The high computational cost of ICARE, while no specific model in ICARE accounts for a significant contribution of this cost.
- To limit the dimensionality of the boundary conditions and to avoid considering all the possible sequences of operator actions.

Concerning this last point, limiting the surrogate model to the vessel (as defined in an ASTEC input deck[6]) has several advantages. First, the vessel is connected to the primary circuit only thanks to two junctions (CONNECTI in ASTEC):

- downcomer to lower plenum,
- core to upper plenum.

In addition, no safety system is directly connected to the vessel. The different automatic or manual actions (safety injection, depressurisation of the primary circuit, cooling of the steam generators) will act on the vessel indirectly by changing the flow conditions at its inlet or outlet. So, the surrogate model will not have to consider those different actions individually as inputs, as illustrated by Figure 7.

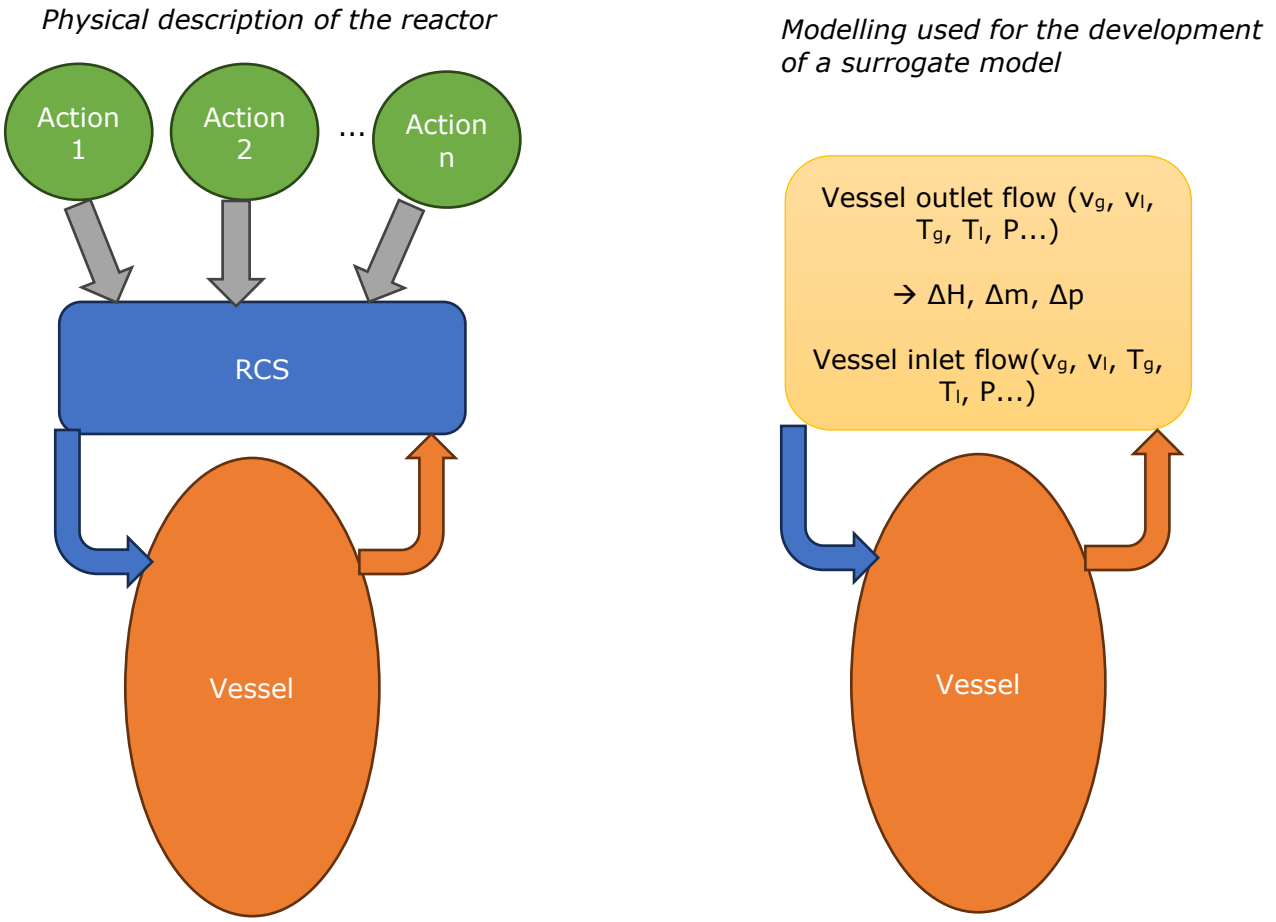


Figure 7: Each action on a safety system connected to the RCS modifies the properties of the flow coming out the vessel and re-entering it, by modifying its momentum (p), mass (m) and enthalpy (H).

The surrogate model for the vessel will have to communicate with other active modules at each macro time-step, which is adapted automatically by ASTEC within user specified boundaries.

[6] In an ASTEC input deck, the domain VESSEL includes the active core, the upper and lower core structures, the core barrel, the downcomer and the lower plenum of the vessel.

The first attempt of the model will consider explicit coupling with the rest of the ASTEC modules, possibly with a linear data interpolation between the beginning and the end of the macro time-step, to provide more realistic conditions for CESAR in the RCS at each of its internal iterations. Semi-implicit coupling can be considered later in the project if the development of the explicit model is achieved successfully. The semi-explicit coupling would require the computation of the derivatives of the model solutions with respect to the boundary conditions. Besides, a modification of the ASTEC's solver would be necessary, which is not planned at this stage.

As explained for the CESAR surrogate model, counter current flows may be encountered in the primary circuit.

Output variables necessary for other modules

The easiest way to couple a global model within ASTEC is by using already existing CONNECTIs. Therefore, in order to transfer output variables to other modules, the vessel model should make use of the CONNECTI from VESSEL to PRIMARY through the following variables:

- Liquid temperature.
- Gas temperature.
- Liquid mass flow rate.
- Gas mass flow rates (steam and hydrogen).
- Mass of fission products:
    - If possible, all fission products considered by ICARE,
    - At least the elements displayed by the simulator interface, while checking the impact on other figures of merit, like the dose rate in the containment. Such a simplification may lead to numerous side effects affecting other modules like SOPHAEROS and ISODOP, which must be evaluated carefully. Therefore, this option will only be considered later in the project.
- External temperature of the vessel wall at two locations (active core and lower plenum) to account for thermal fluxes to the walls of the cavity.

As for the CESAR metamodel, ASTEC will read the necessary thermodynamic properties of the fluid from a database to complete the list of outputs.These magnitudes will be evaluated at each ASTEC's macro timestep (i.e., the timestep at which a module exchanges data to interact with the other modules). The thermal-hydraulic variables will be transferred to CESAR, whereas the fission product masses will be transferred to SOPHAEROS. Limiting the number of considered elements for the fission products must be assessed carefully, because it is a major simplification. In addition, it will certainly prevent from using the SM for the ex-vessel phase of the accident, which will have been trained on the full list of elements.

The surrogate model will also have to calculate the time of the vessel rupture, which depends on the degradation of the wall of the vessel.

Besides, at the failure of the vessel, the CONNECTI from VESSEL to CONTAINMENT should be used to provide the following output variables:

- Composition (masses of the elements) of the corium.
- Temperature of corium.

These output variables will be transferred to CPA and MEDICIS modules.

The isotopic composition of the corium is calculated by ISODOP from the mass of the elements. It is averaged over the vessel and, after the vessel rupture, the vessel and the cavity together.

To avoid huge sensitivity in the simulation in the ex-vessel phase of the accident, the input deck used in ASSAS assumes that all the mass in the core is poured instantaneously into the cavity at the time of the vessel failure.

The surrogate model should be able to correctly predict the timing of the relevant events, such as oxidation runaway for hydrogen generation, the release of volatile fission products, the movement of molten materials, etc.

Output data required by the simulator interface

Of the ICARE/CESAR variables, only a selection will be displayed in the simulator interface concerning the evolution of the core degradation:

- Water level in the vessel
- Reactor subcooling (ΔT_{sat})
- Maximum fuel and/or clad temperature (until the first fuel relocation and the first clad failure respectively)
- Core exit temperature
- Core mass flow rate
- Hydrogen generation: flow rate and total mass (the latter can be directly calculated from the former)
- Fission products release rate: these quantities can be directly calculated thanks to the fission product mass flow rate mentioned above and data provided by ISODOP (see section "outputs").
- Corium description in the core:
    - Temperature,
    - Mass of intact core, magma and debris at different meshes of the core (5 radial rings and 15 axial levels)
- Corium description in the lower plenum:
    - Temperature,
    - Mass of oxide pool, heavy metal pool, light metal pool and debris (the total mass can be directly computed from these quantities)
- Void fraction in all meshes of the VESSEL (these quantities are only necessary for the 3D display of the reactor, which is not a priority for SM)

Most of these magnitudes are not directly given by ICARE and will be calculated by SENSORs defined ad hoc. The surrogate models may calculate the SENSORs directly, or the variables needed by ASTEC to compute them.
3D visualisation is not a priority of the project and would require the void fraction to be displayed for all components and meshes in ASTEC.

Input variables from other ASTEC modules

The input variables are defined in the CONNECTI from the PRIMARY to the VESSEL. These variables represent the thermal-hydraulic state of the primary circuit at the cold leg:

- Liquid temperature.
- Gas temperature.
- Liquid velocity.
- Gas velocity.
- Void fraction.
- Partial pressures of steam and hydrogen,
- Nuclear (for a short period of time before the reactor trip) and residual power.

It is worth noting that the fission products returning to the vessel from the cold leg are directly transferred to the core outlet by SOPHAEROS. ASTEC has no model for Fission Product retention or transport into the vessel region. Thus, Fission Products have no interaction with the core or other structures in the vessel once they have been released.

-

Input variables from operator's actions

The operator actions considered in the frame of the project are defined in the specifications of the simulator (approximately a dozen). The most significant actions are the following

- Water injection from portable equipment into the RCS and/or SGs.
- Pressurizer and SG relief valve opening.
- Induced break of the SG tubes, triggered by the instructor.

Besides these operator actions, several passive or automatic systems can actuate in the plant after an accident or transient. The actuation of these systems can lead to changes of large magnitude in the boundary conditions of the vessel. The considered systems actions are:

- Opening of safety valves of SGs.
- Opening of safety valves of the pressurizer.
- Discharge of the accumulators.
- Induced primary circuit breach.
- Coast down of the reactor coolant pumps.
- Reactor trip (automatic in the considered sequences).
- Actuation of the AFW of the SG (LOCA scenario only).

None of these operator actions, passive systems, or automatisms directly interact with the vessel but with the boundary conditions (e.g., the water inlet flow, the primary pressure, etc.). Therefore, it is hoped that the surrogate model will be able to capture the effects in the core and vessel of any system activation through the input variables. Considering the state of the different safety systems as inputs may improve the accuracy of the surrogate model by giving more contextual information. However, it may affect its generalisation capabilities, for example if new safety systems are included in the simulator. So, both approaches should be tested.

Additional internal state variables

Internal variables are defined as variables used by ASTEC to describe the state of the vessel without being communicated to other modules or to the simulator interface. They account for the history of the core degradation and influence the transfer function between inputs and outputs. Several approaches can be explored to treat them appropriately.

A first approach would be to calculate them comprehensively to stay as close as possible to the physical models. However, this would increase the dimensionality of the problem dramatically, whereas some of the variables are redundant. A study has been carried out to identify a reduced list of variables that enable a perfect reproducibility of calculations in ASTEC. All other quantities can be computed without loss of accuracy from this so-called "filtered database".

The filtered database is a good start to identify a list of independent variables describing the internal state of the vessel. A statistical analysis, including possibly dimension reduction techniques, may show that most information is carried by a reduced number of variables (or combinations of them), with an acceptable loss of precision. Adding more context data (data that is not used directly by the physical models but that may be correlated with the results) may improve results as well. Expert assessment can also be relied upon to identify the variables of

primary importance to describe the core degradation (in addition to the already identified inputs), like:

- Clad oxidation.
- Blockage (portion of the channel available for gas circulation), porosity (fraction of the mesh volume available for the coolant flow) and hydraulic diameter (ratio of the section of the mesh to the wetted perimeter), which account for the blockage of the channel.
- State of the clad (intact or cracked). Fraction of failed claddings.
- Residual power, which can be:
    - Assessed globally (inferred from the delay since the reactor trip and the inventory loss of the vessel),
    - Computed from the composition of the core and the specific radioactive power released by each element (computed by ICARE).

ICARE also computes the composition of the components of the core, including the different layers of the cladding (different metallic and oxide compounds), whose number varies during a calculation. Considering such variables seems to be a substantial complication, so the more general features mentioned above will be preferred in a first stage.

These variables represent the internal state of the vessel and are calculated in each mesh of the vessel. They would be helpful to know the state of the vessel and can facilitate the correct evaluation by the surrogate model of output variables. They can be considered extensively or summarised by synthetic figures of merit. The later can be determined by expert judgment or calculated automatically using dimension reduction techniques.

Data generation

To successfully train the surrogate model, the training database must sufficiently cover the period of the core degradation. Particular attention should be given to magnitudes, such as cladding and fuel temperature, which have a cliff-edge effect, for example, for the fission product release.

Different safety systems can be activated automatically or by the operator from the accident's beginning until the extensive core degradation. They do not directly affect the core conditions but the boundary conditions of the primary circuit.

Surrogate Modelling

Given that the vessel provides as 2D grid representation (Axis , Channel) of the current state (all variables) at time, we will model the spatial and temporal behaviour simultaneously across all variables, instead of treating each variable at each spatial position as an individual univariate time-series. The reason behind this is two-fold: for one, we expect spatial correlation and interaction of the variables in neighboring grid cells, e.g. a rise in temperature in one place will likely cause a temperature rise in the surrounding parts of the vessel.
Furthermore, since we are dealing with physical variables, we expect dependencies and correlations between variables, e.g. between pressure and temperature. Hence, we will model the entire state of the vessel over time. The surrogate model will take the 2D grid of all variables to be modeled over the last n timesteps as input, and predict the next timestep. This model can then be utilized in an autoregressive manner, to produce long term predictions of multiple time-steps. Since such an autoregressive mode accumulates errors relatively fast, we will train the model with an adaptive training curriculum, where in the initial phases we optimize the prediction of one timestep prediction, and in the later phases gradient updates are performed on successively increasing autoregressive prediction windows.

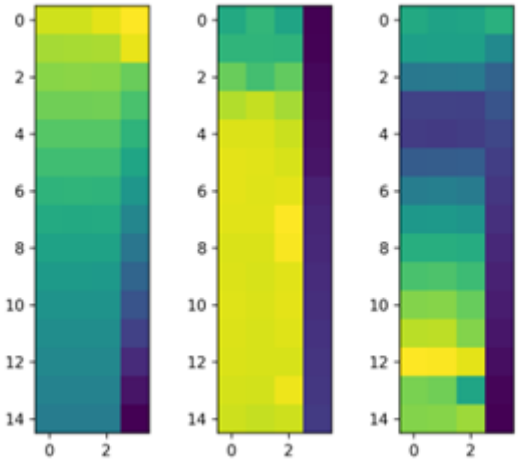


Figure 10: 2D grid of 4 channels and 15 meshes for three different variables (T_sat, T_liq, void)

We will rely on existing neural network architectures to address this task of modelling a 3D (2D + Variables) system over time. A prominent example of such a task is numerical weather prediction, where several surrogate model approaches have recently been published.
Taking analogy from neural operator learning, its advancement into Fourier Neural Operators (FNO) and adaptation to computer vision problems with Vision Transformers, we will utilize an Adaptive Fourier Neural Operator (AFNO) network. This architecture has been successfully applied to weather prediction in form of FourCastNet, for which trainable model code is publicly avaliable on github.
We have already adapted this code to our modelling of the vessel, including a respective dataloader. An example training curve of the training and validation mean absolute percentage error (MAPE) against the persistence baseline is shown in figure 11. Due to the high number of model parameters and the low number of currently available training samples, current test runs do not yet provide sufficient model accuracy.

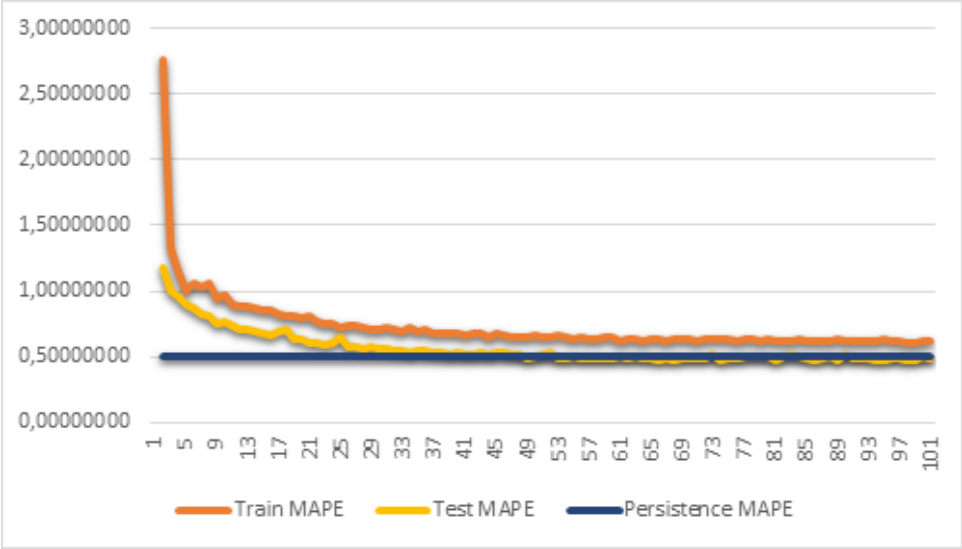


Figure 11: Training of AFNO Transformer on exemplary data set

Post-vessel rupture containment

We are developing surrogate models of the behaviour after reactor pressure vessel (RPV) rupture (phenomenological time TRUP in ASTEC). With the model implemented in the simulator, the vessel will collapse (time called TCOLLAPSE in ASTEC) shortly after its rupture. The corium will then be poured into the reactor cavity and the Molten Core Concrete Interaction (MCCI) will start (TMCC in ASTEC). At that point, we are only interested in phenomena in the containment. The ASTEC modules CESAR and ICARE are turned off to speed up the simulation, since the RCS and vessel dynamics are no longer the priority. In this stage, the time frame of the simulation switches from hours to days. Accordingly, the temporal resolution of the simulation also becomes coarser. The relevant ASTEC modules at this stage are: CPA, SOPHAEROS, MEDICIS and ISODOP.

The principal requirements for a surrogate model of post-rupture containment are:

- the ability to predict the time evolution of a relatively small number of quantities, listed later,
- a very high speedup factor - the simulation can cover days of real time dynamics.

General approach

We identify two different approaches to a surrogate for post-rupture containment**: Global model and time-stepping**, as explained in "Formulation of modelling problems"**.**

The global model approach is the favoured option for post-rupture containment surrogate models. The reasons for this preference are the following.

1. A global model requires more data to learn complex dynamics than a time stepper. The dynamics in the post-rupture phase are relatively simple compared to the early stages of the accident, which makes a global model more feasible.
2. The number of operator actions available in the post-rupture regime is small – only a handful, which makes the design of a global model simpler.

The remainder of this report will focus on the details of the global model formulation. Time stepping remains an option should significant issues arise during the development of a global model.

Data

The analysis of variables in this report is based on simulations produced during the Euratom project **FAST Nuclear Emergency Tools (FASTNET).** The input deck used in FASTNET is not identical to the one in ASSAS. While details may vary, they are similar enough to provide insight into the details of the considered surrogate modelling problem.
Since this report addresses only the post-rupture phase, we discarded scenarios where vessel rupture did not occur. This left us with 14 scenarios.

The FASTNET runs have a different simulation length. To simplify analysis and prepare the data for the first machine learning attempts, we modified it to ensure the same number of time points for each scenario. We achieved this by padding each variable with its last value to the longest simulation time in among the 14 scenarios.

Output variables

The surrogate model predicts a number of quantities as time series. We expect to predict a fixed number of time points and use interpolation to find intermediate values if required.
Based on the specifications of the simulator, we expect to limit the time horizon to one week. The time discretization is expected to be between 10 s and 50 s.

Denoting the set of output quantities as $y_1, y_2, \dots, y_m$, and the time discretization as $t_1, t_2, \dots, t_n$, the outputs of the surrogate model are $y_1(t_1), y_2(t_1), \dots, y_m(t_1), y_1(t_2), y_2(t_2), \dots, y_m(t_2), \dots, y_m(t_n)$.

The output variables are defined in the specifications of the simulator.
Reactor building (containment zone "dome"):

- Pressure (*PRES*) and temperature (*TFLU*)
- Concentrations of steam, H_2, O_2, CO and CO_2 (five real values: XH20, *XH2, XO2, XCO, XCO2 → to be updated: SENSORS may be defined for that purpose*)
- Amount of H2 and CO recombined by the recombiners (two values)
- Containment spray flow rate (one value)
- Containment venting flow (one value)
- Activities of the fission products: Xe-133, I-131 (organic, inorganic, aerosol), I-132 (organic, inorganic, aerosol), CS-134, Cs-136, Cs-137, Te-132, Sr-90, Ru-103, La-140, Cm-242, Pu-241, Mo-99, Sb-127 (18 variables)
- Total dose rate (one real value)

After the vessel rupture, ASTEC does not compute the release of fission products from the circuits to the containment, so the corresponding value displayed by the simulator will be null.

Cavity ("CAVITY")

- Axial and radial concrete erosion (two real values)
- Description of the corium in the cavity (three values):
    - Temperature,
    - Mass of oxide and crust (from which the total mass can be computed).

Sump (containment zone "pool"):

- Water level and temperature (two real values: *WLEV, TLIQ*)

Environment:

- FP activity (same list as for the reactor building dome)
- Release rate of fission products to the environment (total activity in Bq/h)

The current list features 56 real-valued quantities. Since they are values from individual containment zones, there is no exploitable spatial structure.

Figure 8 depicts the trajectories (time dependence) of the output variables for 14 scenarios from the FASTNET data. Most of the variables are well-behaved, without wild jumps or oscillations. Reactor building pressure and temperature, as well as hydrogen concentration, tend to undergo a single peak before settling on a stationary value. On the other hand, activities and FP inventories tend to either increase or decrease monotonically.

Particular attention should be given to sump water level and temperature. In all but one of the scenarios, the sump was empty throughout the post-rupture phase. In such cases, the water temperature is meaningless. In one scenario, the water level drops extremely rapidly. This suggests that water level may be better modelled as a binary variable, indicating whether the sump is full or empty. A possible suggestion for building surrogate models is to engineer the machine learning reward function so as to ignore water temperature when the sump is empty.

Finally, we observe a very steep jump in the behaviour of Caesium activity in one of the scenarios. It may be prudent to investigate whether this is expected physical behaviour or an artefact of the simulation.

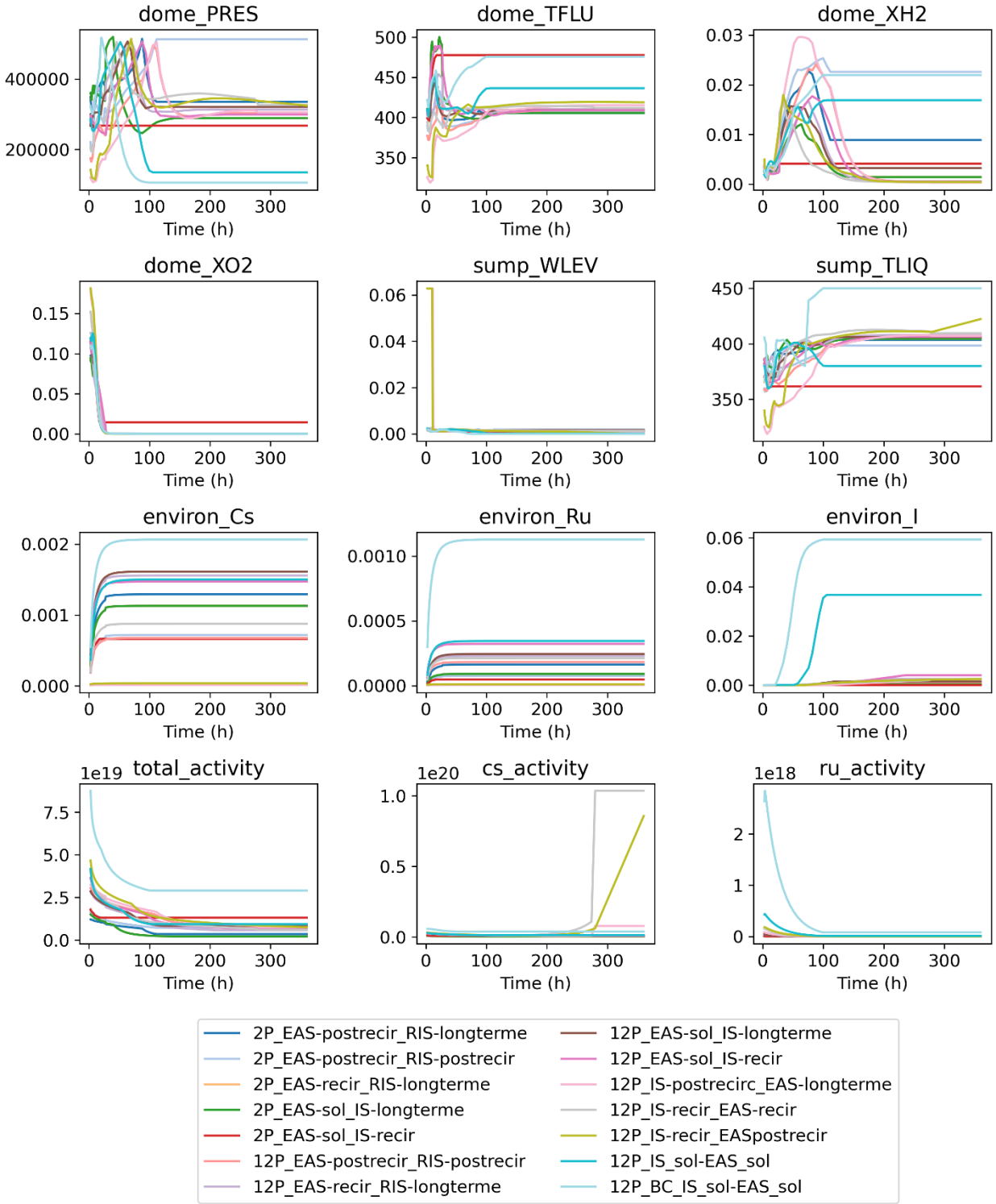


Figure 8. Trajectories of output variables for 14 scenarios from the FASTNET data.

Input variables

The main input of the surrogate model are the values of various physical and chemical quantities at the time of vessel rupture (since after this, ASTEC is turned off and most of these quantities are not simulated anymore). The choice of quantities to include is open to experimentation. The first tentative list of options includes:

- Pressure, and gas temperature in the containment (2 real values in 18 zones),
- Atmospheric composition (6 independant real values in 18 zones)
- Liquid and gas velocity in the containment (1 or 2 real values in 54 connections between zones)
- Wall temperature, with an in-depth profile (1 real value in 62 walls x up to 10 in-depth layers)
- Corium: mass and composition of the crust and the liquid phase (possibly summarized with mass, density, residual power, time elapsed since the reactor trip)

Potential candidates for inclusion are also:

- The full element inventory, with its alpha, beta, gamma specific power and activity (77 elements x 6 real = 462 real values in 2 domains, namely "containment" and "environment")
    - ASTEC considers more than 700 isotopes in each DOMAIN for its physical description, but these quantities can probably be summarised with the variables mentioned above for simplicity, including the time elapsed since the reactor trip.
- The full FP species inventory (2196 real values in 18 zones)
- Module connections (heat flux, source flow, especially from the Reactor Water Storage Tank for the LOCA scenario etc.)

Table 3. Statistical summaries of related input variables.

	Pressure (Pa)	**Gas temperature (K)**	**Atmospheric composition (kg)**	**Fluid, gas velocity (kg/s)**	**Wall temperature (K)**	**Element inventory (kg)**
Mean	$202 \cdot 10^3$	346	$0.15 \cdot 10^9$	5	369	0.5
Standard deviation	$99 \cdot 10^3$	45	$1.1 \cdot 10^9$	40	28	2.7
Max	$359 \cdot 10^3$	438	$9.2 \cdot 10^9$	657	437	56
Median	$175 \cdot 10^3$	354	26	0.04	378	0
Min	$97 \cdot 10^3$	287	0	-30	288	0

The table above presents statistical summaries of the identified input variables. For this analysis, we grouped quantities with the same physical dimensions. The summaries reveal potentially troubling properties for atmospheric composition variables, which feature a large discrepancy between the mean and median, indicating a skewed distribution with severe outliers. Similarly, the majority of element inventory variables are zero. Note that the FASTNET database is limited, featuring only a small number of scenarios. It is possible that these properties would not appear in a more representative database. Furthermore, it is expected that composition variables exhibit a wide range of orders of magnitude. As such, preprocessing, such as a logarithmic transformation, may be useful for model inputs. The inventory of radioactive elements can also be represented as activities, rather than mass. In any case, particular attention should be given to the preprocessing and representation of atmospheric composition and element inventory

variables, with a possible reconsideration of which elements are interesting enough to warrant inclusion among surrogate model inputs.
Furthermore, fluid and gas velocities in zone connections exhibit a high standard deviation in comparison to the average. In other studies (see surrogates of steam generators), very erratic behaviour of velocities with time was observed. Aggregation or smoothing operations may be useful when preprocessing these variables for surrogate model development.

Operator actions

Operator actions are interventions that power plant operators can take in real time during the accident or simulation. For example, in the post-rupture phase, an operator may decide to engage the containment spray system once it has been recovered (LOCA scenario only), which can have a significant effect on the dynamics in the containment. For a surrogate model, operator actions represent a special type of input and require particular consideration. The full list of operator actions is given by the simulator specifications:

- Actuation of containment spray systems after their recovery (LOCA scenario).
- Containment venting.
- Combustion of gases (provided that flammable conditions can be reached, which should not happen with the plant model used for the simulator; this will be checked during the generation of the training database)

These actions are binary – at a given moment, they either happen or do not.

A time stepping surrogate model can incorporate an operator action rather simply by including its binary variable among its inputs. Since the prediction proceeds in steps, the engagement of an operator action is visible to the surrogate model as it happens.

Addressing operator actions in a global model represents a greater complication, since the action can be engaged long after vessel rupture, when the detailed physical conditions in the containment are no longer simulated. At this time, we identified two possible approaches to addressing this:

1. Represent the operator action by the **time of engagement**. An unused action may be represented by a time of engagement, larger than the time horizon of the prediction. For example, a surrogate model at the time of vessel rupture receives the input $t_{CSS} = inf$ alongside the variables identified in Section 5 and predicts the trajectories y_1. The containment spray system is engaged 1 hour after vessel rupture, at which point the model is run again, this time with the input $t_{CSS} = 1h$ alongside the conditions at the time of vessel rupture. It predicts new trajectories y_2. We then use y_1 for $t \leq 1h$ and y_2 for $t > 1h$.
   This method extends simply to more than one type of operator action through the introduction of additional input variables.

2. Train a **separate surrogate model** for each operator action that predicts trajectories based only on the trajectories of outputs, identified in Section 4. For example, a surrogate model would at the time of vessel rupture predict trajectories based on the inputs from Section 5. The containment spray is engaged 1 hour after vessel rupture, at which point the second surrogate model is engaged, which takes as input the first hour of the predictions of the first surrogate model, and predicts the remainder of the time horizon.

   This method extents to more than one type of operator action by training separate surrogate models and chaining their predictions, as long as the operator actions always happen in the same order. If the order can change, surrogate models must likely be

trained for each possible ordering of operator actions, which can represent a significant complication.

A further complication with this approach is that the surrogate models which take as input trajectories of output variables from Section 4, require different machine learning methods than the surrogate model which takes as input the physical conditions at the time of vessel rupture (input variables from Section 5).

Data generation

To successfully train surrogate models, a large training dataset with a good coverage of possible physical conditions is required. Since the simulation of the early stages of the accident is computationally very costly, it is prudent to consider starting the simulations at the time of vessel rupture. We consider two ways of achieving efficient data generation for the post-rupture phase:

1. First, perform a **large** number of full simulations, **densely** covering various early-accident scenarios and operator actions. Then, expand each of the obtained simulations by restarting it at the time of vessel rupture and sampling the available post-rupture operator actions and their timings. In this way, different physical conditions at vessel rupture, as well as different post-rupture dynamics are covered.
   The downside of this approach is that it requires a lot of computational effort to perform a large number of full simulations.

2. First, perform a **small** number of simulations, **sparsely** covering various early-accident scenarios and operator actions. Then, expand each of the obtained simulations by restarting it at the time of vessel rupture and modifying the conditions at the time of vessel rupture by sampling physical quantities in the neighbourhood of their simulated values. Finally, expand each of the restarted simulations by sampling post-rupture operator actions and their timings. In this way, the smaller number of full simulations is alleviated by the sampling of physical conditions at the time of rupture.
   The downside of this approach is that it may produce a number of simulations with physically impossible conditions, which can be difficult to detect and may have a negative effect on the performance of surrogate models. Dimension-reduction techniques may help to identify the latent space in which initial conditions vary: sampling parameter in this reduced domain may help to identify more physical conditions.

Global surrogate of MELCOR

Problem definition

KTH is developing machine learning methods and tools to replace the MELCOR model of a complete reactor and achieve faster than real-time calculation performance. The modelling will consider all in-vessel and ex-vessel phenomena happening before and after vessel failure. MELCOR has several packages that model different parts of the reactor (COR, RN, HS, CAV, CVH, etc.). Developing separate Surrogate Models (SM) for each of the packages may not be possible due to the lack of the access to local variables during runtime. Therefore, we aim to develop an SM of the complete system. Our task is to develop a SM, that takes into account the nature of accident (LOCA or SBO), to assess,

1. Core damage.
2. Relocation of molten material.
3. Vessel and containment failures.
4. Ejection of debris.
5. Fission products inventory and release.
6. Pool conditions.
7. Ex -vessel phenomena – steam explosion, debris bed coolability, pool stratification.

The evolution of the system conditions together with frequently used operator actions will present the effectiveness of said actions in terms of cumulative distribution functions of the fission product release. Practical challenges foreseen in the development include:

1. Time dependency of input and output parameters: how the SM should incorporate it during execution.
2. Generation of the adequate training database:
    a. long running time of MELCOR full system simulation will require a trade-off between the number of simulations and SM accuracy.
3. Selection of an adequate SM model architecture to replicate MELCOR results.

The general approach

A global MELCOR SM can be approximated as an operator, where time dependent initial and boundary conditions are mapped onto the time dependent results. Our task is to develop such an operator. There are several machine learning methods we are considering for the implementation of the MELCOR SM:

1. Data approximation techniques:
    a. E.g. Hierarchical time-steppers HiTSs based on multi-scale flow maps and hierarchical neural network architecture. These methods suffer the curse of dimensionality in transient scenarios. Few events with uncertain order and timing create infinite number branches. Extensive data (transients) might be needed to anticipate all possible actions by the operator.
2. Physics-informed neural networks (PINNS)
    a. The error function incorporates output error and input/output operator error. Such an approach improves training on sparse data sets. The equations describing the physical phenomena must be known, which is not the case for MELCOR.
3. Neural operators (approximates the operator, not data):
    a. Fourier operator (Lu et al. 2022): Has a specific architecture of the branch and a trunk represented by a Fourier series.

b. Maps between spaces of function (Kovachki et al. 2021): Finite dimensional parametric approximation of an operator.
c. DeepONet (Lu et al. 2019): Can be used for simulating dynamics of a complex nonlinear system, corresponding to different boundary and initial conditions without the need for retraining the neural networks. Large data samples are required. If G is an operator taking an input function u, and G(u) is the corresponding output function, for any point y in the domain of G(u), the output G(u)(y) is a real number, as given by Universal Approximation Theorem for Operator illustrated in Figure I below.

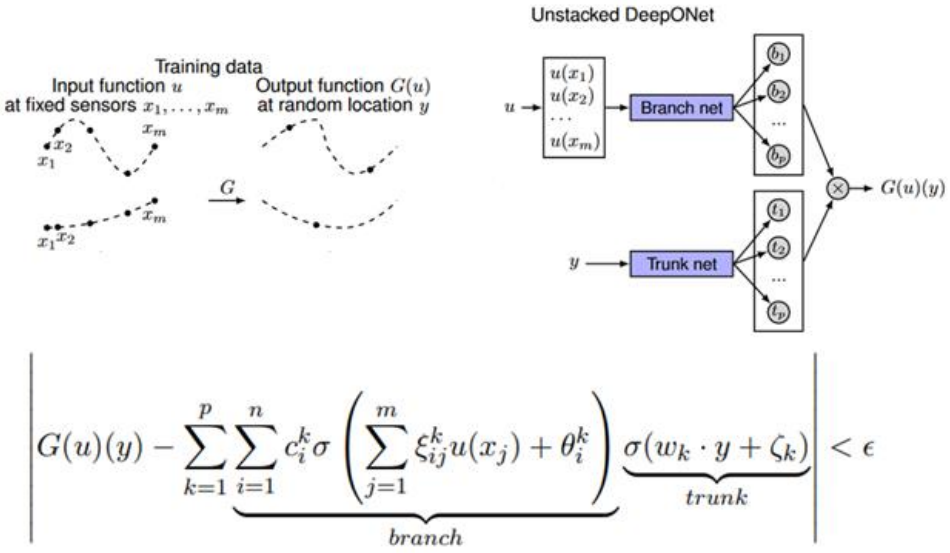


Figure 13. DeepONet architecture [Lu et al, 2019]

Here G functions as an operator for MELCOR, hence, the network takes inputs composed of two parts: u and y, and outputs G(u)(y). The operator approximates solution in spatial and temporal coordinates. Once DeepONet is trained, it can be applied to new input functions, thus producing new results substantially faster than numerical solvers, with low generalization error. Our main focus in this project will be on the developing of DeepONet operator for the MELCOR model of a complete reactor.

Open questions in the implementation of DeepONet for approximation of MELCOR operator are

1. DeepONet requires large input space for training, combining with PINNS may help to reduce the requirement for the size of the input space.
2. Training to predict time step operator. Using different architecture than original HiTSs may provide better solution.

Data

The data of the accidents using MELCOR for the training of SM, will be based on simulations of Nordic BWR. Data from 760 MELCOR simulations of SBO scenarios and 800 MELCOR simulations of LOCA scenarios are now available. The simulations were run for 72hr to cover short- and near-term effects of operator actions.

Input variables

SM will be developed for the whole code. Selection of relevant (needed) parameters for the training of SM will be a non-trivial task requiring a trial and error approach. The total number of parameters is expected to be < 100. Preliminarily, these variables are included in the input, representing the physical, thermal-hydraulic and chemical characteristics of the core, auxiliary systems, cavity, and pool:

1. Pool and fog temperatures.
2. Pool and fog levels.
3. Pool and fog pressures.
4. Gas velocity.
5. Void fraction.
6. Partial pressures of steam and hydrogen.
7.

As a part of the solution database generation the input parameters from the following models were varied:

1. Fission product (FP) release from fuel,
2. Core degradation and relocation,
3. Reactor pressure vessel lower head failure,
4. FP and aerosol dynamics,
5. Spray and pool scrubbing and filters trapping.

50 listed parameters from these models were screened and subsequently 28 were considered as inputs in the sensitivity and uncertainty analysis. These are listed in the table below.

	L	S	PARAMETER	MODEL	DEFAULT	RANGE MIN	RANGE MAX	UNIT
1	*	*	SC710641	CORSOR-Booth diffusion model	381400	241000	381400	J/kg-mole
2		*	SC710651		0.000006	0.000006	0.00001	m
3	*		TUO2ZRO2	Eutectics model	2450	2450	2800	K
4	*		FCELRA	Intercell radiation model	0.25	0.1	0.25	
5		*	PDPOR	Debris formation	0.3	0.3	0.5	
6	*		HFRZSS	Candling model	1000	1000	2500	W/m2-K
7	*	*	SC11312		2400	2100	2500	K
8		*	TPFAIL	Vessel failure model	1273	1273	1600	K
9	*	*	HDBPN		1000	100	1000	W/m2-K
10	*	*	GAMMA	Aerosol dynamics model	1	1	3	
11	*	*	STICK		1	0.5	1	
12		*	RHONOM		1000	1000	4900	kg/m3
13	*	*	TURBDS		0.001	0.00075	0.00125	m2/s3
14	*		SC7111I1	Vapour diffusivity model	4.982	4.2347	5.7293	A
15	*	*	SC7111I2		550	467.5	632.5	K
16		*	SC7111CS1		3.617	3.0745	4.1595	A
17	*		SC7111CS2		97	82.45	111.55	K
18		*	SC7170CS	Hygroscopic aerosol model	3.95	3.3575	4.5425	kg/kg H2O
19		*	SC7170CSI3		0.44	0.374	0.506	kg/kg H2O
20	*		SC7170CSI4		2.25	1.9125	2.5875	kg/kg H2O
21	*	*	SC7170CSM		0.67	0.5695	0.7705	kg/kg H2O
22		*	SC715111	SPARC-90 globule size correlation	3.45	2.9325	3.9675	
23	*		SC71521	SPARC-90 bubble shape model	0.007	0.005	0.008	m
24	*		SC71531	SPARC-90 bubble rise velocity model	7.876	6.6946	9.0574	cm/s
25	*		SC71551	SPARC-90 particle impaction model	1.79182	1.523	2.0606	
26		*	SC71555		1.13893	0.9681	1.3098	
27	*	*	SC71542	SPARC-90 swarm velocity model	0.003011	0.0025593	0.0034626	I-s/cm2
28	*		SC3210	ORIGEN decay curve multiplier	1	1	1.15	I-s/cm2

These parameters together with the thermo-physical parameters should be enough to describe the state of different systems.

For the input related to the time dependent operator actions see the respective section below.

Output variables

The DeepONet model outputs only one parameter (figure of merit [FOM]) at a time. For the purposes of source term analysis, cesium, iodide fractions, hydrogen generation, and timing of vessel failure are calculated. This selection will change as the project progresses, and the simulator specifications are finalized. Some of the simulator outputs identified are,

Reactor building:

1. Mass flow rates,
2. Pressure and temperature,
3. H2 (generation rates, concentration is CVs) and O2 concentration,
4. Total dose rate,
5. Mass of debris in lower head,
6. Water level and core temperature (avg and max).

Wetwell/drywell:

1. Melt release mass and rate and temperature,
2. Water level and temperature.

Response function (FOM):

1. CS release fraction,
2. I2 release fraction,
3. FP inventory: Cs, Ru, I (three real values).

The list of output variables will be updated with the progress of the project.

Operator actions

Operator actions, representing real-time interventions, are crucial inputs for a surrogate model. The operator actions considered within the project framework are outlined in the simulator specifications, numbering approximately a dozen. The key actions include:

1. Water injection from portable equipment into the reactor coolant system and/or steam generators.
2. Opening of pressurizer and steam generator relief valves.
3. Induced break of steam generator tubes by cold water injection, possibly resulting in a primary circuit induced break, initiated by the instructor.

In addition to operator actions, several passive or automatic systems may actuate in the plant post-accident or transient systems. These system actions can lead to significant changes in vessel boundary conditions. Modelling operator actions in full code SM is difficult. We will attempt to:

1. Represent the operator action by the time of engagement, extending to different actions with additional input variables.
2. Train separate surrogate models for each operator action, predicting trajectories based on previous outputs.

The latter method requires different machine learning methods for models based on output trajectories compared to those relying on physical conditions at the time of vessel rupture. The latter method may fit well with the type of data that the SM will be able to track (in terms of CDFs of FOMs).

Data generation

To effectively train surrogate models, a substantial training dataset that comprehensively covers potential phenomena and physical boundary conditions is essential. Given the computational expense of simulating the large number of accidents for various operator actions, we propose to conduct a significant number of full simulations, densely covering various early-accident scenarios and operator actions. Have the trigger for the operator action at appropriate time periods, to end with splinter scenarios. Where each SM could, for example, track a particular splinter scenario.

Conclusions

In this document we overviewed the methodology to be employed in ASSAS. We discussed the roles of different modules of ASTEC and classified the many different possible formulations of surrogate modelling approaches. We identified the challenges associated with each formulation and its potential for speed-up. Based on these factors, we decided on a selection of most promising surrogate modelling approaches to address in ASSAS.

For each of the selected approaches, we detailed the formulation of the surrogate modelling problem and identified the inputs and outputs to the surrogate models. We further analysed small data samples and studied the characteristics of different variables, which helped inform the strategy for generating data in the future. The information collected in this report represents a necessary foundation for the development of surrogate models.

The analysis of the advantages and disadvantages of different surrogate modelling approaches in this report is based on a consideration of the computational efficiency of different ASTEC modules, the properties and the dimensionality of different physical systems and the connections between them, experience with different machine learning methods and experience with modelling severe accidents. More in-depth practical insights on the advantages and disadvantages of considered approaches will be gained during the development of surrogate models and provided in a report on methodology at the end of the project.

References

[1] Isabel Parrado-Rodriguez, Users' expectations – Simulator specifications, ASSAS WP6 report, version 1, 27/02/2024
[2] Lionel Chailan, Loic Bosland, Laure Carénini, Julien Chambarel, Frederic Cousin, et al., "Overview of ASTEC integral code status and perspectives," 9th European Review Meeting on Severe Accident Research (ERMSAR2019), Prague, Czech Republic, March 2019, doi: ⟨irsn-04106726⟩
[3] Antoine Lechevallier, Sylvain Desroziers, Thibault Faney, Eric Flauraud, Frédéric Nataf. Hybrid Newton method for the acceleration of well event handling in the simulation of CO2 storage using supervised learning. 2023. ffhal-04085358v3f

[4] Louis Proffit, "Internship report – ASSAS project" (25th July 2023), https://assas-horizon-euratom.eu/documents-and-data/

[5] W. Ambrosini, R. Bovalini, F. D'Auria, 1990. Evaluation of accuracy of thermal–hydraulic code calculations. Energia Nucleare 7, 5–16.

[6] Pietro Maccari, Andrea Bersano, Stefano Ederli, Fabrizio Gabrielli, Fulvio Mascari, 2023, Validation and uncertainty analysis of ASTEC in early degradation phase against QUENCH-06 experiment, Nuclear Engineering and Design, 414, 112600.

[7] Soheil Kolouri, Se Rim Park, Matthew Thorpe, Dejan Slepcev, Gustavo K. Rohde, Optimal Mass Transport: Signal processing and machine learning applications, IEEE Processing Magazine, July 2017.