NIST Technical Note
NIST TN 2332

Fire Data Generator (FD-Gen) v1.0.0

An automated tool to generate multiple FDS input files

Hongqiang Fang
Wai Cheong Tam

This publication is available free of charge from:
https://doi.org/10.6028/NIST.TN.2332

NIST NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

NIST Technical Note
NIST TN 2332

Fire Data Generator (FD-Gen) v1.0.0

An automated tool to generate multiple FDS input files

Hongqiang Fang
Wai Cheong Tam
Fire Research Division
Engineering Laboratory

This publication is available free of charge from:
https://doi.org/10.6028/NIST.TN.2332

March 2025

U.S. Department of Commerce
Howard Lutnick, Secretary

National Institute of Standards and Technology
Craig Burkhardt, Acting Under Secretary of Commerce for Standards and Technology and Acting NIST Director

NIST TN 2332
March 2025

NIST Technical Series Policies

Publication History
Approved by the NIST Editorial Review Board on 2025-02-27

How to Cite this NIST Technical Series Publication
H. Fang, W. C. Tam, (2025) Fire Data Generator (FD-Gen) v1.0.0. (National Institute of Standards and Technology, Gaithersburg, MD), NIST TN 2332. https://doi.org/10.6028/NIST.TN.2332

Author ORCID IDs
Hongqiang Fang: 0000-0002-9508-0853
Wai Cheong Tam: 0000-0003-3913-0243

Contact Information
waicheong.tam@nist.gov

Abstract

This document serves as the documentation for the Fire Data Generator (FD-Gen), an automated tool designed to streamline the creation of multiple Fire Dynamics Simulator (FDS) input files. By employing Monte Carlo methods to sample relevant fire parameters, FD-Gen creates diverse fire scenarios and seamlessly integrates them into FDS case input files. This tool is designed to efficiently produce large volumes of synthetic fire data, supporting the development and training of machine learning models for smart fire detection and forecasting applications. Additionally, FD-Gen may offer practical benefits for applications such as fire risk assessment and fire investigation, broadening its impact across fire safety and research domains.

Keywords

Fire scenarios, Fire simulation, FDS input files, Monte Carlo, Data sampling, Automated tool

Table of Contents

List of Tables

List of Figures

Author Contributions

Hongqiang Fang: Conceptualization, Methodology, Software, Validation, Visualization, Writing- Original draft preparation; **Wai Cheong Tam**: Conceptualization, Methodology, Supervision, Writing- Reviewing and Editing.

1. Introduction

1.1. Background

Smart fire detection and forecasting is an emerging and promising field within smart firefighting research. Its primary objective is to identify and predict various aspects of fire characteristics, such as fire location and fire intensity. This predictive information has the potential to enhance firefighting and rescue operations by improving situational awareness and strengthening the decision-making abilities of firefighters and incident commanders. Notable application areas include smart fire source localization [1], fire intensity estimation [2], flashover prediction [3], and firefighters' tenability analysis [4].

The use of machine learning (ML) is becoming increasingly prevalent in smart fire detection and forecasting research [5-9]. ML's powerful capability to capture intricate patterns within data has greatly improved the accuracy and efficiency of these tasks. However, developing such models typically requires large amounts of data for effective training, which presents a challenge due to the rarity of fire events and the difficulty in collecting full-scale real-world data. To address this issue, a commonly employed approach is the use of synthetic fire data for ML model development. This synthetic fire data, generated through simulation tools or other techniques, helps to overcome data scarcity, enabling more robust and effective model training.

At present, two widely used fire simulation tools are Fire Growth and Smoke Transport Modeling (CFAST) [10] and the Fire Dynamics Simulator (FDS) [11]. CFAST is a two-zone fire model used to calculate the evolving distribution of smoke, fire gases and temperature throughout compartments of a building during a fire. FDS is a computational fluid dynamics (CFD) model for low-speed flows, with an emphasis on smoke and heat transport from fires. Both tools are developed and maintained by the National Institute of Standards and Technology (NIST) and have undergone extensive verification and validation, ensuring their accuracy and reliability for fire modeling and research applications. Previous studies have employed both simulation tools to generate fire data for ML model development. **Table 1** summarizes the simulation tools and data sizes used in earlier research focused on smart fire detection and forecasting.

Table 1. Summary of the simulation tool and data size employed in previous research.

Year	Author	Problem statement	Synthetic fire data		Reference
			Simulation tool	Sampled fire scenarios	
2020	Tam et al.	To develop classification models that can predict the location of a fire in multi-room compartments	CFAST	1 000	[12]
2020	Wu et al.	To predict the location, size, and ventilation wind speed in a tunnel fire	FDS	100	[2]
2021	Wang et al.	To develop a model for forecasting the potential occurrence of flashover in a single-floor, multi-room compartment fire	CFAST	1 000	[6]

2021	Kou et al.	To determine fire locations and intensity in a complex building with multiple compartments	CFAST	700	[8]
2021	Fang et al.	To identify the stages of fire development in a multiple compartment room	CFAST	6 000	[13]
2021	Wu et al.	To predict real-time ceiling temperature profile in underground tunnel fires	FDS	100	[5]
2022	Wu et al.	To predict fire size and location of tunnel fires with a small-scale demonstration	FDS	400	[14]
2023	Tam et al.	To predict the potential occurrence of flashover in a traditional single-story ranch-style family room	CFAST	110 000	[3]
2023	Fang et al.	To identify the room of fire origin in a multiple-room commercial building	FDS	480	[1]

Although fire simulation has become a valuable tool for generating synthetic fire data with desired characteristics across various fire scenarios, it introduces another challenge: how to efficiently generate the case files for the simulations. Given the large number of simulation cases required to develop an adequate dataset for ML models, manually adjusting parameters and creating case input files one by one is impractical. To address this issue, CFAST currently offers a solution with using the CFAST fire data generator (CData) [15], which automates the generation of CFAST case input files by sampling values from a list of predefined parameter ranges. However, for FDS, this process still largely depends on manually creating the case files one by one, which is labor intensive and inefficient, especially when large datasets are needed for ML model development. Therefore, developing a similar automated tool for FDS is necessary. Such a tool will streamline the workflow for creating multiple FDS case input files and standardize the formation of numerical datasets for ML model development. Additionally, this tool may also provide practical benefits for applications such as fire risk assessment and fire investigation, broadening its impact across fire safety and research domains.

However, it is important to note that while synthetic fire data provide a valuable means for training ML models in fire safety applications, the model's performance heavily depends on the quality of the data used. Users must exercise caution when using FDS to generate training data, especially if the model is intended for real-world fire scenarios. Proper verification and validation of the ML model remain essential before its deployment in practical applications to ensure reliability and accuracy.

1.2. Features of FD-Gen

The Fire Data Generator (FD-Gen) described in this document is a Python-based automated tool designed to streamline the creation of multiple FDS case input files. By leveraging Monte Carlo methods [16], the tool generates random values for key fire parameters and integrates them directly into FDS code lines, enabling users to efficiently produce a large volume of FDS case files within specified parameter ranges. The current version of FD-Gen supports the randomization of various fire parameters, including fire source locations, vent locations, door or window opening sequences, obstruction sizes, fire heat release rate (HRR) curves, and other related parameters.

The key features of FD-Gen include:

- Automated input script checking: automatically checks the FD-Gen input script for errors or inconsistencies.
- Customizable fire parameter sampling: allows users to define and customize sample distributions for various fire parameters.
- Generation of multiple FDS input files: efficiently generates a large number of FDS case input files based on specified parameter ranges.
- Fire parameter sample plotting: provides information summary and visualizations of the sampled fire parameter data, enhancing repeatability and validation.

The remaining sections of this document are organized as follows. **Section 2** provides a brief introduction to the workflow of FD-Gen, serving as a "Getting Started" guide. **Section 3** describes the namelists used to define commands and values within the program, with illustrative examples provided for each. **Section 4** explains the procedures for running FD-Gen, guiding users through the steps to execute the program. **Section 5** includes three practical examples of using FD-Gen to design multiple fire scenarios in building structures. **Section 6** concludes the document with a summary.

2. Getting started

FD-Gen is a Python-based tool designed to facilitate the sampling and generation of FDS case input files using a script-driven approach. The tool accepts a script file composed of a combination of FD-Gen namelist lines and FDS case code lines. The namelist lines specify the rules for sampling fire parameters, allowing FD-Gen to seamlessly insert sampled values into the original parameters within the FDS case code. This script file-based approach facilitates both customization and automation in creating FDS case input files. This technical note provides a comprehensive guide on obtaining and using FD-Gen effectively.

2.1. How to acquire FD-Gen

Our project provides an open-sourced GitHub repository for those interested in obtaining and testing FD-Gen. The current location of the repository is https://github.com/usnistgov/FD-Gen. The repository contains both the source code (in the main_code folder) and the executable file (**FD-Gen.exe**). The FD-Gen executable has been tested and confirmed to work on Windows 10 and Windows 11 running FDS version 6.9.1. For users utilizing the Python source code to operate, the environment requirements for the dependencies are listed below. Ensure that these dependencies are installed in your Python environment before running FD-Gen:

Python	3.11 or higher
Numpy	1.26.4
Pandas	2.2.3
Scipy	1.14.1
Matplotlib	3.9.2
Openpyxl	3.1.5

2.2. FD-Gen workflow

The basic workflow of FD-Gen is shown in **Figure 1**. It consists of four main steps.

Step 1: prepare FDS case file

The FDS case file following the FDS code[1] must be prepared. This file should include key components such as the basic building geometry, fire scenario definition, mesh configuration, surface and material properties, as well as device placement and monitoring information. These elements are crucial for accurately simulating fire dynamics and ensuring the proper functionality of the simulation. The setup process for this input file aligns with the standard procedure used in any fire simulation case in FDS. It is advisable to test the file in the FDS program beforehand to ensure correct configurations before proceeding with subsequent FD-Gen steps. Detailed

[1] FD-Gen was tested using FDS version 6.9.1. Its applicability to earlier versions of FDS remains uncertain and may require additional testing or adjustments to ensure compatibility.

instructions on how to download the executables, manuals, source-code and related utilities of FDS can be found on the official website of NIST FDS project at: https://pages.nist.gov/fds-smv/.

Step 2: write FD-Gen input file

This step involves editing the input file for FD-Gen. To account for variations in fire parameters (e.g., fire source, heat release rate, and door open time), FD-Gen employs a four-letter namelist format, similar to FDS and CFAST, to control the sampling and wrapping processes. These namelist lines are integrated with the FDS code lines, specifying the data and commands required for randomization and replacement within the FDS case file. Detailed instructions on using these namelists are provided in Section 3.

Step 3: execute the FD-Gen program

Step 3-1: script checking

After executing the FD-Gen program, an initial sub-step is included to examine the syntax of the FD-Gen code lines. This step automatically analyzes the FD-Gen input file and identifies the IDs of the parameters designated for randomization in the FDS fire case. Additionally, it ensures that these defined parameters have matching IDs within the FDS code lines. This validation guarantees that the sampled parameter values are accurately integrated during the file wrapping process.

Step 3-2: parameter sampling

During this step, the FD-Gen program processes the prepared input file to generate sample values for the randomized fire parameters. Once the sample values are generated, the program outputs a spreadsheet containing the sampled values for each parameter across all cases. Additionally, other information output files are created, providing detailed information about the design of the sample generators and the randomized parameters. These outputs allow users to review and verify that the synthetic data aligns with their intended design before proceeding to the next step.

Step 3-3: input file wrapping

The final step involves integrating the sampled values into the corresponding fire parameters within the provided FDS case input file. This process generates multiple FDS case input files, each containing unique sets of fire parameter values defined by the user-specified randomization.

Step 4: output the results

The FD-Gen program produces various outputs to assist users in assessing and applying the generated results effectively. These include:

- FDS case input files: A collection of FDS case input files is created, each reflecting unique fire parameter values based on the defined randomization.
- Visual representations of sampled data: FD-Gen generates visualizations of the sampled fire parameters, such as fire source locations, HRR curves, and the probability density functions (PDFs) and cumulative distribution functions (CDFs) of the sampled values.

- Metadata for validation and repeatability: The program records essential details, including information for all randomized fire parameters, the seed value used for randomization, and the total number of generated case files. This ensures the reproducibility of the results and facilitates verification of the data generation process.

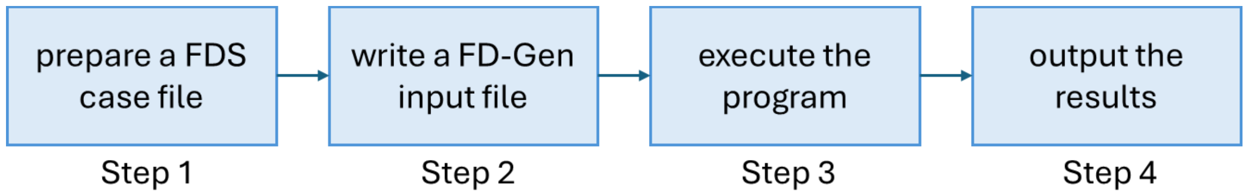


Figure 1. FD-Gen workflow.

3. Defining FD-Gen namelist

FD-Gen employs a four-letter based naming convention to define commends for randomizing and wrapping parameters and values within the FD-Gen input file. This structured approach follows the same rules used in FDS and CFAST, ensuring consistency and enabling an easier transition for existing FDS and CFAST users. This section provides the details regarding the use of these namelists. For convenience, all the namelists and their respective inputs utilized are summarized in **Appendix A**. FD-Gen Input namelist index, serving as a comprehensive reference for commands. Furthermore, test cases for each namelist can be found within the test case folder of the repository: https://github.com/usnistgov/FD-Gen/tree/main/test_cases.

3.1. Basic information

3.1.1. Namelist MAIN

`&MAIN` inputs specify general inputs for a FDS case file creation process including a project name, the total number of cases to be generated, and seeds for the project.

`PROJECT_NAME`: specifies the name of the project for the FDS case file creation process. Using this input, a folder with the same name is generated to store all output files created during the process. Additionally, all FDS case input files will use this project name as a prefix, ensuring consistent organization and easy identification.

`NUMBER_OF_CASES`: specifies the total number of FDS case files to be generated.

`SEEDS`: defines the global seed number that FD-Gen applies for fire parameter sampling. Using the same seed value ensures that FD-Gen produces the same set of cases each time the program is run with the same input file. This allows for repeatability and consistency in FDS case file creation across multiple tries. The default value of `SEEDS` is 1.

3.1.2. Namelist PFSL

`&PFSL` stands for "plotting fire source location". This namelist enables the visualization of sampled fire source locations within the project.

`FSL_ID`: specifies the ID name that is used to represent the randomized fire source location within FD-Gen. This ID acts as the reference to ensure that the randomized fire source locations are identified and plotted correctly.

`SKIP_CASE`: specifies the sampling rate for plotting data. The default value is 1, which means all samples are plotted. For larger datasets, a higher skip rate can be set to reduce visual clutter by plotting a sparser subset of samples, ensuring clearer representation in the figure.

3.1.3. Namelist PHRC

`&PHRC` stands for "plotting heat release rate curves". This namelist enables the plotting of the sampled HRR curves in the project.

HRC_ID: specifies the ID name used to represent the randomized HRR curve within FD-Gen. This ID acts as the reference to ensure that the randomized HRR curve is identified and plotted correctly.

SKIP_CASE: specifies the sampling rate for plotting data. The default value is 1.

3.1.4. Namelist PPSD

&PPSD stands for "plotting parameter sample data". This namelist enables the plotting of the PDFs and CDFs of the sampled data.

PSD_ID: specifies the ID name of sample generators that is developed within the FD-Gen project. This ID acts as the reference to ensure that the samples generated by the sample generator are identified and plotted correctly.

SKIP_CASE: specifies the sampling rate for plotting data. The default value is 1.

3.1.5. Examples

- Script

```
&MAIN PROJECT_NAME='test_script_MAIN', NUMBER_OF_CASES=50, SEEDS=1/
&PFSL FSL_ID='FSL'/
&PHRC HRC_ID='CURVE'/
&PPSD PSD_ID='FS_X','FS_Y','FS_Z'/
```

- Explanation

Under the namelist &MAIN, the basic information for the FD-Gen project is specified. The key parameters include:

PROJECT_NAME: in this example, PROJECT_NAME is set to 'example_commercial'. As a result, all generated FDS case input files will have filenames that use this name as a prefix (e.g.,example_commercial_00000.fds, example_commercial_00001.fds, etc.), as illustrated in **Figure 2**.

NUMBER_OF_CASES and SEEDS: in this example, 10 cases and random seed number of 1 is specified for the project. This information will be record in a text output file, which serves as a record of the project's basic information. An example of the output text file is presented in **Figure 3**.

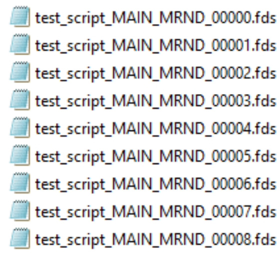


Figure 2. Creation of multiple FDS case input files.

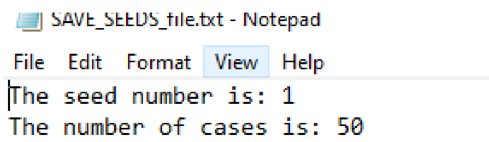


Figure 3. Example of the output text file contains the basic configuration of the project.

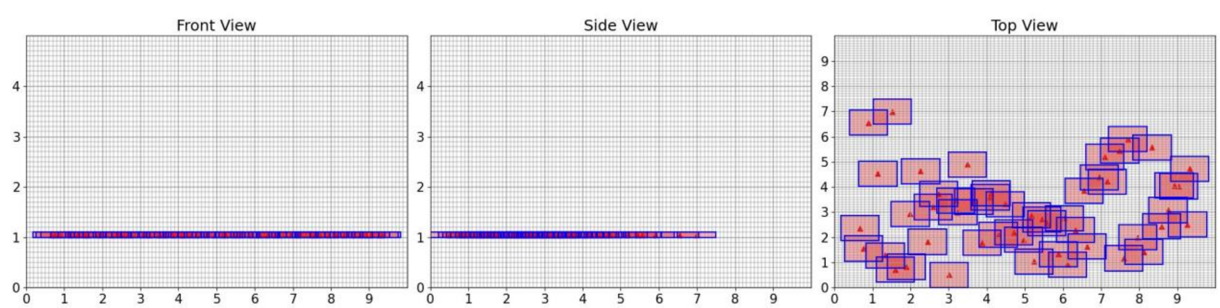
Under the namelists `&PFSL`, `&PHRC`, and `&PPSD`, the plotting of specific randomized fire parameters is specified:

`FSL_ID`: specifies the plotting of the fire source locations. In this example, the project plots the randomized fire source locations under the ID name of `'fire source'`.

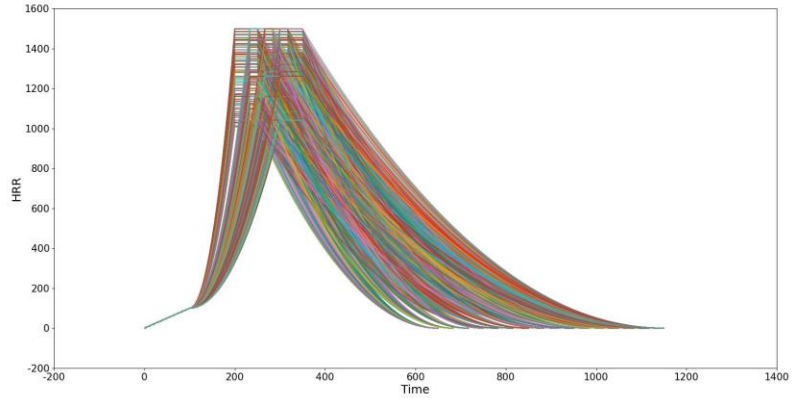
`HRC_ID`: specifies the plotting of the HRR curves. In this example, the project plots the randomized HRR curves under the ID name of `'CURVE'`.

`PSD_ID`: specifies the plotting of the samples from the sample generators designed in the project. In this example, the project plots the samples under the generator ID name of `'FS_X'`, `'FS_Y'`, and `'FS_Z'`.

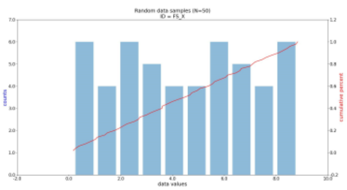
With these commands, FD-Gen identifies the parameters using the specified ID names and generates plots for the corresponding sampled data as defined in the project. The resulting figures are saved as .jpg files and stored in the designated project folder. For this example, the generated plots are shown in **Figure 4**.



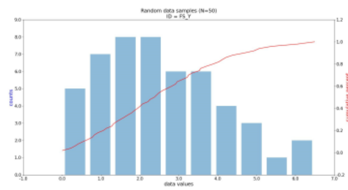
(a) Randomized fire source location - `'fire source'`



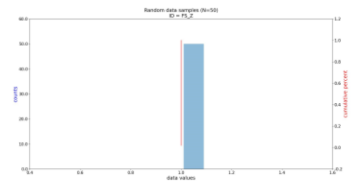
(b) Randomized HRR curves – `'CURVE'`.



(c) Sample data distributions - `'FS_X'`



(d) Sample data distributions - `'FS_Y'`



(e) Sample data distributions - `'FS_Z'`

Figure 4. Example of the project outputs.

3.2. Sample generator

3.2.1. Namelist MRND

The `&MRND` namelist is a key component of FD-Gen, tasked with generating randomized data samples for fire parameters. These samples form the basis for creating new FDS case input files. Within `&MRND`, a random sample generator is configured to tailor the data samples according to specified distributions and value types. The sampling process leverages Latin Hypercube Sampling (LHS) [17] to ensure comprehensive and uniform exploration of the entire parameter range.

`GENERATOR_ID`: a sample generator is identified by a unique name. It is essential to maintain a distinct `GENERATOR_ID` for each sample generator created throughout the FD-Gen input file. This ensures that each generator is properly referenced and avoids conflicts during data sampling and file wrapping process.

`PRE_PARM_INFO_ID`: specifies the ID name of a pre-parameter information input. Pre-parameter information input refers to the data or information that contains relevant details for a specific fire parameter of interest. For example, when designing the sample generator for the parameter for fire source locations, we may apply the pre-parameter information input associated with its ID. This ensures that the sample generator adheres to specific characteristics such as width, length, or other relevant attributes already defined for the fire source location. Such a linkage ensures consistency and precision in parameter sampling.

FYI: a user defined comment that can provide additional information about the input.

DISTRIBUTION_TYPE: identifies the type of distribution used for the parameter data sampling. Currently, allowed distributions are 'CONSTANT', 'LINEAR', 'UNIFORM', 'TRIANGLE', 'NORMAL', 'LOG_NORMAL', 'USER_DEFINE'.

'CONSTANT' option returns sample values with the specified input of CONSTANT. If the additional input of CONSTANT is not provided, the sample generator defaults to the predefined values within the pre-parameter information input. In this case, CONSTANT is set as the average of the boundary defined in the pre-parameter information input.

'LINEAR' option returns sample values ranging from the MINIMUM to MAXIMUM, incremented by 1/NUMBER_OF_CASES. If the additional inputs for MINIMUM and MAXIMUM are not provided, the sample generator defaults to the predefined boundary values specified in the pre-parameter information input. In addition, if the input for NUMBER_OF_CASES is not provided, the sample generator will use the global NUMBER_OF_CASES specified in the &MAIN. Furthermore, 'LINEAR' option also provides an additionally input INCREASE. If INCREASE is applied, the 'LINEAR' option will generate sample values at specified intervals of INCREASE within the defined range.

'UNIFORM' option returns sample values within the range of MINIMUM to MAXIMUM, following a uniform distribution.

'TRIANGLE' option returns sample values within the range of MINIMUM to MAXIMUM, following a triangle distribution. In this distribution, additionally PEAK input must be provided. The PEAK represents the mode, or the value at which the probability density reaches its highest point in the triangle distribution.

'NORMAL' option returns sample values within the range of MINIMUM to MAXIMUM, following a normal distribution. In this case, MEAN and STDEV must be provided. MEAN is the mean value for the normal distribution, and STDEV is the standard deviation value for the distribution.

'LOG_NORMAL' option returns sample values within the range of MINIMUM to MAXIMUM, following a lognormal distribution. In this case, MEAN and STDEV must be provided. MEAN is the mean value for the lognormal distribution, and STDEV is the standard deviation value for the distribution.

'USER_DEFINE' option returns sample values based on a user-provided list of sampled values. When using this mode, SAMPLES can be specified. A list array of sample values needs to be inputted by the user, and the sample generator will directly use these given lists of values for further fire wrapping process.

It is noted that if the additional inputs for MINIMUM and MAXIMUM are not provided for the specified distribution, the sample generator defaults to the predefined boundary values defined in the pre-parameter information input.

RANDOM_SEEDS: identifies the seed number used to generate samples in current sample generator. If RANDOM_SEEDS is not provided, the sample generator will use the global seed number specified in the &MAIN. It is recommended to leave RANDOM_SEEDS vacant in &MRND, as this simplifies repeatability with using only one global seed number to control.

CONSTANT: specifies the constant value in the 'CONSTANT' distribution type option.

MINIMUM: specifies the minimum value range for the sampling.

PEAK: specifies the mode or the value at which the probability density reaches its highest point in the 'TRIANGLE' distribution type option.

MAXIMUM: specifies the maximum value range for the sampling.

MEAN: specifies the mean value in the 'NORMAL' and 'LOG_NORMAL' distribution type option.

STDEV: specifies the standard deviation in the 'NORMAL' and 'LOG_NORMAL' distribution type option.

INCREASE: specifies the increment in the 'LINEAR' distribution type option.

VALUE_TYPE: specifies the value type that outputted by the sample generator. This determines how the sampled values are formatted before goes to next step for wrapping into the FDS case input file. Allowed types are 'INTEGER', 'APPROX', 'REAL', and 'LOGICAL'. Notably, if VALUE_TYPE is not specified, the default value type 'APPROX' will be applied. By default, the APPROX value type maintains a precision of 0.01.

'INTEGER' returns integer value of the sample data.

'APPROX' returns an approximate value of the sample data. Within this value type, additional argument input APPROX_VALUE must be specified.

'REAL' returns real value of the sample data with 10 decimal places, preserving the most original form of the sample data.

'LOGICAL' returns binary value of the sample data, represented as 0 or 1, by applying a threshold to the original sample data. Within this value type, additional argument input LOGICAL_VALUE must be specified.

APPROX_VALUE: specifies the number of digits that should be retained in the final output when the VALUE_TYPE option is set to 'APPROX'. By default, APPROX_VALUE is set to 0.01, meaning that the output values will be rounded to two decimal places unless otherwise specified.

LOGICAL_VALUE: specifies the threshold that is used to conduct the logical data transformation when the VALUE_TYPE option is set to 'LOGICAL'. If a value exceeds this specified threshold, the output will be 1; otherwise, it will be 0.

SHUFFLE_STATES: defines the state of data shuffling in the sample generator. When SHUFFLE_STATES is set to True, the sampled data will be shuffled randomly using the SHUFFLE_RANDOM_SEEDS (described below) specified in the sample generator. By default, SHFFLE_STATES is set to True, meaning that the output sample data is outputted in a random order.

SHUFFLE_RANDOM_SEEDS: identifies the random shuffle seed used for shuffling data in the current sample generator. If SHUFFLE_RANDOM_SEEDS is not provided, the data generator will automatically utilize the shuffling seed generated by the global seed number defined in the &MAIN. It is recommended to leave SHUFFLE_RANDOM_SEEDS blank, as this simplifies repeatability with using only one global seed number.

NUMBER_OF_SAMPLES: specifies the number of sample data to be generated by the defined sample generator. If an input for NUMBER_OF_SAMPLES is not provided, the sample generator will default to using the global NUMBER_OF_CASES specified in the &MAIN. It is recommended to leave NUMBER_OF_SAMPLES blank, as this ensures consistency sample number throughout the overall project (not the case when formulating combinations of parameters using the Cartesian product).

SAMPLES: provides a user-provided list of sample data when 'USER_DEFINE' distribution type option is utilized.

3.2.2. Examples

- Script

```
&MRND GENERATOR_ID='FS_X', PRE_PARM_INFO_ID='fire source_X',
DISTRIBUTION_TYPE='UNIFORM', MINIMUM=0, MAXIMUM=10/

&MRND GENERATOR_ID='FS_Y', PRE_PARM_INFO_ID='fire source_Y',
DISTRIBUTION_TYPE='NORMAL', MINIMUM=0, MAXIMUM=10, MEAN=2, STDEV=2/

&MRND GENERATOR_ID='FS_Z', PRE_PARM_INFO_ID='fire source_Z',
DISTRIBUTION_TYPE='CONSTANT', CONSTANT=1/
```

- Explanation

In this example, a series of sample generators are defined. Under the namelist of &MRND, different data ranges and distribution types are specified in the creation of sample generators. The key parameters include:

GENERATOR_ID: in this example, GENERATOR_ID is set to 'FS_X', 'FS_Y', and 'FS_Z', corresponding to the sampling of randomized fire source locations in the X, Y, and Z directions, respectively.

PRE_PARM_INFO_ID: this input designates the pre-defined information to be inherited by the designed sample generator. The detailed usage of this argument input will be explained in the following sections.

DISTRIBUTION_TYPE: in this example, three different types of distributions are applied, 'UNIFORM', 'NORMAL', and 'CONSTANT'. Within each distribution, CONSTANT, MINIMUM, MAXIMUM, MEAN, and STDEV are specified as required. **Figure 5** illustrates the positions of the randomized fire source locations based on the data generated by these sample generators. **Figure 6** depicts the PDF and CDF of the sample data. It is observed that the data samples follow the distribution types as specified in the &MRND input.

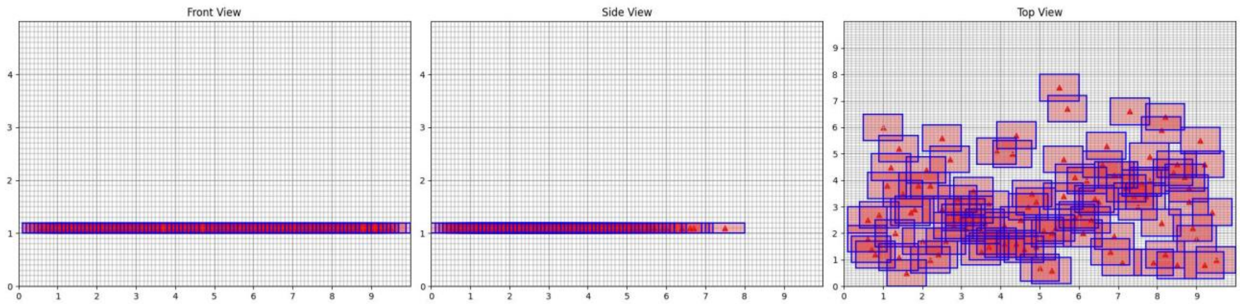


Figure 5. Plot of the positions of sampled fire source locations.

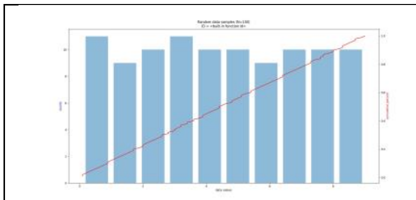
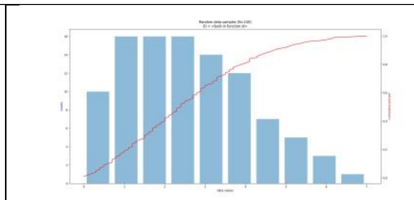
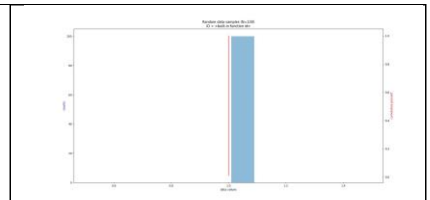
		
(a) Data distributions - 'FS_X'	(b) Data distributions - 'FS_Y'	(c) Data distributions - 'FS_Z'

Figure 6. Plot of the sample data generated by the designed sample generators.

3.3. Fire source location (FSL)

FSL is a namelist in FD-Gen designed to generate sample data and wrap FDS input files, specifically for randomized fire source locations. It exclusively works with the &OBST code lines in the FDS file to identify the fire source and perform the replacement. The FSL functionality is organized into two key namelists: &IFSL and &GFSL.

3.3.1. Namelist IFSL

&IFSL stands for "inputting fire source location" information. It specifies the essential details required for sampling this parameter, referred to as pre-parameter information. The pre-parameter information for fire source location includes the area coverage of the sampling space, the size of the fire burner, which is considered in the FDS, and the mesh size that is applied in the FDS case input file.

PRE_PARM_INFO_ID: specifies the ID name that is used in the FD-Gen input file to identify the pre-parameter information for the parameter.

XB: specifies the area coverage for randomized fire source locations. It is provided in the format XB=0,10,0,10,0,5, which consists of three pairs of coordinates defining the sampling space dimensions. These coordinates typically correspond to the room's length, width, and height, indicating that the fire source will be randomly positioned within the defined spatial boundaries.

FIRE_BURNER_SIZE: defines the dimensions of the fire burner as specified in the FDS case file. FD-Gen uses these dimensions to adjust the sampling range, ensuring that the fire burner does not collide with walls or exceed the room boundaries. The FIRE_BURNER_SIZE is provided in the format, such as FIRE_BURNER_SIZE=1,1,0.2, where the three values represent the length, width, and height of the fire burner, respectively.

FYI: a user defined comment that can provide additional information about the input.

MESH_SIZE: specifies the size of each cell as defined in the FDS case file. Based on this mesh size, FD-Gen can make any generated fire source location aligns with the mesh grid. The MESH_SIZE is provided in the format MESH_SIZE=0.1,0.1,0.1, where each value represents the mesh resolution along the x, y, and z directions. If MESH_SIZE is not specified, users can leave it blank. In such cases, the generated fire source locations will be distributed any places within the specified boundary range.

It is important to note that the current version does not include a feature to prevent collisions with obstructions within the sampling space. To ensure that the fire source or burner does not overlap with any obstruction in complex scenarios, it is recommended to designate an empty sampling space within the room for more appropriate placement.

3.3.2. Namelist GFSL

&GFSL stands for "generating fire source location" data. It specifies the essential details required for generating the sample data for this parameter. The inputs in this namelist include ID, PRE_PARM_INFO_ID, GENERATOR_ID, PRODUCT, and FYI.

ID: identifies the name of the fire parameter to be randomized, as defined in the FDS case file. This ID ensures accurate targeting of the specified fire burner obstruction and maintains consistent referencing throughout the sample data generation and file wrapping processes.

PRE_PARM_INFO_ID: serves as an identifier for the associated pre-parameter information as defined in the &IFSL. Using this ID, the program will retrieve and incorporate the relevant pre-parameter data (e.g., XB, FIRE_BURNER_SIZE, MESH_SIZE) in the processes of sample data generation and file wrapping.

GENERATOR_ID: identifies the sample generators used for parameter data sampling. For parameters such as the fire source location, which involves sampling in three dimensions (X, Y, and Z), three distinct GENERATOR_ID names are required—one for each axis. The syntax for specifying the corresponding generator IDs can be written as: GENERATOR_ID='GENERATOR_1', 'GENERATOR_2', 'GENERATOR_3'.

PRODUCT: provides an option to generate scenarios by combining all sampled fire parameters using the Cartesian product. The default option for this argument is False, meaning that the formation of fire scenarios will continue to use a standard projection rather than incorporating all possible parameter combinations for this item.

FYI: a user defined comment that can provide additional information about the fire source location data generation process.

3.3.3. Examples

- Script

```
============================================================
============ FIRE SOURCE LOCATION ==========================
# FD-Gen code
&IFSL PRE_PARM_INFO_ID='PRE_FSL', XB=0,8.7,22.1,26.3,0,4,
FIRE_BURNER_SIZE=2,1,0.2/
&MRND GENERATOR_ID='FS_X', PRE_PARM_INFO_ID='PRE_FSL_X',
DISTRIBUTION_TYPE='UNIFORM'/
&MRND GENERATOR_ID='FS_Y', PRE_PARM_INFO_ID='PRE_FSL_Y',
DISTRIBUTION_TYPE='NORMAL', MEAN=24, STDEV=2/
&MRND GENERATOR_ID='FS_Z', PRE_PARM_INFO_ID='PRE_FSL_Z',
DISTRIBUTION_TYPE='CONSTANT', CONSTANT=0/
&GFSL ID='fire source', PRE_PARM_INFO_ID='PRE_FSL',
GENERATOR_ID='FS_X','FS_Y','FS_Z'/
============================================================
# FDS code
&OBST ID='fire source', XB=0.7,3.7,22.1,26.3,0.0,3.4,
SURF_IDS='CURVE','INERT','INERT'/
============================================================
```

- Explanation

In this example, the &OBST ID='fire source' in FDS case file is designed to sample, allowing the generation of multiple randomized fire source locations. The key FD-Gen namelists and commands that control this process include &IFSL, &MRND, and &GFSL.

&IFSL

PRE_PARM_INFO_ID: in this example, PRE_PARM_INFO_ID is set to 'PRE_FSL', which designates an identifier for retrieving and referencing the pre-parameter information input related to FSL.

XB: this input defines the sample search space within the room. In this example, the fire source location will be randomly placed within a room, with the following given coordinate ranges: X=[0, 8.7], Y=[22.1, 26.3], Z=[0, 4].

FIRE_BURNER_SIZE: this input specifies the fire burner size as defined in the FDS case input file. In this example, the fire burner dimensions are 2 m long, 1 m wide, and 0.2 m high. This size of the burner will be randomly located in the given range of the search space.

&MRND

GENERATOR_ID: in this example, due to the three axes (X, Y, and Z) of the sample space, three different sample generators are created. GENERATOR_ID of these sample generators are 'FS_X', 'FS_Y', and 'FS_Z'.

PRE_PARM_INFO_ID: under the &MRND namelist, the sample generator can inherit the pre-parameter information from &IFSL. Since there is only one PRE_PARM_INFO_ID specified, but the information needs to be applied to three individual sample generators (for the X, Y, and Z axes), separate identifiers are used for each axis. Specifically, for &IFSL, the following naming formats are applied to represent the inheritance of information for each axis: PRE_PARM_INFO_ID+'_X', PRE_PARM_INFO_ID+'_Y', and PRE_PARM_INFO_ID+'_Z'. As a result, in this example, PRE_PARM_INFO_ID for each individual sample generators are represented in the syntax of 'PRE_FSL_X', 'PRE_FSL_X', and 'PRE_FSL_Z'.

DISTRIBUTION_TYPE: in this setup, three distinct sampling distributions are applied to generate data for the sample generator used to define fire source location parameters. A uniform distribution samples the x-coordinate of the fire source location across the room's length, ranging from 0 m to 8.7 m. A normal distribution with a mean of 24 and a standard deviation of 2 is used to sample the y-coordinate of the fire source location across the room's width, ranging from 22.1 m to 26.3 m. A constant value of 0 is assigned to the z-coordinate, ensuring the burner remains at ground level for accurate placement.

&GFSL

ID: in this example, the ID of the fire parameter is labeled as 'fire source', aligning with the unique ID name defined in FDS.

PRE_PARM_INFO_ID: this argument inherits pre-parameter information from the &IFSL inputs. Therefore, it maintains consistency with the PRE_PARM_INFO_ID specified in &IFSL, using 'PRE_FSL'.

GENERATOR_ID: this argument specifies the use of the defined sample generators, which is align with the GENERATOR_ID specified in &MRND. It includes three sample generators listed as GENERATOR_ID='FS_X', 'FS_Y', 'FS_Z', each corresponding to different spatial parameters for sampling the fire source coordinates.

With the example script executed in FD-Gen, the sampled parameter data will be incorporated into the new FDS case files. The corresponding line of the FDS code for the fire source location will be updated with the new sampled data, as illustrated below:

Before:

```
&OBST ID='fire source', XB=0.7,3.7,22.1,26.3,0.0,0.2,
SURF_IDS='CURVE','INERT','INERT'/
```

After (example of a single FDS input file):

```
&OBST ID='fire source', XB=3.8, 5.8, 24.6, 25.6, 0.0, 0.2,
SURF_IDS='CURVE','INERT','INERT'/
```

It is observed that the fire burner has been relocated to a new position, creating a distinct scenario for a different fire source location. A plot illustrating all the sampled fire source locations for this example, encompassing 10 cases, is presented in **Figure 7.**

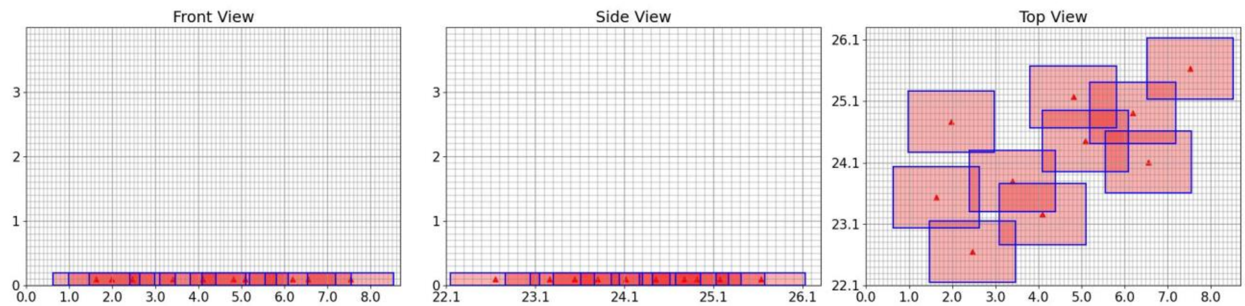


Figure 7. Plot of the sampled fire source locations.

3.4. Vent position (VTP)

`VTP` is a namelist in FD-Gen designed to generate sample data and wrap FDS input files, specifically for reallocating the positions of parameters associated with `&VENT` (such as burner surfaces and exhaust fans) in the FDS code lines. The commands contain two key namelists `&IVTP` and `&GVTP`.

3.4.1. Namelist IVTP

`&IVTP` stands for "inputting vent position" information. It inputs the necessary pre-parameter information for reallocating vent positions during the data sampling and file wrapping process. This pre-parameter information includes the plane on which the vent is positioned, the area coverage for sampling the vent positions, the size of the vent and the mesh size as defined in the FDS case.

`PRE_PARM_INFO_ID:` specifies the ID name that is used in the FD-Gen input file to identify the pre-parameter information input for the parameter.

`PLANE:` specifies the fixed plane of the vent to be positioned, as the vent appears only on a single plane.

`MOVE_RANGE:` identifies the sampling space for vent re-positioning. It is provided in the form `MOVE_RANGE=0,1,-5,5,0,0`, where the three pairs of coordinates represent the dimensions of the sampling space. In this input, at least one pair of coordinates must keep the same value, indicating the plane on which the vent is positioned. This should align with the plane specified in `PLANE` to maintain consistency.

`VENT_SIZE`: specifies the size of the vent as designed in the FDS case input file. This input allows FD-Gen to consider the vent's size during re-positioning, ensuring that vent positions do not overlap with or exceed area boundaries. The `VENT_SIZE` is provided in the format `VENT_SIZE=1,1`, where the quantities represent the length and width of the vent, respectively.

`FYI`: a user defined comment that can provide additional information about the input.

`MESH_SIZE`: identifies the size of each cell as defined in FDS case input file. Based on this mesh size, FD-Gen can make any generated vent position is aligned with the mesh grid. The `MESH_SIZE` is provided in the form `MESH_SIZE=0.1,0.1,0.1`, where each value represents the mesh resolution along the x, y, and z directions. If `MESH_SIZE` is not specified, users can leave it blank. In such cases, the generated vent locations will be distributed any places within the specified boundary range.

It is important to note that the current version does not include a feature to prevent collisions with obstructions within the sampling space. To ensure that the vent does not overlap with any obstruction in complex scenarios, it is recommended to designate an empty sampling space within the plane for more appropriate placement.

3.4.2. Namelist GVTP

`&GVTP` stands for "generating vent position" samples. It specifies the essential details required for generating the sample data for this parameter. The arguments in this namelist include `ID`, `PRE_PARM_INFO_ID`, `GENERATOR_ID`, `PRODUCT`, and `FYI`.

`ID`: specifies the ID name of fire parameter that needs to be randomized as defined in the FDS case file. This ID ensures that the correct vent information is identified and consistently referenced during the sample data generation and file wrapping process.

`PRE_PARM_INFO_ID`: serves as an identifier for the associated pre-parameter information as defined in the `&IVTP`. Using this ID, the program will retrieve and incorporate the relevant pre-parameter data (e.g., `PLANE`, `MOVE_RANGE`, `VENT_SIZE`, `MESH_SIZE`) in the sample data generation and file wrapping process.

`GENERATOR_ID`: identifies the sample generator used for parameter sampling. For parameters such as vent position, which involves sampling in three dimensions (X, Y, and Z), three distinct `GENERATOR_ID` are required—one for each axis. The syntax for specifying the corresponding generator IDs can be written as: `GENERATOR_ID='GENERATOR_1', 'GENERATOR_2', 'GENERATOR_3'.`

`PRODUCT`: provides an option to generate scenarios by combining all sampled fire parameters using the Cartesian product. The default option for this argument is `False`, meaning that the formation of fire scenarios will continue to use a standard projection rather than incorporating all possible parameter combinations for this item.

FYI: a user defined comment that can provide additional information about the VTP data generation process.

3.4.3. Examples

- Script

```
===========================================================
=========== VENT POSITION =================================
# FD-Gen code
&IVTP PRE_PARM_INFO_ID='VENT POSITION', PLANE='Z', MOVE_RANGE=0,1,
-25,25,0,0, VENT_SIZE=0.5,20/
&MRND GENERATOR_ID='VENT_X', PRE_PARM_INFO_ID='VENT POSITION_X',
DISTRIBUTION_TYPE='UNIFORM'/
&MRND GENERATOR_ID='VENT_Y', PRE_PARM_INFO_ID='VENT POSITION_Y',
DISTRIBUTION_TYPE='UNIFORM'/
&MRND GENERATOR_ID='VENT_Z', PRE_PARM_INFO_ID='VENT POSITION_Z',
DISTRIBUTION_TYPE='CONSTANT'/
&GVTP ID='VENT_FIRE',  PRE_PARM_INFO_ID='VENT POSITION',
GENERATOR_ID='VENT_X','VENT_Y','VENT_Z'/
===========================================================
# FDS code
&VENT ID='VENT_FIRE', XB=0.0,1.0,-10,10,0,0, SURF_ID='LINEFIRE'/
===========================================================
```

- Explanation

This example illustrates the sampling and wrapping of parameters associated with VTP in the FDS namelist &VENT. When defining randomized fire parameters, such as fire source locations or vents, within the &VENT FDS namelist, the corresponding VTP namelists must be utilized. Key commands governing this sampling process include &IVTP, &MRND, and &GVTP.

&IVTP

PRE_PARM_INFO_ID: in this example, PRE_PARM_INFO_ID is set to 'VENT POSITION', which designates an identifier for retrieving and referencing the pre-parameter information input related to VTP.

PLANE: this input defines the plane where the sampling takes place. In the given example, the fire vent source is positioned on the ground, making the input of argument and its value written as PLANE='Z'.

MOVE_RANGE: this argument defines the search range within the specified plane. In this example, the vent window is constrained to a rectangular area on the ground with dimensions of 1 m in length and 50 m in width. Additionally, since the fire vent source is positioned at ground level, the last pair of values in the range is set to 0,0 to indicate no vertical displacement. The complete MOVE_RANGE is represented as MOVE_RANGE=0,1, -25,25,0,0, specifying the x, y, and z bounds for sampling within the area range.

VENT_SIZE: this input specifies the size of the vent window as defined in the FDS case input file. In this example, the vent window is set at 20 m in width and 0.5 m in length.

&MRND

GENERATOR_ID: in this example, due to the three axes (X, Y, and Z) of the sample space, three different sample generators are created. GENERATOR_ID of these sample generators is 'VENT_X', 'VENT_Y', and 'VENT_Z'.

PRE_PARM_INFO_ID: within the &MRND namelist, the sample generator can inherit the pre-parameter information from &IVTP. Since there is only one PRE_PARM_INFO_ID, but the information needs to be applied to three individual sample generators (for the X, Y, and Z axes), separate identifiers are used for each axis. Specifically, for &IVTP, the following naming formats are applied to represent the inheritance of information for each axis: PRE_PARM_INFO_ID+'_X', PRE_PARM_INFO_ID+'_Y', and PRE_PARM_INFO_ID+'_Z'. Therefore, in this example, PRE_PARM_INFO_ID for each individual sample generators are represented in the syntax of 'VENT POSITION_X', 'VENT POSITION_Y', and 'VENT POSITION_Z'.

DISTRIBUTION_TYPE: In this setup, both uniform and constant distributions are applied. The uniform distribution governs the sample generator for 'VENT_X' and 'VENT_Y', allowing for randomized sampling across a defined range. Additionally, a constant value is assigned to the sample generator for 'VENT_Z', ensuring a fixed value is maintained.

&GVTP

ID: in this example, the ID of the vent parameter is labeled as 'VENT_FIRE', aligning with the unique ID name defined in FDS.

PRE_PARM_INFO_ID: this input inherits pre-parameter information from the &IVTP inputs. Therefore, it maintains consistency with the PRE_PARM_INFO_ID specified in &IVTP, using 'VENT POSITION'.

GENERATOR_ID: this input specifies the use of the defined sample generators, which are align with the GENERATOR_ID specified in &MRND. It includes three sample generators listed as GENERATOR_ID='VENT_X','VENT_Y','VENT_Z', each corresponding to different spatial parameters for sampling the vent position coordinates.

With the example script executed in FD-Gen, the sampled parameter data will be incorporated into the new FDS case input files. The corresponding line of the FDS code for the vent position will be updated with the new sampled data, as illustrated below:

Before:

```
&VENT ID='VENT_FIRE', XB=0.0,1.0,-10,10,0,0, SURF_ID='LINEFIRE'/
```

After (example of a single FDS input file):

```
&VENT ID='VENT_FIRE', XB=0.23, 0.73, -19.95, 0.05, 0.0, 0.0,
SURF_ID='LINEFIRE'/
```

From the example, it can be observed that the vent position has been relocated to a new position, thereby creating a distinct scenario with a unique vent location.

3.5. Door or window open time (DWT)

DWT is a namelist in FD-Gen designed to generate sample data and wrap FDS input files, specifically for reconfiguring door and window open times associated with &OBST and &HOLE in the FDS code lines. The commands contain two key namelists &IDWT and &GDWT.

3.5.1. Namelist IDWT

&IDWT refers to "Inputting door window open time" information, which provides the necessary pre-parameter information for generating samples on the time during which portions of doors or windows are controlled. Below are some key arguments typically included.

PRE_PARM_INFO_ID: specifies the ID name that is used in the FD-Gen input file to identify the pre-parameter information for the parameter.

OBST_ID: specifies the ID name as designed in the FDS case file to identify the door or window items that need to be controlled. Typically, doors are referenced under the FDS namelist &OBST, and windows are referenced under the FDS namelist &HOLE.

XB: identifies the geometry for the door or window items. XB is given in the form XB=9.4,9.6,23,24,0.0,2.0. This notation consists of three pairs of coordinates representing the three dimensions of the geometry for door or window items.

OPEN_DIRECTION: this argument input specifies the opening direction of a door or window considered in the FDS case scenario. Using this information, the original door or window can be divided into sections and subsequently removed in specific directions. Accepted opening directions include 'X+', 'X-', 'Y+', 'Y-', 'Z+', and 'Z-', consistent with the directions in FDS.

QUANTITY_NAME: specifies the QUANTITY parameter as defined in the FDS case input file, which determines the control mechanism for door or window openings based on specific quantities. Common inputs for QUANTITY in FDS code include 'TIME', 'TEMPERATURE', etc.

PORTION: specifies the portions into which the window or door is divided for controlled opening. For example, if a window opens in two stages—first halfway and then fully—the PORTION parameter should be defined as PORTION=0.5,1. This denotes that the window initially opens to 50% of its total capacity before transitioning to 100 %. For a three-stage opening sequence, the PORTION parameter can be defined as PORTION=0.33,0.66,1.

DEVC_ID: similar to OBST_ID, the DEVC_ID specifies the identifier as defined in the FDS input file under the namelist &DEVC. It serves as a link to locate the &DEVC input that holds specific open time data for the door or window items.

DOOR_OR_WINDOW_OPEN_TIME_RANGE: this argument input specifies the sampling space for the time variable in DWT. Given the defined portions for staged openings, the DOOR_OR_WINDOW_OPEN_TIME_RANGE parameter is provided in pairs. Each pair represents the minimum and maximum time range for a particular stage of opening. For instance, if a window opens in two stages, the input might be defined as DOOR_OR_WINDOW_OPEN_TIME_RANGE=100,151,200,250. This means that the first stage of the window's opening will occur randomly within 100 s to 151 s, and the second stage will follow, opening randomly between 200 s and 250 s.

FYI: a user defined comment that can provide additional information about the input.

3.5.2. Namelist GDWT

&GDWT stands for "generating door or window open time" samples. It specifies the essential details required for generating the sample data for this parameter. The argument inputs in this namelist include ID, PRE_PARM_INFO_ID, GENERATOR_ID, PRODUCT, and FYI.

ID: specifies the ID name of fire parameter that needs to be randomized as defined in the FDS case file.

PRE_PARM_INFO_ID: serves as an identifier for the associated pre-parameter information as defined in the &IDWT.

GENERATOR_ID: identifies the sample generators used for parameter sampling. Based on the portion information inputted in &IGWT, the number of required generators should match the number of stages for the opening. For example, if the door or window opens in two stages, the sample generators must be defined accordingly, denoted as GENERATOR_ID='dot_timer1', 'dot_timer2'. Two sample generators are included and defined.

PRODUCT: provides an option to generate scenarios by combining all sampled fire parameters using the Cartesian product. The default option for this argument is False, meaning that the formation of fire scenarios will continue to use a standard projection rather than incorporating all possible parameter combinations for this item.

FYI: a user defined comment that can provide additional information about the DWT data generation process.

3.5.3. Examples

- Script

```
============================================================
======= Door open time =====================================
# FD-Gen code
&IDWT PRE_PARM_INFO_ID='door open time',
      OBST_ID='door24_1','door24_2',
      XB=9.4,9.6,23.15,24.3,0.0,2.0, OPEN_DIRECTION='Y-'
      QUANTITY_NAME='TIME', PORTION=0.5,1
      DEVC_ID='dot_timer1','dot_timer2'
      DOOR_OR_WINDOW_OPEN_TIME_RANGE=100,151,200,250 /
&MRND GENERATOR_ID='dot_timer1', PRE_PARM_INFO_ID='door open time_1',
DISTRIBUTION_TYPE='UNIFORM'/
&MRND GENERATOR_ID='dot_timer2', PRE_PARM_INFO_ID='door open time_2',
DISTRIBUTION_TYPE='LINEAR'/
&GDWT ID='door24',  PRE_PARM_INFO_ID='door open time',
GENERATOR_ID='dot_timer1','dot_timer2'/
===========================================================
# FDS code
&DEVC ID='dot_timer1', QUANTITY='TIME', XYZ=0.0,26.3,0.0,
SETPOINT=500, INITIAL_STATE=.TRUE./
&DEVC ID='dot_timer2', QUANTITY='TIME', XYZ=0.0,26.3,0.0,
SETPOINT=500, INITIAL_STATE=.TRUE./
&OBST ID='door24_1', XB=9.4,9.6,23.15,24.3,0.0,2.0, RGB=153,102,0,
SURF_ID='INERT', DEVC_ID='dot_timer1'/
&OBST ID='door24_2', XB=9.4,9.6,23.15,24.3,0.0,2.0, RGB=153,102,0,
SURF_ID='INERT', DEVC_ID='dot_timer2'/
===========================================================
```

- Explanation

In this example, sampling and wrapping the parameters associated with DWT is detailed. A two-stage opening of the door is presented in this example.

&IDWT

PRE_PARM_INFO_ID: in this example, PRE_PARM_INFO_ID is set to 'door open time', which designates an identifier for retrieving and referencing the pre-parameter information related to DWT.

OBST_ID: this input identifies the specific &OBST IDs as defined in the FDS input file. In cases involving multi-stage openings, multiple &OBST IDs may be specified. For example, in this two-stage opening scenario, two &OBST IDs (i.e., 'door24_1' and 'door24_2') in the FDS code lines should be included.

XB: this argument input defines the geometry of the door to be placed. In this example, the parameter XB=9.4,9.6,23.15,24.3,0.0,2.0 specifies the spatial coordinates of the door.

OPEN_DIRECTION: this argument input specifies the direction in which the door will open. In this example, the value indicates that the door opens along the Y-axis in the negative direction. This means the door will be split along the Y-axis, and as a result, the door disappears or is removed from the area from the Y=23.15 to Y=24.30.

QUANTITY_NAME: the quantity name used to control the door opening in this example is 'TIME'. This argument remains consistent with the corresponding input in the FDS case input file.

PORTION: the portion specified in this example is PORTION=0.5,1, which indicates that the door is divided into two parts and will be opened in two stages: initially opening to 50 % and then fully opening to 100 %. In the first stage, one part of the door is removed based on the time value provided by 'dot_timer1'. In the second stage, the remaining part of the door is removed according to the time value specified by 'dot_timer2'. This approach automates the code line generation for a controlled, staged opening of the door within FDS.

DEVC_ID: this argument identifies the specific &DEVC IDs as designed in the FDS case input file. In this example, due to the two-stage opening, two &DEVC IDs are specified, denoted as 'dot_timer1' and 'dot_timer2'.

DOOR_OR_WINDOW_OPEN_TIME_RANGE: in this example, the sampling range for the DWT is specified as DOOR_OR_WINDOW_OPEN_TIME_RANGE=100,151,200,250. This range is defined according to the QUANTITY_NAME parameter, which is set to 'TIME', indicating that the data being sampled belongs to time variable. Given that the door is designed to open in two stages, the range defines that the first stage of the door opening will occur within the time frame of 100 s to 151 s, while the second stage will occur between 200 s to 250 s. This argument input ensures that the door opens in a controlled, staged manner with precise timing intervals for each stage.

&MRND

GENERATOR_ID: in this example, due to the two stages of door opening, two different sample generators are created. GENERATOR_ID of these two sample generators includes 'dot_timer1' and 'dot_timer2'.

PRE_PARM_INFO_ID: within the &MRND namelist, the sample generator can inherit the pre-parameter information from &IDWT. Since there is only one PRE_PARM_INFO_ID, but the information needs to be applied to two individual sample generators (for the two stages of door opening), separate identifiers are used for each sample generator. Specifically, the following naming formats are applied to represent the inheritance of information for each opening stage: PRE_PARM_INFO_ID+'_1', PRE_PARM_INFO_ID+'_2', etc. Therefore, for this example, 'door open time_1' and 'door open time_2' are applied.

3. DISTRIBUTION_TYPE: in this example, uniform and linear distributions are respectively applied to sample data from the specified ranges.

&GFSL

ID: in this example, the ID of the door open time sampling task is labeled as 'door24'.

PRE_PARM_INFO_ID: this argument input inherits pre-parameter information from the &IDWT inputs. Therefore, it maintains consistency with the PRE_PARM_INFO_ID specified in &IDWT, using 'door open time'.

GENERATOR_ID: this argument specifies the use of the defined sample generators, aligned with the GENERATOR_ID specified in &MRND. It includes two sample generators, listed as GENERATOR_ID='dot_timer1','dot_timer2', each corresponding to different sample data variations for the door opening time.

With the example script executed in FD-Gen, the sampled parameter data will be incorporated into the FDS input file. The corresponding line of the FDS script for the door open time will be updated with the new sampled data, as illustrated below:

Before:

```
&DEVC ID='dot_timer1', QUANTITY='TIME', XYZ=0.0,26.3,0.0,
SETPOINT=500, INITIAL_STATE=.TRUE./

&DEVC ID='dot_timer2', QUANTITY='TIME', XYZ=0.0,26.3,0.0,
SETPOINT=500, INITIAL_STATE=.TRUE./

&OBST ID='door24_1', XB=9.4,9.6,23.15,24.3,0.0,2.0, RGB=153,102,0,
SURF_ID='INERT', DEVC_ID='dot_timer1'/

&OBST ID='door24_2', XB=9.4,9.6,23.15,24.3,0.0,2.0, RGB=153,102,0,
SURF_ID='INERT', DEVC_ID='dot_timer2'/
```

After (example of a single FDS input file):

```
&DEVC ID='dot_timer1', QUANTITY='TIME', XYZ=0.0,26.3,0.0,
SETPOINT=129.0, INITIAL_STATE=.TRUE./

&DEVC ID='dot_timer2', QUANTITY='TIME', XYZ=0.0,26.3,0.0,
SETPOINT=240.0, INITIAL_STATE=.TRUE./

&OBST ID='door24_1', XB=9.4, 9.6, 23.725, 24.3, 0.0, 2.0,
RGB=153,102,0, SURF_ID='INERT', DEVC_ID='dot_timer1', /
```

```
&OBST ID='door24_2', XB=9.4, 9.6, 23.15, 23.725, 0.0, 2.0,
RGB=153,102,0, SURF_ID='INERT', DEVC_ID='dot_timer2', /
```

It is observed that the door opening times for the two `&DEVC` ID entries have been altered. Additionally, these modified times correspond to the newly created separate door items.

3.6. Random obstruction (RXB)

`RXB` is a namelist in FD-Gen designed to generate sample data and wrap FDS input files, specifically for creating random or variable sized items such as obstructions, holes, and vents. The functionality is linked to the relative namelists in FDS, such as `&OBST`, `&VENT`, `&HOLE`. This command contains two key namelists `&IRXB` and `&GRXB`.

3.6.1. Namelist IRXB

`&IRXB` refers to "inputting random obstruction" information. It specifies key details necessary for generating random variable sized items during the data sampling and file wrapping process.

`PRE_PARM_INFO_ID:` specifies the ID name that is used in the FD-Gen input file to identify the pre-parameter information input for the parameter.

`NAMELIST:` specifies the corresponding namelist used for this parameter in the FDS input file, where `&OBST` is applied for obstructions, `&VENT` for vent items, and `&HOLE` for hole items.

`INITIAL_POINT:` the coordinates of the fixed point for this variable-sized item, with the designated fixed location corresponding to its X-min, Y-min and, Z-min corner. The format includes three values representing the X, Y, and Z coordinates of the point.

`LENGTH_RANGE`: defines the sampling space for the length of the item. The notation is expressed as a pair of values representing the minimum and maximum bounds of the range.

`WIDTH_RANGE:` defines the sampling space for the width of the item.

`HEIGHT_RANGE:` defines the sampling space for the height of the item.

`FYI:` A user defined comment that can provide additional information about the input.

3.6.2. Namelist GRXB

`&GRXB` refers to "generating random obstruction" samples. It specifies the essential details required for generating the sample data for this parameter. The arguments in this namelist include `ID`, `PRE_PARM_INFO_ID`, `GENERATOR_ID`, `PRODUCT`, and `FYI`.

`ID:` specifies the ID name of fire parameter that needs to be randomized in the FDS case file. This ID ensures that the correct variable sized item is identified and consistently referenced during the sample data generation and file wrapping process.

PRE_PARM_INFO_ID: serves as an identifier for the associated pre-parameter information as defined in the &IRXB. Using this ID, the program retrieves and incorporates the relevant pre-parameter information in the sample data generation and file wrapping process.

GENERATOR_ID: in the context of RXB, where length, width, and height need to be sampled individually, three distinct GENERATOR_ID are required. Each GENERATOR_ID corresponds to a separate generator handling the sampling for one of the three dimensions. The syntax for specifying the corresponding generator IDs can be written as: GENERATOR_ID='GENERATOR_1', 'GENERATOR_2', 'GENERATOR_3'.

PRODUCT: provides an option to generate scenarios by combining all sampled fire parameters using the Cartesian product. The default option for this argument is False, meaning that the formation of fire scenarios will continue to use a standard projection rather than incorporating all possible parameter combinations for this item.

FYI: a user defined comment that can provide additional information about the RXB data generation process.

3.6.3. Examples

- Script

```
============================================================
======= random obstruction ================================
# FD-Gen code
&IRXB PRE_PARM_INFO_ID='obstruction',
      NAMELIST='OBST'
      INITIAL_POINT=0,0,0
      LENGTH_RANGE=4,8, WIDTH_RANGE=1,3, HEIGHT_RANGE=2,5/
&MRND GENERATOR_ID='OBS_x', PRE_PARM_INFO_ID='obstruction_L',
DISTRIBUTION_TYPE='UNIFORM', VALUE_TYPE='APPROX', APPROX_VALUE=0.01/
&MRND GENERATOR_ID='OBS_y', PRE_PARM_INFO_ID='obstruction_W',
DISTRIBUTION_TYPE='UNIFORM', VALUE_TYPE='APPROX', APPROX_VALUE=0.01/
&MRND GENERATOR_ID='OBS_z', PRE_PARM_INFO_ID='obstruction_H',
DISTRIBUTION_TYPE='UNIFORM', VALUE_TYPE='APPROX', APPROX_VALUE=0.01/
&GRXB ID='obstruction', PRE_PARM_INFO_ID='obstruction',
GENERATOR_ID='OBS_x','OBS_y','OBS_z'/
===========================================================
# FDS code
&OBST ID='obstruction', XB=0.0,4.0,0.0,2.0,0.0,2.0,
COLOR='GRAY 94', SURF_ID='INERT'/
===========================================================
```

o Explanation

This example details the process of sampling the parameters within RXB. A random variable sized obstruction, identified by the ID name 'obstruction', is specified in the FDS input file.

&IRXB

PRE_PARM_INFO_ID: in this example, PRE_PARM_INFO_ID is set to 'obstruction', which designates an identifier for retrieving and referencing the pre-parameter information related to RXB.

NAMELIST: this argument corresponds to the namelist used for the parameter in the FDS case input file. In this example, where the item is a variable sized obstruction, the argument is assigned the value 'OBST', aligning with the namelist used in FDS.

INITIAL_POINT: the fixed point for the obstruction is specified using the input INITIAL_POINT=0,0,0. This indicates that the obstruction is positioned with its base at the point (0,0,0). The length, width, and height of the obstruction are then added relative to this point.

LENGTH_RANGE: specifies the sampling range for the obstruction's length, set between 4 m and 8 m.

WIDTH_RANGE: specifies the range for the obstruction's width, set between 1 m and 3 m.

HEIGHT_RANGE: Indicates the range for the obstruction's height, set between 2 m and 5 m.

&MRND

GENERATOR_ID: In this example, due to the three dimensions (length, width, and height) of the sample space, three different sample generators are created. GENERATOR_ID of the three sample generators is 'OBS_x', 'OBS_y', and 'OBS_z'.

PRE_PARM_INFO_ID: within the &MRND namelist, the sample generator can inherit the pre-parameter information using the argument PRE_PARM_INFO_ID. Since there is only one PRE_PARM_INFO_ID, but the information needs to be applied to three individual sample generators (for the length, width, and height), separate identifiers are used for each dimension. Specifically, for &IRXB, the following are applied to represent the inheritance of information for each dimension: PRE_PARM_INFO_ID+'_L', PRE_PARM_INFO_ID+'_W', and PRE_PARM_INFO_ID+'_H'. Therefore, in this example, PRE_PARM_INFO_ID for each individual sample generators are represented in the syntax of 'obstruction_L', 'obstruction_W', and 'obstruction_H'.

DISTRIBUTION_TYPE: in this setup, three distinct sampling distributions are applied to generate the sample data for a random sized obstruction. The length of the obstruction is sampled using a uniform distribution within a range of 4 m to 8 m. Similarly, the width of the obstruction is determined by a uniform distribution ranging from 1 m to 3 m. For the height, a uniform distribution is also used, sampling values between 2 m and 5 m.

&GFSL

ID: In this example, the ID of the RXB is labeled as ' obstruction', aligning with the unique ID name in FDS case input file.

PRE_PARM_INFO_ID: this argument inherits pre-parameter information from the &IRXB inputs. Therefore, it maintains consistency with the PRE_PARM_INFO_ID specified in &IRXB.

GENERATOR_ID: this argument specifies the use of the defined sample generator, which is align with the GENERATOR_ID specified in &MRND. It includes three sample generators listed as GENERATOR_ID='OBS_x','OBS_y','OBS_z', each corresponding to one dimension of the information required by RXB.

With the example script executed in FD-Gen, the sampled parameter data will be incorporated into the FDS input file. The corresponding line in the FDS script for the obstruction will be updated with the newly sampled data, as illustrated below:

Before:

```
&OBST ID='obstruction', XB=9.4,9.6,23.15,24.3,0.0,2.0,
COLOR='GRAY 94', SURF_ID='INERT'/
```

After (example of a single FDS input file):

```
&OBST ID='obstruction', XB=0, 6.27, 0, 2.61, 0, 4.98, COLOR='GRAY 94',
SURF_ID='INERT'/
```

It is observed that the XB for the &OBST ID 'obstruction' has been altered. The updated size of the obstruction has been created and is reflected in the new FDS code lines.

3.7. Other parameter (OTH)

OTH is a namelist in FD-Gen designed to generate sample data and wrap FDS input files, specifically for fire parameters other than these considered above. This command is provided to accommodate situations that require sampling of additional variables in the FDS input file. This command contains two key namelists &IOTH and &GOTH.

3.7.1. Namelist IOTH

&IOTH refers to "Inputting other" information, which specifies the key details necessary for generating sample data for parameters not mentioned above that need to be randomized in the FDS case input file.

PRE_PARM_INFO_ID: specifies the ID name that is used in the FD-Gen input file to identify the pre-parameter information for the parameter.

`NAMELIST`: specifies the corresponding namelist for this parameter as defined in the FDS case input file.

`ARG_NAME`: specifies the name argument input within the namelist that the parameter needs to be randomized.

`OTHERS_RANGE`: defines the sampling space for the parameter. The notation is expressed as a pair of values representing the minimum and maximum bounds of the range.

`FYI`: a user defined comment that can provide additional information about the input.

3.7.2. Namelist GOTH

`&GOTH` refers to generate samples for `OTH`. It specifies the essential details required for generating the sample data for this parameter. The arguments in this namelist include `ID`, `PRE_PARM_INFO_ID`, `GENERATOR_ID`, `PRODUCT`, and `FYI`.

`ID`: specifies the ID name of fire parameter that needs to be randomized as defined in the FDS case file.

`PRE_PARM_INFO_ID`: serves as an identifier for the associated pre-parameter information as defined in the `&IOTH`. Using this ID, the program will retrieve and incorporate the relevant pre-parameter information in the sample data generation and file wrapping process.

`GENERATOR_ID`: identifies the sample generator used for parameter data sampling. In the `IOTH` context, only one sample generator needs to be defined.

`PRODUCT`: provides an option to generate scenarios by combining all sampled fire parameters using the Cartesian product. The default option for this argument is `False`, meaning that the formation of fire scenarios will continue to use a standard projection rather than incorporating all possible parameter combinations for this item.

`FYI`: a user defined comment that can provide additional information about the OTH data generation process.

3.7.3. Examples

- Script

```
============================================================
======= other information ==================================
# FD-Gen code
&IOTH PRE_PARM_INFO_ID='setpoint_value', NAMELIST='DEVC',
ARG_NAME='SETPOINT', OTHERS_RANGE=300, 350/
&MRND GENERATOR_ID='generator1', PRE_PARM_INFO_ID='setpoint_value',
DISTRIBUTION_TYPE='UNIFORM', VALUE_TYPE='APPROX', APPROX_VALUE=1/
&GOTH ID='FRONT DOOR TIMER', PRE_PARM_INFO_ID='setpoint_value',
GENERATOR_ID='generator1'/
```

```
============================================================
# FDS code
&DEVC ID='FRONT DOOR TIMER', QUANTITY='TIME', XYZ=-0.1,-0.4,0.1,
SETPOINT= 300.0, INITIAL_STATE=.TRUE. /
============================================================
```

- Explanation

This example details the process of sampling the parameters within `OTH`. An activated time value for a timer is randomized in this example.

`&IOTH`

`PRE_PARM_INFO_ID`: in this example, `PRE_PARM_INFO_ID` is set to `'setpoint_value'`, which designates an identifier for retrieving and referencing the pre-parameter information related to `OTH`.

`NAMELIST`: corresponds to the namelist of the parameter as defined in the FDS case input file. In this case, where the item relates to timer control, the argument is set to `'DEVC'`, aligning with the namelist used in FDS.

`ARG_NAME`: specifies the variable that needs to be randomized in the relevant line of FDS code. In this example, `ARG_NAME` is set to `'SETPOINT'`, indicating the specific parameter to be sampled.

`OTHERS_RANGE`: defines the range of setpoint values to be sampled, which, in this example, spans from 300 s to 350 s.

`&MRND`

`GENERATOR_ID`: in this example, `GENERATOR_ID` is assigned with value `'generator1'` to specify the generator used for parameter data sampling .

`PRE_PARM_INFO_ID`: the generator inherits the pre-parameter information from `'setpoint_value'`, as specified in `&IOTH`.

`DISTRIBUTION_TYPE`: a uniform distribution is applied to sample data within the range of 300 s to 350 s, representing the randomized setpoint of the device in this example.

`&GFSL`

`ID`: in this example, the ID is labeled as `'FRONT DOOR TIMER'`, aligning with the unique ID name in FDS case input file.

`PRE_PARM_INFO_ID`: this argument inherits pre-parameter information from the `&IOTH` inputs. Therefore, it maintains consistency with the `PRE_PARM_INFO_ID` specified in `&IOTH`.

GENERATOR_ID: this argument specifies the use of the defined sample generator, which is align with the GENERATOR_ID specified in &MRND. In this case, GENERATOR_ID is set to 'generator1'.

With the example executed in FD-Gen, the sampled parameter data will be incorporated into the FDS input file. The corresponding line in the FDS script for the OTH will be updated with the newly sampled data, as illustrated below:

Before:

```
&DEVC ID='FRONT DOOR TIMER', QUANTITY='TIME', XYZ=-0.1,-0.4,0.1,
SETPOINT= 300.0, INITIAL_STATE=.TRUE. /
```

After (example of a single FDS input file):

```
&DEVC ID='FRONT DOOR TIMER', QUANTITY='TIME', XYZ=-0.1,-0.4,0.1,
SETPOINT=341.0, INITIAL_STATE=.TRUE./
```

It is observed that the SETPOINT for the DEVC ID 'FRONT DOOR TIMER' has been altered. The new sampled value has been created and is reflected in the new FDS code lines.

3.8. HRR curve (HRC)

OTH is the namelist of FD-Gen responsible for generating sample data and integrating them into FDS input files specifically for randomized HRR curves. It comprises five essential namelists. &IFTD and &GFTD is responsible for generating sample data for individual fire time duration (FTD). &IMHR and &GMHR generates sample data for the maximum heat release rate (MHR) for individual fire scenarios, enabling variability in fire intensity. &GHRC combines the sample data from fire time durations and maximum heat release rates to generate the randomized HRR curves. Finally, these are seamlessly incorporated into FDS input files.

3.8.1. Namelist IFTD

&IFTD stands for "inputting fire time duration" information. It provides the necessary pre-parameter details required for generating sample data for FTD. The HRR curve is customized by stages, hence this pre-parameter information mainly specifies the sampling range for the time duration at various stages of fire development. Figure **8** illustrates the arguments used in designing the time duration across various stages of fire development.

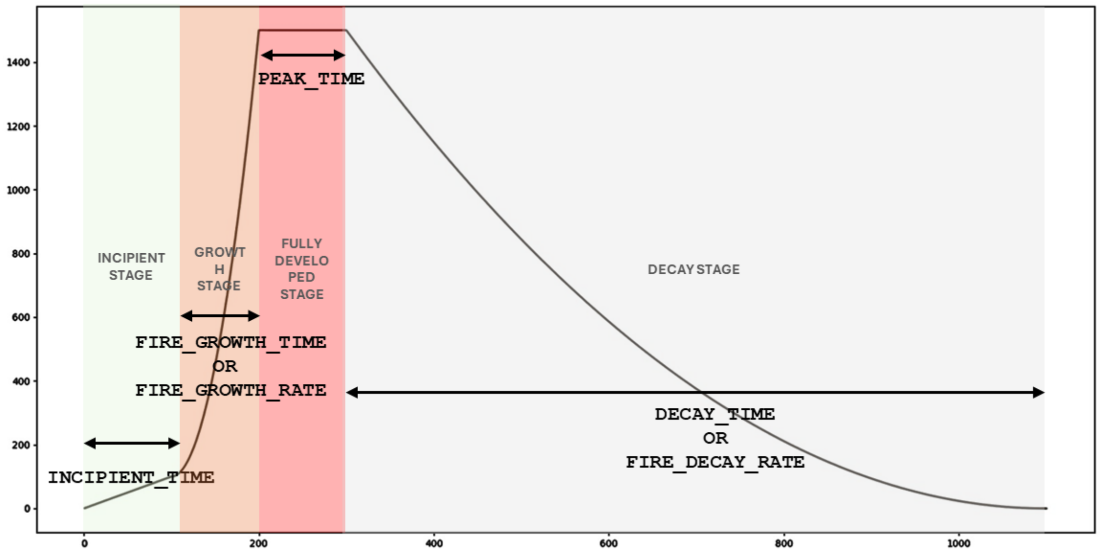


Figure 8. schematic diagram of the arguments used in the namelist &IFTD. A typical fire HRR curves divided into stages is illustrated.

PRE_PARM_INFO_ID: specifies the ID name that is used in the FD-Gen input file to identify the pre-parameter information input for the parameter.

INCIPIENT_TIME: specifies the sampling range for the incipient time variable within the FTD. It defines the duration of the incipient stage, indicating how long this phase of fire development lasts.

FIRE_GROWTH_TIME: specifies the sampling range for the fire growth time variable within the FTD. It defines the duration of the fire growth stage, indicating how long this phase of fire development lasts.

FIRE_GROWTH_RATE: specifies the sampling range for the fire growth rate variable within the FTD. It defines the rate at which the fire develops during its growth stage. Notably, since FIRE_GROWTH_TIME and FIRE_GROWTH_RATE both define the same growth stage in the HRR curve, only one of these inputs should be used for this fire parameter to avoid redundancy or conflicts.

PEAK_TIME: specifies the sampling range for the peak time variable within the FTD. It defines the duration of the fully developed stage, indicating how long this phase of fire development lasts.

DECAY_TIME: specifies the sampling range for the decay time variable within the FTD. This input defines the duration of the decay stage, indicating how long this phase of fire development lasts.

FIRE_DECAY_RATE: specifies the sampling range for the fire decay rate variable within the FTD. This input defines the rate at which the fire develops during its decay stage. Notably, since DECAY_TIME and FIRE_DECAY_RATE both define the same growth stage in the HRR curve, only one of these inputs should be used for this fire parameter to avoid redundancy or conflicts.

FYI: A user defined comment that can provide additional information about the input.

3.8.2. Namelist GFTD

&GFTD stands for "generating fire time duration" samples. It specifies the essential details required for generating the sample data for this parameter.

ID: specifies the ID name that is used in the FD-Gen input file to represent the data samples generated for FTD.

PRE_PARM_INFO_ID: serves as an identifier for the associated pre-parameter information as defined in the &IFTD.

GENERATOR_ID: identifies the sample generator used for parameter sampling. Based on the information provided in &IFTD, four sample generators should be specified, representing the incipient, fire growth, fully developed, and decay stages of fire development, respectively.

FYI: A user defined comment that can provide additional information for FTD sample generator.

3.8.3. Namelist IMHR

&IMHR stands for "inputting maximum heat release rate" information. It provides the necessary pre-parameter details required for generating sample data for MHR. The MHR is customized based on the stages of fire development. Therefore, the sampling range for the incipient, fully developed, and decay stages of fire development is specified in this namelist. Figure **9** illustrates the arguments used in designing the maximum HRR values across various stages of fire development.

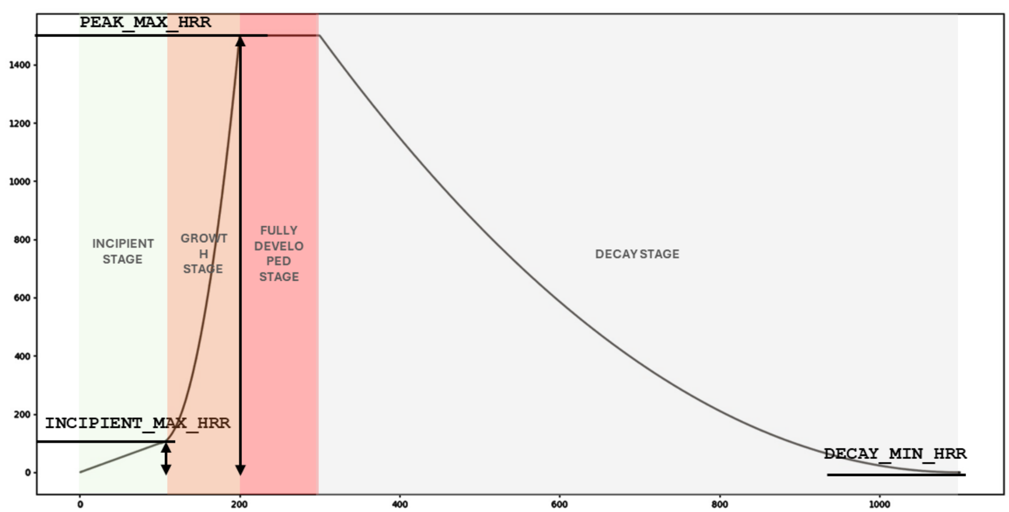


Figure 9. schematic diagram of the arguments used in the namelist &IMHR. A typical fire HRR curves divided into stages is illustrated.

PRE_PARM_INFO_ID: specifies the ID name that is used in the FD-Gen input file to identify the pre-parameter information input for the parameter.

INCIPIENT_MAX_HRR: specifies the sampling range for the maximum HRR during the incipient stage of fire development within the MHR. This input defines the range for the peak HRR reached during this initial stage, setting the upper intensity limit before the fire transitions to subsequent growth phases.

PEAK_MAX_HRR: specifies the sampling range for the peak maximum HRR variable within the MHR. It defines the range for the maximum HRR reached during the fully developed stage of fire development, representing the fire's highest intensity.

DECAY_MIN_HRR: specifies the sampling range for the minimum HRR variable during the decay stage within the MHR. This input defines the range for the minimum HRR at the conclusion of the fire's decay stage, capturing the fire's lowest intensity.

3.8.4. Namelist GMHR

&GMHR stands for "generating maximum heat release rate" samples. It specifies the essential details required for generating the sample data for this parameter.

ID: specifies the ID name that is used in the FD-Gen input file to represent the data samples generated for MHR.

PRE_PARM_INFO_ID: serves as an identifier for the associated pre-parameter information input as defined in the &IMHR.

GENERATOR_ID: identifies the sample generators used for parameter sampling. Based on the information provided in &IMHR, three sample generators should be specified, representing the sample generator of incipient maximum HRR, peak maximum HRR, and decay minimum HRR, respectively.

FYI: a user defined comment that can provide additional information for MHR sample generator.

3.8.5. Namelist GHRC

&GHRC refers to generate the HRR curves. It specifies the essential details required for generating the HRR samples and integrating them into the FDS case input file. The argument input in this namelist include ID, FIRE_TIME_DURATION_GENERATOR_ID, MAX_HRR_GENERATOR_ID, FIRE_PATTERN, START_TIME, END_TIME, TIME_FREQUENCY, HRR_FLUCTUATION, HRR_FLUCTUATION_RATE, TIME_SLICE_SAMPLES, HRR_SAMPLES, PRODUCT, and FYI.

ID: specifies the ID name for the HRR curve as defined in the FDS case input file.

FIRE_TIME_DURATION_GENERATOR_ID: identifies the FTD ID used for fire time duration data sampling. Here, it is noted that if multiple items of HRR curves are defined, the same number of ID names for FTD should be applied, denoted as FIRE_TIME_DURATION_GENERATOR_ID='FTD1','FTD2',….

MAX_HRR_GENERATOR_ID: identifies the MHR ID used for maximum fire HRR data sampling. Here, it is noted that if multiple items of HRR curves are defined, the same number of ID names of MHR should be written, denoted as MAX_HRR_GENERATOR_ID ='MHR1','MHR2',….

FIRE_PATTERN: specifies the pattern of the HRR curve to be followed. It is defined in the format of an example of FIRE_PATTERN=1,2,0,2; where the four values represent the pattern of the HRR curve at different stages. Each value corresponds to an exponent index that indicates the pattern: 0(Constant), 1(linear increase), 2 or large (exponential increase). For example, FIRE_PATTERN=1,2,0,2 represents an HRR curve with a linear increase during the incipient stage, a squared increase during the growth stage, steady during the fully developed stage, and a squared decrease during the decay stage.

START_TIME: specifies the start time of the item being burned. If START_TIME is specified, the HRR curve will be generated beginning at this specified time. If not defined, it will default to start at 0 s, aligning with the beginning of the fire simulation.

END_TIME: specifies the end time for the item being burned. If END_TIME is specified, the HRR curve will terminate at this specified time. If not defined, the simulation will default to terminate at 1800 s from the start of the fire simulation or after all stages of fire development have been completed, whichever occurs first, provided the total fire duration is less than 1800 s.

TIME_FREQUENCY: defines the frequency at which the HRR curve data is written to the FDS case input file. By default, the value is set to 1 s, meaning the changes in the designed HRR curve are inputted every second of the fire simulation. This ensures that the HRR curve is accurately represented in the FDS input file at the specified intervals.

HRR_FLUCTUATION: provides a mode for sampling HRR curves with a fixed variation. When activated, this mode generates multiple HRR curves by adjusting max HRR either upward or downward by a specified percentage, enabling controlled variability in fire scenarios. The default setting for this input is False, indicating that the mode is not activated unless explicitly specified.

HRR_FLUCTUATION_RATE: specifies the rate of variation in the HRR_FLUCTUATION mode.

TIME_SLICE_SAMPLES: specifies a user-defined list of time frame values for the HRR curve. These values define specific moments in time for which corresponding HRR data points will be included.

HRR_SAMPLES: a user-defined list of HRR values corresponding to the time frames specified in TIME_SLICE_SAMPLES. These values represent the heat release rate at each defined time point and are directly wrapped into the FDS case input file, ensuring that the HRR curve aligns with the user's design.

`PRODUCT:` provides an option to generate scenarios by combining all sampled fire parameters using the Cartesian product. The default option for this argument is `False`, meaning that the formation of fire scenarios will continue to use a standard projection rather than incorporating all possible parameter combinations for this item.

`FYI:` a user defined comment that can provide additional information about the HRR curve data generation process.

3.8.1. Examples

- Script

```
===========================================================
======= HRR CURVE =========================================
# FD-Gen code
&IFTD PRE_PARM_INFO_ID='FTD', INCIPIENT_TIME=100,100,
FIRE_GROWTH_TIME=100,200, PEAK_TIME=50,150, DECAY_TIME=400,800/
&MRND GENERATOR_ID='FG1', PRE_PARM_INFO_ID='FTD_I',
DISTRIBUTION_TYPE='CONSTANT'/
&MRND GENERATOR_ID='FG2', PRE_PARM_INFO_ID='FTD_G',
DISTRIBUTION_TYPE='NORMAL', MEAN=120, STDEV=50/
&MRND GENERATOR_ID='FG3', PRE_PARM_INFO_ID='FTD_P',
DISTRIBUTION_TYPE='UNIFORM'/
&MRND GENERATOR_ID='FG4', PRE_PARM_INFO_ID='FTD_D',
DISTRIBUTION_TYPE='UNIFORM'/
# max hrr
&IMHR PRE_PARM_INFO_ID='MAX_HRR', INCIPIENT_MAX_HRR=50,50,
PEAK_MAX_HRR=1000,8000, DECAY_MIN_HRR=0,0/
&MRND GENERATOR_ID='incipient', PRE_PARM_INFO_ID='MAX_HRR_I',
DISTRIBUTION_TYPE='CONSTANT'/
&MRND GENERATOR_ID='peak', PRE_PARM_INFO_ID='MAX_HRR_P',
DISTRIBUTION_TYPE='UNIFORM'/
&MRND GENERATOR_ID='decay', PRE_PARM_INFO_ID='MAX_HRR_D',
DISTRIBUTION_TYPE='CONSTANT'/
# generate the curve
&GFTD ID='FTD', PRE_PARM_INFO_ID='FTD',
GENERATOR_ID='FG1','FG2','FG3','FG4'/
&GMHR ID='MAX_HRR', PRE_PARM_INFO_ID='MAX_HRR',
GENERATOR_ID='incipient','peak','decay'/
&GHRC ID='CURVE', FIRE_TIME_DURATION_GENERATOR_ID='FTD',
MAX_HRR_GENERATOR_ID='MAX_HRR', FIRE_PATTERN=1,2,0,2/
===========================================================
```

```
# FDS code
&SURF ID='CURVE', HRRPUA=436, RAMP_Q='CURVE_fireramp', COLOR='RED'/
==============================================================
```

- Explanation

In this example, the FD-Gen script of generating FDS case input files for randomized HRR curves associated with a single ignited item is detailed.

`&IFTD`

`PRE_PARM_INFO_ID`: in this example, `PRE_PARM_INFO_ID` is set to `'FTD'`, which designates an identifier for retrieving and referencing the pre-parameter information related to `FTD`.

`INCIPIENT_TIME`: the input for the time duration during the incipient stage of fire development is specified as a range from 100 s to 100 s, indicating a fixed time duration for the incipient stage.

`FIRE_GROWTH_TIME`: the input for the time duration during the growth stage of fire development is specified as a range from 100 s to 200 s. This indicates that the fire's growth stage will last for a minimum of 100 s and a maximum of 200 s, allowing for variability in the fire development scenarios.

`PEAK_TIME`: specifies the time duration for the fully developed stage of fire development. The input range is defined from 50 s to 150 s, indicating that this stage will last a minimum of 50 s and a maximum of 150 s, representing variability in the peak fire intensity phase.

`DECAY_TIME`: specifies the time duration for the decay stage of fire development. The input range is defined from 400 to 800 s, indicating that the fire's decay stage will last a minimum of 400 s and a maximum of 800 s, reflecting the gradual reduction in fire intensity over time.

`&MRND`

`GENERATOR_ID`: four sample generators are required to define the time durations for the four stages of fire development. In this example, the sample generators are specified as `'FG1'`, `'FG2'`, `'FG3'`, and `'FG4'`.

`PRE_PARM_INFO_ID`: within the sample generators, the generator can inherit the pre-parameter information from `&IFTD`. Since there is only one `PRE_PARM_INFO_ID`, but the information needs to be applied to four individual sample generators (for four stages of fire development), separate identifiers are used for each variable. Specifically, for `&IFTD`, the following identifiers are applied to represent the inheritance of information for each fire development stage: `PRE_PARM_INFO_ID+'_I'`, `PRE_PARM_INFO_ID+'_G'`, `PRE_PARM_INFO_ID+'_P'`, and `PRE_PARM_INFO_ID+'_D'`. Additionally, if `FIRE_GROWTH_RATE` or `FIRE_DECAY_RATE` is used in place of `FIRE_GROWTH_TIME` or `DECAY_TIME`, the corresponding variable names should be `PRE_PARM_INFO_ID+'_GR'` or PRE_PARM_INFO_ID+'_DR'. Therefore, in this example, PRE_PARM_INFO_ID for each individual sample generators are represented in the syntax of 'FTD_I', 'FTD_G', 'FTD_P', and 'FTD_D'.

DISTRIBUTION_TYPE: in this setup, four distinct sampling distributions are used to generate sample data for variables in FTD. The time duration of the incipient stage remains constant. The time duration during the fire growth stage is sampled using a normal distribution. The time duration of the fully developed stage is sampled using a uniform distribution. The decay stage also uses a uniform distribution.

&IMHR

PRE_PARM_INFO_ID: in this example, PRE_PARM_INFO_ID is set to 'MAX_HRR', which designates an identifier for retrieving and referencing the pre-parameter information input related to MHR.

INCIPIENT_MAX_HRR: specifies the maximum HRR during the incipient stage of fire development. In this example, the value is fixed at 50 kW, indicating a fixed peak HRR during this initial stage.

PEAK_MAX_HRR: specifies the maximum HRR during the fully developed stage of fire development. The range is set between 1000 kW and 8000 kW, allowing for variability in the peak HRR during the fully developed stage across different scenarios.

DECAY_MIN_HRR: specifies the minimum HRR during the decay stage of fire development. Here, the value is fixed at 0 kW, indicating that the HRR curve is designed to reduce to zero at the end of the fire simulation, fully extinguishing the fire.

&MRND

GENERATOR_ID: three sample generators are required to define the maximum or minimum HRR of incipient, fully developed, and decay. In this example, the sample generators are specified as 'incipient', 'peak', and 'decay'.

PRE_PARM_INFO_ID: within the sample generators, the generator can inherit the pre-parameter information from &IMHR. Since there is only one PRE_PARM_INFO_ID, but the information needs to be applied to three individual sample generators (for maximum or minimum HRR of incipient, fully developed, and decay stages), separate identifiers are used for each variable. Specifically, for &IMHR, the following identifiers are applied to represent the inheritance of information for each variable: PRE_PARM_INFO_ID+'_I', PRE_PARM_INFO_ID+'_P', and PRE_PARM_INFO_ID+'_D'. Therefore, in this example, PRE_PARM_INFO_ID for each individual sample generators are represented in the syntax of 'MAX_HRR_I', MAX_HRR_P', and 'MAX_HRR_D'.

DISTRIBUTION_TYPE: in this setup, three distinct sampling distributions are used to generate sample data for variables in MHR. The maximum HRR during the incipient stage remains constant. The maximum HRR during the fully developed stage is sampled using a uniform distribution. The minimum HRR during the decay stage remains constant in zero.

&GFTD

ID: the ID name of FTD data samples is labeled as 'FTD'.

PRE_PARM_INFO_ID: this input inherits pre-parameter information input from the &IFTD inputs. Therefore, it maintains consistency with the PRE_PARM_INFO_ID specified in &IFTD, using 'FTD'.

GENERATOR_ID: this argument input specifies the use of the defined sample generator, which is align with the GENERATOR_ID specified in &MRND. It includes four sample generators listed as GENERATOR_ID='FG1','FG2','FG3','FG4', each corresponding to the data samples for a specific stage of fire development.

&GMHR

ID: the ID name of MHR data samples is labeled as 'MAX_HRR'.

PRE_PARM_INFO_ID: this input inherits pre-parameter information from the &IMHR inputs. Therefore, it maintains consistency with the PRE_PARM_INFO_ID specified in &IMHR, using 'MAX_HRR'.

GENERATOR_ID: this input specifies the use of the defined sample generators, which is align with the GENERATOR_ID specified in &MRND. It includes three sample generators listed as GENERATOR_ID='incipient','peak','decay', each corresponding to the data samples for a specific stage of fire development.

&GHRC

ID: in this example, the ID name of the generation of HRR curves is labeled as ' CURVE', aligning with the unique ID name defined in FDS code lines.

FIRE_TIME_DURATION_GENERATOR_ID: the value of this input is 'FTD', indicating using the designed FTD data samples to generate the HRR curves.

MAX_HRR_GENERATOR_ID: the value of this input is ' MAX_HRR', indicating using the designed MHR data samples to generate the HRR curves.

FIRE_PATTERN: the fire pattern in this example is 1, 2, 0, 2, representing an HRR curve with a linear increase during the incipient stage, a squared increase during the growth stage, a steady value during the fully developed stage, and a squared decrease during the decay stage.

With the example executed in FD-Gen, the sampled HRR curves will be incorporated into the FDS case input file. The corresponding line of the FDS script will be updated with the new sampled data, as illustrated below:

Before:

```
&SURF ID='CURVE', HRRPUA=4360, RAMP_Q='CURVE_fireramp', COLOR='RED'/
```

(When preparing the FD-Gen input file, the FDS code line for HRR should be formatted as above, ensuring that HRRPUA and RAMP_Q are explicitly included. Additionally, all existing &RAMP lines must be removed to avoid conflicts with the new &RAMP code lines generated by FD-Gen.)

After (example of a single FDS input file):

```
&SURF ID='CURVE', HRRPUA=3877.0, RAMP_Q='CURVE_fireramp', COLOR='RED'/
&RAMP ID='CURVE_fireramp', T=1.0, F=0.00012896569512509673 /
&RAMP ID='CURVE_fireramp', T=2.0, F=0.00025793139025019347 /
&RAMP ID='CURVE_fireramp', T=3.0, F=0.0003868970853752 9015 /
&RAMP ID='CURVE_fireramp', T=4.0, F=0.0005158627805003869 /
&RAMP ID='CURVE_fireramp', T=5.0, F=0.0006448284756254836 /
&RAMP ID='CURVE_fireramp', T=6.0, F=0.0007737941707505803 /
…
```

It is observed that the designed HRR curve is integrated into the FDS case input file, with the ramp of the HRR curve specified for every second. A plot of all the sampled HRR curves in this example (10 cases) is displayed in **Figure 10**.

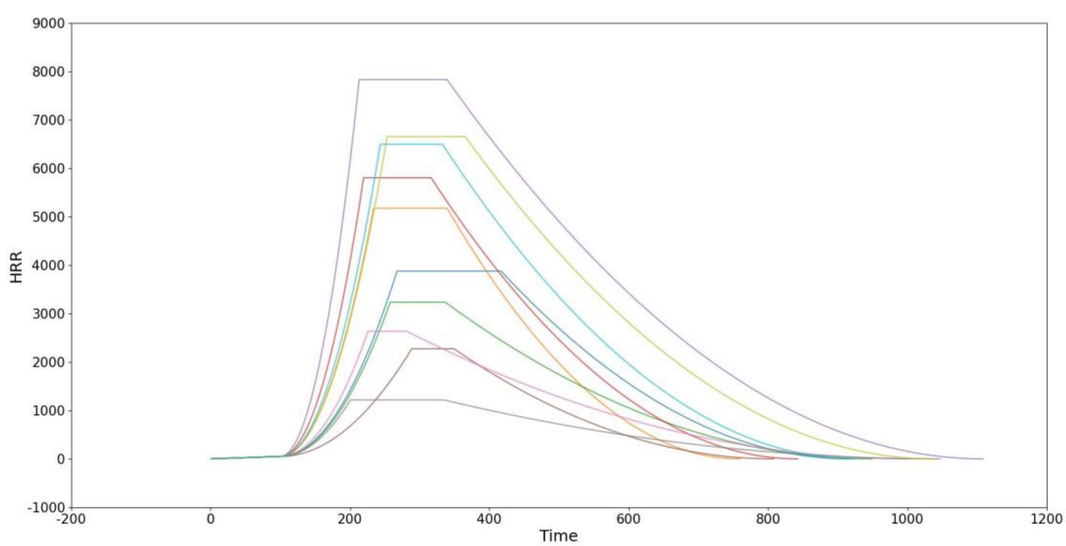


Figure 10. Plot of the sampled HRR curves.

4. Creating multiple FDS input files

By compiling the FD-Gen input file using the namelists above, we can customize random fire scenarios with a wide range of fire parameters, such as random fire source locations, door opening times, and HRR curves. This section details the process and steps of using the compiled FD-Gen input file to create multiple FDS case input files. To illustrate, the process is demonstrated using the same example of generating random HRR curves in section 3.8.

4.1. Step 1: Start

4.1.1. Navigate to the FD-Gen Program Directory

Open a terminal and use the command to navigate to the location of the FD-Gen main program.

4.1.2. Run the FD-Gen Input File

If you are using the executable file, run the following command:

```
.\FD-Gen <FD-Gen project name>.fds
```

If you are using the Python script directly, use the following command:

```
.\python FDSdata.py <FD-Gen project name>.fds
```

4.2. Step 2: script checking

Once the FD-Gen input file is executed, the program reads the FD-Gen input file and performs basic code checking. The following information may be indicated in the terminal:

4.2.1. Project basic information

```
========================================
===========  Basic INFO ================
PROJECT_NAME          ==> test_script_HRC
CURRENT PATH          ==> ...\test_script_HRC
NUMBER_OF_CASES       ==> 10
SEEDS                 ==> 1
========================================
SAVE_SEEDS file created successfully.
========================================
```

First, project basic information will show up. Information such as project name, path of the file, number of cases, and seed number will be indicated. The output folder `SEEDS_FOLDER` containing the information of seed number of the total number of cases will be created.

4.2.2. Script checking information

```
========================================
========== Script checking =============
=====> Check script length
FDS script total length:            15
Data generator script total length:  12
```

Second, the program runs an auto-check to identify errors or inconsistencies in the FD-Gen input file. These above lines of information indicate the total number of FDS code lines and FD-Gen code lines. In the FD-Gen input file, any code line that does not use the FD-Gen namelist will be counted as an FDS code line.

```
=====> Check PARAMETER PRE-INPUT information
PRE_PARM_INFO_ID: FTD              ==>     Namelist - IFTD
PRE_PARM_INFO_ID: MAX_HRR          ==>     Namelist - IMHR
```

This part of the context provides a checking result of the pre-parameter information entered in the FD-Gen input file (all the code lines to start with `I***`). It lists the ID name used to identify the pre-parameter along with its corresponding namelist class.

```
=====> Check SAMPLE GENERATOR information
GENERATOR_ID: FG1                  ==>     PRE_PARM_INFO_ID = FTD_I
GENERATOR_ID: FG2                  ==>     PRE_PARM_INFO_ID = FTD_G
GENERATOR_ID: FG3                  ==>     PRE_PARM_INFO_ID = FTD_P
GENERATOR_ID: FG4                  ==>     PRE_PARM_INFO_ID = FTD_D
GENERATOR_ID: incipient            ==>     PRE_PARM_INFO_ID =
MAX_HRR_I
GENERATOR_ID: peak                 ==>     PRE_PARM_INFO_ID =
MAX_HRR_P
GENERATOR_ID: decay                ==>     PRE_PARM_INFO_ID =
MAX_HRR_D
```

This part of the context provides a checking result of the sample generator entered in the FD-Gen input file (all the code lines to start with `MRND`). The ID name used to identify the sample generator along with its corresponding ID name for pre-parameter information input is indicated.

```
=====> Check matched FDS ID
ID: FTD                   ==>NOT applicable     FDS ID
=====> Check matched PRE_PARM_INFO_ID
```

```
ID: FTD                    ==>Match found     PRE_PARM_INFO_ID: FTD
=====> Check matched GENERATOR_ID
ID: FTD                    ==>Match found     GENERATOR_ID: FG1
ID: FTD                    ==>Match found     GENERATOR_ID: FG2
ID: FTD                    ==>Match found     GENERATOR_ID: FG3
ID: FTD                    ==>Match found     GENERATOR_ID: FG4
```

This section of context provides a checking result of all fire parameters designated for sampling in the FD-Gen input file (all the code lines to start with `G***`). The first step involves verifying whether the parameter's ID name matches a corresponding parameter ID in the FDS code. This match is essential, as the ID name determines the precise FDS code line where the sampled data will be integrated into the FDS case input file. It is therefore critical, as emphasized in the namelist section, that the ID name used for sample generation remains consistent with the one defined in the FDS case input file. In this example, the checking result is marked as `NOT applicable`, since the sample parameter, FTD-fire time duration, is solely used for generating samples and is later transformed into HRR curve data. This parameter does not require direct integration into the FDS case input file. For a regular fire parameter, the result would display `Match found`, along with the corresponding ID name from the FDS code, confirming successful identification and alignment.

Subsequently, the process checks the `PRE_PARM_INFO_ID` to determine whether the fire parameter is linked to any specific pre-parameter information input. If such a link exists, a `Match Found` will be displayed, along with the linked pre-parameter information ID.

Lastly, the process examines the parameter's linked sample generator IDs. If the sample generators are identified, a `Match  Found` will be displayed, along with the generator's ID name.

```
=====> Check matched FDS ID
ID: MAX_HRR                ==>NOT applicable     FDS ID
=====> Check matched PRE_PARM_INFO_ID
ID: MAX_HRR                ==>Match found     PRE_PARM_INFO_ID: MAX_HRR
=====> Check matched GENERATOR_ID
ID: MAX_HRR                ==>Match found     GENERATOR_ID: incipient
ID: MAX_HRR                ==>Match found     GENERATOR_ID: peak
ID: MAX_HRR                ==>Match found     GENERATOR_ID: decay
```

The same checking process is applied to the fire parameter with the ID name `MAX_HRR`. In this example, `MAX_HRR` represents the maximum heat release rate data samples of the HRR curves. Thus, it is also marked as `Not Applicable` in the FDS ID name checking process, as it does not require a match in the FDS ID name. However, matches for the pre-parameter and sample generator are still available. All linked pre-parameter IDs and sample generators are listed accordingly.

```
=====> Check matched FDS ID
ID: CURVE               ==>Match found     FDS ID
=====> Check matched PRE_PARM_INFO_ID
ID: CURVE               ==>NOT applicable     PRE_PARM_INFO_ID
=====> Check matched GENERATOR_ID
ID: CURVE               ==>NOT applicable     GENERATOR_ID
========================================
========================================
```

Eventually, the checking process is applied to the HRR curve input, which is a regular fire parameter influencing the file wrapping in the FDS case input file. In this example, the parameter is denoted as `CURVE`. If the checking process does not find `CURVE` matching the parameter ID name used in FDS code lines, a `No Match Found` indicator will appear. Additionally, asterisks (`*`) and a `WARNING` indicator will be displayed to highlight the issue detected during the checking process. Although this warning does not automatically terminate the FD-Gen program, it is strongly recommended to carefully review the FD-Gen input file to identify and resolve any unmatched parameter issues. Ensuring consistency between the inputs and parameter defined in FDS case input file is crucial for proper functionality. Similarly, match checking for the pre-parameter and sample generator is also conducted. However, in this case, the checking process is not applicable for the ongoing parameter, as the HRR curve parameter only utilizes the data samples from `FTD` and `MHR` for curve data creation. Details regarding the use of `&GHRC` for generating randomized HRR curves are provided in the Namelist `&GHRC`.

```
Do you want to proceed to the SAMPLING step? (Y/N):
```

After completing the script checking process, a message will prompt for input. If a `Y` or `y` response is provided, the FD-Gen program will automatically proceed to the sampling step, generating sample data as designed in the input file. Conversely, if an `N` or `n` response is given, the program will automatically terminate, allowing you to review, edit the input file, and redo the script checking process as needed.

4.3. Step 3: parameter sampling

Once the program goes into the sampling process, the program follows the FD-Gen code lines to sample the fire parameter data samples as designed. The following information may be indicated in the terminal:

4.3.1. Sampling progress information

```
========================================
======= Parameter data sampling ========
========================================
```

```
== Parameter (fire time duration) - FTD ==
==> Generating samples...
GENERATOR_ID = FG1
CONSTANT: Successfully applied
APPROX: Successfully applied
GENERATOR_ID = FG2
NORMAL: Successfully applied
APPROX: Successfully applied
GENERATOR_ID = FG3
UNIFORM: Successfully applied
APPROX: Successfully applied
GENERATOR_ID = FG4
UNIFORM: Successfully applied
APPROX: Successfully applied
==> Data samples generated Successfully.
========================================
```

The text information outlines the sampling process for all sample generators specified for each parameter in the FD-Gen input file. For each generator, the generator ID, distribution type, and value type applied are displayed to provide a clear indication of how the sampling process is progressing. In this example, all the sample data for fire parameter under the ID name `FTD` is successfully generated.

```
========================================
== Parameter (max HRR) - MAX_HRR ==
==> Generating samples...
GENERATOR_ID = incipient
CONSTANT: Successfully applied
APPROX: Successfully applied
GENERATOR_ID = peak
UNIFORM: Successfully applied
APPROX: Successfully applied
GENERATOR_ID = decay
CONSTANT: Successfully applied
APPROX: Successfully applied
==> Data samples generated Successfully.
========================================
```

In a same way, the sample data for fire parameter under the ID name MAX_HRR is successfully generated.

```
========================================
== Parameter (fire HRR curve) - CURVE ==
==> Generating samples...
==> Data samples generated Successfully.
========================================
```

Finally, the HRR curve data under the ID name CURVE is successfully created.

4.3.2. Saved parameter information

```
*****PARAMETER_FILE file created successfully.
========================================
========================================
```

After sampling all the fire parameters, the output folder PARAMETER_FILE_FOLDER will be created automatically. It contains three files inside:

(1) Generator_info.txt

This file provides comprehensive information utilized by the sample generator for generating sample data. For example, detailed information about the generator ID name FG1 is attached below. Within the context, all argument inputs used by the sample generator to produce the data samples are summarized.

```
GENERATOR_ID: FG1
1 Parameter ID: ['FTD'],
2 FYI: None,
3 Distribution Type: CONSTANT,
4 Random Seeds: 37,
5 Constant: [100],
6 Minimum: None,
7 Peak: None,
8 Maximum: None,
9 Mean: None,
10 Stdev: None,
11 Increase: None,
12 Value Type: APPROX,
13 Approx Value: 1.0,
14 Logical Value: None,
```

```
Shuffle State: True,
Shuffle Random Seeds: 12,
Number of Samples: 10,
Samples: [100. 100. 100. 100. 100. 100. 100. 100. 100. 100.]
```

(2) Param_brief.txt

This file provides comprehensive information regarding the fire parameter taking into consideration in the FD-Gen project. For example, detailed information about the parameter ID name `CURVE` is attached below. Within the context, all argument inputs to produce the specific samples of the HRR curves are summarized.

```
ID: CURVE
Fire Time Duration Generator ID: ['FTD'],
Max HRR Generator ID: ['MAX_HRR'],
Fire Pattern: [1, 2, 0, 2],
FYI: None,
Time Frequency: 1,
Time Start: 0,
Time End: 1800,
HRR Fluctuation: False,
HRR Fluctuation Rate: 0,
Number of Samples: 10,
Time Slice Samples: [1.,2.,3.,4.,5.,6.,7.,8.,9.,10.,11.…]
HRR Samples:[…, 5.33937545e+01, 5.48860876e+01, 5.66496148e+01,
5.86843361e+01, 6.09902515e+01, 6.35673610e+01, 6.64156646e+01,
6.95351624e+01, 7.29258543e+01, 7.65877403e+01, 8.05208204e+01,
8.47250947e+01, 8.92005630e+01, 9.39472255e+01, 9.89650821e+01,…]
```

(3) Pararmeter_outputs.xlsx

The output also includes a spreadsheet detailing the values of fire parameters that will be wrapped into the FDS case input file for each scenario. The following table, as shown in **Table 2**, represents this spreadsheet output. Each row corresponds to a specific fire scenario case, displaying the applied values for `FTD` and `MHR` at different stages of fire development for that case. The data in this spreadsheet serves as the foundational samples for embedding randomized information into multiple FDS case input files.

```
Do you want to proceed to the WRAPPING step? (Y/N):
```

After completing the sampling process, a message will prompt for input. If a `Y` or `y` response is provided, the FD-Gen program will automatically proceed to the file wrapping step, using the sampled data to generate the FDS case files. Before proceeding, it is recommended to review the data information files located in the PARAMETER_FILE_FOLDER. This ensures that the generated data samples align with your intended design. If the parameter samples are found to deviate from the intended design, you can respond with N or n, and the program will terminate. This allows you to review and edit the input file until the parameter samples are properly generated as designed.

Table 2. Spreadsheet output of sampled fire scenario details.

Cases	<FTD> ID - FTD .Incipient	<FTD> ID - FTD .Growth	<FTD> ID - FTD .Peak	<FTD> ID - FTD .Decay	<MHR> ID - MAX_HRR .Incipient	<MHR> ID - MAX_HRR .Peak	<MHR> ID - MAX_HRR .Decay	<HRC> ID - CURVE .Fire time duration	<HRC> ID - CURVE .HRR samples
0	100	168	149	500	50	3877	0	…	…
1	100	134	105	420	50	5174	0	…	…
2	100	158	79	613	50	3236	0	…	…
3	100	120	97	526	50	5806	0	…	…
4	100	113	126	769	50	7833	0	…	…
5	100	189	61	459	50	2273	0	…	…
6	100	126	56	721	50	2634	0	…	…
7	100	101	134	713	50	1218	0	…	…
8	100	153	112	671	50	6656	0	…	…
9	100	143	89	593	50	6497	0	…	…

4.4. Step 4: parameter wrapping

The file wrapping process is responsible for generating the FDS case input files based on the sampled data. The terminal output may include the following information:

4.4.1. File wrapping information

```
========================================
======  Parameter data wrapping ========
*****CASE_FOLDER file created successfully.
========================================
========================================
```

The text information indicates the completion of the file wrapping process. The CASE_FOLDER will be created automatically, containing all the generated new FDS case input files. These files will be named following the format <project name>_00000.fds, <project name>_00001.fds, and so on. Within each file, the parameter values are wrapped with the generated data, ensuring that the sampled values are correctly integrated into the FDS case input files.

4.4.2. Data plotting information

```
========================================
======  Parameter data plotting ========
*****OUTPUT_FOLDER file created successfully.
========================================
========================================
```

In addition to generating the FDS case input files, the FD-Gen program will also plot the fire parameters specified in the FD-Gen input file by namelist &MAIN. The resulting figures for these fire parameters will be saved in the OUTPUT_FOLDER, providing a visual representation of the specified parameter distributions and sampled values.

5. Examples

In this section, several case examples are conducted to illustrate the process of generating multiple FDS case files tailored to specific fire scenarios. Among these examples, we also demonstrate advanced functionalities, including the use of the `PRODUCT` to generate scenarios by combining all sampled fire parameters using the Cartesian product. Other features covered include HRR curves with multiple item ignition and enabling `HRR_FLUCTUATION` for sampling HRR curves with a fixed variation.

5.1. Fire in a commercial building structure

This example demonstrates an FD-Gen project designed to generate multiple random fire scenarios within a commercial building layout. The building geometry used in this example is based on the works of [1] and [4]. This geometry serves as a foundation for developing ML models aimed at fire source localization and firefighter tenability analyses during fire emergency response operations. Table **3** provides a detailed summary of the randomized fire parameters considered in this example. The floor plan of the building structure is illustrated in **Figure 11**.

Table 3. Summary of fire parameters considered in the commercial building fire scenario design.

Fire parameters		Unit	Distribution types	Min	Max	Constant	Mean(μ)	SD(σ)
Fire source location	Fire source (X_{loc})	m	Uniform	0	8.7	-	-	-
	Fire source (Y_{loc})	m	Uniform	22.1	26.3	-	-	-
	Fire source (Z_{loc})	m	Constant	-	-	0	-	-
HRR curves	Incipient time duration	s	Constant	-	-	100	-	-
	Fire growth time duration	s	Normal	100	200	-	120	50
	Fully developed time duration	s	Uniform	50	150	-	-	-
	Decay time duration	s	Uniform	400	800	-	-	-
	Maximum HRR (Incipient)	kW	Constant	-	-	50	-	-
	Maximum HRR (peak)	kW	Uniform	1000	8000		-	-
	Minimum HRR (decay)	kW	Constant	-	-	0	-	-
Door open time	Stage 1	s	Uniform	100	150	-	-	-
	Stage 2	s	Uniform	200	250	-	-	-
	Stage 3	s	Uniform	250	300	-	-	-

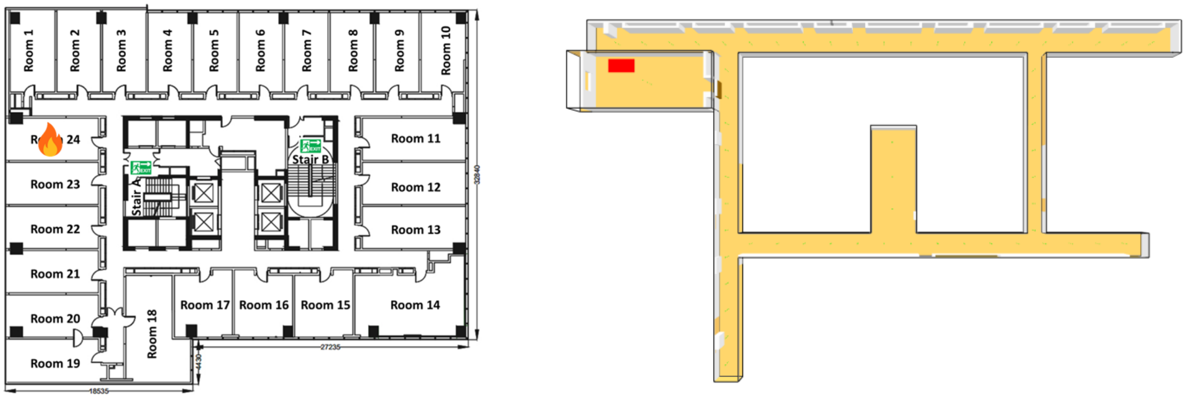


Figure 11. Floor plan of the commercial building structure.

5.1.1. FD-Gen code breakdown

To provide a clear illustration, the script lines within the FD-Gen input file is explained below. These listed script lines outline the patterns and rules that the considered fire parameters should adhere to during the generation process. By specifying these rules, the script ensures that the resulting FDS case input file is configured correctly for the desired fire scenarios. The attached script includes only the FD-Gen code sections relevant to the randomized fire parameters. Sections of the script unrelated to these parameters and belonging to the original FDS code have been omitted here for clarity. The complete FD-Gen input file for this example is available in the GitHub repository's example folder for reference and use.

```
==============================================================
=========== Main INFO =======================================
&MAIN PROJECT_NAME='example_commercial', NUMBER_OF_CASES=10, SEEDS=10/
&PFSL FSL_ID='fire source'/
&PHRC HRC_ID='CURVE'/
&PPSD PSD_ID='dot_timer1','dot_timer2','dot_timer3'/
==============================================================
```

First, the basic information for the FD-Gen project is specified, including the project name, the number of cases to be generated, and the seed number for randomization. Additionally, this example includes commands to plot two fire parameters: `'fire source'` and `'CURVE'`. These parameters are specified using the `&PFSL` and `&PPSD` namelists, representing plots of random fire source locations and HRR curves, respectively. Furthermore, the pattern of sampled data in the sample generators `'dot_timer1'`,`'dot_timer2'`, and `'dot_timer3'` is also specified, providing a visual representation of the sample data.

```
==============================================================
======= Door open time ======================================
# FD-Gen code
```

```
&IDWT PRE_PARM_INFO_ID='door open time',
      OBST_ID='door24_1','door24_2','door24_3'
      XB=9.4,9.6,23.15,24.3,0.0,2.0, OPEN_DIRECTION='Y-'
      QUANTITY_NAME='TIME', PORTION=0.5,0.75,1
      DEVC_ID='dot_timer1','dot_timer2','dot_timer3'
      DOOR_OR_WINDOW_OPEN_TIME_RANGE=100,150,200,250,250,300 /
&MRND GENERATOR_ID='dot_timer1', PRE_PARM_INFO_ID='door open time_1',
DISTRIBUTION_TYPE='UNIFORM'/
&MRND GENERATOR_ID='dot_timer2', PRE_PARM_INFO_ID='door open time_2',
DISTRIBUTION_TYPE='UNIFORM'/
&MRND GENERATOR_ID='dot_timer3', PRE_PARM_INFO_ID='door open time_3',
DISTRIBUTION_TYPE='UNIFORM'/
&GDWT ID='door24', PRE_PARM_INFO_ID='door open time',
GENERATOR_ID='dot_timer1','dot_timer2','dot_timer3'/
==========================================================
# FDS code
&DEVC ID='dot_timer1', QUANTITY='TIME', XYZ=0.0,26.3,0.0,
SETPOINT=500, INITIAL_STATE=.TRUE./
&DEVC ID='dot_timer2', QUANTITY='TIME', XYZ=0.0,26.3,0.0,
SETPOINT=500, INITIAL_STATE=.TRUE./
&DEVC ID='dot_timer3', QUANTITY='TIME', XYZ=0.0,26.3,0.0,
SETPOINT=500, INITIAL_STATE=.TRUE./
&OBST ID='door24_1', XB=9.4,9.6,23.15,24.3,0.0,2.0, RGB=153,102,0,
SURF_ID='INERT', DEVC_ID='dot_timer1'/
&OBST ID='door24_2', XB=9.4,9.6,23.15,24.3,0.0,2.0, RGB=153,102,0,
SURF_ID='INERT', DEVC_ID='dot_timer2'/
&OBST ID='door24_3', XB=9.4,9.6,23.15,24.3,0.0,2.0, RGB=153,102,0,
SURF_ID='INERT', DEVC_ID='dot_timer3'/
==========================================================
```

The second part of the script specifies the design of the door opening sequence. Using the `&IDWT` namelist, the script defines a three-stage door opening process. The room door is configured to open from the Y-direction in increments of ½ open, ¾ open, and fully open at each stage. For each stage, the door opens between 100 s to 150 s, 200 s to 250 s, and 250 s to 300 s after fire ignition, respectively. Subsequently, the sample generator for the parameter is defined using the `&MRND` namelist. Three sample generators are created to correspond to the three door opening stages defined in `&IDWT`. Each generator employs a uniform distribution, ensuring that the door open time within each interval is sampled uniformly. Finally, the `&GDWT` namelist is used to generate sample data and integrate the samples into the FDS case files, utilizing the pre-parameter information and sample generators as defined. In the subsequent FDS code, the ID names in `&DEVC` and `&OBST` are kept consistent with those defined in the FD-Gen script to ensure correspondence during the file wrapping process. Additionally, three lines of FDS code in `&DEVC` and `&OBST` are included to represent the three stages of the door opening process.

```
============================================================
================== HRR CURVE ===============================
# FD-Gen code
# fire time duration
&IFTD PRE_PARM_INFO_ID='FTD', INCIPIENT_TIME=100,100,
FIRE_GROWTH_TIME=100,200, PEAK_TIME=50,150, DECAY_TIME=400,800/
&MRND GENERATOR_ID='FG1', PRE_PARM_INFO_ID='FTD_I',
DISTRIBUTION_TYPE='CONSTANT'/
&MRND GENERATOR_ID='FG2', PRE_PARM_INFO_ID='FTD_G',
DISTRIBUTION_TYPE='NORMAL', MEAN=120, STDEV=50/
&MRND GENERATOR_ID='FG3', PRE_PARM_INFO_ID='FTD_P',
DISTRIBUTION_TYPE='UNIFORM'/
&MRND GENERATOR_ID='FG4', PRE_PARM_INFO_ID='FTD_D',
DISTRIBUTION_TYPE='UNIFORM'/
# max HRR
&IMHR PRE_PARM_INFO_ID='MAX_HRR', INCIPIENT_MAX_HRR=50,50,
PEAK_MAX_HRR=1000,8000, DECAY_MIN_HRR=0,0/
&MRND GENERATOR_ID='incipient', PRE_PARM_INFO_ID='MAX_HRR_I',
DISTRIBUTION_TYPE='CONSTANT'/
&MRND GENERATOR_ID='peak', PRE_PARM_INFO_ID='MAX_HRR_P',
DISTRIBUTION_TYPE='UNIFORM'/
&MRND GENERATOR_ID='decay', PRE_PARM_INFO_ID='MAX_HRR_D',
DISTRIBUTION_TYPE='CONSTANT'/
# generate HRR curves
&GFTD ID='FTD', PRE_PARM_INFO_ID='FTD',
GENERATOR_ID='FG1','FG2','FG3','FG4'/
&GMHR ID='MAX_HRR', PRE_PARM_INFO_ID='MAX_HRR',
GENERATOR_ID='incipient','peak','decay'/
&GHRC ID='CURVE', FIRE_TIME_DURATION_GENERATOR_ID='FTD',
MAX_HRR_GENERATOR_ID='MAX_HRR', FIRE_PATTERN=1,2,0,2/
============================================================
# FDS code
&SURF ID='CURVE', HRRPUA=436, RAMP_Q='CURVE_fireramp', COLOR='RED' /
============================================================
```

This section breaks down the script lines for designing HRR curves in the fire scenarios. Using the `&IFTD` and `&IMHR` namelists, the script defines the ranges of time duration and maximum HRR range for each stage of fire development. Based on the provided pre-parameter information input, the sample generator automatically draws data from these specified ranges. In this example, uniform, constant, and normal distribution types are employed to sample data across different fire development stages. Subsequently, the &GFTD and &GMHR are utilized to generate samples for the corresponding fire parameters, specifically with the ID of 'FTD' and 'MAX_HRR'. Finally, &GHRC namelist is applied to generate multiple HRR curves using the generated data samples of 'FTD' and 'MAX_HRR'. In the &GHRC namelist, the FIRE_PATTERN is specified as 1,2,0,2, indicating that the fire grows with a linear increase during the incipient stage, a squared increase during the growth stage, remains steady during the fully developed stage, and decreases with a squared pattern during the decay stage. In the subsequent FDS code, the FDS code corresponding to this part of the FD-Gen script consists of a single line of FDS code. This line includes the basic HRR information, such as ID, HRRPUA, RAMP_Q. The ID name for the parameter is designated as 'CURVE', ensuring consistency with the parameter ID defined in the FD-Gen &GHRC. This alignment guarantees proper correspondence during the file wrapping process, ensuring that the generated FDS case input file integrates the sampled HRR curve data accurately.

```
===========================================================

======= FIRE SOURCE LOCATION ==============================

&IFSL PRE_PARM_INFO_ID='fire source', XB=0,8.7,22.1,26.3,0,4,
FIRE_BURNER_SIZE=2,1,0.2, MESH_SIZE=0.2,0.2,0.2/

&MRND GENERATOR_ID='FS_X', PRE_PARM_INFO_ID='fire source_X',
DISTRIBUTION_TYPE='UNIFORM'/

&MRND GENERATOR_ID='FS_Y', PRE_PARM_INFO_ID='fire source_Y',
DISTRIBUTION_TYPE='UNIFORM'/

&MRND GENERATOR_ID='FS_Z', PRE_PARM_INFO_ID='fire source_Z',
DISTRIBUTION_TYPE='CONSTANT', CONSTANT=0/

&GFSL ID='fire source', PRE_PARM_INFO_ID='fire source',
GENERATOR_ID='FS_X','FS_Y','FS_Z'/

===========================================================

&OBST ID='fire source', XB=0,3.7,22.1,26.3,0.0,3.4,
SURF_IDS='CURVE','INERT','INERT'/

===========================================================
```

This part of the script specifies the design of the randomized fire source location. Using the &IFSL namelist, the script defines ranges and other relevant information related to fire source locations. Based on the information provided in &IFSL, the sample space XB is set as the coordinates of room 24, ensuring that the fire sources are randomly positioned within the room. Within the sample generators defined in the script, uniform distributions are applied to sample data for the fire source location along the room's length and width, while the height of the burner is fixed at the floor level. Finally, using the &GFSL namelist, the sample data for fire source locations is generated and integrated into the FDS case input file during the file wrapping process.

5.1.2. Example outputs

5.1.2.1. Example of created FDS case input files

(1) Door open time FDS code lines

```
============================================================
======= Door open time =====================================
===========================================================
&DEVC ID='dot_timer1', QUANTITY='TIME', XYZ=0.0,26.3,0.0,
SETPOINT=112.0, INITIAL_STATE=.TRUE./
&DEVC ID='dot_timer2', QUANTITY='TIME', XYZ=0.0,26.3,0.0,
SETPOINT=247.0, INITIAL_STATE=.TRUE./
&DEVC ID='dot_timer3', QUANTITY='TIME', XYZ=0.0,26.3,0.0,
SETPOINT=261.0, INITIAL_STATE=.TRUE./
&OBST ID='door24_1', XB=9.4, 9.6, 23.916666666666668, 24.3, 0.0, 2.0,
RGB=153,102,0, SURF_ID='INERT', DEVC_ID='dot_timer1', /
&OBST ID='door24_2', XB=9.4, 9.6, 23.533333333333333,
23.916666666666668, 0.0, 2.0, RGB=153,102,0, SURF_ID='INERT',
DEVC_ID='dot_timer2', /
&OBST ID='door24_3', XB=9.4, 9.6, 23.15, 23.533333333333333, 0.0, 2.0,
RGB=153,102,0, SURF_ID='INERT', DEVC_ID='dot_timer3', /
===========================================================
```

In the generated FDS case input files, it is observed that each file includes three stages of door opening, with each stage controlled by a sampled setpoint time. These setpoint times govern the door's transitions through its opening phases. Moreover, the entire door boundary is automatically divided into three sections, with each section corresponding to one stage of the door opening process.

(2) HRR curve FDS code lines

```
============================================================
======= HRR CURVE ==========================================
# max hrr
# generate the curve
============================================================
&SURF ID='CURVE', HRRPUA=6768.0, RAMP_Q='CURVE_fireramp', COLOR='RED'
/
&RAMP ID='CURVE_fireramp', T=1.0, F=7.387706855791962e-05 /
&RAMP ID='CURVE_fireramp', T=2.0, F=0.00014775413711583924 /
&RAMP ID='CURVE_fireramp', T=3.0, F=0.00022163120567375886 /
&RAMP ID='CURVE_fireramp', T=4.0, F=0.0002955082742316785 /
```

```
&RAMP ID='CURVE_fireramp', T=5.0, F=0.0003693853427895981 /
&RAMP ID='CURVE_fireramp', T=6.0, F=0.0004432624113475177 /
…
===========================================================
```

Additionally, the FDS code lines defining the random HRR curves include ramps for the HRR, with values recorded every second. These ramps accurately reflect the fire growth patterns specified in the FD-Gen script, ensuring consistency and alignment with the predefined fire scenarios.

(3) Fire source location FDS code lines

```
===========================================================
======= FIRE SOURCE LOCATION ==============================
===========================================================
&OBST ID='fire source', XB=2.6, 4.6, 22.8, 23.8, 0.0, 0.2,
SURF_IDS='CURVE','INERT','INERT'/
===========================================================
```

Finaly, the FDS code lines also include the randomized fire source locations. The coordinates of these fire sources follow the distribution type specified in the associated sample generators, ensuring variability across scenarios while adhering to the design rules.

5.1.2.2. Plots of the sampled fire scenarios

In addition to generating multiple FDS case input files, the output results of the example also include plots of the sampled fire parameter data. In this example, the plots exhibit the randomized HRR curves, fire source locations, and the sample data for door opening times. The random HRR curves are depicted in **Figure 12**. The random fire source locations are presented in **Figure 13**, and the CDF and PDF plots for the sample data of door opening are show in **Figure 14**.

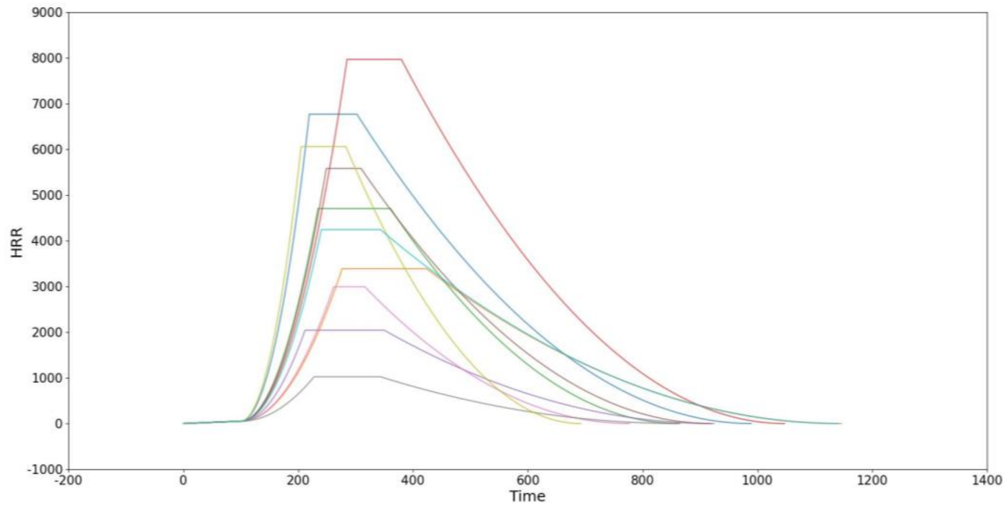


Figure 12. Random HRR curves.

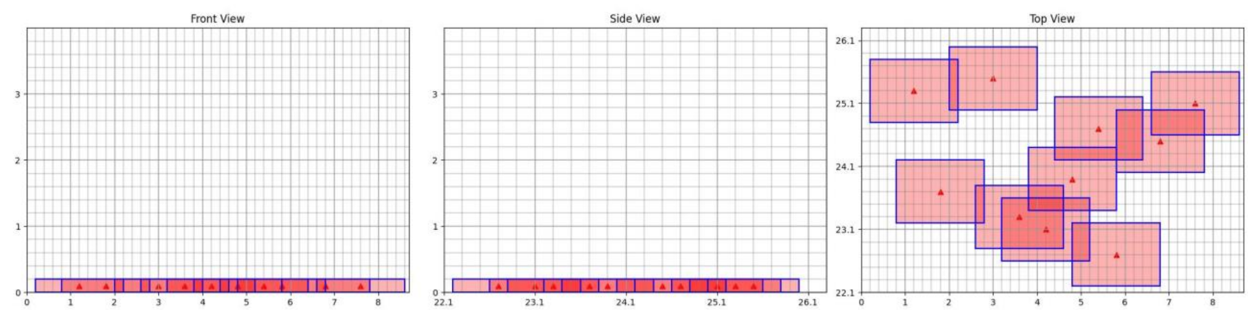
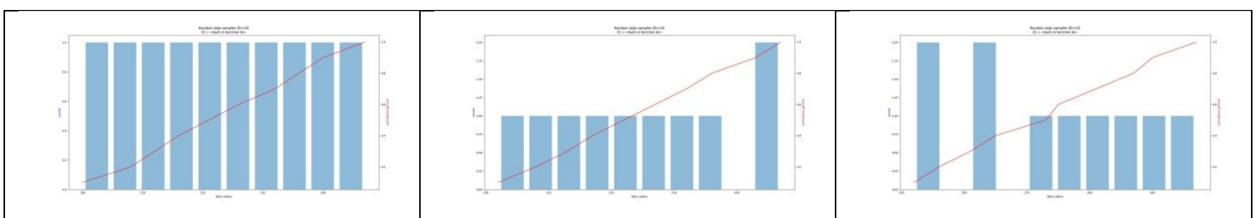


Figure 13. Random fire source locations.



'dot_timer1'	'dot_timer2'	'dot_timer3'

Figure 14. Sample data for door opening times.

5.2. CSTB Tunnel fire test

This example demonstrates an FD-Gen project designed to generate multiple random fire scenarios within a tunnel environment. The original FDS case file is based on the FDS validation case file `CSTB_Tunnel_Test_2.fds`[2], which was used to validate the FDS model against fire experiments conducted by the French building research laboratory, Centre Scientifique et Technique du Bâtiment (CSTB), between 2005 and 2008 [18]. For this example, we utilize this tunnel geometry to illustrate how FD-Gen can be used to create a variety of fire scenarios in a tunnel environment. Additionally, advanced functionalities, the use of the `PRODUCT` to generate scenarios by combining all sampled fire parameters using the Cartesian product and design of HRR curves with multiple item ignition, are tested. **Table 4** provides a detailed summary of the randomized fire parameters used in this example. **Figure 15** illustrates the geometry of the tunnel structure in Smokeview.

Table 4. Summary of fire parameters considered in the CSTB Tunnel fire scenario design.

Fire parameters		Unit	Distribution types	Min	Max	Constant	Mean(μ)	SD(σ)
Fire source location	Fire source (X_{loc})	m	Uniform	0	8.7	-	-	-
	Fire source (Y_{loc})	m	Uniform	22.1	26.3	-	-	-
	Fire source (Z_{loc})	m	Constant	-	-	0	-	-
HRR curves-item 1	Incipient time duration	s	Constant	-	-	100	-	-
	Fire growth time duration	s	Linear	200	300	-	-	-
	Fully developed time duration	s	Constant	-	-	300	-	-
	Decay time duration	s	Linear	400	600	-	-	-
	Maximum HRR (Incipient)	kW	Constant	-	-	100	-	-
	Maximum HRR (peak)	kW	Linear	3000	4000		-	-
	Minimum HRR (decay)	kW	Constant	-	-	500	-	-
HRR curves-item 2	Incipient time duration	s	Constant	-	-	100	-	-
	Fire growth time duration	s	Constant	-	-	100	-	-
	Fully developed time duration	s	Constant	-	-	50	-	-
	Decay time duration	s	Linear	400	500	-	-	-
	Maximum HRR (Incipient)	kW	Constant	-	-	500	-	-
	Maximum HRR (peak)	kW	Constant	-	-	800	-	-
	Minimum HRR (decay)	kW	Constant	-	-	0	-	-
Vent fan	volume flow	m^3/s	Linear	10	15	-	-	-

Figure 15. Building geometry of the CSTB tunnel.

[2] Retrieved from https://github.com/firemodels/fds/blob/master/Validation/CSTB_Tunnel/FDS_Input_Files/CSTB_Tunnel_Test_2.fds

5.2.1. FD-Gen code breakdown

To create the customized fire scenarios using FD-Gen in this example, a script of FD-Gen input file is provided below. The script defines all the patterns and rules that the randomized fire parameters must follow. By specifying these rules, the script ensures that the resulting FDS case input file is configured correctly for the desired fire scenarios. The attached script includes only the FD-Gen code sections relevant to the randomized fire parameters. Sections of the script unrelated to these parameters and belonging to the FDS code have been omitted here for clarity. The complete FD-Gen input file for this example is available is available in the GitHub repository's example folder for reference and use.

```
===========================================================
=========== Main INFO =====================================
&MAIN PROJECT_NAME='example_tunnel', NUMBER_OF_CASES=384, SEEDS=35/
&PFSL FSL_ID='tunnel_fire'/
&PHRC HRC_ID='BURNER'/
===========================================================
```

First, the basic information for the FD-Gen project is specified, including the project name, the number of cases to generate, and the seed number for randomization. This example includes commands to visualize two key fire parameters: `'tunnel_fire'` and `'BURNER'`. These parameters are specified using the `&PFSL` and `&PPSD` namelists, representing plotting of random fire source locations and HRR curves, respectively. Additionally, this example demonstrates the use of the `PRODUCT` functionality, which facilitates the generation of fire scenarios by applying the Cartesian product to the selected fire parameters. As a result, fire cases are systematically created by combining four fan volume flow rates, 16 distinct HRR curves, and six unique fire source locations. This approach yields a total of 384 fire scenarios, each representing a unique combination of these parameters.

```
===========================================================
======= OTHER INFORMATION (fan volume flow) ===============
# FD-Gen code
&IOTH PRE_PARM_INFO_ID='fan_volume_flow', NAMELIST='SURF',
ARG_NAME='VOLUME_FLOW', OTHERS_RANGE=10, 15/
&MRND GENERATOR_ID='Fan1', PRE_PARM_INFO_ID='fan_volume_flow',
DISTRIBUTION_TYPE='LINEAR', VALUE_TYPE='APPROX', APPROX_VALUE=0.01,
NUMBER_OF_SAMPLES=4/
&GOTH ID='FAN', PRE_PARM_INFO_ID='fan_volume_flow',
GENERATOR_ID='Fan1', PRODUCT=True/
==========================================================
# FDS code
&SURF ID='FAN', VOLUME_FLOW=12, COLOR='BLUE' /
==========================================================
```

This part of the script defines the fire parameter associated with fan volume flow. In this example, the fan's volume flow in the tunnel is randomized within a range of 10 m^3/s to 15 m^3/s. This pre-parameter information is specified using the `&IOTH` namelist, where `NAMELIST='SURF'` and `ARG_NAME='VOLUME_FLOW'` are provided to link the parameter to corresponding FDS code lines. Additionally, the sample generator is configured using the `&MRND` namelist, employing a linear distribution to generate four distinct values within the defined range. To facilitate comprehensive combinations with other fire parameters, `PRODUCT=True` is included in the `&GOTH` namelist. This ensures the sampled vent parameter is incorporated into the Cartesian product, allowing the generation of all possible fire scenario combinations. Additionally, the ID name of `&GOTH` is set as `'FAN'`, matching the ID name used in the FDS code line. Furthermore, `PRE_PARM_INFO_ID` and `GENERATOR_ID` are specified as same as `&IOTH` and `&MRND`, respectively.

```
===========================================================
=========== HRR CURVE =====================================
# FD-Gen code
# fire time duration
&IFTD PRE_PARM_INFO_ID='FTD1', INCIPIENT_TIME=100,100,
FIRE_GROWTH_TIME=200,300, PEAK_TIME=300,300, DECAY_TIME=400,600/
&MRND GENERATOR_ID='1', PRE_PARM_INFO_ID='FTD1_I',
DISTRIBUTION_TYPE='LINEAR', SHUFFLE_STATES=False, NUMBER_OF_SAMPLES=1/
&MRND GENERATOR_ID='2', PRE_PARM_INFO_ID='FTD1_G',
DISTRIBUTION_TYPE='LINEAR', SHUFFLE_STATES=False, NUMBER_OF_SAMPLES=2/
&MRND GENERATOR_ID='3', PRE_PARM_INFO_ID='FTD1_P',
DISTRIBUTION_TYPE='LINEAR', SHUFFLE_STATES=False, NUMBER_OF_SAMPLES=1/
&MRND GENERATOR_ID='4', PRE_PARM_INFO_ID='FTD1_D',
DISTRIBUTION_TYPE='LINEAR', SHUFFLE_STATES=False, NUMBER_OF_SAMPLES=2/

&IFTD PRE_PARM_INFO_ID='FTD2', INCIPIENT_TIME=100,100,
FIRE_GROWTH_TIME=100,100, PEAK_TIME=50,50, DECAY_TIME=400,500/
&MRND GENERATOR_ID='5', PRE_PARM_INFO_ID='FTD2_I',
DISTRIBUTION_TYPE='CONSTANT', NUMBER_OF_SAMPLES=1/
&MRND GENERATOR_ID='6', PRE_PARM_INFO_ID='FTD2_G',
DISTRIBUTION_TYPE='CONSTANT', NUMBER_OF_SAMPLES=1/
&MRND GENERATOR_ID='7', PRE_PARM_INFO_ID='FTD2_P',
DISTRIBUTION_TYPE='CONSTANT', NUMBER_OF_SAMPLES=1/
&MRND GENERATOR_ID='8', PRE_PARM_INFO_ID='FTD2_D',
DISTRIBUTION_TYPE='LINEAR', SHUFFLE_STATES=False, NUMBER_OF_SAMPLES=2/

&GFTD ID='FTD1', PRE_PARM_INFO_ID='FTD1',
GENERATOR_ID='1','2','3','4'/
```

```
&GFTD ID='FTD2', PRE_PARM_INFO_ID='FTD2',
GENERATOR_ID='5','6','7','8'/


# max hrr

&IMHR PRE_PARM_INFO_ID='MAX_HRR1', INCIPIENT_MAX_HRR=100,100,
PEAK_MAX_HRR=3000,4000, DECAY_MIN_HRR=500,500/

&MRND GENERATOR_ID='incipient1', PRE_PARM_INFO_ID='MAX_HRR1_I',
DISTRIBUTION_TYPE='CONSTANT', SHUFFLE_STATES=False,
NUMBER_OF_SAMPLES=1/

&MRND GENERATOR_ID='peak1', PRE_PARM_INFO_ID='MAX_HRR1_P',
DISTRIBUTION_TYPE='LINEAR', SHUFFLE_STATES=False, NUMBER_OF_SAMPLES=2/

&MRND GENERATOR_ID='decay1', PRE_PARM_INFO_ID='MAX_HRR1_D',
DISTRIBUTION_TYPE='CONSTANT', SHUFFLE_STATES=False,
NUMBER_OF_SAMPLES=1/


&IMHR PRE_PARM_INFO_ID='MAX_HRR2', INCIPIENT_MAX_HRR=500,500,
PEAK_MAX_HRR=800,800, DECAY_MIN_HRR=0,0/

&MRND GENERATOR_ID='incipient2', PRE_PARM_INFO_ID='MAX_HRR2_I',
DISTRIBUTION_TYPE='CONSTANT', NUMBER_OF_SAMPLES=1/

&MRND GENERATOR_ID='peak2', PRE_PARM_INFO_ID='MAX_HRR2_P',
DISTRIBUTION_TYPE='CONSTANT', NUMBER_OF_SAMPLES=1/

&MRND GENERATOR_ID='decay2', PRE_PARM_INFO_ID='MAX_HRR2_D',
DISTRIBUTION_TYPE='CONSTANT', NUMBER_OF_SAMPLES=1/


&GMHR ID='MAX_HRR1', PRE_PARM_INFO_ID='MAX_HRR1',
GENERATOR_ID='incipient1','peak1','decay1'/

&GMHR ID='MAX_HRR2', PRE_PARM_INFO_ID='MAX_HRR2',
GENERATOR_ID='incipient2','peak2','decay2'/


# generate the curve

&GHRC ID='BURNER', FIRE_TIME_DURATION_GENERATOR_ID='FTD1','FTD2',
MAX_HRR_GENERATOR_ID='MAX_HRR1','MAX_HRR2', FIRE_PATTERN=1,2,0,2,
END_TIME=2000, TIME_FREQUENCY=20, PRODUCT=True/

============================================================

# FDS code

&SURF ID='BURNER', HRRPUA=3930, RAMP_Q='BURNER_fireramp',
COLOR='ORANGE' /

============================================================
```

This part of the script specifies the HRR curves for this example. A more complex HRR curve is designed here, assuming that two items are ignited adjacent to each other, resulting in two observable HRR peaks. By defining two complete stages of fire development and generating the HRR curves with the corresponding FTD and MHR samples, randomized HRR curves with two peaks can be created.

For the first peak, &IFTD is specified with an incipient time lasting 100 s, a fire growth time ranging between 200 s and 300 s, a peak time of 300 s, and a decay time lasting between 400 s and 500 s. Subsequently, &MRND is defined with generator IDs 1, 2, 3, and 4, each applying a linear distribution and retrieving essential pre-parameter information from the &IFTD input. For the &IMHR input, the incipient maximum HRR is set to 100 kW, the peak maximum HRR ranges between 3000 kW and 4000 kW, and the decay minimum HRR is set to 500 kW. By utilizing &MRND to shape the sampling process of the maximum HRR, we apply a linear distribution to sample data within the given range. Additionally, by specifying NUMBER_OF_SAMPLES within &MRND to limit the number of samples, we generate and combine these samples, ultimately obtaining eight distinct curve shapes for the first peak.

Regarding the second peak, we specify the time durations for fire development as follows. the incipient stage lasts 100 s, the fire growth phase lasts 100 s, the peak stage lasts 50 s, and the decay phase lasts between 400 s and 500 s. Regarding the maximum HRR for the second peak, to ensure continuity from the first peak, the INCIPIENT_MAX_HRR is set to 500 kW, aligning with the minimum decay HRR of the first peak. Additionally, the peak maximum HRR is set to 800 kW, while the decay minimum HRR is set to 0 kW, indicating complete extinguishment of the fire. Based on the design for the second peak, two distinct curve shapes are established.

Finally, in the &GHRC namelist, FIRE_TIME_DURATION_GENERATOR_ID and MAX_HRR_GENERATOR_ID are assigned two consecutive sample generator IDs for FTD and MHR, representing the two entire stages of fire development. The FIRE_PATTERN is specified as 1,2,0,2, indicating both the curves follow the following fire behavior: a linear increase during the incipient stage, a squared increase during the growth stage, a steady value during the fully developed stage, and a squared decrease during the decay stage. The END_TIME is set to 2000 s, marking the end of the HRR curves, while TIME_FREQUENCY is specified as 20 s, indicating that HRR data values are updated every 20 s. To facilitate comprehensive combinations with other fire parameters, the option PRODUCT=True is also included. Based on the provided script, a total of 16 distinct HRR curves (8 for the first peak and 2 for the second peak) are generated. These curve samples combine with other parameters to form all possible fire case scenarios.

```
============================================================

======= FIRE SOURCE LOCATION ==============================

# FD-Gen code

&IFSL PRE_PARM_INFO_ID='fire source', XB=-17,25,-1.2,1.2,0,2,
FIRE_BURNER_SIZE=1,0.5,0.4, MESH_SIZE=0.1,0.1,0.1/

&MRND GENERATOR_ID='FS_X', PRE_PARM_INFO_ID='fire source_X',
DISTRIBUTION_TYPE='LINEAR', NUMBER_OF_SAMPLES=6/

&MRND GENERATOR_ID='FS_Y', PRE_PARM_INFO_ID='fire source_Y',
DISTRIBUTION_TYPE='CONSTANT', CONSTANT=-0.25, NUMBER_OF_SAMPLES=1/
```

```
&MRND GENERATOR_ID='FS_Z', PRE_PARM_INFO_ID='fire source_Z',
DISTRIBUTION_TYPE='CONSTANT', CONSTANT=0, NUMBER_OF_SAMPLES=1/

&GFSL ID='tunnel_fire', PRE_PARM_INFO_ID='fire source',
GENERATOR_ID='FS_X','FS_Y','FS_Z', PRODUCT=True/

===========================================================

# FDS code

&OBST ID='tunnel_fire', XB=-0.5,0.5,-0.2,0.3,0.0,0.4,
SURF_IDS='BURNER','WALL','WALL' /

===========================================================
```

The script in this section defines the randomized fire source locations. Based on the `&IFSL` and `&MRND` input, the sample space is set as random positions along the x-axis, following a linear distribution, with six distinct locations sampled within the defined range. To enable combinations with the other fire parameters (fire curve and vent fan), the `PRODUCT=True` option is also included, ensuring that every sampled location integrates seamlessly with other randomized parameters. This comprehensive approach generates all possible fire scenario combinations, resulting in 384 (4×16×6) unique fire cases and ensuring robust coverage of potential fire scenarios.

5.2.2. Example outputs

5.2.2.1. Example of created FDS case input files

(1) Vent fan FDS code lines

```
===========================================================
======= OTHER INFORMATION (fan volume flow) ===============
===========================================================
&SURF ID='FAN', VOLUME_FLOW=13.33, COLOR='BLUE' /
===========================================================
```

In the generated FDS case input files, each file features a distinct volume flow value for the "`FAN`" parameter, precisely following the value designed in the fire scenarios.

(2) HRR curve FDS code lines

```
===========================================================
======= HRR CURVE =========================================
# fire time duration - first peak
# max hrr - first peak
# generate the curve - first peak
# fire time duration - second peak
```

```
# max hrr - second peak
# generate the curve - second peak
# entire process
===========================================================
&SURF ID='BURNER', HRRPUA=3000.0, RAMP_Q='BURNER_fireramp',
COLOR='ORANGE' /

&RAMP ID='BURNER_fireramp', T=1.0, F=0.0003333333333333333 /
&RAMP ID='BURNER_fireramp', T=25.75, F=0.008583333333333333 /
&RAMP ID='BURNER_fireramp', T=50.5, F=0.016833333333333332 /
&RAMP ID='BURNER_fireramp', T=75.25, F=0.025083333333333332 /
&RAMP ID='BURNER_fireramp', T=100.0, F=0.03333333333333333 /
&RAMP ID='BURNER_fireramp', T=101.0, F=0.0333575 /
…
===========================================================
```

Regarding the FDS code lines defining the random HRR curves, distinct fire curve is integrated into the `&RAMP` input. The HRR ramps include two distinct peaks, consistent with the design outlined in the FD-Gen input file.

(3) Fire source location FDS code lines

```
===========================================================
======= FIRE SOURCE LOCATION ==============================
===========================================================
&OBST ID='tunnel_fire', XB=-8.8, -7.8, -0.2, 0.3, 0.0, 0.4,
SURF_IDS='BURNER','WALL','WALL' /
===========================================================
```

The randomized fire source locations are also included in the FDS code lines, with their coordinates adhering to the value range specified in the associated FD-Gen input file.

5.2.2.2. Plots of the sampled fire scenarios

In addition to generating multiple FDS case input files, the outputs of this example also include visualizations of the sampled fire parameter data. Specifically, the randomized fire source locations and HRR curves are presented. It is observed that the sampled HRR curves exhibit two distinct peaks as designed, as shown in **Figure 16**. Also, due to the limited number of samples defined within `&MRND`, a total of 16 distinct HRR curves are generated.

Figure 17 illustrates the sampled fire source locations. Using a linear distribution, six possible fire source locations are evenly spaced along the y-axis within the defined range. Finally, by applying the Cartesian product to combine 16 distinct HRR curves, 6 fire source locations, and 4 vent fan flow values, the program successfully generates 384 unique fire case scenarios, ensuring a comprehensive exploration of the parameter value range. Figure **18** provides a screenshot of the generated sample scenarios, demonstrating the successful implementation of the Cartesian product across all fire parameters. As a result, a comprehensive combination of fire scenarios is achieved.

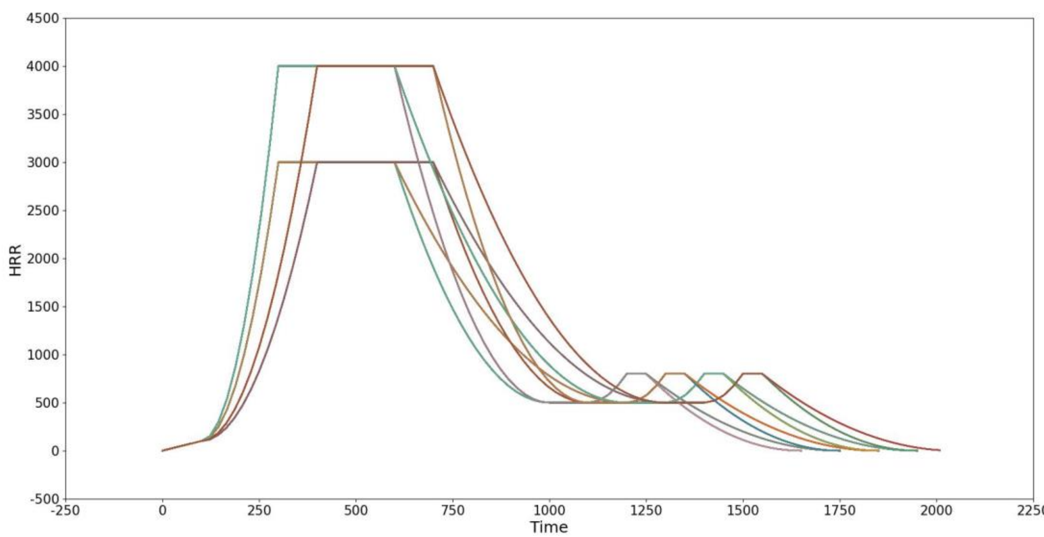


Figure 16. Sampled HRR curves.

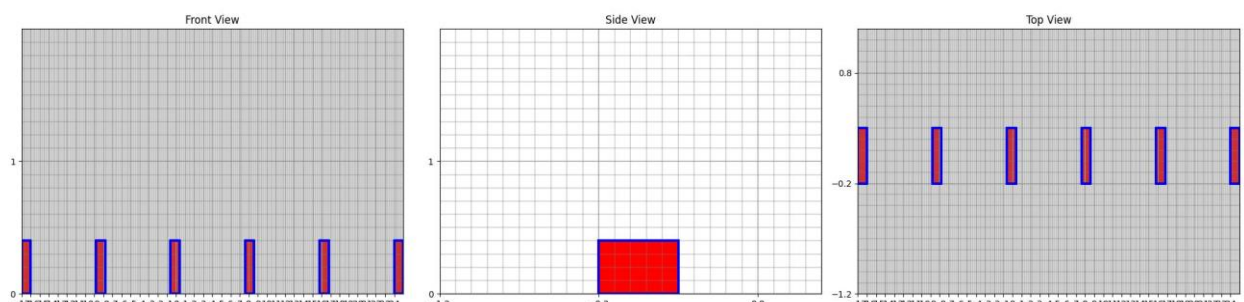


Figure 17. Sampled fire source locations.

1	Cases	OBST_ID - tunne	TH> ID - F	D - FTD1 .In	D - FTD1 .(	ID - FTD1	ID - FTD1 .	D - FTD2 .In	D - FTD2 .(	ID - FTD2	ID - FTD2 .	MAX_HRR	- MAX_H	- MAX_HR	MAX_HRR	- MAX_H	- MAX_HR	RNER .Fire	URNER .HRR samples
2	0	[-8.8, -7.8000000(	13.33	100	200	300	400	100	100	50	400	100	3000	500	500	800	0	[1.000000(	[1.00000000e+00 2.57500000e
3	1	[-8.8, -7.8000000(	13.33	100	200	300	400	100	100	50	500	100	3000	500	500	800	0	[1.000000(	[1.00000000e+00 2.57500000e
4	2	[-8.8, -7.8000000(	13.33	100	200	300	600	100	100	50	400	100	3000	500	500	800	0	[1.000000(	[1.00000000e+00 2.57500000e
5	3	[-8.8, -7.8000000(	13.33	100	200	300	600	100	100	50	500	100	3000	500	500	800	0	[1.000000(	[1.00000000e+00 2.57500000e
6	4	[-8.8, -7.8000000(	13.33	100	300	300	400	100	100	50	400	100	3000	500	500	800	0	[1.000000(	[1.00000000e+00 2.57500000e
7	5	[-8.8, -7.8000000(	13.33	100	300	300	400	100	100	50	500	100	3000	500	500	800	0	[1.000000(	[1.00000000e+00 2.57500000e
8	6	[-8.8, -7.8000000(	13.33	100	300	300	600	100	100	50	400	100	3000	500	500	800	0	[1.000000(	[1.00000000e+00 2.57500000e
9	7	[-8.8, -7.8000000(	13.33	100	300	300	600	100	100	50	500	100	3000	500	500	800	0	[1.000000(	[1.00000000e+00 2.57500000e
10	8	[-8.8, -7.8000000(	13.33	100	200	300	400	100	100	50	400	100	4000	500	500	800	0	[1.000000(	[1.00000000e+00 2.57500000e
11	9	[-8.8, -7.8000000(	13.33	100	200	300	400	100	100	50	500	100	4000	500	500	800	0	[1.000000(	[1.00000000e+00 2.57500000e
12	10	[-8.8, -7.8000000(	13.33	100	200	300	600	100	100	50	400	100	4000	500	500	800	0	[1.000000(	[1.00000000e+00 2.57500000e
13	11	[-8.8, -7.8000000(	13.33	100	200	300	600	100	100	50	500	100	4000	500	500	800	0	[1.000000(	[1.00000000e+00 2.57500000e
14	12	[-8.8, -7.8000000(	13.33	100	300	300	400	100	100	50	400	100	4000	500	500	800	0	[1.000000(	[1.00000000e+00 2.57500000e
15	13	[-8.8, -7.8000000(	13.33	100	300	300	400	100	100	50	500	100	4000	500	500	800	0	[1.000000(	[1.00000000e+00 2.57500000e
16	14	[-8.8, -7.8000000(	13.33	100	300	300	600	100	100	50	400	100	4000	500	500	800	0	[1.000000(	[1.00000000e+00 2.57500000e
17	15	[-8.8, -7.8000000(	13.33	100	300	300	600	100	100	50	500	100	4000	500	500	800	0	[1.000000(	[1.00000000e+00 2.57500000e
18	16	[-8.8, -7.8000000(	15	100	200	300	400	100	100	50	400	100	3000	500	500	800	0	[1.000000(	[1.00000000e+00 2.57500000e
19	17	[-8.8, -7.8000000(	15	100	200	300	400	100	100	50	500	100	3000	500	500	800	0	[1.000000(	[1.00000000e+00 2.57500000e
20	18	[-8.8, -7.8000000(	15	100	200	300	600	100	100	50	400	100	3000	500	500	800	0	[1.000000(	[1.00000000e+00 2.57500000e
21	19	[-8.8, -7.8000000(	15	100	200	300	600	100	100	50	500	100	3000	500	500	800	0	[1.000000(	[1.00000000e+00 2.57500000e
22	20	[-8.8, -7.8000000(	15	100	300	300	400	100	100	50	400	100	3000	500	500	800	0	[1.000000(	[1.00000000e+00 2.57500000e
23	21	[-8.8, -7.8000000(	15	100	300	300	400	100	100	50	500	100	3000	500	500	800	0	[1.000000(	[1.00000000e+00 2.57500000e
24	22	[-8.8, -7.8000000(	15	100	300	300	600	100	100	50	400	100	3000	500	500	800	0	[1.000000(	[1.00000000e+00 2.57500000e
25	23	[-8.8, -7.8000000(	15	100	300	300	600	100	100	50	500	100	3000	500	500	800	0	[1.000000(	[1.00000000e+00 2.57500000e
26	24	[-8.8, -7.8000000(	15	100	200	300	400	100	100	50	400	100	4000	500	500	800	0	[1.000000(	[1.00000000e+00 2.57500000e
27	25	[-8.8, -7.8000000(	15	100	200	300	400	100	100	50	500	100	4000	500	500	800	0	[1.000000(	[1.00000000e+00 2.57500000e
28	26	[-8.8, -7.8000000(	15	100	200	300	600	100	100	50	400	100	4000	500	500	800	0	[1.000000(	[1.00000000e+00 2.57500000e
29	27	[-8.8, -7.8000000(	15	100	200	300	600	100	100	50	500	100	4000	500	500	800	0	[1.000000(	[1.00000000e+00 2.57500000e

Figure 18. Screenshot of the spreadsheet of sampled fire scenario details.

5.3. Single-story residential room fire

This example demonstrates an FD-Gen project designed to generate multiple fire scenarios within a single-story residential room. The original FDS case file is based on the FDS validation case file `Single_Story_Gas_1.fds`[3], which is used to validate the FDS model against fire experiments in a one-story ranch-style house conducted by The Firefighter Safety Research Institute (FSRI) of Underwriters Laboratory (UL) [19, 20]. For this example, we utilize this geometry to illustrate how FD-Gen can be used to create a variety of fire scenarios in this kind of environment. Furthermore, the `HRR_FLUCTUATION` function for sampling HRR curves with a fixed variation is tested. **Table 5** provides a detailed summary of the randomized fire parameters used in this example. The floor plan of the building structure is illustrated in **Figure 19**.

Table 5. Summary of fire parameters considered in the residential fire scenario design.

Fire parameters		Unit	Distribution types	Min	Max	Constant	Mean(μ)	SD(σ)
Fire source location	Fire source (X_{loc})	m	Uniform	3.8	9.2	-	-	-
	Fire source (Y_{loc})	m	Uniform	0.1	3.8	-	-	-
	Fire source (Z_{loc})	m	Constant	-	-	0.1	-	-
HRR curves	Incipient time duration	s	Constant	-	-	0	-	-
	Fire growth time duration	s	Constant	-	-	200	-	-
	Fully developed time duration	s	Constant	-	-	300	-	-
	Decay time duration	s	Constant	-	-	100	-	-
	Maximum HRR (Incipient)	kW	Constant	-	-	0	-	-

[3]Retrieved from: https://github.com/firemodels/fds/blob/master/Validation/UL_NIJ_Houses/FDS_Input_Files/Single_Story_Gas_1.fds

	Maximum HRR (peak)	kW	Constant	-	-	2500	-	-
	Minimum HRR (decay)	kW	Constant	-	-	0	-	-
Door and window open time	'FRONT DOOR TIMER'	s	Uniform	300	350	-	-	-
	'KITCH DOOR TIMER'	s	Uniform	1200	1350	-	-	-
	'BR1 WINDOW TIMER'	s	Uniform	600	700	-	-	-
	'BR1 WIND 2 TIMER'	s	Uniform	1200	1350	-	-	-
	'BR2 WINDOW TIMER'	s	Uniform	1200	1350	-	-	-
	'BR3 WINDOW TIMER'	s	Uniform	1200	1350	-	-	-
	'KIT WINDOW TIMER'	s	Uniform	1200	1350	-	-	-
	'DR WINDOW TIMER'	s	Uniform	1200	1350	-	-	-
	'LR WINDOW TIMER'	s	Uniform	1200	1350	-	-	-

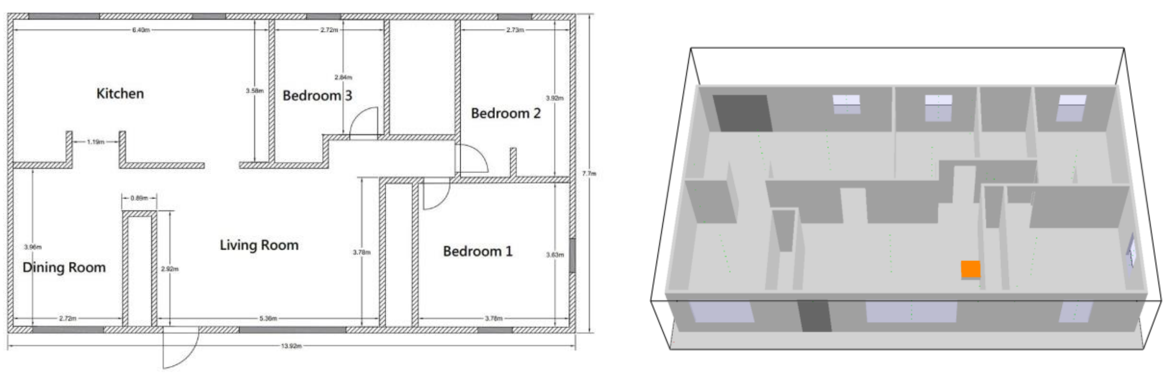


Figure 19. Building geometry of the residential structure.

5.3.1. FD-Gen code breakdown

To create the customized fire scenarios using FD-Gen in this example, a script of FD-Gen input file is provided below. The script defines all the patterns and rules that the randomized fire parameters must follow. By specifying these rules, the script ensures that the resulting FDS case input file is configured correctly for the desired fire scenarios. The attached script includes only the FD-Gen code sections relevant to the randomized fire parameters. Sections of the script unrelated to these parameters and belonging to the FDS code have been omitted here for clarity. The complete FD-Gen input file for this example is available in the GitHub repository's example folder for reference and use.

```
==============================================================
=========== Main INFO ======================================
```

```
&MAIN PROJECT_NAME='example_single_story', NUMBER_OF_CASES=100,
SEEDS=20/

&PFSL FSL_ID='room1'/

&PHRC HRC_ID='BURNER'/

&PPSD PSD_ID='DOT_3','DOT_1'/
===========================================================
```

First, the basic information for the FD-Gen project is specified, including the project name, the number of cases to generate, and the seed number for randomization. Additionally, this example includes commands to plot two fire parameters: `'room1'` and `' BURNER'`. These parameters are specified using the `&PFSL` and `&PPSD` namelists, representing random fire source locations and HRR curves, respectively. Furthermore, the pattern of sampled data in the sample generators `'DOT_3' and 'DOT_1'` is also specified, providing a visual representation of the selected sample data for door open time.

```
===========================================================

=========== HRR CURVE =====================================

#FD-Gen code

# fire time duration

&IFTD PRE_PARM_INFO_ID='FTD', INCIPIENT_TIME=0,0,
FIRE_GROWTH_TIME=200,200, PEAK_TIME=300,300, DECAY_TIME=100,100/

&MRND GENERATOR_ID='FG1', PRE_PARM_INFO_ID='FTD_I',
DISTRIBUTION_TYPE='CONSTANT'/

&MRND GENERATOR_ID='FG2', PRE_PARM_INFO_ID='FTD_G',
DISTRIBUTION_TYPE='CONSTANT'/

&MRND GENERATOR_ID='FG3', PRE_PARM_INFO_ID='FTD_P',
DISTRIBUTION_TYPE='CONSTANT'/

&MRND GENERATOR_ID='FG4', PRE_PARM_INFO_ID='FTD_D',
DISTRIBUTION_TYPE='CONSTANT'/

&GFTD ID='FTD', PRE_PARM_INFO_ID='FTD',
GENERATOR_ID='FG1','FG2','FG3','FG4'/

# max hrr

&IMHR PRE_PARM_INFO_ID='MAX_HRR', INCIPIENT_MAX_HRR=0,0,
PEAK_MAX_HRR=2500,2500, DECAY_MIN_HRR=0,0/

&MRND GENERATOR_ID='incipient', PRE_PARM_INFO_ID='MAX_HRR_I',
DISTRIBUTION_TYPE='CONSTANT'/

&MRND GENERATOR_ID='peak', PRE_PARM_INFO_ID='MAX_HRR_P',
DISTRIBUTION_TYPE='CONSTANT'/

&MRND GENERATOR_ID='decay', PRE_PARM_INFO_ID='MAX_HRR_D',
DISTRIBUTION_TYPE='CONSTANT'/

&GMHR ID='MAX_HRR', PRE_PARM_INFO_ID='MAX_HRR',
GENERATOR_ID='incipient','peak','decay'/
```

```
# generate the curve

&GHRC ID='BURNER', FIRE_TIME_DURATION_GENERATOR_ID='FTD',
MAX_HRR_GENERATOR_ID='MAX_HRR', TIME_FREQUENCY=10, END_TIME=600,
FIRE_PATTERN=1,2,0,2, HRR_FLUCTUATION=True, HRR_FLUCTUATION_RATE=0.4/

===========================================================

#FDS code

&SURF ID='BURNER', HRRPUA=694., COLOR='ORANGE', RAMP_Q='burner' /

===========================================================
```

This part of the script outlines the design of HRR curves for this example, with a focus on testing the `HRR_FLUCTUATION` functionality. Based on the script, a fixed fire time duration is established for fire development using `&IFTD`, `&MRND`, and `&GFTD` commands. The time durations for each stage of the fire are defined as follows: the incipient stage lasts 0 s, the growth stage extends for 200 s, the fully developed stage lasts 300 s, and the decay stage continues for 100 s. Regarding the maximum HRR, a fixed randomness is applied, with values set as follows: 0 kW for the incipient stage, 2500 kW at peak, and 0 kW for the decay stage. This configuration establishes a fixed fire development by omitting the incipient stage, allowing the fire to rapidly reach the fully developed phase, sustain for a designated period, and then decay. The fire curve design represents a simplified burning process commonly observed in combustible items, ensuring a structured yet representative fire scenario.

Afterward, within the `&GHRC` namelist, the `HRR_FLUCTUATION` feature is enabled to generate multiple HRR curves by applying a fixed variation to the baseline HRR curve. This variation adjusts the baseline curve either upward or downward by a specified percentage. When `HRR_FLUCTUATION` is set to True, the HRR ramp design must consist of a single, well-defined fire curve. Therefore, in this example, the sample generators for `FTD` and `MHR` are configured with constant values to define a single exact HRR ramp. Within the `&GHRC` namelist, `FIRE_TIME_DURATION_GENERATOR_ID` and `MAX_HRR_GENERATOR_ID` are assigned with predefined sample data `'FTD'` and `'MAX_HRR'`. The `END_TIME` is set to 600 s, and `TIME_FREQUENCY` is specified as 10 s, meaning that HRR data values are updated every 10 s throughout the 600-s duration. The fire behavior is further specified using the `FIRE_PATTERN`, defined as `1,2,0,2`. This pattern indicates the following fire behavior: a linear increase during the incipient stage, a squared increase during the growth stage, a steady value during the fully developed stage, and a squared decrease during the decay stage. Finally, `HRR_FLUCTUATION` is set to `True`, with the `HRR_FLUCTUATION_RATE` given as 0.4, indicating that the sampled HRR curves will be shifted upward or downward by a maximum of 40%.

```
===========================================================

======= FIRE SOURCE LOCATION ==============================

#FD-Gen code

&IFSL PRE_PARM_INFO_ID='fire source', XB=3.8,9.2,0.1,3.8,0,0.2,
FIRE_BURNER_SIZE=0.6,0.6,0.2, MESH_SIZE=0.1,0.1,0.1/

&MRND GENERATOR_ID='FS_X', PRE_PARM_INFO_ID='fire source_X',
DISTRIBUTION_TYPE='UNIFORM'/
```

```
&MRND GENERATOR_ID='FS_Y', PRE_PARM_INFO_ID='fire source_Y',
DISTRIBUTION_TYPE='UNIFORM'/

&MRND GENERATOR_ID='FS_Z', PRE_PARM_INFO_ID='fire source_Z',
DISTRIBUTION_TYPE='CONSTANT'/

&GFSL ID='room1', PRE_PARM_INFO_ID='fire source',
GENERATOR_ID='FS_X','FS_Y','FS_Z'/

============================================================

#FDS code

&OBST ID='room1', XB= 8.7,9.3,1.6,2.2,0.4,0.6,
SURF_IDS='BURNER','WALL','WALL' /

============================================================
```

This section of the script defines the fire source location setup. Using the `&IFSL` namelist, the script specifies random fire source locations within the living room of the structure. The sample space, as defined in `&IFSL`, is set to the living room's coordinates, ensuring that fire sources are positioned randomly within this area. Within the sample generators, uniform distributions are applied to determine the burner's location along the room's length and width, while the burner's height is fixed at floor level. Finally, using the `&GFSL` namelist, the sampled fire source location data is generated and integrated into the FDS case input file during the file-wrapping process.

```
============================================================

======= OTHER INFORMATION (Door open time) ================

#FD-Gen code

# for FRONT DOOR

&IOTH PRE_PARM_INFO_ID='setpoint_value1', NAMELIST='DEVC',
ARG_NAME='SETPOINT', OTHERS_RANGE=300, 350/

&MRND GENERATOR_ID='DOT_1', PRE_PARM_INFO_ID='setpoint_value1',
DISTRIBUTION_TYPE='UNIFORM'/

&GOTH ID='FRONT DOOR TIMER',  PRE_PARM_INFO_ID='setpoint_value1',
GENERATOR_ID='DOT_1'/

# for other doors

&IOTH PRE_PARM_INFO_ID='setpoint_value2', NAMELIST='DEVC',
ARG_NAME='SETPOINT', OTHERS_RANGE=1200, 1350/

&MRND GENERATOR_ID='DOT_2', PRE_PARM_INFO_ID='setpoint_value2',
DISTRIBUTION_TYPE='UNIFORM'/

&GOTH ID='KITCH DOOR TIMER',  PRE_PARM_INFO_ID='setpoint_value2',
GENERATOR_ID='DOT_2'/

&GOTH ID='BR1 WIND 2 TIMER',  PRE_PARM_INFO_ID='setpoint_value2',
GENERATOR_ID='DOT_2'/

&GOTH ID='BR2 WINDOW TIMER',  PRE_PARM_INFO_ID='setpoint_value2',
GENERATOR_ID='DOT_2'/
```

```
&GOTH ID='BR3 WINDOW TIMER',  PRE_PARM_INFO_ID='setpoint_value2',
GENERATOR_ID='DOT_2'/

&GOTH ID='KIT WINDOW TIMER',  PRE_PARM_INFO_ID='setpoint_value2',
GENERATOR_ID='DOT_2'/

&GOTH ID='DR  WINDOW TIMER',  PRE_PARM_INFO_ID='setpoint_value2',
GENERATOR_ID='DOT_2'/

&GOTH ID='LR  WINDOW TIMER',  PRE_PARM_INFO_ID='setpoint_value2',
GENERATOR_ID='DOT_2'/

# for BR1 WINDOW

&IOTH PRE_PARM_INFO_ID='setpoint_value3', NAMELIST='DEVC',
ARG_NAME='SETPOINT', OTHERS_RANGE=600, 700/

&MRND GENERATOR_ID='DOT_3', PRE_PARM_INFO_ID='setpoint_value3',
DISTRIBUTION_TYPE='UNIFORM'/

&GOTH ID='BR1 WINDOW TIMER',  PRE_PARM_INFO_ID='setpoint_value3',
GENERATOR_ID='DOT_3'/

==========================================================

#FDS code

&DEVC ID='FRONT DOOR TIMER', QUANTITY='TIME', XYZ=-0.1,-0.4,0.1,
SETPOINT= 300.0, INITIAL_STATE=.TRUE. /

&DEVC ID='KITCH DOOR TIMER', QUANTITY='TIME', XYZ=-0.1,-0.4,0.1,
SETPOINT=1320.0, INITIAL_STATE=.TRUE. /

&DEVC ID='BR1 WINDOW TIMER', QUANTITY='TIME', XYZ=-0.1,-0.4,0.1,
SETPOINT= 600.0, INITIAL_STATE=.TRUE. /

&DEVC ID='BR1 WIND 2 TIMER', QUANTITY='TIME', XYZ=-0.1,-0.4,0.1,
SETPOINT=1220.0, INITIAL_STATE=.TRUE. /

&DEVC ID='BR2 WINDOW TIMER', QUANTITY='TIME', XYZ=-0.1,-0.4,0.1,
SETPOINT=1230.0, INITIAL_STATE=.TRUE. /

&DEVC ID='BR3 WINDOW TIMER', QUANTITY='TIME', XYZ=-0.1,-0.4,0.1,
SETPOINT=1250.0, INITIAL_STATE=.TRUE. /

&DEVC ID='KIT WINDOW TIMER', QUANTITY='TIME', XYZ=-0.1,-0.4,0.1,
SETPOINT=1260.0, INITIAL_STATE=.TRUE. /

&DEVC ID='DR  WINDOW TIMER', QUANTITY='TIME', XYZ=-0.1,-0.4,0.1,
SETPOINT=1275.0, INITIAL_STATE=.TRUE. /

&DEVC ID='LR  WINDOW TIMER', QUANTITY='TIME', XYZ=-0.1,-0.4,0.1,
SETPOINT=1200.0, INITIAL_STATE=.TRUE. /

==========================================================
```

This section of the script describes the configuration for the window break and door opening times. Unlike scenarios requiring reassignment of door positions to enable multiple stages of door operation, this example we employ `OTH` inputs instead of `DWT` inputs. The `&IOTH` is configured with `NAMELIST='DEVC'` and `ARG_NAME='SETPOINT'` to directly link to the

FDS code lines that need modification. This approach simplifies the configuration process by directly assigning opening times without additional positional definitions.

In this example, multiple doors and windows are configured for controlled opening events. First, the random opening time for the front door is addressed. The front door is set to open between approximately 300s and 350s. To achieve this, the FD-Gen script is written to sample the data for this parameter. The `&IOTH` namelist defines the range for the data, and a uniform distribution is specified using the `&MRND` namelist. The designed sample generator is then assigned in the `&GOTH` namelist to generate the sample data specifically for the front door. To ensure proper linkage, the `&GOTH ID` name matches the front door `&DEVC ID` name in the FDS code, enabling seamless integration of the sampled data into the FDS case input file. Subsequently, sample generator labeled as `'DOT_2'` is designed to sample data within the time range of 1200 s to 1350 s, representing the latest possible opening event time. In this specified time frame, components such as the kitchen door, bedroom 1 window, and bedroom 2 window, among others, may experience opening events. This enables the use of a single sample generator to sample data for multiple fire parameters, streamlining the configuration process. To implement this, the `&GOTH` namelist is employed to assign unique IDs to the various doors and windows while maintaining the same `PRE_PARM_INFO_ID='setpoint_value2'` and `GENERATOR_ID='DOT_2'`. This ensures consistency and efficiency in generating the required sample data for these parameters. Finally, `'setpoint_value3'` is set with a range of 600 s to 700 s. The sample generator based on this range is especially for data samples of `'BR1 WINDOW TIMER'`.

Finally, these settings are processed by the program, replacing the setpoint values for `&DEVC` in FDS to control the random opening of doors and windows.

5.3.2. Example outputs

5.3.2.1. Example of created FDS case input files

(1) HRR curve FDS code lines

```
===========================================================
=========== HRR CURVE =====================================
# max hrr
# generate the curve
===========================================================
&SURF ID='BURNER', HRRPUA=1702.020202020219, COLOR='ORANGE',
RAMP_Q='BURNER_fireramp', /


&RAMP ID='BURNER_fireramp', T=1.0, F=2.5e-05 /
```

```
&RAMP ID='BURNER_fireramp', T=11.473684210526315,
F=0.003291135734072021б /

&RAMP ID='BURNER_fireramp', T=21.94736842105263,
F=0.012042174515235456 /

&RAMP ID='BURNER_fireramp', T=32.421052631578945,
F=0.026278116343490303 /

&RAMP ID='BURNER_fireramp', T=42.89473684210526, F=0.04599896121883656
/

&RAMP ID='BURNER_fireramp', T=53.368421052631575,
F=0.07120470914127423 /

&RAMP ID='BURNER_fireramp', T=63.84210526315789, F=0.10189536011080332
/

&RAMP ID='BURNER_fireramp', T=74.3157894736842, F=0.13807091412742378
/
```

…

In the generated FDS case input files, the FDS code lines define the detailed ramps for the HRR curve as designed, ensuring accurate representation of the fire development process.

(2) Fire source location FDS code lines

```
===========================================================

======= FIRE SOURCE LOCATION ===============================

===========================================================

&OBST ID='room1', XB=7.2, 7.8, 2.1, 2.7, 0.0, 0.2,
SURF_IDS='BURNER','WALL','WALL' /

===========================================================
```

Additionally, each file features a unique fire source location, ensuring variability across the scenarios.

(3) door and window open FDS code lines

```
===========================================================

======= OTHER INFORMATION (Door open time) ================

===========================================================

&DEVC ID='FRONT DOOR TIMER', QUANTITY='TIME', XYZ=-0.1,-0.4,0.1,
SETPOINT=307.0,  INITIAL_STATE=.TRUE. /

&DEVC ID='KITCH DOOR TIMER', QUANTITY='TIME', XYZ=-0.1,-0.4,0.1,
SETPOINT=1292.0, INITIAL_STATE=.TRUE. /

&DEVC ID='BR1 WINDOW TIMER', QUANTITY='TIME', XYZ=-0.1,-0.4,0.1,
SETPOINT=644.0,  INITIAL_STATE=.TRUE. /

&DEVC ID='BR1 WIND 2 TIMER', QUANTITY='TIME', XYZ=-0.1,-0.4,0.1,
SETPOINT=1250.0, INITIAL_STATE=.TRUE. /
```

```
&DEVC ID='BR2 WINDOW TIMER', QUANTITY='TIME', XYZ=-0.1,-0.4,0.1,
SETPOINT=1231.0, INITIAL_STATE=.TRUE. /
&DEVC ID='BR3 WINDOW TIMER', QUANTITY='TIME', XYZ=-0.1,-0.4,0.1,
SETPOINT=1317.0, INITIAL_STATE=.TRUE. /
&DEVC ID='KIT WINDOW TIMER', QUANTITY='TIME', XYZ=-0.1,-0.4,0.1,
SETPOINT=1204.0, INITIAL_STATE=.TRUE. /
&DEVC ID='DR  WINDOW TIMER', QUANTITY='TIME', XYZ=-0.1,-0.4,0.1,
SETPOINT=1283.0, INITIAL_STATE=.TRUE. /
&DEVC ID='LR  WINDOW TIMER', QUANTITY='TIME', XYZ=-0.1,-0.4,0.1,
SETPOINT=1269.0, INITIAL_STATE=.TRUE. /
=========================================================
```

Furthermore, the door and window opening times for multiple doors and windows are accurately represented. The setpoint values for these events adhere precisely to the distribution type defined in the corresponding FD-Gen input script.

5.3.2.2. Plots of the sampled fire scenarios

In addition to generating multiple FDS case input files, the output results of this example also include visualizations of the sampled fire parameter data. Specifically, the randomized fire source locations and HRR curves are presented. **Figure 20** illustrates the HRR curves generated from the input script. The sampled HRR curves are observed to fluctuate within a specific range, demonstrating the functionality of `HRR_FLUCTUATION=True`. **Figure 21** illustrates the sampled fire source locations. All fire source locations are positioned randomly in the given room area. Finally, as depicted in **Figure 22**, the CDF and PDF plots for `'DOT_1' and 'DOT_3'` indicate that the sampled data follows a uniform distribution, consistent with the distribution type defined in the input file.

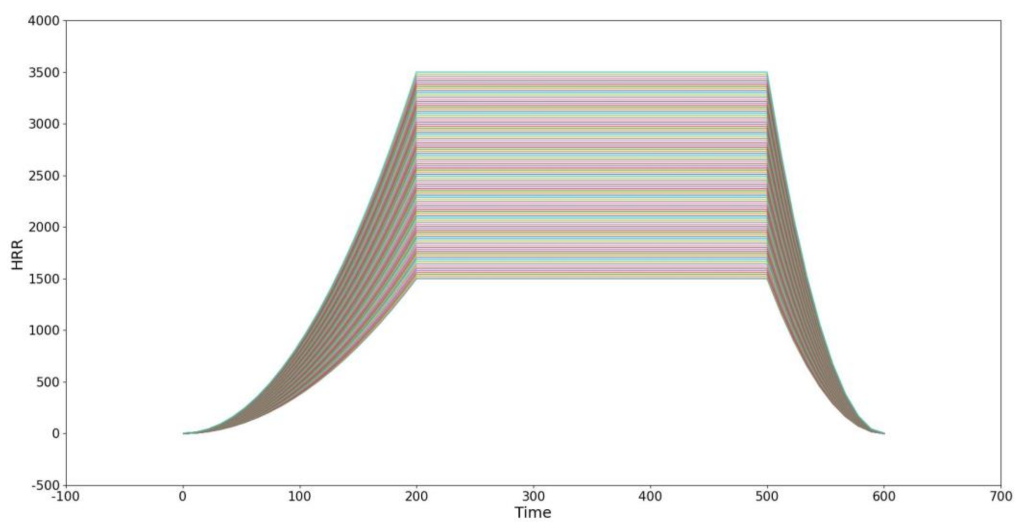


Figure 20. Random HRR curves.

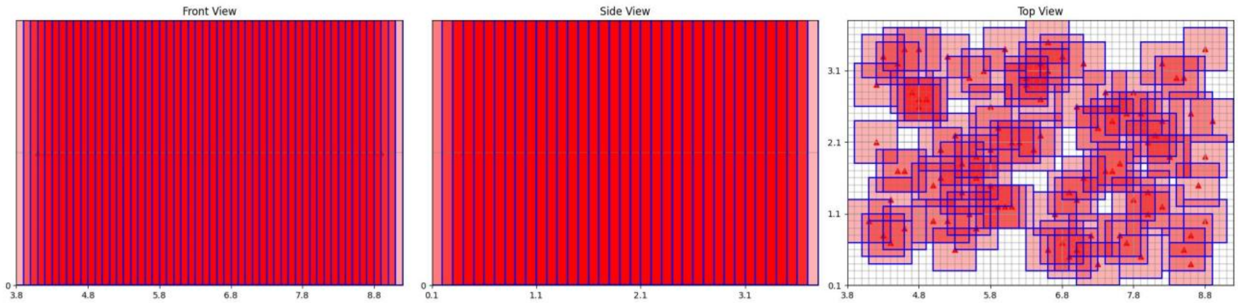


Figure 21. Random fire source locations.

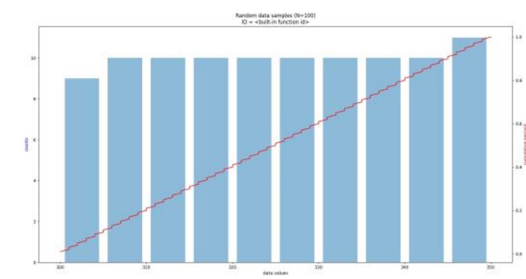
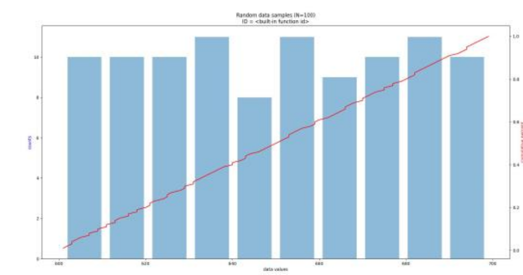
	
'DOT_1'	'DOT_3'

Figure 22. Sample data for door open and window break time.

6. Summary

This publication introduces the Fire Data Generator (FD-Gen), a Python-based automated tool developed to streamline the creation of multiple Fire Dynamics Simulator (FDS) case input files. By employing Monte Carlo methods, FD-Gen assigns random values to key fire parameters, allowing users to efficiently create a large set of fire scenarios within FDS case files. The documentation provides a comprehensive overview, including a background, general introduction, definitions of various namelists used, operation procedure, and detailed examples.

In the current version of FD-Gen, various fire parameters can be specified, including fire source locations, vent locations, multi-stage door and window opening times, obstruction sizes, fire HRR curves, and other relevant details. Regarding sampling methods, FD-Gen supports multiple distribution types, including constant, linear, uniform, triangular, normal, and lognormal distributions. Additionally, FD-Gen offers unique functionalities such as defining different stages of door openings, generating HRR curves for multiple item ignitions, and applying the Cartesian product to fire parameter combinations. These features enable diverse fire parameter randomization patterns, allowing for flexible and realistic fire scenario design. Furthermore, FD-Gen streamlines the FDS input file generation process through its automated file-wrapping mechanism, ensuring that the generated fire scenarios are accurately written into and seamlessly integrated with the FDS case input files.

The development of FD-Gen streamlines the process of generating multiple FDS case input files for various fire scenarios, significantly enhancing the efficiency of constructing numerical databases for machine learning model development in specific fire engineering problems. Furthermore, this tool holds the potential for expansion into applications such as precise fire risk assessments and fire investigations, thereby broadening its scope and utility in fire safety engineering and research. The next steps involve providing ongoing support, addressing software issues if they arise, and actively monitoring the types of issues users apply the program to solve. The feedback for the users will guide future improvements and feature enhancements.

To provide feedback or comments on using FD-Gen, users can use issue tracker on the FD-Gen GitHub repository page for reporting bugs, suggesting improvements, or seeking support.

References

[1] Fang H, Xu M, Zhang B, Lo SM (2023) Enabling fire source localization in building fire emergencies with a machine learning-based inverse modeling approach. *Journal of Building Engineering* 78:107605. https://doi.org/https://doi.org/10.1016/j.jobe.2023.107605

[2] Wu X, Park Y, Li A, Huang X, Xiao F, Usmani A (2021) Smart Detection of Fire Source in Tunnel Based on the Numerical Database and Artificial Intelligence. *Fire Technology* 57(2):657-682. https://doi.org/10.1007/s10694-020-00985-z

[3] Tam WC, Fu EY, Li J, Peacock R, Reneke P, Ngai G, Leong HV, Cleary T, Huang MX (2023) Real-time flashover prediction model for multi-compartment building structures using attention based recurrent neural networks. *Expert Systems with Applications* 223:119899. https://doi.org/https://doi.org/10.1016/j.eswa.2023.119899

[4] Tong Q, William Stroup D, Cheong Tam W, Fang H (2024) Detecting Firefighter's Thermal Risks in a Commercial Building Structure Using Machine Learning. *Journal of Physics: Conference Series* 2885(1):012082. https://doi.org/10.1088/1742-6596/2885/1/012082

[5] Wu XQ, Zhang XN, Huang XY, Xiao F, Usmani A (2022) A real-time forecast of tunnel fire based on numerical database and artificial intelligence. *Build Simul-China* 15(4):511-524. https://doi.org/10.1007/s12273-021-0775-x

[6] Wang J, Tam WC, Jia Y, Peacock R, Reneke P, Fu EY, Cleary T (2021) P-Flash – A machine learning-based model for flashover prediction using recovered temperature data. *Fire Safety Journal* 122:103341. https://doi.org/https://doi.org/10.1016/j.firesaf.2021.103341

[7] Ji W, Li G-Q, Zhu S, Li J, Qi H, Wang Y (2024) Machine learning-driven real-time identification of large-space building fires and forecast of temperature development. *Expert Systems with Applications* 255:124758. https://doi.org/https://doi.org/10.1016/j.eswa.2024.124758

[8] Kou L, Wang X, Guo X, Zhu J, Zhang H (2021) Deep learning based inverse model for building fire source location and intensity estimation. *Fire Safety Journal* 121:103310. https://doi.org/https://doi.org/10.1016/j.firesaf.2021.103310

[9] Fan L, Tam WC, Tong Q, Fu EY, Liang T (2023) An explainable machine learning based flashover prediction model using dimension-wise class activation map. *Fire Safety Journal* 140:103849. https://doi.org/https://doi.org/10.1016/j.firesaf.2023.103849

[10] Peacock R, Jones W, Reneke P, Forney G (2005) CFAST: Consolidated Model of Fire Growth and Smoke Transport (Version 6). User's Guide. (Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD).

[11] McGrattan K, McDermott R, Weinschenk C, Forney G (2013) Fire Dynamics Simulator Users Guide, Sixth Edition. (Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD).

[12] Tam WC, Fu EY, Peacock R, Reneke P, Wang J, Li J, Cleary T (2023) Generating Synthetic Sensor Data to Facilitate Machine Learning Paradigm for Prediction of Building Fire Hazard. *Fire Technology* 59(6):3027-3048. https://doi.org/10.1007/s10694-020-01022-9

[13] Fang H, Lo SM, Zhang Y, Shen Y (2021) Development of a machine-learning approach for identifying the stages of fire development in residential room fires. *Fire Safety Journal* 126:103469. https://doi.org/https://doi.org/10.1016/j.firesaf.2021.103469

[14] Wu X, Zhang X, Jiang Y, Huang X, Huang GGQ, Usmani A (2022) An intelligent tunnel firefighting system and small-scale demonstration. *Tunnelling and Underground Space Technology* 120:104301. https://doi.org/https://doi.org/10.1016/j.tust.2021.104301

[15] Reneke PA, Peacock RD, Gilbert SW, Cleary T (2021) CFAST - Consolidated Fire and Smoke Transport (Version 7) Volume 5: CFAST Fire Data Generator (CData). (Technical Note (NIST TN), National Institute of Standards and Technology, Gaithersburg, MD).

[16] Raychaudhuri S (2008) Introduction to monte carlo simulation. *2008 Winter simulation conference*, (IEEE), pp 91-100.

[17] Shields MD , Zhang J (2016) The generalization of Latin hypercube sampling. *Reliability Engineering & System Safety* 148:96-108.

[18] Blanchard E, Boulet P, Fromy P, Desanghere S, Carlotti P, Vantelon JP, Garo JP (2014) Experimental and Numerical Study of the Interaction Between Water Mist and Fire in an Intermediate Test Tunnel. *Fire Technology* 50(3):565-587. https://doi.org/10.1007/s10694-013-0323-z

[19] McKinnon MB , Weinschenk C (2021) Validation of CFD fire model pressure predictions for modern residential style structures. *Fire Safety Journal* 126:103466. https://doi.org/https://doi.org/10.1016/j.firesaf.2021.103466

[20] Madrzykowski D , Weinschenk C (2019) Impact of fixed ventilation on fire damage patterns in full-scale structures. *National Criminal Justice Reference Service* 252831.

Appendix A. FD-Gen Input namelist index

&MAIN (Basic information)

Parameter	Type	Reference	Units	Default value
PROJECT_NAME	Character	Section 3.1.1		
NUMBER_OF_CASES	Integer	Section 3.1.1		
SEEDS	Integer	Section 3.1.1		1

&PFSL (Fire source location plotting)

Parameter	Type	Reference	Units	Default value
FSL_ID	Character	Section 3.1.2		
SKIP_CASE	Integer	Section 3.1.2		1

&PHRC (HRR curve plotting)

Parameter	Type	Reference	Units	Default value
HRC_ID	Character	Section 3.1.3		
SKIP_CASE	Integer	Section 3.1.3		1

&PPSD (Sample generator data plotting)

Parameter	Type	Reference	Units	Default value
PSD_ID	Character	Section 3.1.4		
SKIP_CASE	Integer	Section 3.1.4		1

&MRND (Sample generator design)

Parameter	Type	Reference	Units	Default value
GENERATOR_ID	Character	Section 3.2.1		
PRE_PARM_INFO_ID	Character	Section 3.2.1		
FYI	Character	Section 3.2.1		
DISTRIBUTION_TYPE	Character	Section 3.2.1		
RANDOM_SEEDS	Integer	Section 3.2.1		Global seed number
CONSTANT	Real	Section 3.2.1		
MINIMUM	Real	Section 3.2.1		
PEAK	Real	Section 3.2.1		
MAXIMUM	Real	Section 3.2.1		
MEAN	Real	Section 3.2.1		
STDEV	Real	Section 3.2.1		
INCREASE	Real	Section 3.2.1		
VALUE_TYPE	Character	Section 3.2.1		'APPROX'
APPROX_VALUE	Real	Section 3.2.1		0.01
LOGICAL_VALUE	Real	Section 3.2.1		

SHUFFLE_STATES	Logical	Section 3.2.1		True
SHUFFLE_RANDOM_SEEDS	Integer	Section 3.2.1		Global seed number
NUMBER_OF_SAMPLES	Integer	Section 3.2.1		Global sample number
SAMPLES	Real array	Section 3.2.1		

&IFSL (Inputting pre-parameter information of fire source location)

Parameter	Type	Reference	Units	Default value
PRE_PARM_INFO_ID	Character	Section 3.3.1		Input for &MRND: PRE_PARM_INFO_ID+'_X'; PRE_PARM_INFO_ID+'_Y'; PRE_PARM_INFO_ID+'_Z'
XB	Real array	Section 3.3.1	m	
FIRE_BURNER_SIZE	Real array	Section 3.3.1	m	
FYI	Character	Section 3.3.1		
MESH_SIZE	Real array	Section 3.3.1	m	

&GFSL (Generating fire source location code lines)

Parameter	Type	Reference	Units	Default value
ID	Character	Section 3.3.2		
PRE_PARM_INFO_ID	Character	Section 3.3.2		
GENERATOR_ID	Character array	Section 3.3.2		
PRODUCT	Logical	Section 3.3.2		False
FYI	Character	Section 3.3.2		

&IVTP (Inputting pre-parameter information of vent position)

Parameter	Type	Reference	Units	Default value
PRE_PARM_INFO_ID	Character	Section 3.4.1		Input for &MRND: PRE_PARM_INFO_ID+'_X'; PRE_PARM_INFO_ID+'_Y'; PRE_PARM_INFO_ID+'_Z'
PLANE	Character	Section 3.4.1		
MOVE_RANGE	Real array	Section 3.4.1	m	

VENT_SIZE	Real array	Section 3.4.1	m	
FYI	Character	Section 3.4.1		
MESH_SIZE	Real array	Section 3.4.1	m	

&GVTP (Generating vent position code lines)

Parameter	Type	Reference	Units	Default value
ID	Character	Section 3.4.2		
PRE_PARM_INFO_ID	Character	Section 3.4.2		
GENERATOR_ID	Character array	Section 3.4.2		
PRODUCT	Logical	Section 3.4.2		False
FYI	Character	Section 3.4.2		

&IDWT (Inputting pre-parameter information of door open time)

Parameter	Type	Reference	Units	Default value
PRE_PARM_INFO_ID	Character	Section 3.5.1		Input for &MRND: PRE_PARM_INFO_ID+'_1'; PRE_PARM_INFO_ID+'_2'; …
OBST_ID	Character	Section 3.5.1		
XB	Real array	Section 3.5.1	m	
OPEN_DIRECTION	Character	Section 3.5.1		
QUANTITY_NAME	Character	Section 3.5.1		
PORTION	Real array	Section 3.5.1		
DEVC_ID	Character	Section 3.5.1		
DOOR_OR_WINDOW_OPEN_TIME_RANGE	Real array	Section 3.5.1	s	
FYI	Character	Section 3.5.1		

&GDWT (Generating door open time code lines)

Parameter	Type	Reference	Units	Default value
ID	Character	Section 3.5.2		
PRE_PARM_INFO_ID	Character	Section 3.5.2		
GENERATOR_ID	Character array	Section 3.5.2		
PRODUCT	Logical	Section 3.5.2		False
FYI	Character	Section 3.5.2		

&IRXB (Inputting pre-parameter information of random obstruction）

Parameter	Type	Reference	Units	Default value
PRE_PARM_INFO_ID	Character	Section 3.6.1		Input for &MRND: PRE_PARM_INFO_ID+'_L'; PRE_PARM_INFO_ID+'_W'; PRE_PARM_INFO_ID+'_H'
NAMELIST	Character	Section 3.6.1		
INITIAL_POINT	Real array	Section 3.6.1	m	
LENGTH_RANGE	Real array	Section 3.6.1	m	
WIDTH_RANGE	Real array	Section 3.6.1	m	
HEIGHT_RANGE	Real array	Section 3.6.1	m	
FYI	Character	Section 3.6.1		

&GRXB (Generating random obstruction code lines）

Parameter	Type	Reference	Units	Default value
ID	Character	Section 3.6.2		
PRE_PARM_INFO_ID	Character	Section 3.6.2		
GENERATOR_ID	Character array	Section 3.6.2		
PRODUCT	Logical	Section 3.6.2		False
FYI	Character	Section 3.6.2		

&IOTH (Inputting pre-parameter information of other variables）

Parameter	Type	Reference	Units	Default value
PRE_PARM_INFO_ID	Character	Section 3.7.1		Input for &MRND: PRE_PARM_INFO_ID
NAMELIST	Character	Section 3.7.1		

ARG_NAME	Character	Section 3.7.1		
OTHERS_RANGE	Real array	Section 3.7.1	m	
FYI	Character	Section 3.7.1		

&GOTH (Generating other variable code lines)

Parameter	Type	Reference	Units	Default value
ID	Character	Section 3.7.2		
PRE_PARM_INFO_ID	Character	Section 3.7.2		
GENERATOR_ID	Character array	Section 3.7.2		
PRODUCT	Logical	Section 3.7.2		False
FYI	Character	Section 3.7.2		

&IFTD (Inputting pre-parameter information of fire time duration)

Parameter	Type	Reference	Units	Default value
PRE_PARM_INFO_ID	Character	Section 3.8.1		Input for &MRND: PRE_PARM_INFO_ID+'_I'; PRE_PARM_INFO_ID+'_G'; PRE_PARM_INFO_ID+'_GR' PRE_PARM_INFO_ID+'_P'; PRE_PARM_INFO_ID+'_D'; PRE_PARM_INFO_ID+'_DR'
INCIPIENT_TIME	Real array	Section 3.8.1	s	
FIRE_GROWTH_TIME	Real array	Section 3.8.1	s	
FIRE_GROWTH_RATE	Real array	Section 3.8.1	kW/s^2	
PEAK_TIME	Real array	Section 3.8.1	s	
DECAY_TIME	Real array	Section 3.8.1	s	
FIRE_DECAY_RATE	Real array	Section 3.8.1	kW/s^2	
FYI	Character	Section 3.8.1		

&GFTD (Generating fire time duration sample data)

Parameter	Type	Reference	Units	Default value
ID	Character	Section 3.8.2		

PRE_PARM_INFO_ID	Character	Section 3.8.2		
GENERATOR_ID	Character array	Section 3.8.2		
FYI	Character	Section 3.8.2		

&IMHR (Inputting pre-parameter information of maximum HRR)

Parameter	**Type**	**Reference**	**Units**	**Default value**
PRE_PARM_INFO_ID	Character	Section 3.8.3		Input for &MRND: PRE_PARM_INFO_ID+'_I'; PRE_PARM_INFO_ID+'_P'; PRE_PARM_INFO_ID+'_D'
INCIPIENT_MAX_HRR	Real array	Section 3.8.3	kW	
PEAK_MAX_HRR	Real array	Section 3.8.3	kW	
DECAY_MIN_HRR	Real array	Section 3.8.3	kW	

&GFTD (Generating maximum HRR sample data)

Parameter	**Type**	**Reference**	**Units**	**Default value**
ID	Character	Section 3.8.4		
PRE_PARM_INFO_ID	Character	Section 3.8.4		
GENERATOR_ID	Character array	Section 3.8.4		
FYI	Character	Section 3.8.4		

&GHRC (Generating HRR curve code lines)

Parameter	**Type**	**Reference**	**Units**	**Default value**
ID	Character	Section 3.8.5		
FIRE_TIME_DURATION_GENERATOR_ID	Character array	Section 3.8.5		
MAX_HRR_GENERATOR_ID	Character array	Section 3.8.5		
FIRE_PATTERN	Integer array	Section 3.8.5		
START_TIME	Real	Section 3.8.5	s	0
END_TIME	Real	Section 3.8.5	s	1800
TIME_FREQUENCY	Integer	Section 3.8.5	s	1

HRR_FLUCTUATION	Logical	Section 3.8.5		False
HRR_FLUCTUATION_RATE	Real	Section 3.8.5		
TIME_SLICE_SAMPLES	Real array	Section 3.8.5	s	
HRR_SAMPLES	Real array	Section 3.8.5	kW	
PRODUCT	Logical	Section 3.8.5		False
FYI	Character	Section 3.8.5		