FRAMEWORK TO GENERATE CFD-BASED SYNTHETIC AI DATA

Ivan D. Tomanović,
Srđan V. Belošević,
Aleksandar R. Milićević,
and Nenad Đ. Crnomarković

Department of Thermal Engineering and Energy,
"Vinca" Institute of Nuclear Sciences – National Institute of the Republic of Serbia,
University of Belgrade

Growing role of AI and data quality

- Recent years saw rapid growth in AI application and model development, directly correlated to growing computational power of modern computers.
- Training successful AI models requires **large**, **diverse**, and high-quality datasets.
- Quality of dataset is crucial for reliable model performance
- Common issues with datasets are **duplication**, inconsistency, **missing** or invalid data.

AI in power production and need to use synthetic data

- In coal-fired power plants, AI helps **optimize the power plant operation** in part or as a whole.
- Real-world datasets are often **incomplete or missing** operating conditions, since the plant must operate within certain parameters.
- **Synthetic data** can fill gaps — must balance *realism* and *coverage*.
- Good example of synthetic data is given by Abdallah Tariq Hasan Alabed et al. They were using **>75% synthetic data** and **improved AI performance** in classification tasks.

Our framework for synthetic data generation

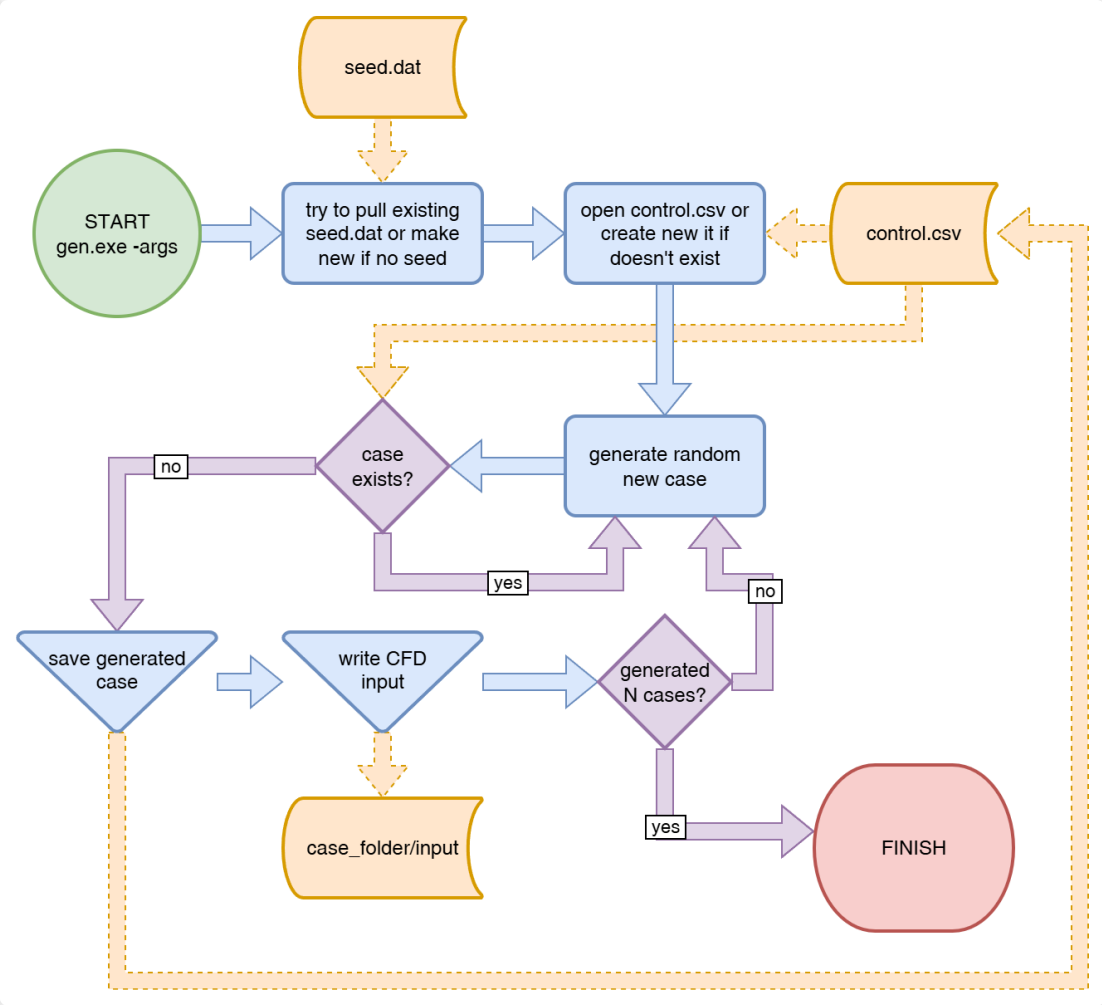
- Combines validated **CFD models** with new software tools providing reliable and validated CFD based outputs to fill gaps in existing data.
- **Synthetic datasets** for AI training are thus based on realistic power plant simulations, and simulations data is expected to have good agreement with real-world measurements.

- Framework components:

1. Random test-case generator
2. Test-case distribution & grouping
3. Execution & load management
4. Data merging & post-processing

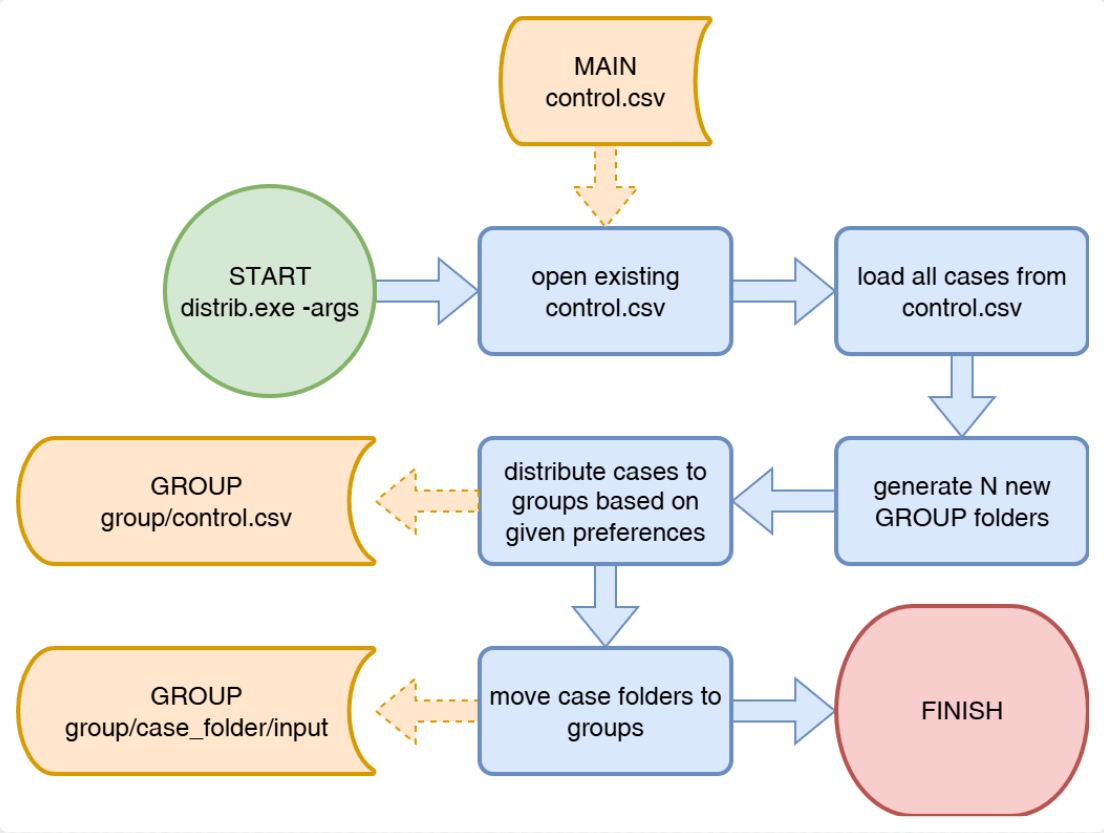
Random test-case generator structure

- Based on user input the program generates the exact number of test-cases.
- Embedded pseudo-random functions are used to ensure random distribution of varied parameters over given ranges.
- Generator ensures that cases are non-duplicate by comparing them with existing ones.
- The *control.csv* is updated with each new unique test-case.
- CFD input data is written in structured case folders as the case is generated.



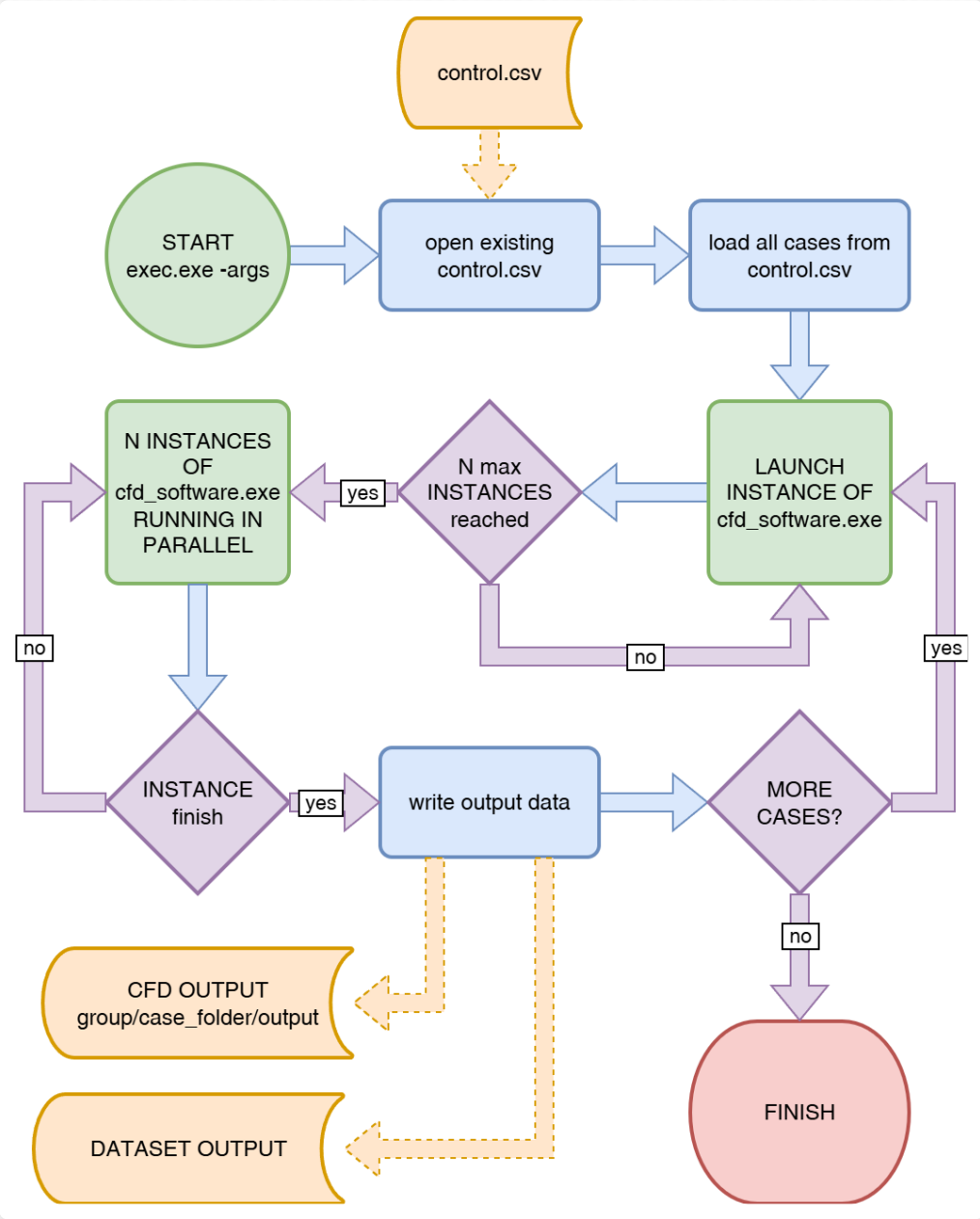
Structure of the tool for distribution and grouping of test-cases

- This tool allows distribution of simulations over several separated machines.
- Based on existing *control.csv* and internal markings on existing and completed cases it creates group folders.
- Test-cases are distributed based on input parameters given at tool run time.
- Each folder can be independently moved to a new machine and the in-house CFD software can be executed there.



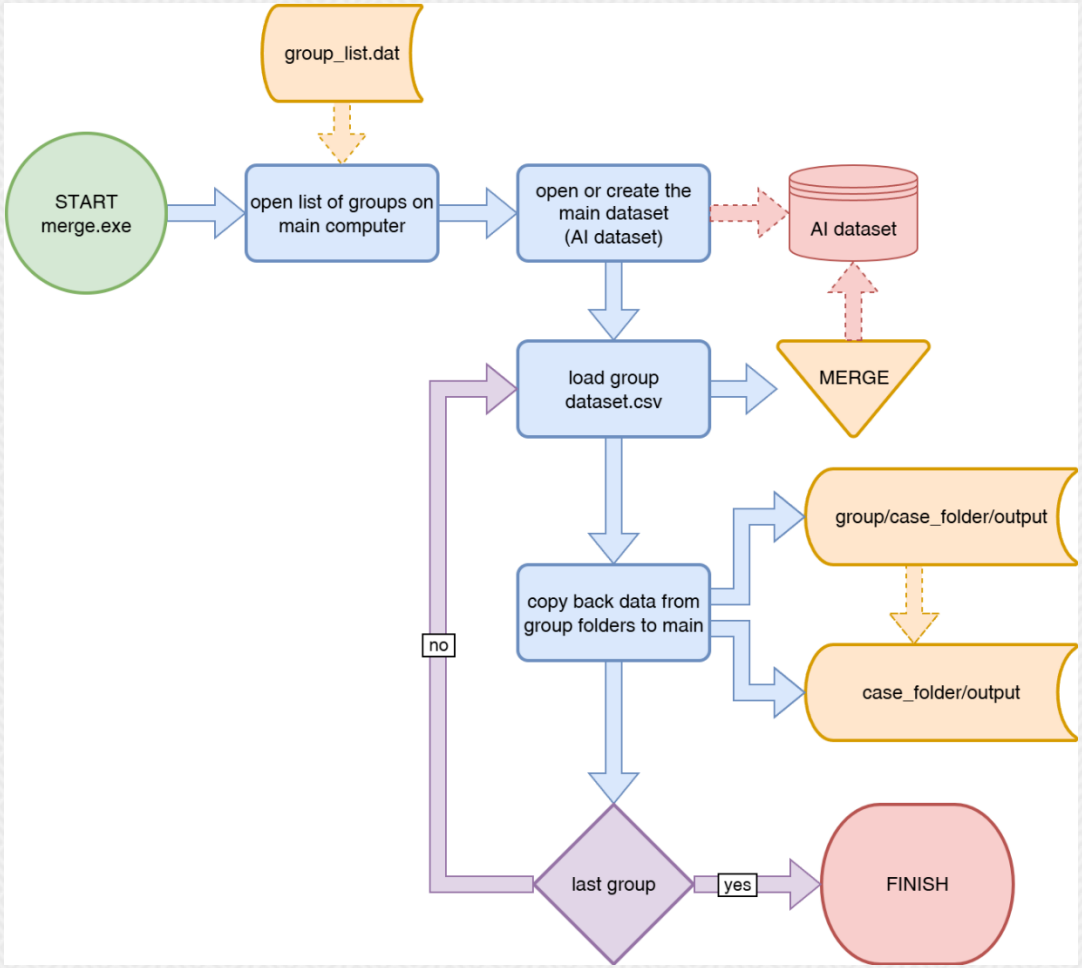
Execution control and load balancing tool

- This tool has task to manage number of running test-cases, as well as to ensure proper data storage in dataset and CFD outputs.
- User can give the number of cores/threads, and the tool will ensure that no more than given number of cases are executed in parallel.
- Tool is capable of resuming simulations from last completed. This is useful in case of unexpected crashes, or planned pauses.



Data collection tool and dataset merge

- After bringing all groups together this tool cycles through all folders and collects data from local *control.csv* files bringing it to a single file in root.
- Output dataset is formed containing all necessary data intended for AI training.
- CFD outputs are sorted in their respective folders under root folder.



Automated Framework for CFD-Based Synthetic Data Generation

- As previously shown, the developed framework integrates several tools to automate creation and management of synthetic datasets for AI training.
- It is divided into two main parts:
- A framework for automated data generation,
- and an in-house developed CFD model.

Automated Framework for CFD-Based Synthetic Data Generation

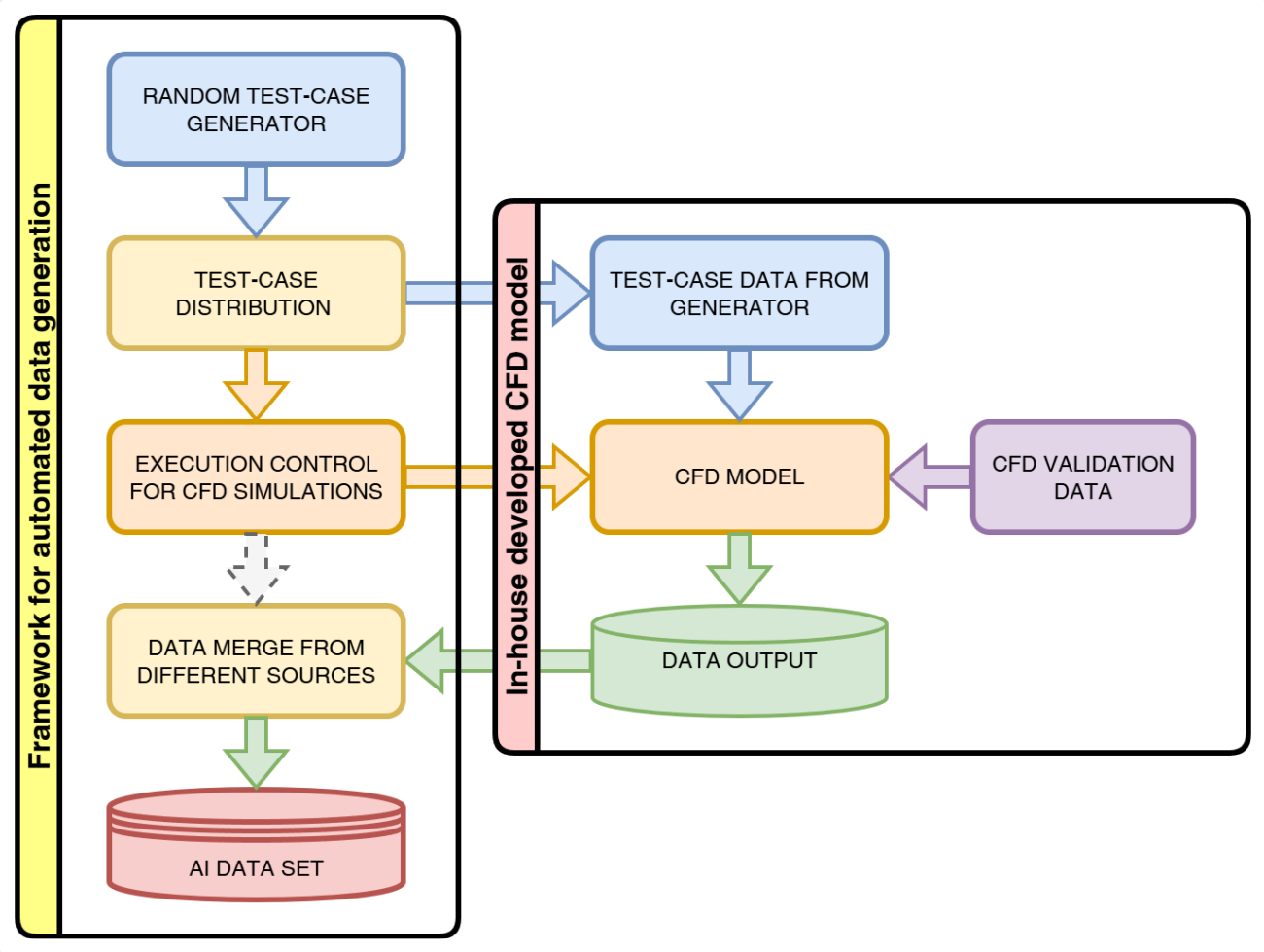
Framework for Automated Data Generation

- **Random test-case generator** creates diverse input scenarios.
- **Test-case distribution** organizes and assigns cases for execution.
- **Execution control** manages CFD simulation runs and balances computational load.
- **Data merge** consolidates results from multiple runs into a unified **AI dataset**.

In-house Developed CFD Model

- Receives **test-case data**. Model is previously validated using experimental data.
- Performs **validated CFD simulations** to generate **realistic data outputs**.
- These **outputs** are automatically **fed back into the framework** to form the final dataset.

Example framework



Conclusions

- The developed framework enables **rapid generation of AI training datasets** using outputs from a **validated in-house CFD model**.
- Automation and integration with control and load-balancing tools **minimize manual work** and **allow large-scale dataset generation**.
- The **framework is scalable** — adaptable to available computational resources and number of test-cases.
- It **provides synthetic data across a wide range of operating parameters**, often unavailable from measurements obtained under real plant operating conditions.
- Supports **creation of hybrid datasets** combining experimental and synthetic data.
- Allows for **training of reliable and efficient AI models** for industrial-scale process prediction and optimization.

Thank you for your attention

Questions and discussion welcome

You can reach us at *tccrg.edu.rs*