



OPEn HPC theRmomechanical tools
for the development of eAtf fuels

Deliverable D3.3 – Procedure for input data transfer in OFFBEAT

Version 1 – 18/07/2025



Funded by the European Union

Disclaimer

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

While this document has been prepared with care, the authors and their employers provide no warranty concerning the content and shall not be liable for any direct, incidental, or consequential damages that may result from the use of the information, or the data contained in it. Reproduction is authorised provided the material is unabridged and the source is acknowledged.

D3.3 version 1 Procedure for input data transfer in OFFBEAT

Document type	Deliverable
Document number	D3.3 version 1
Document title	Procedure for input data transfer in OFFBEAT
Authors	L. Verma, I. Clifford (PSI), A. Sclaro (EPFL), G. Zullo, D. Pizzocri, L. Luzzi (POLIMI), A. Jambrina (VTT), P. Van-Uffelen, A. Schubert, Z. Soti (JRC)
Release date	18/07/2025
Contributing partners	PSI, EPFL, POLIMI, VTT, JRC
Dissemination level	Public

Version	Short description	PMO	Main author	WP leader	Coordinator
1	First release	S. de Grandis (Sintec) 27/05/2025	L. Verma (PSI) 20/05/2025	J. Heikinheimo (VTT) 04/07/2025	B. Michel (CEA) 18/07/2025

Abstract

This deliverable presents the outcomes of the task 3.3 in WP 3 of the OperaHPC Project. The main objective of this task was to provide one-way coupling routines for 3D neutronics, thermal hydraulics, physical and chemical calculations with the 3D thermomechanical simulation using the OFFBEAT code. These coupling routines include a Serpent to OFFBEAT coupling and a coupling between the SCIENTIX fission gas release model and OFFBEAT. As an output of this task, a restart interface for transferring the results of industrial fuel performance codes as initial conditions for a subsequent OFFBEAT simulation was developed. The methodology for this restart interface along with testing using industrial fuel performance codes are presented in this deliverable.

Table of contents

Disclaimer	2
Abstract	3
Table of contents.....	4
Glossary	5
1 Introduction	6
2 Neutronics coupling between OFFBEAT-Serpent	7
2.1 Reactor physics applications	7
2.2 Kraken computational framework	7
2.3 High-fidelity sequence	8
2.3.1 Universal multi-physics interface	9
2.3.2 On-the-fly temperature treatment of cross sections	9
2.3.3 Domain decomposition.....	9
2.3.4 Advanced geometry types	9
2.4 Serpent-OFFBEAT coupling methodology	10
2.4.1 Unstructured mesh-based geometry type (OpenFOAM-based)	10
2.4.2 Coupling methodology.....	10
2.4.3 Proof-of-concept.....	11
2.4.4 Next steps	13
3 OFFBEAT-SCIANTIX coupling.....	14
3.1 Description of SCIANTIX model	14
3.2 Coupling with OFFBEAT	14
4 OFFBEAT restart interface	16
4.1 Restart Methodology.....	16
4.2 TRANSURANUS to OFFBEAT coupling.....	17
4.2.1 Summary of Previous Validation of the TRANSURANUS/OFFBEAT Coupling	17
4.3 Falcon to OFFBEAT coupling.....	19
4.3.1 Developments in support of the coupling	19
4.3.2 Testing the restart interface	19
5 Conclusions	22
References	23
Appendix A: Restart Variables	25
Appendix B: Intermediate file format	29

Glossary

AFEN	analytic function expansion nodal
BEAVRS	Benchmark for Evaluation And Validation of Reactor Simulations
CRAM	Chebyshev Rational Approximation Method
CSG	constructive solid geometry
DBRC	Doppler-broadening Rejection Correction
FENM	flux expansion nodal method
FGR	Fission Gas Release
FPC	Fuel Performance Codes
PCMI	Pellet-Cladding Mechanical Interaction
PIE	Post-Irradiation Examination
TMS	Target Motion Sampling

1 Introduction

The existing industrial codes are well-validated and provide reliable and comprehensive analyses of fuel thermal, mechanical and neutronic behaviour. OFFBEAT [1] can take advantage of these codes for specific solvers or phenomena. Serpent [2], for example, is a widely used neutronics code and can be utilized to provide 3D pin power distributions as input conditions to OFFBEAT. Similarly, SCIANTIX [3] is a 0-D fission gas release model that is embedded into OFFBEAT. As the SCIANTIX model is improved, the coupling of the newer versions with OFFBEAT is verified and tested. A thorough coupling methodology between these well-validated and reliable industrial codes and OFFBEAT is then needed. Another approach developed within the task is to utilize widely validated 2D/1.5D fuel behaviour codes, such as TRANSURANUS [4] or Falcon [5], with OFFBEAT. These industrial fuel performance codes can be coupled with OFFBEAT to get a more detailed 3D analysis of a selected transient. For example, the base irradiation thermomechanical behavior can be simulated rapidly using the traditional codes, and the resulting pre-transient conditions can be transferred to OFFBEAT for computationally more demanding 3D transient analysis.

Coupling methodologies between the industrial codes and OFFBEAT have been explored within the task, and an interface for OFFBEAT restart following a base irradiation simulation using an industrial fuel behaviour code have been developed and tested.

Apart from PSI, who act as a task leader, the partners involved in this task are EPFL, POLIMI, JRC and VTT. The dedicated roles assigned to each partner within this task were as follows:

- VTT and EPFL to collaborate on the neutronics coupling, with routines created to couple 2D/3D fuel behaviour simulations performed with OFFBEAT with 3D neutronics calculations performed with Serpent.
- PSI, EPFL, POLIMI and JRC to focus on the restart interface of OFFBEAT. All partners to contribute towards defining a consistent methodology of supplying suitable restart data to OFFBEAT based on reference 1.5D simulations. Key focus areas were the data requirements, defining a common intermediate data format and creation of new tools/utilities/functionality to facilitate the restart. Each partner to separately create tools to convert the output from their respective 1.5D codes to the common intermediate format.
 - PSI to develop a restart interface to the Falcon fuel behaviour code.
 - EPFL and JRC to develop a restart interface to the TRANSURANUS fuel behaviour code.
 - POLIMI to contribute towards defining and developing the restart interface to OFFBEAT, with focus on the coupling with SCIANTIX and in general to the quantities related to fission gas behaviour.

The outcomes of the task are described in the following sections, focussing on specific coupling methodology and codes.

2 Neutronics coupling between OFFBEAT-Serpent

Serpent [2] is a continuous-energy Monte Carlo neutron and photon transport code that offers built-in capabilities for both static (eigenvalue and fixed-source) and dynamic simulations, fuel depletion, group-constant generation, multi-physics coupling, and variance reduction. Currently, Serpent is developed as part of the Kraken multi-physics framework [6], which provides dedicated tools for core-level reactor physics analyses. Within Kraken, Serpent can be utilized either as a high-fidelity neutronics solver or to generate homogenized group constants for the Ants nodal neutronics code. The neutron and photon transport modes in Serpent allow the code to be employed for various stand-alone applications beyond reactor physics, including radiation shielding and fusion neutronics.

To provide a solid and understanding framework to the work conducted in the OperaHPC project under the Serpent-OFFBEAT coupling task, the key features in the context of reactor physics and Kraken computational framework are briefly introduced.

2.1 Reactor physics applications

Serpent has been developed with a pragmatic approach, focusing on reactor physics applications from the project's outset. Neutron interaction data is obtained from standard ACE format libraries, which support all major reaction modes and provide separate models for thermal scattering and unresolved resonance probability table sampling. Additionally, Doppler-broadening Rejection Correction (DBRC) is available as a separate model for resonance scattering at low energy.

The software includes a built-in fuel depletion solver that combines neutron reactions with ENDF radioactive decay and fission yield data. The Bateman equations are solved using the Chebyshev Rational Approximation Method (CRAM) [7]. Burnable materials can be automatically divided into depletion sub-zones to improve spatial resolution. In addition to solid fuels, the depletion solver enables the modelling of continuous material streams in liquid-fuelled reactors.

The standard geometry model utilized by Serpent is based on a three-dimensional, universe-based constructive solid geometry (CSG) type, which is adequate for most reactor applications involving regular structures. For micro-particle fuel types and pebble-bed reactors, Serpent features an explicit geometry model that allows for descriptions of the geometry without major approximations [8]. Fragmented fuel and other stochastic heterogeneous configurations can be modelled using a geometry type based on Voronoi tessellation [9]. More complicated core structures can be modelled using CAD and mesh-based geometry types.

Serpent is also suitable for coupled reactor physics applications, serving either as a high-fidelity neutronics solver or for generating homogenised cross sections for a reduced-order transport code. At VTT, the approach relies on the Kraken computational framework [6], which manages the two-way coupling to thermal hydraulics and fuel behaviour solvers.

2.2 Kraken computational framework

Kraken [6] is VTT's multi-physics, multi-scale computational platform designed to solve the coupled reactor physics problem. It features a driver module and an interface that facilitates the exchange of input and output data between the physics solvers, along with pre- and post-processing capabilities. The coupling is intended to be code-agnostic; however, the default configuration includes neutronics, thermal hydraulics and fuel behaviour solvers developed at VTT. The computational sequences are illustrated in **Figure 1**.

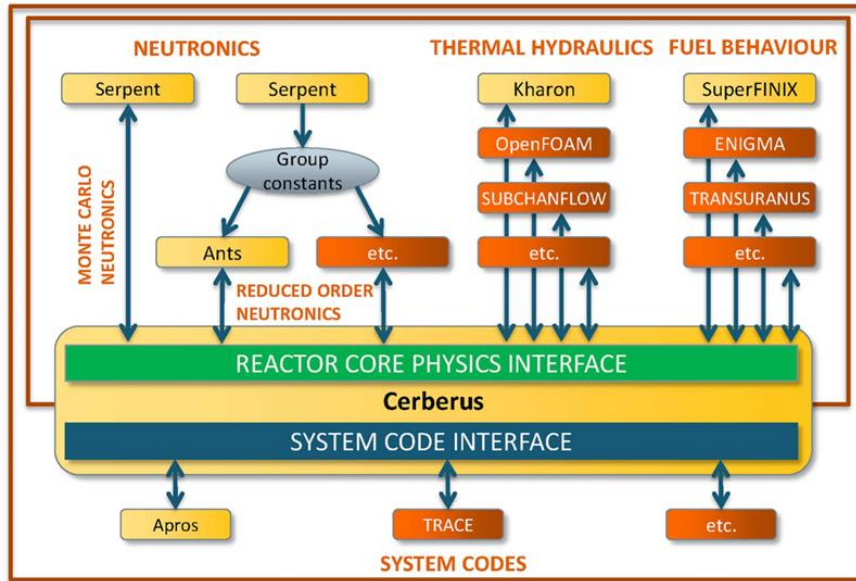


Figure 1: The Kraken computational framework. Data transfer is handled through the driver module Cerberus, which also handles the external coupling to system-scale simulations. The default configuration is based on modular solvers developed at VTT (yellow). Various third-party solvers (orange) have also been coupled to Kraken, such as the SUBCHANFLOW sub-channel thermal hydraulics code [9][10], and the TRANSURANUS fuel performance code [4].

Neutronics calculations in Kraken are based on the Serpent code. In the reduced-order sequence, Serpent is used to produce homogenised group constants for the multi-group nodal neutronics program Ants. Ants solves eigenvalue or fixed-source steady-state, depletion, and time-dependent problems for rectangular or hexagonal core geometries [12], [13] and [14]. Hexagonal cores can also be modelled with a triangular nodal geometry [15]. The nodal diffusion solution is based on a combination of the flux expansion nodal method (FENM) and the analytic function expansion nodal method (AFEN). Ants supports pin power reconstruction and microscopic depletion of nodal nuclide compositions [16] and [17].

The fuel behaviour solution is provided by the SuperFINIX core-level fuel behaviour solver [18], which delegates the modelling of individual or representative fuel rods to the FINIX fuel behaviour module [19]. In steady-state and fuel cycle simulations, the core-level thermal hydraulics solution can be obtained using the time-independent two-phase closed-channel porous-medium thermal hydraulics solver Kharon, which was originally developed as a placeholder for more advanced methodology. Time-dependent simulations and problems requiring more accurate modelling of mixing phenomena have been handled using an OpenFOAM-based solver applying the porous-medium approximation.

The core physics model in Kraken can also be coupled to system-scale simulations via external boundary conditions. At VTT, the external coupling is typically established to the Apros process simulator [20], but Kraken has also been coupled to the TRACE code [21][22].

2.3 High-fidelity sequence

When used as part of the high-fidelity calculation sequence in Kraken, Serpent is integrated with the other physics solvers through the multi-physics interface in the Cerberus module. Temperature and density distributions are introduced into the Monte Carlo simulation from the coupled solvers, while heat deposition distributions from neutron and photon interactions transmitted in the opposite direction.

The features and capabilities that enable such coupling and provide context for the Serpent-OFFBEAT coupling, are discussed below.

2.3.1 Universal multi-physics interface

The multi-physics coupling scheme was implemented in Serpent before developing the Kraken framework [23]. The methodology is based on the complete separation of state-point information from the geometry description. Density and temperature distributions are superimposed over the geometry, eliminating the need for modifications, such as material subdivision, in the underlying model. The methodology supports various interface types, including an unstructured OpenFOAM format polyhedral mesh [24] and a dedicated interface type for fuel performance coupling [25].

While most interface formats depend on spatially discretized distributions, the methodology in Serpent is based on a rejection sampling, allowing temperatures and densities to exhibit continuous changes. Density variations are accounted for by applying a simple multiplier to the underlying material density [26], but variations in temperature involve more detailed physics.

2.3.2 On-the-fly temperature treatment of cross sections

The ACE data format provides continuous-energy cross sections for neutron interactions. Additionally, cross sections in the unresolved resonance region can be managed through probability table sampling, while molecular and lattice binding effects are addressed by specific thermal scattering cross sections and $S(\alpha,\beta)$ scattering laws. Each data type is pre-generated for a specific temperature. However, in multi-physics calculations, the temperature must be adjustable to reflect local variation.

The methodology applied in Serpent for continuous-energy cross sections utilize the Target Motion Sampling (TMS) method [27]. For every neutron collision, the energy and direction of motion of the target nuclide are sampled from a Maxwellian based distribution. A coordinate transformation is performed to the target-at-rest frame of reference, enabling the use of cross sections generated for the corresponding temperature. Temperature treatment for thermal scattering data relies on interpolation between fixed temperatures [28]. Similar interpolation technique has been implemented for unresolved resonance probability table sampling [29].

2.3.3 Domain decomposition

The default optimisation mode in Serpent employs pre-generated macroscopic total cross sections to avoid the costly operation of summing over nuclides each time a new track length is sampled from a material [30]. While this approach results in a large memory footprint for burnup calculation, which increases with the number of burnable materials, full-core burnup calculations exceed the limits of the conventional methodology. With millions of depletion zones, even storing the nuclide compositions requires excessive computer memory. To address this, domain decomposition is used, dividing the geometry among computational nodes, with each operating within its own memory space. Neutrons crossing a domain boundary are transferred on-the-fly to another node during the transport simulation.

In Serpent, domain decomposition relies on a collision-based approach [31]. Instead of fixed boundaries defined by the geometry, domains are assigned on material basis. Neutrons are transferred to another node when a collision occurs in a material belonging to a different domain. The division is applied only to burnable materials based on their location in the global geometry. Limiting decomposition to fuel materials helps reduce the computational overhead, as most collisions typically happen in the moderator, where the material data is shared by all nodes.

2.3.4 Advanced geometry types

The standard CSG geometry type may not always be ideal for modelling complex irregular systems. Constructing large, intricate geometries requires significant user effort and is susceptible to input errors and

D3.3 version 1 Procedure for input data transfer in OFFBEAT

computational bottlenecks. To address this, Serpent allows the importing of CAD- and mesh-based models. All geometry types operate at the universe level, enabling the use of different types for different parts of the modelled system.

The CAD-based geometry type relies on the STL file format, which is commonly used in various 3D applications and supported by most CAD tools. The geometry is comprised of solid volumes defined by tessellated surfaces. The computational cost of the tracking routine is significantly reduced by an adaptive search mesh applied on top of the CAD components [32].

The CAD solids are surface models without any internal structure. For applications requiring spatial subdivision it is possible to use a mesh-based model instead [33]. The geometry type is based on the same OpenFOAM polyhedral mesh (so-called “polymesh”) that is also used by the multi-physics interface.

2.4 Serpent-OFFBEAT coupling methodology

OFFBEAT [1] is a state-of-the-art multi-dimensional fuel performance code based on the open-source C++ library OpenFOAM. As a first step towards an OpenFOAM-based high-fidelity multi-physics coupled sequence (e.g. handled by Cerberus, Kraken multi-physics driver) is to develop the individual features that enables the unstructured-mesh-based solution and transferral of quantities of interest. At this point, the focus has been on the Serpent-side, one-way coupling between Serpent-OFFBEAT.

2.4.1 Unstructured mesh-based geometry type (OpenFOAM-based)

The unstructured-mesh-based geometry is a by-product of the unstructured-mesh-based multi-physics interface. It can be used to create solely a geometry universe-based on an unstructured mesh (solid 1), but also possible to create the geometry and bring in temperature and/or density data on the same mesh, superimposed (solid 3). In the latter case, most of the parameters are given in a separate multi-physics interface file (ifc – interface type: 7, 8, 9), which can be re-read by Serpent to update the fields between coupled calculation iterations, if necessary.

The input format is based on the OpenFOAM polyhedral mesh format (so-called “polymesh”) – the mesh can consist of arbitrary polyhedrons if they follow the OpenFOAM file-format and some mesh-quality requirements.

Serpent does the final particle tracking in a tetrahedral mesh, which is automatically generated by splitting other polyhedra into tetrahedra (internal conversion). Tetrahedrons are used in the particle tracking as each of their face surfaces is defined by three points (the vertices of the face) this is exactly the number of points required to uniquely define a plane. Since Serpent tests whether a point is inside a cell by testing whether it is on the correct side of each of the surfaces making up the cell, it is important that the surfaces bounding the cell are uniquely defined.

Faces with more than three points may not be entirely planar (either due to mesh generation/deformation) or limited numerical precision. To ensure that these kinds of faces are treated in the same way in both the owner cell and the neighbour cell the mesh is split into tetrahedrons before the neutron tracking.

Serpent automatically runs some basic tests for the mesh right after reading the data, reporting inconsistencies, and triggering the split to tetrahedra, if needed. After that, some additional tests are run to verify the final mesh.

2.4.2 Coupling methodology

The one-way coupling between Serpent-OFFBEAT is built on the unstructured-mesh-geometry type: an OpenFOAM-based mesh for both the transport and burnup solutions (**Figure 2**).

D3.3 version 1 Procedure for input data transfer in OFFBEAT

The two available options are:

- Standard approach: (.det and .dep files) using the standard detector and burnup outputs for an unstructured-mesh geometry.
- Interface approach: a multi-physics interface transfers the flux/power distributions (i.e., derived from the transport solution) and a depletion interface transfers the burnup/isotopic compositions (i.e., derived from the depletion/activation solution).

Based on the unstructured mesh-geometry type, a depletion based on an unstructured mesh has been developed to integrate the OpenFOAM capabilities within the depletion solver.

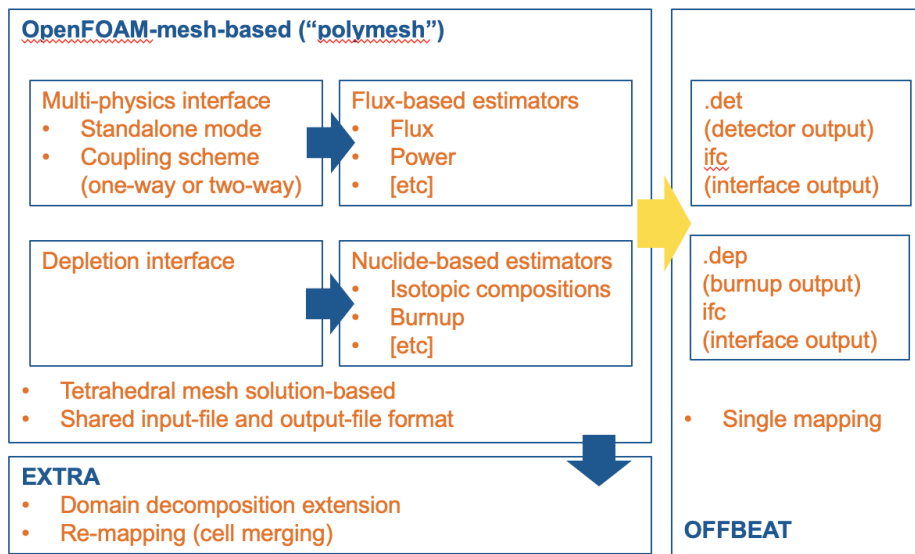


Figure 2: Serpent-OFFBEAT coupling diagram (and key features).

Both interfaces are built on the OpenFOAM polyhedral format, so-called "polymesh". Serpent solves the transport/burnup system on the tetrahedral meshes (with an internal conversion from any other mesh types, e.g., hexahedral). Additionally, they shared the same inputs (mesh/parameters) and output (format/mapping).

Targeting some computational performance, two extra features have been included: a) domain decomposition (extension to unstructured-mesh-based geometries) by enabling multi-node burnup solution, re-mapping or cell merging either for reducing the statistical variance or simplifying the computational scheme.

2.4.3 Proof-of-concept

Due to the complexity in the verification of the implementation of the methodology given: a) the unstructured nature of the OpenFOAM mesh, b) the original vs. native mesh-solution in Serpent (i.e., polyhedral vs. tetrahedral mesh), and c) the stochastic nature of Serpent Monte Carlo code, standard reactor physics or fuel behaviour benchmarks were not used to test the new features. Instead, a proof-of-concept approach was adopted using a simple but robust "toy problem" designed to eliminate ambiguity.

The "Toy problem" considered is a 2x2x2 cubes geometry – a non-standard variation of the Rubik's Cube. The geometry is simple enough to build one-to-one model based on standard CSG universe-based (Figure 3, left-top) and OpenFOAM-mesh-based geometries, enabling variations of the setup.

Doing so, one-to-one models, both the transport and burnup solutions should be identical, providing unambiguous confirmation of the success of the implemented features.

D3.3 version 1 Procedure for input data transfer in OFFBEAT

The two basic configurations are:

1. CSG (8 cubes) vs OpenFOAM (8 hexahedral cells) – see **Figure 3**, right-top.
2. CSG (8x5, 8 cubes divided by planes into 5 tetrahedrons each) vs. OpenFOAM (40 tetrahedral cells) – see **Figure 3**, left-bottom.

Additionally, a mixed case, to show the feature of re-mapping, i.e. from an OpenFOAM tetrahedral based setup to a cube-like geometry (**Figure 3**, right-bottom).

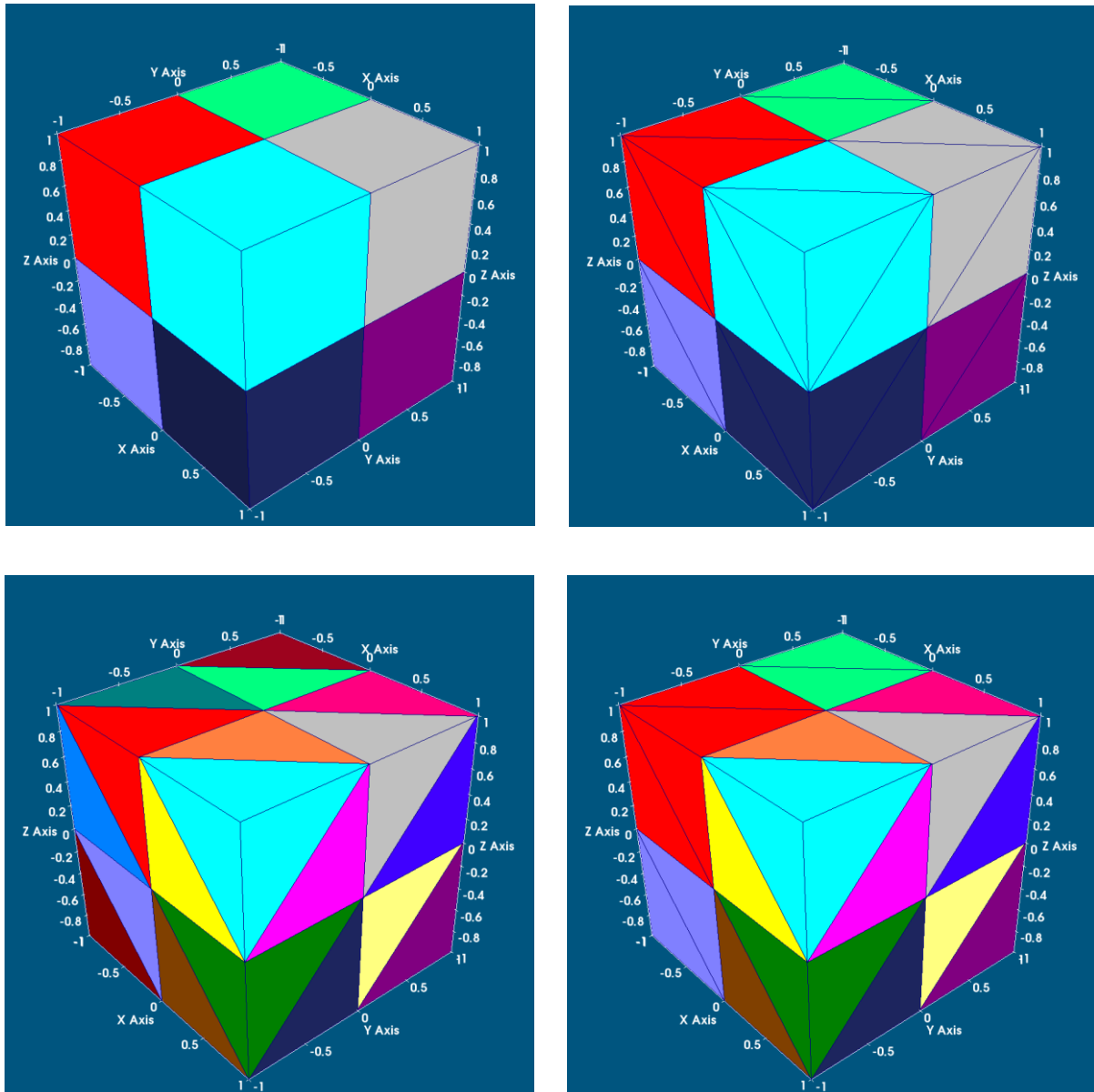


Figure 3: CSG-based 2x2x2 cubes (left-top), hexahedral-mesh-based (right-top), tetrahedral-mesh-based (left-bottom), and re-mapping: tetrahedral-mesh-based merged into hexahedral-mesh/cubes.

The different configurations have been successfully tested using different input options (e.g., calculation modes, tracking algorithms, sources, etc.). More complex verification (i.e. more intricate geometries) of the new features in Serpent do not provide more insight at this stage of the one-way coupling.

2.4.4 Next steps

Continuing the work developed in OperaHPC WP3 and reported in this deliverable, two key tasks are envisioned as natural lines of research:

- i. Validation:
The Serpent-OFFBEAT coupling has been thoroughly verified with simple but unequivocal models, however, non-realistic ones. Therefore, the validation task with Serpent-OFFBEAT test cases is the following action. Starting, for example, with the Task 3.1 setup (presented in deliverable D3.1), and a comparison between standard vs. OpenFOAM-mesh-based geometry in the depletion and/or transient, RIA, case.
- ii. Collaboration with EPFL:
Further features, more advanced and topical for both reactor physics and fuel-performance behaviour, can be developed by the partnership of VTT and EPFL, developers of Serpent (and Kraken) and OFFBEAT, respectively. Including, e.g., optimization and performance features, two-way-coupling, etc.
- iii. Kraken computational framework:
Based on the initial coupling developments presented in this report, OFFBEAT can also be brought into the Kraken computational framework in the future, as a solver on its own and as part of the coupled calculation sequences

3 OFFBEAT-SCIANTIX coupling

3.1 Description of SCIANTIX model

SCIANTIX [3] is an open source 0D mesoscale code developed at POLIMI to model the behaviour of gaseous and volatile fission products in nuclear fuel. Version 2.0 (<https://github.com/sciantix/sciantix-official>) features an object-oriented architecture and advanced physics-based models describing the production, transport, and trapping of xenon, krypton, and helium in the fuel matrix and in high burnup structures. SCIANTIX accounts for intragranular/intergranular diffusion, bubble nucleation and growth, gas release and swelling, and includes transient effects. It can operate in standalone mode or be coupled with other codes, including fuel performance and multiphysics platforms.

The internal structure of the SCIANTIX 2.0 code reflects a modular and maintainable design. The code folders are organized into the SCIANTIX directory, within the following directories,

```

SCIANTIX/
├── Make/                # Compilation rules and scripts
├── include/            # C++ headers defining classes and models
│   ├── classes/       # Core data structures and variables
│   ├── file_manager/  # Input/output and file parsing utilities
│   ├── models/        # Physics-based models (e.g. diffusion, trapping)
│   ├── namespaces/    # Global constants and namespaces
│   └── operations/    # Numerical operations and utilities
└── src/               # C++ source files implementing the logic
    ├── classes/       # Class implementations
    ├── file_manager/  # File reading/writing procedures
    ├── models/        # Model implementations
    ├── namespaces/    # Namespace initialisations
    └── operations/    # Supporting operations and algorithms

```

3.2 Coupling with OFFBEAT

POLIMI integrated the new Version 2.0 of SCIANTIX into the OFFBEAT code, in a dedicated development branch named (<https://gitlab.com/AlessandroScolaro/offbeat-official/-/tree/development/sciantix2>) in a dedicated class, `fgrSCIANTIX`. This class acts as a two-way interface between the OFFBEAT thermomechanical solver and the SCIANTIX model. At each time step and for each fuel cell, the interface exchanges field values such as temperature, burnup, hydrostatic stress, and fuel density. These are used to initialize and run the new SCIANTIX simulation object (through *initialize*, *execute*, *update* commands), which computes fission gas release, swelling, bubble characteristics, and other internal variables. These outputs are then transferred back to OFFBEAT and incorporated in the mechanical calculations.

The class manages a comprehensive set of scalar fields for xenon and krypton behaviour (e.g., grain-boundary and grain-matrix inventories, released gas, bubble radii and densities), and supports restart operations via OpenFOAM registry mechanisms. It enables transient 3D simulations in OFFBEAT with a detailed and validated description of fission gas evolution, ensuring physical consistency with past fuel history.

D3.3 version 1 Procedure for input data transfer in OFFBEAT

The class `fgrSCIANTIX` is instantiated and used within the OFFBEAT driver (`offbeat.C`) through the base class `fissionGasRelease`. Its methods `correct()` and `updateVariables()` are called at each outer iteration and time step respectively, ensuring that SCIANTIX is tightly integrated in the thermomechanical solution loop and receives the necessary field updates.

A performance test was performed on the IFA-432 Rod 1 configuration, comparing the execution time of a default OFFBEAT simulation with one using SCIANTIX 2. The results showed only a marginal overhead: 300 seconds for the default simulation versus 312 seconds for the coupled version, confirming the efficiency of the integration.

This work was mainly addressed during two dedicated research visits (mobility periods) at EPFL: one in October 2023 (two weeks), and one in February–March 2024, during which the integration of SCIANTIX into OFFBEAT and on the restart interface, beside the validation effort pertaining the Task 5.2, were carried out.

4 OFFBEAT restart interface

A restart interface to OFFBEAT after industrial fuel performance code calculations is developed and tested within this task and is presented in this section. First, the methodology adopted for the restart is discussed followed by the demonstration for the usage and testing of the restart utility after fuel performance calculations using TRANSURANUS and Falcon codes.

4.1 Restart Methodology

The utility developed for the restart of OFFBEAT after an industrial fuel performance code calculation is called ‘map1DFields’. As the name suggests, this utility maps the 1D fields from conventional codes like TRANSURANUS or Falcon to a 3D OFFBEAT geometry. A representation of the input data transfer from industrial fuel performance codes to OFFBEAT is represented in **Figure 4**. The 1D calculations done for each axial slice of the industrial code can be transferred to OFFBEAT using the map1DFields utility to perform more detailed 2D r- θ or 3D analyses.

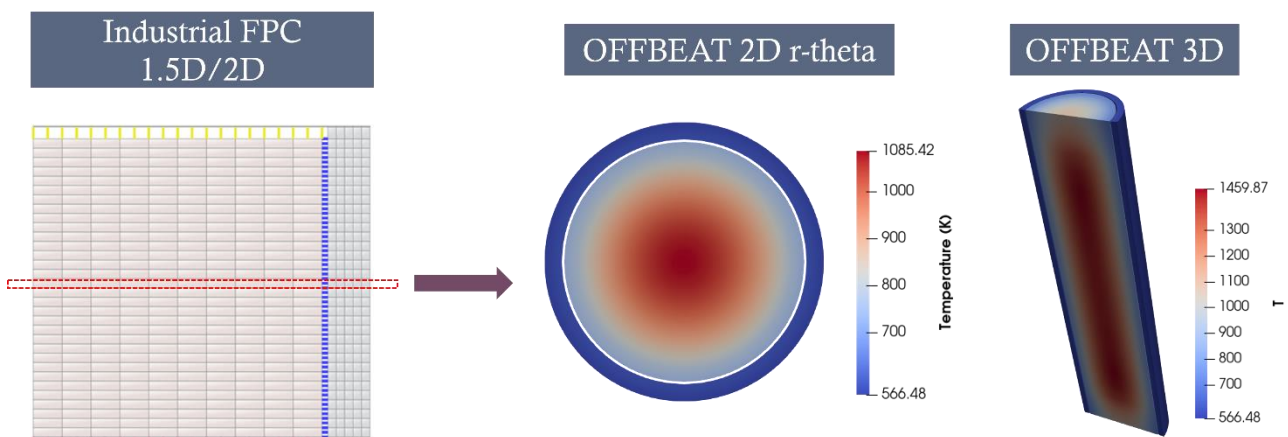


Figure 4: Representation of input data transfer from 1D to 3D geometry.

The methodology adopted for the restart interface includes obtaining the results from the industrial fuel performance codes and creating an intermediate file containing the radial-axial fields of interest. This intermediate file is then fed to the map1DFields utility which interpolates the fields into a 3D geometry at the set time of restart. The algorithm to follow for the restart is presented in **Figure 5** and detailed below:

1. The industrial fuel performance code calculation will be carried out.
2. The output from the industrial code calculation will be written into an intermediate file. Each code user will develop their own scripts to convert the output to the intermediate format.
3. The intermediate file will follow a consistent format. It will be an OpenFOAM dictionary containing the fields of interest in a 2D radial-axial (r-z) table. Examples of fields needed are presented in **Appendix A** and an example format of the intermediate file is shown in **Appendix B**.
4. The intermediate file is provided to the map1DFields utility, which transforms the 2D r-z fields into a 3D OFFBEAT mesh, which is created a priori.
5. The utility will return the transformed fields in the ‘restart time’ directory of the OFFBEAT restart case.
6. Once the time directory with the fields is obtained, the OFFBEAT case can be set up as usual by providing the boundary conditions and other simulation conditions.

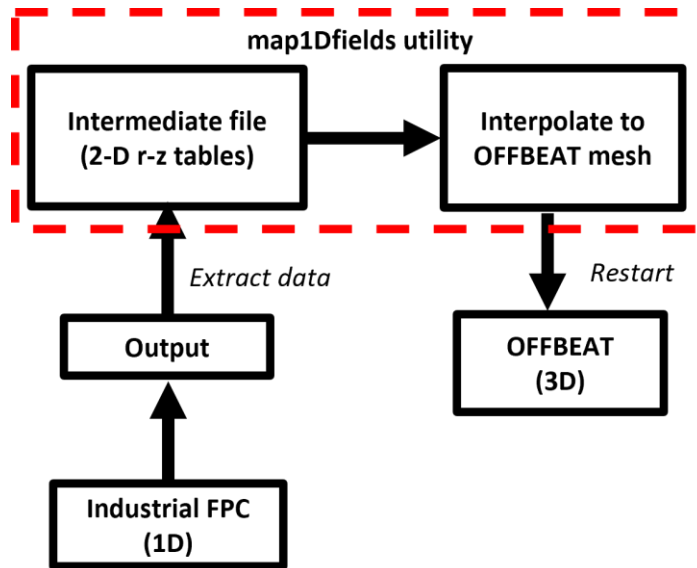


Figure 5: Algorithm for an industrial fuel performance code to OFFBEAT restart.

The utility can be found in the git repository of the OFFBEAT code at [utilities/map1DFields · development/FalcOFFBEAT · foam-for-nuclear project / OFFBEAT · GitLab](https://github.com/foam-for-nuclear-project/foam-for-nuclear-project/tree/development/FalcOFFBEAT). Here you can find the source code for the utility along with the make files. Once it is compiled, to use the utility, simply copy your intermediate file named ‘map1DFieldsDict’ to the OFFBEAT case */system* folder. Next run *map1DFields* command and the utility will create the restart time directory in your current OFFBEAT Case folder with the transferred fields.

In the following, two test cases for the restart interface are presented. The first uses TRANSURANUS fuel performance code as the industrial code and the second uses Falcon as the initial industrial code for subsequent OFFBEAT restart.

4.2 TRANSURANUS to OFFBEAT coupling

To test the TRANSURANUS/OFFBEAT coupling, the AN3 rod from the Risoe3 experiment was used as a reference case. This rod has been previously analysed in earlier studies with the initial version of the interface [34] and no additional validation was conducted within this task. However, a set of quality checks was performed to compare the TRANSURANUS end-of-base irradiation output with the OFFBEAT post-mapping initial conditions at the start of the ramp, using the new interface. These checks ensured that all relevant fields (e.g., temperature, stress, strain, internal pressure) were correctly transferred from TRANSURANUS to OFFBEAT.

This confirms the technical functionality of the new interface and validates its readiness for use in coupled TRANSURANUS+OFFBEAT workflows. Further validation efforts on the Risoe3 rods experimental database using the updated TRANSURANUS-to-OFFBEAT restart routine will be addressed in other work packages of the OperaHPC project.

To ensure continuity and completeness of this document, we summarize below the results of the initial validation effort, performed, and published prior to the restructuring of the interface.

4.2.1 Summary of Previous Validation of the TRANSURANUS/OFFBEAT Coupling

A full demonstration of the TRANSURANUS/OFFBEAT restart methodology was previously carried out and published in Progress in Nuclear Energy [34]. The methodology involved using TRANSURANUS to simulate the base irradiation of the AN3 rod and restarting the simulation in OFFBEAT to analyze the 72-hour bump test, focusing on pellet-cladding mechanical interaction (PCMI).

D3.3 version 1 Procedure for input data transfer in OFFBEAT

The mapping process from TRANSURANUS to OFFBEAT was verified by comparing key geometric quantities (fuel radius, cladding inner/outer radius) before and after the restart. Discrepancies were below $1\ \mu\text{m}$.

A comparison between the coupled simulation and a standalone TRANSURANUS run showed good agreement. For instance, the fuel centerline temperature differed by less than 50 K (see **Figure 6**), and the integral fission gas release (FGR) differed by $\sim 1\%$ (**Figure 7**). These results indicate not only the correctness of the restart but also a high level of consistency between the physical models used in TRANSURANUS and those implemented in OFFBEAT. The agreement confirms that the transition from TRANSURANUS to OFFBEAT does not introduce artificial discontinuities or mismatches in the predictions, at least for the ramps considered in the study.

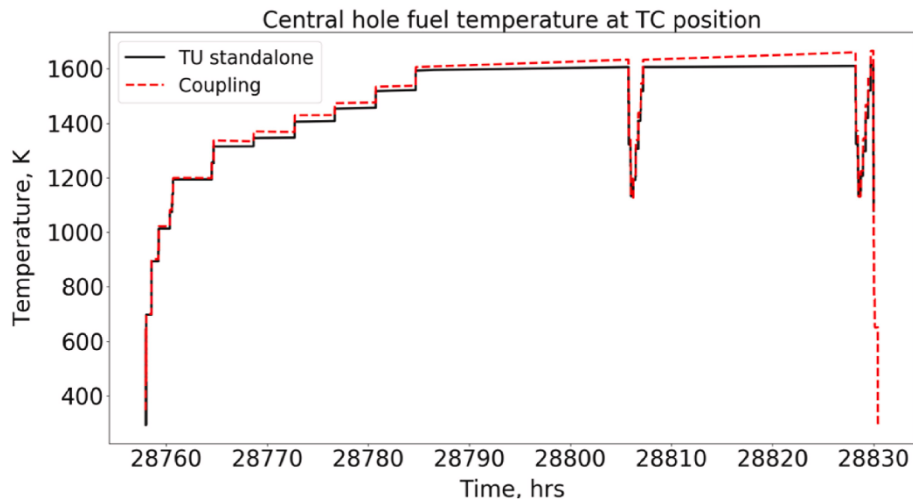


Figure 6: Fuel centreline temperature at the TC location during the AN3 bump test: comparison between standalone TRANSURANUS and coupled TRANSURANUS/OFFBEAT simulation [adapted from [34]].

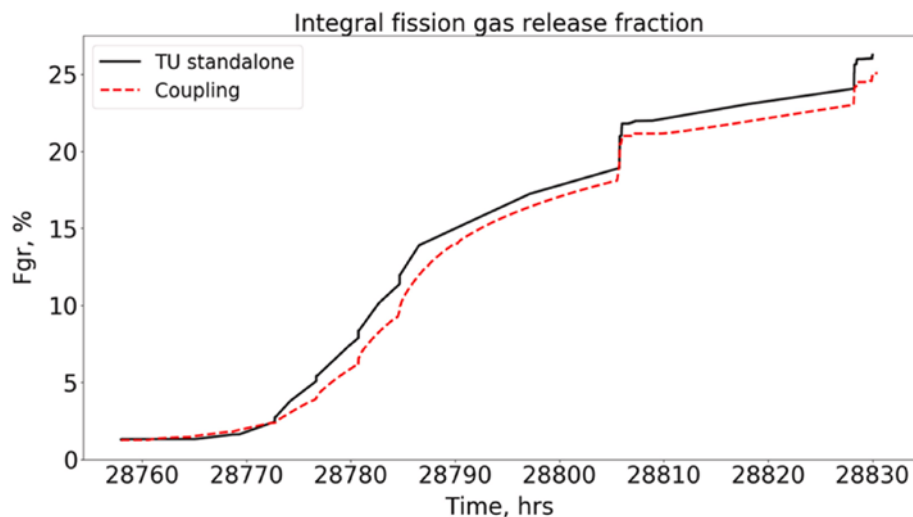


Figure 7: Fission gas release during the AN3 bump test: standalone TRANSURANUS vs. coupled simulation [adapted from [34]].

The coupled simulation results were also compared to Post-Irradiation Examination (PIE) data from the Risø-3 experiment. The predicted average cladding diameter increase during the bump ($\sim 17\ \mu\text{m}$) was in reasonably good agreement with the measured value ($\sim 25\ \mu\text{m}$). Similarly, the simulated ridge height ($\sim 3\ \mu\text{m}$) was below but comparable to the measured $\sim 13\ \mu\text{m}$. These results highlight areas where further refinement

D3.3 version 1 Procedure for input data transfer in OFFBEAT

of models (e.g., recovery of pellet strength after contact, or relocation recovery mechanisms) could help to improve quantitative agreement, but they do not detract from the validity of the overall coupling approach.

These results demonstrated that the coupling methodology was capable of supporting high-fidelity transient simulations based on realistic base irradiation histories. The task 3.3 in the OperaHPC project generalizes and modularizes this interface for broader use with different 1D industrial codes.

4.3 Falcon to OFFBEAT coupling

The updated and generalized restart interface developed in this task of OperaHPC was tested for another industrial fuel performance code, Falcon [5]. Falcon is the reference code for fuel behaviour analysis at PSI. It is a 2D fuel behaviour code developed by EPRI and has been verified and validated to a great extent over the course of its development and usage. Falcon supports 2D axisymmetric analysis (r-z) of full-length fuel rods and can also be used for further detailed analysis of 2D slices (r- θ) at selected axial locations to study radial and angular effects. Based on a robust finite element numerical structure, Falcon is capable of analysing both steady state and transient fuel behaviour with a seamless transition between the two modes.

To test the restart interface, the BEAVRS benchmark [35] rod was used as the reference case for Falcon and subsequent OFFBEAT simulation. In order for the restart interface to work, some additional implementations and assumptions were needed for the Falcon simulation. These are discussed below before testing the restart interface with the BEAVRS rod.

4.3.1 Developments in support of the coupling

Unlike TRANSURANUS, some models are not compatible between Falcon and OFFBEAT. For example, the fission gas release model in OFFBEAT is SCIANTIX, which is not available in Falcon. OFFBEAT restart using SCIANTIX requires some specific input parameters which would not be available from Falcon. Similarly, the models for the fuel-cladding gap vary in the two codes. To deal with such compatibility issues, respective measures were adopted. For example, if the fission gas release needs to be modelled, then a parallel OFFBEAT simulation would have to be carried out to obtain the SCIANTIX fields at the time of restart. Further, the power histories and gap gas compositions, including the gas pressure are obtained from a corresponding full Falcon simulation to get the time-dependent histories and used in OFFBEAT. To facilitate this, a new gapGas model was implemented in OFFBEAT to enable time-dependent gap gas compositions and gas pressure histories.

Other encountered issues were obtaining the different components of the strain (e.g., elastic, plastic, creep) and further different creep rates contributing to the total creep strain (primary, secondary, thermal, irradiation...) from Falcon. Falcon does not provide these fields separately, however OFFBEAT requires these fields to have the correct conditions at the time of restart. To account for these inconsistencies, Falcon source code was modified to obtain the elastic and inelastic strains separately. Then, it is assumed that for a base irradiation simulation, the contribution of plasticity is significantly lower than that of creep and based on this assumption, the total inelastic strain is used as the creep strain field to be passed on to OFFBEAT.

With these assumptions and implementations, the restart interface was tested for the BEAVRS benchmark rod. The boundary conditions for the BEAVRS benchmark were obtained in Task 3.1 (reported in deliverable D3.1) and it is briefly described in the next section.

4.3.2 Testing the restart interface

As mentioned, the restart interface was tested for the BEAVRS benchmark rod. The Benchmark for Evaluation And Validation of Reactor Simulations (BEAVRS) was led by the MIT Computational Reactor Physics Group. The reactor described in BEAVRS is a Westinghouse PWR, rated at 3411 MWth, with two cycles of measured

D3.3 version 1 Procedure for input data transfer in OFFBEAT

operational data. The benchmark specification provides a detailed description of fuel assemblies, burnable absorbers, in-core fission detectors, core loading patterns, and numerous in-vessel components. The duration of the first and second cycles were 574.94 days and 315.28 days, respectively, with a cooling period of 30 days between the two. The power history, axial power profile and other boundary conditions were obtained in Task 3.1.

For the Falcon simulation, a 2D axisymmetric geometry of the full rod was considered based on the geometry specifications provided in the benchmark. The mesh was composed of 10 and 5 radial positions for the fuel and the cladding, respectively with 32 axial positions. The axisymmetric geometry is depicted in **Figure 8 (a)**.

Falcon simulation for the full rod was carried out for the first cycle of operation ($t = 574.94$ days). For the restart demonstration, the outputs from Falcon at $t = 400.64$ days were used. This Falcon output (obtained in the HDF5 file format) was used via a Python script to create the fields and other data in the format of the intermediate file map1DFieldsDict. The demonstration comprises obtaining Falcon outputs at $t = 400.64$ days and passing these to OFFBEAT 3D geometry to run the remaining of the first cycle of operation.

For the subsequent OFFBEAT simulation, we focus on the axial position no. 17 in Falcon, which is located in the centre of the rod axially. The radial dimensions of the fuel rod are used as it is but only this axial length is simulated for an OFFBEAT 3D case. The axial position height was 114.27 mm. The OFFBEAT mesh was composed of 40 cells for the fuel and 10 cells for the cladding in the radial direction, 20 cells for the axial, and 40 cells in the azimuthal direction. The 3D geometry used for the OFFBEAT simulation is depicted in **Figure 8 (b)**.

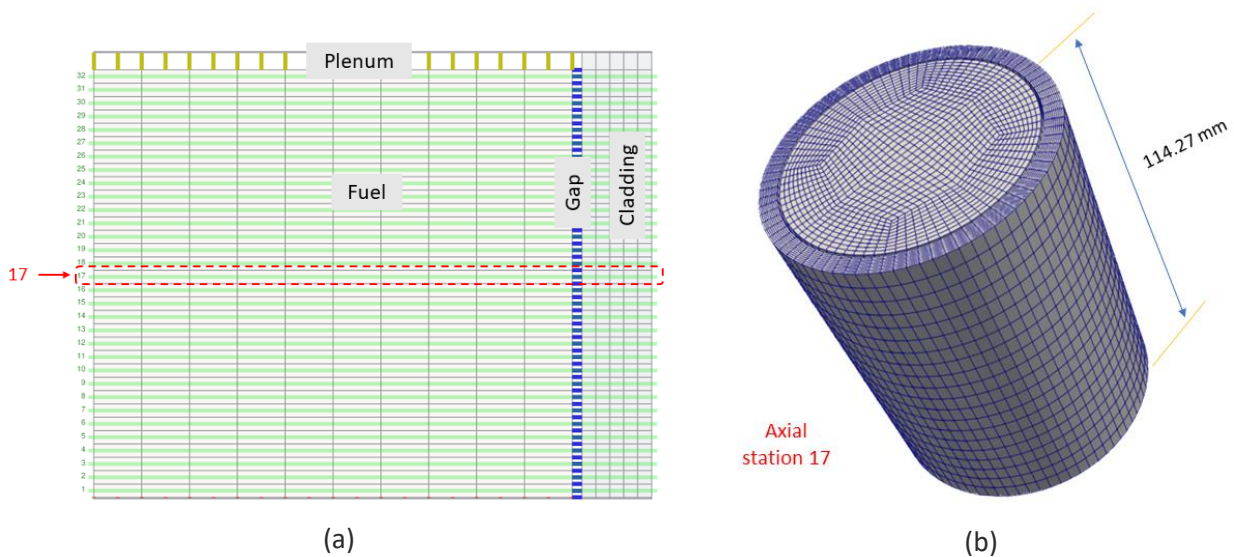


Figure 8: (a) Falcon 2D axisymmetric geometry and axial position of interest, (b) OFFBEAT 3D geometry for axial position of interest

The fields transferred from Falcon included the temperature, stress and strains, burnup, fast fluence, creep strain, along with the gap gas composition and pressure, and the power histories. The Falcon output at the end of the set restart time was compared with the OFFBEAT post-mapping initial conditions at the restart time, using the new interface. **Figure 9(a)** shows the temperature field post mapping in the OFFBEAT 3D geometry at the time of restart. The temperatures at the fuel centreline, fuel outer surface, and cladding inner and outer surfaces were compared to Falcon temperatures and found to be mapped accurately.

Figure 9(b) shows the gap gas pressure for the complete first cycle, using the Falcon simulation and with OFFBEAT after restart. The gap pressure histories are transferred from Falcon for the coupled simulation

D3.3 version 1 Procedure for input data transfer in OFFBEAT

using OFFBEAT, and the overlap of the values indicate proper implementation of the new gapGas model implemented in OFFBEAT. Similar results are observed for the power histories.

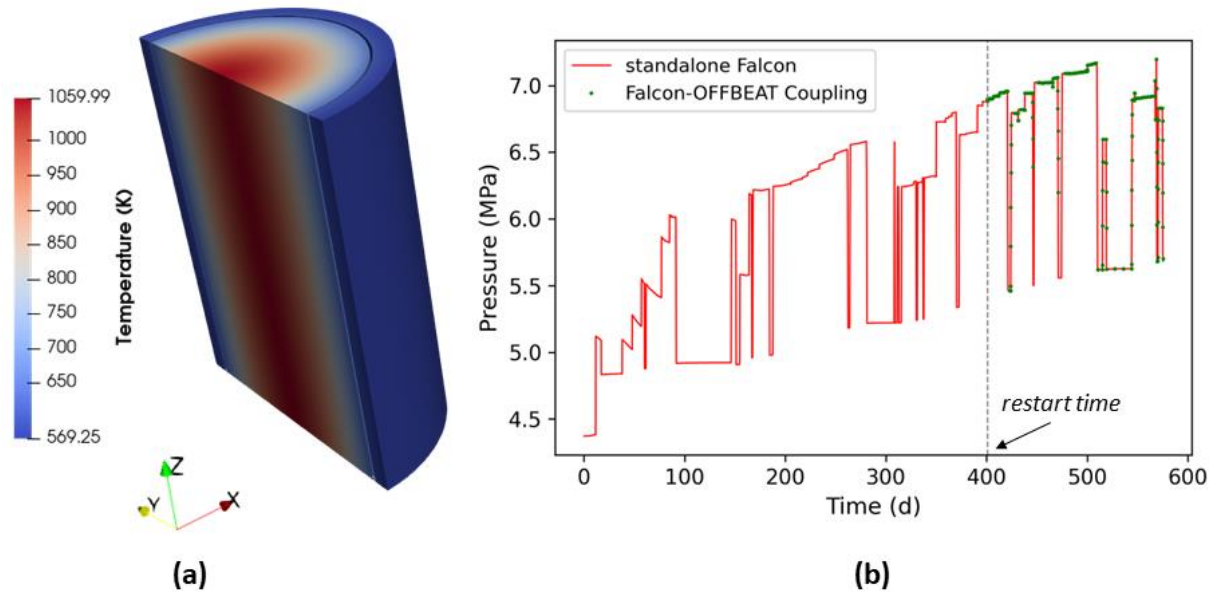


Figure 9: (a) The mapped temperature field in the OFFBEAT 3D geometry at the time of restart, (b) the gap gas pressure provided as time history.

The new and generalized utility for the restart interface was found to be efficient in transferring the fields from a Falcon run for an OFFBEAT restart. However, unlike TRANSURANUS, Falcon has some limitations on the output fields it can provide, as well as inconsistent models with OFFBEAT. This leads to results diverging from the standalone Falcon results after the restart. However, this is observed also when standalone Falcon 2Dr-z results are compared to standalone OFFBEAT 2D results for the full rod. The restart methodology, which is the focus of this task, is able to transfer the fields from an industrial fuel performance code to OFFBEAT with good precision. Further investigation would be required to generate the necessary fields from Falcon to enable a coupled simulation and to refine the results thus obtained. Once the base-irradiation results are satisfactory, the next step would be to test the coupled simulation for a Falcon base-irradiation followed by a subsequent transient calculation using OFFBEAT. These will be explored in the future and exchanges of the results will be carried out with the OperaHPC project.

5 Conclusions

The main objective of this task was to provide one-way coupling routines between industrial codes and OFFBEAT. This included coupling neutronics, thermal hydraulics, physical and chemical calculations from widely used codes with the 3D thermomechanical simulation using the OFFBEAT code. In the framework of this task, a neutronics coupling routine from Serpent to OFFBEAT was developed and presented in this deliverable. The primary step towards coupling included the proper implementation of individual features that enable the unstructured-mesh-based solution and transferral of quantities of interest. The verification of the implementation of the methodology was done against a simple proof-of-concept solution. Further, a coupling between the SCIANTIX fission gas release model and OFFBEAT was developed and a new version, SCIANTIX 2, was embedded within OFFBEAT. The execution time overhead for a SCIANTIX 2 coupled to OFFBEAT versus a default OFFBEAT calculation was found to be marginal (~12 s). The final output of this task was the development and testing of a restart interface for transferring the results of industrial fuel performance codes as initial conditions for a subsequent OFFBEAT simulation. The initial version of the restart interface was validated for the AN3 rod using TRANSURANUS as the initial fuel performance code. The restart interface was restructured and generalized for any industrial fuel performance code within this task. The new generalized restart interface was tested with Falcon fuel behaviour code and was found efficient in transferring the fields from a Falcon run to an OFFBEAT restart.

All the sub-tasks within this task were completed and reported in this deliverable. The major issues faced were reported and the possible solutions presented. Especially for the demonstration of the restart interface between Falcon and OFFBEAT, the incompatibility of models in the two codes and available output data from Falcon created several difficulties. Further investigation would be required to generate the necessary fields from Falcon to enable a coupled simulation and to refine the results thus obtained.

The developments made in this task are available in the respective domains of the codes involved. For example, the developments for the Serpent to OFFBEAT coupling will be included in the latest release of Serpent and would be part of the Kraken framework at VTT. A publication based on this work is also envisaged for the future. The OFFBEAT-SCIANTIX coupling is embedded in the OFFBEAT release version along with the different verification tests. The map1DFields utility for the restart interface along with a working example can be found in the OFFBEAT repository. A publication on the restart methodology and its demonstration is also envisaged for the future.

References

- [1] A. Scolaro, I. Clifford, C. Fiorina, A. Pautz, The OFFBEAT multi-dimensional fuel behavior solver, *Nuclear Engineering and Design* vol. 358, 110416, 2020, <https://doi.org/10.1016/j.nucengdes.2019.110416>
- [2] J. Leppänen, M. Pusa, T. Viitanen, V. Valtavirta, T. Kaltiaisenaho, The Serpent Monte Carlo code: Status, development and applications in 2013, *Annals of Nuclear Energy* vol. 82, pp. 142-150, 2015, <https://doi.org/10.1016/j.anucene.2014.08.024>
- [3] D. Pizzocri, T. Barani, L. Luzzi, SCIANITX: A new open-source multi-scale code for fission gas behaviour modelling designed for nuclear fuel performance codes, *Journal of Nuclear Materials* vol. 532, 152042, 2020, <https://doi.org/10.1016/j.jnucmat.2020.152042>
- [4] K. Lassmann, TRANSURANUS: a fuel rod analysis code ready for use, *Journal of Nuclear Materials*, vol. 188, pp. 285-302, 1992, <https://doi.org/10.1016/B978-0-444-89571-4.50046-3>
- [5] Y. Rashid, R. Dunham, R. Montgomery, Fuel analysis and licensing code: FALCON MOD01, *EPRI Report 1011308*, 2004
- [6] J. Leppänen, V. Valtavirta, A. Rintala, V. Hovi, R. Tuominen, J. Peltonen, M. Hirvensalo, E. Dorval, U. Lauranto, R. Komu, *Energies* 15, 876 (2022)
- [7] M. Pusa, J. Leppänen, *Nucl. Sci. Eng.* 164, 140 (2010)
- [8] V. Rintala, H. Suikkanen, J. Leppänen, R. Kyrki-Rajamäki, *Ann. Nucl. Energy* 77, 223 (2015)
- [9] J. Leppänen, New Stochastic Geometry Type Based on Voronoi Tessellation in the Serpent 2 Monte Carlo Code, in *Proc. M&C 2023*, Niagara Falls, Canada, Aug. 13–17 (2023)
- [10] U. Imke, V. Sanchez Espinoza, *Sci. Technol. Nucl. Install.* 2012, 465059 (2012)
- [11] R. Tuominen, V. Valtavirta, *Ann. Nucl. Energy* 180, 109447 (2023)
- [12] V. Sahlberg, A. Rintala, Development and First Results of a New Rectangular Nodal Diffusion Solver of Ants, in *Proc. PHYSOR 2018*, Cancun, Mexico, Apr. 22–26 (2018)
- [13] A. Rintala, V. Sahlberg, *Kerntechnik* 84, 252 (2019)
- [14] A. Rintala, U. Lauranto, *Ann. Nucl. Energy* 190, 109868 (2023)
- [15] M. Hirvensalo, A. Rintala, V. Sahlberg, Triangular geometry model for Ants nodal neutronics solver, in *Proc. M&C 2021*, Virtual conference, Oct. 3–7 (2021)
- [16] V. Valtavirta, A. Rintala, U. Lauranto, *Ann. Nucl. Energy* 179, 109384 (2022)
- [17] A. Rintala, V. Valtavirta, J. Leppänen, *Ann. Nucl. Energy* 164, 108603 (2021)
- [18] V. Valtavirta, J. Peltonen, U. Lauranto, J. Leppänen, SuperFINIX – A Flexible-Fidelity Core Level Fuel Behavior Solver for Multi-Physics Applications, in *Proc. NENE 2019*, Portorož, Slovenia, Sept. 9–12 (2019)
- [19] T. Ikonen, H. Loukusa, E. Syrjälähti, V. Valtavirta, J. Leppänen, V. Tulkki, *Ann. Nucl. Energy* 84, 111 (2015)
- [20] Apros process simulator website, <https://www.apros.fi/en>
- [21] U.S. Nuclear Regulatory Commission, TRACE V5.0 PATCH 6 THEORY MANUAL – Field Equations, Solution Methods, and Physical Models, Washington, DC. (2020)
- [22] R. Tuominen, R. Komu, V. Valtavirta, Coupling of TRACE with Nodal Neutronics Code Ants Using the Exterior Communications Interface and VTT's Multiphysics Driver Cerberus, in *Proc. PHYSOR 2022*, Pittsburgh, PA, May 15–20 (2022)
- [23] J. Leppänen, T. Viitanen, V. Valtavirta, *Trans. Am. Nucl. Soc.* 107, 1165 (2012)
- [24] J. Leppänen, V. Valtavirta, T. Viitanen, M. Aufiero, Unstructured Mesh Based Multi-Physics Interface for CFD Code Coupling in the Serpent 2 Monte Carlo Code, in *Proc. PHYSOR 2014*, Kyoto, Japan Sep. 28 - Oct. 3 (2014)
- [25] V. Valtavirta, J. Leppänen, T. Viitanen, *Ann. Nucl. Energy* 100, 50 (2017)
- [26] J. Leppänen, *Nucl. Sci. Eng.* 174, 318 (2013)
- [27] T. Viitanen, J. Leppänen, *Nucl. Sci. Eng.* 171, 165 (2012)
- [28] T. Viitanen, J. Leppänen, New Interpolation Capabilities for Thermal Scattering Data in Serpent 2, in *Proc. PHYSOR 2016*, Sun Valley, ID, May 1–6 (2016)

D3.3 version 1 Procedure for input data transfer in OFFBEAT

- [29] J. Leppänen, On-the-fly Temperature Treatment Routine for Unresolved Resonance Probability Table Sampling in Serpent 2, in Proc. M&C 2025, Denver, CO, Apr. 27-30 (2025).
- [30] J. Leppänen, A. Isotalo, Burnup calculation methodology in the Serpent 2 Monte Carlo code, in Proc. PHYSOR 2012, Knoxville, TN, Apr. 15–20 (2012)
- [31] M. García, J. Leppänen, V. Sanchez-Espinoza, Ann. Nucl. Energy 152, 108026 (2021)
- [32] J. Leppänen, Ann. Nucl. Energy 176, 109259 (2022)
- [33] J. Leppänen, M. Aufiero, Development of an Unstructured Mesh Based Geometry Model in the Serpent 2 Monte Carlo Code, in Proc. PHYSOR 2014, Kyoto, Japan Sep. 28 - Oct. 3 (2014)
- [34] A. Scolaro, P. Van Uffelen, A. Schubert, C. Fiorina, E. Brunetto, I. Clifford, A. Pautz, Towards coupling conventional with high-fidelity fuel behavior analysis tools, Progress in Nuclear Energy, vol 152, 104357, 2022, <https://doi.org/10.1016/j.pnucene.2022.104357>
- [35] N. Horelik, B. Herman, B. Forget, K. Smith, Benchmark for Evaluation And Validation of Reactor Simulations (BEAVRS), v1.0.1, Proc. Int. Conf. Mathematics and Computational Methods Applied to Nuc. Sci. & Eng., Sun Valley, Idaho 2013, <https://github.com/mit-crpg/BEAVRS>

Appendix A: Restart Variables

The following table lists the fields/variables that need to be transferred from the industrial fuel performance codes to OFFBEAT via the intermediate file. Here,

- Description: Name of the field.
- Label: Name of the field as used in OFFBEAT.
- Units: Unit of the field (preferably in SI units, although a unit conversion multiplier would be provided in the intermediate file for each field).
- From 1.5D Code: Defines whether the field is derived from the industrial fuel performance code or not. If not, then the data must be defined by the user.
- Initial Condition: Is this variable an initial/restart condition for the downstream OFFBEAT simulation? If so, then the value at the time of restart is needed.
- Boundary Condition: Is this variable a boundary condition in the downstream OFFBEAT simulation? If so, then time-dependent data for the whole simulation will be required.
- Comment: Any comment regarding the field/variable

Description	Label	Units	From 1.5D Code	Initial Condition	Boundary Condition	Comment
Fuel/cladding temperatures	T	K	y	y		
Cladding outer surface temperature BCs	-	K			y	
Axial power history (LHGR)	-	W/m			y	Axial LHGR
Burnup	Bu	MWd/tHM	y	y		
Fast fluence	fastFluence	n/m2	y	y		
Coolant pressure history	-	Pa			y	
Fuel/cladding stress - axial	sigma_z	Pa	y	y	y	If we don't assume plain strain, then we need to supply an axial stress BC.
Fuel/cladding stress - radial	sigma_r	Pa	y	y		
Fuel/cladding stress - azimuthal/hoop	sigma_hoop	Pa	y	y		
Relocation strain	epsilonRelocation	-	y	y		
Fuel/cladding swelling strain	epsilonSwelling	-	y	y		
Fuel densification (sintering) strain	epsilonDensification	-	y	y		

D3.3 version 1 Procedure for input data transfer in OFFBEAT

Creep strain - radial	epsilonCreep_r	-	y	y		
Creep strain - axial	epsilonCreep_z	-	y	y		
Creep strain - hoop	epsilonCreep_hoop	-	y	y		Theta (hoop) component will need to be calculated somehow if not available from the 1.D code
Plastic strain - radial	epsilonPlastic_r	-	y	y		
Plastic strain - axial	epsilonPlastic_z	-	y	y		
Plastic strain - hoop	epsilonPlastic_hoop	-	y	y		Theta (hoop) component will need to be calculated somehow if not available from the 1.D code. If not available, e.g. code doesn't provide separate plastic/creep strain outputs then we can probably assume plastic strain is zero, i.e. only creep occurs during normal operation.
Gap gas pressure history	P_gap	Pa	y		y	
Gap gas composition (Helium)		-	y		y	
Gap gas composition (Xenon)		-	y		y	
Gap gas composition (Argon)		-	y		y	
Gap gas composition (Krypton)		-	y		y	
Gap gas composition (Neon)		-	y		y	
Gap gas composition (Radon)		-	y		y	
SCIANTIX variables:						* Not available from all 1.5D codes due to different FGR models being used. If not available, we can run OFFBEAT and the 1.5D code in parallel to generate the necessary data.
Fission gas produced	Gas_produced	mol/m3	y*	y		

D3.3 version 1 Procedure for input data transfer in OFFBEAT

Fission gas produced in grain	Gas_grain	mol/m3	y*	y		
Fission gas produced in grain solution	Gas_grain_solution	mol/m3	y*	y		
Fission gas produced in grain bubbles	Gas_grain_bubbles	mol/m3	y*	y		
Intragranular bubble concentration	Intragranular_bubble_concentration	n/m3	y*	y		
Intragranular bubble radius	Intragranular_bubble_radius	m	y*	y		
Intergranular bubble concentration	Intergranular_bubble_concentration	n/m3	y*	y		
Intergranular bubble radius	Intergranular_bubble_radius	m	y*	y		
Intergranular bubble area	Intergranular_bubble_area	m ²	y*	y		
Intergranular bubble volume	Intergranular_bubble_volume	m ³	y*	y		
Intergranular atoms per bubble	Intergranular_atoms_per_bubble	at/n	y*	y		
Intergranular vacancies per bubble	Intergranular_vacancies_per_bubble		y*	y		
Fission gas released	Gas_released	mol/m3	y*	y		
Fission gas in grain boundary	Gas_boundary	mol/m3	y*	y		
Helium produced	He_produced	mol/m3	y*	y		
Helium in grain	He_grain	mol/m3	y*	y		
Helium in grain boundary	He_boundary	mol/m3	y*	y		
Xenon produced	Xe_produced	mol/m3	y*	y		
Xenon in grain	Xe_grain	mol/m3	y*	y		
Xenon in grain boundary	Xe_boundary	mol/m3	y*	y		
Krypton produced	Kr_produced	mol/m3	y*	y		
Krypton in grain	Kr_grain	mol/m3	y*	y		
Krypton in grain boundary	Kr_boundary	mol/m3	y*	y		
Intragranular gas swelling	intragranularGasSwelling	-	y*	y		

D3.3 version 1 Procedure for input data transfer in OFFBEAT

Intergranular gas swelling	intergranularGasSwelling	-	y*	y		
Effective burnup	Effective_burn_up	MWd/tHM	y*	y		
General and geometric data:						
Restart time	time	s	y	y		The time at which OFFBEAT restart is initiated.
Fuel inner radius	riFuel	m				Geometrical data
Fuel outer radius	roFuel	m				r-z tables for the fields would be defined at these locations
Cladding inner radius	riClad	m				
Cladding outer radius	roClad	m				
Axial locations	axialPositions	m				
Radial locations	radialPositions	m				

Appendix B: Intermediate file format

The intermediate file is to be named *map1DFieldsDict*. It is in the OpenFOAM dictionary format. The format of the files looks like as shown in **Figure 10** below.

```

/*-----*- C++ -*-----*\
=====
\\ / F i e l d           | OpenFOAM: The Open Source CFD Toolbox
\\ / O p e r a t i o n   | Website: https://openfoam.org
  \\ / A n d              | Version: 9
  \\ / M a n i p u l a t i o n |
/*-----*\

FoamFile
{
  format      ascii;
  class       dictionary;
  location    "system";
  object      map1DFieldsDict;
}
// ***** //

// Unit conversion factors for length (to m) and time (to s)
convertToMeters      0.001;
convertToSeconds     3600.000;

// General Info
nAxialSlices 0;
time 28758.000;

// Geometric Data
riFuel 0.00;
roFuel 0.0039218;
riClad 0.0040005;
roClad 0.0045720;
axialPositions ( 48.900 97.800 146.700 195.600 244.500 286.000 );
radialPositions ( 0.0001961 0.0013726 0.0021570 0.0029414 0.0037257 0.0040576 0.0042862 0.0045148 );

fields
{
  Sigma
  {
    radialData
    (
      ( 0.000000000E+00 3.0692902217E-03 6.1331048484E-03 ... -4.9590122489E-02 )
      ...
    );

    tangentialData
    (
      ( 0.000000000E+00 0.000000000E+00 0.000000000E+00 ... 0.000000000E+00 )
    );

    axialData
    (
      ( 4.1133328544E-02 4.1133328544E-02 4.1133328544E-02 ... -6.9410956948E-02 )
    );
  }
}

```

Figure 10: Example snapshot of the format of the intermediate file.

The header of the file is the generic OpenFOAM dictionary header. The Unit conversion factors enable to set the units of space and time for the simulation case. For example, if the units of axial and radial locations are in mm, then `convertToMeters 0.001`; scales the unit to m. Next, the General info sets the time at which the

D3.3 version 1 Procedure for input data transfer in OFFBEAT

restart begins. The transformed fields resulting from the map1DFields utility would be saved to the time directory with this (restart) time. The Geometric data sets the inner and outer radii of the fuel and cladding and the axial and radial positions at which the fields are tabulated in 2D r-z format. Finally, the fields sub-dictionary sets the different fields needed to be transferred from the industrial fuel performance code to OFFBEAT. A more detailed description of the format to use for the fields is shown in **Figure 11**.

```
fields
{
  Sigma
  {
    name      sigma;
    type      symmTensor;
    unit      [1 -1 -2 0 0 0];
    conversionFactor  1.0;

    radialData
    (
      ( 1.3546849800e+08  1.3057375491e+08  1.2157704477e+08  ...  -1.4220204745e+07 )
      ( 2.0780077767e+08  5.0098385113e+07  3.6372093013e+07  ...  -1.5429618832e+07 )
      ( 2.0715983267e+08  4.6663046529e+07  3.1579043686e+07  ...  -1.4265158027e+07 )
      ( 2.0673323999e+08  4.5545178918e+07  2.9370160934e+07  ...  -1.4875374800e+07 )
      ( 2.0682971515e+08  4.5206745757e+07  2.8540335930e+07  ...  -1.4097798914e+07 )
      ( 2.0719216289e+08  4.8445797435e+07  3.3743327686e+07  ...  -1.4367373315e+07 )
      ( 1.2195632925e+08  1.1726700220e+08  1.0976963403e+08  ...  -1.2300369723e+07 )
    );
  }
};
```

Figure 11: Definition of fields in the 2D r-z table format.

Each of the fields has five attributes: name, type, unit, conversionFactor and data. For the example shown in Figure 11, the field Sigma has attributes:

- name: sigma, which is the name of the field in OFFBEAT,
- type: symmTensor, as stress is a symmetric tensor in OFFBEAT,
- unit: the dimensional unit in [MASS, LENGTH, TIME, TEMPERATURE, MOLES, CURRENT, LUMINOUS INTENSITY],
- conversionFactor: for unit conversion (eg., 1 for Pa or 1e6 for MPa),
- data: the 2D r-z data table. For vector/tensor fields, the data is defined as the components: radialData, axialData and tangentialData. For scalar fields, it is only data.

A complete working example of map1DFieldsDict can be found along with the utility in the git repository: [utilities/map1DFields · development/FalcOFFBEAT · foam-for-nuclear project / OFFBEAT · GitLab](https://github.com/operahpc/foam-for-nuclear-project/tree/development/FalcOFFBEAT)