Developing an automated fluid activation residence time CFD database

Plasma Physics and Controlled Fusion

OPEN ACCESS

RECEIVED
31 December 2025

REVISED
17 March 2026

ACCEPTED FOR PUBLICATION
8 April 2026

PUBLISHED
20 April 2026

PAPER

Developing an automated fluid activation residence time CFD database

L R Humphrey*, T A Berry, A J Dubas, C L Grove, C R Shand and R Worrall

United Kingdom Atomic Energy Authority, Culham Campus, Abingdon, Oxfordshire OX14 3DB, United Kingdom

* Author to whom any correspondence should be addressed.

E-mail: luke.humphrey@ukaea.uk

Keywords: machine learning, CFD, fluid activation

Abstract

Fluid activation is of critical importance to fusion power plant design, as it impacts dose rates to maintenance personnel and equipment, heating in sensitive components, and radionuclide inventories with implications for accident scenarios and waste. The levels of fluid activation in a system are dependent on the neutron flux spectrum and the exposure time of the fluid. Accurately modelling the fluid irradiation history for a pipe system requires computational fluid dynamics (CFD) to determine the residence time distribution (RTD) of fluid particles passing through each component. However, performing CFD on whole pipe systems can be computationally expensive which limits frequency of design iterations.

To address this concern, a new code was developed: FARBASE (the Fluid Activation Residence time dataBASE). This paper details the development and functionality of FARBASE, focusing on its two key features:

- An automated CFD pipeline which accepts a parametric description of a pipe component under given flow conditions, generates the pipe geometry, runs a steady-state OpenFOAM simulation, and returns the resulting RTD.
- A Gaussian process regression (GPR) surrogate model which can be trained on the CFD database and queried to provide uncertainty quantified predictions of RTDs. Where the uncertainty of the GPR prediction exceeds a given threshold, FARBASE can automatically perform additional CFD to update the database, improving the accuracy of future predictions.

Work is ongoing to utilise FARBASE in UKAEA's GammaFlow fluid activation code, providing RTDs which are used to determine the production and decay rates of key radionuclides in each component of a basic water circuit. This will be extended in future to model benchmark experiments and validate the combined tool, which aims to provide an efficient and standardised approach to modelling activation in complex fluid circuits.

1. Introduction

In fusion devices, materials surrounding the plasma become activated by exposure to high-intensity, high-energy neutron fluxes. This exposure creates radionuclides which decay and emit secondary gamma rays and electrons, leading to a residual radiation field which can persist long after the device shuts down.

The resulting shut-down dose rate (SDDR) is an important safety consideration. In active nuclear fission power plants, SDDRs are the majority source of radiation exposure to workers [1]. Future fusion power plants (FPPs) will also need to design for acceptable SDDRs.

Table 1. Neutron capture reactions on nuclides present in water. (Fusion-relevant reactions are in bold).

Reaction	Radionuclide	Half-life (s)	Threshold (MeV)	Decay radiation
^{16}O(n,p)	**^{16}N**	**7.1**	**10**	**γ (6.1, 7.1 MeV)**
^{16}O(n,2n)	^{15}O	122	20	γ (0.5 MeV)
^{17}O(n,p)	**^{17}N**	**4.2**	**9**	**n (0.4, 1.2 MeV)**
^{18}O(n,γ)	^{19}O	26	< 1	γ (0.2, 1.4 MeV)

To ensure that FPPs are operated and maintained safely, radioactivity must be monitored. However, to design a commercially-viable FPP, the nature of activation must also be accurately predicted and mitigated in the initial design. Simulation provides a cost-effective route to predictive design.

Typically, SDDR calculations assume material is static during irradiation and decay. However, this is not the case for moving fluids which experience a changing neutron flux during operation, and can also transport activated material outside of primary shielding. This has implications for dose rates to workers, heating in sensitive components and electronics, waste management and compliance, and in accident scenarios for example loss of coolant accidents.

In fusion devices, flowing materials likely to be present include:

- Coolants
  - H_2O (water)
  - CO_2 (carbon-dioxide)
  - He (helium)
  - N_2 (nitrogen)
- Breeders
  - PbLi (lithium-lead)
  - Molten Salts (FLiBe, etc)

In this work, we focus specifically on water, whose activation by neutron exposure is shown in table 1. Transmutation of ^{16}O primarily generates ^{16}N and ^{17}N. ^{16}N produces high-energy gammas which contribute directly to dose rate, while ^{17}N produces lower-energy neutrons which cause secondary activation of materials around the radioisotope at the point of decay. The short half-lives of these radionuclides (7.1 s and 4.2 s respectively) mean that the exact flow behaviour of the system has a significant impact on the resultant activity; accurate fluid modelling is important for an accurate activity calculation.

Computational fluid dynamics (CFD) simulations enable the distribution of flow paths in a fluid system to be predicted. Previous works [2–5] have showed that incorporating CFD into fluid activation calculations improved model accuracy: providing a closer match to experimental results compared to the conventional approach (assuming a flat flow distribution throughout the fluid circuit).

In particular, CFD significantly improves fluid activation modelling for components with a variation in flow rates, whether due to laminar flows or turbulent flows through certain components, such as bends and tanks. Residence times are not constant in such components, that is to say: the length of time a given fluid particle spends in a given section of pipe varies and can be tallied in a distribution. These CFD-derived residence time distributions (RTDs) can then be utilised in fluid activation calculations to better account for fluid dynamic effects.

Despite its advantages, CFD has limitations. Fluid modelling is expensive, both computationally and in terms of the time required to set up and analyse the simulation. Modelling entire complex fluid circuits at high fidelity can be prohibitively expensive, and this approach limits frequency of design iterations since each small change to one part of the system requires a full re-run of the entire system analysis. In addition, CFD is not required where a flat flow distribution is known to be an adequate approximation, such as the fast-flowing, turbulent, straight cylindrical pipe sections which make up a significant proportion of many coolant water systems.

Thus, a need was identified for a surrogate modelling approach which can provide a sufficiently accurate approximate solution for residence times in parametric pipe sections relatively quickly and cheaply.

In the present work, the Fluid Activation Residence time dataBASE (FARBASE) is described. In FARBASE, Gaussian process regression (GPR) is used to fit a statistical model to available data. The advantages of a GPR model (compared to alternatives, such as a neural network) are that they are quick

to train and provide uncertainty-quantified predictions which can be used as a metric to determine where additional CFD data would be most useful. In addition, GPRs are considered to scale well up to ∼20 dimensions in most successful applications [6], which is sufficient dimensionality to describe the majority of pipe system components.

FARBASE provides two key functions:

- A CFD pipeline for calculating RTDs from parametrically described pipe components using Gmsh [7], OpenCascade [8] and OpenFOAM [9].
- A surrogate modelling module for interpolating between data by fitting to a database of previously-calculated RTDs.

FARBASE is intended for use by fluid activation codes, namely GammaFlow [2], to efficiently incorporate CFD-derived RTDs into fluid activation calculations at a reduced computational cost.

FARBASE itself makes no assumptions about the irradiation conditions of the pipe system being modelled, simply outputting the RTD for a given section. When using these RTDs in fluid activation calculations, it is implicitly assumed that the irradiation intensity is spatially invariant in these sections. Where spatial variance is expected to be significant, a more bespoke treatment of that section may be required.

2. Methods

The FARBASE code is written in Python. The core surrogate modelling and RTD prediction functionality may be used standalone if the user provides an appropriate training database. However, when used alongside an OpenFOAM installation, FARBASE can automatically deploy simulations to build a training database or extend an existing one.

A compatible instance of OpenFOAM must be installed in the computing environment being used to deploy the pipeline. FARBASE has been developed using the OpenFOAM Foundation (.org) distribution, for which the latest version at the time of writing is OpenFOAM v13 [9]. It is recommended to use this version with FARBASE, though other versions may remain compatible.

FARBASE is not yet packaged with training data, though it is intended to do so in a future version. At present, the user must first construct an initial training database, then make predictions with a set uncertainty threshold which, when exceeded, will trigger additional CFD to extend the database and improve prediction accuracy in the region of the queried point in parameter space.

2.1. FARBASE usage

The expected user workflow is shown in figure 1. Stepping through the flowchart:

- First, a GPR model is initialised by selecting a component from the `components` module. Each component class contains the required information (i.e. the input parameters and the number of outlets) required to initialise the surrogate model with the appropriate dimensions.
- Then, the user may either load an existing dataset or generate a new one. Data will be checked for validity against the chosen component, ensuring compatibility. In most cases, this requires that data was generated using that component type, though some cross-compatibility is present. For example, a variable-angle elbow database may be used when specifying a fixed 90° elbow component, in which case only data from simulations with that angle will be loaded.
- To generate data, the user passes in the number of data they would like to generate alongside the bounds for each dimension in the parameter space for their selected component. Inputs consist of the geometric parameters required to describe the component (for example length and radius) alongside flow parameters required for OpenFOAM's `incompressibleFluid` solver (kinematic viscosity and volumetric flow rate). A choice of either Latin hypercube or Sobol sampling are used to select candidates for simulation, each ensuring a good spread of data throughout the bounded parameter space. The candidate simulations are then deployed in sequence.
- Simulations of straight cylindrical pipes may use an analytical solution as a rapid proxy for simulation. This feature is primarily intended for developer usage. For all other simulations, an OpenFOAM case directory is created and populated according to user settings, the pipe component geometry is then generated which forms the basis for the CFD mesh. The simulation is then run. Once converged,

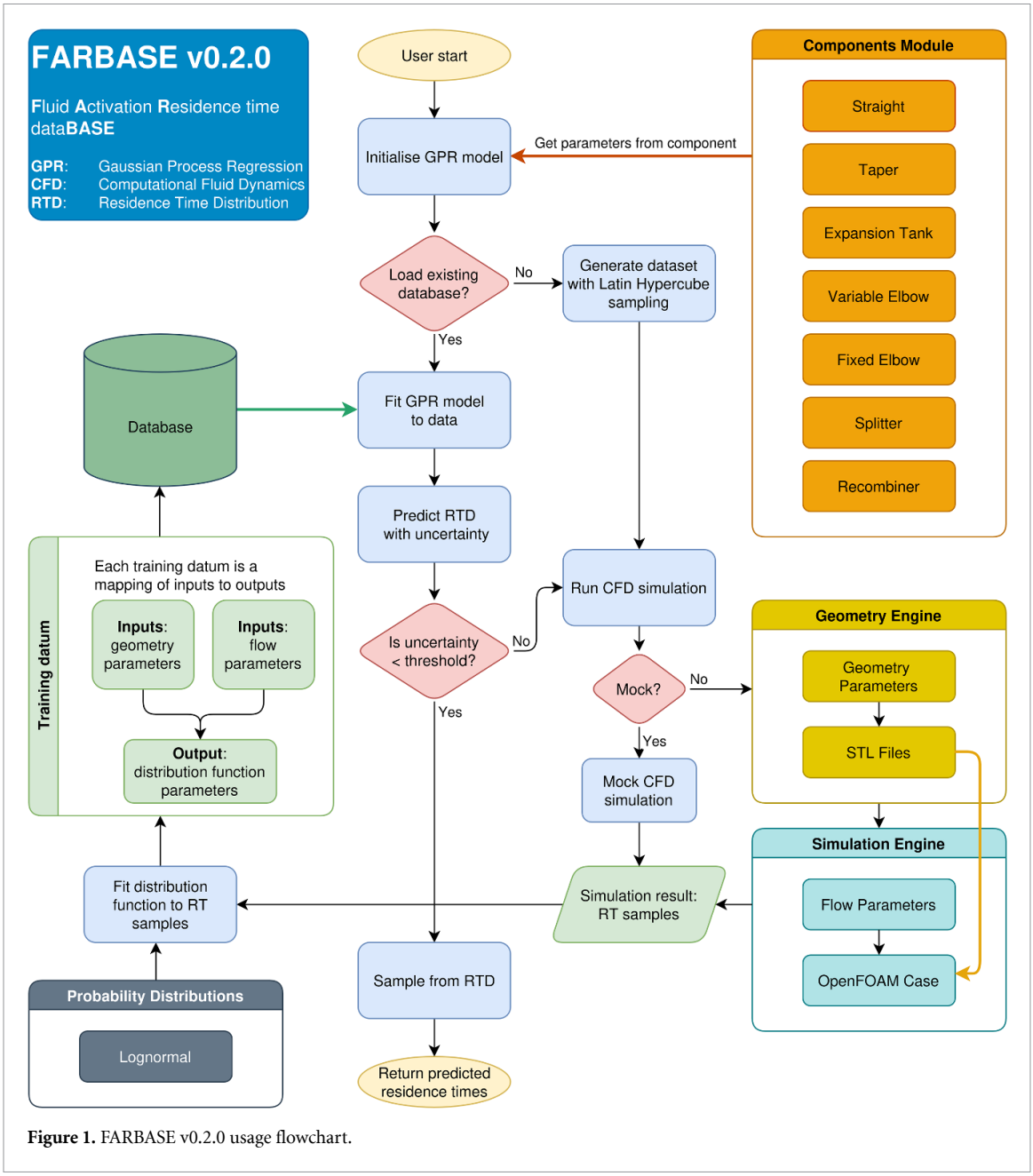


Figure 1. FARBASE v0.2.0 usage flowchart.

a RTD is sampled from each outlet on the component and used to fit a probability density function (PDF).

- For each simulation, the training data extracted is the mapping of the inputs (geometric and flow parameters) to the outputs (the PDF parameters describing the RTD for each outlet). Thus, while simulation data may be retained at the user's discretion, the storage footprint for the FARBASE training database itself is considerably reduced, consisting of a small number of scalar values per data point.
- Once training data is loaded to memory, the GPR model can be fit and queried to generate predicted RTDs by interpolating between available data.
- Each prediction is provided alongside its coefficient of variation; if this metric is below the chosen threshold (by default, 25%) the user will be prompted to deploy an additional simulation which will automatically extend the database. If this is done, the prediction will then be recalculated based on the additional data.
- The GPR prediction takes the form of a PDF, but the user may also request a discrete number of residence times to form a distribution, in which case these will be randomly sampled from the PDF.

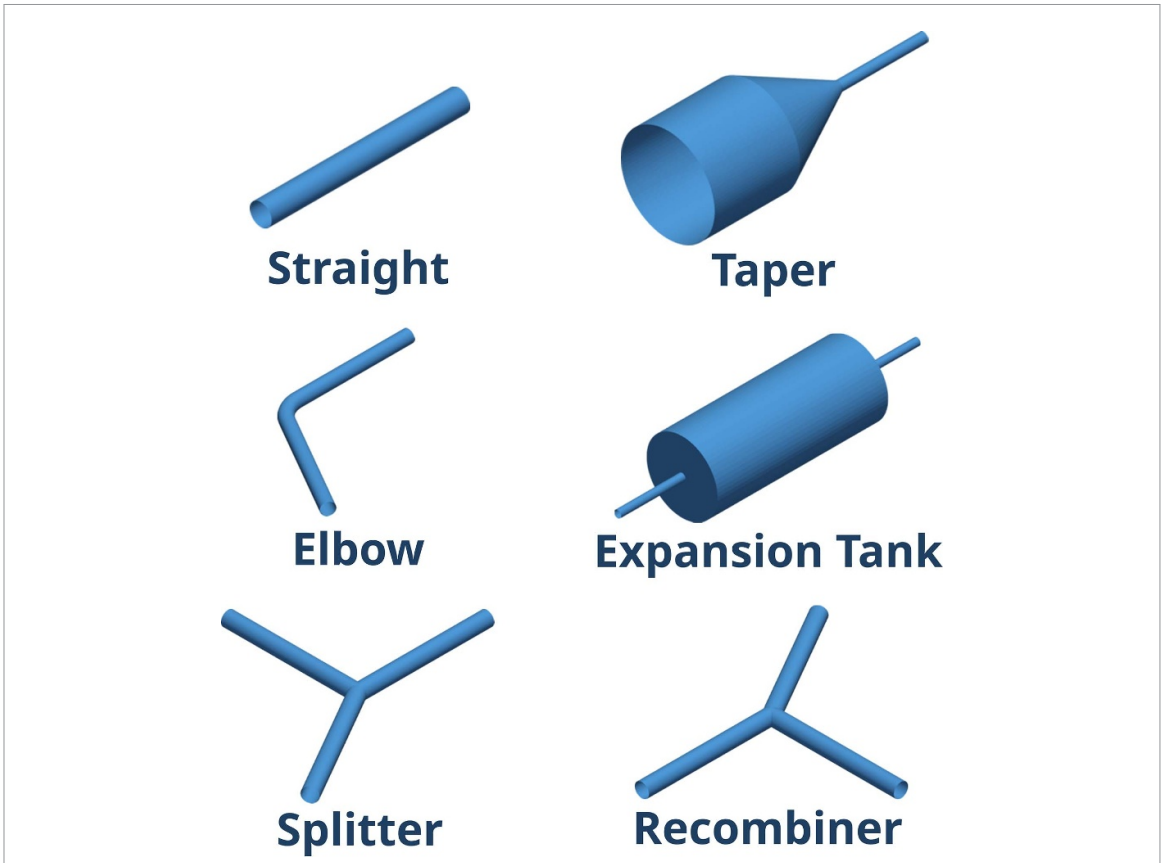


Figure 2. 3D geometries for each component type available in FARBASE.

2.2. Components

The `components` module is comprised of the user-facing classes for each component type, as visualised in figure 2. Each class links together the associated parameter names, geometry class, and probability distribution associated with that component type to be provided to the surrogate model and CFD pipeline. The components included in the module are as follows:

- `Straight`, a straight cylindrical pipe section
- `Taper`, a truncated conical pipe section with cylindrical inlet and outlet sections of different radii
- `ExpansionTank`, a cylindrical in-line expansion tank, also known as a delay tank
- `VariableElbow`, a cylindrical elbow section of variable angle ranging from 0°–180°
- `FixedElbow`, a cylindrical elbow section of fixed angle, used for building a surrogate model for common elbow angles such as 45°, 90°, and 180°
- `Splitter`, a cylindrical pipe splitter with one inlet and two outlets
- `Recombiner`, a cylindrical pipe recombiner with two inlets and one outlet

2.3. Automated CFD pipeline

Users need only interact with the `GaussianProcessor` class and the `component` library. The CFD workflow described in this section is handled automatically by FARBASE.

2.3.1. Geometry

The first module in the CFD pipeline is `geometry`, whose primary function is to generate surface geometry files required to set up the simulation. This functionality utilises the Gmsh CAD engine [7], combining native features with Gmsh's wrapping of OpenCascade [8].

An abstract base class `GeometryABC` defines the interface to be used by every geometry subclass. Each subclass is associated with a particular pipe system component geometry. Once a subclass is instantiated from user-supplied dimensions, the `write_stls` method may be called to generate STL files describing the surface geometry of the component's inlet(s), outlet(s), and walls.

Table 2. OpenFOAM simulation parameters for the automated CFD engine.

Parameter	Field	Value
Transport model		`Newtonian`
Turbulence model		`kOmegaSST`
Time scheme		`steadyState`
Grad scheme		`Gauss linear`
Div scheme	`U`	`bounded Gauss limitedLinearV`
	`k`	`bounded Gauss limitedLinear`
	`omega`	`bounded Gauss limitedLinear`
	`age`	`bounded Gauss upwind`
Tolerance	`p`	10^{-6}
	`U k omega`	10^{-5}
	`age`	0.001
Relaxation factors	`p`	0.3
	`U k omega`	0.7
	`age`	1
Residual controls	`p`	10^{-4} (`lofi`) or 10^{-6} (`hifi`)
	`U k omega`	10^{-4} (`lofi`) or 10^{-6} (`hifi`)

In addition, each geometry class is able to return several associated variables required by the simulation set up:

- The number of inlets.
- The number of outlets and their directions.
- The coordinates of a grid of points on each outlet.
- The narrowest and widest diameters.
- The number of symmetry planes.
- The coordinates describing a bounding box surrounding the geometry.
- The coordinates of an arbitrary point within the geometry.

Depending on the geometry of a given component, there may be 0–2 planes of symmetry; it is often desirable to exploit these when running the simulation. To support this, `domain` includes the flag `use_symmetry_planes` (on by default) which causes the method to return either the full domain, the positive half or positive quadrant (as appropriate). When the simulation mesh is constructed, only the subsection of the pipe volume inside both the surface geometry and the domain will be meshed and the appropriate planes will be treated as symmetric boundaries.

2.3.2. Simulation

The second module in the CFD pipeline is `simulation`, which handles construction of the OpenFOAM case directory, writing required files, deployment of the simulation, and reading of simulation outputs from file.

FARBASE CFD is solved for steady-state incompressible-flow, second order in space, using OpenFOAM's native `incompressibleFluid` solver. Additional details on the simulation parameters are shown in table 2.

Regardless of component, the simulations follow the same sequence: `blockMesh`, `snappyHexMesh`, `decomposePar`, `foamRun`, `reconstructPar`, `foamPostProcess`. (With the exception that when running in serial, `decomposePar` and `reconstructPar` are omitted.)

Stepping through this sequence, first a cuboid background mesh is generated using `blockMesh`. The domain is given by the component's `Geometry` subclass, accounting for symmetry planes and applying a two cell buffer on all other sides to ensure the component walls are not coincident with the mesh at this stage. The mesh is comprised of square cells; the cellsize is set to the smallest of two values:

- The size corresponding to four cells across the component's narrowest diameter.
- The size corresponding to a target number of cells across the widest diameter. This target is 16 in `lofi` mode and 32 in `hifi` mode.

Second, a castellated mesh is generated by running `snappyHexMesh` with the surface geometry files. This function performs iterations of deletion and refinement for cells beyond and coincident with the surface respectively. In `lofi` mode, two levels of castellation are used, while in `hifi` mode, four are used. In each mode, mesh quality is controlled according to the defaults provided in `meshQualityDict`.

Third, if multiple processes are available, the mesh is separated into a number partitions equal to the number of processes by `decomposePar` using the `scotch` algorithm [10]. Then, the solver is executed by `foamRun`, until convergence criteria are met. If `decomposePar` was run earlier, the solved mesh is then reconstructed with `reconstructPar`.

Finally, the `age` field is computed for the latest timestep (i.e. the steady-state solution) according to equation (1):

$$\nabla \cdot (\phi t) = 1 \tag{1}$$

where ϕ volumetric flux and t is the age, or the time taken for a fluid particle to travel from an inlet to the location. (Note that diffusion is not accounted for in this formula, so its validity is limited to cases where diffusion transport is dominated by momentum transport, which is broadly the case for turbulent fluids.) Velocity and age values are then sampled for a grid of `boundaryProbes` on each outlet and written to CSV file.

The resulting samples constitute a spatial distribution of residence times corresponding to the solution, alongside the velocity values which may be used as weights to calculate the corresponding temporal distribution.

Following the OpenFOAM sequence, this weighted resampling is performed within FARBASE (using Pandas [11]) and a PDF is fit to the resulting distribution to acquire the reduced form used by the FARBASE surrogate model. Which PDF to use is given by the component type, which points to one of the classes from FARBASE's `distributions` library. In the present version, however, all components use an x-offset log-normal distribution. FARBASE has been developed with extensibility of supported distribution functions in mind, anticipating that complex components may require more complex functions to fit to the resulting distributions.

2.3.3. Mock simulation

To support the testing and development of FARBASE, a module was developed to mock an OpenFOAM CFD simulation by sampling residence times from an analytical solution to fluid flow through a straight cylindrical pipe section.

Specifically, the Hagen–Poiseuille formula [12] is used to provide fluid particle velocity v as a function of radial distance r from the pipe central axis for laminar flows. By applying a power law, it is also possible to derive an approximate formula for the radial profile of turbulent flows. The Reynolds number Re is used to determine whether a flow is laminar or turbulent. For a cylindrical pipe of radius R, Re is calculated as shown in equation (2):

$$Re = \frac{QD_{\mathrm{H}}}{A\nu} \tag{2}$$

where Q is the volumetric flow rate, D_{H} is the hydraulic diameter (which for a cylindrical pipe is just the internal diameter $D = 2R$), $A = \pi R^2$ is the cross-sectional area, and ν is the kinematic viscosity. Thus, the Reynolds number can be determined from the available input parameters.

The flow is considered turbulent when $Re \geqslant 2300$, else laminar. Transitional regimes are not considered. The maximum velocity for each regime is given by equation (3), which is used to calculate the radial profile given by equation (4),

$$v_{\max} = \begin{cases} 2\frac{Q}{A}, & Re < 2300 \\ \frac{(n+1)(2n+1)}{(2n^2)} \cdot \frac{Q}{A}, & Re \geqslant 2300 \end{cases} \tag{3}$$

$$v(r) = \begin{cases} v_{\max} \cdot \left(1 - \left(\frac{r}{R}\right)^2\right), & Re < 2300 \\ v_{\max} \cdot \left(1 - \frac{r}{R}\right)^{\frac{1}{n}}, & Re \geqslant 2300 \end{cases} \tag{4}$$

where $v_{\max} = v(r = 0)$ is the maximum velocity which occurs at the centre of the pipe and n is the flow index, often taken as $n = 7$ as a rule of thumb [12].

The residence time t of a particle can then be given as a function of r by dividing the pipe length L by the velocity $v(r)$, shown in equation (5):

$$t(r) = \frac{L}{v(r)}. \tag{5}$$

From this, a RTD is acquired by randomly sampling residence times from a grid of points on the pipe outlet, as weighted by $v(r)$.

Although developed primarily for testing and development purposes, this module may in principle be used in analysis, which could allow compute budget to be focussed on more complex components. However, validation work is required before an assessment of the approximation's accuracy can be given.

2.4. Surrogate model

FARBASE's surrogate modelling functionality is handled by the `gaussianprocess` module, which consists of a single class `GaussianProcessor`. This class contains the user-facing methods for saving and loading datasets, building the surrogate model, querying the model with a set of inputs, sampling from predicted RTDs, and deploying simulations to update the database.

To instantiate the class, it must be passed an instance of a `PipeSystemComponent` subclass. The optional input `num_sim_procs` may also be passed here, which provides the number of MPI processes to use for any deployed simulations. Note that the surrogate modelling itself does not require parallel processing; this value is only used for deployed simulations. The Gaussian processes are implemented using Scikit-learn [13]. One instance of `GaussianProcessRegressor` is used per outlet on the component, such that the mapping between inputs and outlet RTDs is treated as an independent black box function for each outlet. Each GPR uses a standard radial basis function kernel `RBF` alongside a white noise kernel `WhiteKernel` to handle any stochastic noise induced by random sampling. Inputs are preprocessed using `StandardScaler` and `normalize_y` is set to `True` to avoid the impact of unit length scales on the GPR fit.

3. Preliminary study

In this section, a preliminary study is presented for an expansion tank component. As discussed in section 1, this component is known to feature recirculating currents which have a significant impact on fluid activation.

The OpenFOAM case is generated via the FARBASE automated CFD pipeline described in section 2.3, using the `lofi` setting. The simulation input parameters are shown in table 3. The mesh generated by `snappyHexMesh` and used in the simulation is shown in figure 3.

The converged simulation is post-processed using three different methods:

(i) Sampling the `age` field from a regular grid of points on the outlet surface (the method currently implemented in FARBASE).

(ii) Computing the integration time from a set of streamlines seeded randomly on the inlet (corresponding to the approach used in [2]).

(iii) Computing the integration time from a set of streamlines seeded randomly within a spherical domain overlapping the full tank radius.

To support fluid activation workflows, the distribution of interest is one which describes the residence times of fluid particles leaving a component per timestep. Given boundary probes and streamline seeds are both spatially distributed, each method also includes a conversion to a temporal distribution. For method 1, 10 000 boundary probes were used to arrive at a spatial distribution. To convert to a temporal distribution, 1000 samples were drawn with replacement, using outlet velocity as the relative weights. For methods 2 and 3, 1000 streamlines were seeded to arrive at a spatial distribution. To convert temporal distribution, a weighted histogram was plotted using the inverse of the integration time as the weights (such that each streamline contributes its weight to the bin count, rather than 1).

3.1. Results

The simulation terminated after 1588 iterations, with a pressure residual of 9.9×10^{-3} and a mean velocity residual of 1.74×10^{-4}. These residuals are significantly larger than the controls stated in table 2, but are considered sufficiently converged for the purposes of this preliminary study since the simulation

Table 3. Geometric dimensions and flow conditions for the expansion tank under study. R_1, L_1 are the radius and length respectively of both the inlet and outlet sections; R_2, L_2 are the radius and length respectively of the wider tank section; ν is the kinematic viscosity; and VFR is the volumetric flow rate.

R_1 (cm)	R_2 (cm)	L_1 (cm)	L_2 (cm)	ν ($m^2\ s^{-1}$)	VFR ($m^3\ s^{-1}$)
0.550	5.50	10.0	26.5	1.00×10^{-6}	1.70×10^{-4}

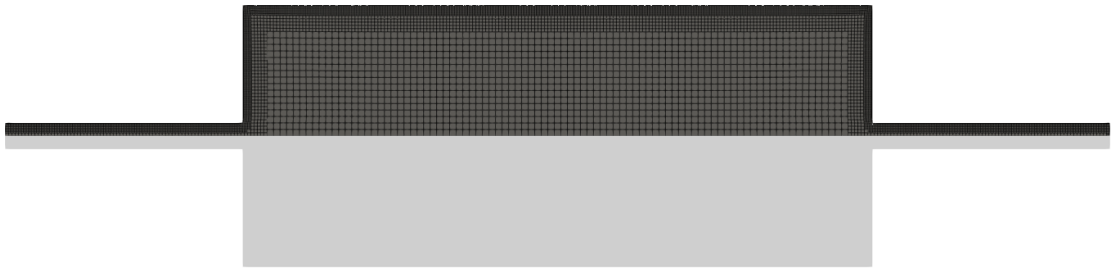


Figure 3. Mesh of expansion tank quadrant generated via FARBASE using the geometric parameters given in table 3. Two symmetry planes are used; for clarity, the full component surface is overlaid with 10% opacity.

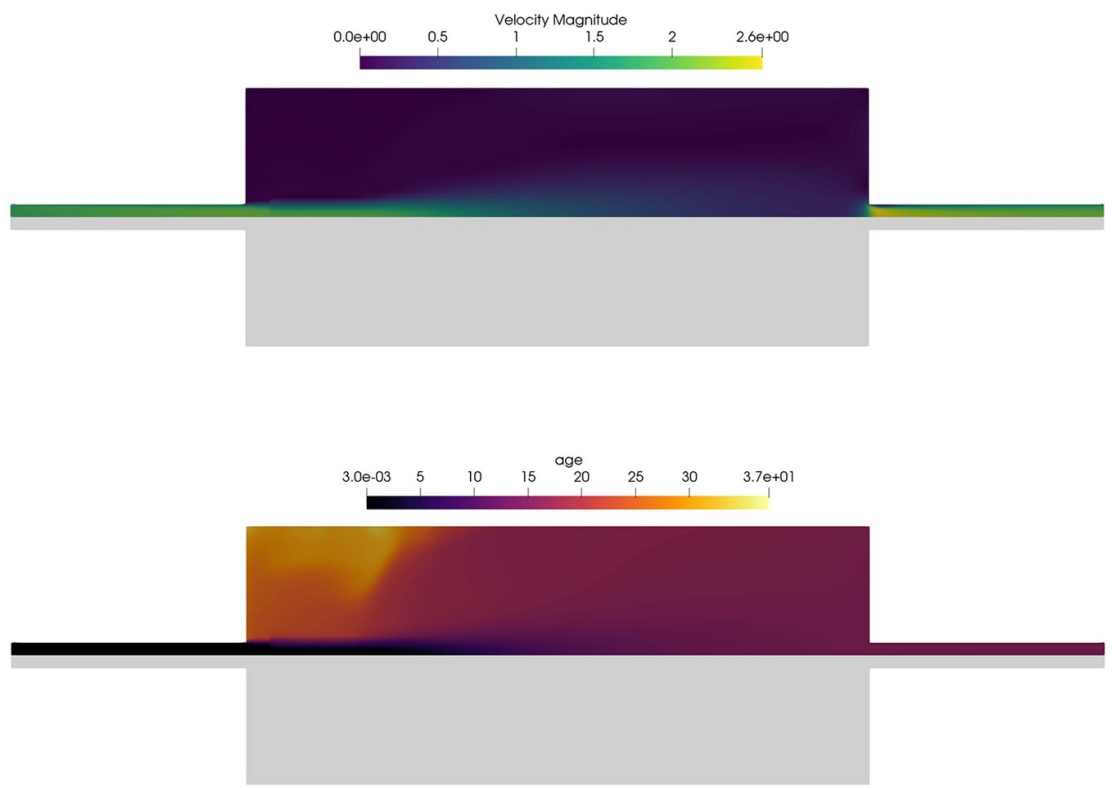


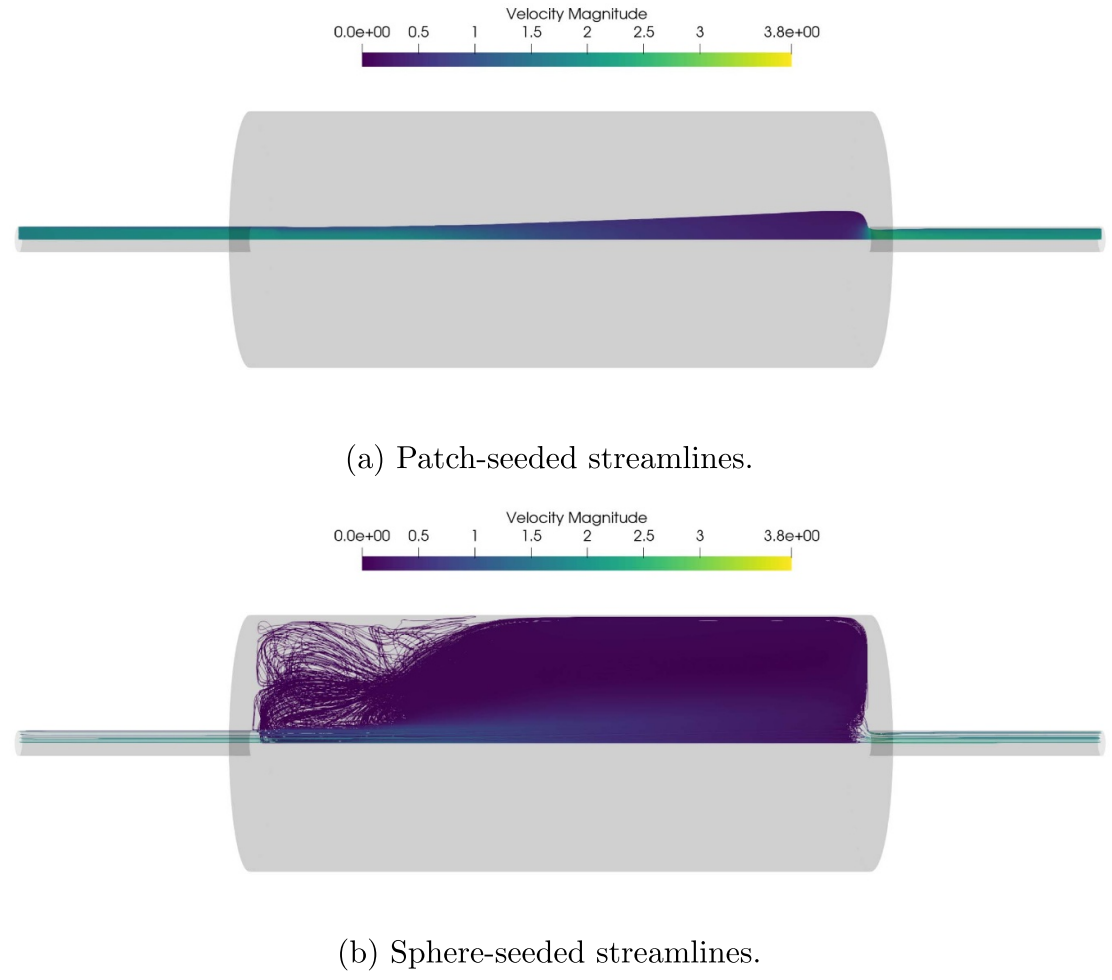
Figure 4. OpenFOAM results showing the steady-state velocity and age fields in the y–z plane for the simulated expansion tank quadrant. The flow direction is left to right.

result qualitatively produces the expected recirculation patterns for the expansion tank component. Thus, it constitutes an adequate test case for the comparison of RTD sampling methods.

Figure 4 shows the velocity and age fields for the steady-state solution while figure 5 shows the flow pattern and figure 6 shows the streamlines computed by each method. From these figures, the expected recirculation flow can be seen within the tank body, with the maximum of the age field positioned in the back of the tank (high radius, low axial distance). By contrast, the mean age at the outlet is considerably shorter.

Figure 7 and table 4 describe the RTDs sampled by each post-processing method:

Figure 5. Line integral convolution showing the steady-state flow pattern in the y–z plane for the simulated expansion tank quadrant. The flow direction is left to right.



(a) Patch-seeded streamlines.

(b) Sphere-seeded streamlines.

Figure 6. Streamlines computed via OpenFOAM for the steady-state expansion tank quadrant. The flow direction is left to right.

(i) For method 1, the distribution's range is relatively small and is dominated by a single-value. The recirculation region is not represented, since all samples were taken from the outlet and thus correspond only to the mean flow.

(ii) For method 2, the distribution resembles a log-normal distribution with a steep initial peak followed by a gradual tail. The recirculation region is not represented, since all streamlines exited the component without entering this region. Four spurious data were manually removed from the

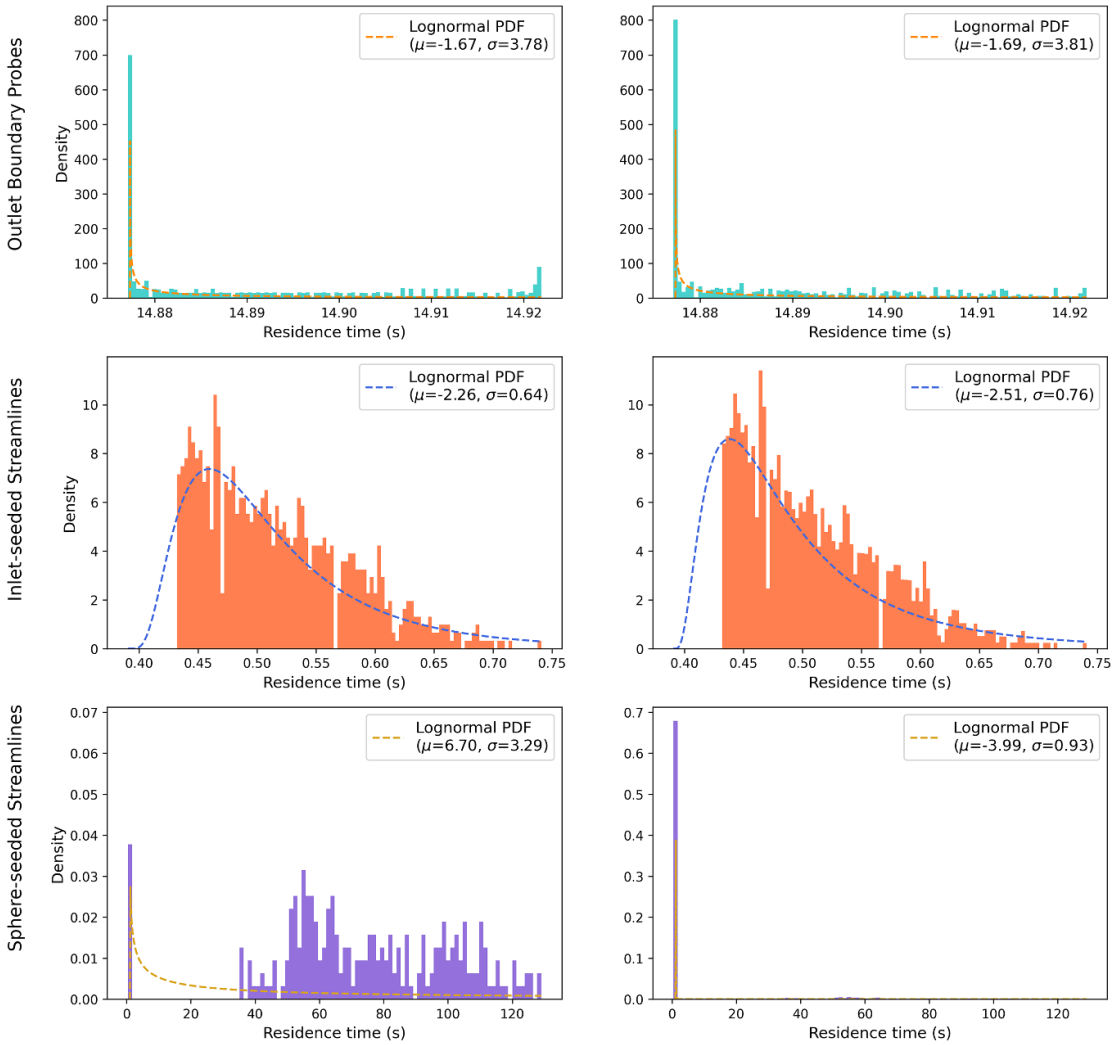


Figure 7. Histograms with fitted PDFs showing the RTDs for each post-processing method. The left side shows the data directly extracted from the simulation (boundary probes / streamlines) while the right side shows the processed data after applying weights to arrive at a temporal distribution.

Table 4. Average residence times for the unweighted (t_{avg}) and weighted ($t_{avg,w}$) RTDs shown in figure 7 alongside the ideal residence time $t_{ideal} = V/\text{VFR}$ where the expansion tank volume $V = 2(\pi R_1{}^2 L_1) + \pi R_2{}^2 L_2$.

Sampling method	t_{avg} (s)	$t_{avg,w}$ (s)	t_{ideal} (s)
Method 1 (Outlet boundary probes)	14.89	14.89	14.93
Method 2 (Inlet seeded streamlines)	0.5164	0.5094	14.93
Method 3 (Sphere seeded streamlines)	73.83	9.433	14.93

distribution. These streamlines terminated close to their source locations with very short integration times ranging 27–329 μs, likely due to numerical artefacts.

(iii) For method 3, the distribution contains a mixture of streamlines with similar times to method 2 all in the leftmost histogram bin, with a separate cluster of residence times corresponding to streamlines seeded in the recirculation regions. The latter did not exit the component and as a result were terminated upon reaching the maximum number of segments (19 999). Thus, their residence times may be considered arbitrarily high. In the weighted histogram, these high-valued contributions become barely visible.

3.2. Discussion

Comparing these methods: streamlines prove more effective at capturing the shape of an RTD compared to the boundary probes method presently implemented in FARBASE, which produces an overly narrow

distribution dominated by the mean flow conditions. Further, the residence times given by the boundary probes method do not directly correspond to either the particle tracks from the bulk flow ($t_{res} \sim 0.5\,s$) or the recirculating flow ($t_{res} > 25\,s$), so even if representative of mean flow, are unlikely to accurately represent the residence times experienced by any given fluid particle, which limits application to fluid activation contexts.

Both streamline methods provided a better range of values in the distribution. It is noted in the result for inlet-seeded streamlines that the fitted PDF uses a minimum residence time lower than the minimum in the dataset, as this gives the best fit to the data overall. Samples taken from this PDF could return residence times shorter than are actually realistic for the given component and flow conditions. One option to mitigate this would be to set the minimum residence-time value directly and only allow the log-normal shape and locations parameters (μ and σ respectively) to be adjusted when fitting the curve. Another option would be to implement alternative distribution functions which could be used to improve the fit. This could include the use of higher order functions, noting that these would also increase the dimensionality of the surrogate model's output space.

Comparing streamline methods, inlet-seeding does not capture recirculation current while sphere-seeding does so by assigning arbitrarily high residence times which prevent the fitting of a meaningful PDF. In both cases, spurious data may also arise from streamlines which terminate due to numerical artefacts, as occurred in the inlet-seeded streamlines. To address both of these issues, a streamline filtering procedure has been implemented which filters out any streamlines which do not connect from an inlet to an outlet, given some tolerance. Considering application more generally to components which may have multiple inlets or outlets, it is more appropriate to seed at a single outlet rather than an inlet, given that the distribution of interest is the RTD for flow exiting one pipe section and entering another. Given these points, the steady-state strategy chosen for future versions of FARBASE is a filtered outlet-seeding method which returns streamlines which connect to a given outlet from any inlet (with a tolerance of 10^{-6} m to account for negligible numerical deviations).

Recirculation regions pose a non-trivial issue to fluid activation modelling. In steady-state conditions, outside of irradiated zones, these regions may be largely irrelevant to the overall activation calculation. However, if the recirculation region exists in an irradiated zone, the fluid parcel contained there would likely become activated up to the point of saturation, until the flow conditions change such that the fluid escapes recirculation and is transferred to other parts of the system; this scenario is, by contrast, highly relevant to the overall activation calculation. Future work is needed to account for such scenarios.

4. Future work

Work is ongoing to integrate FARBASE into fluid activation modelling workflows. Specifically, FARBASE will provide RTDs to another UKAEA-developed code: GammaFlow, which will incorporate them into fluid activation calculations.

In order to validate FARBASE-assisted fluid activation calculations and compare them to the conventional approach, a benchmark case will be run using a known fluid activation system, such as the FNG [14] or KATANA [15] experiments. It is the desire of the authors to release an initial stable version of FARBASE following this validation work and with the addition of code sustainability features such as a complete test suite and continuous integration pipeline.

Further development of FARBASE will explore the following options:

- Use of streamlines to generate RTDs.
- Performance of mesh scaling studies to determine sensible default meshing parameters for nominal geometries.
- Addition of mesh quality checks to the automated pipeline, with the option to automatically refine the mesh until checks pass. This would enable the pipeline to produce suitable meshes for a wider region of the geometric parameter space.
- Implementation of transient simulation setup in the automated pipeline, to support the study of transient effects such as start-up and shut-down of fluid coolant systems.
- Use of dimensionless parameters in the surrogate model input space, potentially allowing for greater inference from training data at different spatial scales.

- Application to activated corrosion product modelling for more accurate transport of corroded solubles and crud in fluid circuits. This may necessitate development of FARBASE to consider diffusive processes and particulate transport, including consideration of the Schmidt number for the flow.

How best to account for stagnation and recirculation remains an important unanswered question in the field that will need to be addressed. However, if high-fidelity simulations are required, this only adds to the rationale to develop and maintain surrogate models for interpolating between these expensive simulations.

5. Conclusion

A new code, FARBASE, has been developed to flexibly and efficiently integrate CFD into fluid activation calculations. The code can provide estimated RTDs for a range of parametrically described fluid circuit components. Where data is insufficient to provide a reliable estimate, additional data can be generated by automatic deployment of an OpenFOAM CFD simulation.

A preliminary study was presented, comparing three approaches to deriving an RTD from the steady-state solution for an expansion tank component simulated via FARBASE. Of these, the inlet-seeded streamlines method is deemed the most appropriate method to use in future versions of the code, though it is noted that recirculation currents are not accounted for.

Work is ongoing to integrate FARBASE into GammaFlow fluid activation calculations and validate the results against known experimental benchmarks such as FNG and KATANA.

Acknowledgments

This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion) and from the EPSRC [Grant number EP/W006839/1]. To obtain further information on the data and models underlying this paper please contact PublicationsManager@ukaea.uk. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://doi.org/10.14468/qhvt-0h35 [16].

ORCID iDs

L R Humphrey 0000-0001-5066-6036
C L Grove 0000-0002-3630-5637
C R Shand 0000-0003-2259-8288

References

[1] Nuclear Energy Agency 2021 *Occupational Exposures at Nuclear Power Plants: Twenty-Eighth Annual Report of the Isoe Programme, 2018* (*Radiation Protection (Oecd)*) (available at: www.oecd-nea.org/jcms/pl_57736/occupational-exposures-at-nuclear-power-plants-2018)
[2] Berry T A, Nobs C R, Dubas A, Worrall R, Eade T, Naish J and Packer L W 2021 *Fusion Eng. Des.* **173** 112894
[3] Pampin R, Cau F, Fabbri M, Izquierdo J and Portone A 2021 *Nucl. Fusion* **61** 036003
[4] De Pietri M, Alguacil J, Rodríguez E and Juárez R 2023 *Comput. Phys. Commun.* **291** 108807
[5] Chiovaro P, Quartararo A, Avona P, Bongioví G, Di Maio P, Giambrone S, Moscato I and Vallone E 2024 *Nucl. Fusion* **64** 046016
[6] Binois M and Wycoff N 2022 *ACM Trans. Evol. Learn. Optim.* **2** 1–26
[7] Geuzaine C and Remacle J F 2009 *Int. J. Numer. Methods Eng.* **79** 1309–31
[8] Open Cascade SAS 2025 Open CASCADE technology (OCCT) (available at: https://dev.opencascade.org/)
[9] The OpenFOAM Foundation 2025 OpenFOAM® version 13: Open source CFD toolbox (available at: https://openfoam.org/release/13/)

[10] Wang M, Tang Y, Guo X and Ren X 2012 Performance analysis of the graph-partitioning algorithms used in OpenFOAM *2012 IEEE 5th Int. Conf. on Advanced Computational Intelligence (ICACI)* pp 99–104
[11] Wes M 2010 Data structures for statistical computing in Python *Proc. 9th Python in Science Conf.* ed S van der Walt and J Millman pp 56–61
[12] White F M 1999 *Fluid Mechanics* (McGraw-hill)
[13] Pedregosa F *et al* 2011 *J. Mach. Learn. Res.* **12** 2825–30
[14] Andreoli F *et al* 2020 *EPJ Web Conf.* **239** 21002
[15] Kotnik D, Peric J, Govekar D, Snoj L and Lengar I 2024 *Nucl. Eng. Technol.* **57** 103233
[16] Humphrey L R 2026 Data supporting the paper: Developing an automated fluid activation residence time CFD database *UKAEA Open Data* (https://doi.org/10.14468/qhvt-0h35)