Optimisation of a Hydrogen-Fuelled Parametric Strut Injector Using an Automated Workflow CFD Method

Nicholas C. W. Treleaven*
Safran Tech
Rue des Jeunes Bois, Châteaufort,
78114 Magny-Les-Hameaux, France
Email: nicholas.treleaven@safrangroup.com

Guillaume J. J. Fournier
Safran Tech
Rue des Jeunes Bois, Châteaufort,
78114 Magny-Les-Hameaux, France

Julien Leparoux
Safran Tech
Rue des Jeunes Bois, Châteaufort,
78114 Magny-Les-Hameaux, France

Renaud Mercier

Safran Tech

Rue des Jeunes Bois, Châteaufort,

78114 Magny-Les-Hameaux, France

*Address all correspondence to this author.

ABSTRACT

A strut injector for burning hydrogen has been optimised using an automated workflow system. The design process began with a parametric CAD model that allowed for various design features to be altered and various different designs to be generated from a given list of parameters. In order to choose the optimal set of injectors to cover the design space, an optimised design of experiments (DOE) method was used to automatically choose the parameters that best spanned the design space. One hundred candidate designs were chosen and a script used to generate a series of stereolithography (STL) files for each design. The STL files were then uploaded to a supercomputer for CFD analysis. For each of the 100 designs, a 4 step process was followed to generate the required data, this included a automated mesh generation step, field initialisation step, mesh adaptation step and finally an LES all within the YALES2 numerical framework. These 400 simulations were automated using an automatic workflow management process which limited the quantity of human intervention required and massively boosted productivity. In order to reduce the time required for post-processing and the amount of data required, the simulations relied heavily on a on-the-fly post-processing methodology which reduced the complex time-unsteady flow fields to a small number of quantified outputs of interest that measured the suitability of each design such as the pressure drop across the injector and the efficiency of the mixing process. At the conclusion of these simulations, automated scripts translated these outputs into a smaller set of parameters that could be used to compare each design and allow subsequent optimisation and surrogate modelling. Several surrogate modelling methods were attempted with mixed results however a simple classification methodology quickly identified the parameters of interest.

NOMENCLATURE

A_IN Injector inlet angle

A_OUT Injector outlet angle

a_χ SDR coefficient

a_ξ Mixedness coefficient

b_χ SDR exponent

b_ξ Mixedness exponent

E Averaging (Expectation) operator

k Kernel function

K Kernel function matrix

L_B Airgate height

L_G Airgate length

L_H Distance between injector and airgate

L_O Injector offset

L_OG Offset between injector trailing edge and airgate

L_X Injector thickness

L_Y Injector width

L_Z Injector length

N_I Number of hydrogen injection holes

$\mathcal{N}$ Gaussian Process operator

R_IN Injector inlet radius

T_{max} Maximum adiabatic flamelet temperature

x,y,z physical coordinates

$\mathbf{X}$ surrogate model input

$\mathbf{Y}$ surrogate model output

Z mixture fraction

Z_0 global mixture fraction

ΔP Pressure drop across injector

ϕ equivalence ratio

μ mean vector

ρ covariance parameter

σ standard deviation

σ_0 standard deviation of Gaussian noise signal

χ Scalar Dissipation Rate (SDR)

X 1D averaged Scalar Dissipation Rate (SDR)

ξ Mixedness

Ξ 1D averaged Mixedness

1 INTRODUCTION

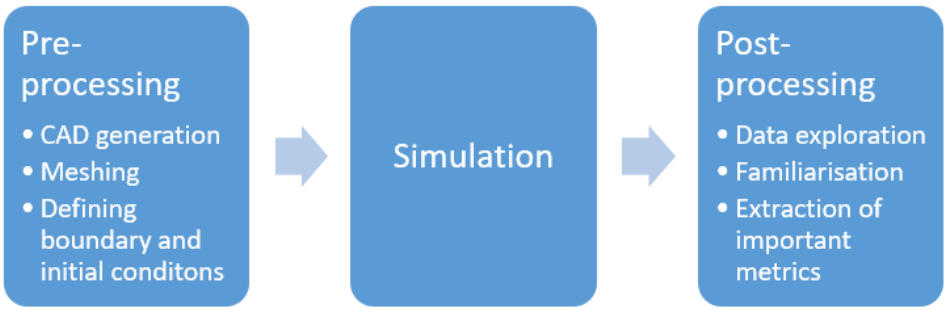
Increasing the efficiency of aerodynamic early-stage design is a subject of interest for rapidly exploring new design spaces that are becoming available due to the arrival of new manufacturing methods; and in the specific context of aeroengines, the onset of new fuels, such as hydrogen. Early stage design is difficult because of the large range of

possible designs and correspondingly large design space. This necessarily requires a large number of experiments or simulations to cover all of the possible designs, which either entails a long lead-time, a high cost, or both. Any methods, therefore, that can reduce the time or effort required to cover this early-stage design are extremely worthwhile. Among others, this includes: parametrisation of design, optimising the design of experiments (DOE), automated workflows for testing, reduced-order or low-dimensional modelling, and automated optimisation. Parametrisation of design helps in multiple ways :

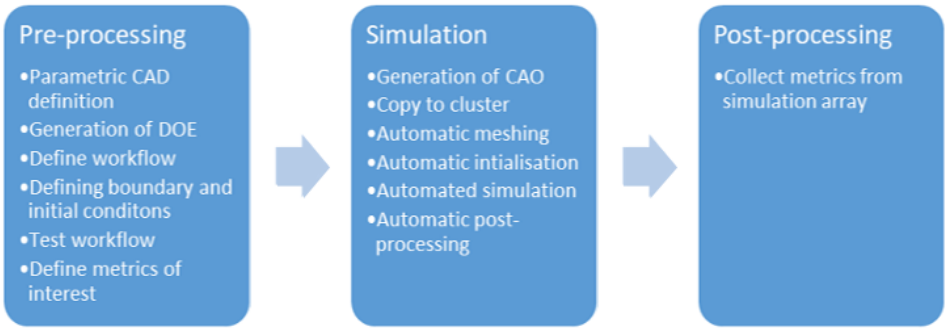
1. It ensure that designs are developed and evaluated in a systematic way that avoids accidental duplication.
2. It quantifies the design space through a number of adjustable numerical parameters.
3. It allows for optimisation as the gradient of any chosen objective function with respect to any design change is quantified through the gradient with respect to the design parameters.

Optimisation of the design of experiments (DOE) is essential as it allows the designer to cover the design space with the absolute minimum number of experiments or evaluations. This accelerates the design process and reduces costs. Even with an optimised DOE to reduce the number of test concepts, evaluating potential designs (through experimental testing or full-order numerical simulation) is expensive and time consuming. This difficulty explains why an automated testing procedure can be of benefit. In this case, the automated workflow procedure means that after the parameters of each test design are chosen, there is very little human intervention. This allows the designer to work on multiple projects in parallel, significantly improving productivity. One important aspect of such a workflow, specific to CFD simulations is the use of *live*, *real-time*, *on-the-fly*, or *streaming* post-processing methods. In a traditional CFD workflow, the beginning point is the pre-treatment stage, including tasks such as defining the computational mesh. The second point is the simulation itself and the final stage is the post-processing of the simulation data. The cost of the first stage can be minimised by using an automated meshing strategy where the geometry inputs (file types, boundary conditions, etc.) are standardised such that after a single mesh is generated manually, the meshing procedure can be generalised and applied automatically to a number of different concepts. The second stage of the process, the simulation itself, can be optimised through either the efficiency of the code (physical time per CPU hours spent) or through improving the parallel efficiency of the code (number of processors that can be used at the same time). The final stage, post-processing, can require a huge number of man hours, especially in the case where the designer has limited experience with the flow and needs to visually check the form of the flow before the relevant statistics or metrics are extracted. Although this remains the case for the present study, once the initial familiarisation of the flow was completed, metrics were selected to be computed *on-the fly*. This capability was aided through the post-processing capabilities of the YALES2 framework, where planes of data can be extracted at a predetermined time step and quantities of interest calculated before being refined down to small number of low-dimensional output values

that can be periodically saved to disk. Three recent studies that make use of various elements of the proposed design strategy are a multifidelety optimisation of a hydrogen injector [1], an experimental optimisation of a pilot injector for an industrial gas turbine [2], and the optimised rescaling of a liquid fueled aero-engine injector [3]. A more traditional combustor design process would rely not only on CFD calculations but also on low order models, built and optimised using data collected during previous design studies [4]. The generation of designs, meshes and running of calculations may be (partially) automated, howeve the evaluation of designs, based on CFD results, is done manually and relies heavily on the expertise of the combustion chamber designer [5, 6]. The use of artificial intelligence in the design process has tended to centre on either developing low order models of engine performance [7] or on improving the performance of reacting flow solvers [8]. The first aspect limits the number of operating points that must be tested to ensure wide operability limits, while the second helps to accelerate the resolution of the reacting flow fields of a previously untested design.



(a) The traditional CFD workflow.



(b) The automated workflow.

Fig. 1: The classic CFD workflow and the new automated workflow used in this study.

The design methodology described in this study was used to explore and optimise the geometry of a hydrogen

fuel injector for aero-engine applications. The technology used for this injector is a form of microinjection that has previously only been used for supersonic combustion refered to as a *strut injector*. The chosen application for this technology is to replace a singular conventional swirled fuel spray nozzle with an array of these strut injectors for the main combustion chamber of a gas turbine. The principle of all micro-injection concepts is to limit the production of nitrous-oxides through miniaturising the size of the flames and hence the recirculation zones and residence times of the hot gases produced during combustion. In a similar type of injector, *micromix*, the air is injected into the combustion chamber through a large number of small holes in an *air-guiding panel*. Each of these holes produces a jet of air that is then fed with a small amount of hydrogen, injected perpendicularly into each air jet. This jet-in-crossflow configuration leads to a series of vortices and small recirculation zones close to the injector face. The problem with such an injection system is that due to the high flame stability of hydrogen, the flame can easily become attached to the injector head that may subsequently leads to unacceptable levels of heat stress and premature wear.

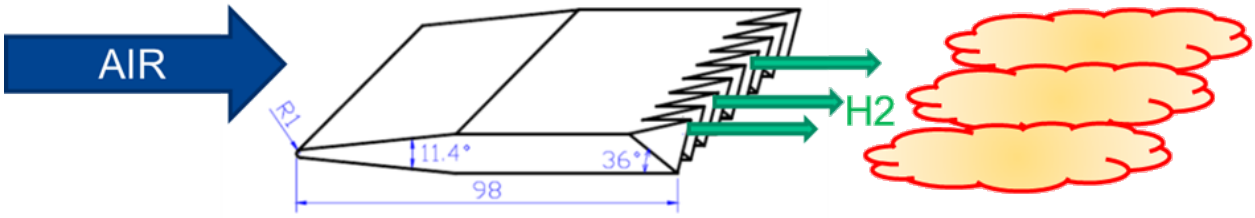


Fig. 2: The strut injector concept (Adapted from [9]).

Strut injectors attempt to solve this problem by instead injecting the hydrogen in the same direction as the surrounding airflow (Fig. 16a). The air-guiding panel is also simplified into a series of channels with a single strut injector consisting of a hydrogen fuel line which act as an obstacle in the flow. Although the rate of mixing between the hydrogen and air is reduced as compared to a jet-in-crossflow configuration, the likelihood of the flame becoming attached to the injector is reduced due to either one, or all of the following:

- The mixture being too rich in (or close to) the hydrogen jet for combustion to occur.
- The mixture being too lean in the regions between the hydrogen jets to sustain combustion.
- The local velocity of the flow being higher that the turbulent flame speed close to the injector.
- The scalar dissipation rate or strain being too high to sustain combustion close to the injector.

In order to enhance the effects of points 3 and 4 above, strut injectors rely on a series of *teeth* on the trailing edge of the injector to simultaneously aid mixing and to locally quench the flame close to the injector.

This paper begins with an explanation of the computational, design, and surrogate modelling methodologies

where the generation of the concepts, design of experiments, LES and low-dimensional models are discussed. This is followed by a presentation of the results of this study and a brief attempt at optimisation using a simple classification method. Finally, a discussion is included on how the approach could be improved.

2 METHODOLOGY

2.1 Definition of the parametric geometry

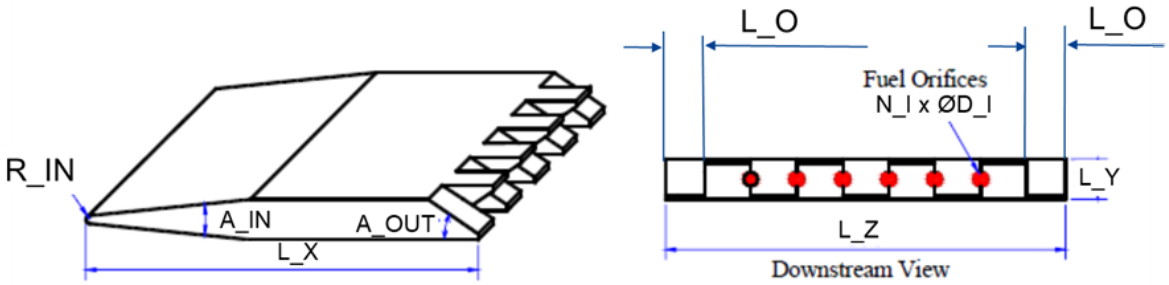


Fig. 3: The parametric strut injector (adapted from [9]).

The strut injector design chosen for this study is based on one of the concepts presented by Sunami et al. [9] for a supersonic combustion application. It consists of a wedge shaped strut (Fig. 3) with an inlet radius (R_IN), and wedge angle (A_IN). The wedge is limited at an overall strut thickness (L_Y). On the trailing edge a number (N_I) of hydrogen injections holes are added between a certain number (N_I+1) of teeth of triangular cross-section to help mix the hydrogen with the surrounding air. The triangular teeth are given an exit angle of A_OUT and alternate in orientation, with every second tooth facing up and those in between facing down. The strut also has a length (L_X) and a width (L_Z) with an offset (L_O) which independently defines the outermost teeth which do not have hydrogen injection holes.

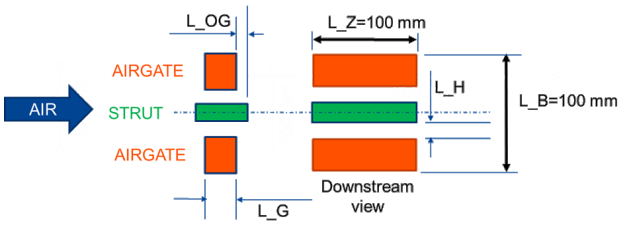
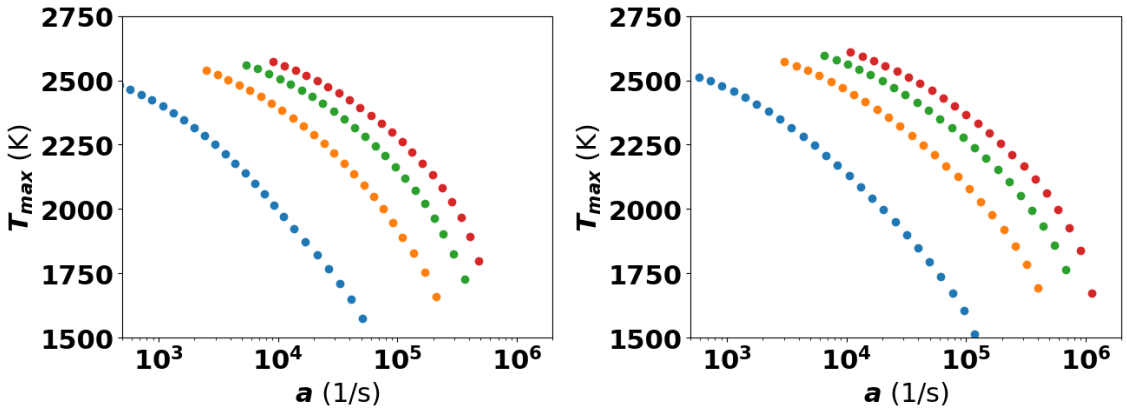


Fig. 4: Definition of the airgate.

In order to increase the velocity of the air close to the injector a simple airgate was also added to the geometry. The air gate is defined by its height (L_B) and its width (L_Z) which is set to be the same as the injector. The size of the opening for the air to pass through the airgate is then defined by the air gate height (L_H), the length of the airgate (L_G) and an additional offset (L_OG) is added to allow for configurations where the air gate is in front of, or behind the strut. In order for this strut to be appropriate for later testing in an experimental facility, the width of the injector was fixed at L_Z= 100mm and the offset set to L_O = 32.5 mm. Excluding these parameters gives a design space with 10 degrees of freedom. The limits of each of these parameters are shown in Table 2. A parametric CAD model was built of this injector in *CATIA* allowing rapid generation of the CAD model over a wide range of design parameters. A *visual basic* script was also written to read an *excel* file containing the parameters chosen for a number of injectors and sequentially generate the CAD model for each and save them as STL (stereolithography) files that could be read by the CFD and meshing software *YALES2*.

2.2 Definition of the output metrics



(a) 300 K.

(b) 700 K.

Fig. 5: The flame temperature for a non-premixed flame with an air temperature of 300 K and 700 K given as a function of the flame strain rate. Blue symbols: 1 bar, orange: 5 bar, green: 9 bar, red: 13 bar.

The ideal injector geometry would result in low overall NOx emissions, a stable flame, low heat transfer to the injector and maintain its performance over a wide range of operating conditions. In order to achieve this goal, the ideal injector should produce a lifted flame [10]. As already mentioned a key part of this capability is linked to a high level of scalar dissipation rate close to the injector. The scalar dissipation rate is closely linked to the flame strain rate [11], whose effect on the hydrogen flame can be simulated with a 1D counterflow flame. Figure 5 shows the maximum temperature of a 1D hydrogen air counterflow diffusion flame as a function of strain rate, air flow

temperature and pressure, computed using the *cantera* package [12] with the SanDiego mechanism [13, 14]. The simulations begin at the lowest strain rate (the most leftward point for each pressure and temperature) and the strain rate is then progressively increased until the flame extinguishes. The strain rate that leads to flame quenching for each case is the point furthest to right in each set of data points. The quenching strain rate (or scalar dissipation rate) shows a strong relationship with temperature and pressure with both higher pressures and temperatures associated with higher flame quenching strain rates. This shows that as the combustion chamber pressure and temperature increases, the harder it is to maintain a lifted flame.

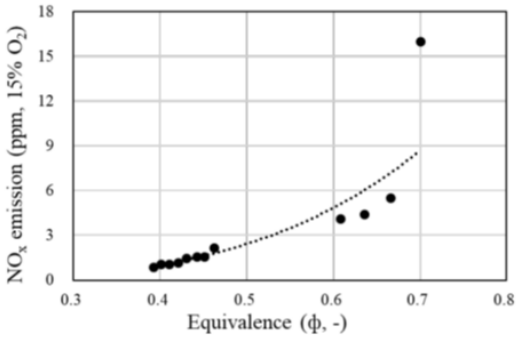


Fig. 6: NOx emissions for a multi-tube combustor as a function of equivalence ratio (from Cho et al. [15]).

Figure 6 shows the effect of equivalence ratio on the production of NOx for another type of microinjection [15]. The figure indicates that the closer the equivalence ratio is to one, the more NOx is produced. It is also important to note that the increase is not linear and an equivalence ratio of 0.7 results in a NOx production of more than two times the rate of an equivalence ratio of 0.66. This non-linear behaviour highlights an important property of a good injector design: the hydrogen should be quickly mixed with air before combustion. Because the hydrogen is injected without premixing, the equivalence ratio is very large close to the injector, however due to the overall lean equivalence ratio of the system, the mixture at the exit of the chamber is also lean. The goal is therefore to avoid the flame burning when the equivalence ratio is close to one. In order to reduce the likelihood of this occurring, the flow should be homogenised as quickly as possible.

Whilst running a set of reacting simulations would give more information, the first two criteria for a good design, *a lifted flame* with *low NOx*, can be developed using cold simulations with a passive scalar to represent the injection and mixing of hydrogen. Unfortunately, the third and fourth criteria (*a stable flame* and a *broad operating range*), cannot be tested with a cold simulation. However, at the very early stages of the design process, *thermoacoustic stability* cannot easily be assessed as the geometry of the combustion chamber has not yet been defined. The *operating range*

of the combustor is also difficult to test because it requires testing the design over a broad range of conditions either by simulation or experiment. At this stage of the design process, it is too expensive due to the large number of candidate designs. The metrics used to define the cost function for the optimal design are hence chosen to be related to the scalar dissipation rate and the mixedness of the flow downstream of the injector. Cold, incompressible LES simulations with a passive scalar to represent hydrogen are chosen as a lower cost alternative to reacting LES.

2.2.1 The scalar dissipation rate (SDR)

The Scalar Dissipation Rate (SDR) is defined as [11]:

$$\chi = 2\mathcal{D}\left|\frac{\partial Z}{\partial x}\right|^2 \tag{1}$$

where Z is the passive scalar mixture fraction and $\mathcal{D}$ is the diffusivity according to Fick's law. One option would be to generate time average fields of the scalar dissipation rate and compare different designs manually. The problem is that with a large number of designs, the comparison becomes very time consuming. A better option is to calculate the average scalar dissipation rate over a series of planes with their normal in the x direction, at different distances downstream from the training edge of the injector. Ten planes were defined in the interval x=[0.0,0.1]. On each plane, 100x100 equidistant points were sampled at each time step and used to compute the surface average of each metric of interest (Fig. 7).

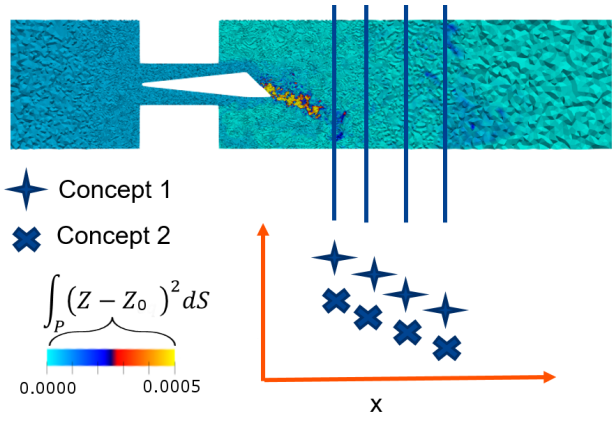


Fig. 7: A diagram showing how the downstream planes were used to integrate the quantities of interest at each downstream location.

2.2.2 *Mixedness*

A scalar is also defined to measure the overall efficiency of mixing:

$$\xi = (Z - Z_0)^2 \tag{2}$$

where $Z_0 = 0.007$ is the overall global mixture fraction. The plane and temporal averaged values of χ and ξ, X and Ξ respectively can be plotted as a function of distance downstream from the injector trailing edge. Because these values are still a function of space, they present a difficulty to any future optimisation or surrogate modelling task. In order to simplify the task of any future surrogate modelling, the number and dimensionality of the target outputs should be reduced. The dissipation of a passive scalar in a fully developed turbulent planar jet or wake evolves according to [16]:

$$\overline{u}\frac{\partial \overline{Z}}{\partial x} + \overline{v}\frac{\partial \overline{Z}}{\partial y} = \Gamma\frac{\partial^2 \overline{Z}}{\partial y^2} - \frac{\partial \overline{v'Z'}}{\partial y} \tag{3}$$

where the overbar represent a ensemble average and the dash represents the instantaneous (mean subtracted) quantity, this is very similar as the equation for the mean axial velocity $\overline{u}$. The mixture fraction of the passive scalar is therefore expected to vary as [16]:

$$\overline{Z}(x) \propto x^{-1/2} \tag{4}$$

It is therefore assumed that the evolution of Ξ and X are of the form:

$$\Xi = a_\xi x^{-b_\xi} \tag{5}$$

and

$$X = a_\chi x^{-b_\chi} \tag{6}$$

Results from the simulations are therefore used to find the values of best fit of a_ξ, a_χ, b_ξ, and b_χ for each concept. Because the curve fit assumes that the wake is fully developed, it is not surprising that the inclusion of the value at the trailing edge injector plane leads to lower accuracy fit. The values of X and Ξ at the trailing edge of the injectors are therefore included as separate optimisation parameters ($X(x = 0)$ and $\Xi(x = 0)$). Table 1 shows the list of output values that were approximated by the surrogate models along with an indication of whether the value should be maximised or minimised during the optimisation.

Output	Parameter	**Symbol**	**Optimum**
1	Pressure loss (%)	ΔP	< 5
2	Global maximum SDR	max χ	maximised
3	SDR at trailing edge	$X(x = 0)$	maximised
4	SDR coefficient 1	a_χ	maximised
5	SDR coefficient 2	b_χ	maximised
6	mixedness at trailing edge	$\Xi(x = 0)$	maximised
7	mixedness coefficient 1	a_ξ	maximised
8	mixedness coefficient 2	b_ξ	maximised

Table 1: List of outputs collected from the simulations.

2.2.3 Global maximum scalar dissipation rate and pressure drop

An additional parameter of interest is the global maximum scalar dissipation rate, which may not necessarily lie in one of the extracted planes. It is calculated by finding the maximum value of SDR in the 3D domain. Finally, the overall pressure drop across the injector and air gate is also calculated, which is an important design parameter for the combustion chamber. The pressure drop is calculated using an area average of the static pressure over the inlet and outlet patches of the computational domain.

2.3 Definition of the workflow

The workflow process is broken into 3 main parts: preparation, automation and result collection. Preparation remains the most labour intensive process as it requires the operator to setup and test the automated parts of the workflow to ensure that the workflow can continue without supervision. The automation step is designed to be as autonomous as possible to limit the interaction required from the user and the result collection stage is designed such that it is either entirely automated or, requires only very small quanties of input data so that the results can be analysed

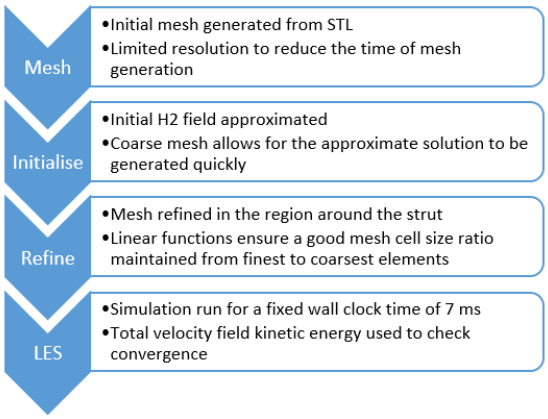


Fig. 8: The four simulations run for each new concept automatically during the workflow.

efficienty in real-time.

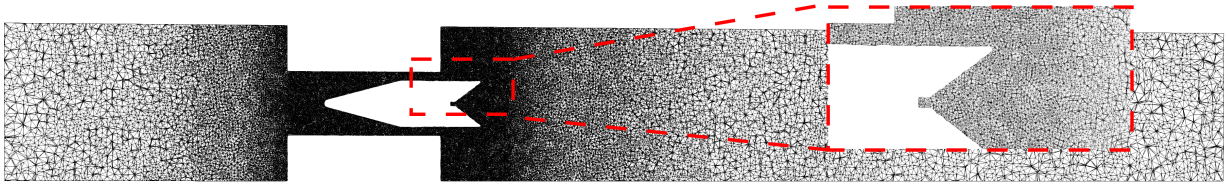
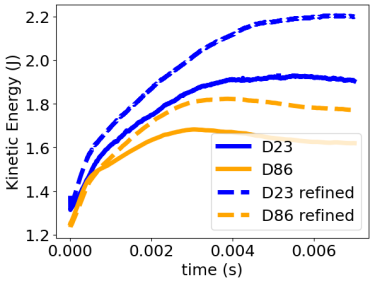
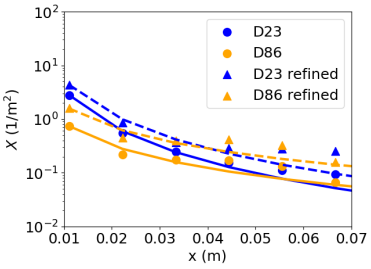


Fig. 9: The (unrefined) LES mesh for an optimal (D86) design.

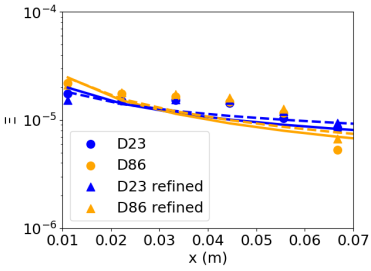
The preparation step involved the setup of four *test* stages of the simulation (Fig. 8): Mesh, Initialise, Refine and LES. The mesh step runs a simulation to generate an initial coarse mesh of the domain using only the input STL files (also automatically generated from the parametric CAD model). The full domain included an STL file that corresponded to the body of the injector, an STL file that represented the airgate and an STL file that represented the injection holes for the hydrogen at the trailing edge of the injectors. These STL files were placed into a rectangular domain automatically generated by the CFD solver *YALES2* with dimensions of $800 \times 100 \times 100$ mm. The mesh was then generated using the methods described in [17, 18]. The initial mesh was heavily refined at the hydrogen injector holes in order to capture the finest surface geometries (0.2 mm for the injection holes, 0.3 mm for the struts and 0.6 mm for the airgate). Once the mesh generation step is completed, the initialisation step begins, which involves an initial guess of the mixture fraction field ($Z = 0$ upstream of the injector, $Z = Z_0$ downstream) and a subsequent poorly refined LES simulation run for 24 hours until the plume of high mixture fraction is sufficiently downstream. The third step was then to refine the mesh to LES quality using the methods described in [19] using the MMG library [20, 21]. The LES mesh had a maximum mesh cell density of 1 mm close to the injector (with smaller surface



(a) Integral Kinetic Energy as a function of time.



(b) Average Scalar Dissipation Rate as a function of axial position. Symbols = values extracted from simulations at predefined planes. Lines = lines of best fit used in the optimisation study.



(c) Average Mixedness as a function of axial position. Symbols = values extracted from simulations at predefined planes. Lines = lines of best fit used in the optimisation study.

Fig. 10: A comparison of three important statistics between an optimal (D86) and sub-optimal (D23) design. Solid line = original unrefined mesh density used in the optimisation study. Dashed line = refined mesh.

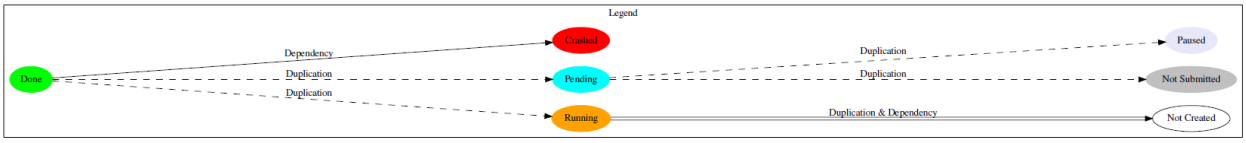


Fig. 11: The output of *workflow.py –plot*.

densities maintained from the initial mesh) and linearly expanded towards the inlet and outlet with a maximum density of 7 mm. This resulted in a final mesh cell count of around 10 million cells. Figure 9 shows the mesh used for an optimal case (D86), this mesh is typical of the meshes used throughout the study. The final stage is the LES which is run for a total of 7 ms, during which, the statistics are collected for further analysis.

The mesh resolutions chosen were tested by refining the mesh by a factor of two for an optimal (D86) and a sub-optimal (D23) design. The linear expansions of the mesh density before and after the injectors was maintained, however the minimum and maximum cells sizes were reduced to 0.5 and 3.5 mm, leading to meshes of around 80 million cells. The temproral evolution of the global integrated kinetic energy and spatial evolution of the SDR and mixedness are shown in Fig. 10. The higher resolution of the meshes results in higher levels of kinetic energy due to a greater propotion of turbulent scales being resolved in the refined simulations. However, as can be seen by the spatial evolution of the mixedness and scalar dissipation rate, the change in mesh resolution does not significantly alter the statistics used for the optimisation. Some variation is seen for the case of the scalar dissipation rate however this only affects the absolute values of scalar dissipation rate and does not alter either the slope of the lines of best fit (b_ξ, and b_χ) or the ranking of the concepts in terms of $X(x = 0)$.

Once the robustness of these four simulation steps was validated, the workflow was initiated with all of the 100

geometries to be analysed. *YALES2* workflows are launched in three automated stages. The first stage copied the files from the *test* directories into the working directory and automatically changed the input files to point to the different geometries to be tested. An additional list of strut lengths was also read so that the dimensions of the refine mesh zone around each injector corresponded to the each injector length. A second automated stage entered each of the now 400 simulation folders (100 geometries $\times$ 4 simulations) and compiled the *YALES2* code ready for submission. After the 400 executables were compiled, a final automated step launched the submission process. The first 100 simulations were submitted to the scheduler. At the completion of each simulation a script was called to link the most recent simulation's output to the next simulation step. This process was monitored automatically by using the *workflow.py –plot* command which generated an output (Figure 11) showing which simulations had completed, which were still pending and which had crashed and needed further input from the user. The workflow ran on a supercomputing cluster for around 4 days in physical time, during which 1.5 million CPU hours were consumed.

2.4 Design of Experiments (DOE)

Before a surrogate model can be constructed or a design can be optimised, data must be collected from either simulation or experiment. If the design process is begun without previous experience or data available, a number of experiments and/or simulations must be planned. Simulations and experiments are expensive and as such the number should be minimised. At the same time, the response surface obtained for different output parameters of interest can be highly irregular and non-linear with respect to the input design parameters and as such, in order to prevent the design falling into a local, but not global minimum, the span covered by the set of simulations or experiment should be maximised. The worst choice for efficiency but optimal for covering the design space is to conduct an experiment for multiple levels of each design parameters whilst keeping all the others constant. In the case of finding the sensitivity with repsect to ten design parameters at two different levels, a complete set of results would require 1024 simulations.

Latin Hypercube Sampling (LHS) improves the efficiency of the DOE by considering each level of each parameter only once. In the given example, this would reduce the number of required simulations from 1024 to 20. This assumes however that the non-linear interactions between multiple inputs and each output are minimal. The choice of the values of each input design parameter for a LHS DOE is not fixed though and the LHS can be optimised to improve the space filling of the method. The *MaxProj LHS* (Maximum Projection Latin Hypercube) [22] optimises the LHS such that it fills a maximum of space assuming no non-linear interaction but also optimises for non-linear interactions of higher and higher order with progressively smaller and smaller weightings. Unfortunately in this design one of the design parameters (N_I, the number of injection holes) is a categorical input (it must be an integer value). For this type of input, the DOE must be adapted to the Sliced Maximin-Distance LHS method [23]. This is achieved using

the SLHD (Sliced Latin Hypercube Design) package [*https://cran.r-project.org/web/packages/SLHD/index.html*]. In a pure (non-sliced) LHS DOE, a good rule of thumb is that there should be at least 10 designs tested for each input parameter. However, the slicing of the domain using different categories usually entails an increase in the size of the DOE. Due to the cost of the simulations in this study, the 10 designs per input where maintained leading to a total DOE size of 100 designs across 4 different numbers of injection holes. The design parameters and their ranges tested are listed in Table 2. Figure 12 shows some of the designs tested.

Input	Parameter	**Min**	**Max**	**Categories**
1	N_I			[2,3,4,5]
2	D_I (mm)	1	5	
3	L_X (mm)	20	100	
4	L_Y (mm)	5	30	
5	R_IN (mm)	1	5	
6	A_IN (degrees)	10	45	
7	A_OUT (degrees)	10	80	
8	L_H (mm)	5	10	
9	L_OG (mm)	-10	50	
10	L_G (mm)	10	100	

Table 2: Range of design parameters

2.5 LES Solver

The meshing, simulations and initial post-processing were done using the *YALES2* software package [24]. Due to the computational cost and uncertainty associated with a reacting simulation of turbulent hydrogen flames, a hot combustion simulation strategy was avoided and replaced by a surrogate cold simulation. The LES simulations were solved using the variable density solver of YALES2 using the WALE subgrid scale model [25]. It solves the low-Mach Navier-Stokes equations [26] using the finite volume method on unstructured meshes using a projection method [27]. The time advancement used a 4th order TFV4A method with a 4th order accurate central spatial integration for the equations of density and momentum [28]. A single transported scalar was solved to represent the mixture fraction (Z) using MUSCL scheme with the second-order accurate flux limiter SUPERBEE [29] and 3rd order accurate temporal integration. The local properties of the fluid were calculated as a linear function of the local mixture fraction based on the properties of air ($Z = 0$) and hydrogen ($Z = 1$). The laminar Schmidt number was set to 0.204 which is

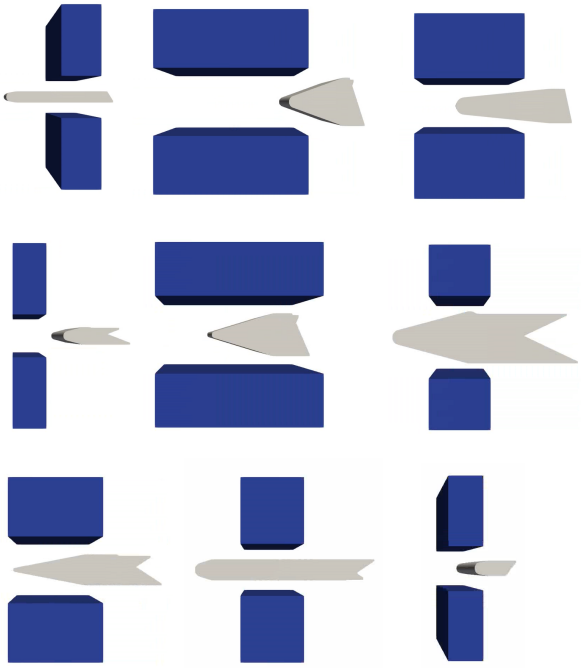


Fig. 12: Side views of nine of the one hundred designs tested. Blue = injector/strut, red = hydrogen injection holes, grey = airgate.

approximately equal to the Schmidt number of hydrogen in air at $Z = 0.007$ at 400 K, which is equal to the global hydrogen mixture fraction (Z_0). Hydrogen was injected through the hydrogen injection holes at 300 K and air was injected at the domain inlet boundary at 450 K with a mass flow rate of 103 g/s.

2.6 Surrogate modelling

2.6.1 LASSO model

The LASSO (least absolute shrinkage and selection operator) model [30] was developed as an alternative to the least squares method that helps to boost the accuracy and interpretability of the derived model. It seeks a model of the form:

$$f^* = \sum_{i=0}^{N} \beta_i \mathbf{X}_i \tag{7}$$

where $\mathbf{X}_i$ are the input vector components, β_i is the set of coefficients and f^* is the optimal estimator for the input data function $f(\mathbf{X}_i)$. The model is chosen such that:

$$\sum_{i=0}^{N} |\beta_i| \leq t \tag{8}$$

where t is a tuning parameter. The aim of this restriction is to drive some of the model coefficients to zero such that the dimensionality of the resulting model is reduced and that only the strongest interactions between the model inputs and output are retained. The optimisation of this model is performed using the *glmnet* package [31, 32]. The LASSO models were built within the in-house Safran data exploration and optimisation platform NEXT STEP. An open-source version called *Lagun* [https://gitlab.com/drti/lagun] is freely available.

2.6.2 ACOSSO

ACOSSO (Adaptive COmponent Selection Shrinkage Operator) [33] is similar to the LASSO model except that non-linear interactions between the inputs $(\mathbf{X_i}, ..., \mathbf{X_p})$ are considered (in general up to any order but in this study limited to order 2):

$$f(\mathbf{X}_i, ..., X_p) = f_0 + \sum_{i=1}^{p} f_i(\mathbf{X}_i) + \sum_{i<j}^{p} f_{ij}(\mathbf{X}_i, \mathbf{X}_j) \tag{9}$$

The functions f are chosen to be elements of a reproducing kernel Hilbert space, and similarly to the LASSO model, the model coefficents are forced towards zero by penalising the sum of the norms of the coefficients. This allows for non-linear effects to be included in a LASSO type model. The ACOSSO model was built within NEXT STEP.

2.6.3 Kriging model

A kriging model or Gaussian Process regression (GPR) is a type of machine learning that builds a model from a set of data and simultaneously gives an estimate of the certainty of the model prediction. Battacharya et al. [34] recently used such a model, extended to take into account inputs of variable fidelity, to predict the emissions of a gas turbine running on hydrogen. GPR assumes that the output of a function f with a vector of inputs $\mathbf{X}$ is:

$$\mathbf{y} = f(\mathbf{X}) + \epsilon \tag{10}$$

where ϵ is a noise term assumed to follow a normal distribution. If it is assumed that the input vector signal is a random variable following its own distribution, the function can be approximated as a Gaussian process:

$$f(\mathbf{X}) \sim \mathcal{N}(\mu(\mathbf{X}), K) \tag{11}$$

where $\mu(\mathbf{X}) = E[f(\mathbf{X})]$ is the mean vector, $K_{ij} = k(\mathbf{X_i}, \mathbf{X_j})$ is the matrix of kernel functions that defines the statistical correlation between two set of inputs and $\mathcal{N}$ is the multivariate normal distribution function. The kernel functions are defined as:

$$k(\mathbf{X_i}, \mathbf{X_j}) = E[(f(\mathbf{X_i}) - \mu(\mathbf{X_i}))(f(\mathbf{X_j} - \mu(\mathbf{X_j}))] \tag{12}$$

The defined Gaussian process will then be used to estimate a set of unknown outputs ($\mathbf{f}^*$) from an additional set of untested inputs ($\mathbf{X}^*$), the combined system is written:

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(\mathbf{X}) \\ \mu(\mathbf{X}^*) \end{bmatrix}, \begin{bmatrix} K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I & K(\mathbf{X}, \mathbf{X}^*) \\ K(\mathbf{X}^*, \mathbf{X}) & K(\mathbf{X}^*, \mathbf{X}^*) \end{bmatrix} \right) \tag{13}$$

where there is expected to be a Gaussian noise signal with variance σ_n^2 in the input data and I is the identity matrix. Assuming that the model data has been normalised such as $\mu(\mathbf{X}) = \mu(\mathbf{X}^*) = 0$, the predicted output $f^*(\mathbf{X}^*)$ is given by:

$$\begin{aligned} f^* \sim \mathcal{N}(K^*(\mathbf{X}^*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I]^{-1} f, K(\mathbf{X}^*, \mathbf{X}^*) \\ -K(\mathbf{X}^*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I]^{-1} K(\mathbf{X}, \mathbf{X}^*)) \end{aligned} \tag{14}$$

In this study the kernel function is assumed to be the Matérn 5/2 function:

$$k_\nu(\tau) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\frac{\tau}{\rho}\right)^\nu K_\nu \left(\sqrt{2\nu}\frac{\tau}{\rho}\right) \tag{15}$$

where Γ is the gamma function, σ is the standard deviation, K_ν is the modified Bessel function of the second kind, ρ is a non-negative covariance parameter and $\nu = 5/2$. The surrogate model is trained within NEXT STEP using the Robust GaSP package [35].

2.6.4 O-POD

The Output based Proper Orthogonal Decomposition (O-POD) [36] uses the Snapshot POD method [37] to build a set of modes that statistically link the input and output data, and approximate the outputs based on a given set of inputs. The model inputs and outputs are used to generate an augmented matrix:

$$\boldsymbol{\varphi} = [\mathbf{X}, \mathbf{y}] \tag{16}$$

where $\mathbf{X}$ is the set of input data and $\mathbf{y}$ is the set of output data that will be used to train the model. The columns of $\mathbf{X}$ and $\mathbf{y}$ consist of the different input and output data available and the rows consists of the values of this data for different tests or simulations. The POD is then applied to this augmented matrix:

$$\boldsymbol{\varphi} = \sum_{i=0}^{N_r} a_i \boldsymbol{\theta}_i \tag{17}$$

with

$$\boldsymbol{\theta}_i = [\theta_i, \psi_i] \tag{18}$$

a_i is a column vector of temporal coefficients that correspond to the different tests or simulations but that are independent of the measurement (input or output) while θ and ψ are modes that statistically link the behaviour of the inputs

and outputs across all of the data set. These modes are constructed such that the original data is optimally represented by the smallest number of modes (N_r) and therefore the POD has the advantage of acting as a noise filter as non-statistically significant white noise is pushed into the higher order modes that can be excluded from the model with a minimal effect on the model accuracy. Once these modes are constructed they can be used to interpolate the output solution $\mathbf{y}^*$ from an unknown input vector $\mathbf{X}^*$.

The interpolation is done by finding the values of a_i that closest represent the new, untested model inputs $\mathbf{X}^*$. In POD, this is done by projection of the model inputs and outputs ($\boldsymbol{\varphi}$) onto the POD modes ($\boldsymbol{\theta}$) however, in the case of O-POD, only the values of $\mathbf{X}^*$ are known and not $\mathbf{y}^*$. The interpolation is therefore done using the least square method by finding a_i^* such that:

$$\min \sum_{j=1}^{N_{in}} \left(\sum_{i=1}^{N_r} a_i^* \theta_{i,j} - {\mathbf{X}^*}_j \right)^2 \tag{19}$$

which is easily solved using the *numpy.linalg* package [38] and N_{in} are the number of model inputs. Finally the unknown model outputs can be found using:

$$\boldsymbol{y}^* = \sum_{i=0}^{N_r} a_i^* \psi_i \tag{20}$$

3 RESULTS AND DISCUSSION

3.1 Automated workflow

The automated workflow progressed as expected although there were some difficulties encountered with the meshing of the geometries. The mesh solver of YALES2 is sensitive to the mesh density of the chosen geometry STL selected by the user and is also sensitive to the relative target size of the volume mesh cells and the nearby surfaces. Because the geometry changes between each design, in some cases the values chosen by the user were inappropriate for the chosen geometry resulting in a crash of the meshing program. The workflow would therefore be aided by both a more robust algorithm and the automatic switching of the target mesh cell sizes without user intervention. There remained ten simulations that could not be meshed by the solver and this is a focus for improving the robustness of the chosen approach in the future. Nevertheless, the failure rate (10 out of 100) is similar to the experience of colleagues working in the turbomachinery space who perform geometry optimisation using commercial tools.

After the geometries were meshed, the workflow progressed as expected. The freshly meshed geometries were

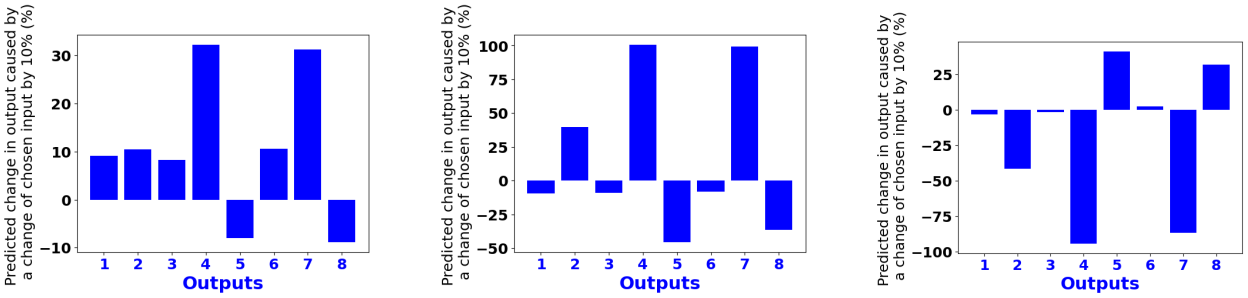
initialised and run with the coarse mesh over 24 hours to reduce the time of LES computation. Then the domain was refined to the LES standard and the simulation run for 7 ms while the statistics were collected. The calculation of the output scalars, and their integration on the planes of interest was done by YALES2 and the results printed to a text file at each time step. A small script was run at the end of the simulation to assemble all of these outputs across all of the simulations into one file for final post-processing.

Before the surrogate models could be built, the data was time averaged. The convergence of all of the simulations was checked by plotting the volume integral of the kinetic energy and the time-averaging was done over the last 5000 samples which corresponded with a global time to convergence (5 - 7 ms, Figure 10a). Lines of best fit were made using the *scipy.optimize* [39] python package and the resulting coefficients and time averaged values were saved into a text file for futher analysis.

3.2 Surrogate modelling

Type	ΔP	max χ	$X(x=0)$	a_χ	b_χ	$\Xi(x=0)$	a_ξ	b_ξ
LASSO	0.007	0.075	0.500	-0.013	0.410	0.393	0.089	0.526
ACOSSO	0.012	-0.024	0.61	0.001	0.571	0.640	0.109	0.638
Kriging	0.010	0.086	0.575	0.008	0.531	0.541	0.207	0.563

Table 3: The output of NEXT STEP showing the Q^2 accuracy of the LASSO, ACOSSO and Kriging models. The poorest quality models are highlighted in red.



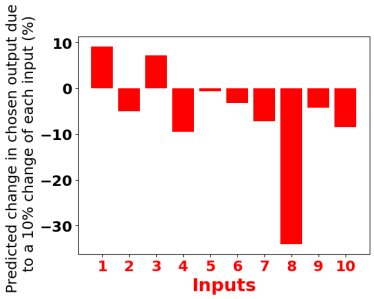
(a) Number of hydrogen injection holes (N_I).

(b) Height of the strut (L_Y).

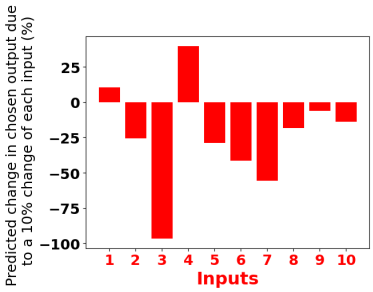
(c) Angle at the leading edge of the strut (A_IN).

Fig. 13: The change in outputs predicted by the O-POD model with a 10% increase in the values of 3 different inputs.

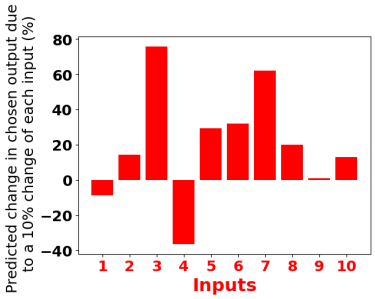
The resulting simulations outputs were imported into NEXT STEP along with the corresponding design param-



(a) The pressure drop accross the strut.



(b) The global maximum of SDR.



(c) The rate of mixing.

Fig. 14: The change in select outputs predicted by the O-POD model with a 10% increase in the values of all inputs.

eters as inputs. These combined results were then used to build surrogate models for each output using LASSO, ACOSSO and Kriging type models. The error of each surrogate model was evaluated using a leave one out analysis where a single sample of input and output data is removed from the set and the surrogate model computed a second time. This is then repeated for every data point. If the surrogate model is robust and accurate it should accurately reproduce the value of the left out data point. The error is therefore calculated between each model and the left out data using the Q^2 criterion:

$$Q^2 = 1 - \frac{\frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} (Y_i^{\text{true}} - Y_i^{\text{pred}})^2}{\frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} (Y_i^{\text{true}} - \overline{Y}_i^{\text{true}})^2} \tag{21}$$

where Y is the data output being tested and the subscripts *pred* and *true* represent the predicted values from the model and the true values respectively. The results of the surrogate modelling in NEXT STEP is shown in Tab. 3. The accuracy of all of the tested surrogate models falls below the limit of a "good" model ($Q^2 > 0.85$). The surrogate modes do especially badly at predicting the pressure drop, maximum SDR, and the a_X and a_Ξ coefficients. The other outputs are more easily predicted but still fall short. Globally, the ACOSSO models do the best out of the models available which can be explained by the strong non-linear physics that are expected to be represented by the models. The LASSO models are clearly the worst which further enforces this hypothesis. The limitation of the ACOSSO models to only second order interactions limits the number of inputs whose effects must be combined together to predict the output response observed. One option at this point would be to continue running simulations at the points where the models are least precise in order to reduce the uncertainty and boost the accuracy of the surrogate models. Although this remains a possibility, this action was put on hold until it is sure that other types of models can not produce better results than those already available within NEXT STEP.

The advantage of a POD based model over the previous tested models is that the POD can capture the non-linear interaction between different inputs (even though each mode individually can only capture the linear relationship between two inputs) and that the POD has the natural property of pushing the error and noise into the higher order modes. This means that a POD based model can tune out some of the noise to focus only on the correlations that matter at the cost of reducing the fidelity of the model. A detailed error analysis of this model has not yet been undertaken however some of the results are presented in Figs. 13 & 14. The first set of figures tests the sensitivity of the design to changes in the input parameters. Figure 13 shows the effect of increasing each input value independently by 10% on the outputs starting from the mean values (R_IN = 2.9 mm , A_IN = 29° , L_Y = 18 mm, N_I = 4, A_OUT = 48°, L_X = 74 mm, L_H = 7.5 mm, L_OG = 21 mm, L_G = 57 mm). Three inputs are chosen: the number of hydrogen injection holes, the height of the strut, and the angle of the leading edge. Increasing the number of injection holes (and therefore teeth) is associated with a large increase in the values of a_X and a_Ξ as well as a modest increase in the other outputs but a decrease in the values of b_X and b_Ξ. This shows that increasing the geometric complexity of the injector increases the local SDR at the injector exit but with the effect of decreasing the rate of mixing. This is consistent with the observations made in the theoretical study of Marant and Cossu [40] who used linear stability analysis to show that the optimum mixing is achieved using vortices and streaks (flow structures) of the largest size. The height of the strut can be seen to decrease slightly the rate of mixing but also to increase the maximum value of the SDR in the flow field. An increasing inlet angle is seen to increase the rate of mixing and also reduce the global maximum of the SDR. An increased inlet angle is likely to promote a region of separation at the point where the inlet section of the strut meets the top and bottom surfaces and this appears to improve mixing further downstream.

An alternative approach is shown in Fig. 14 where the effects on each output are plotted as a function of which input is modified. The larger the bar, the larger the effect on the chosen output. The pressure drop is shown first and, as expected, the largest effect on the pressure drop is the size of the air gate between the injector and the surrounding structure. The global maximum SDR is shown next and indicates that the largest increase in global SDR is achieved by having a shorter (decreasing L_X (Input 3)) or fatter strut (increasing L_Y (Input 4). Finally, the rate of mixing b_Ξ is shown and indicates that the largest influence is driven also the length of the injector, but with the reverse effect: an increase of L_X (Input 3) or reduction in the value of L_Y (Input 4) leads to an increase in the rate of mixing. A shorter or fatter injector leads to a larger maximum SDR most likely due to large separations at the trailing edge. A longer injector might promote the generation of turbulent eddies in the boundary layer, leading to better mixing and small separations at the trailing edge. This relationship shows that there is globally a trade-off to be made between the rate of mixing and having the highest local SDR close to the injector.

3.3 Optimisation

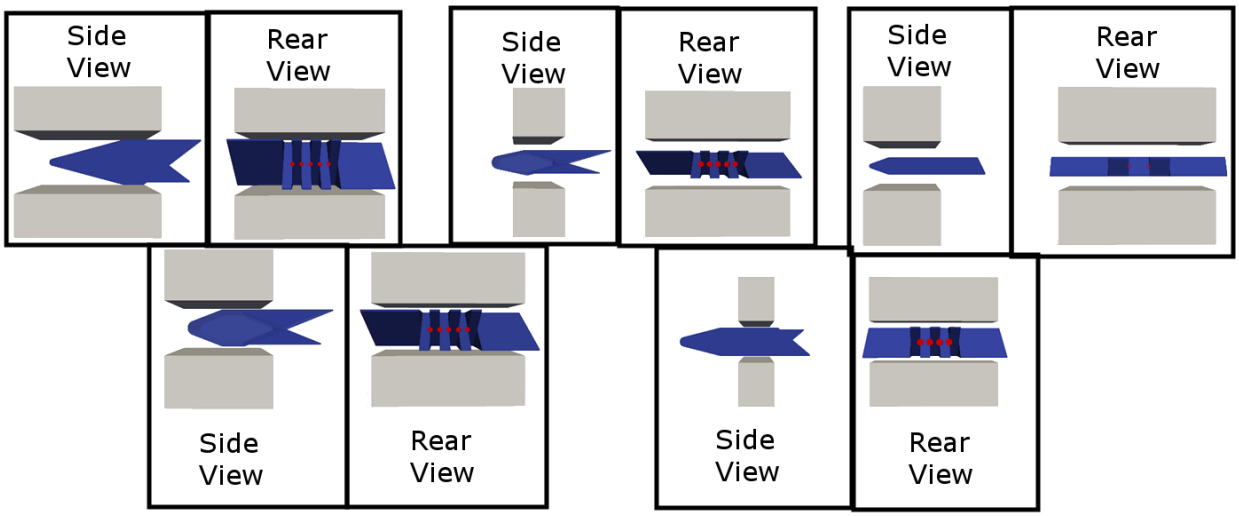


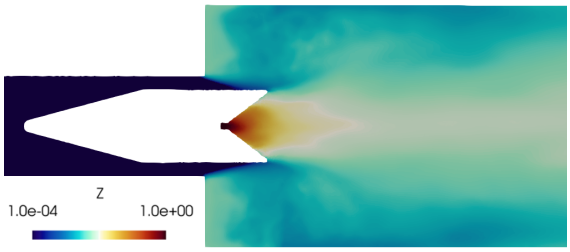
Fig. 15: The five designs chosen as being the most optimal. Blue = injector/strut, red = hydrogen injection holes, grey = airgate.

After defining the surrogate models the injectors can be optimised. There are however a number of practical difficulties in doing so:

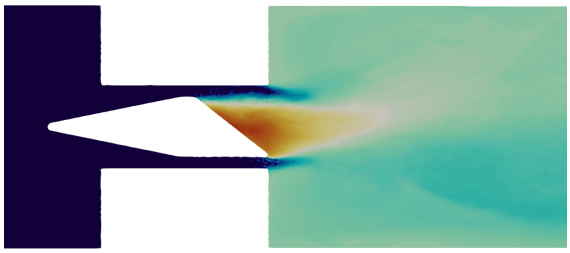
- There is no surrogate model of good enough quality generated within NEXT STEP and that can be used within the NEXT STEP optimisation routines.
- The relationship between the inputs and outputs appears to be highly non-linear and difficult to interpret, therefore optimisation without the right tools runs the risk of falling into local minima.
- The optimisation is clearly multi-objective in nature and without reacting experiments, simulations or other tools, it is difficult to assign weights to the criteria.

Due to these restrictions a simple classification routine is used as a demonstration and to get an idea of the kinds of struts that appear to have desirable properties. Refering to Tab. 1, the desirable characteristics of an injector are chosen to be:

- The largest possible value of SDR at the trailing edge to prevent the flame from attaching to the injector.
- The largest possible mixing rate.
- The smallest possible pressure drop.
- The largest possible value of global SDR.



(a) D86: An optimal design.



(b) D23: A sub-optimal design.

Fig. 16: The mixture fraction field for an optimal and sub-optimal design. The color scale is logorithmic in order to highlight the differences between the designs.

These criteria were applied one by one to the 90 concepts and the bottom 50% of the injectors were eliminated at each stage. After applying the 4 criteria, 5 injectors remained and these are shown in Fig. 15. The results demonstrate the power of the human eyeball: although the surrogate models struggle to identify the design features leading to the optimal design, it is reasonable obvious that these designs have common features:

- The injectors are long.
- The displacement between the end of the air gate and the end of the injector is always positive (L_OG<0).
- The injectors can either be thick with small inlet/outlet angles and large numbers of hydrogen injector holes or thinner with larger inlet/outlet angles and a smaller number of hydrogen injector holes

Figure 16 shows the hydrogen mass fraction contours along the centreline of the combustion chamber for an optimal (D86) and sub-optimal injector (D23). The color scale is logarithmic to better highlight the difference in the two designs. It can be noted that in the case of the sub-optimal design, the high hydrogen containing wake behind the strut is larger than in the optimal design. The most noticable dfference in the two designs in this plane is the relative distance from the end of the air canal to the end of the injector. The protrusion of the optimal design into the combustion chamber leads to a (double) backwards facing step, while the sub-optimal design is closer to a wake configuration. The presence of the backwards facing steps leads to a large adverse pressure gradient at the dump-plane which encourages the growth of turbulent fluctuations in the boundary layer close to the injector wall. At the rearwards

tip of the injetor, these fluctuations lead to a more rapid mixing of the wake region. For the sub-optimal design, the adverse pressure gradient is only triggered at the trailing edge of the injector and therefore it takes more time for the turbulent fluctuations to grow and mix out the high-hydrogen containing wake.

Future work will aim to link the results from this study to results from reacting simulations, experiments and correlations in order to define the weightings that should be used in a future multi-object optimisation routine. A good place to start is the work of Leroy et al. [41] who developed a relationship between the NOx produced and the design parameters of a jet flame and the work of Marragou et al. [10] who developed a theory to discern whether or not a hydrogen flame will attach itself to an injector based on the triple flame speed and the local velocity field.

4 CONCLUSION

An automated design workflow has been demonstrated for an early stage design of hydrogen fuelled strut injector. The strut design was parametrised using ten adjustable parameters and an optimal LHS DOE developed to test the sensitivity of the design to the input parameters. Eight design outputs were identified and an automated workflow developed using YALES2 to generate the outputs in real time to minimise the amount of user intervention. After the workflow was completed, several surrogate models were tested in the hope that they could be used to optimise a future design. The surrogate models identified some interesting trends in the sensitivity of the outputs to the input parameters but were not sufficiently accurate to enable optimisation without additional simulation or experimental input. Future work will aim to link these cold flow simulation results to the real-life reacting case. This will include the use of simple correlations and also the integration of a multi-fidelity design approach where surrogate models are built to accommodate different input data of varying fidelity (LES, RANS simulations, experiments, correlations) in order to optimise a design with a minimal level of human intervention and with a minimum of high-fidelity, expensive data.

ACKNOWLEDGEMENTS

This work was granted access to the HPC resources of TGCC under the allocation A0142B13460 made by GENCI. This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101056865 (HESTIA). The research work presented was also funded through the National H2Tech program granted by the French Délégation Générale de l'Aviation Civile (DGAC). The authors would also like to thank Sebastien Trocherie from Safran Engineering Services who prepared the CAD models and the ongoing support of thier colleagues at Safran Tech and the CORIA laboratory.

REFERENCES

[1] Farjon, P., Bertier, N., Dubreuil, S., and Morio, J., 2023. "Towards the multifidelity optimization of h2-air injector for aircraft propulsion". In Turbo Expo: Power for Land, Sea, and Air, Vol. 86953, American Society of Mechanical Engineers, p. V03AT04A017.

[2] Reumschüssel, J. M., von Saldern, J. G., Ćosić, B., and Paschereit, C. O., 2024. "Multi-objective experimental combustor development using surrogate model-based optimization". *Journal of Engineering for Gas Turbines and Power,* ***146***(3).

[3] Gövert, S., Gruhlke, P., Behrendt, T., and Janus, B., 2023. "Scaling of lean aeronautical gas turbine combustors". In Turbo Expo: Power for Land, Sea, and Air, Vol. 86953, American Society of Mechanical Engineers, p. V03AT04A042.

[4] Pegemanyfar, N., Pfitzner, M., Eggels, R., von der Bank, R., and Zedda, M., 2006. "Development of an automated preliminary combustion chamber design tool". In Turbo Expo: Power for Land, Sea, and Air, Vol. 42363, pp. 327–336.

[5] Tangirala, V. E., Tolpadi, A. K., Danis, A. M., and Mongia, H., 2000. "Parametric modeling approach to gas turbine combustor design". In Turbo Expo: Power for Land, Sea, and Air, Vol. 78552, American Society of Mechanical Engineers, p. V002T02A048.

[6] Dudebout, R., Reynolds, B., and Molla-Hosseini, K., 2004. "Integrated process for cfd modeling and optimization of gas turbine combustors". In Turbo Expo: Power for Land, Sea, and Air, Vol. 41669, pp. 679–686.

[7] Asgari, H., Chen, X., Menhaj, M. B., and Sainudiin, R., 2013. "Artificial neural network–based system identification for a single-shaft gas turbine". *Journal of Engineering for Gas Turbines and Power,* ***135***(9), p. 092601.

[8] Chi, C., Janiga, G., and Thévenin, D., 2021. "On-the-fly artificial neural network for chemical kinetics in direct numerical simulations of premixed combustion". *Combustion and Flame,* ***226***, pp. 467–477.

[9] Sunami, T., Magre, P., Bresson, A., Grisch, F., ORAIN, M., and Kodera, M., 2005. "Experimental Study of Strut Injectors in a Supersonic Combustor Using OH-PLIF". In AIAA/CIRA 13th International Space Planes and Hypersonics Systems and Technologies Conference, American Institute of Aeronautics and Astronautics.

[10] Marragou, S., Magnes, H., Aniello, A., Guiberti, T. F., Selle, L., Poinsot, T., and Schuller, T., 2023. "Modeling of H2/air flame stabilization regime above coaxial dual swirl injectors". *Combustion and Flame,* ***255***, Sept., p. 112908.

[11] Poinsot, T., and Veynante, D., 2022. *Theoretical and Numerical Combustion*. AFNIL, May.

[12] Goodwin, D. G., Moffat, H. K., Schoegl, I., Speth, R. L., and Weber, B. W., 2023. Cantera: An Object-oriented Software Toolkit for Chemical Kinetics, Thermodynamics, and Transport Processes, Aug.

[13] Curran, H. J., 2019. "Developing detailed chemical kinetic mechanisms for fuel combustion". *Proceedings of the Combustion Institute,* ***37***(1), pp. 57–81.

[14] Gimenez-Lopez, J., Rasmussen, C. T., Hashemi, H., Alzueta, M. U., Gao, Y., Marshall, P., Goldsmith, C. F., and Glarborg, P., 2016. "Experimental and Kinetic Modeling Study of C2H2 Oxidation at High Pressure". *International Journal of Chemical Kinetics,* ***48***(11), pp. 724–738.

[15] Cho, E.-S., Jeong, H., Hwang, J., and Kim, M., 2022. "A Novel 100% Hydrogen Gas Turbine Combustor Development for Industrial Use". In Proceedings of ASME Turbo Expo 2022: Turbomachinery Technical Conference and Exposition, Vol. Volume 3A: Combustion, Fuels, and Emissions, American Society of Mechanical Engineers, p. V03AT04A020.

[16] Pope, S. B., 2000. *Turbulent Flows*. Cambridge: Cambridge University Press, Aug.

[17] Meynet, S., Moureau, V., Lartigue, G., and Hadjadj, A., 2021. "Automatic surface and volume mesh generation for roughness-resolved LES of additive-manufacturing heat exchangers". In 13th International ERCOFTAC symposium on engineering, turbulence, modelling and measurements (ETMM13).

[18] Grenouilloux, A., Leparoux, J., Moureau, V., Balarac, G., Berthelon, T., Mercier, R., Bernard, M., Bénard, P., Lartigue, G., and Métais, O., 2023. "Toward the use of LES for industrial complex geometries. Part I: automatic mesh definition". *Journal of Turbulence,* ***24***(6-7), pp. 280–310.

[19] Leparoux, J., Mercier, R., Moureau, V., and Musaefendic, H., 2018. "Primary atomization simulation applied to a jet in crossflow aeronautical injector with dynamic mesh adaptation".

[20] Dapogny, C., Dobrzynski, C., and Frey, P., 2014. "Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems". *Journal of Computational Physics,* ***262***, Apr., pp. 358–378.

[21] Dobrzynski, C., and Frey, P., 2008. "Anisotropic Delaunay Mesh Adaptation for Unsteady Simulations". In Proceedings of the 17th International Meshing Roundtable, Springer, pp. 177–194.

[22] Joseph, V. R., Gul, E., and Ba, S., 2015. "Maximum projection designs for computer experiments". *Biometrika,* ***102***(2), Mar., pp. 371–380.

[23] Ba, S., Myers, W. R., and Brenneman, W. A., 2015. "Optimal Sliced Latin Hypercube Designs". *Technometrics,* ***57***(4), pp. 479–487.

[24] Moureau, V., Domingo, P., and Vervisch, L., 2011. "Design of a massively parallel CFD code for complex geometries". *Comptes Rendus Mécanique,* ***339***(2), pp. 141–148. High Performance Computing.

[25] Nicoud, F., and Ducros, F., 1999. "Subgrid-Scale Stress Modelling Based on the Square of the Velocity Gradient Tensor". *Flow, Turbulence and Combustion,* ***62***, Sept., pp. 183–200.

[26] Pierce, C. D., and Moin, P., 2004. "Progress-variable approach for large-eddy simulation of non-premixed turbulent combustion". *Journal of Fluid Mechanics,* ***504***, Apr., pp. 73–97.

[27] Chorin, A. J., 1968. "Numerical solution of the Navier-Stokes equations". *Mathematics of Computation,* ***22***(104), Oct., pp. 745–762.

[28] Kraushaar, M., 2011. "Application of the compressible and low-mach number approaches to large-eddy simulation of turbulent flows in aero-engines". Ph.D. Thesis, Institut National Polytechnique de Toulouse, Dec.

[29] Roe, P. L., 1985. "Some contributions to the modelling of discontinuous flows". In Large-Scale Computations in Fluid Mechanics, pp. 163–193.

[30] Tibshirani, R., 1996. "Regression Shrinkage and Selection via the Lasso". *Journal of the Royal Statistical Society. Series B (Methodological),* ***58***(1), pp. 267–288.

[31] Friedman, J. H., Hastie, T., and Tibshirani, R., 2010. "Regularization Paths for Generalized Linear Models via Coordinate Descent". *Journal of Statistical Software,* ***33***(1), Feb., pp. 1–22.

[32] Simon, N., Friedman, J. H., Hastie, T., and Tibshirani, R., 2011. "Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent". *Journal of Statistical Software,* ***39***(5), Mar., pp. 1–13.

[33] Storlie, C. B., Bondell, H. D., Reich, B. J., and Zhang, H. H., 2011. "Surface Estimation, Variable Selection, and the Nonparametric Oracle Property". *Statistica Sinica,* ***21***(2), Apr., pp. 679–705.

[34] Bhattacharya, C., Christopher, J., Thierry, D., Biruduganti, M., Supekar, S., and Dasgupta, D., 2023. "Data-Driven Surrogate Modeling of Microturbine Combustors Burning Hydrogen Blends". In Proceedings of ASME Turbo Expo 2023: Turbomachinery Technical Conference and Exposition, Vol. Volume 9: Microturbines, Turbochargers, and Small Turbomachines; Oil and Gas Applications, American Society of Mechanical Engineers, p. V009T18A011.

[35] Gu, M., Palomo, J., and Berger, J. O., 2019. "RobustGaSP: Robust Gaussian Stochastic Process Emulation in R". *The R Journal,* ***11***(1), June, pp. 112–136.

[36] Christ, P., 2019. "Modeling of Automotive HVAC Units Using Proper Orthogonal Decomposition". Ph.D. Thesis, Technical University of Munich.

[37] Sirovich, L., 1987. "Turbulence and the dynamics of coherent structures. I. Coherent structures". *Quarterly of Applied Mathematics,* ***45***(3), Oct., pp. 561–571.

[38] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E., 2020. "Array Programming with NumPy". *Nature,* ***585***(7825), Sept., pp. 357–362.

[39] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., Vijaykumar, A., Bardelli, A. P., Rothberg, A., Hilboll, A., Kloeckner, A., Scopatz, A., Lee, A., Rokem, A., Woods, C. N., Fulton, C., Masson, C., Häggström, C., Fitzgerald, C., Nicholson, D. A., Hagen, D. R., Pasechnik, D. V., Olivetti, E., Martin, E., Wieser, E., Silva, F., Lenders, F., Wilhelm, F., Young, G., Price, G. A., Ingold, G.-L., Allen, G. E., Lee, G. R., Audren, H., Probst, I., Dietrich, J. P., Silterra, J., Webber, J. T., Slavič, J., Nothman, J., Buchner, J., Kulick, J., Schönberger, J. L., de Miranda Cardoso, J. V., Reimer, J., Harrington, J., Rodríguez, J. L. C., Nunez-Iglesias, J., Kuczynski, J., Tritz, K., Thoma, M., Newville, M., Kümmerer, M., Bolingbroke, M., Tartre, M., Pak, M., Smith, N. J., Nowaczyk, N., Shebanov, N., Pavlyk, O., Brodtkorb, P. A., Lee, P., McGibbon, R. T., Feldbauer, R., Lewis, S., Tygier, S., Sievert, S., Vigna, S., Peterson, S., More, S., Pudlik, T., Oshima, T., Pingel, T. J., Robitaille, T. P., Spura, T., Jones, T. R., Cera, T., Leslie, T., Zito, T., Krauss, T., Upadhyay, U., Halchenko, Y. O., Vázquez-Baeza, Y., and SciPy 1.0 Contributors, 2020. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". *Nature Methods,* ***17***(3), Mar., pp. 261–272.

[40] Marant, M., and Cossu, C., 2018. "Influence of optimally amplified streamwise streaks on the Kelvin–Helmholtz instability". *Journal of Fluid Mechanics,* ***838***, Mar., pp. 478–500.

[41] Leroy, M., Mirat, C., Renaud, A., Puggelli, S., Zurbach, S., and Vicquelin, R., 2024. "Structure and NOx Emissions of Stratified Hydrogen-Air Flames Stabilized on a Coaxial Injector". *Journal of Engineering for Gas Turbines and Power,* ***146***(3), Mar., p. 031012.