# MULTI-OBJECTIVE SURROGATE-BASED OPTIMIZATION METHOD FOR THE SCALING OF COMBUSTION CHAMBERS

**B. Lambert**
German Aerospace Center (DLR)
Institute of Combustion Technology
Pfaffenwaldring 38–40
D-70569 Stuttgart, Germany
Email: baptiste.lambert@dlr.de

**G. Eckel**
German Aerospace Center (DLR)
Institute of Combustion Technology
Pfaffenwaldring 38–40
D-70569 Stuttgart, Germany

**P. Le Clercq**
German Aerospace Center (DLR)
Institute of Combustion Technology
Pfaffenwaldring 38–40
D-70569 Stuttgart, Germany

**M. Zahn**
GE Aviation Advanced Technology
Freisinger Landstrasse 50
85748 Garching, Germany

**T. Ripplinger**
GE Aviation Advanced Technology
Freisinger Landstrasse 50
85748 Garching, Germany

**ABSTRACT**

*In this study, we developed an efficient computer-aided design tool for scaling combustor designs. From a limited number of fluid simulations, an original design is scaled while preserving most of the flow properties, using a multi-objective optimization method and deep neural network or kriging surrogate models. The accuracy and robustness of the method were first tested with a simple geometry. Investigations have shown a strong sensitivity of the surrogate models to the sample distribution, which can be reduced using a strategic sampling method. Subsequently, the geometry was scaled down to a factor of two using both surrogate models while preserving most of the flow features.*

Keywords: multi-objective optimization, combustion, scaling.

**NOMENCLATURE**

$n_l$ Number of hidden layers of an ANN.

$n_n$ Number of nodes per hidden layers of an ANN.

$x_k$ Input of the k-th nodes of an ANN.

$w_k$ Weight on the k-th nodes of an ANN.

$y$ Output of an ANN.

$b$ ANN bias.

$D$ Collection of design used to build a surrogate model.

$\Omega$ Fluid domain.

$U$ Fluid velocity.

$\rho$ Fluid density.

$p$ Fluid pressure.

$Y_k$ Mass fraction of the k-th species.

$\boldsymbol{V}_k$ Diffusion velocity of the k-th species.

$\dot{\omega}_k$ Reaction rate of the k-th species.

$\dot{\omega}_T$ Heat release due to combustion.

$\tau$ Fluid stress tensor.

$\mu$ Fluid dynamic viscosity.

$h$ Fluid enthalpy.

$T$ Fluid temperature.

$\lambda$ Fluid heat conductivity.

$C_p$ Fluid specific heat capacity.

$\Delta h_f^0$ Fluid heat formation.

$\boldsymbol{S}_S^h$ Heat fluxes.

$\omega_i$ Chemical source.

$M_i$ Species molecular mass.

$N_s$ Number of species.

$v'_{i_r}$ Reaction stoichiometric coefficient.

$A_j$ The Arrhenius coefficient.

$\boldsymbol{x}_p$ Droplet position.

$\boldsymbol{u}_p$ Droplet velocity.

$T_p$ Droplet temperature.

$d_p$ Droplet diameter.

$c_p$ Droplet drag coefficient.

$\Delta h_{vap}$ Specific enthalpy of evaporation.

$c_{pl}$ Specific heat capacity.

$\dot{Q}_{surf}$ Surface heat flow rate.

$\mathbb{E}$ Relative error (eq. 12).

$\alpha$ Geometrical scale ratio of a current design compared to the reference design.

## 1. INTRODUCTION

In recent years, innovations in combustion systems have been driven by the need of the industry to comply with stricter environmental regulations while remaining competitive. Computer-aided design has become a standard in the industry to cost-effectively accelerate R&D. However, developing brand-new designs still comes at significant costs and risks. In this context, it is desirable to scale proven and tested designs to different power classes and sizes. Recent improvements in Selective Laser Melting and 3D printing[1, 2] facilitate the manufacturing of complex geometries at scale. The classical approach to scaling a design is to use dimensional analysis. From the governing equations of the flow, a set of necessary conditions for two designs to have similar flow properties can be extracted using the Buckingham Pi theorem. However, the scaling of a combustor is found to be practically impossible even with steady aerothermochemistry[3–6], due to the multi-physic and highly non-linear nature of the phenomena involved in combustion. In other words, there are too many simultaneous constraints on the similarities between scales that it is simply impossible to satisfy them all at the same time. Hence, a looser definition of similarity is generally used, where only design-critical quantities are preserved. To this end, we propose a multi-objective optimization method that focuses on scaling a design and preserving key quantities such as velocity and temperature distributions. This article is organized as follows. In Sec. 2 the surrogate models and optimization method used are described in depth. Sec. 3 details the numerical methods used to solve the flow dynamics in the combustion chamber. Subsequently, the accuracy

of the surrogate-based optimization method is assessed with a simple test case in Sec. 4. Finally, a reference design is down- and up-scaled using two different surrogate models.

## 2. METHODOLOGY

The proposed method to scale a combustor follows the same flow as a classical surrogate-based optimization. Instead of directly optimizing a physical variable of interest (e.g. fluid temperature), the deviation of this variable on a scaled geometry from its value in the reference case at scale 1:1 is set as an objective function. The method pipeline is summarized in Fig. 1.

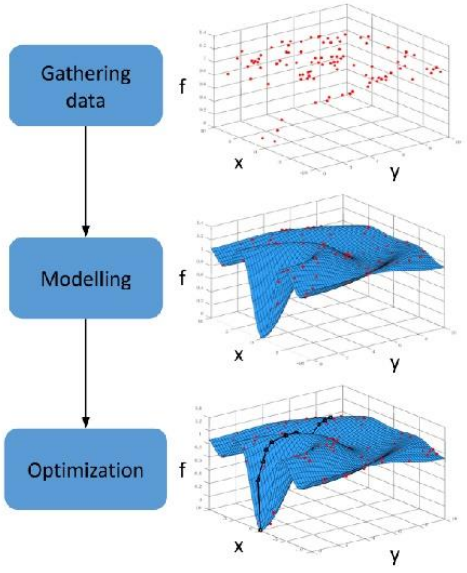


**FIGURE 1:** Illustration of the surrogate-based optimization pipeline for a single objective $f$ and two design parameters $x$ and $y$.

First, information on the dependence of the objective function $f$ (e.g. deviation of the fuel velocity field in a downscaled combustor compared to the original design) according to the design parameter ($x$, $y$), i.e. the operating variables with an influence on the objective function (e.g. fuel mass flow rate), need to be gathered. In this paper's case, Computational Fluid Dynamics (CFD) simulations are carried out on a down-scaled version of a given combustor for different sets of design parameters, to evaluate the objective functions in the design space. In Fig 1 each red dot is the result of a CFD for a given design ($x$, $y$), and the collection of designs D is near-randomly spread in the design space using the Latin Hypercube Sampling[7] (LHS). Details regarding the CFD simulations are discussed in Sec.

3. As CFD simulations tend to be very costly and time consuming, in particular for combustion problems, a surrogate model is built using D to approximate the value of the objective function where no CFD were carried out (visualized by the blue surface in Fig. 1). Building a surrogate model with a very limited number of samples can be a challenge based on the complexity of the functions modelled. Hence, two models from the library DAKOTA[8] (Sec. 2.2) were considered for cross-validation with our in-house model (Sec 2.1) in the benchmarks in Sec 4.

Finally, a multi-objective optimization method is run to find the optimal designs, using the surrogate model each time the objective functions need to be approximated. An example of the path taken by the optimizer in the design space is illustrated by the black line in Fig 1 to the optimal design $(x, y)_{opt} = \{(x, y),\ f(x, y) = \min f\}$.

### 2.1 Deep Neural Network Based Surrogate Model

The surrogate model proposed in this paper is built on an ensemble of Deep Neural Networks (DNN) trained to approximate the objective functions for given design parameters.

#### 2.1.1 Deep Neural Networks

Artificial Neural Networks (ANN) are composite functions organized into a logical sequence of layers. The inputs of the network are fed to the first layer. Subsequent layers take as input the output of some of the functions in the previous layer. Finally, the last layer in the network maps to output data also called labels. DNN generally refers to ANN with multiple hidden layers. The more complex structure of DNN enables the network to better model highly non-linear systems.

The layers are made of a collection of nodes also called neurons. The number of neurons on the input and output layer is equal to the number of design parameters and objective functions, respectively. For dense ANN, each neuron of a given hidden layer $l$ receives input from all of the nodes ($x_k$) on the previous layer $l-1$, or from a bias ($b$) and computes an output $y$. Each input has an associated weight $w$, which is assigned on the basis of its relative importance to other inputs. The node applies an activation function $f$ to the weighted sum of its inputs:

$$y = f\left(\sum_k w_k x_k + b\right). \quad (1)$$

For all nodes in the hidden layer, the ReLU activation function is used while a simple linear function is considered for the nodes on the output layer. With the backpropagation algorithm, the ANN weights ($w_k$) are modified to fit the ANN to the objective functions. In this paper, the Python-based framework Keras from Tensorflow[9] is used to build and train ANNs. Once the structure is set, the ANN is trained using the samples from the training data set (75% of $D$) and the Adam optimization method[10]. At each epoch, the training and validation losses are computed using the training and validation data set (10% of $D$), respectively. Both losses indicate the quality of the ANN and are used to decide when to stop the training. Finally, the accuracy of trained ANN is evaluated using the testing data set (the last 15% of $D$).

### 2.1.2 The DNN Surrogate Model

The optimal structure of an ANN for a given problem is a priori unknown. Therefore, different structures are randomly generated following a given template. In this paper, dense feed forward ANNs are made of $n_l$ identical hidden layers of $n_n$ nodes. For each ANN, $n_l$ and $n_n$ are set randomly such that $n_l \in [1,10]$ and $n_n \in [5,500]$. Due to the high cost of CFD simulations, the training data set available is relatively small compared to the data-set size generally used in ANN regressions. This increases the risk of overfitting the ANN to the data and so reduces the final ANN accuracy when fed new unseen inputs. To reduce this risk, different early stopping strategies can be used and we choose to interrupt the training of the ANN if the validation loss is val_loss $\leq 10^{-3}$.

Once a large collection of ANN is trained, the accuracy of each ANN is assessed by comparing the ANN predictions to the computed objective functions for all design points in the testing data set. The ANNs are then sorted according to their accuracy. All ANNs with an error less than 5% of the most accurate ANN, are selected. The average of the individual predictions of the selected ANNs is defined to be the overall prediction of the resulting surrogate model. This selection step also helps to eliminate over-fitting ANNs since they often performed poorly on the testing data set and they are, therefore, discarded.

In general, ANN-based regression models can be constructed either from a multiple output ANN, which approximates all the objective functions at once or from a collection of single output ANNs, each one approximating a specific objective function. If the

objective functions are not independent of each another, multiple output ANN models will generally be more accurate, since the ANN will model the dependencies between the objective functions.

### 2.2 Dakota Surrogate Models

Alternative surrogate models can be found in the open-source software DAKOTA developed at Sandia National Laboratory[8]. Among all the surrogate models provided in DAKOTA, two surrogate models were considered: the artificial neural network which is in nature comparable to our in-house model (Sec. 2.1.2) and the Gaussian process which was the most reliable during the benchmark test (see Sec. 4). Results obtained with the Gaussian process were used in particular as a reference to validate the DNN model.

#### 2.2.1 The Artificial Neural Network Model

The ANN surface fitting method in Dakota employs a Stochastic Layered Perceptron (SLP) artificial neural network based on the direct training approach of Zimmerman[11]. The SLP ANN method is designed to have a lower training cost than traditional back-propagation neural networks. The functional form of the SLP ANN model is:

$$\hat{f}(x) = \tanh(\tanh(xA_0 + \theta_0)\,A_1 + \theta_1), \qquad (2)$$

where $x$ is the current point in n-dimensional parameter space, and the terms $A_0$, $\theta_0$, $A_1$, $\theta_1$ are the matrices and vectors that correspond to the neuron weights and offset values in the ANN model. These terms are computed during the ANN training process. The SLP ANN is a non-parametric surface fitting method which means, unlike kriging (Sec. 2.2.2), the ANN surface is not guaranteed to exactly match the response values of the data points from which it was constructed[8]. Furthermore, one SLP ANN is trained for each objective function and so cannot model potential non-linear interactions between objective functions.

#### 2.2.1 Gaussian Process (GP) or Kriging Interpolation

The kriging interpolation model is based on the notion that the sample response values exhibit spatial correlation, with response values modelled via a Gaussian process around each sample location. In other words, samples taken close together are likely to have

highly correlated response values, whereas samples taken far apart are unlikely to have highly correlated response values. Kriging methods have wide utility due to their ability to accommodate irregularly spaced data, their ability to model general surfaces that have many peaks and valleys, and their exact interpolation of the given sample response values[12] . These methods are also known to generally perform well compared to peers when only small training datasets are available. The specific form of the kriging model implemented in Dakota is described in Giunta and Watson[13] and Romero, et al.[14], and it is based on the work of Koehler and Owen[15]. The form of the kriging model is

$$\hat{f}(x) = \hat{\beta} + r(x)^T \boldsymbol{R}^{-1}\left(f - \hat{\beta}\mathbb{1}\right), \tag{3}$$

where $\hat{\beta}$ is the generalized least squares estimate of the mean response; *r(x)* is an N×1 vector of correlations between the current point x and all N sample sites in the parameter space; $\boldsymbol{R}$ is the N×N correlation matrix of all N sample sites; *f* is the vector of N sample site response values; 1 is an N×1 unit vector. The terms in the correlation vector $r = (r_i)_{[\![1,N]\!]}$ and matrix $\boldsymbol{R} = \left(\boldsymbol{R}_{i,j}\right)_{[\![1,N]\!]}$ are computed using a Gaussian correlation function:

$$\begin{cases} r_i = \exp\left[-\sum_{k=1}^{n} \theta_k |\, x_k - x_k^{(i)}|^2\right], \\ R_{i,j} = \exp\left[-\sum_{k=1}^{n} \theta_k |\, x_k^{(i)} - x_k^{(j)}|^2\right], \end{cases} \tag{4}$$

with n the number of dimensions of the parameter space, and $\theta$ is the n×1 vector of correlation parameters, which is computed using maximum likelihood estimation. This kriging method works well for a sparse set of sample points in n-dimensional parameter space, but as the number of samples increases, the correlation matrix $\boldsymbol{R}$ becomes ill-conditioned which leads to a numerically unstable surrogate model.

### 2.3 Optimization Method

Once the surrogate models are built on a given set of training samples, optimal designs are located using the Nondominated Sorting Genetic Algorithm II (NGSA-II). NGSA-II is an improvement of NGSA developed by Deb[16] and is a typical multi-objective

evolutionary algorithm (MORAs). In particular, NGSA-II is a fast and elitist multi-objective genetic algorithm, that overcomes the three main criticism of the original NGSA approach: high computational complexity of non-dominated sorting, lack of elitism, and need for specifying the sharing parameters. The NGSA-II is directly integrated into our toolchain using the Python library PyMOO[17].

## 3. NUMERICAL METHOD

The optimization toolchain is tested on a bluff body spray flame. The reacting multi-phase flow system is solved by applying a two-way coupled RANS/Lagrange method based on two DLR in-house solvers. The gaseous phase is calculated using the pressure-based code THETA (Turbulent Heat Release Extension of the TAU Code) while the dispersed liquid phase is computed using the Lagrangian particle tracking code SPRAYSIM. As the focus of the paper at hand is on the optimization and scaling methodology, the CFD models were selected to be stable, reliable, and with short turnaround times (24 CPU.h). However, this is not a limitation of the toolchain per se as these models can be easily replaced by more sophisticated, more accurate but also more costly numerical models.

### 3.1 Governing Equation and Numerical Methods

#### 3.1.1 Gaseous Phase

The flow in the gaseous phase is considered an ideal gas and is modelled using steady Reynolds-Averaged Navier–Stokes (RANS) equations[18]:

$$\begin{cases} \partial_t\langle\rho\rangle + \partial_{x_i}(\langle\rho\rangle\widetilde{U}_i) = \dot{S}_\rho, \\ \partial_t(\langle\rho\rangle\widetilde{U}_i) + \partial_{x_i}\left(\langle\rho\rangle\widetilde{U}_i\widetilde{U}_j\right) = \partial_{x_i}\left(\langle\tau_{ij}\rangle - \langle\rho\rangle\widetilde{U_i''U_j''}\right) - \partial_{x_j}\langle\rho\rangle + \langle\rho\rangle g_i + F_i + \dot{S}_{\rho u}, \\ \partial_t(\langle\rho\rangle\widetilde{Y}_k) + \partial_{x_i}(\langle\rho\rangle\widetilde{U}_i\widetilde{Y}_k) = -\partial_{x_i}\left(\langle V_{k,i}Y_k\rangle + \langle\rho\rangle\widetilde{U_i''Y_k''}\right) + \langle\dot{\omega}_k\rangle + \dot{S}_{Y_i}\ for\ k \in [\![1, N_S]\!], \\ \partial_t(\langle\rho\rangle\widetilde{h}) + \partial_{x_i}(\langle\rho\rangle\widetilde{U}_i\widetilde{h}) = \langle\dot{\omega}_T\rangle + \langle\frac{Dp}{Dt}\rangle + \langle\tau_{ij}\partial_{x_j}U_i\rangle + \partial_{x_i}\left(\langle\lambda\partial_{x_i}T\rangle - \langle\rho\rangle\widetilde{U_i''h''}\right) - \partial_{x_i}\langle\rho\sum_{k=1}^{N_S}V_{k,i}Y_kh_k\rangle + \dot{S}_h, \\ \frac{Dp}{Dt} = \partial_t\langle p\rangle + \widetilde{U}_i\partial_{x_i}\langle p\rangle + \langle U_i''\partial_{x_i}p\rangle, \end{cases} \quad (5)$$

with $\boldsymbol{U}$, $\rho$, $T$, and $Y_i$ the velocity vector, the density, the temperature and the species mass fraction, respectively. The heat release due to combustion and k-th species reaction rate are denoted $\dot{\omega}_T$ and $\dot{\omega}_k$, respectively. The RANS equations assumed that any quantity $f$ may be split into a mass weighted-average (Favre average) and a fluctuation component as: $f = \tilde{f} + f''$,with $f'' = 0$. Reynolds averages are denoted $\langle . \rangle$.

Spray source terms are denoted $\dot{S}_\rho$, $\dot{S}_{\rho u}$, $\dot{S}_{Y_k}$, and $\dot{S}_h$. Gravity is also considered and included in the total external force term $\boldsymbol{F}_i$. The gaseous phase is assumed Newtonian. Therefore, the stress tensor $\tau$ can be related, as a first approximation, to the averaged velocity gradients, resulting in the Boussinesq approximation. In Cartesian coordinates it can be written as follows:

$$\tau_{ij} = \mu\left(\partial_{x_j} U_i + \partial_{x_i} U_j\right) - \delta_{ij}\left(\frac{2}{3}\mu\right)\partial_{x_k} U_k \qquad (6)$$

with $\mu$ the dynamic viscosity of the gaseous phase. The turbulent Reynolds stress tensor $\boldsymbol{T} = \langle\rho\rangle\widetilde{U_i''U_j''}$ is calculated using the k-ω SST closure equations. Since only low-Mach number flows are considered, the static enthalpy $h$ is defined as follows:

$$h(T, Y_i) = \sum_{i=1}^{N_s} Y_i\left(\Delta h_{f_i}^0 + \int_{T^0}^{T} C_p(T')dT'\right), \qquad (7)$$

where $\Delta h_{f_i}^0$ and $C_p$ are the heat formation and the specific heat capacity at constant pressure, respectively. The heat fluxes $\boldsymbol{S}_S^h$ are modelled according to the Fourier's law:

$$\boldsymbol{S}_S^h = -\lambda\nabla T + \rho\sum_{i=1}^{N_s} h_i Y_i \boldsymbol{V}_i, \qquad (8)$$

with the heat conductivity denoted $\lambda$. The convective velocity $\boldsymbol{v}$ is defined as the sum of the flow velocity $\boldsymbol{U}$ and the diffusion velocity $\boldsymbol{V}$ which can be modelled using Fick's law.

Lastly, the chemical source term $\omega_i$ is defined for the $N_s$ species as the sum over all $N_r$ elementary reactions:

$$\omega_i = M_i \sum_{r=1}^{2N_r} \left(\nu''_{i_r} - \nu'_{i_r}\right) k_r \prod_{j=1}^{N_s} \left[A_j\right]^{\nu'_{j_r}}, \qquad (9)$$

where $M_i$, $\nu'_{i_r}$ and $A_j$ are the species molecular mass, the reaction stoichiometric coefficient, and the Arrhenius coefficients, respectively.

The chemical reactions are modeled using the Eddy Dissipation Model (EDM)[19], which is based on the assumption that the chemical reactions occur infinitely fast compared to the time scale of the transport process. This assumption, also known as Fast Chemistry, is often made for non-premixed flame as it reduces the complexity of the chemical model.

The governing equations 5 are discretized and solved using 3D unstructured dual grids allowing for the simulation of flows in and around complex geometries. A second-order upwind scheme and a central scheme are used to calculate the convective and diffusive fluxes, respectively. The pressure-velocity coupling is based on the SIMPLE algorithm[20]. The FGMRES method is preconditioned by a single multigrid V-cycle and the BiCGStab method with Jacobi preconditioning is applied to solve the Poisson equation for the pressure correction and the other transport equations, respectively.

### 3.1.2 Liquid Phase

For the liquid phase, droplet size and velocity distributions were prescribed approximating the outcome of the atomization process. Droplet dispersion and evaporation were calculated using a Lagrangian particle tracking approach based on the following ordinary differential equations (ODEs):

$$\begin{cases} \dfrac{d\boldsymbol{x}_p}{dt} = \boldsymbol{u}_p, \\ \dfrac{d\boldsymbol{u}_p}{dt} = \dfrac{3}{4}\dfrac{c_d}{d_p}\dfrac{\rho_g}{\rho_l}\left\|\boldsymbol{u}_g - \boldsymbol{u}_p\right\| \cdot \left(\boldsymbol{u}_g - \boldsymbol{u}_p\right) + \left(1 - \dfrac{\rho_g}{\rho_p}\right)\boldsymbol{g}, \\ \dfrac{dT_p}{dt} = \dfrac{6}{\pi d_p^3}\dfrac{\dot{m}_{surf}\Delta h_{vap}\left(T_p\right) + \dot{Q}_{surf}}{\rho_l c_{pl}}, \\ \dfrac{d\left(d_p\right)}{dt} = \dfrac{d_p}{3}\dfrac{1}{\rho_l}\dfrac{\rho_l}{dt} - \dfrac{2}{\rho_l}\dfrac{\dot{m}_{vap}}{\pi d_p^2}, \end{cases} \qquad (10)$$

where $\boldsymbol{x}_p$, $\boldsymbol{u}_p$, $T_p$, $d_p$, and $c_d$ are the position, velocity, temperature, diameter and drag coefficient of the tracked droplet. Drag formulae were taken from Clift and Grace[21]. The indices g and l stand for gas and liquid, respectively. The specific enthalpy of evaporation, the specific heat capacity, and the surface heat flow rate are denoted $\Delta h_{vap}$, $c_{pl}$, and $\dot{Q}_{surf}$, respectively. Droplet evaporation was modelled by a variant of the evaporation model of Abramzon and Sirignano[22]. Relying on the point source approximation, the filtered spray source terms in the gas phase equations (Eq. 5) were computed following the approach presented in [23].

### 2.3 Description of the Benchmark Reference Case

The toolchain was tested using a simple 100×20×20 cm combustion chamber design with a 10 cm cubic obstacle at the center (Fig. 2). Pressurized air at $P_{air}$ = 20 bar enters the chamber by the lower boundary (Z = 0) at a mass flow rate $\dot{m}_{air} = 0.5$ kg/s and temperature $T_{air} = 800$ K. The chamber top boundary (Z = 1 m) is set as a pressure outlet, while all the other boundaries are considered adiabatic walls. Jet-A fuel is injected at an angle of 45◦ with respect to Oz through an annulus (4.5 mm and 5 mm inner and outer radius, respectively) at the mass flow rate $\dot{m}_{spray} = 5.0 \cdot 10 - 3$ kg/s, and temperature $T_{spray} = 300$ K. Droplet sizes were prescribed following a Rosin-Rammler distribution with a constant spread of $q$ = 2.8. The kerosene properties were taken from the single species surrogate formulation of Rachner[24]. The whole fluid domain Ω was meshed using 3 million tetrahedral elements. A CFD simulation of the reference case is run to obtain the fluid density, temperature and velocity field needed to evaluate the toolchain objective functions described in Sec. 3.3

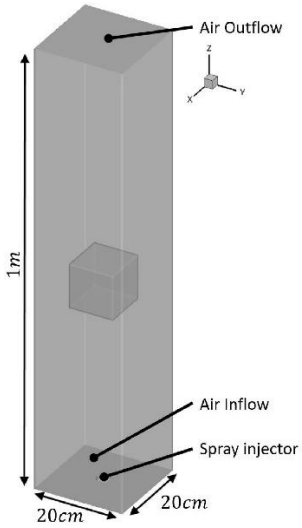


**FIGURE 2:** 3D representation of the reference geometry.

### 2.3 Design Parameters and Objective Functions

All optimizations were performed varying three design parameters: the air mass flow rate $\dot{m}_{air}$, the liquid mass flow rate $\dot{m}_{spray}$, and the droplet scaling ratio *DS*. For example, if *DS* < 1, all droplet sizes of the given design are reduced (i.e. scaled by a factor *DS*) but still follow the same nondimensional droplet size distribution as in the reference case. The reference case was defined to have the following design parameter values: $\left(\dot{m}_{air}, DS, \dot{m}_{spray}\right)_{ref} = (0.5\ \text{kg/s}, 1\ , 5.0 \cdot 10^{-3}\ \text{kg/s})$. The choice of these design parameters is based on their influences on the gaseous mixture velocity, temperature, and density that we wish to preserve while scaling. More design parameters could be considered such as the type of fuel. However, the cost of the method increases significantly as more design parameters are considered because of the increasing number of CFD needed to build accurate surrogate models.

The choice of objective functions varies from project to project. We choose to look for designs with similar features compared to a reference case. Thus, each objective function is defined as the average difference between a current case and the reference:

$$f_{\phi,I} = \frac{1}{|I|}\int_I |\phi - \phi_{ref}|dv \sim \frac{1}{|I|}\sum_{x\in I} |\phi(x) - \phi_{ref}(x) \tag{11}$$

where *I* is a sub-domain of $\Omega$, $\phi$ and $\phi_{ref}$ are selected features (e.g. the temperature field) in the current and reference case, respectively. $f_{\phi,I}$ is numerically approximated using the value of $\phi$ computed on the grid element *x* of the domain *I*. If not stated otherwise, ten objective functions $f_{\phi,I}$ are built in the optimization toolchain based on the density, temperature and velocity fields of the gaseous mixture in the two planes: $\phi = \{\rho, T, U_x, U_y, U_z\}$ and the two center planes *I* = {YZ,XZ}.

## 4. BENCHMARKS

At first, the toolchain was used to reproduce a reference case without any geometrical scaling. By knowing exactly the reference design parameter values, the accuracy and robustness of the toolchain can be accurately assessed without the extra layer of uncertainty that comes with scaling. Due to randomization at different stages of the optimization process, such as the distribution of the design

samples, optimizations using identical inputs can give slightly different results. Therefore, five optimizations were run independently to obtain an averaged approximation of the optimal solution.

In the three sections below, the accuracy of the toolchain is assessed by computing the relative error $\mathbb{E}$ of the optimum found compared to the reference:

$$\mathbb{E} = \frac{1}{3}\sum_{k=1}^{3}\frac{|x_k^{opt} - x_k^{ref}|}{x_k^{ref}}, \qquad (12)$$

With $x_k^{opt}$, $x_k^{ref}$ the design parameters of the optimum and reference, respectively.

### 4.1 Surrogate Model Sensitivity to the Number of Samples

The size of the training data set usually has a significant impact on the quality of the surrogate model. The more samples are computed, the more information is gathered on the objective functions and the better the training of the surrogate model is expected. However, as CFD simulations are expensive, only a limited number of samples can be realistically computed. This results in training data sets, which tend to be relatively small compared to standard ones, especially for ANNs. To evaluate the sensitivity to the number of samples, several optimization runs were performed using different data set sizes. Furthermore, the surrogate models could take advantage of objective function symmetries based on the design space definition, which would artificially increase the accuracy of the method. Therefore, the sensitivity analysis was carried out on two different design domains to reduce symmetry biases. Figures 3 and 4 show the evolution of the error $\mathbb{E}$ according to the number of samples generated with the LHS method looking at two different design domains.

For all senarii, the ANN model from Dakota performs poorly compared to the two other models. This underperformance could be explained by the small size of the training data set or the simple structure of SLP ANN, compared to the ANNs used in the DNN based model, which fails to model highly non-linear objective functions. Therefore, the ANN model will not be considered further in this study.

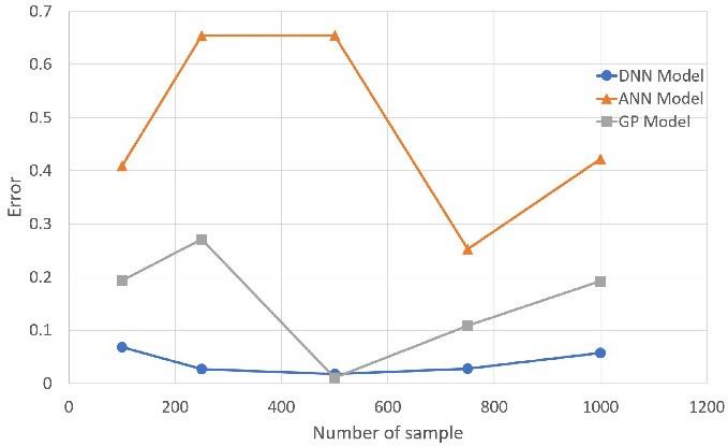


**FIGURE 3:** Evolution of the error $\mathbb{E}$ depending on the number of samples generated with the LHS method in the design domain [0.08 kg/s, 0.8 kg/s]×[0.8, 1.2]×[0.001 kg/s, 0.01 kg/s].

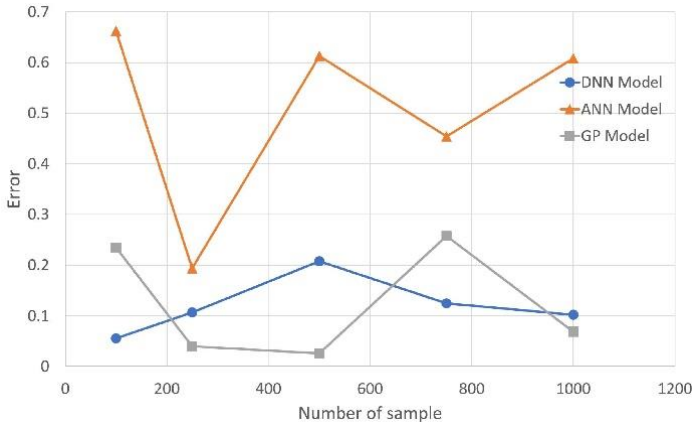


**FIGURE 4:** Evolution of the error $\mathbb{E}$ depending on the number of samples generated with the LHS method in the design domain [0.08 kg/s, 0.8 kg/s]×[0.8, 1.2]×[0.001 kg/s, 0.01 kg/s].

The two remaining models have similar accuracy. In Fig. 3, the DNN model appears to perform better than the GP model. However, the design parameters of the reference are exactly in the middle of the design domain. The DNN model seems to take advantage of some objective function symmetries that significantly improve its accuracy compared to cases where the design domain is not centered onto the solution (Fig. 4). Overall, there is no clear correlation between the model accuracy and the size of the training data set, within the size range considered here. Therefore, increasing the size of the training data set, in the feasibility range imposed by the CFD costs, does

not guarantee a more accurate solution, and more effort should rather be invested in refining other model hyperparameters to improve the method's accuracy.

### 4.2 Sensitivity to Initial Sample Distribution

As seen previously, the size of the data set is not directly correlated to the method's accuracy. However, it was observed that the distribution of these samples in the design space influenced the quality of the surrogate model training. To highlight this influence, three different distributions for the training data set have been generated using three different seed numbers with the LHS method. A seed number is an integer used to initialize the pseudorandom sample generator built in the LHS method. It enables the reproduction of the experiences as two independent runs of the LHS method using the same seed number will generate the same sample distribution.

The relative errors $\mathbb{E}$ obtained using either the DNN surrogate model or the GP model are summarized in the table 1. With either surrogate model, the results clearly show the influence of the sample distribution on the optimization accuracy. Similar results were found with up to 10000 samples, different design spaces sizes, and positions relative to the reference (not shown here).

| Seed Number | DNN model | GP Model |
|---|---|---|
| 42 | $2.69 \pm 1.05\%$ | $1.32 \pm 0.40\%$ |
| 420 | $2.62 \pm 1.22\%$ | $50.06 \pm 1.20\%$ |
| 4200 | $1.44 \pm 6.18\%$ | $8.48 \pm 0.22\%$ |

**TABLE 1:** Relative error $\mathbb{E}$ for a seed number variation with 500 samples generated with the LHS method in the design domain [0.1 kg/s, 1.0 kg/s]×[0.8, 1.2]×[0.001 kg/s, 0.01 kg/s]. The reported values are the averaged relative errors and standard deviations obtained out of five independent optimization runs.

The rather small training data sets could be the reason for this apparent sensitivity of the results to the sample distribution. However, data sets with more than a thousand samples are unrealistic due to the CFD simulation cost. Furthermore, there is no apparent way to

know a priori which seed number would generate an optimal distribution. Therefore, an alternative strategy to the LHS method is needed to sample strategically in the design space in order to minimize the influence of the sample distribution on the optimization results.

### 4.3 A Nested Re-sampling Method

To reduce the influence of the design sample distribution and thus improve the accuracy of the toolchain, a nested re-sampling method was additionally tested. First, samples are generated in the design space using the LHS method and then simulated to gather information on the objective functions. Based on the assumption that optimal designs are more likely to be close to samples with the current minimum error, the design domain is reduced to contain only 50% of the initial samples which have the smallest mean objective function. New samples are then added to the design domain using the LHS method. The process is repeated until the maximum number of computable samples is reached. To avoid computing the same design twice, new samples too close to an existing sample are discarded before simulating the batch of new design samples.

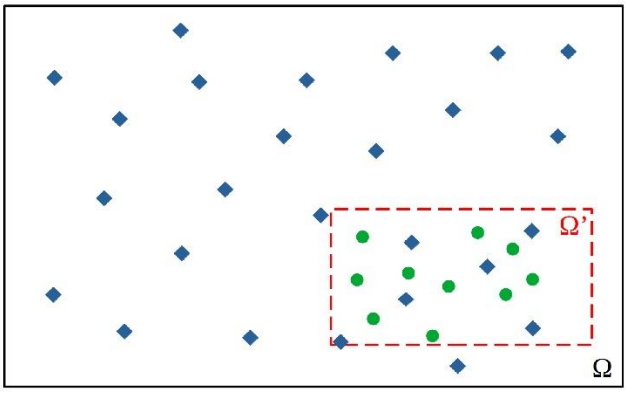


**FIGURE 5:** Visualization of the resampling method in a region Ω′ of the design space Ω. New samples (circles) are added based on the location of existing samples (diamonds) and their averaged objective functions.

Two scenarii using the re-sampling method were tested. Starting with the same initial 400 design sample distribution, 100 new samples were added either all at once in one resampling cycle or in two re-sampling cycles, each increasing the number of samples by 50. Both scenarii were compared to the 500 samples equivalent obtained without re-sampling. In all three cases, a set of runs was

conducted with the same five LHS seed numbers {42, 420, 4200, 42000, 420000}. Table 2 summarizes the average errors E of these runs.

| Sampling | DNN model | GP Model |
|---|---|---|
| 500 | $2.25 \pm 0.58\%$ | $19.96 \pm 21.49\%$ |
| 400 + 100 | $3.38 \pm 2.16\%$ | $11.15 \pm 6.70\%$ |
| 400 +50 + 50 | $2.61 \pm 0.26\%$ | $6.58 \pm 6.82\%$ |

**TABLE 2:** Relative errors of the optimized design compared reference design values for no, one or two re-sampling cycles. Error values are the average of $\mathbb{E}$ obtained with LHS seed numbers {42, 420, 4200, 42000, 420000}. The reported values are the averaged relative errors and standard deviations obtained out of five independent optimization runs.

It is noteworthy that a re-sampling cycle using the DNN surrogate model and concentrating the design samples in a neighborhood close to the last found optimum does not necessarily improve the accuracy compared to the case without re-sampling (see errors after the first resampling cycle). Furthermore, there is the risk to focus on a specific region of the optimization space and thus finding only a local optimum. Nevertheless, results in Tab. 2 show the potential of re-sampling methods to improve the toolchain accuracy and robustness. The uncertainty is improved by 38% with DNN models and 68% with the GP model if two re-sampling cycles are performed. In other words, the nested re-sampling method decreases the sensitivity of the optimization error to the LHS seed number, i.e. the sample distribution

## 5. GEOMETRICAL SCALING

In this section, the toolchain is used to find scaled designs with similar thermo-physical properties to the reference case described in sec. 3.2. Simulations are made with combustion chamber geometry scaled by a factor $\alpha$, while all geometrical features remained identical to the reference.

### 5.1 Scaling Approach

Flow similarity to the reference case was measured using ten objective functions $f_{\phi,I}$ with $\phi = \{\rho, T, U_x, U_y, U_z\}$ and $I$ = {YZ, XZ}. With the same design parameters $(\dot{m}_{air}, DS, \dot{m}_{spray})$ as in the previous section, five different scale factors were investigated: $\alpha \in$ {1.25, 0.875, 0.75, 0.625, 0.5}. Based on the observation made in section 4.3, surrogate models were built from a 500-sample data set generated from an initial distribution of 300 samples using the LHS method and two refinement cycles of 100 samples using the nested re-sampling method.

Figure 6 shows the locations of each optimal design found for α ∈ {1.25,0.875,0.75,0.625,0.5} using the DNN (circle) or GP (triangle) surrogate models. Both surrogate models give similar results, except for the case $\alpha$ = 0.5.

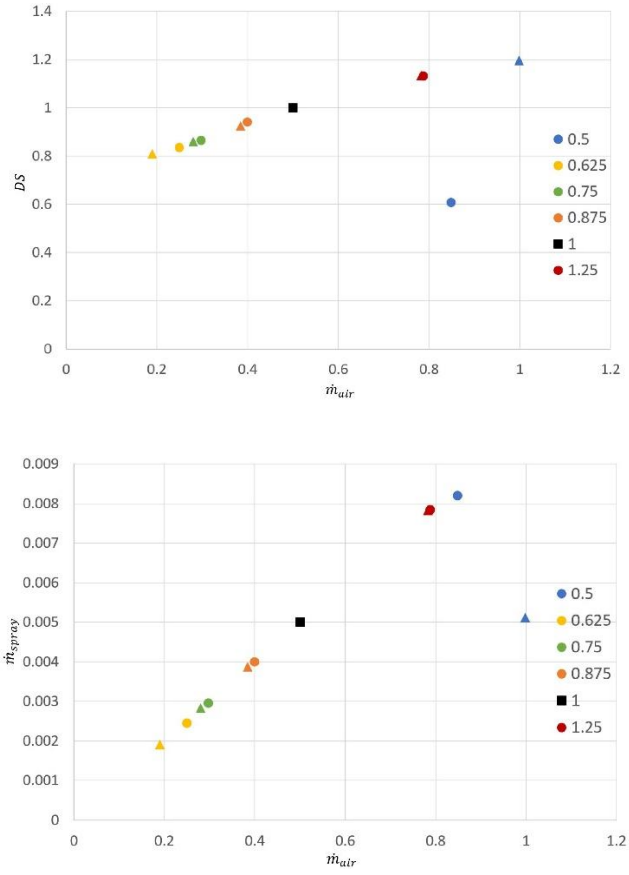


**FIGURE 6:** Distribution of the optimal design parameter found using the DNN (circle) and GP (triangle) surrogate model scaling ratio α ∈ {1.25, 0.875, 0.75, 0.625, 0.5}.

To assess the accuracy of the surrogate models, the objective functions $f_{\phi,I}$ were considered. For each optimal design, $f_{\phi,I}$ were normalized by the average value of $\phi$ in the domain $I$ of the reference case. Then, the global accuracy of the surrogate models was evaluated at each scale $\alpha$ by computing the average value of the normalized objective functions, displayed in figure 7. For $\alpha \in [0.625, 1.25]$, the accuracy of the surrogate models is similar and decrease as the $|1 - \alpha|$ increase. For $\alpha = 0.5$, the average of the normalized objective functions is one order of magnitude higher than at any other scale (31.07 and 85.14 with the DNN and GP model, respectively). With large size variations, the flow dynamics change drastically from the reference which makes it challenging for the toolchain to find an optimal design by varying only three parameters, even with a loose definition of similarity. Optimal designs could be found for extreme α by considering more design parameters and structural alterations of the geometry.

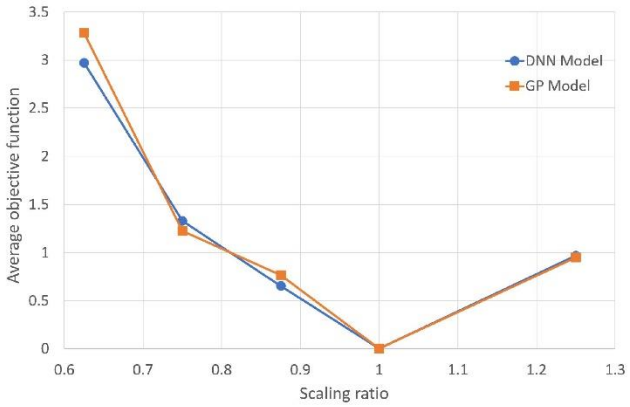


**FIGURE 7:** Evolution of the objective function averaged according to the scaling ratio using the DNN and GP model. The case $\alpha = 0.5$ is arbitrarily excluded for visibility.

### 5.2 Spatial Distribution of Similarities and Dissimilarities

As the objective functions in this study reflect only the overall error, it is advisable to also gain insight on the distribution of the differences between a scaled design and the reference. Thereby, it should be avoided that a design was considered similar to the reference while having high differences concentrated in a small region of the domain. Figure 8, 9, and 10 show exemplarily for a scaling factor α = 0.75 the local deviation of the density, temperature, and the Z-velocity field optimized by the DNN model (left) and the GP model (middle) from their respective reference fields (right). Negligible differences are blanked out to highlight regions where the computed optimal solution is dissimilar to the reference.

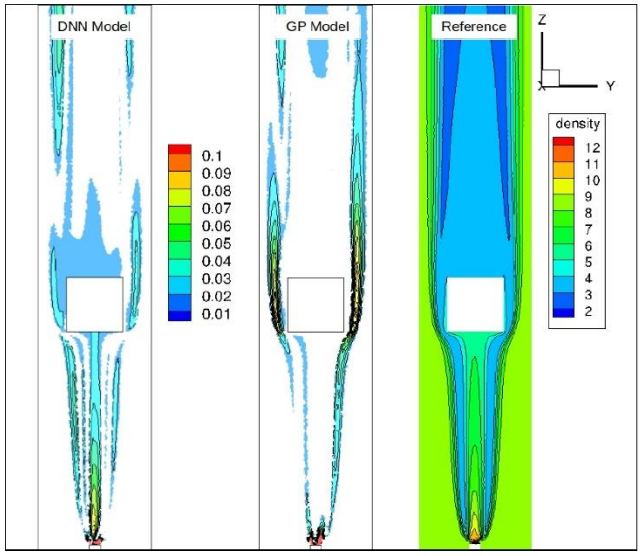


**FIGURE 8:** L2 error of the density field obtained with $\alpha = 0.75$ in $I = YZ$ and the DNN (left) or GP (middle) surrogate model, compared to the reference field (left). The density field of the reference slice is displayed in kg/m$^3$.

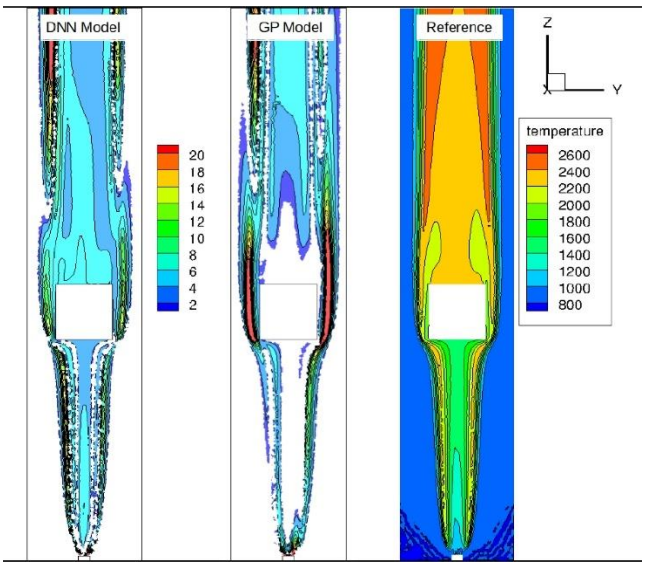


**FIGURE 9:** L2 error of the temperature field obtained with $\alpha = 0.75$ in $I = YZ$ and the DNN (left) or GP (middle) surrogate model, compared to the reference field (left). The temperature field of the reference slice is displayed in K.

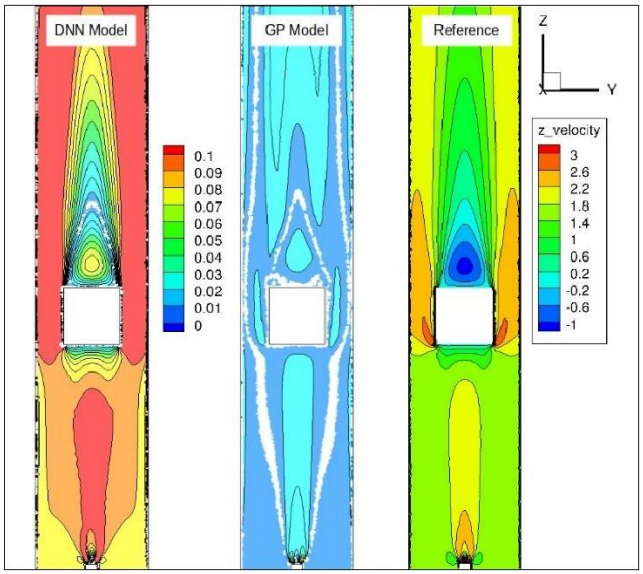


**FIGURE 10:** L2 error of the z-velocity field obtained with $\alpha = 0.75$ in $I = YZ$ and the DNN (left) or GP (middle) surrogate model, compared to the reference field (left). The velocity field of the reference slice is displayed in m/s.

Therefore, it might be advisable to use both models for cross-model validation. This is a viable approach as it does not result in a significant cost increase of the overall process being primarily dominated by the CFD simulations. Furthermore, the re-sampling method is independent of the surrogate models.

## 6. CONCLUSION AND OUTLOOK

It was shown that surrogate-based optimization can be a valuable tool for design optimization and scaling. Even though perfect similarity from a mathematical standpoint is difficult (or impossible) to achieve, a methodology was presented to scale an existing design while preserving most of the flow features. A simple geometry was scaled down to a factor of two using the Gaussian process and deep neural network-based surrogate models. Both surrogates showed significant sensitivity to the sample distribution and gave similar results. Presumably, the DNN accuracy could be improved with a more advanced network structure and hyperparameters optimization. Future works will focus on the improvement of the sampling method and applying the methodology to a more realistic combustor with geometrical features defined as design parameters.

**ACKNOWLEDGEMENTS**

The research leading to these results has received funding from the German Federal Aviation Research Program LuFo V-3.

**AUTHOR CONTRIBUTIONS**

B. Lambert: Methodology, Software, Formal analysis, Investigation, Writing - Original Draft.

G. Eckel: Methodology, Conceptualization, Writing - Review & Editing.

P. Le Clercq: Conceptualization, Supervision, Funding acquisition.

M. Zahn and T. Ripplinger: Supervision, Project administration, Funding acquisition.

**REFERENCES**

[1] I. Alexandridis, A. Avranas, K. K., and J. Kechagias, 2018. "Design and 3d printing of a two-stroke engine with a low cost 3d printer: A case study". *Conference: 3rd International Conference on: Applied Physics, System Science and Computers At: Dubrovnik, Croatia Affiliation: Technological Educational Institute of Thessaly*, 09.

[2] J. Gray, and C. Depcik, 2020. "Review of additive manufacturing for internal combustion engine components". *SAE International Journal of Engines*, **13**, 0.

[3] J. Hulka, 2008. "Scaling of performance in liquid propellant rocket engine combustion devices". *44th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*.

[4] D. Stewart, 1956. "Scaling of gas turbine combustion systems". *Selected Combustion Problems, II, AGARD Combustion Colloquium*, pp. 384-413.

[5] S. S. Penner, 1955. "Similarity analysis for chemical reactors and the scaling of liquid fuel rocket engines". *Combustion Research and Reviews, AGARD*, pp. 140-162.

[6] O. Delabroy, F. Lacas, B. Labegorre, and J.-M. Samaniego, 1998. "Parametres de similitude pour la combustion diphasique". *Revue Générale de Thermique*, **37**(11), pp. 934-953.

[7] M. D. McKay, R. J. B., and W. J. Conover, 1979. "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code". *Technometrics. American Statistical Association*, **21**(2), pp. 239-245.

[8] B. M. Adams, W. J. Bohnhoff, K. R. Dalbey, M. S. Ebeida, J. P. Eddy, M. S. Eldred, R. W. Hooper, P. D. Hough, K. T. Hu, J. D. Jakeman, M. Khalil, K. A. Maupin, J. A. Monschke, E. M. Ridgway, A. Rushdi, D. T. Seidl, J. A. Stephens, and J. G. Winokur, 10/2019. *Dakota, a multilevel parallel objectoriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Developers Manual*.

[9] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, 2015. *Tensor-Flow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org*.

[10] D. P. Kingma, and J. Ba, 2014. "Adam: A method for stochastic optimization". *International Conference on Learning Representations*.

[11] D. C. Zimmerman, 09/1996. "Genetic algorithms for navigating expensive and complex design spaces". *Technical report, Final Report for Sandia National Laboratories.*

[12] A. A. Giunta, L. P. Swiler, S. L. Brown, M. S. Eldred, M. D. Richards, and E. C. Cyr, 09/2006. "The surfpack software library for surrogate modeling of sparse irregularly spaced multidimensional data". *AIAA*, **3.**

[13] A. A. Giunta, and L. T. Watson, 1996. "A comparison of approximation modeling techniques: polynomial versus interpolating models". *AIAA 98-4758 in Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization,* pp. 392-404.

[14] V. J. Romero, L. P. Swiler, A. A. G., 2004. "Construction of response surfaces based on progressive-lattice-sampling experimental designs". *Structural Safety*, **2**(2), pp. 201-219.

[15] J. R. Koehler, and A. B. Owen, 1996. *Computer experiments*, Vol. 13. Elsevier Science.

[16] K. Deb, A. Pratap, S. A., and T. Meyarivan, 2002. "A fast and elitist multi-objective genetic algorithm: NSGA-II". *IEEE Transactions on Evolutionary Computation*, **6**(2), pp. 182-197.

[17] J. Blank, and K. Deb, 2020. "pymoo: Multi-objective optimization in python". *IEEE Access*, **8**, pp. 89497-89509.

[18] T. Poinsot, and V. D., 2005. *Theoretical and Numerical Combustion*. Edwards.

[19] B. Magnussen, and B. Hjertager, 1976. "On mathematical modelling of turbulent combustion with special emphasis on soot formation and combustion". *In Proceedings of the 16th International Symposium on Combustion*, pp. 719-729.

[20] L. S. Caretto, A. D. Gosmam, S. V. P., and D. B. Spalding, 1973. "Two calculation procedures for steady, three-dimensional flows with recirculation". *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics*, pp. 60-68.

[21] R. Clift, JR. Grace, M. W., 1978. *Low-Dissipation Low-Dispersion Second-Order Scheme for Unstructured Finite Volume Flow Solvers*. Academic Press New York.

[22] B. Abramzon, and W. Sirignano, 1989. "Droplet vaporization model for spray combustion calculations". *International Journal of Heat and Mass Transfer*, **32**(9), pp. 1605-1618.

[23] G. Eckel, M. Rachner, P. L. Clercq, and M. Aigner, 2016. "Semi-empirical model for the unsteady shear breakup of liquid jets in cross-flow". *Atomization and Sprays*, **26**(7), pp. 687-712.

[24] M. Rachner, 1998. Die stoffeigenschaften von kerosin jet a-1. DLR-Mitteilung 98-01, DLR - German Aerospace Center