

Machine Learning based reduced models for the aerothermodynamic and aerodynamic wall quantities in hypersonic rarefied conditions

Marc Schouler^{a,*}, Ysolde Prévèreaud^a, Luc Mieussens^b

^aONERA / DMPE, Université de Toulouse, F-31055 Toulouse, France

^bBordeaux INP, Univ. Bordeaux, CNRS, IMB, UMR 5251, F-33400 Talence, France

Abstract

Since their development at the end of the 50s, panel methods were widely used for the fast simulation of aerospace objects reentry. Although improvements were proposed for the continuum regime formulations, the bridging functions usually employed in the transitional regime did not go through major changes since then. With the current interest in designing Very Low Earth Orbit (VLEO) satellites and more efficient reentry vehicles, a greater level of preciseness is now required for the fast computation of the aerodynamic (AED) and aerothermodynamic (ATD) wall quantities in rarefied regime. In this context, this paper presents a new way of dealing with the major limitations of the bridging methods by using Machine Learning (ML) based surrogate models. Hence, kriging models are first developed to compute the pressure and heat flux stagnation coefficients ($C_{p,stag}$, $C_{h,stag}$) in the transitional regime and Artificial Neural Networks (ANN) are then trained to compute the pressure, friction and heat flux coefficient distributions (C_p , C_f , C_h) in any point of an object surface in the rarefied portion of its reentry. Both models are finally evaluated by comparison with DSMC computations representative of reentry conditions.

Keywords: Surrogate models, Aerothermodynamic, Aerodynamic, Rarefied hypersonic flow, Earth reentry

1. Introduction

Recent progress in air-breathing electric propulsion make Very Low Earth Orbit (VLEO) satellites a reality in the upcoming years [1]. In this context, orbits as low as 100 km are deemed possible for Earth Observation (EO) satellites [2, 3, 4]. In such conditions, the drag and heat flux estimations are paramount for the proper design of such satellites. In the meantime, the development of efficient and reusable reentry vehicles manoeuvrable at high altitudes requires an accurate computation of the aerodynamic coefficients for a precise prediction of their trajectory. Since high fidelity tools cannot be used to simulate the whole life cycle of a satellite nor the aerodynamic (AED) and aerothermodynamic (ATD) constraint at every moment of an object's reentry, reentry codes are usually used instead.

There are two categories of reentry codes. The first one is called oriented object codes which are the most common and which directly compute integrated drag C_D and heat flux C_H coefficients for some predefined geometries. The second category corresponds to spacecraft oriented codes. These are based on panel methods [5, 6, 7] whose strength is to rapidly compute the AED and ATD coefficients of any object and at any instant of its reentry. In this aim, according to the rarefaction regime (free molecular to continuum), the corresponding simplified formulations are used to compute local quantities (C_p , C_f , C_h) that can be integrated over the surface to obtain

*. Corresponding author

Email address: schouler.marc@gmail.com (Marc Schouler)

global coefficients (C_L , C_D , *etc.*). In the transitional regime, bridging functions $f_b(\cdot)$ are used according to the following relationship:

$$C_{x,\text{trans}} = C_{x,\text{cont}} + (C_{x,\text{fm}} - C_{x,\text{cont}})f_b(Kn), \quad (1)$$

where Kn is the Knudsen number that measures the degree of rarefaction of the gas, and the bridging function verifies $f_b(Kn_{\text{cont}}) = 0$, $f_b(Kn_{\text{fm}}) = 1$ where Kn_{cont} and Kn_{fm} are the Knudsen numbers indicating the limits of the continuum and free molecular regime respectively. Moreover, $C_{x,\text{trans}}$ is any global or local coefficient in the transitional regime (*i.e.* such as $Kn_{\text{cont}} \leq Kn \leq Kn_{\text{fm}}$) and $C_{x,\text{cont}}$, $C_{x,\text{fm}}$ respectively denote the value in the continuum and free molecular regimes.

Since the first bridging function proposed by Martino [8], several formulations were derived [9, 10, 11, 12, 13]. Their application to flight data [14] have for instance proved the relevance of such method for the reconstruction of the C_A/C_N ratio measured during the Viking I entry of the Mars atmosphere. Nevertheless, when the bridging function is not parametrized for a specific object, its generalisation capacities suffer from the relative aspect of the rarefaction parameter ($Kn = \lambda_\infty/L_{\text{ref}}$) which is highly dependent on the choice of the reference length. Moreover, the non-absolute value of the continuum and free molecular limits in terms of Knudsen number is another important limitation. Finally, as evident from equation (1), the error due to the distribution formulations in the limit regimes fatally propagates through the transitional regime.

Thus, in the aim of improving the preciseness of the panel methods in the rarefied regime, efforts were carried out by Falchi in the frame of his PhD [15, 16, 17, 18]. However, Falchi's work mostly focuses on sharp geometries (cube, cylinder, ...) characteristic of debris shapes while in our work, we are more concerned with aerodynamic shapes. The last decades, Machine Learning (ML) based techniques became highly popular in engineering [19, 20] and especially in fluid mechanics where both global [21] and local [22] surrogate models were developed for hypersonic conditions. More recently, reduced models were even built for hypersonic rarefied conditions. Indeed, in [23], Walsh *et al.* trained Radial Basis Functions (RBF) to reduce the drag coefficient through shape optimisation of VLEO satellites. In [24], Sia and Mehta built kriging models for the drag coefficient computation of simple debris shapes (cube, sphere, cylinder) and neural networks for the computation of their heat flux distribution. Nonetheless, these models are restrictive and expensive in the sense that for each geometry at a fixed attitude, 100 DSMC computations are used to build the corresponding kriging and neural network models.

In this work, the objective was to develop surrogate models theoretically applicable to any kind of geometry and to any level of rarefaction. Hence, a two-step approach was adopted. The first step consists in computing the stagnation pressure and heat flux as a function of the flight conditions with kriging models. Then, Artificial Neural Networks (ANN) are fed with the adequate stagnation quantities and geometric variables in order to compute the pressure, friction and heat flux coefficients at any point on the surface of the geometry. The paper is organized as follows. The first sections give an overview of all the computational techniques involved in this work. Hence the current approaches and their limitations are first introduced. Then, the chosen machine learning methods are briefly described. This part ends with the presentation of the computational approaches used for the training and the evaluation of the developed surrogates. In the next section, the design variables and the Design of Experiments (DOE) are discussed and described. In the fifth section, the results of all surrogate models are presented and compared to the results obtained with the Direct Simulation Monte-Carlo (DSMC) code **SPARTA** [25] and the vehicle oriented reentry code **ARES** [7]. Finally, the limits and advantages of the newly designed models are discussed in the conclusion where perspectives are drawn.

2. The pannel methods: principles and limitations

Examples of object and vehicle oriented reentry codes are given in [6, 18]. In this section, we only consider the ONERA reentry code **ARES** which serves to illustrate the panel methods

and the limitations that we are looking to improve. **ARES**, which was formerly known as MUSIC/FAST [7], is a spacecraft oriented code divided in four solvers:

- (i) **ATMOS**: which gathers several Terrestrial and Martian atmospheric models.
- (ii) **FAST**: which contains the AED and ATD models.
- (iii) **MODETHEC**: which models the 3D heat transfer inside the materials and their potential thermo-chemical degradation.
- (iv) **MUSIC**: a Guidance Navigation and Control (GNC) flight dynamics code capable of modelling controlled and uncontrolled orbital and reentry flight phases.

The coupling of all these instances enable the fast modelling of a complete space mission and the panel methods which are discussed hereafter are part of the **FAST** code.

2.1. Panel methods main principle

When considering a discretized geometry, each surface element constitutes a panel. For each panel, one can compute its local inclination angle θ as the angle between the outward normal \vec{n} of the element and the freestream velocity \vec{U}_∞ (Figure 1). As done in this work, it is often easier to work with the local incidence angle $\alpha = \pi/2 - \theta$. From the local incidence angle, it is possible to determine whether an element is impinged by the incoming flow or whether it lies in the shadow of the freestream (Figure 2). In case of a complex geometry, some elements may lie in the shadow of other parts and the local angle of incidence alone does not suffice. Sophisticated projection methods can then be used in complements [7].

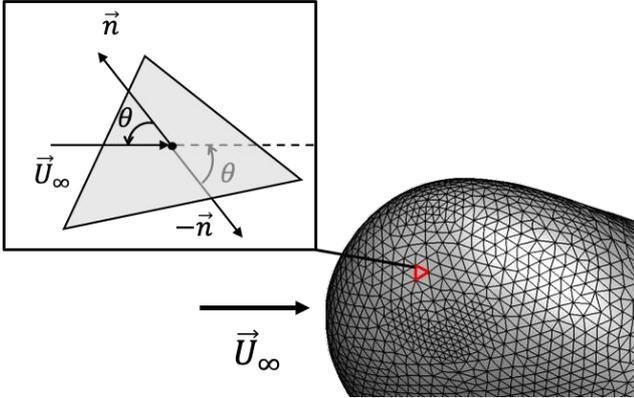


FIGURE 1 – Local inclination angle θ for a given panel.

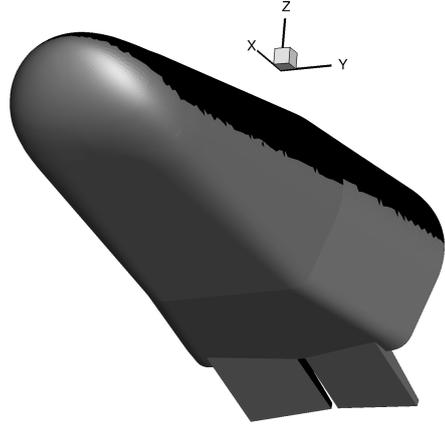


FIGURE 2 – Freestream impinged elements (grey) and shadow elements (black) for the IXV flying with a 47° angle of attack.

In **ARES**, the local curvature radius $R_{n,loc}$ can also be computed at each element and a stagnation detection algorithm automatically identifies the position of the stagnation point according to the flight attitude [7].

Once all panels have been adequately characterized, the panel methods described in the next sections can be applied.

2.2. Transitional regime

As discussed in introduction, a lot of bridging functions were developed during the pre 2000s space exploration period. For all local quantities (C_p , C_f , C_h), **ARES** uses Blanchard's generalized \sin^n bridging functions of expression [12]:

$$\begin{aligned} f_b(Kn) &= \sin^n(\pi[a_1 + a_2 \log(Kn)]), \\ n &= a_3 + a_4 \log(Kn), \end{aligned} \quad (2)$$

where $(a_i)_{i=1..4}$ are parameters of adjustment dependent on the considered local quantity and that were calibrated for a sphere in transitional regime [7]. Hence, with this function and equation (1), any quantity computed at one panel in the continuum and free molecular regime can be expressed at any Kn across the transitional regime.

In the context of bridging functions, the continuum and free molecular limits can be hard to track and the choice of Kn_{cont} and Kn_{fm} is therefore crucial. As indicated in literature, in the case of the Viking I reentry [14], $Kn_{\text{cont}} = 2 \times 10^{-3}$ and $Kn_{\text{fm}} = 25$ were used while sometimes other values such as $Kn_{\text{cont}} = 10^{-3}$, $Kn_{\text{fm}} = 10$ [11] or even $Kn_{\text{fm}} = 100$ [12] were preferred. Hence, this choice is somehow arbitrary which is why the most conservative values ($Kn_{\text{cont}} = 10^{-3}$, $Kn_{\text{fm}} = 100$) are used in **ARES**. Nevertheless, the relevance of the bridging estimation relies on the Knudsen number definition which, according to the selected reference length (L_{ref}), can significantly fluctuate. For instance for the DLR hypersonic transport system Spaceliner [26], the nose radius R_n is about 0.2 m while the total length of the vehicle L is of the order of 60 m. In this case, the unique level of rarefaction resulting from a given altitude can be represented by two radically different Knudsen numbers. In this case, an arbitrary choice for the reference length is left to the code user.

2.3. Free molecular regime

In the absence of chemical surface reactions, surface properties can be analytically determined by balancing the incident and reflected fluxes of normal momentum (pressure), of tangential momentum (friction) and of kinetic plus internal energy (heat flux) [27]. Since the incident fluxes are at equilibrium, the velocity distribution function is a Maxwellian and they can easily be computed. However, the reflected fluxes are not trivial and a gas-surface interaction law is used to express the reflected flux as a function of the incident flux and the reflected flux at thermal equilibrium with the wall. For a Maxwell interaction law of accommodation coefficient w , the following equations are obtained [27]:

$$C_p = \frac{1}{s^2} \left[\left(\frac{2-w}{\sqrt{\pi}} s \sin(\alpha) + \frac{w}{2} \sqrt{\frac{T_w}{T_\infty}} \right) \exp(-s^2 \sin^2(\alpha)) + \left\{ (2-w) \left(s^2 \sin^2(\alpha) + \frac{1}{2} \right) + \frac{w}{2} \sqrt{\frac{\pi T_w}{T_\infty}} s \sin(\alpha) \right\} (1 + \text{erf}(s \sin(\alpha))) \right], \quad (3)$$

$$C_f = \frac{w \cos(\alpha)}{s\sqrt{\pi}} \left[\exp(-s^2 \sin^2(\alpha)) + \sqrt{\pi} s \sin(\alpha) \{1 + \text{erf}(s \sin(\alpha))\} \right], \quad (4)$$

$$q = w \frac{\rho_\infty}{4\sqrt{\pi}} \left(\frac{2kT_\infty}{m} \right)^{3/2} \left(\left[s^2 + \frac{\gamma}{\gamma-1} - \left\{ \frac{1}{2} \frac{\gamma+1}{\gamma-1} \right\} \left(\frac{T_w}{T_\infty} \right) \right] \left[\exp(-s^2 \sin^2(\alpha)) + \sqrt{\pi} s \sin(\alpha) \{1 + \text{erf}(s \sin(\alpha))\} \right] - \frac{1}{2} \exp(-s^2 \sin^2(\alpha)) \right), \quad (5)$$

where T_w and T_∞ respectively denote the wall and freestream temperatures, m and γ are the mass and adiabatic coefficient of the gas, ρ_∞ is the freestream density, s is the freestream speed ratio:

$$s = U_\infty \sqrt{\frac{m}{2kT_\infty}}, \quad (6)$$

with k the Boltzmann constant and the $\text{erf}(\cdot)$ function is given by:

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) dt. \quad (7)$$

In free molecular conditions ($Kn \gg 1$), the wall quantities obtained by DSMC with the same accommodation coefficient w exactly converge towards these analytical solutions.

2.4. Continuum regime

In continuum regime, the friction contribution to the aerodynamic forces is negligible in comparison to the pressure contribution which is why C_f is simply neglected. For the pressure and heat flux distributions, a two-step approach is employed: the stagnation value is computed first and the distribution is computed afterwards.

As explained in [7], in **ARES**, the pressure stagnation value $C_{p,\text{stag}}$ is computed from an iterative procedure based on the Mollier table and the Rankine-Hugoniot equations. Then, the distribution of C_p is obtained with the modified Newton method:

$$C_p = C_{p,\text{stag}} \sin^2(\alpha), \quad (8)$$

when the panel is directly impinged by the flow and $C_p = 0$ (*i.e.* $p = p_\infty$) when the panel is in the shadow (Figure 2).

For the stagnation heat flux, many correlation functions exist [18, 28] and most of them are available in **ARES** [7]. Here, only the Johnston & Brandis formulation [29] is presented:

$$q_{\text{stag}} = 7.455 \times 10^{-9} \rho_\infty^{0.4705} U_\infty^{3.089} R_n^{-0.52}, \quad (9)$$

where q_{stag} has units of W/cm^2 , ρ_∞ is expressed in kg/m^3 , U_∞ in m/s and R_n in m . This correlation was derived from around 340 CFD simulations with velocities comprised between 3 and 9.5 km/s under a super-catalytic wall assumption. Still in **ARES**, the heat-flux distribution is then obtained for any panel with Verant-Lefrançois formula:

$$q = q_{\text{stag}} \left(\frac{p}{p_{\text{stag}}} \right)^a \left(\frac{R_n}{R_{n,\text{loc}}} \right)^b, \quad (10)$$

where the parameters a and b were obtained from 3D simulations of the IXV in continuum regime and are given in [7]. Other simple formulations exist for the distribution like that in **SCARAB** [6]:

$$q = q_{\text{stag}} (0.1 + 0.9 \sin(\alpha)), \quad (11)$$

or that of Murzinov [30] that was recently used by Singh and Schwartzentruber [31]:

$$q = q_{\text{stag}} (0.55 - 0.45 \cos(2\alpha)). \quad (12)$$

These formulations are however very simplistic and cannot adequately represent complex distributions for non-spherical shapes.

In case of a significant error in the wall quantities in continuum regime, it is fatally propagated by the bridging function across the lower part of the transitional regime. For this reason and those mentioned in section 2.2, ML techniques were investigated. Since, the two-step approach presented in this section is particularly convenient, a similar procedure was adopted.

3. Machine learning based reduced models

Let \mathbf{x} be a vector of design variables (inputs) and y the corresponding response (output) obtained with a high fidelity code. The functional relationship between the inputs and output is then:

$$y = f(\mathbf{x}). \quad (13)$$

The surrogate model of the code provides the approximation \hat{y} that is related to the inputs by the relation :

$$\hat{y} = \hat{f}(\mathbf{x}). \quad (14)$$

and to the output by the relation :

$$y = \hat{y} + \epsilon, \quad (15)$$

where ϵ is the error due to both approximation and measurement [19]. The objective of metamodeling is then to build a collection of n sampling points $\mathbf{x}^{(i)}$ and responses $y^{(i)}$ forming the plan of experiments. This plan is finally used to train the metamodel to learn the relationship $\hat{f}(\cdot)$ between the inputs and outputs.

3.1. Response surface model

Response surface models (RSM) are simple surrogates which consist of a polynomial regression. Such models assume that the error ϵ is normally distributed with mean zero and standard deviation σ [19]. The approximation is then defined by the order of the polynomial. In most cases, RSM are linear (first order):

$$\hat{y} = \beta_0 + \sum_{i=1}^d \beta_i x_i, \quad (16)$$

or quadratic (second order):

$$\hat{y} = \beta_0 + \sum_{i=1}^d \beta_i x_i + \sum_{i=1}^d \beta_{ii} x_i^2 + \sum_{i=1}^d \sum_{j \geq i}^d \beta_{ij} x_i x_j. \quad (17)$$

The parameter d is the number of design variables and the regression parameters β_i and β_{ij} are obtained by least squares regression by fitting the data $(\mathbf{x}^{(i)}, y^{(i)})$ [19, 21].

RSM models constitute a good way of identifying whether a simple polynomial approximation is sufficient or if a more sophisticated learning method is required. In this work, it was used as a comparative tool for the kriging method described in next section.

3.2. Kriging

Kriging is an interpolation method expressed as:

$$\hat{y}(\mathbf{x}) = f(\mathbf{x}) + z(\mathbf{x}), \quad (18)$$

where $f(\cdot)$ is a low-order polynomial which, in the case of ordinary kriging, as considered here, reduces to a constant β_0 to be determined. The $z(\cdot)$ function is a Gaussian stochastic function which represents the realization of a random process with zero mean and variance σ^2 . Kriging is then simply a mean with a random field given by:

$$\hat{y}(\mathbf{x}) = \beta_0 + z(\mathbf{x}), \quad (19)$$

where the stochastic process $z(\cdot)$ has covariance:

$$Cov(z) = \sigma^2 \mathbf{R}(x^i, x^j), \quad (20)$$

with $\mathbf{R}(x^i, x^j)$ the correlation matrix of size $n \times n$ and n the number of sample points. The correlation matrix is generally written:

$$\mathbf{R}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \exp \left(\sum_{l=1}^d \theta_l |x_l^{(i)} - x_l^{(j)}|^p \right). \quad (21)$$

A full interpretation of the weight factors θ and of the parameter p is available in [20]. In practice, large values of θ_l indicate a strong correlation, *i.e.* sensitivity to the l^{th} design parameter and p

reflects the smoothness of the approximation [21]. Although p can be adjusted according to the data, it is often taken as an integer. Hence, when $p = 1$, the correlation function \mathbf{R} (also called *kernel*) is *absolute exponential* and when $p = 2$, the kernel is *Gaussian*. The parameters σ^2 , β_0 and θ are then computed as maximum likelihood estimates (MLE) [20, 21]:

$$\tilde{\beta}_0 = (\mathbf{1}^t \mathbf{R}^{-1} \mathbf{1})^{-1} \mathbf{1}^t \mathbf{R}^{-1} \mathbf{y}, \quad (22)$$

$$\tilde{\sigma}^2 = \frac{1}{n} (\mathbf{y} - \mathbf{1} \tilde{\beta}_0)^t \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1} \tilde{\beta}_0), \quad (23)$$

with $\mathbf{1}$ a $1 \times n$ vector of ones. The optimal weight vector $\tilde{\theta}$ is the one that maximizes the reduced likelihood:

$$\frac{n}{2} \ln(\tilde{\sigma}^2) - \frac{1}{2} \ln(\mathbf{R}). \quad (24)$$

It is usually determined thanks to an optimization procedure. Finally, the kriging model, which consists in the optimal response at a new point \mathbf{x}^* , is given by the mean of the Gaussian process conditioned by the vector of responses \mathbf{y} and writes [20]:

$$\hat{y}(\mathbf{x}^*) = \tilde{\beta}_0 + \mathbf{r}(\mathbf{x}^*)^t \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1} \tilde{\beta}_0), \quad (25)$$

where $\mathbf{r}(\mathbf{x}^*)$ is the correlation vector between \mathbf{x}^* and all sampling points. The variance of this Gaussian process is given by:

$$\hat{\sigma}(\mathbf{x}^*) = \tilde{\sigma}^2 (1 - \mathbf{r}(\mathbf{x}^*)^t \mathbf{R}^{-1} \mathbf{r}(\mathbf{x}^*)), \quad (26)$$

and is a way of quantifying the confidence in the kriging model.

When dealing with noisy data, one might prefer a regressing kriging instead of an interpolating kriging. Such model can easily be obtained by introducing a constant λ in equations (25)-(28) [20]:

$$\hat{y}(\mathbf{x}^*) = \tilde{\beta}_0 + \mathbf{r}(\mathbf{x}^*)^t (\mathbf{R}^{-1} + \tilde{\lambda}) (\mathbf{y} - \mathbf{1} \tilde{\beta}_0), \quad (27)$$

$$\hat{\sigma}(\mathbf{x}^*) = \tilde{\sigma}^2 \left(1 - \mathbf{r}(\mathbf{x}^*)^t (\mathbf{R}^{-1} + \tilde{\lambda}) \mathbf{r}(\mathbf{x}^*) \right), \quad (28)$$

where $\tilde{\lambda}$ is the MLE of λ that can be determined via an optimization procedure analogous to the one described earlier.

Because of the stochastic nature of the DSMC, the numerically derived quantities are noisy and the regressing kriging was therefore employed for the computation of the stagnation pressure and heat flux coefficients.

3.3. Artificial Neural Networks

Artificial Neural Network (ANN) are part of Deep Learning (DL), a subcategory of ML in which the depth notion makes reference to the successive layers of representations usually learnt by neural networks [32]. A short description of ANN is provided in [19] but the fast changes in the field make it outdated so newer references such as Chollet's book [32] are more relevant for an accurate and exhaustive description of DL methods.

In the context of this work, the focus is given on a specific class of neural networks called *multilayer perceptron*. Such networks are composed of *fully connected* (or *dense*) layers and, in the case of a scalar regressing network, of a single output (Figure 3). The complexity of such network is characterized by its number of hidden layers L and their respective numbers of neurons n_l . In the case of a *feedforward* network, the information travels directly from the input $\mathbf{x} = (x_1, \dots, x_d)$

to the outputs \hat{y} . For each neurone $k \in \llbracket 1, n_1 \rrbracket$ of the first layer denoted (1), the inputs are reduced and activated as follows:

$$\begin{aligned} z_k^{(1)} &= \sum_{i=1}^d w_{k,i}^{(1)} x_i + b_k^{(1)}, \\ s_k^{(1)} &= \sigma^{(1)}(z_k^{(1)}), \end{aligned} \quad (29)$$

where $w_{k,i}$ and b_k respectively denote the weights and the bias of the k^{th} neurone while $\sigma^{(1)}(\cdot)$ is the activation function of the first layer. Similarly, for the l^{th} layer and its neurone $k \in \llbracket 1, n_l \rrbracket$, equation (31) becomes:

$$\begin{aligned} z_k^{(l)} &= \sum_{i=1}^{n_{l-1}} w_{k,i}^{(l)} s_i^{(l-1)} + b_k^{(l)}, \\ s_k^{(l)} &= \sigma^{(l)}(z_k^{(l)}). \end{aligned} \quad (30)$$

The single output is finally given by the following equations:

$$\begin{aligned} z^{(L+1)} &= \sum_{i=1}^{n_L} w_i^{(L+1)} s_i^{(L)} + b^{(L+1)}, \\ \hat{y} &= s^{(L+1)} = \sigma^{(L+1)}(z^{(L+1)}). \end{aligned} \quad (31)$$

In practice, the same activation functions (*e.g.* sigmoid, ReLU, tanh) are used for all layers except the last one where the identity function is preferred in case of scalar regression [32]. For each input, the corresponding error E is then defined as:

$$E = \frac{1}{2}(y - \hat{y})^2, \quad (32)$$

which is minimized by adjusting the weights and biases of the network. Usually, the error is computed for a set of inputs called *batch* and for each network parameter, their contribution to the error is estimated through the derivatives of the error $\left(\frac{\partial E}{\partial w_{k,i}^{(l)}}, \frac{\partial E}{\partial b_k^{(l)}} \right)$ for k, i and l between 1 and n_l , 1 and n_{l-1} , and 1 and $L + 1$, respectively. The parameters are then updated after each passing of a batch through the network according to the following relations:

$$\begin{aligned} w_{k,i}^{(l)} &= w_{k,i}^{(l)} - \lambda \frac{\partial E}{\partial w_{k,i}^{(l)}}, \\ b_k^{(l)} &= b_k^{(l)} - \lambda \frac{\partial E}{\partial b_k^{(l)}}, \end{aligned} \quad (33)$$

where $\lambda \in \mathbb{R}_+^*$ is the *learning rate* (or *step*). Because the derivative computation can be tricky, they are obtained with the *chain rule* which exploits a fundamental property of multivariate functions: $\frac{\partial y}{\partial x} = \frac{\partial y}{\partial u} \frac{\partial u}{\partial x}$. This way, the *backpropagation* algorithm consists of calculating the derivative of the error with respect to the output and to retropropagate it through the network neuron after neuron and layer after layer. Finally when the batches are selected randomly, this parameters adjustment procedure is called *stochastic gradient descent* (SGD). Optimizers can be used in order to make the learning process more efficient and regularization techniques reduce the chances of *overfitting* to the data [32].

As opposed to the network's parameters (*i.e.* the values of the weights and biases), the *hyperparameters* refer to all the other adjustable parameters including the number of layers, the number of neurons in each of them, the learning rate or the batch size. The determination of an optimal architecture (*i.e.* set of hyperparameters) can be reached by trial and error or by hyperparameter tuning [24, 32]. When dealing with small networks like in this case, the former

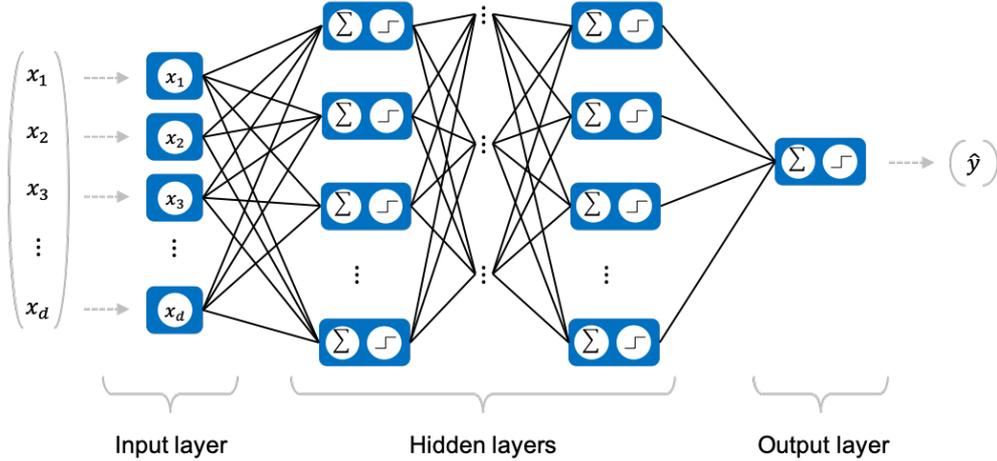


FIGURE 3 – Graphical representation of a multilayer perceptron with a single output (inspired from [33]).

method is usually sufficient and was therefore employed.

ANN is a learning method capable of dealing with huge amount of data and robust to *outliers*. It was therefore chosen as a regressing technique for the pressure, friction and heat flux distribution computation.

4. High-fidelity fluid solvers

Now that the ML methods have been described, the next sections focus on the two high fidelity methods that were used to generate the data necessary to train and evaluate them.

4.1. Direct Simulation Monte-Carlo with **SPARTA**

In rarefied conditions encountered at high altitude and characterized by low densities, the rate of intermolecular collision is not high enough to maintain a state of equilibrium. A significant deviation from such state leads to the failure of the Navier-Stokes equations and gas-kinetic methods must be preferred. In this context, Bird’s DSMC [27] is the most popular method for the simulation of hypersonic rarefied flows. This stochastic algorithm provides a numerical solution of the Boltzmann equation for any Knudsen number. More details about DSMC and its current issues are available in [27, 34, 35, 36] and its validation by comparison to experimental data was investigated in [37, 38].

In order to guarantee the physical relevance of the results, several criteria must be met when using DSMC. Hence, to prevent particles to interact with particles farther than a mean free path away, the cell size (Δx) must be kept smaller than a fraction of the local mean free path (λ). In the same way, to make sure that particles do not move over distances greater than a mean free path, the time step (Δt) must be kept smaller than the local mean collision time (τ_{coll}). Finally, the statistical convergence requires that a certain number of particles (usually of the order of ten) is contained in each cell. For all the DSMC computations performed with **SPARTA** [25] in the frame of this work, a particular care was taken to satisfy these requirements. Since the limiting and most critical criteria is generally the grid refinement, **SPARTA** adaptive refinement procedure was used. This binary refinement procedure illustrated in [25, 38] enables to locally refine the mesh where Δx becomes greater than λ .

Unless specified otherwise, **SPARTA**’s following physical models were used. Since we are only concerned with orbital reentry velocities ($U_\infty < 10$ km/s), the ionization and radiative effects were neglected and a five species air model was used. The chemical reactions were modelled

with Bird’s TCE model [27] and Park’s kinetic rate of reactions. The rotational mode of energy was assumed continuous with a constant number of collision ($Z_{\text{rot}} = 5$) while the vibrational mode was assumed discrete with a variable number of collision given by Millikan-White model ($Z_{\text{vib}} = Z_{\text{vib}}^{\text{MW}}(T)$) [27, 34]. The translational-vibrational and translational-rotational energetic transfers were simulated with the Borgnakke-Larsen algorithm and the No Time Counter (NTC) selection procedure [27, 34]. Following the findings of a recent paper [38], the Variable Hard Sphere (VHS) collision model was used with Stephani’s parameters [39]. Finally, the gas-surface chemical reactions were not considered and the vehicles wall was assumed fully diffuse with a constant temperature.

4.2. Computational Fluid Dynamics with **CASL**

In near continuum conditions ($Kn < 10^{-2}$), the computational cost of the DSMC computation is generally prohibitive. For such conditions, the stagnation pressure and heat flux were then obtained with the CFD code **CASL**.

CASL is the 1D spheric equivalent of the CFD code **CELHyO** [40, 41]. It was recently used by Van Ghele for the study of the Intermediate eXperimental Vehicle (IXV) in continuum regime [42]. The particularity of **CASL** is that it is capable of precisely computing the heat flux and pressure at the stagnation point of any geometry assimilated to the equivalent sphere of nose radius R_n equals to the stagnation curvature radius. For the present computations, the Navier-Stokes equations were solved with an implicit time step and a second order HUS-Van-Leer scheme in space.

In accordance to the DSMC computations, no surface reactions was considered and a five species air model was used. Because of the negligible influence of simulating the vibrational mode (effect smaller than 5% on the stagnation heat flux), the flow was assumed in thermal equilibrium and Park’s kinetic rate of reaction was used to simulate the chemical nonequilibrium. The species viscosity was computed with Blottner’s model and the mixture viscosity was modeled with Armaly-Sutton’s formula. The diffusion was obtained with Gubernatis formula and the thermal conductivity with Wilke’s and Montchick’s [43].

5. Reduced models generation

As mentioned in the introduction, two categories of surrogate models were developed in this work. The first category computes the pressure and heat flux coefficients at the stagnation point while the second locally compute the pressure, heat-flux and friction coefficient value at any location of the geometry. For each kind of model, this section presents the selected variables of design and the structure of the DOE employed for the training and validation.

5.1. Surrogate models for the stagnation pressure and heat flux coefficients

5.1.1. Design variables

In order to build accurate surrogate models for the stagnation pressure and heat flux coefficients, the first question is to determine what minimal amount of geometric and atmospheric quantities is necessary. In hypersonic rarefied conditions, the most relevant non-dimensional numbers are the Knudsen and Mach numbers. In addition, the literature and the engineering models introduced in section 2 show that in continuum regime, the stagnation heat flux can be computed from contextual quantities (*e.g.* ρ_∞ , U_∞ , R_n). Similarly, in free molecular regime, the expression of the stagnation heat flux can be derived from equation (5) with $\alpha = 0^\circ$. Under the hypersonic and fully diffuse wall assumptions (*i.e.* $s \gg 1$ and $w = 1$), it only depends on U_∞ , $R_{s,\text{air}}$ and T_∞ . For the pressure coefficient, the same observation holds and in the transitional regime, only the Knudsen number is needed to propagate the values of the continuum and free molecular regimes.

Since the Knudsen and the Mach numbers encompass all the environmental values and according to the ML principle which recommends to avoid redundant information among the design variables, these two numbers appear as the most natural candidates.

Finally, by noticing that the reference length L_{ref} used in the Knudsen number calculation depends on the context, the following choice $L_{\text{ref}} = 2R_n$ (with R_n the stagnation point curvature) allows for the most relevant geometric quantity to be included in the design variables. The surrogates models for both coefficients have the following form:

$$\hat{C}_{h/p,\text{stag}} = \hat{f}(Kn, M_\infty) = \hat{f}\left(\frac{\lambda_{N_2,\infty}^{\text{VHS}}}{2R_n}, \frac{U_\infty}{\sqrt{\gamma R_{\text{sair}} T_\infty}}\right). \quad (34)$$

In order to avoid any ambiguity about the value of R_n and to reduce the computational cost of the simulations, the database described in next section was built with a unit sphere ($R_n = 1$ m) in hypersonic rarefied regime.

5.1.2. Design of experiments

Once the design variables are selected, the next step consists in generating the appropriate numerical database to explore the plan of experiments. In order to create a DOE (Kn, M_∞) corresponding to rarefied reentry conditions, the exploration domain must first be limited in terms of objects size and atmospheric conditions. Considering the frame of this study, the following conditions were chosen:

- The objects size varies between 0.1 and 50 m.
- The altitudes are comprised between 200 and 85 km.
- The reentry velocity goes from 6 to 10 km/s.

By using the **US62** ($h < 90$ km) and **Jacchia77** ($h \geq 90$ km) atmospheric models [44, 45], the associated numeric density n_∞ and atmospheric temperature T_∞ give the following conditions:

- $M_\infty \in [8; 14]$, $Kn \in [4; 1980]$ at 200 km.
- $M_\infty \in [22; 37]$, $Kn \in [1.4 \times 10^{-4}; 7 \times 10^{-2}]$ at 200 km.

These limits define the *physical domain* illustrated in grey on Figure 4. Because the temperature can vary at a fixed altitude and considering the limiting Knudsen number in continuum regime ($Kn \simeq 10^{-3}$) and in free molecular regime ($Kn \simeq 100$), the following domain is obtained:

- For $Kn = 100$, $M_\infty \in [8; 20]$.
- For $Kn = 10^{-3}$, $M_\infty \in [20; 40]$.

However, because of the non-linear variation of T_∞ between the considered altitudes, the domain of experiments simply limited by these boundaries is not relevant as is. This aspect is illustrated in Figure 4 where several limit points obtained at altitudes between 200 and 85 km are plotted in black. The domain of experiments represented in red dashed line on the same Figure was then considered instead.

A 25 points latin hypercube sampling (LHS) plan was then constructed inside this domain. Because the physical domain is not canonical, a logarithmic bilinear transformation was built to go from one reference frame to another. The obtained DOE in the canonical and physical reference frames is represented in Figures 5 and 6.

Because $(Kn, M_\infty) \mapsto (U_\infty, T_\infty, n_\infty, X_{O_2}, X_{N_2}, X_O)$ is a non-bijective application, the atmospheric and flight conditions were obtained with the following procedure:

- (i) The temperature T_∞ is assumed constant and equal to 160 K.
- (ii) The geometries are all spheres of unit radius ($R_n = 1$ m).
- (iii) The mean free path $\lambda_{N_2,\infty}^{\text{VHS}}$ and the numeric density n_∞ are obtained from (Kn, R_n, T_∞) .
- (iv) Knowing n_∞ , the compositions (X_{O_2}, X_{N_2}, X_O) are interpolated from the **Jacchia77** atmospheric table.

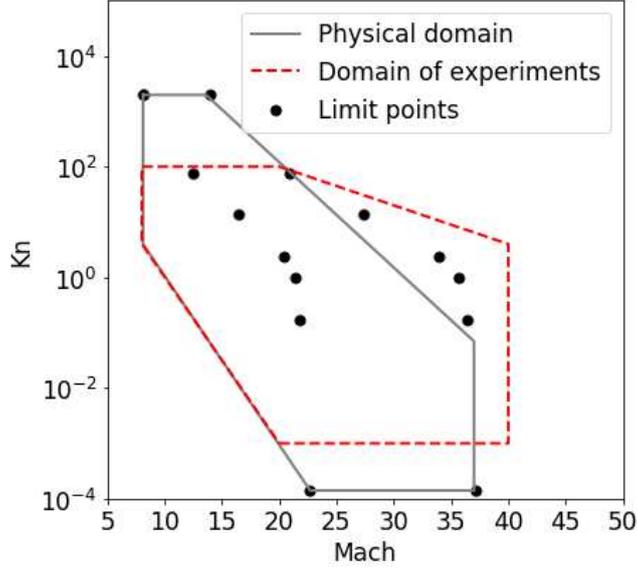


FIGURE 4 – Physical and experiments domains.

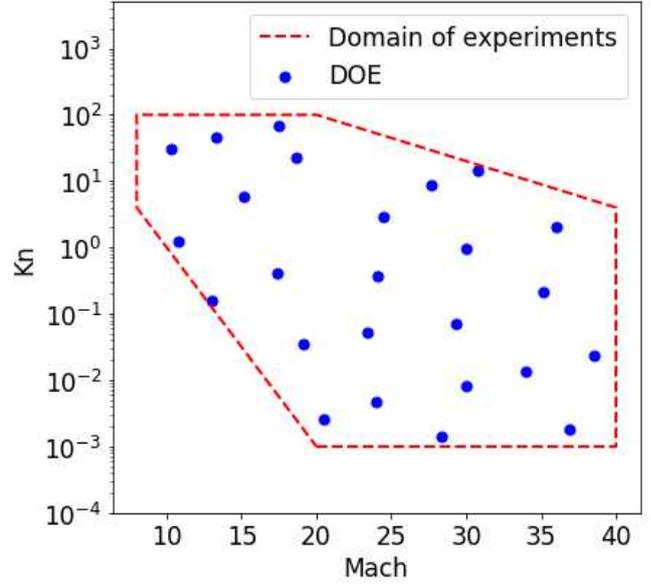
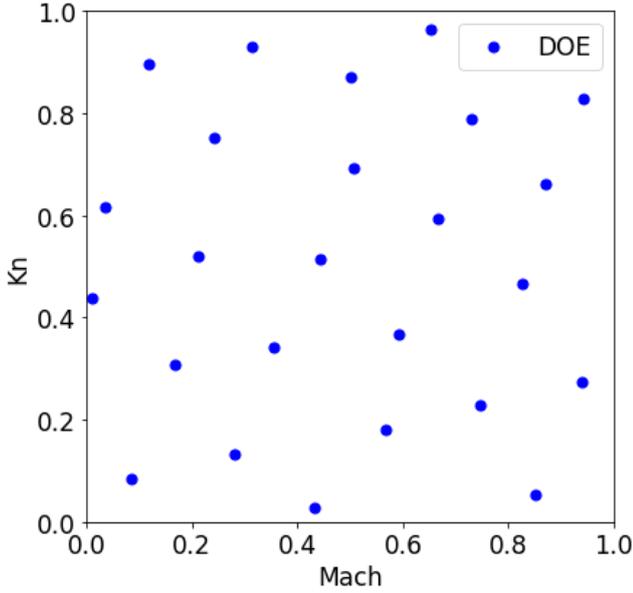


FIGURE 5 – LHS DOE in the canonical reference frame. FIGURE 6 – LHS DOE in the physical reference frame.

- (v) The specific gas constant $R_{s,\text{air}}$ is then computed and its combination with T_∞ and M_∞ gives U_∞ .
- (vi) The wall temperature T_w is finally estimated from radiative equilibrium according to Moss method [46], with a constant emissivity $\epsilon_w = 0.85$.

The conditions of the 25 training points resulting from this method are given in Table 1. For the most rarefied conditions ($Kn \geq 10^{-2}$), 2D-axisymmetric DSMC computations were performed with **SPARTA**. As discussed in section 4.1, a particular care was taken to ensure the respect of the convergence metrics. Hence, for the most rarefied simulation ($Kn = 67$), the time step, grid size and number of particles per cell along the stagnation line were chosen as follows: $\Delta t = 5 \times 10^{-4} \tau_{\text{coll},\infty}$, $\Delta x = 0.005 \lambda_\infty$ and $N_{p/c} > 10$. For the case $Kn = 1.4 \times 10^{-2}$, the time step and grid size were taken $\Delta t = \min(\tau_{\text{coll}})/5$, $\Delta x = \lambda_\infty/4$ and an automatic refinement procedure was set up to reduce Δx up to 8 times where $\lambda/\Delta x \leq 1.5$. Examples of results are shown in Figures 7 and 8. For the near-continuum conditions ($10^{-3} < Kn < 10^{-2}$), the simulations were

performed with **CASL** in the numerical conditions described in section 4.2.

M_∞	Kn	n_∞ (\cdot/m^3)	X_{O_2}	X_{N_2}	X_O	U_∞ (m/s)	T_w (K)
CFD							
28.4	1.4×10^{-3}	4.69×10^{20}	0.221	0.779	0	7214	1597
36.9	1.8×10^{-3}	3.59×10^{20}	0.221	0.779	0	9383	1881
20.4	2.6×10^{-3}	2.49×10^{20}	0.221	0.779	0	5199	1154
24	4.7×10^{-3}	1.40×10^{20}	0.221	0.779	0	6108	1212
30	8.1×10^{-3}	8.08×10^{19}	0.221	0.779	0	7636	1337
DSMC							
33.9	1.4×10^{-2}	4.76×10^{19}	0.233	0.767	0.009	8643	1384
38.5	2.4×10^{-2}	2.75×10^{19}	0.196	0.786	0.018	9830	1411
19.01	3.4×10^{-2}	1.91×10^{19}	0.186	0.785	0.029	4890	798
23.4	5.1×10^{-2}	1.27×10^{19}	0.178	0.784	0.038	6017	885
29.3	6.9×10^{-2}	9.45×10^{18}	0.170	0.784	0.0460	7547	1011
13.0	1.6×10^{-1}	4.18×10^{18}	0.147	0.779	0.074	3371	499
35.2	2.2×10^{-1}	3.00×10^{18}	0.133	0.774	0.093	9161	1009
24.1	3.8×10^{-1}	1.72×10^{18}	0.114	0.765	0.121	6324	672
17.3	4.1×10^{-1}	1.61×10^{18}	0.112	0.763	0.125	4559	517
30.0	9.5×10^{-1}	6.91×10^{17}	0.089	0.741	0.170	7993	633
10.8	1.2	5.38×10^{17}	0.085	0.733	0.182	2886	284
36.0	2.0	3.22×10^{17}	0.077	0.714	0.209	9675	601
24.4	2.9	2.25×10^{17}	0.072	0.697	0.231	6601	413
15.2	5.8	1.12×10^{17}	0.064	0.661	0.275	4141	253
27.7	8.9	7.33×10^{16}	0.058	0.634	0.308	7634	347
30.8	14.1	4.63×10^{16}	0.052	0.601	0.347	8561	336
18.6	22.8	2.87×10^{16}	0.046	0.561	0.393	5240	220
10.3	30.9	2.12×10^{16}	0.042	0.538	0.420	2913	171
13.3	45.6	1.43×10^{16}	0.036	0.491	0.473	3824	176
17.4	67.0	9.75×10^{15}	0.032	0.460	0.508	5071	184

Conditions obtained with $T_\infty = 160$ K, $Kn = \lambda_{N_2, \infty}^{VHS} / (2R_n)$ and $R_n = 1$ m.

TABLE 1 – Atmospheric conditions of the flight points used for the training of the kriging models.

For the validation, Singh and Schwartzenruher work [31] provided 29 points inside our plan of experiments for the validation of the $C_{h,stag}$ prediction model. In addition, 23 validation points for both the $C_{p,stag}$ and $C_{h,stag}$ prediction models were extracted from the DSMC 3D database (denoted DB 3D) that was used for the training of the neural network models (see section 5.2.2). For each of these simulations performed with 3D geometries (Apollo, OREX and IXV), the corresponding Kn was computed from the stagnation curvature R_n . These validation points are represented in Figure 9.

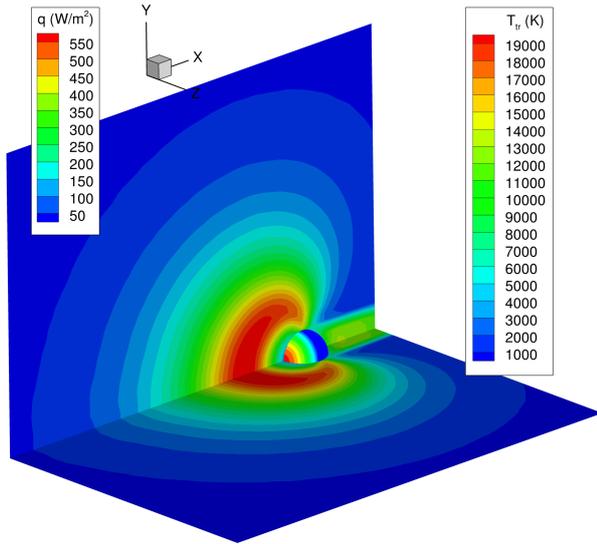


FIGURE 7 – Temperature field and wall heat flux for the DSMC simulation of the $M_\infty = 30.8$, $Kn = 14.1$ case.

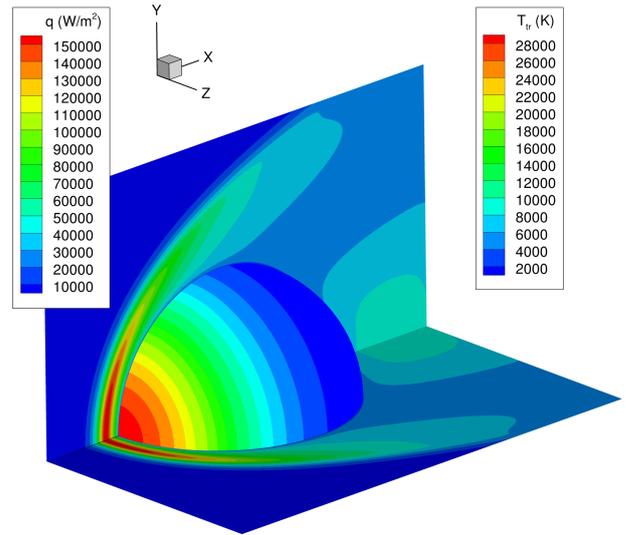


FIGURE 8 – Temperature field and wall heat flux for the DSMC Simulation of the $M_\infty = 33.9$, $Kn = 1.4 \times 10^{-2}$ case.

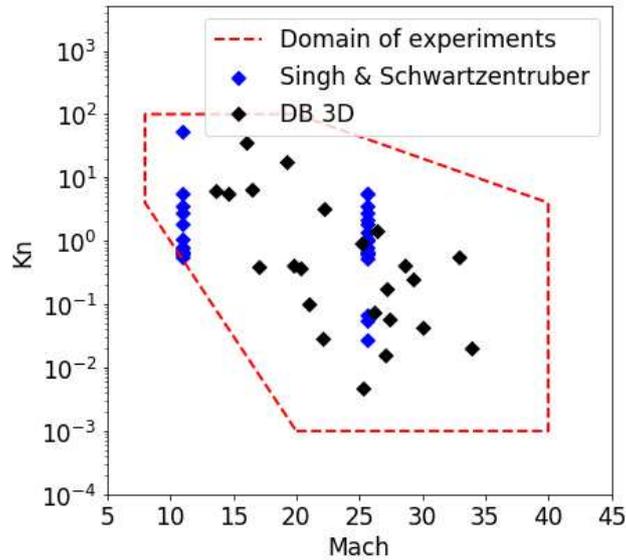


FIGURE 9 – Validation points.

5.2. Reduced models for the distributions of pressure, friction and heat flux coefficients

Since most reentry vehicles can be divided in three main categories (see Table 2), this section presents a way to build reduced models as generic as possible by considering only three geometries belonging to each of these categories.

Types of geometry	Examples of vehicles
Sphere-cone	OREX, Stardust, Ram-C, Pathfinder
Transportation vehicle	IXV, Space Shuttle, Spaceliner
Reentry probe	Apollo, ARD, Orion, Crew Dragon

TABLE 2 – Reentry vehicle typology.

5.2.1. Design variables

In complement to the distribution formulas presented in section 2, the analysis of atmospheric reentries (Apollo 6, Apollo AS-202, OREX and the IXV) performed in the frame of previous studies [37, 38] enabled the identification and confirmation of the most relevant design variables that are to be accounted for in our new models. This analysis is presented below.

For the pressure coefficient, the modified Newton method indicates that only $C_{p,\text{stag}}$ and α are required to compute the C_p distribution in the continuum regime. In the free molecular regime, the pressure depends on some atmospheric variables ($T_\infty, \rho_\infty, U_\infty$), on the accommodation coefficient w and on the local incidence angle α . For both formulations, the stagnation value $C_{p,\text{stag}}$ corresponds to the elements where $\alpha = 90^\circ$ which is why these two quantities must be included in the new model. As for the accommodation coefficient, although its impact in highly rarefied conditions is significant, gas-surface interaction are assumed fully diffuse in this work and this parameter was not considered. For the investigated vehicles (Apollo, OREX, IXV), the corresponding reentries in rarefied conditions did not evidence the need to include any other parameter and a model of the form $\hat{C}_p = \hat{f}(C_{p,\text{stag}}, \alpha)$ was considered.

For the heat flux coefficient in continuum regime, the distribution formulations introduced in section 2.4 all depend on the stagnation coefficient $C_{h,\text{stag}}$ and on the local incidence angle α . The most complete formulation of Vérant-Lefrançois also depends on the local curvature via the ratio $R_n/R_{n,\text{loc}}$. In free molecular regime, similarly to the pressure distribution, the distribution depends on the local incidence angle, the accommodation coefficient and some atmospheric variables. For the same reasons, the design variables must include α and $C_{h,\text{stag}}$ while w was not accounted for. In addition, the observation of the heat flux distribution at various altitudes along the IXV windward centerline [38] shows how the level of rarefaction changes its behaviour. Indeed, for $Kn \leq 10^{-1}$, the heat flux decreases along the windward as the distance to the stagnation point (here denoted d_{stag}) increases while this effect is not visible in highly rarefied conditions. Another rarefaction effect is observed at the vehicle's shoulder where the peak of heat flux strongly rises when Kn diminishes. Assuming that the rarefaction level can be described by $C_{h,\text{stag}}$, only the parameters $R_n/R_{n,\text{loc}}$ and d_{stag} were included to the design variables. In order to ensure the model's stability even for particular cases where $R_n/R_{n,\text{loc}} \rightarrow \infty$, the following limiter was implemented:

$$\left[\frac{R_n}{R_{n,\text{loc}}} > 10 \right] \Rightarrow \left[\frac{R_n}{R_{n,\text{loc}}} = 10 \right]. \quad (35)$$

Finally, a model of the form $\hat{C}_h = \hat{f}(C_{h,\text{stag}}, \alpha, d_{\text{stag}}, R_n/R_{n,\text{loc}})$ was investigated.

The friction coefficient is, a priori, a more complex quantity to deal with. Indeed, in continuum regime, no generic formulation exists. In free molecular regime, an analytical formulation analogous to that of the pressure and heat flux was introduced in section 2.3. It depends on α and on some atmospheric quantities. However, in this case, the stagnation value is zero and the same approach as that used for C_p and C_h cannot be transposed. In addition, the observation of the friction distribution along the IXV windward centerline [38] suggests that the rarefaction level and $R_n/R_{n,\text{loc}}$ influence C_f while the role of d_{stag} is not evident from these results. A model of the form $\hat{C}_f = \hat{f}(C_{h,\text{stag}}, \alpha, R_n/R_{n,\text{loc}})$ was then investigated.

At this stage, one notices that none of the considered model explicitly includes Kn and that the rarefaction level is assumed to be measured by the stagnation values. Furthermore, the three distribution models are scalar regressing models and any regressing surrogate could have been employed instead. Nevertheless, the reduced model design phase evidenced issues related to the

structure and size of the training database; this is why neural networks emerged as the most suitable candidate.

5.2.2. Design of experiments

As opposed to the kriging models introduced in section 5.1 where the design variables (Kn , M_∞) are global, the input parameters for the C_p , C_h , and C_f distributions are local. An immediate consequence of this difference is that instead of having one sampling point per simulation, each result provides as many sampling points as there are of surface elements on the considered geometry. Indeed, each element of the surface mesh has its own set of values (α , d_{stag} , $R_n/R_{n,\text{loc}}$). However, since each model depends on a stagnation quantity ($C_{p,\text{stag}}$, $C_{h,\text{stag}}$) and that each simulation only provides one value for each of them, the exploration of the plan of experiments is much less extensive in this dimension.

In such conditions and under the constraint of a reasonable number of simulations, a 27 computation DOE was built with the objective of homogeneously exploring the design space. Hence, because varying thoroughly each geometric variable across its interval of variation ($\alpha \in [-90^\circ; 90^\circ]$, $R_n/R_{n,\text{loc}} \in [0; 10]$, $d_{\text{stag}} \in [0; 4]$ m) is hard with a fixed set of only 3 geometries (Apollo, OREX, IXV), the database was built in order to explore the design space at best and at minimal cost. The 3D database was therefore computed from a pseudo-random sampling¹ of a vehicle, an altitude ($h \in [95; 150]$ km), a velocity ($U_\infty \in [6; 10]$ km/s) and an angle of attack (AoA $\in [0^\circ; 180^\circ]$). For the three considered vehicles, this altitude interval offers a good coverage of the transitional regime while enabling the respect of the DSMC convergence metrics. Concerning the velocity interval, this choice is consistent with the range of orbital reentry velocities for which ionization effects are negligible. Finally, for the angles of attack, although values greater than 45° are not relevant from a trajectory point of view, this choice allows us to take advantage of the geometric diversity of the vehicles afterbody hence providing a better exploration of the design space.

The corresponding atmospheric data were then extracted from the **Jacchia77** model. In addition to these 27 simulations, 3 computations obtained in the frame of previous studies [37, 38] were appended forming a training database of 30 simulations. This database is presented in Table 3 and illustrated in Figure 10. In the table, the non-indicated Kn correspond to special cases a flat face faces the upstream flow and thus for which no equivalent stagnation curvature is known. In accordance with Cholet’s recommendations [32], a training-validation-test procedure was used. Hence two complementary databases composed of respectively 5 and 8 computations were designed to validate and test the trained models. The corresponding simulations are listed in Table 4. The validation database was elaborated in order to promote the models performing the best in challenging flight conditions. That is, it was made of simulations with specific altitudes and angles of attack exhibiting phenomena hard to retrieve with panel methods. In the same way, the test database was designed to ensure that the models performing the best on the validation database also gave good results on different but similarly challenging flight conditions. In addition, new geometries were included to the test database in order to evaluate the generalisation capacities of the selected models. In particular, the Spaceliner was considered with the aim of identifying the limits of the constructed models when dealing with geometries far greater than that of the training database.

Again, the computations were performed in accordance with the models listed in section 4.1. Similarly to the 2D-axisymmetric computations, a particular care was taken to ensure the respect of the DSMC metrics. Hence, for all simulations, the following criteria were imposed: $\Delta t \leq \min(\tau_{\text{coll}})/5$, $\Delta x \leq \lambda_\infty/4$. An adaptive refinement allowing to divide Δx down to $\Delta x/8$

1. The sampling was constrained so that altitudes could only be multiple of 5 km and velocities were rounded to one decimal. The sampling was also biased with respect to the altitude and the angle of attack so that altitudes below 110 km and angle of attack smaller than 90° were more likely to be selected.

was used and the number of numerical particles was increased until $N_{p/c} \geq 10$ in the front of the body. Finally, for the particular case of the IXV, the flaps are responsible for specific phenomena jeopardizing the proper learning of local features in their vicinity. This part of the geometry was therefore truncated in the training database.

Id	Vehicle	Altitude (km)	Velocity (km/s)	Angle of Attack ($^\circ$)	Kn	M_∞
1	IXV	95	9.4	60	2.1×10^{-2}	33.9
2	IXV	100	7.1	170	-	25.1
3	IXV	100	7.4	30	7.5×10^{-2}	26.2
4	IXV	110	9.3	20	4×10^{-1}	28.6
5	IXV	120	6.9	180	3.9×10^{-1}	17
6	IXV	130	7.0	30	5.4	14.6
7	*IXV	90	7.46	47	7.9×10^{-3}	28.2
8	OREX	100	8.5	130	-	30.1
9	OREX	105	6.2	100	-	21.1
10	OREX	140	8.9	110	-	16.4
11	OREX	95	7.5	20	1.6×10^{-2}	27.1
12	OREX	110	9.5	10	2.5×10^{-1}	29.2
13	OREX	120	8.0	40	4.1×10^{-1}	19.8
14	OREX	100	8.5	30	4.2×10^{-2}	30.1
15	OREX	105	6.2	0	1×10^{-1}	21.1
16	OREX	140	8.9	10	6.6	16.4
17	Apollo	95	7.6	100	5.8×10^{-2}	27.4
18	Apollo	105	6.0	90	3.7×10^{-1}	20.4
19	Apollo	105	9.7	140	5.6×10^{-1}	32.9
20	Apollo	110	8.2	80	9.3×10^{-1}	25.2
21	Apollo	130	9.2	150	18	19.2
22	Apollo	140	7.4	120	6.2	13.7
23	Apollo	150	9.5	40	35.2	16
24	Apollo	120	9.0	40	3.2	22.2
25	Apollo	95	7.0	20	4.7×10^{-3}	25.3
26	Apollo	105	8.0	30	1.8×10^{-1}	27.2
27	Apollo	105	6.5	0	2.9×10^{-2}	22.1
28	Apollo	110	8.6	180	1.5	26.5
29	*Apollo	200	9.6	-25	65.4	13
30	*Apollo	95	8.3	-19.4	4.0×10^{-3}	30

*Simulations performed in the frame of the Apollo [37] and IXV [38] reentry analysis.

TABLE 3 – Training database for the neural network models.

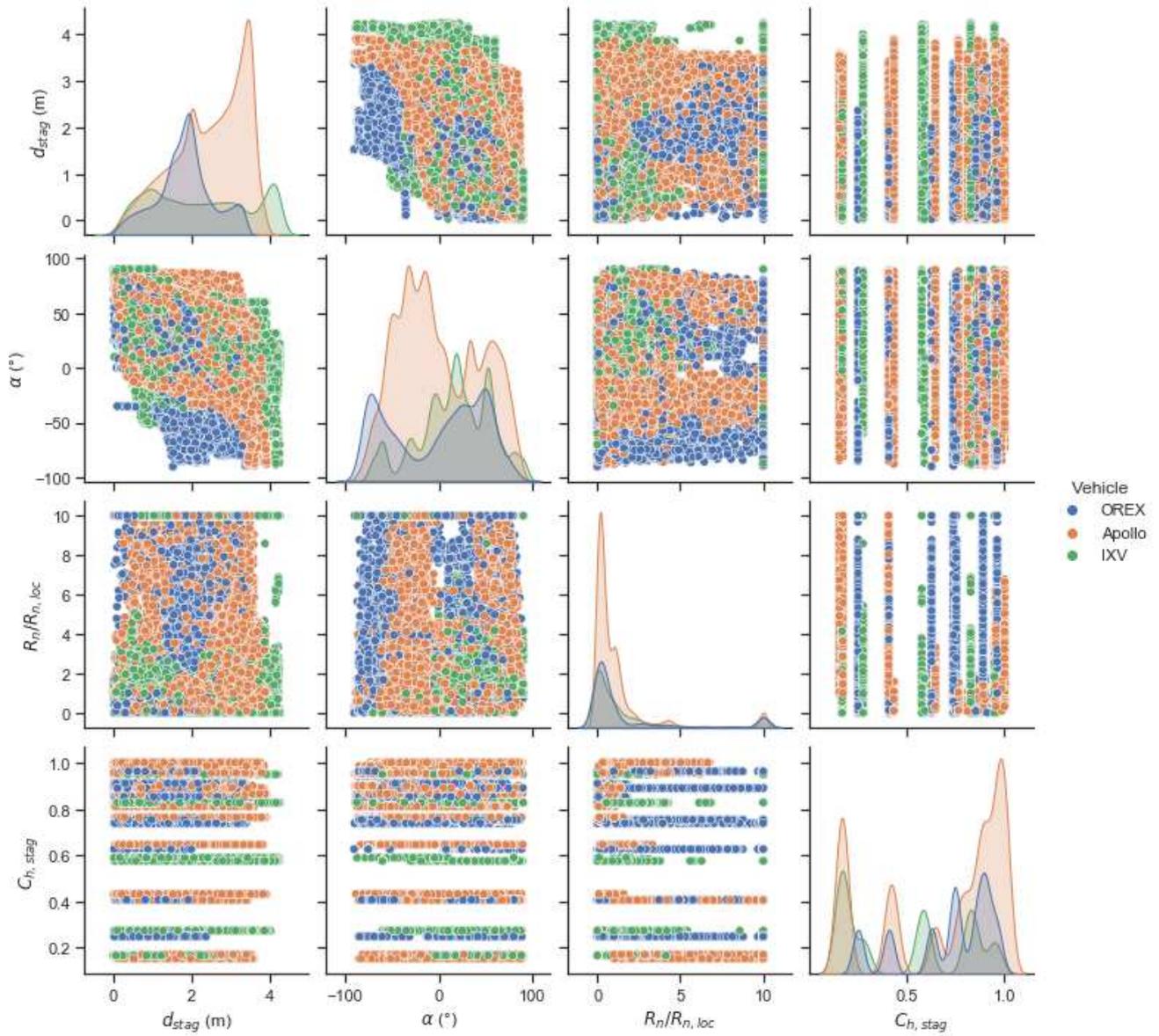


FIGURE 10 – Representation of the training dataset for the C_h model.

Id	Vehicle	Altitude (km)	Velocity (km/s)	Angle of Attack (°)	Kn	M_∞
Validation						
A	*IXV	105	7.5	45	1.3×10^{-1}	26.9
B	*OREX	101	7.5	0	4.3×10^{-2}	26.1
C	*Apollo	105	8.3	-25.2	9.6×10^{-2}	28.1
D	*Apollo	115	9.6	-25	4.5×10^{-1}	26.3
E	*Apollo	170	9.6	-25	25.1	14.4
Test						
I	*Apollo	100	8.3	-24.6	3.3×10^{-2}	29.3
II	*Apollo	130	9.6	-25	2.6	20.1
III	*IXV	115	7.4	47	9.1×10^{-1}	20.3
IV	*OREX	105	7.5	0	8.9×10^{-2}	25.1
V	Sphere	110	8	0	3.3×10^{-1}	24.6
VI	Ram-C	105	10	0	8.4×10^{-1}	34
VII	Orion	100	7.6	-25	4.5×10^{-2}	26.9
VIII	Spaceliner	150	7	45	70.3	16

*Simulations performed in the frame of the Apollo, OREX [37] and IXV [38] reentry analysis.

TABLE 4 – Validation and test databases for the neural network models.

6. Results

This section first presents the results obtained with the kriging models for the $C_{p,stag}$ and $C_{h,stag}$ coefficients. Then, the C_p , C_h and C_f distributions obtained with the neural networks are discussed.

6.1. Stagnation coefficients

The LHS DOE introduced in section 5.1 and the various surrogate models discussed in this section (*i.e.* RSM and kriging) were built with **SMT**, an open access toolbox [47, 48] co-developed by several research teams including NASA, the University of Michigan, ISAE-SUPAERO and ONERA.

6.1.1. Results for $C_{p,stag}$

The results obtained with several kriging and RMS models are summarized in Table 5 where the RMSE and MRE respectively denote the root mean squared and mean relative errors defined by the following expressions:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}, \quad (36)$$

$$\text{MRE} = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{y_i}, \quad (37)$$

where n is the number of validation points, $(y_i)_{i=1..n}$ are the values obtained by numerical modeling and $(\hat{y}_i)_{i=1..n}$ are the values predicted by the surrogates.

One notices that all the reduced model have a similar level of performance which suggests that a simple quasi-linear relation exists between $C_{p,stag}$ and the design variables (Kn, M_∞). However, the comparison of the 3D results plotted in the canonical reference frame and that were obtained with a linear RSM (Figure 11a) and with kriging using a Gaussian correlation (Figure

11c) show radically different behaviour for an equivalent level of preciseness. Hence, in spite of the small MRE for the linear model, the response is physically incorrect. In the same way, the ordinary kriging with an exponential correlation function (Figure 11b) provides the lowest error but the noisy response is inconsistent with the expected tendency. For the kriging models using a Gaussian and matern52², as expected, the S shaped curve is visible for the greatest Mach numbers (Figure 11d). As a consequence, these model seem more relevant despite their non-maximal level of preciseness.

Surrogate model	RMSE	MRE (%)
Ordinary kriging with a Gaussian kernel	0.0197	1.494
Ordinary kriging with an exponential kernel	0.0184	1.376
Ordinary kriging with a matern52 kernel	0.0197	1.521
Linear RSM	0.0194	1.517
Quadratic RSM	0.0179	1.448

TABLE 5 – Errors in the $C_{p,stag}$ prediction obtained with various reduced models.

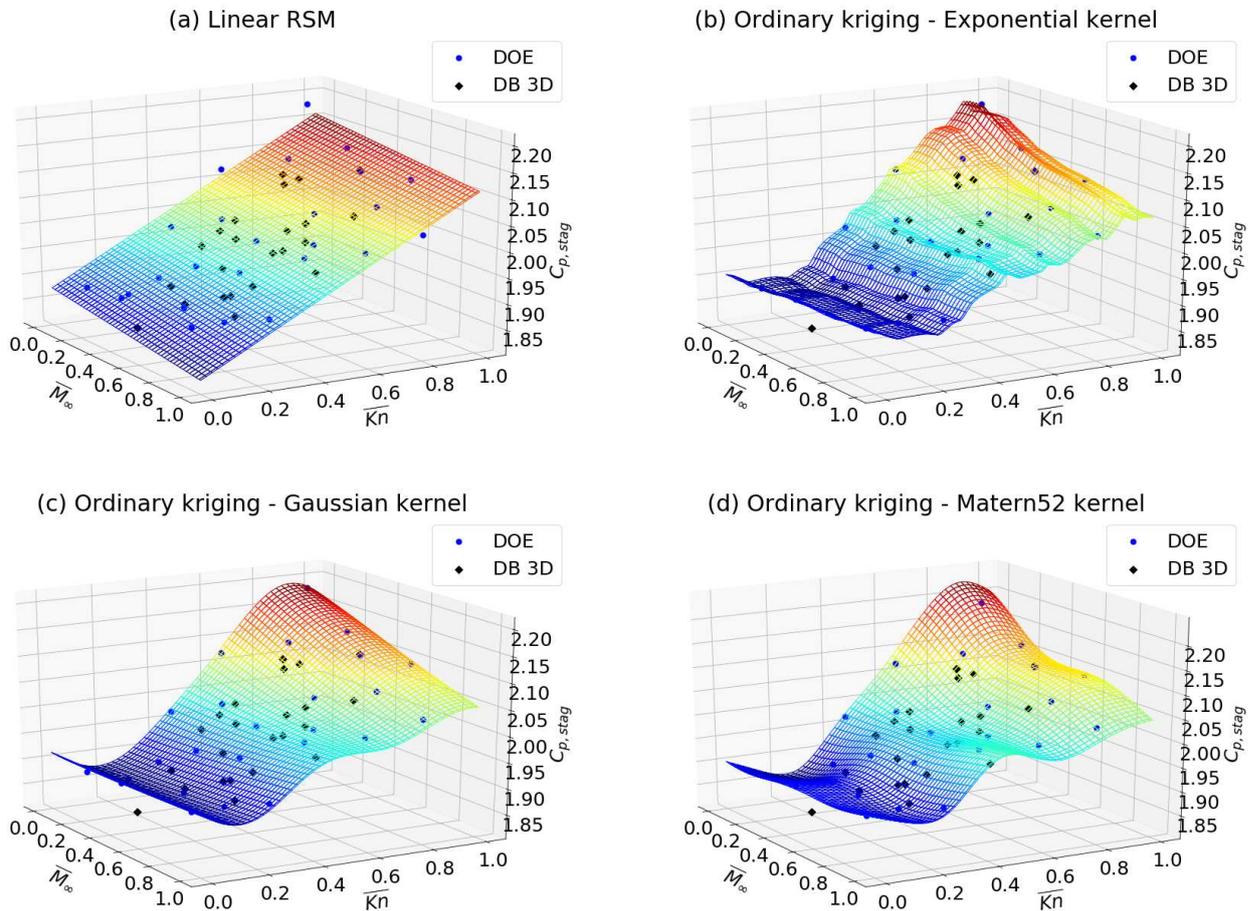


FIGURE 11 – Stagnation pressure coefficients $C_{p,stag}$ obtained with various surrogate models.

From a general perspective, the preciseness of the models could be due to the small variation range of $C_{p,stag}$ which only varies between 1.8 and 2.2 leaving little room for an absurd prediction.

2. The Matérn 5/2 kernel written matern52 is a specific correlation function of form:

$$r(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \prod_{k=1}^d \left(1 + \sqrt{5}\theta_k |x_k^{(i)} - x_k^{(j)}| + \frac{5}{3}\theta_k^2 (x_k^{(i)} - x_k^{(j)})^2 \right) \exp(-\sqrt{5}\theta_k |x_k^{(i)} - x_k^{(j)}|).$$

Nevertheless, the outliers circled in red in Figure 12 indicate that part of this error is due to the stagnation point detection algorithm. Indeed, when handling noisy data coming from DSMC computations, fluctuations in the pressure of the order of a few percent around the stagnation point can result in the extraction of an erroneous value of $C_{p,stag}$, *i.e.* smaller than the actual $C_{p,stag}$. For these 3 points for instance, the knowledge of the real value would slightly bring them closer to the model response. Nonetheless, such adjustment would *in fine* only weakly affect the MRE that is already smaller than 2%.

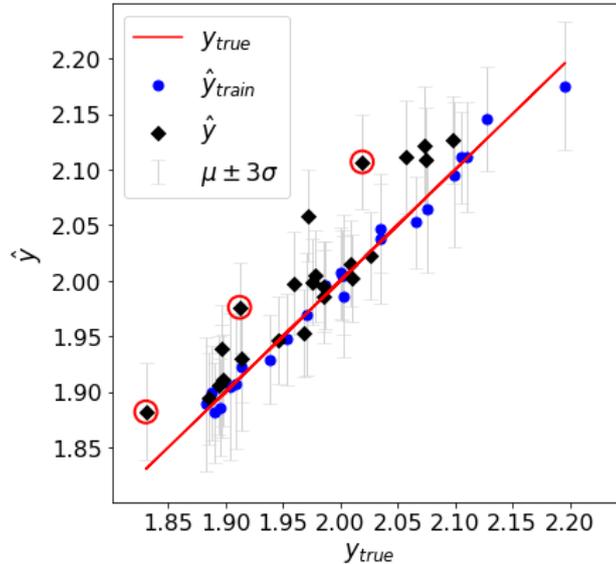


FIGURE 12 – Comparison of the real $C_{p,stag}$ and the one predicted with kriging and the Gaussian kernel.

In order to quantify the benefit of this new model in comparison to the existing model, the validation points were also simulated with **ARES**. For the 23 points of DB 3D, the simulations were performed considering that each vehicle is simply represented by spheres of nose radius equal to their stagnation curvature. For all cases (Figure 9), a MRE of 3.66% was obtained with the existing model, which is a bit more than twice the error obtained with the statistical models.

Among the surrogate models investigated in this section, all of them perform better than the one in **ARES**. However, only the kriging models with the Gaussian and matern52 correlation functions have a physically relevant behaviour. The Gaussian correlation provides a level of precision slightly better which is why it appears as the most suitable model.

6.1.2. Results for $C_{h,stag}$

For the stagnation heat flux coefficient, two databases were used for the validation. Thus, data from Singh and Schwartzentruber [31] and 3D computations from the neural network training database were employed. The results obtained with the surrogate models are summarized in Table 6. As opposed to the stagnation pressure coefficient, these errors show a significantly better level of performance for the kriging models in comparison to the linear and quadratic RSM. Indeed, the 3D view of the results in Figure 13 confirms the non-linear trend of the response. Since the sampling point of Singh and Schwartzentruber database are more localized in the upper rarefied region where $C_{h,stag}$ has a quadratic tendency (Figure 14a), the quadratic model performs well for this particular case. For the kriging models, a global S shape is observed as expected with a small dependence on the Mach number. Again, the Gaussian kernel provides the smoothest and the most relevant response while the exponential correlation function is, as expected, more noisy.

For Singh and Schwartzentruber database, Figure 14a shows that for the less rarefied points, the discrepancy between the kriging prediction and the DSMC results increases. This underesti-

Surrogate model	RMSE (%) [Singh]	MRE (%) [Singh]	RMSE [DB 3D]	MRE(%) [DB 3D]
Gaussian kriging	0.056	4.349	0.0541	6.046
Exponential kriging	0.064	5.534	0.0529	5.789
Matern52 kriging	0.0576	4.499	0.0531	5.803
Linear RSM	0.149	13.78	0.145	14.43
Quadratic RSM	0.0657	5.68	0.0798	9.146

TABLE 6 – Errors in the $C_{h,\text{stag}}$ prediction obtained with various reduced models. The corresponding database is indicated between brackets.

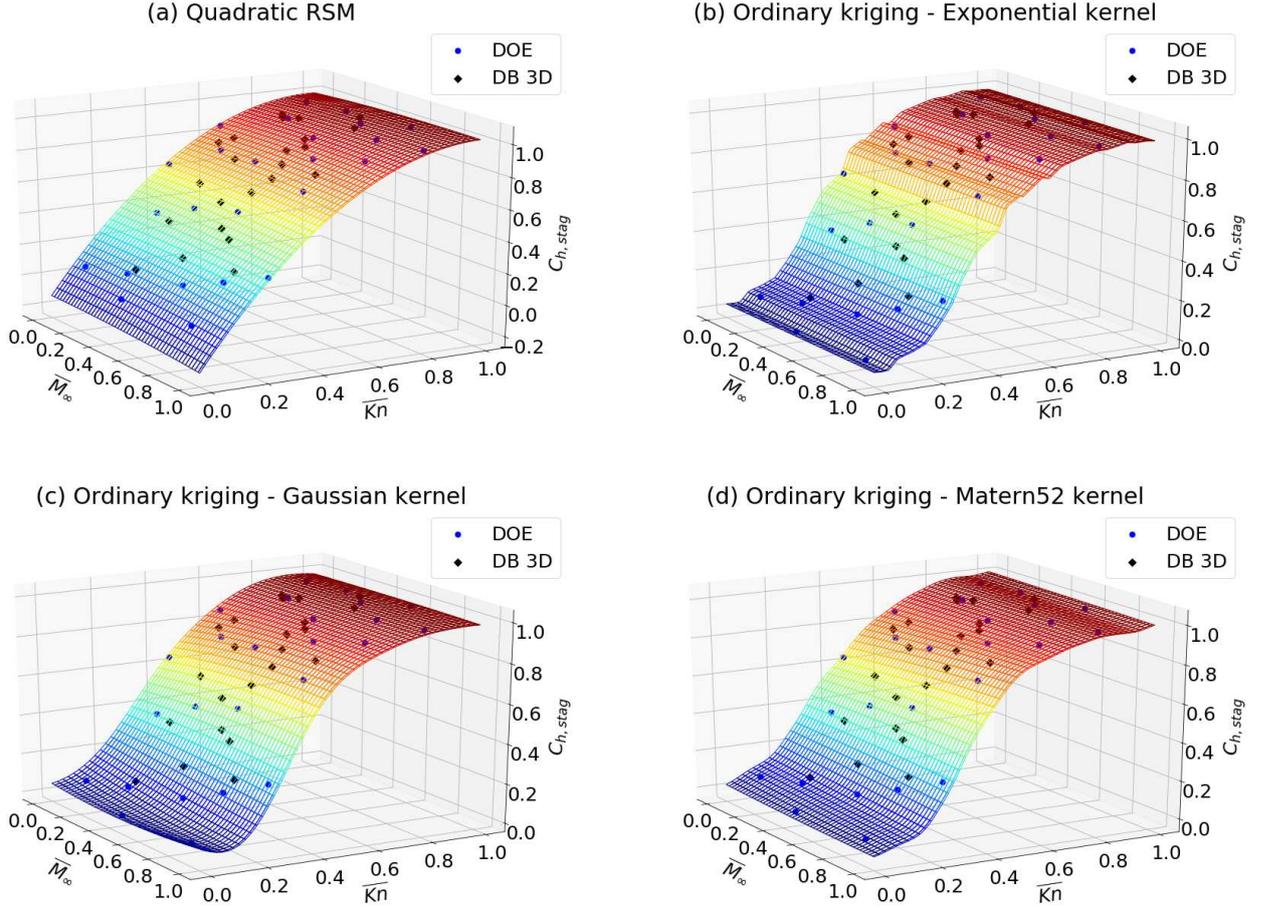


FIGURE 13 – Stagnation heat flux coefficients $C_{h,\text{stag}}$ obtained with various surrogate models.

mation of the DSMC results is however not observed for the second database (Figure 14b). This behaviour is probably due to the difference in the VHS collision model employed for the DSMC simulations. Indeed, as explained in [31], the authors used Bird’s parameters which, as discussed in [38], results in a greater heat flux than that obtained with Stephani’s set of parameters [39] when $Kn < 10^{-1}$. For the two databases, the mean relative error is of the order of 5% which is quite remarkable given the wide variation range of $C_{h,\text{stag}}$.

Finally, similarly to the stagnation pressure coefficient, **ARES** was used to estimate the stagnation heat flux of the DB 3D validation points. For this quantity, several correlations exist in the continuum regime which is why several formulations were tested. Since these are generally obtained under a super-catalytic wall assumption, only the points of the database for which no catalytic effects are expected were considered. A mean relative error comprised between 12.2 %

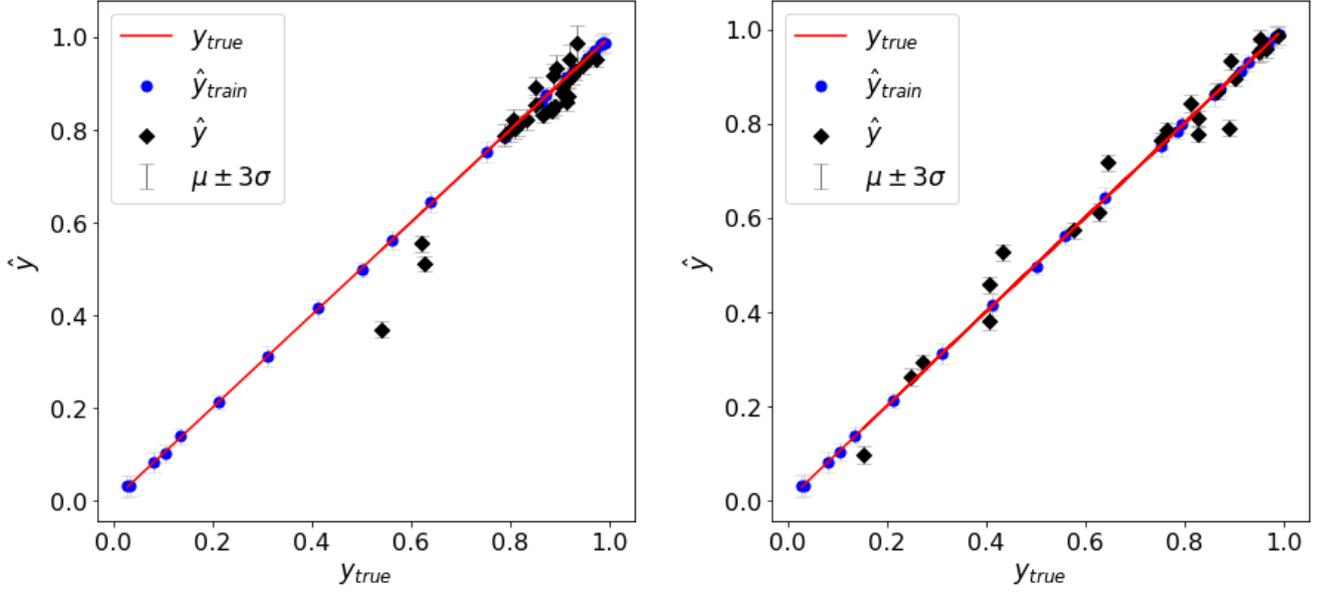


FIGURE 14 – Comparison of the real $C_{h,stag}$ and the one predicted with kriging and the Gaussian kernel for the Singh and Schwartzentruber (left) and DB 3D (right) databases.

(Johnston and Brandis [29]) and 13.1 % (Sutton-Graves [49]) was obtained. This time again, the mean relative error of the kriging models is half that of the existing model.

The models developed in this section constitute a first way of significant improvement with respect to the existing models. Indeed, the kriging models obtained provide errors twice smaller than that of **ARES** for both $C_{p,stag}$ and $C_{h,stag}$. In both cases, the kriging model with a matern52 and a Gaussian correlation functions perform very well and provide a physically relevant tendency. These models thus constitute a solid alternative to the currently employed models.

6.2. Distribution coefficients

The neural networks discussed in this section were built with the library **keras** [32, 50]. Once a specific architecture was chosen by evaluation on the validation database, the selected network was finally tested by comparison with the DSMC results. Here, the test results are presented in two ways. In one side, the distributions predicted by the networks are compared to the DSMC distributions (left figures). In parallel, a qualitative comparison is made by plotting the distributions obtained by DSMC, with **ARES** and with the networks in a normalized form (*i.e.* divided by their respective maximal values) along the vehicles centerline (right figures). This way, the tendencies can be compared independently from the error committed at the stagnation point.

6.2.1. Results for C_p

The first network is of the form $\hat{C}_p = \hat{f}(C_{p,\text{stag}}, \alpha)$. Because of the simplicity of this formulation and considering the good level of precision of the existing methods, a major improvement is not expected for this model. The selected network is made of 3 hidden layers [32, 64, 128] with ReLU activation functions. It was trained with an Adam Optimizer and *weight decay* regularization was used to prevent the chances of overfitting [32].

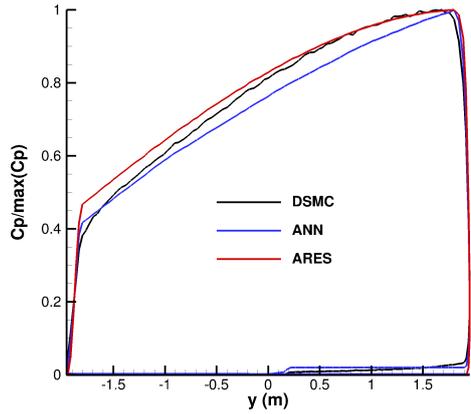
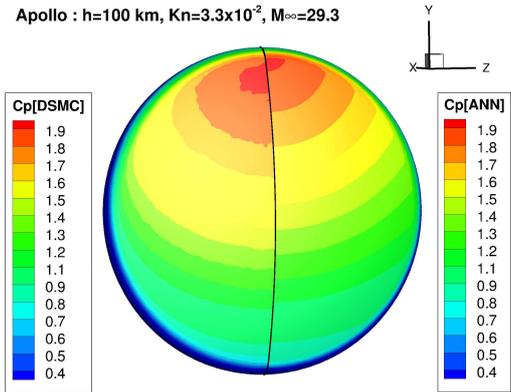
Figure 15 shows the results for the 4 geometries the network was trained on (cases I to IV in Table 4). For the Apollo case at 100 km (I), a good agreement is obtained. However, the DSMC stagnation point differs from the elements where α is close to 90° . For this reason, the peak of C_p predicted by the reduced models (ANN and **ARES**) is further away from the DSMC peak. At 130 km (II), one observes the excellent agreement between all results.

For the IXV case (III), a good agreement is obtained between the DSMC and the ANN. In addition, one denotes that the noisy DSMC values in the stagnation region are smoothed by the reduced models. For both surrogate models, the tendencies are relevant and the ANN results seem to underestimate the DSMC distribution while **ARES** has a tendency to overestimate it. For OREX (IV), a very good agreement is observed between the ANN and the DSMC results. In comparison to **ARES**, the results along the centerline indicate that the ANN performs slightly poorer on the spherical part.

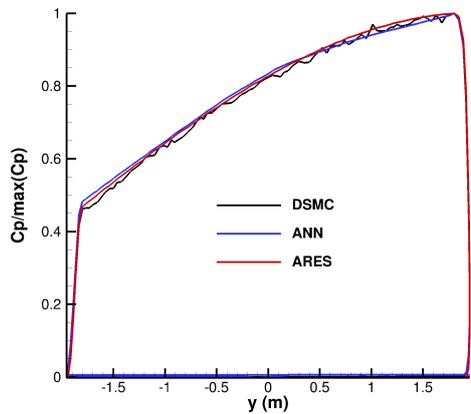
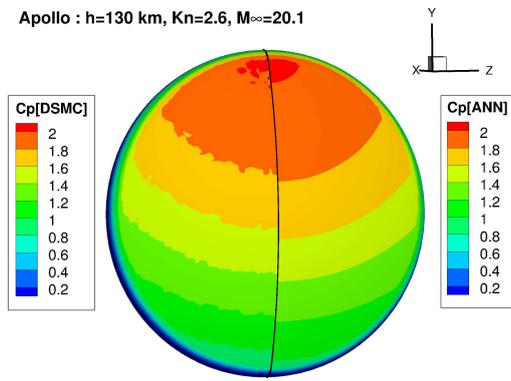
The results with respect to the new geometries (cases V to VIII, not included in the training database) are represented in Figure 16. For the elementary case of a Sphere (V), an excellent agreement is obtained with the ANN. One also notices that for the angles $\theta \geq 50^\circ$ along the sphere, the ANN performs better than **ARES**. For Ram-C, the 9° sphere-cone (VI), an excellent agreement is obtained with the two reduced models. For Orion (VII), a similar behaviour as that of Apollo (III) is observed and, as expected, a better estimation is provided by the ANN model in the shadow region where the results of **ARES** are compromised by the continuum regime $C_p = 0$ assumption. Finally for the Spaceliner, the distributions show that the ANN model slightly underestimate the stagnation value (relative difference of 9%) because the stagnation point is located in a region where $\alpha \simeq 76^\circ$. Overall, the tendency is however excellent as opposed to that obtained with **ARES**.

For the C_p coefficient, the results indicate that an equivalent level of preciseness is observed for the ANN and **ARES** in most cases. However, since the new model does not suffer from the continuum regime assumption (*i.e.* $C_p = 0$ in the shadow region), for such elements, it leads to better results in the transitional regime. Finally, the analysis of the Spaceliner case suggests that in spite of a training limited to 3 specific geometries, the ANN model is reliable even for radically different geometries.

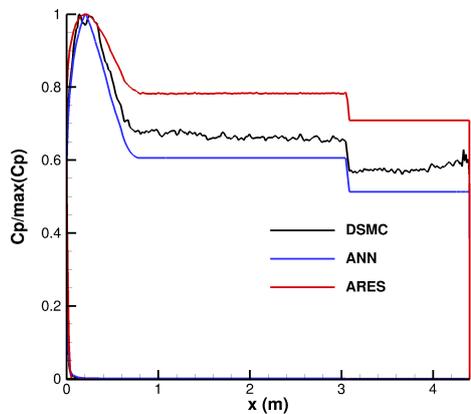
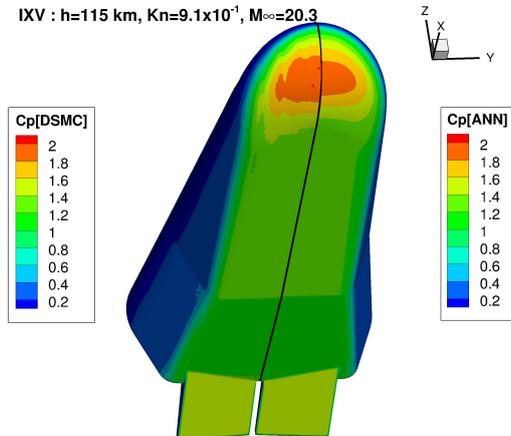
Apollo : $h=100$ km, $Kn=3.3 \times 10^{-2}$, $M_\infty=29.3$



Apollo : $h=130$ km, $Kn=2.6$, $M_\infty=20.1$



IXV : $h=115$ km, $Kn=9.1 \times 10^{-1}$, $M_\infty=20.3$



OREX : $h=105$ km, $Kn=8.9 \times 10^{-2}$, $M_\infty=25.1$

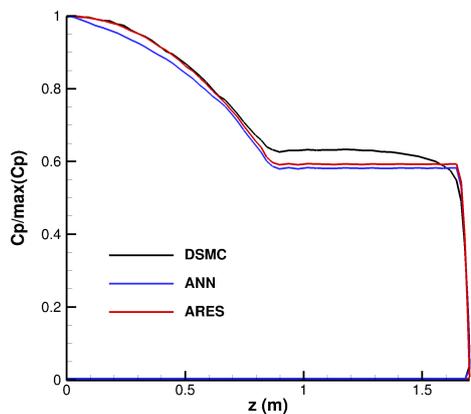
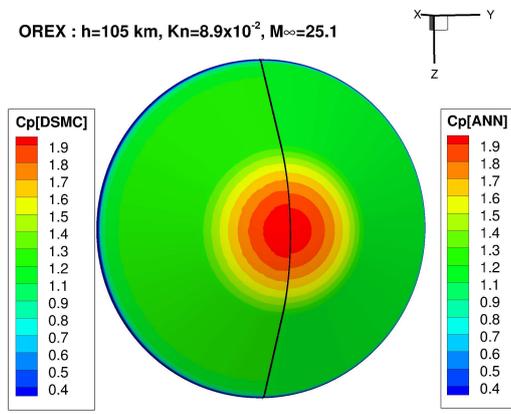


FIGURE 15 – Comparison of the C_p coefficient obtained with the neural network (ANN), with SPARTA (DSMC) and with ARES for Apollo (I-II), the IXV (III) and OREX (IV).

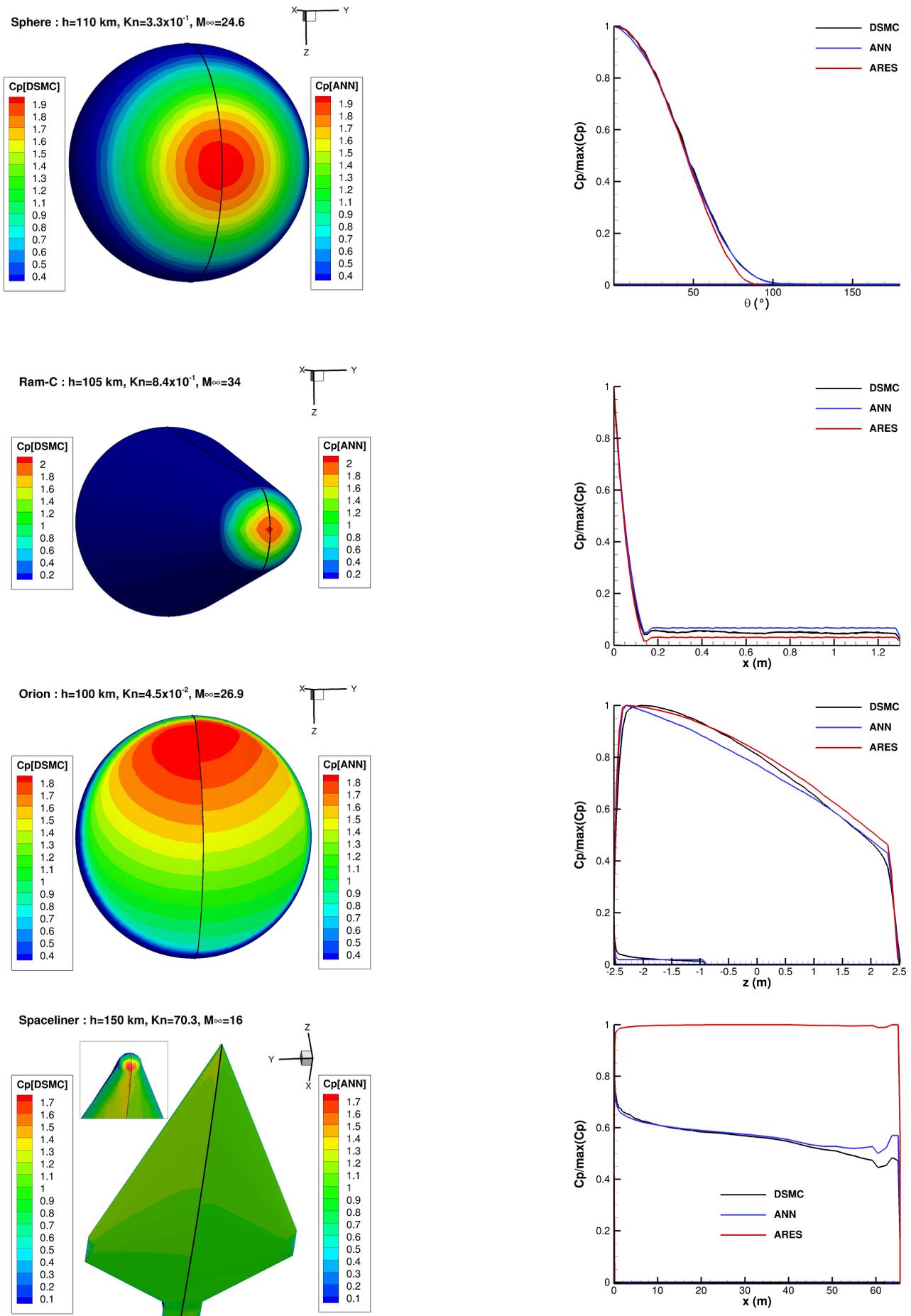


FIGURE 16 – Comparison of the C_p coefficient obtained with the neural network (ANN), with SPARTA (DSMC) and with ARES for a Sphere (V), Ram-C (VI), Orion (VII) and the Spaceliner (VIII).

6.2.2. Results for C_h

The second network is of the form $\hat{C}_h = \hat{f}(C_{h,\text{stag}}, \alpha, d_{\text{stag}}, R_n/R_{n,\text{loc}})$. It is made of 4 hidden layers [32, 64, 128, 256] and ReLU activation functions. The same optimizer and regularization methods as for the C_p model were employed. Furthermore, as discussed at the end of section 6.1.2, the comparison between **ARES** and the DSMC computations are relevant only if catalytic effects are negligible. For some cases, this is not true and only the ANN results are discussed.

Figure 17 shows the results for simulations I to IV. In the case of Apollo (I), the contours indicate an excellent agreement for the peak of heat flux in the shoulder region. Further down the spherical portion, a discrepancy is however noticeable. Overall, a good agreement is observed. For the 130 km case (II), a good agreement is obtained but a small erroneous peak is predicted with the ANN model leading to an overestimation of the heat flux.

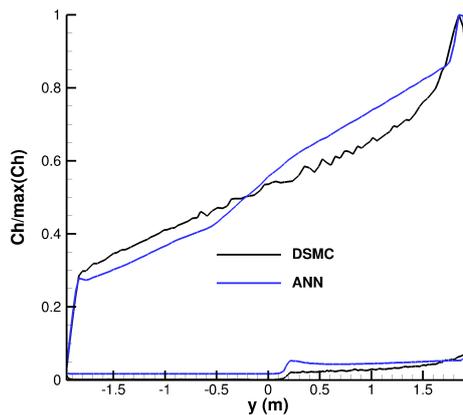
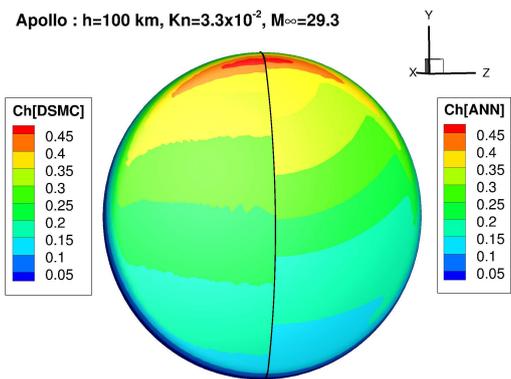
For the IXV (III), the contours indicate a very good agreement in terms of amplitude in the stagnation region. As for the results along the centerline, the results show an abusive dependence of the response to d_{stag} that is not expected for such level of rarefaction and that leads to the underestimation of the DSMC results. For this case, **ARES** and the ANN predictions suffer from an equivalent error relatively to the stagnation value. For the OREX case (IV), the contours indicate a 3 to 8% relative difference between the ANN and DSMC results in the nose region. As for the distribution tendency, the comparison with **ARES** shows a poorer behaviour along the spherical section but a good capture of the heat flux diminution along the conical portion.

Concerning the new geometries, the results are given in Figure 18 and the following points were noted. For the sphere (V), similarly to the OREX case (IV), a linear trend is observed in the stagnation region where the heat flux is slightly underestimated. Further along the centerline ($\theta \geq 50^\circ$), the behaviour is well reproduced with the ANN model and more accurately than with **ARES**. For the Ram-C case (VI), a slight overestimation is observed in the stagnation region (relative difference of 4%) but the overall tendency is well retrieved.

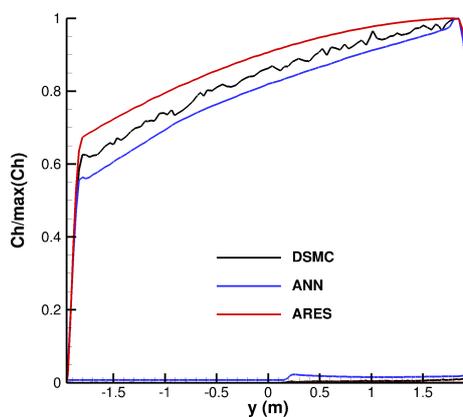
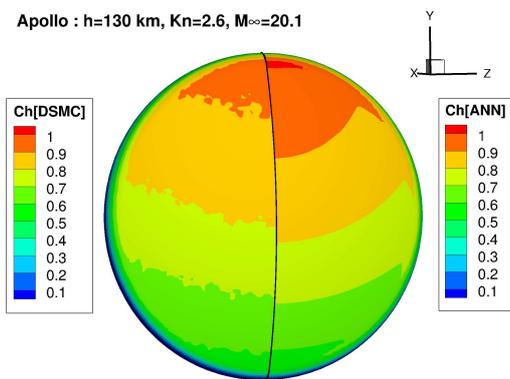
For Orion (VII), similarly to the Apollo (I) case, the peak of heat flux is precisely reproduced and the same underestimation is noticed further down the spherical section. Finally for the Spaceliner (VIII), the results show that this time, the stagnation value is well retrieved but the overall tendency is absurd. This aspect was actually expected since the ANN model uses an absolute distance and was only trained with objects of size smaller than 4 m.

The results presented in this section suggest that the targeted dependency were quite well assimilated for most cases. Indeed, as illustrated by the Apollo (III) and Orion (VI) cases, the characteristic heat flux peak in the shoulder region of probes reentering in incidence is very well reproduced by the ANN model while it constitutes a known limitation of the existing models [7]. In the same way, the OREX case (IV) shows that the dependence to d_{stag} along the conical section, which is another limiting points of the current models, is well inferred by the ANN. However, because altitudes around 100 km are more represented in the database, the rarefaction dependence was not perfectly assimilated as evident from the Apollo (II) and IXV (III) cases. This could be addressed by increasing the number of simulations at higher altitudes. Finally, the Spaceliner points out the consequence of using an absolute distance as a design parameter which suggests that some kind of non-dimensional parameter might be more appropriate to deal with such geometries. Another possibility would be to include bigger geometries in the training database. In first approach, scaling the Spaceliner d_{stag} parameter according to the maximal value in the training database yields a much better result (Figure 19).

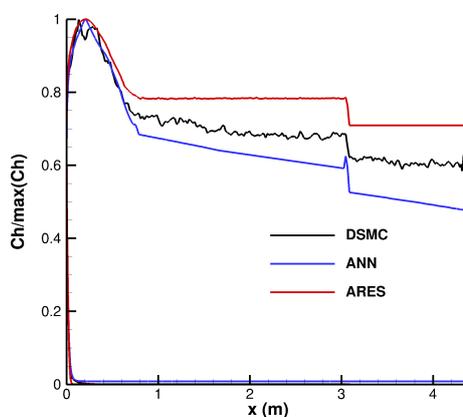
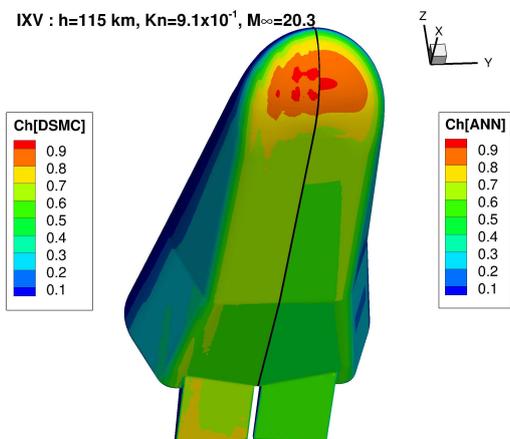
Apollo : $h=100$ km, $Kn=3.3 \times 10^{-2}$, $M_\infty=29.3$



Apollo : $h=130$ km, $Kn=2.6$, $M_\infty=20.1$



IXV : $h=115$ km, $Kn=9.1 \times 10^{-1}$, $M_\infty=20.3$



OREX : $h=105$ km, $Kn=8.9 \times 10^{-2}$, $M_\infty=25.1$

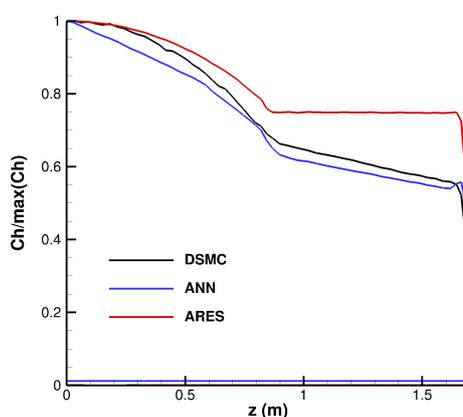
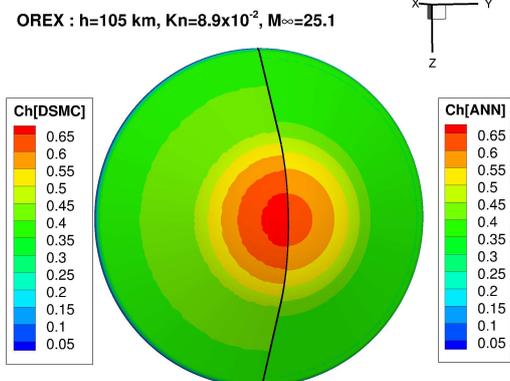


FIGURE 17 – Comparison of the C_h coefficient obtained with the neural network (ANN), with **SPARTA** (DSMC) and with **ARES** for Apollo (I-II), the IXV (III) and OREX (IV).

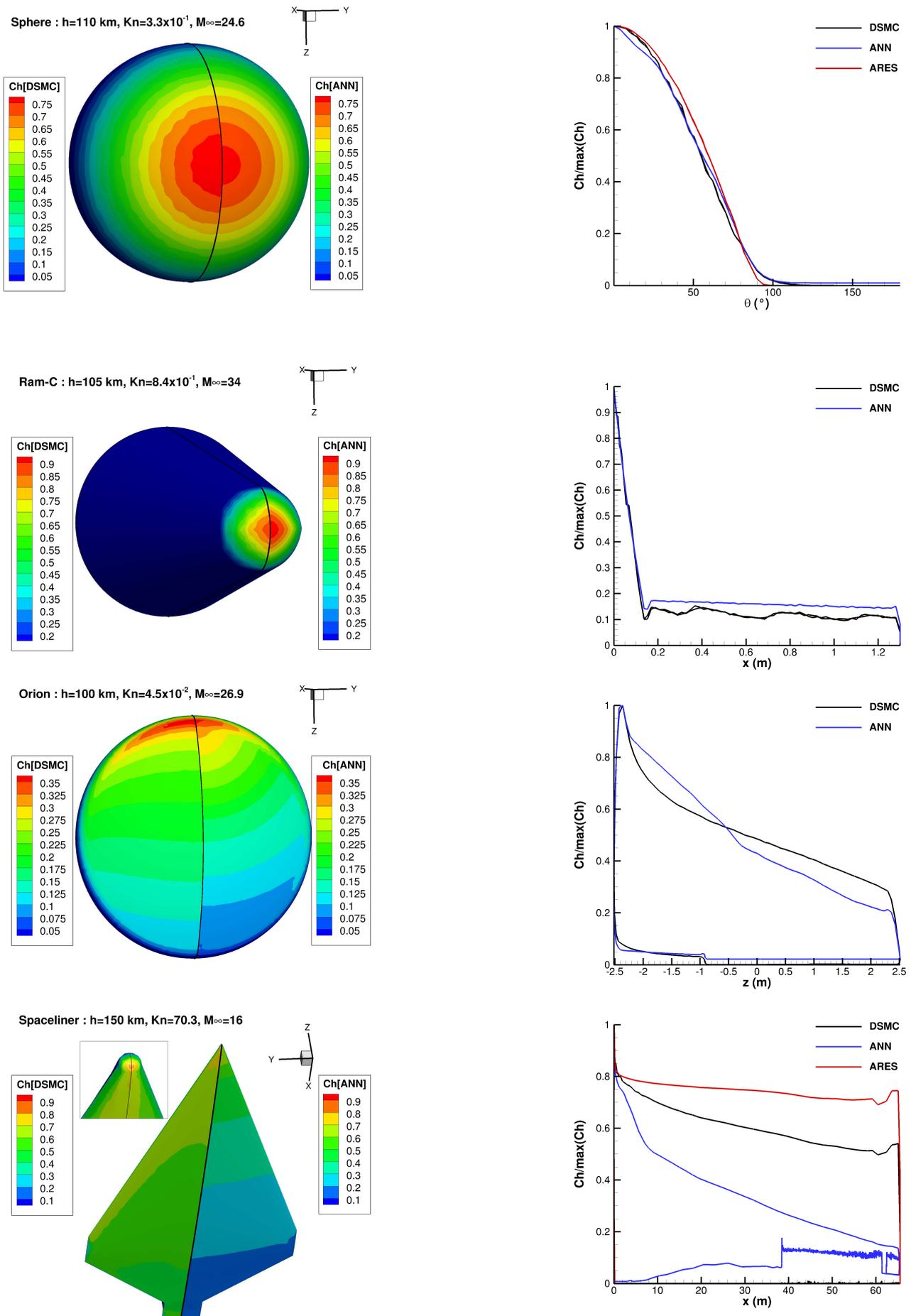


FIGURE 18 – Comparison of the C_h coefficient obtained with the neural network (ANN), with **SPARTA** (DSMC) and with **ARES** for a Sphere (V), Ram-C (VI), Orion (VII) and the Spaceliner(VIII).

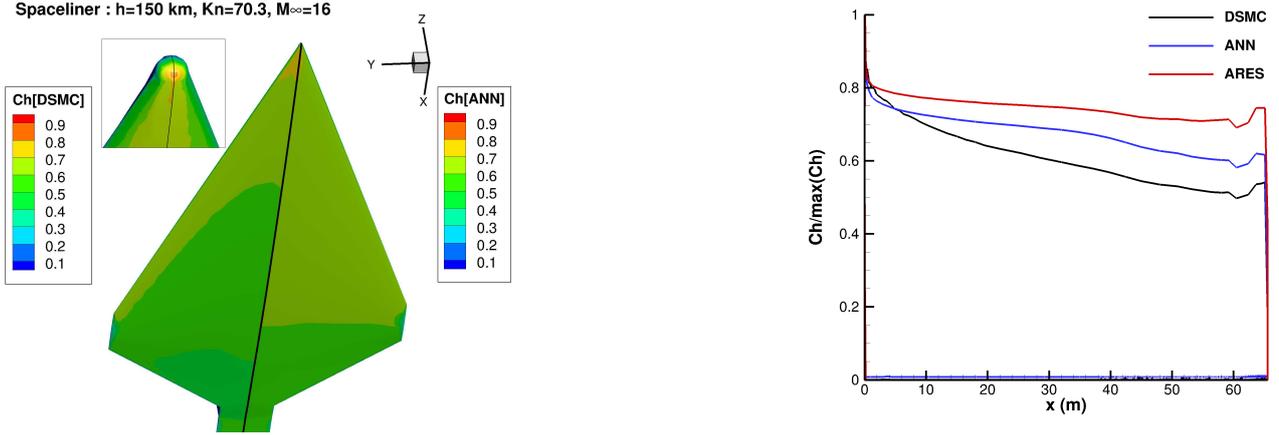


FIGURE 19 – Comparison of the C_h coefficient obtained with the neural network (ANN), with **SPARTA** (DSMC) and with **ARES** for the Spaceliner(VIII) and a scaled d_{stag} parameter.

6.2.3. Results for C_f

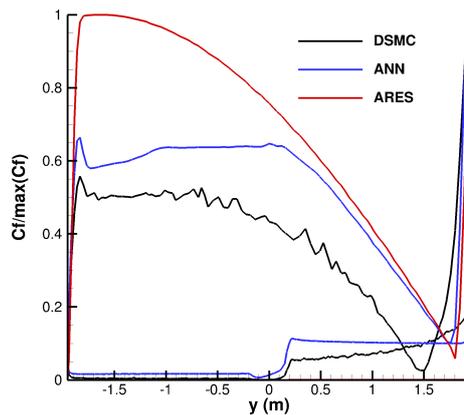
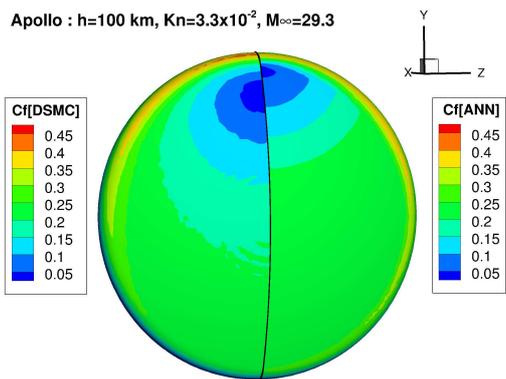
The third and last model has the following form: $\hat{C}_f = \hat{f}(C_{h,stag}, \alpha, R_n/R_{n,loc})$. The selected architecture is made of 3 hidden layers [32, 64, 128] and the activation functions are ReLU functions. Similarly to the two other networks, an Adam optimizer and weight decay were used.

Figure 20 gives the results for the first test simulations (I to IV). Similarly to what was observed for C_p and the Apollo case at 100 km (I), the stagnation point is not exactly where $\alpha \simeq 90^\circ$ which is why the minimal value of C_f predicted with the reduced models is slightly shifted away from that of the DSMC computation. As for the amplitude of the distribution further down the spherical section, the ANN model is more precise relatively to the maximal value in the shoulder region. At 130 km (II), a good agreement is observed for both **ARES** and the ANN but the stagnation value predicted by the ANN is highly overestimated. For the IXV (III) and OREX (IV) cases, the tendencies are accurately retrieved with the ANN model but the value at the stagnation point is still overestimated.

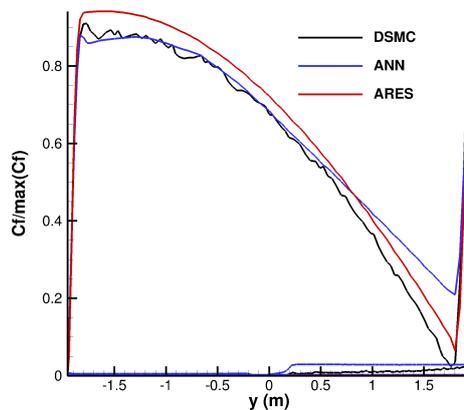
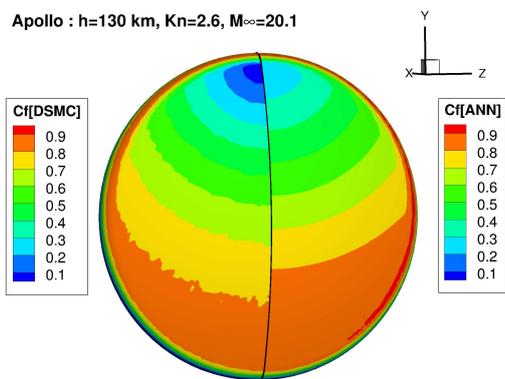
For the new geometries (V to VIII) represented in Figure 21, the same observation is noted for the sphere case (V) where the tendency is accurate but the stagnation value is again highly overestimated. For the Ram-C case (VI) and similarly to **ARES**, an excellent agreement is obtained with the ANN model. For Orion (VII), the same shifting of the stagnation point is observed but overall, the ANN model is in much better agreement with the DSMC results than **ARES**. Finally for the Spaceliner case (VIII), a similarly poor result is obtained with both **ARES** and the ANN. This time, the effect of viscosity which results in the bended tendency along the centerline is not captured by any of the reduced models. This behaviour is characteristic of physical phenomena enhanced by long flat geometries and even with the addition of a distance parameter could hardly be assimilated with the considered database.

As discussed in section 5.2.1, the friction coefficient C_f is particular in that it is characterized by a zero value in the stagnation region. The adopted approach relies on the use of $C_{h,stag}$ in order to infer the appropriate behaviour according to the rarefaction level. Except for the particular case of the Spaceliner, the expected behaviour was well retrieved but the ANN model also evidenced a tendency to overestimate C_f in the stagnation region. Presently, no clear way of improvement has emerged with respect to this problem. As for the Spaceliner, the poor results obtained with both **ARES** and the ANN suggest that, for this kind of geometry promoting viscous effects, a specifically trained model might be needed.

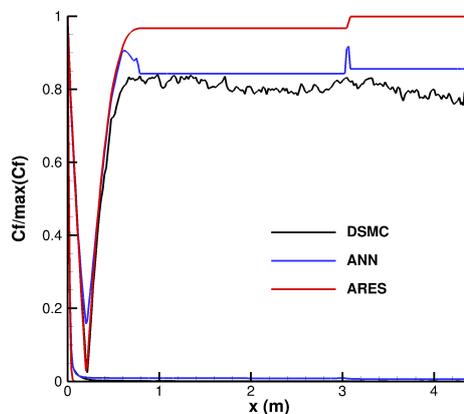
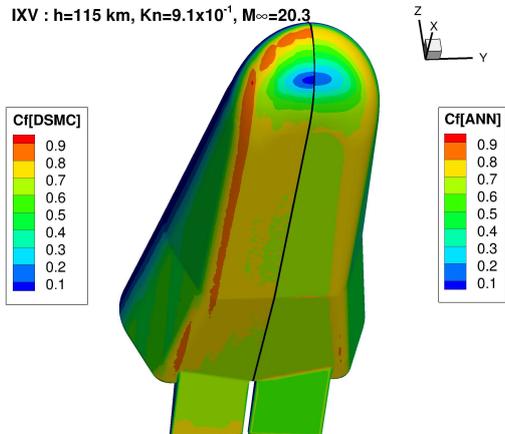
Apollo : $h=100$ km, $Kn=3.3 \times 10^{-2}$, $M_\infty=29.3$



Apollo : $h=130$ km, $Kn=2.6$, $M_\infty=20.1$



IXV : $h=115$ km, $Kn=9.1 \times 10^{-1}$, $M_\infty=20.3$



OREX : $h=105$ km, $Kn=8.9 \times 10^{-2}$, $M_\infty=25.1$

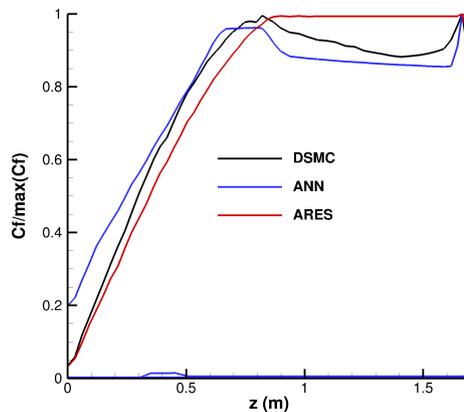
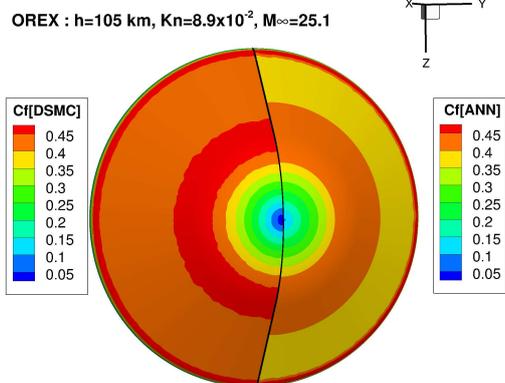


FIGURE 20 – Comparison of the C_f coefficient obtained with the neural network (ANN), with **SPARTA** (DSMC) and with **ARES** for Apollo (I-II), the IXV (III) and OREX (IV).

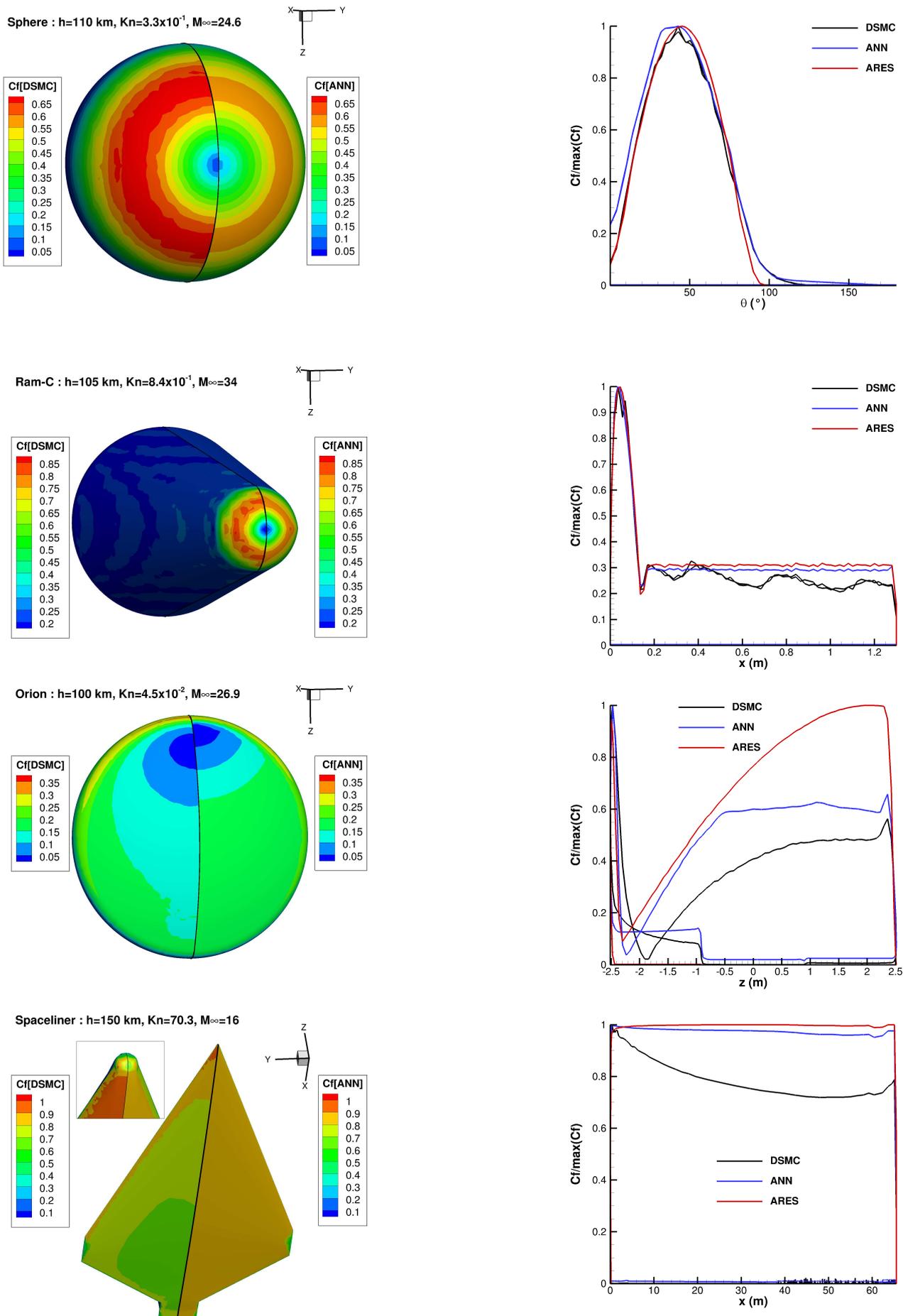


FIGURE 21 – Comparison of the C_f coefficient obtained with the neural network (ANN), with SPARTA (DSMC) and with ARES for a Sphere (V), Ram-C (VI), Orion (VII) and the Spaceliner (VIII).