POLITECNICO DI MILANO
Dipartimento
di
Ingegneria Aerospaziale

KUNGLIGA TEKNISKA HÖGSKOLAN
Department of Aeronautical
and Vehicle Engineering

Adaptive Fidelity Aero Data Generation

Relatori: Prof. Sergio Ricci
Prof. Arthur Rizzi

Correlatore: Prof. Jesper Oppelstrup

Tesina di laurea di:
Daniele Santini Matr. 707896

Anno accademico 2008/2009

Sommario

Durante la fase di progetto e design di un aereo é importante verificare che il velivolo sia in grado di rispondere alle specifiche di progetto. A tale scopo si effetuano test in galleria del vento, e qualora i risultati non fossero soddisfacenti si modifica il progetto di conseguenza. Le stime provenienti da tali test sono accurate e affidabili, ma l'utilizzo della galleria del vento, delle opportune strumentazioni e la creazione di modelli che riproducano il piú fedelmente possibile il reale velivolo costituiscono dei costi aggiuntivi. Per poter ridurre tali costi prima dei test in galleria si utilizzano dei modelli computazionali in grado di simulare la distribuzione delle forze aerodinamiche con un buon indice di fedeltá; alcuni esempi sono i programmi Datcom, Tornado, ed in generale programmi di implementazione degli algoritmi risolutivi delle equazioni di Eulero. Esistono diversi programmi di questo tipo, ognuno col proprio livello di accuratezza. Di norma maggiore é la precisione del modello usato maggiore é il costo computazionale per il suo utilizzo, il che puó risultare un'ulteriore complicazione in fase di progetto[1].
Al fine di ovviare tale problema é stata creata una procedura chiamata "Data Fusion Approach" il cui scopo é quello di fornire delle stime sufficientemente accurate tramite la fusione di stime a basso livello di precisione con poche misurazioni piú accurate. I primi dovrebbero determinare l'andamenteno delle quantitá in analisi, mentre i secondi dovrebbero correggere tali "trend" ottenendo delle distribuzioni piú vicine a quelle reali. Questo approccio risulta sufficientemente accurato per semplici casi, ma perde di fedeltá nel momento in cui viene applicato a casistiche meno lineari.
Nonostante ció, il concetto alla base della procedura, vale a dire il progressivo miglioramento degli andamenti stimati tramite aggiunta di ulteriori misurazioni, puó essere comunque utilizzato al fine di ottimizzare l'utilizzo dei modelli computazionali. In questo report viene presentata una procedura iterativa basata su tale concetto, la quale una volta applicata permette di generare database contenenti migliaia di stime con poche centinaia di computazioni, con conseguente risparmio in elaborazione e tempo.
Prima viene introdotta e descritta nel dettaglio la "Data Fusion Approach", si procede poi con l'analisi degli algoritmi di interpolazione conosciuti col nome "Kriging", si introducono i criteri di campionamento che permettono di identificare le misure da aggiungere per le correzioni nel processo iterativo,

[1] Utilizzo di supercomputers in grado di gestire database di dimensioni notevoli, nonché lunghi tempi di attesa per ottenere le stime

ed infine si mostrano i risultati provenienti dall'applicazione di tale procedura al simulatore "Tornado"; i risultati mostrati sono relativi ad un Boeing 747 e ad un caso ipotetico creato al fine di evidenziare la qualitá delle stime anche in casi piú complessi.

Abstract

The object of this report is to introduce an iterative procedure based on the basic concept of the data fusion approach whose aim is to optimize the usage of computational models. By using such a procedure the estimates are highly accurate and require lower computational costs, speeding up the usage of these models.

Keywords: Computer Experiments, Correlation Parameter, Data Fusion, Extrapolation, Kriging, Sampling

Contents

List of Figures

List of Tables

Chapter 1

Introduction

During the design phase of an aircraft it is important to verify that such a project is able to respond to the specifics the airplane itself is designed for. One way to do that is to test a model of the designed aircraft in a wind tunnel session, and according to the consequent results modify the project whenever it seems necessary to. Unfortunately this kind of test can be quite expensive, and that would increase the total cost of the aircraft production. In order to reduce those costs computational models are available, and by using them it is possible to get a good estimate of the aerodynamic properties and of the performances of the airplane. That is why simulators such as Datcom, Tornado or Euler Equations have been developed.
Datcom (Data Compendium) is a program written in Fortran IV, whose aim is to derive some aerodata by using some statistical result obtained for certain aircraft configurations and then adapting them to the actual airplane by introducing its geometrical and envelope features. It allows to analyze the aircraft aerodynamic characteristics in all of the flight regimes, although some of them are limited or not accessible depending on the aircraft attitude. As stated in [10], its fundamental purpose is to provide a systematic summary of methods for estimating stability and control characteristics in preliminary design applications. Consistent with this philosophy, the development of the Digital Datcom computer program is an approach to provide rapid and economical estimation of aerodynamic stability and control characteristics. As stated in [6], it requires flight conditions and geometry description in order to estimate the required output. Its estimate are quite accurate for most of the conventional aircraft for subsonic and supersonic regimes. Transonic data are found from subsonic and supersonic data, leading to uncertainty in the results. This method is based on inviscid flow assumption, and friction correction are later added to the results. It can be downloaded on [14].

Tornado is a 3D-vortex lattice program with flexible wake coded in Matlab. It is based on the Vortex Lattice Method, which works with the actual geometry of the airplane wing and tail[1], dividing them into a certain number of panels in which both a lifting vortex and a control point are located. In this way the program can take the effects of the mutual induction of the lifting surfaces into account, and it can also estimate the entity of the loads both spanwise and chordwise, although it shows some problem when it comes to handle transonic regimes. Its outputs are: 3D forces acting on each panel, aerodynamic coefficients in both body and wind axis and stability derivatives with respect to angle of attack, angle of sideslip, angular rates and rudder deflections. It can be downloaded on [15].
The Euler equations are a simplified version of the Navier-Stokes ones, which have been gotten by assuming with zero viscosity and heat conduction terms. The Euler equations can be applied to compressible as well as to incompressible flow by using either an appropriate equation of state or that the divergence of the flow velocity field is zero, respectively.
In order to simulate the force distribution over the aircraft, some program such as CEASIOM are still developing as well. As it is possible to read in [3] the aim of this program is exactly to make the aircraft conceptual design phase easier and cheaper.
There are several kind of computational models, all of them with different levels of accuracy. Usually the best way to get precise results is to use the model with the highest accuracy, but that comes together with high computational costs. For this reason the data fusion approach has been developed. As it is possible to read in [5], this is a procedure that let to obtain quite accurate results by using a data set coming from a poorly accurate model and improving them by using few results coming from more accurate ones. This approach is based on the concept that the initial data set gives the trend of the analyzed quantity, while the high-fidelity models adjust such a trend to its actual values, and is effective as far as it is used on simple cases, in which the force distribution is quite linear and predictable. As this methodology is applied on more complicated and less linear cases the results worsen in quality, making the entire procedure less reliable.
Nonetheless the idea of improving a trend by adding additional samples can still be used to optimize the usage of the computational models previously mentioned, in order to get accurate results with less computational costs. Based on this concept an iterative procedure has been developed, which is applied on one model, the initial trend is given by a data set collecting few

[1]For the time being the fuselage is not simulated, but they are working on a new version of the program in which both the fuselage contribution to the drag and the wing stall are taken into account.

estimates of the studied responses and the final data set collects thousands of them.
This report goes through the theoretical bases behind the procedure, showing also the features of the several toolboxes and simulators used in the program, and at last shows some results coming from its application on a simple case, a Boeing 747 with no control surfaces, and on a hypothetical case which was created to demonstrate that this approach can handle complicated and non-linear responses as well.

Chapter 2

Data Fusion Approach

This chapter gives an introduction to the Data Fusion Approach, with a brief discussion on the concept of low fidelity and high fidelity results.

Flight simulation and aircraft performance estimates require a database of look-up tables for aerodynamic forces and moments as functions of the flight state - velocity, angle of attack, sideslip, control surface deflections, etc. The database is a set of tables of all the required forces and moments with around ten independent variables/dimensions and it is queried by interpolation. A ten-dimensional table of sufficient resolution would be enormous, and it is simplified by exploiting weak dependencies and approximate linearization into a number of three-dimensional or two-dimensional tables.
The task addressed in this work is to fill such a 3-D $m \times n \times k$ table by computation of only a small number of strategically chosen points from which the rest can be interpolated by some method for interpolation from an "unstructured" set of points. Kriging with a linear trend was employed, implemented in the DACE Toolbox as referred in [4], and an iterative procedure, which adds points incrementally until an accuracy estimate criterion is satisfied, as well.
The independent variables in the aerodatabase application are angle of attack α, Mach number M and side slip angle β, $x = (M, \alpha, \beta)$, and the dependent variables are coefficients of lift , drag, pitching moment, roll moment, and yaw moment, summarized into $f = (C_L, C_D, C_m, C_{roll}, C_n)$, $f_1 = C_L(M, \alpha, \beta)$, etc. The iterative procedure previously mentioned is based on the basic concept of the so called data fusion approach.

The aim of this procedure is to "fuse" low fidelity data with a few high-quality ones so that the resulting data set is quite accurate, as close as possible to the high fidelity one, and cheap to get computationally speaking. As explained in [5], this approach is based on the estimate of the so called increment function $\beta(x)$, which is defined as follows

$$\beta(x) = f_{hf}(x) - f_{lf}(x), \tag{2.1}$$

where f_{hf} and f_{lf} represent the estimates of the analyzed quantity by an high-fidelity and a low-fidelity method respectively. By this definition, $\beta(x)$ can be considered as an indicator of the error in the low fidelity estimate. This parameter is first estimated in specific points which are localized by following specific criteria[1], then it gets interpolated over the computational domain and at last it is used to correct the low-fidelity results getting something supposedly more accurate, as shown in the following equation

$$f(x) \approx f_{lf}(x) + \beta(x), \quad \forall x \in \overrightarrow{X}, \tag{2.2}$$

where $\overrightarrow{X}$ is the vector containing all the design sites. According to [5], the low-fidelity data is used to predict trends, while high fidelity data is used to provide absolute values, so that combining them together the final results turn out to be both quite accurate and cheap computationally speaking. This is the standard data fusion procedure, in this study everything is inserted in an iterative loop, so that the final results can be more accurate, but still being computationally cheap. Moreover just one model is used, so the aim of the procedure is not to improve the results coming from a poorly accurate model anymore, but to optimize the usage of the model itself. In this way the number of computation required to get the analyzed responses will be smaller than the number of element of the 3-D table the model is used for.
However, when it comes to categorize low fidelity and high fidelity models it is not possible to classify them universally, but such a classification depends on the purpose of the study one is leading. For instance, considering Datcom and Tornado [2], according to [10] and [12] when it comes to analyze the aerodynamic forces distribution over an aircraft flying in subsonic regime the Tornado results seems to be more reliable since they come from calculations evaluated on the real aircraft while Datcom gives results based on statistics. On the other hand, since Tornado is based on the Vortex Lattice Method, everything coming from Tornado at an high Mach number M is no longer reliable and its usage is limited to the subsonic regime, so for analyses in

[1] See Chapter 4 for more informations about them.
[2] See Chapter 5 for a detailed description of the models.

transonic regime Datcom plays the role of high fidelity model. The following pictures show such results for a simple case such as the *Boeing 747* with no control surfaces coming from the usage of the two models mentioned above for an angle of attack α of 6°. These responses have been obtained by using the CEASIOM package, which contained all the geometrical information regarding this aircraft, both for the Datcom estimates and for the Tornado ones

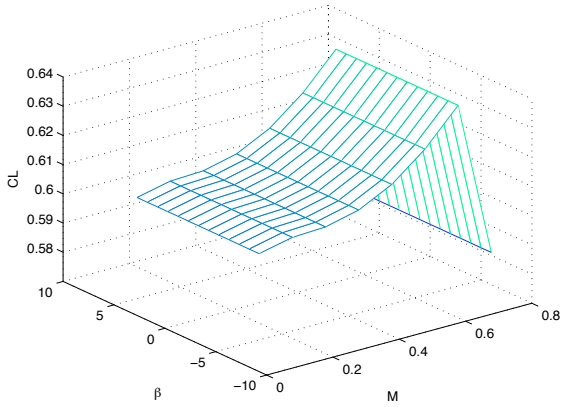

Figure 2.1: Datcom Results, $\alpha = 6°$.

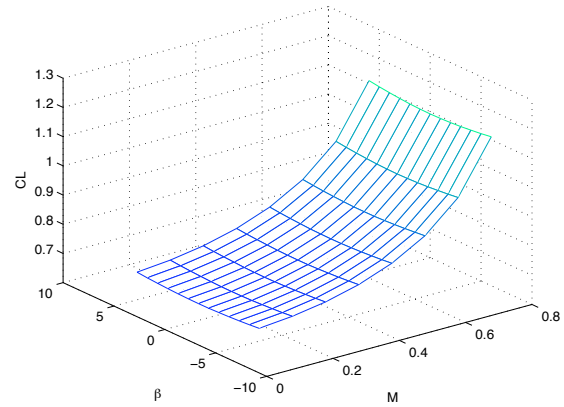

Figure 2.2: Tornado Results, $\alpha = 6°$.

The transonic Tornado results are obtained by the Prandtl-Glauert correction, and as it is possible to notice, beyond $M = 0.6$ these estimates are not reliable. Moreover, in the subsonic regime Tornado shows a slight sideslip angle dependence which is totally ignored by Datcom[3], and this is another reason to choose Tornado as an high fidelity model in this regime. Another thing that may be noticed is that the lift coefficient estimates for low Mach numbers are quite different, and between the two of them the Tornado ones are more reliable since the model works with the actual geometry of the wing, while Datcom adjusts statistical results to the analyzed case. Although Tornado does not consider some term such as the effect of the fuselage, in subsonic regime its estimates should be more accurate.
Anyway, both of them can be seen as low fidelity model if compared to the Euler Equation model, which in turn has low fidelity if compared to a DNS. In conclusion, it is not possible to classify a model univocally and a priori, but its fidelity level depends on the work conditions and on the models available to the user.

[3]Such a discrepancy is more evident without using the Prandtl-Glauert correction.

Chapter 3

Kriging

This chapter goes through the specific of Kriging, how it works and what it requires in order to work, introducing also the actually used toolbox, the DACE Kriging Toolbox.

Kriging has been characterized as "Optimal interpolation based on regression against observed values of surrounding data points, weighted according to spatial covariance values". As stated in [7], all interpolation algorithms, such as inverse distance squared, splines, radial basis functions, triangulation, etc., estimate the value at a given location as a weighted sum of data values at surrounding locations. Almost all assign weights according to functions that give a decreasing weight with increasing separation distance. Kriging assigns weights according to a user-chosen and data-driven weighting function, rather than an arbitrary function, but it is still just an interpolation algorithm and will give very similar results to others in many cases

- if the data locations are densely and uniformly distributed throughout the domain of interest, any interpolation algorithm will do;
- if the data locations fall in a few clusters with large gaps in between, the estimates will be unreliable regardless of interpolation algorithm;
- Almost all interpolation algorithms will underestimate the highs and overestimate the lows.

Kriging has few strong points

- it helps compensate for the effects of data clustering, assigning individual points within a cluster less weight than isolated data points, or treating clusters more like single points;
- it gives estimates of error - known as "kriging variance" - along with estimate of the value itself. Note, however, that the error map is basically a scaled version of a map of distance to nearest data point;
- The available estimation error provides basis for adaptive procedures for building tables, such as done in this work.

Kriging, initially developed by D. Krige and G.Matheron, originated in geostatistics and then became a widespread technique for interpolation in multidimensional, unstructured sets of points, [4]. Kriging uses the model

$$\hat{y} = F(\gamma_{:}, x) + z(x). \tag{3.1}$$

to define the value $\hat{f}$ from the *regression model* $F(\gamma, x)$ and the "residual" which is treated as a random function $z(x)$.

The regression model is given by a linear combination of p chosen functions g_j

$$F(\gamma, x) = \sum_{j=1}^{p} \gamma_j g_j(x) \tag{3.2}$$

The random function is a stochastic process which is assumed to have mean zero and covariance between $z(w)$ and $z(x)$ equal to

$$E[z(w) z(x)] = \sigma^2 R(\theta, w, x), \tag{3.3}$$

where σ^2 is the process variance of the response, and $R(\theta, w, x)$ the correlation model with parameter θ standing for the correlation function parameter.

The general definition of the correlation model is

$$R(\theta, w, x) = \prod_{j=1}^{n} R_j(\theta, w_j - x_j), \tag{3.4}$$

where n is the number of independent variable.

This correlation function parameter controls the range l around which the measurement affects its surroundings. It has to be defined per each independent variable, and the relation between θ and l is

$$\theta \propto \frac{1}{l^2} \tag{3.5}$$

The higher is the value of θ, the smaller is the size of the region in which the sampled values affect the appearance of the resulting response.
In this study the Matlab DACE kriging toolbox has been used, it can be downloaded on [16]. By using this toolbox, the increment function $\beta(x)$ introduced in Chapter 2 is estimated over the computational domain, given few exact estimates on specific points x_0. As stated in [6], the accuracy of kriging function always increases with an increase in the number of samples, although the computational time gets longer. The usage of such a package requires the user to establish both the regression and the correlation model. The choice of such terms is based on the quantities the toolbox is dealing with, and the following sections go through the detail of such choices.

3.1 Regression and Correlation Models

Usually low order polynomials are used for the regression model. Numerical problems were encountered with quadratic regression, so a linear model was chosen here,

$$g_1(x) = 1, \quad g_2(x) = x_1, \quad ..., \quad g_{n+1}(x) = x_n. \tag{3.6}$$

The choice of the correlation one depends on the analyzed phenomenon. The picture shows a few commonly used correlation function for $0 \leq d \leq 2$, where d is the distance between design sites[1]

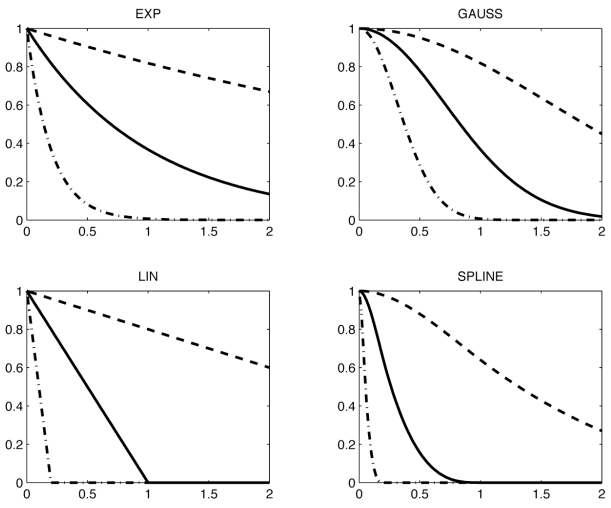

Figure 3.1: Correlation Functions.

[1]Points in the computational domain, see [3] for more explanations.

Usually for continuously differentiable f Gaussian, cubic or spline functions are chosen, since they are several times differentiable at $d = 0$, as explained in [3], and in this study the Gaussian one has been used. The following formula represents such a model

$$R(\theta, w, x) = \prod_{j=1}^{n} \exp\left(-\theta_j |w_j - x_j|^2\right) = \prod_{j=1}^{n} \exp\left[-\left(\frac{|w_j - x_j|}{l_j}\right)^2\right] \tag{3.7}$$

As stated in the previous paragraph, the correlation function parameter θ_j must be estimated for each independent variable. The choice of such a parameter determine the results of the kriging interpolations, usually linear responses need small value of this amount while nonlinear ones require larger value, meaning that the single data point affects locally the final result. However, guessing its value can be notoriously difficult, particularly using data with more than 2 independent variables. Therefore, the DACE kriging toolbox uses a maximum likelihood estimate to find the statistically best θ, once its lower and upper limits are defined. It basically determines θ^* that solves

$$\min_{\theta} \{\psi(\theta) \equiv |R|^{\frac{1}{m}} \sigma^2\} \tag{3.8}$$

Optimizing this likelihood function, however, is a very hard problem, since the region around the minimum is very flat, causing problems for any optimization algorithm. The following figure shows a typical trend of the function previously stated

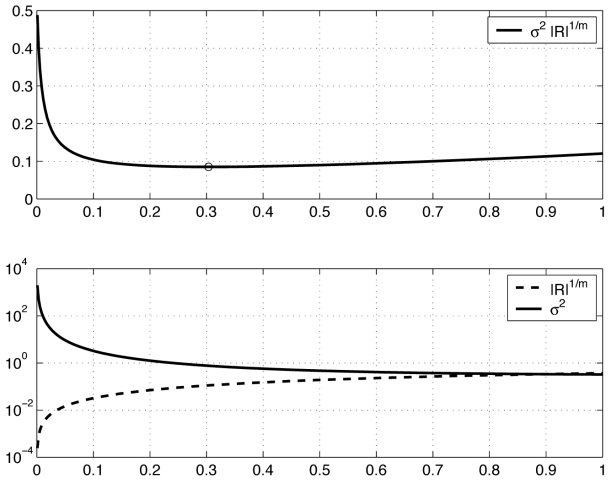

Figure 3.2: Correlation Function Parameter: Estimate.

Given the flatness of the graph, in order to define the correlation model that

suits the analyzed responses the best it is not necessary to estimate θ^* precisely, any value of the correlation function parameter within a small range around the statistically best θ would give an accurate model. For this reason the used toolbox does not really estimate the minimum of $\psi(\theta)$, but it stops after at most 4 iterative steps giving a result that is a good compromise between efficiency and desired accuracy, as stated in [2]. Moreover, the regression function is affected by the number of used data points. The larger is such a number the greater is the degree of local information available, and consequently the estimate of the correlation function parameter changes depending on it.
The lower limit for the estimate of this parameter is usually set as close to zero as possible[2], the upper one depends on the linearity of the analyzed response as previously stated.

[2]Given its definition it can not be exactly equal to zero.

Chapter 4

Procedure Features

This chapter goes through the DFIP specifics focusing on the sampling criteria, the criterion used to stop the iterative procedure, and the user interface.

Applying the concepts described in the previous chapter, an optimizing procedure has been developed, which basically works following these steps

- it loads the initial data set input by the user obtained by using a certain model, which collects few results;
- it defines the final computational domain with the dimensions input by the user;
- it samples few points on which the computational model will be applied again;
- it fuses the previous data set with the estimates on the sampling points by using interpolation algoriths;
- it repeats the last two points iteratively until the stop criterion is satisfied.

The core of the program is basically given by the choice of proper sampling criteria. In fact once these criteria get properly established it does not matter what kind model the program is supplied with, and that makes the usage of the program itself more flexible, giving the possibility of using it, or at least part of it, even in other engineering fields.

4.1 Sampling Criteria

As stated in the introduction of this chapter, the choice of the right points where the high fidelity model must be applied on is a very important step. There are several elements that must be taken into account in order to select the best candidates as such points. Some of them are more important than others, but still it is necessary to consider all of them, their relative importance can be defined by setting the right priority for their execution. Those terms are the increment function $\beta(x)$, the Mean Squared Error MSE, the geometrical feature of the surface that describes the distribution of the estimated coefficients *geom* and at last the density of the sampling point in the computational domain *void*. Once the total number of sampling points N is fixed, it can be considered as equal to the summation of sampling points coming from the listed elements, as shown in the following equation

$$N = N_{\beta(x)} + N_{MSE} + N_{geom} + N_{void}. \tag{4.1}$$

They have been listed in their priority order. As it is possible to notice, the increment function term and the mean squared error one are the ones with the highest priority. That is due to the fact that they represent how much the current distribution of the analyzed quantities fails to represent the actual one. As stated in Chapter 2, $\beta(x)$ can be seen as an expression of the error in the estimate of a certain quantity due to the usage of models with different accuracy level, and since this kind of error seems to be larger than the one due to the usage of the kriging toolbox, it has higher priority.

4.1.1 Increment Function $\beta(x)$

As stated in Chapter 2, from the data fusion procedure it is possible to estimate $\beta(x)$, which in this case represents the difference between the results at a certain step and the those ones associated to the previous step. The value of such a term is supposed to decrease in the iterative loop, converging to zero. So, once the loop is over, the distribution of $\beta(x)$ for a two sweeping variables case should be a flat plate. First the program tries to get all of the sampling points from the $\beta(x)$ analysis, so that

$$N_{\beta(x)} = N. \tag{4.2}$$

For each force coefficient f_i it checks the $\beta(x)$ distribution, identifies the maximum/minimum points for such a distribution, which supposedly represent the worst estimates in the current iterative step, and uses them as sampling points. That means that

$$N_{\beta(x),i} = \frac{N_{\beta(x)}}{n} = \left\{ x \mid \frac{\partial \beta(x)}{\partial x_j} = 0 \right\}, \ \forall f_i \tag{4.3}$$

where n represents the number of analyzed quantities. Since it is not possible to say a priori which force coefficient requires a more thorough analysis, the available number of sampling points has been equally split among the studied responses. Once these points are gotten, they are collected all together defining a unique matrix of sampling points. Since such points are obtained analyzing individually each non-dimensional coefficient, it might happen that several coefficients share same sampling points, and that could generate some problem in the data fusion procedure[1]. In order to avoid such a problem, the program checks whether there are some double point; and in that case it eliminates the extra points. As a consequence of that, at the end of such an analysis there could be two different cases

1. $N_{\beta(x)} < N$: the program goes on with the next step in the sampling procedure;

2. $N_{\beta(x)} \geq N$: the program stops the sampling procedure taking the first N sampling points.

An important thing to highlight is that this step occurs anytime $\beta(x)$ is available, that means that before the iterative loop occurs it is skipped.
The following picture shows an example of sampling following the criteria mentioned above for a two sweeping variables case

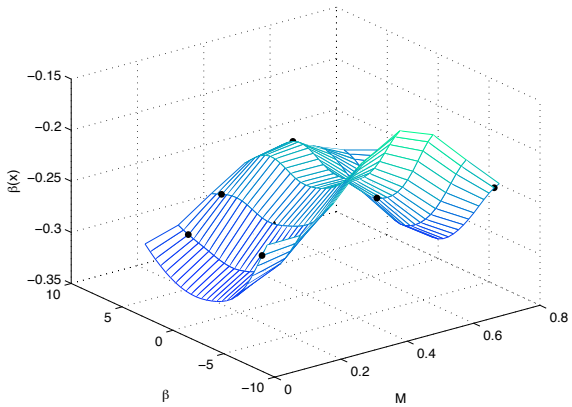

Figure 4.1: Increment Function Sampling.

[1]The DACE Toolbox does not allow multiple design sites.

4.1.2 Mean Standard Error MSE

As stated in [3], the approximation models used to estimate the analyzed variables are deterministic, so they lack of random error and the only one they are affected is MSE, which is related to the used correlation models. The larger MSE in a certain point, the worse the interpolation in that point, so this factor must be taken into account. Once the $\beta(x)$ step is over, the program goes on and gives half of the remaining sampling points to the MSE analysis, getting

$$N_{MSE} = \frac{N - N_{\beta(x)}}{2}. \tag{4.4}$$

$$N_{MSE} = \{x \mid MSE(x) = MSE_{max}\}. \tag{4.5}$$

In this case the available number of sampling points is not split among the several force coefficients, since all of them are estimated over the computational domain by using the same correlation model, so the MSE distribution is the same as well. The following picture shows an example of sampling following the criteria mentioned above for a two sweeping variables case

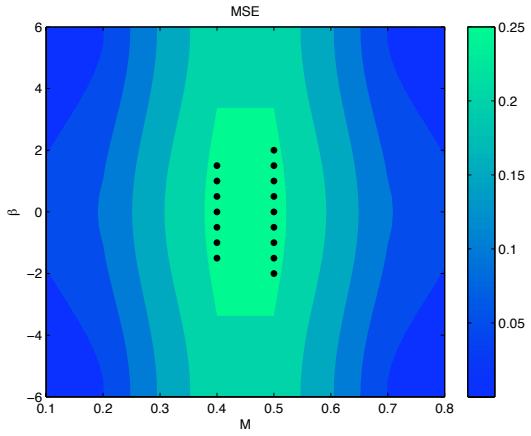

Figure 4.2: Mean Standard Error Sampling.

4.1.3 Geometrical Features *geom*

Another important element to take into account is the shape of the surface describing the distribution of the estimated variables. According to the analytical geometry bases, to sketch a generic function maximum, minimum, inflection and border points have to be identified. That would give a general idea of the function trend. However, several tests showed that while the border points analysis is quite effective, the maximum/minimum and inflection

points analyses are not[2], slowing the entire procedure down. For this reason only the border points has been considered, reducing the sampling procedure as follows

$$N_{geom} = N - N_{\beta(x)} - N_{MSE} \tag{4.6}$$

$$N_{geom} = \{x \mid x \in \partial\Omega\}. \tag{4.7}$$

In the equation above $\partial\Omega$ stands for the border of the surface in a two sweeping variables case, analyzed side by side. In a three sweeping variables case it would represent the external surfaces described by the conditions $\alpha = \alpha_1$, $\alpha = \alpha_{end}$, $M = M_1$, $M = M_{end}$, $\beta = \beta_1$ and $\beta = \beta_{end}$.
The picture below shows an example of the sampling criteria mentioned above for a two sweeping variables case

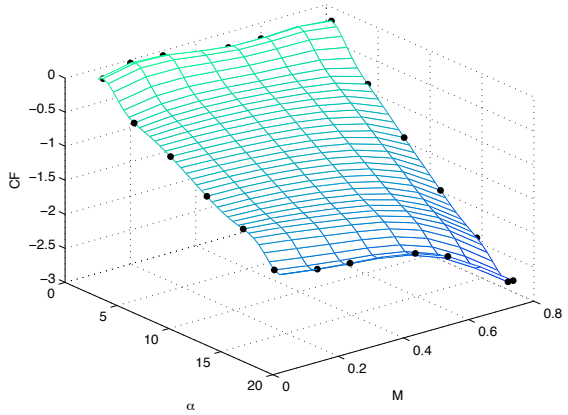

Figure 4.3: Geometrical Sampling.

After the double-points check, the final result is

$$N_{geom} \leq N - N_{\beta(x)} - N_{MSE} \tag{4.8}$$

In this case the double-points analysis checks whether the external surfaces share common sampling points or not. If there are some point left, the procedure goes through the last sampling criterion.

4.1.4 Sampling Point Density *void*

At last, assuming that there are some point left it could be useful to check those spots in the computational domain where there are no sampling points and to fill them with the remaining points. In order to do that, the program

[2]That is due to the fact that they are applied on local results in the iterative loop.

identifies the largest empty sphere in the computational domain and then it locates one point in its center. Such a step is repeated as many times as the number of left points is. The following equations recap such a step

$$N_{void} = N - N_{\beta(x)} - N_{MSE} - N_{geom} \tag{4.9}$$

$$N_{void} = \{x \mid \text{sphere} = \emptyset \ \wedge \ r = \|x - x_{sampled}\| = r_{max}\}, \tag{4.10}$$

where r is the radius of the sphere, and $x_{sampled}$ is one of the sampling points already gotten. Also in this case it is not necessary to check for the double-points.
The following picture shows an example of sampling following the criteria mentioned above

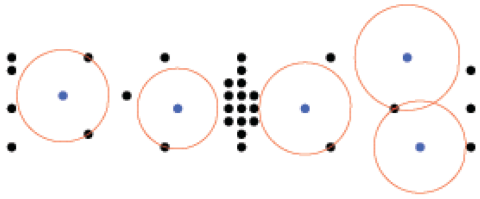

Figure 4.4: Sampling Point Density Sampling.

In conclusion this criterion basically does the same thing the MSE criterion does, that is filling up the empty spaces in the computational domain[3]. The only difference is that while the MSE criterion somehow takes the distribution of the analyzed variables into account, the *void* criterion considers only the distribution of the sampling points. That means that theoretically well estimated areas but poor of sampling points would be filled up by this criterion.

4.1.5 Statistical Results

This section collects some statistic regarding the usage of the criteria listed above, in order to better understand how the final results are influenced by their usage.

The number of sampling points coming from the $\beta(x)$ analysis is always smaller than the maximum number of sampling points established by the

[3]High mean squared errors are due to a lack of points in that area.

user, and usually leaves a quite large number of points for the next analyses. That is probably due to the good quality of the results coming from the used kriging toolbox.

The number of sampling points coming from the MSE analysis is almost as big as the number of points allowed to this criterion, which is largely conditioned by the number of sampling points left from the $\beta(x)$ analysis. That means that the double-point analysis does not find so many overlapping points, and probably the MSE distribution for each force coefficient is different.

The geometrical criterion usually uses almost all the remaining points, leaving to the *void* analysis few sampling points.

The following table collects the percentages of occurring of the listed criteria, recapping what stated in the paragraphs above, allowing 50 sampling points per each iterative step

$\beta(x)$	MSE	*geom*	*void*
23.2%	34%	39.6%	3.2%

Table 4.1: Criteria Statistical Results.

4.2 Stop Criterion

Since the procedure runs iteratively, it needs a criterion to stop itself at a certain point. This criterion is based on checking the average value of the increment function expressed as a percentage of an indicative maximum value of the corresponding force coefficient $C_{F,i}$ in the current iterative step, $\bar{\beta}_{i,\%}$. As stated in Chapter 2, such a parameter represents the error between one solution and the previous one estimated in the iterative loop. So, $\bar{\beta}_{i,\%}$ is defined as follows

$$\bar{\beta}_{i,\%} = \frac{\bar{\beta}_i}{\max(f_i)} \cdot 100, \ \forall f_i, \tag{4.11}$$

where $\bar{\beta}_i$ is the increment function $\beta(x)$ averaged over the whole computational domain associated to f_i. Calling m the number of element in the computational domain

$$\bar{\beta}_i = \frac{\sum_{i=m}^{m} \beta_i(x)}{m}, \ \forall f_i. \tag{4.12}$$

Then $\bar{\beta}_{i,\%}$ is compared to a relative tolerance factor *tol* chosen by the user, so that the user himself can set the program according to his needs.

Moreover, since the procedure is dealing with variables depending on three sweeping amounts, such variables would be portrayed by 4D surfaces. When the number of sampling points increases, the 4D surface coming from the usage of the kriging toolbox could be either very close to the previous estimate or totally different, and there is no way to say which one of the two cases occurs a priori, so it may happen that $\bar{\beta}_{i,\%}$ satisfies the tolerance factor by a lucky strike. In order to counter such a problem, the stop criterion must be satisfied three times in a row. After that it is reasonable to assume that the convergence is reached, although that means adding some computation that anyway refines the final results. That also means that the minimum number of iterative steps is equal to 3.

4.3 User Interface

In order to test such a procedure on some aeronautical case, it has been coded in Matlab, creating a program that applies the procedure itself on Tornado, one of the aeronautical simulator introduced in Chapter 1. It has been supplied with an user interface which resembles the Tornado one. This choice is not related to any low fidelity/high fidelity consideration, but on the fact that a multiple choices interface would simplify the usage of the program itself. Moreover, showing the several possibilities directly on the command windows does not requires further m-files that would code the possible GUI interface, without compromising its ease of use. Among the such choices, the program offers the possibility to choose both the initial data set and the geometrical features once they are collected in their proper folders, so it can be used to estimate the aerodynamic properties of another aircraft without directly modifying the code. It asks the user to input the dimension of the final data set, the number of the sampling points to use in each iterative step, the correlation lengths for each independent variable, the aircraft model, whether to use the Prandtl-Glauert correction or not in the Tornado estimates and at last the relative tolerance factor. Regarding the dimension of the final data set, it is limited to at most 10000 results per force coefficient. Every time the kriging toolbox has to interpolate/extrapolate over that limit, Matlab runs out of RAM memory and the program stops. For this reason the dimension of the final data set must be set in order to get a manageable computational domain.

Once its sequence is activated the program displays the number of sampling points used in each iterative step and how they are distributed in the four points described in Section 4.1. Once the procedure is over, the program recaps the total number of sampling points and how they are distributed over the several criteria, and it gives the user the possibility to display some graph representing the estimates responses.
The following figure portrays such an interface

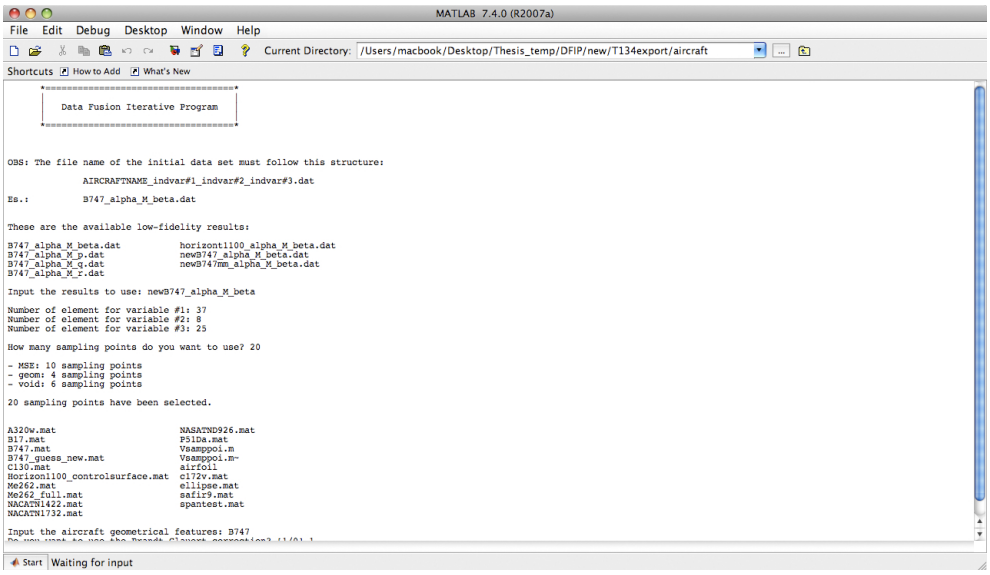

Figure 4.5: Matlab User Interface.

Chapter 5

Test Case

This chapter shows the results gotten by using the iterative procedure and analyzes them changing some parameter such as the number of sampling points per iterative step and comparing them with the actual results.

In order to show the quality of what comes from the usage of such a procedure, it has been tested first on a simple case, such as a *Boeing 747* with no control surfaces, then a hypothetical case which was created in order to challenge the procedure itself[1]. The initial dataset is quite poor, just 8 values in the first test case and 4 values in the second one. By using this iterative procedure the purpose is to generate a larger and more accurate dataset limiting the computational costs. In order to show such results by 3D surfaces, one of the three sweeping variables have been fixed one at the time, first the angle of attack α at 13°, then the Mach number M at 0.3 and at last the sideslip angle β at 6°[2]. The used aerodynamic model was Tornado, which has been set with a fixed wake and the Prandtl-Glauert correction has been used on the final results, although one has to remember that it works in subsonic regime, so any result related to a Mach number larger than 0.6 is not reliable.

[1] Those responses

[2] These are just example parameters, they do not have a particular meaning.

5.1 Results

5.1.1 Boeing 747

All these results shown below have been gotten by using a relative tolerance factor $tol = 1\%$ and applying the Prandtl-Glauert correction. The force distributions for this case are quite linear, so the correlation function parameters are defined assuming that each data point affects the whole response. That means that the lengths that define such a parameter as stated in Chapter 3 have to be at least as large as the range of each independent variable. The following table collects these values

l_1	l_2	l_3
18	0.7	12

Table 5.1: Correlation Lengths.

These amounts have been used to define the upper limits of θ, then the DACE kriging toolbox calculates its statistically best estimate.
The accuracy of the final estimates should not depend on the number of sampling points per iterative step, assuming a proper choice of the correlation lengths. To verify that, the following figure sketches the trend of the averaged difference between the estimated results and the exact ones[3] expressed as a percentage of an indicative value of the maximum force coefficients $e_{\%,i}$

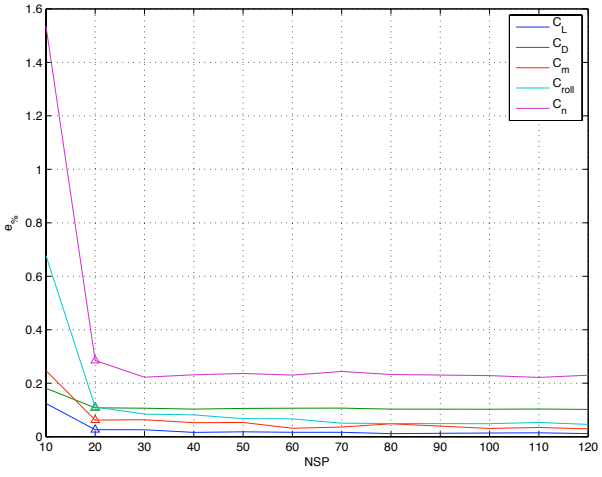

Figure 5.1: $e_{\%}$ Trend.

[3]Results that come from the standard usage of the computational model on the established domain.

As it is possible to notice, the error assumes an unexpected values for $NSP = 10$, but it is about the same for all the other cases. A reasonable explanation for this behavior at $NSP = 10$ is that since the program adds few sampling points in each iterative step, the changes in the results in each step may be not considerable, and consequently the stop criterion is satisfied even though the final response is not highly accurate. However, even in this case the largest error is of about 1.6%, so still quite reliable. Anyway, since this behavior can be expected a good rule of thumb is to use a number of sampling points per iterative step which let to use at least the first three sampling criteria stated in Chater 4.1, in this case $NSP \geq 20$.
The presence of bumps on the different curves are due to the chosen correlation lengths for that specific case, but in general it is possible to observe that the accuracy of the results is quite the same. The following table collects some of the values of $e_{\%}$

		Force Coefficients				
		C_L	C_D	C_m	C_{roll}	C_n
	20	0.03	0.11	0.06	0.11	0.28
	30	0.03	0.11	0.06	0.08	0.22
	40	0.02	0.10	0.05	0.08	0.23
NPS	**50**	0.02	0.11	0.05	0.07	0.24
	60	0.02	0.11	0.03	0.07	0.23
	70	0.02	0.11	0.04	0.05	0.24
	80	0.01	0.10	0.05	0.05	0.23

Table 5.2: Error Percentage on Final Result.

The number of sampling points NSP must be chosen in order to be sure that the final results converge to an accurate estimate in short computational time. The following figure, sketching how the number of iterative steps NIS changes with NSP and how the total number of sampling points $TNSP$ changes with NSP, can better explain that phenomenon. The triangles identify $NSP = 20$, and the red lines represent the average trends of these parameters. As it is possible to notice, $TNSP$ tends to increase with NSP as it was expected, while NIS tends to decrease, meaning that the regression model gets more accurate. The values of these two parameters at a specific value of NSP are related to the way the used kriging toolbox works, so it can not be estimated a priori and it should not surprise that the actual trends do not go directly from high values to lower one or vice versa as NSP increases

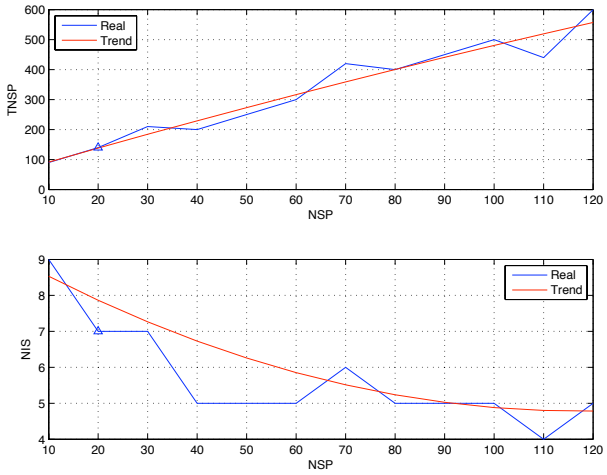

Figure 5.2: $TNSP$ and NIS Trends.

The following graphs sketches some of the results coming from the usage of the procedure at the attitude stated at the beginning of this chapter and setting the number of sampling points per iterative step as equal to 20. Each one of them is followed by the contour plot of the difference between the estimated results and the exact ones, so that one can better appreciate their quality. On the contour plots the sampling points that influenced the corresponding distribution are sketched as well. These specific graphs have been chosen because these results are quite sensible to the choice of the correlation model, so by representing them one can appreciate even better the quality of the final results.

Since the total number of sampling point is quite small in this case, the representation of the used sampling points does not help too much understanding how such results have been gotten; but generally speaking that could show the critical areas on which the program focused in order to get the final responses.

The total number of sampling points used to get a database of 7400 elements is equal to 140, which means that the program saved about the $98,1\%$ of the computations to get them.

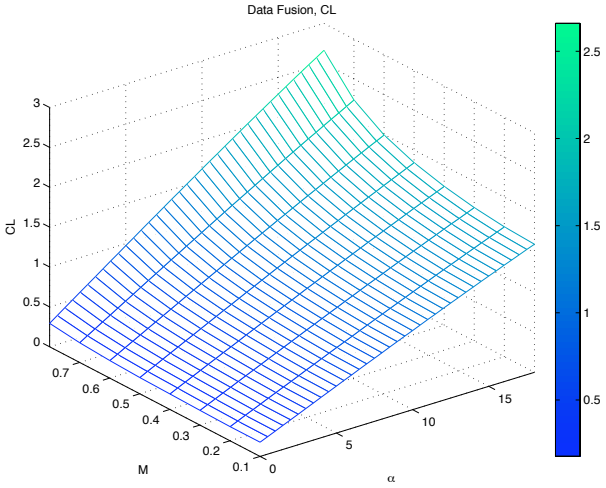

Figure 5.3: Data Fusion, C_L, $\beta = 6°$, $tol = 1$.

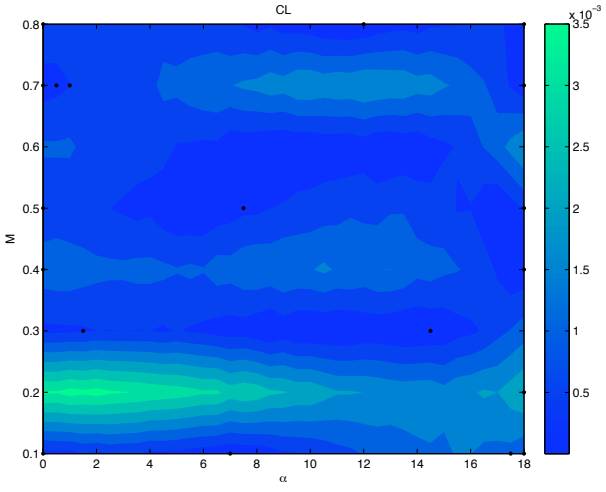

Figure 5.4: Error Distribution, C_L, $\beta = 6°$, $tol = 1$.

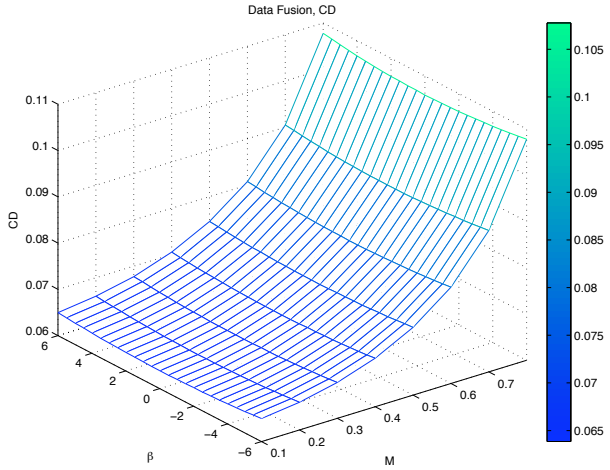

Figure 5.5: Data Fusion, C_D, $\alpha = 13°$, $tol = 1$.

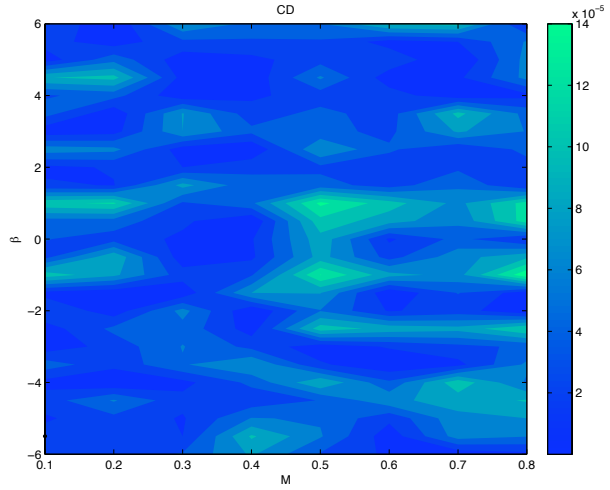

Figure 5.6: Error Distribution, C_D, $\alpha = 13°$, $tol = 1$.

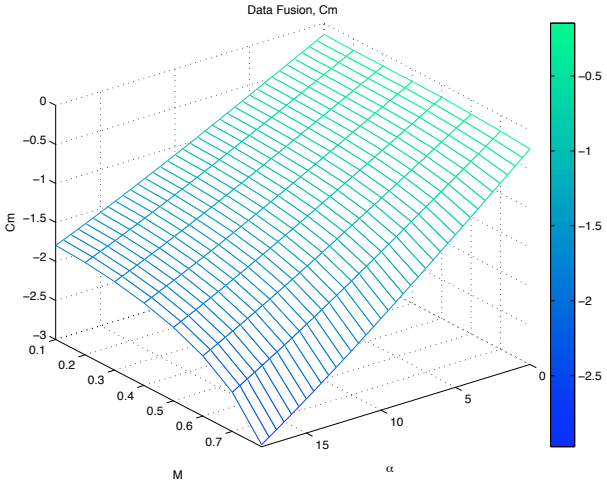

Figure 5.7: Data Fusion, C_m, $\beta = 6°$, $tol = 1$.

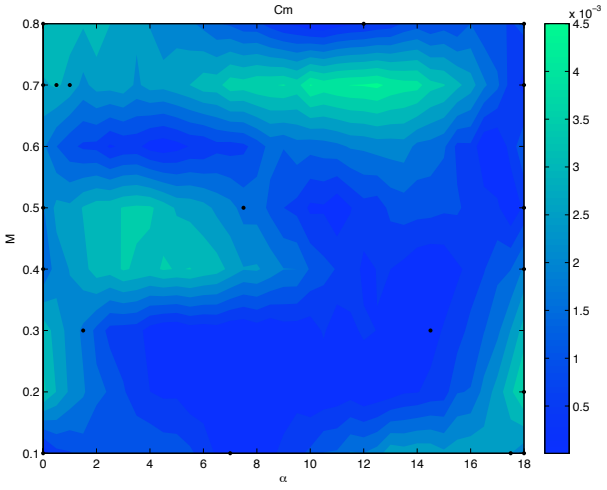

Figure 5.8: Error Distribution, C_m, $\beta = 6°$, $tol = 1$.

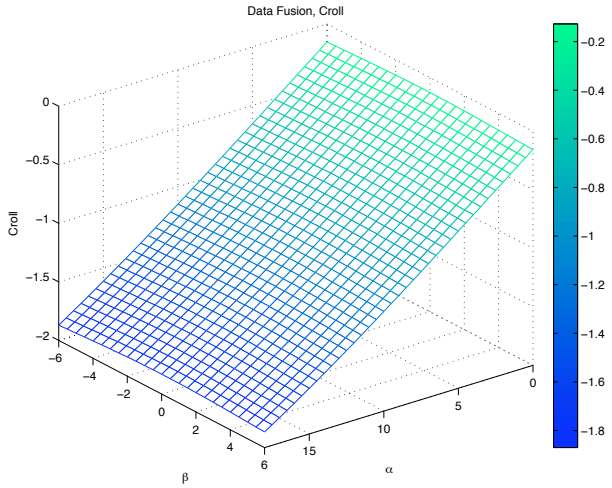

Figure 5.9: Data Fusion, C_{roll}, $M = 0.3$, $tol = 1$.

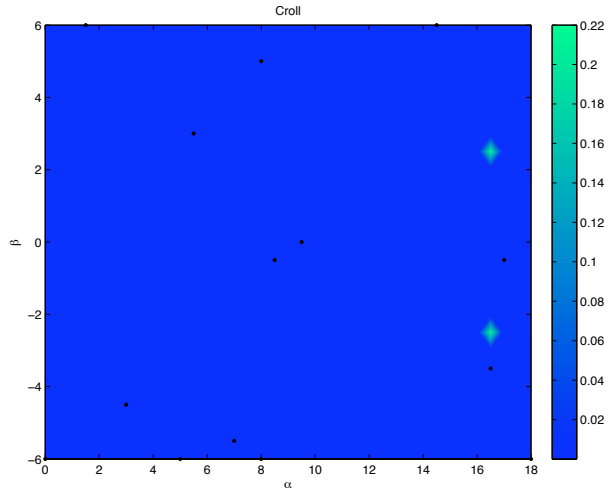

Figure 5.10: Error Distribution, C_{roll}, $M = 0.3$, $tol = 1$.

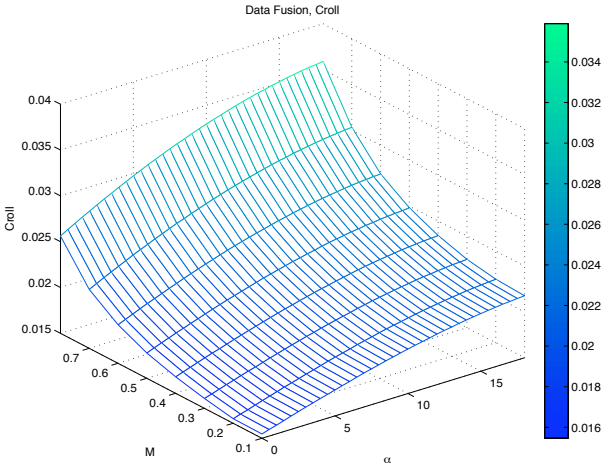

Figure 5.11: Data Fusion, C_{roll}, $\beta = 6°$, $tol = 1$.

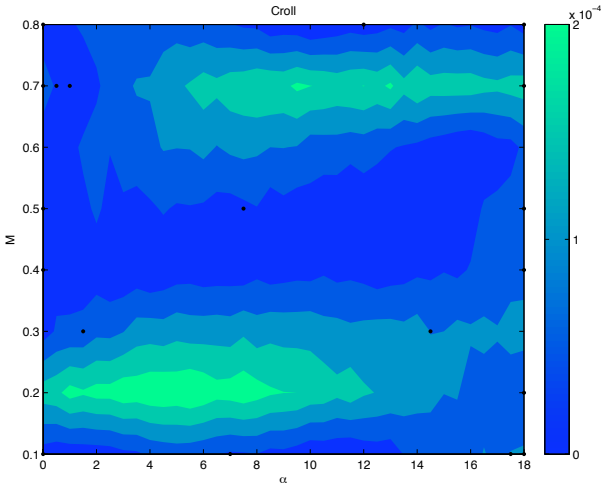

Figure 5.12: Error Distribution, C_{roll}, $\beta = 6°$, $tol = 1$.

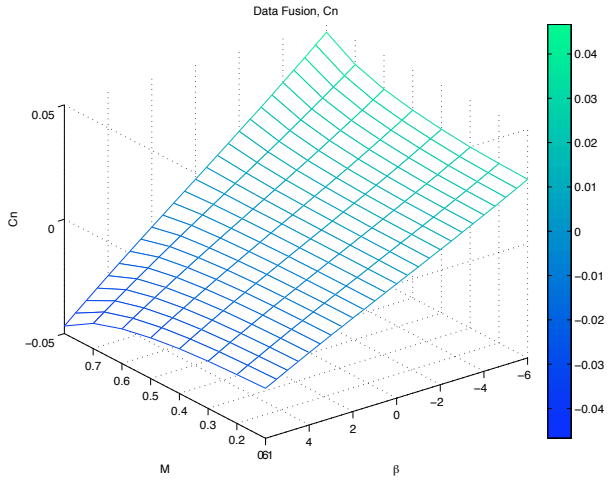

Figure 5.13: Data Fusion, C_n, $\alpha = 13°$, $tol = 1$.

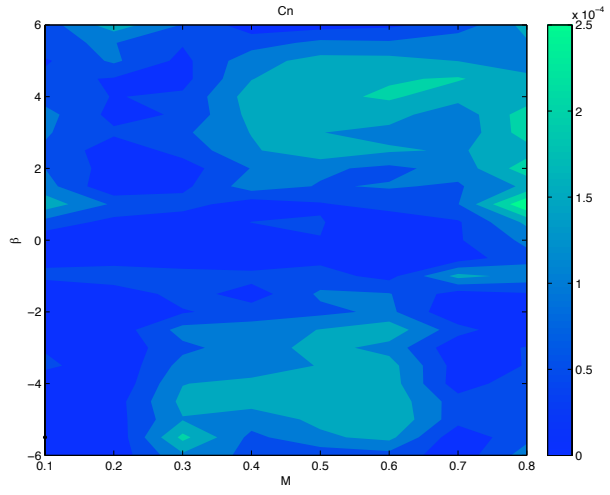

Figure 5.14: Error Distribution, C_n, $\alpha = 13°$, $tol = 1$.

5.1.2 Hypothetical Case

In order to verify the quality of the results coming from the application of this procedure, it has been tested on the two nonlinear responses described by the following equations

$$w(x,y) = y\sin(x) - \cos(y)\cos(x).^3 - 2\sin(n(y))\sin(m(x)) - e^{-p(x)} \quad (5.1)$$

$$z(x,y) = \sin(y)^3\sin(x)^2 + \sin(y)\cos(x) + e^{-q(y)} \quad (5.2)$$

$$m(x) = \frac{2x+\pi}{4} \qquad n(y) = \frac{2y+\pi}{4}$$

$$p(x) = \frac{2x+\pi}{\pi} \qquad q(y) = \frac{2y+\pi}{\pi}$$

$$x, y \in \left[-\frac{\pi}{2}, \frac{3}{2}\pi\right]. \quad (5.3)$$

The following figures sketches such distributions

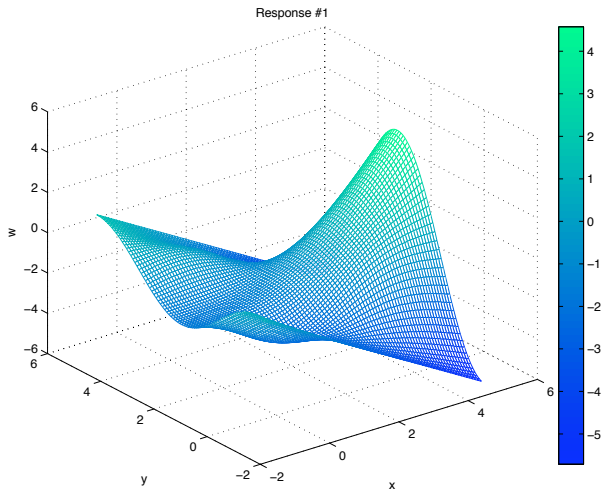

Figure 5.15: Second Test Case - 1.

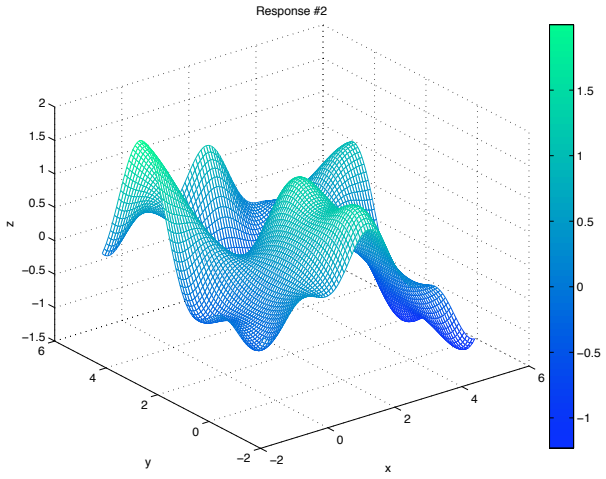

Figure 5.16: Second Test Case - 2.

These are the results obtained by using the iterative procedure on this new dataset. They have been obtained by using $NSP = 20$ and a relative tolerance factor of 1%. Given the non-linearity of these responses, the correlation function parameters have been defined so that each data point affects the final results just locally. Their upper limits have been set based on an affected range equal to 1/5 of the independent variables ranges for the first response and 1/10 for the second one[4].

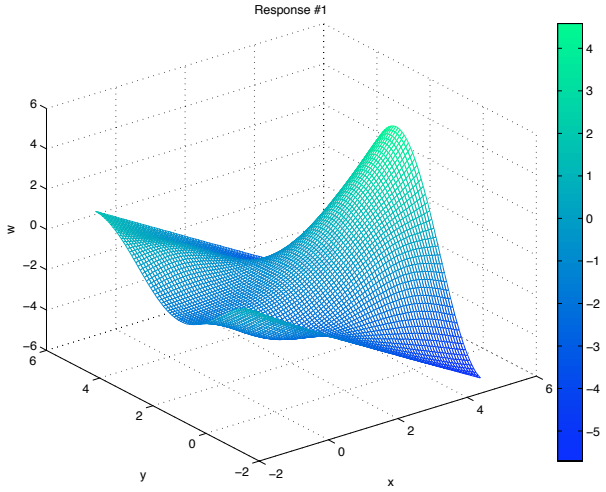

Figure 5.17: Data Fusion Results - 1.

[4]Since the second response is more nonlinear than the first one its correlation function parameters have to be larger.

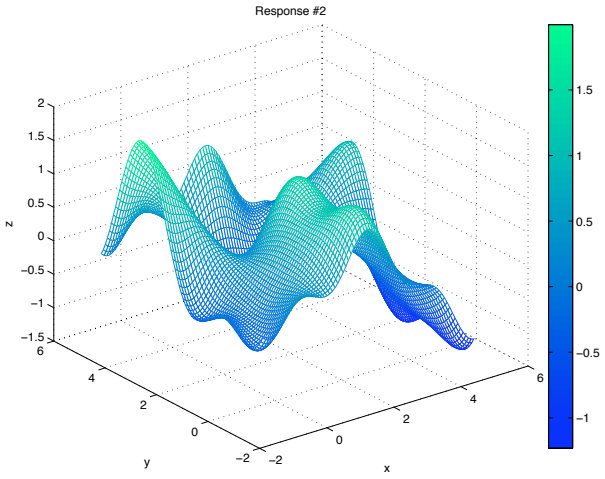

Figure 5.18: Data Fusion Results - 2.

As it is possible to see those results are quite alike to the expected ones, meaning that the program seems to work properly. To better appreciate the quality of these results, the following graphs sketch the difference between the estimated results and the exact ones, visualizing also the used sampling points to simplify the analysis

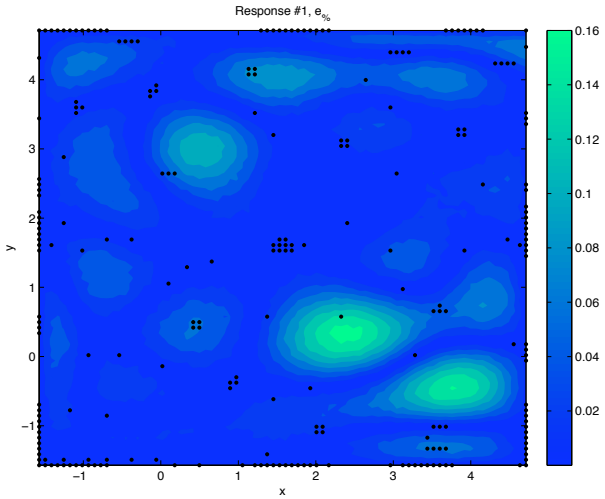

Figure 5.19: Error Distribution - 1.

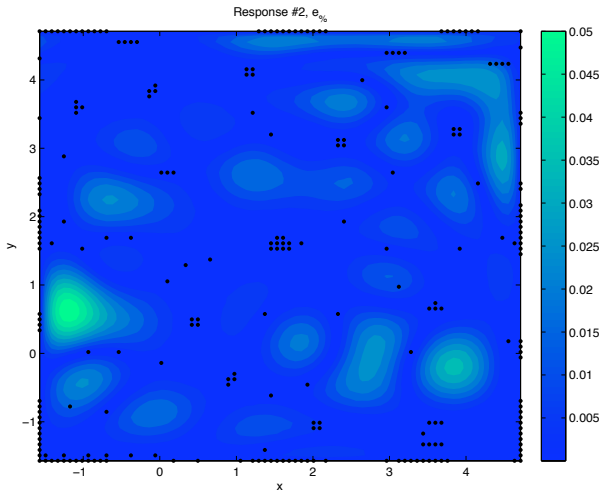

Figure 5.20: Error Distribution - 2.

It is possible to notice how some sampling point seems to follow a sort of pattern. This is due to the application of the sampling criteria, meaning that those were probably the most critical areas for the program to get the final results, so it filled them with data points in order to fix the problem.
Going through the statistical results, the following figure sketches the trend of the percentage errors $e_{\%}$

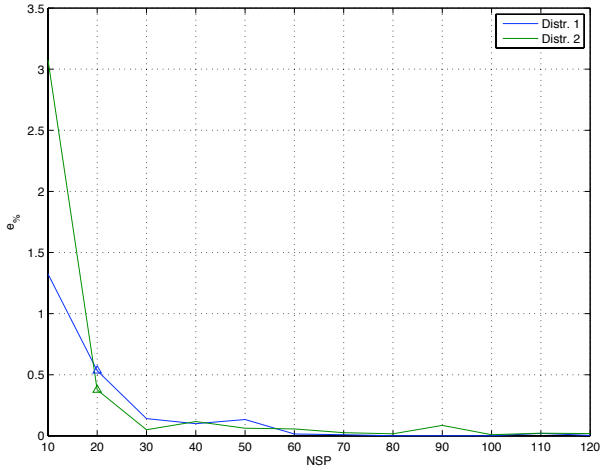

Figure 5.21: $e_{\%}$ Trend, Test Case 2.

Also in this case it is possible to observe that for $NSP \geq 20$ the percentage errors are about constant for the two responses, as it was expected, showing the same behavior noticed in the first test case. The following table collects

some result to analyze it in detail

		Responses	
		#1	#2
	20	0.53	0.38
	30	0.14	0.05
	40	0.10	0.12
NSP	**50**	0.13	0.06
	60	0.02	0.06
	70	0.01	0.03
	80	0.00	0.02

Table 5.3: Error Percentage on Final Result, Test Case 2.

The same tendency noticed in the first test case can be noted also for the $TNSP$ and NIS, as skeyched in the following figure

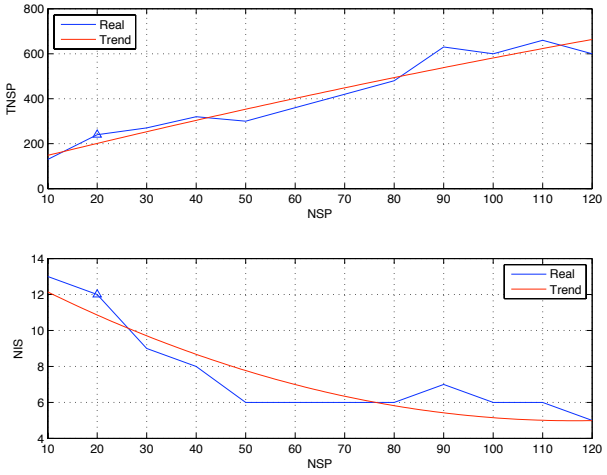

Figure 5.22: $TNSP$ and NIS Trends, Test Case 2.

Also in this case the results show that as the number of sampling points increases, the number of iterative step decreases and the total number of sampling points increases as well.
In the test case NSP is equal to 20 giving a total number of sampling points equal to 240 to generate a database of 6400 responses, saving about the $96,25\%$ of the computations.

Chapter 6

Conclusion

In conclusion, the procedure seems to work properly. The final results are satisfying and saving a large number of computations. As stated in Chapter 4, the core of the procedure is given by the sampling function and the correcting steps, then it is just supplied with the computational model whose usage has to be optimized, and with a kriging package which not necessarily has to be the DACE toolbox. Generally speaking, the model itself could be any model,and since the procedure has been developed to work with generic data set, one could also think to use it with models that are not necessarily related to aerodynamics. The sampling function would remain intact as long as the user is working with at most three independent variables data set.
The only limit in this analysis is given by the fact that the used kriging toolbox requires to know the regularity of the final responses in order to define a proper upper limit for the correlation function parameter, but this is just related to the specific package that as been used in this study.

Bibliography

[1] M. Ghoreyski, K. J. Badcock, M. A. Woodgate, *"Accelerating the Numerical Generation of Aerodynamic Models for Flight Simulation"*, University of Liverpool, Liverpool, England L69 3GH, United Kingdom.

[2] S. N. Lophaven, H. B. Nielsen, J. Søndergaard, *"Aspects of the Matlab Toolbox DACE"*, IMM Technical University of Denmark, 2002.

[3] R. Von Kaenel, A. Rizzi, J. Oppelstrup, T. Goetzendorf-Grabowski, M. Ghoreyshi, L. Cavagna, A. Bérard, *"CEASIOM: Simulating Stability & Control with CFD/CSM in Aircraft Conceptual Design"*, ICAS 2008.

[4] S. N. Lophaven, H. B. Nielsen, J. Søndergaar, *"DACE A Matlab Kriging Toolbox"*, IMM Technical University of Denmark, 2002.

[5] C. Y. Tang, K. Gee, S. L. Lawrence, *"Generation of Aerodynamic Data using a Design of Experiment and Data Fusion Approach"*, AIAA, 2005.

[6] M. Ghoreyshi, K. J. Badcock, M. A. Woodgate, *"Integration of Multi-Fidelity Methods for Generating an Aerodynamic Model for Flight Simulation"*, University of Liverpool, Liverpool, England L69 3GH, United Kingdom.

[7] G. Bohling, *"Kriging"*, Kansas Geological Survey.

[8] J. P. C. Kleijnen, *"Kriging Metamodelling in Simulation A Review"*, Tilburg University.

[9] M. Zagoraiou, A. B. Antognini, *"On the Optimal Design for Gaussian Ordinary Kriging with Exponential Correlation Structure"*, Universitá di Bologna.

[10] J. Lee, J. H. Kwon, *"On the Use of Kriging in the Interpolation of Fluid-Structure Interaction Analysis"*, Computational Fluid Dynamics JOURNAL.

[11] *"The Usaf Stability And Control Datcom"*, Public Domain Aeronautical Software, December 1999.

[12] F. Jurecka, M. Ganser, K. U. Bletzinger, *"Update scheme for sequential spatial correlation approximations in robust design optimisation"*, ScienceDirect.

[13] T. Melin, *"Users guide and reference manual for Tornado 1.0"*, Royal Institute of Technology (KTH), Department of aeronautics, 2000-12.

[14] *http://www.holycows.net*.

[15] *http://www.redhammer.se*.

[16] *http://www.wikipedia.org*.

[17] *http://www2.imm.dtu.dk*.