UNIVERSIDAD POLITÉCNICA DE MADRID

Escuela Técnica Superior de Ingenieros Aeronáuticos
Departamento de Vehículos Aeroespaciales

DOCTORAL THESIS

**Aerodynamic database error
filtering via high order singular
value decomposition**

Author:
Artur Andrzej
JARZABEK

Supervisors:
Dr. José Manuel
VEGA DE PRADA

Dr. José Manuel
PERALES PERALES

2016

UNIVERSIDAD POLITÉCNICA DE MADRID

Abstract

Escuela Técnica Superior de Ingenieros Aeronáuticos

Departamento de Vehículos Aeroespaciales

Doctor of Philosophy

Aerodynamic database error filtering via high order singular value decomposition

by Artur Andrzej JARZABEK

A method is presented to filter errors out in multidimensional databases. The method does not require any a priori information about the nature the errors. In particular, the errors need not to be small, neither random, nor exhibit zero mean. Instead, they are only required to be relatively uncorrelated to the clean information contained in the database. The method is based on an improved extension of a seminal iterative gappy reconstruction method (able to reconstruct lost information at known positions in the database) due to Everson and Sirovich (1995).

The improved gappy reconstruction method is evolved as an error filtering method in two steps, since it is adapted to first (a) identify the error locations in the database and then (b) reconstruct the information in these locations by treating the associated data as gappy data. The resulting method filters out $O(1)$ errors in an efficient fashion, both when these are random and when they are systematic, and also both when they concentrated and when they are spread along the database. The performance of the method is first illustrated using a two-dimensional toy model database resulting from discretizing a transcendental function and then tested on two CFD-calculated, three-dimensional aerodynamic databases containing the pressure coefficient on the surface of a wing for varying values of the angle of attack.

A more general performance analysis of the method is presented with the intention of quantifying the randomness factor the method admits maintaining a correct performance and secondly, quantifying the size of error the method can detect. Lastly, some improvements of the method are proposed with their respective verification.

UNIVERSIDAD POLITÉCNICA DE MADRID

Resumen

Escuela Técnica Superior de Ingenieros Aeronáuticos

Departamento de Vehículos Aeroespaciales

Doctorado

Filtrado de errores en bases de datos aerodinámicas a través de la descomposición en valores singulares para tensores de alto orden

por Artur Andrzej JARZABEK

Esta Tesis presenta un nuevo método para filtrar errores en bases de datos multidimensionales. Este método no precisa ninguna información a priori sobre la naturaleza de los errores. En concreto, los errores no deben ser necesariamente pequeños, ni de distribución aleatoria ni tener media cero. El único requerimiento es que no estén correlados con la información limpia propia de la base de datos.

Este nuevo método se basa en una extensión mejorada del método básico de reconstrucción de huecos (capaz de reconstruir la información que falta de una base de datos multidimensional en posiciones conocidas) inventado por Everson y Sirovich (1995).

El método de reconstrucción de huecos mejorado ha evolucionado como un método de filtrado de errores de dos pasos: en primer lugar, (a) identifica las posiciones en la base de datos afectadas por los errores y después, (b) reconstruye la información en dichas posiciones tratando la información de éstas como información desconocida. El método resultante filtra errores $O(1)$ de forma eficiente, tanto si son errores aleatorios como sistemáticos e incluso si su distribución en la base de datos está concentrada o esparcida por ella.

Primero, se ilustra el funcionamiento del método con una base de datos modelo bidimensional, que resulta de la discretización de una función transcendental. Posteriormente, se presentan algunos casos prácticos de aplicación del método a dos bases de datos tridimensionales aerodinámicas que contienen la distribución de presiones sobre un ala a varios ángulos de ataque. Estas bases de datos resultan de modelos numéricos calculados en CFD.

Acknowledgements

The work in this Thesis has been funded by the Spanish Ministry of Science and Innovation under Grant TRA2013–45808-R.

I would like to thank my supervisors José Manuel Vega de Prada and José Manuel Perales Perales for their help and support in the process of preparing and writing this Thesis. I am also in debt to Miguel Ángel González Hernández who in the first place encouraged me to start my PhD studies and has always served with his vast knowledge in experimental aerodynamics. Finally, I would like to give thanks to my colleague Ana Isabel Moreno López with whom I could discuss any doubts or theories that I had and that together we were capable of solving any difficulty.

Lastly, I would like to thank me family and friends, especially my girlfriend Karolina, for constantly supporting me through all those years and providing the much needed distraction when necessary.

Contents

Abstract	iii
Resumen	v
Acknowledgements	vii
1 Introduction	1
2 Mathematical Background	7
2.1 SVD and POD	7
2.2 HOSVD	9
3 Filtering method and its validation	15
3.1 Cleaning two-dimensional databases	15
3.1.1 The improved gappy reconstruction SVD method	15
3.1.2 The error cleaning SVD method	17
3.2 Cleaning high-dimensional databases	19
3.2.1 The improved gappy reconstruction HOSVD method	19
3.2.2 The error cleaning HOSVD method	21
3.3 Toy model database	22
3.4 Performance of the method on the toy model	25
3.4.1 Random errors at randomly chosen database positions	25
3.4.2 Systematic errors errors at randomly chosen database po- sitions	30
3.4.3 Random errors at concentrated database positions	32
3.4.4 Systematic errors at concentrated database positions	34
3.5 Conclusions	35
4 Application to a CFD-generated aerodynamic database	37
4.1 Model definition	37
4.2 Error filtering	40
4.2.1 Random errors at randomly located positions	40
4.2.2 Systematic errors at randomly located positions	43
4.2.3 Random errors at concentrated positions	45
4.2.4 Systematic errors at concentrated positions	47
5 Method performance analysis	49
5.1 Randomness influence for varying contamination	50
5.1.1 Random errors at randomly chosen database positions	50
5.1.2 Systematic errors at randomly chosen database positions	54

5.1.3	Random errors at concentrated database positions	58
5.2	Randomness influence for varying error magnitude	62
5.2.1	Random errors at randomly chosen database positions . .	62
5.2.2	Systematic errors at randomly chosen database positions .	66
5.2.3	Random errors at concentrated database positions	69
6	Automatic tuning of method parameters	73
6.1	Mode number selection using MSE	73
6.2	Verification	75
6.2.1	Random errors at randomly chosen database positions . .	76
6.2.2	Systematic errors at randomly chosen database positions .	80
6.2.3	Random errors at concentrated database positions	83
7	Conclusions and Future Work	87
A	Chapter 5, Section 5.1 coarse databases tables and figures	91
B	Chapter 5, Section 5.2 coarse databases tables and figures	99
C	Chapter 6 coarse databases tables and figures	107
	Bibliography	118

Chapter 1

Introduction

Multidimensional databases are needed in both scientific investigation and daily engineering practice. The size of these databases has grown substantially mainly due to the increase of both the data collection capabilities and the computational power of modern-day computers. Aerospace engineering is no exception where large databases are used in flight simulation, flight data collection, aerodynamic design, certification, etc. In the case of aerodynamic design huge databases are generated using both computational fluid dynamics (CFD) or wind tunnel/flight tests.

However, the quality of these databases may need to be improved since they are usually affected by unavoidable discretization and experimental errors. The nature of these errors depends on various factors. In particular:

- Intrinsic noise (often called background noise) in experimental tests, especially of electronic origin, produce errors that are usually small (smaller than 0.1%), random and spread along the database. The exception is when the weak analogue signal needs to be amplified, amplifying the noise as well [1].
- Ill-functioning sensors and the intrusive effect of sensors themselves produce errors that are usually systematic. These errors may be either localized or not, and either small or $O(1)$.
- Human errors and mistakes may either exhibit a random nature or be systematic, and are usually $O(1)$ and isolated.
- Discretization errors [2] in CFD generated databases may either be small (the ideal situation) or significant (due to constraints in both development time and available computational resources), and may be concentrated in specific regions (e.g., near the leading edge and the wing tip when simulating subsonic flows around aircraft wings) or spread along the database. Excluding round-off errors, these errors are usually systematic.

Increasing the quality of the databases requires filtering these errors out, which is the main object of this Thesis. As a main idea, roughly speaking, the methods to be developed consist in two steps. Namely, the methods will proceed by first identifying the erroneous points and then reconstructing them by using the genuine physical information that is present in the clean data.

Filtering errors greatly depends on their nature and magnitude. As anticipated above, errors can be either random or systematic. The latter exhibit their own redundancies, which may be known or not. Their magnitude may be small (e.g. specialised equipment measurement uncertainty) or not (obstructions of sensors such as pitot tube icing), and their distribution can be localized (one ill-functioning probe) or spread along the database (ill-functioning sensor acquiring data from all probes). Standard error filtering methods developed in classical error analysis [3] and digital signal processing [4, 5] are usually based on assumptions that require some previous knowledge about the nature of either the noise (e.g. high-frequency, zero-mean, Gaussian) that is being filtered and/or the underlying clean signal (e.g., linear, polynomial, harmonic). Unfortunately, especially in engineering applications, obtaining this information may be either impossible or require a time-consuming pre-process. Increasing the fidelity in experimental databases by repeating the experimental procedure several times is nearly always time consuming and very costly. Therefore, standard error filtering methods may apply to some noisy technological databases but not to others due to problems with elucidating the nature of the errors. For instance, the above mentioned $O(1)$ localized errors resulting from ill-functioning sensors in wind tunnel and flight tests are not adequate for standard methods. Instead, these errors are usually identified and filtered out in industry via manual plot-by-plot scanning, which is both very expensive and time consuming. As a result this Thesis will focus on medium (5-10%) and large (greater than 20%) errors.

The above mentioned databases share a very important feature, namely they are strongly correlated both in time and the physical space due to physical laws that govern the underlying physical problem. In the case of CFD applied to compressible aerodynamics the physics will be described by the Navier-Stokes, mass and energy conservation equations. A natural conclusion from this observation would be to take advantage of those patterns (redundancies) underlying the nature of those databases in order to identify errors, eliminate them and interpolate their correct values. Hence, those correlation, or so called redundancies, are fundamental filtering method developed in this Thesis.

Nowadays virtually all the databases are stored electronically and since the storage capacity of modern computers is constantly increasing to accommodate for it, it is the data quality that matters. Saving data, whether it originates from engineering application or not, requires a certain structure for it to be stored efficiently, pre-processed, etc. Furthermore, most databases depend on a set of parameters. For instance, in aeronautical wind tunnel tests these parameters can be either the flow parameters that vary (Re number, Mach number, angle of attack, etc.) or changes in the geometry (wing sweep, dihedral angle, etc.). Conveniently for data processing, this type of databases can be stored as higher order tensors. Similarly to a matrix or a vector, which is a 2- or 1-dimensional tensor (or a tensor of order 2 or 1) respectively, tensors can be represented as an organised multidimensional array of numerical values and hence the data can be stored retaining their correlation in the parameter space which, as mentioned

in previous paragraph, is essential for correctly identifying and eliminating errors. The need to analyse such information contained in such databases gave rise to a family of methods called *multiway data analysis* [6]. There are many kinds of methods which depend greatly on the application and the way they approach the multidimensionality of the data. The two that are the most well know and commonly applied are PARAFAC and Tucker models [7]. The former one stands for *Parallel Factor Analysis* while the latter was given the name of the mathematician who made a substantial contribution to the development of multiway data analysis and his contribution is fundamental from the point of view of this Thesis.

In order to lay the foundations for the filtering method, we need to start with truncated singular value decomposition (SVD) [8], which is somewhat equivalent to proper orthogonal decomposition (POD) [9]. SVD can be seen as a method for matrices used for identifying dimensions, called *modes*, which have the greatest influence on the variation of the data. By ordering those modes by their energy, i.e. the degree of influence, it exploits redundancies among the two original dimensions, also called *directions*, and produces a lower dimensional approximation. SVD has many applications, for instance database generation [10, 11, 12], derivation of reduced order models [13, 14, 15, 16, 17, 18], compression [19, 20] and image processing [21]. Additionally, it has a natural basic ability to filter noise out [22] from two-dimensional databases (namely, matrices). In other words, truncation retaining the most energetic modes cleans the database if noise is both uncorrelated to and less energetic than the retained modes. A much more efficient error filtering method will be obtained using *gappy-SVD*, or *gappy-POD*, an iterative method to reconstruct lost data in two-dimensional databases invented by Everson and Sirovich [23]. The robustness of this method was first improved independently by Beckers and Rixen [24] and Venturi and Karniadakis [25] for clean databases (exhibiting no significant noise) and was further improved to treat noisy databases. This second improvement is very important because, as recognized in [25], the former improvement of the Everson-Sirovich method was very sensitive to background noise in the non-gappy positions.

However, in principle, SVD can be only applied to matrices and thus it loses the redundancies across more than two dimensions. A solution to this difficulty is the so-called higher order SVD (HOSVD) which is an extension to tensors [26] of standard SVD invented by Tucker [27, 28] in 1960s and popularized more recently by de Lathuwer et al. [29, 30]. The most direct application of HOSVD is signal processing such as written digits recognition [31]. However, HOSVD has also been shown to be capable of database generation [12] and compression [32]. As reported in the latter, the fact that the compression factor growth is exponential with respect to the number of database dimension means that HOSVD mitigates the effect of dimensionality curse. It also facilitates further data analysis and processing as it allows to efficiently interpolate data in multidimensional space resulting from physical properties of the system. A

substantial increase in computational time efficiency has been shown for reciprocating engines, requiring real-time control [33], and conceptual design of aircraft components [34]. However, from the point of view of this Thesis, the most important application of HOSVD is employing it in reconstructing missing data in multidimensional databases contaminated with noise [35, 36]. The method called *gappy-noisy-HOSVD method* is based on the iterative method to reconstruct lost data in two-dimensional databases invented by Everson and Sirovich, later improved independently by Beckers et al. and Venturi et al. The main contribution is its expansion to multidimensional databases and the optimisation of the number of retained modes which minimises the noise influence on data reconstruction.

It is worth noting that until now there has been very little research on SVD-based error filtering methods. The closest resemblance to an error filtering method was a systematic error correction done for weather predictions [37]. This method corrected raw observation data, prone to outliers, by replacing their leading modes with the corresponding numerical analysis modes [38]. This method takes advantage of the already known noise filtering property and does not strictly look for individual errors but corrects the whole database. Against this background, the object of this thesis is to develop a flexible methodology able to filter out both systematic and random errors and reconstruct the missing data. Moreover, the method will be effective regardless of the application, that is whether the database is of engineering or any other origin, and the errors may result from various sources, with no *a priori* knowledge about their nature, i.e. they can be small or not, and either localized or spread along the whole database. The only requirement is that the database is numerical and in the form of a tensor.

As mentioned above, the method is based on a double-iterative gappy-noisy-HOSVD method. The first objective to be fulfilled is to identify the erroneous information. This will be done in the first iterative loop, where the database is repeatedly reconstructed using SVD, retaining a fixed number of modes, and compared to the original database. The reconstructed points that differ significantly from the original ones are marked as erroneous and their position in the database is updated every iteration. After the first loop is terminated, the method enters into the second iterative loop, the number of retained modes is incremented and the process is repeated. Once the second loop is finished, an optimal number of modes calculated and the erroneous point positions are permanently stored. Once again the whole process is repeated but this time a percentage of erroneous points positions are known and will be treated as gappy points (i.e. equal to zero) in order to avoid contaminating the database reconstruction. This final steps forms the third and last iterative loop.

The fact that the error positions can be identified iteratively as those database positions in which the discrepancy between the given database and its SVD reconstruction is largest is a very simple idea. The triple-iterative *error filtering method* is the core of this Thesis and will enable the method to recognize very well both random and systematic errors without any *a priori* knowledge of the

clean database and the nature of the errors. Additionally, once the error positions are known, they are treated as gappy points and are reconstructed using missing data reconstruction methods based on SVD, which yield very small reconstruction errors [35]. Finally, replacing SVD with HOSVD will allow to treat multidimensional databases in an analogous manner, which is one of the important aspects of this Thesis. Another object of this Thesis is to minimise the number of pre-defined parameters which in certain occasions can be a limitation. One of the most crucial parameters, which can significantly influence the method's performance, is the minimum and maximum modes that the method can retain.

The verification of the error filtering method functioning will be done using a toy model two-and three-dimensional database obtained upon discretization of a transcendental function which is going to be contaminated by both random and systematic errors at random and concentrated locations. It will be followed by a verification using a three-dimensional aerodynamic database comprising of pressure distribution over a wing for various angles of attack. The database will be obtained using open-source CFD package called OpenFOAM® using Spalart-Allmaras one-equation turbulence model [39]. The size of the model will be determined to simulate a real wind tunnel experiment where there is limited space for pressure sensors. Analogically, the database will be contaminated by both systematic and random errors placed at random and concentrated positions, especially in the regions of high pressure gradients. Apart from the creation of the CFD database, all applications will be performed using MATLAB on a desktop PC, with a i7-4930K CPU @ 3.40GHz processor meaning that the method does not require much processing power. Applying it to the examples in this Thesis required always less than a couple of minutes to finish, even for the largest databases.

This Thesis is organized as follows. The mathematical background of POD, SVD and HOSVD is outlined in Chapter 2 in order to familiarise the reader with the topic and to establish necessary notation. It will be followed by Chapter 3, where the principles of gappy reconstruction for matrices based on SVD are recalled and adapted for the purposes of error filtering. A comprehensive explanation is given on the structure of the filtering method, outlining the necessary conditions needed to be met in order for the method to function properly. The method is then extended to higher-order tensors by replacing SVD and HOSVD. For the purpose of this Thesis, the methodology is illustrated only for a third order tensor due to clarity reasons and the fact that the expansion of the theory to higher dimensions is straightforward. Afterwards, the definition of the toy model is given and discretized to 2D and 3D meshes required for testing purposes. The Chapter is concluded with a thorough analysis of the filtering method's application to those meshes which are contaminated by either random or systematic at random or concentrated locations. The Chapter 4 describes the CFD model used to obtain the 3D aerodynamic database (two spatial dimensions and the angle of attack). The database is later contaminated with errors analogously to the toy model databases and the results obtained

from the filtering method application are thoroughly explained. In Chapter 5 a performance analysis is done in order to determine the functioning limits of the filtering method, both in terms of error contamination level and its magnitude. The analysis is performed for both random and systematic errors at at random and concentrated locations. In the penultimate Chapter 6 a modification to the method is given which allows to automatically assign the minimum and maximum number of modes that can be retained by the method. The result of applying this modification to the method are compared against the pre-defined minimum and maximum number of retained modes used in the previous analyses. Finally, Chapter 7 summarises the findings of this Thesis and suggests future research activities that could be perused to further expand the research on the topic of error filtering methods based on SVD and HOSVD.

Chapter 2

Mathematical Background

The aim of this chapter is to recall the standard mathematical techniques of matrix and tensor decomposition, on which the filtering method is based.

2.1 SVD and POD

The POD of a system of vectors and the SVD of the associated matrix (whose columns are the vectors) are closely related to each other, even though they have a different origin.

SVD seems to have been first invented independently by Beltrami and Jordan (in 1873-74) for square matrices associated with bilinear forms, and extended for rectangular matrices by Eckart and Young [40]. The SVD of a $I \times J$ -matrix A is given by [8] $U\Sigma V^\top$, or in terms of the elements of the matrix, as

$$A = U\Sigma V^\top \quad \text{or} \quad A_{i_1 i_2} = \sum_{l=1}^r \sigma_l U_{i_1}^l V_{i_2}^l, \quad (2.1)$$

where r is the rank of A . Here, $U_i^1, \dots, U_i^r$ and $V_j^1, \dots, V_j^r$ (known as *left and right SVD-modes*, respectively) are orthonormal eigenvectors of the positive definite matrices $AA^\top$ and $A^\top A$, respectively, and $\sigma_1, \dots, \sigma_r$ (known as *singular values*) are the square roots of the associated eigenvalues.

Computational efficiency in the calculation of the SVD modes is improved applying a QR decomposition to the matrix A , as $A = Q \cdot R \cdot E$, where Q is orthogonal, R is upper triangular, and E is a column permutation matrix. Thus, $A^\top A = E^\top \cdot R^\top \cdot R \cdot E$, which is quite advantageous when the number of rows of A is much larger than the number of columns. Once the right-modes have been calculated as the eigenvectors of $A^\top A$ associated with nonzero singular values, the left modes are calculated as $U^k = \sigma_k^{-1} A \cdot V^k$. If the number of columns is larger than the number of rows, the improvement above is applied to $A^\top$. Since the object of this thesis is to develop a filtering method, SVD decomposition will be calculated using a very efficient MATLAB command 'svd' with 'econ'

option. The $I_1 \times r$ -matrix U and the $I_2 \times r$ -matrix V are such that

$$U \cdot U^\top = V \cdot V^\top = r \times r \text{ unit matrix}, \quad (2.2)$$

where U and V are matrices formed from left and right SVD-modes respectively and the r elements of the diagonal matrix Σ are the singular values sorted in decreasing order. If the expansion (2.1) is truncated to $m < r$ terms, as

$$A_{i_1 i_2}^{\text{trunc}} = \sum_{l=1}^m \sigma_l U_{i_1}^l V_{i_2}^l, \quad (2.3)$$

then the root mean square error (RMSE) of the approximation coincides with the following truncation error estimate (TEE), as

$$\text{RMSE} \equiv \sqrt{\frac{1}{I_1 I_2} \sum_{i_1, i_2} (A_{i_1 i_2} - A_{i_1 i_2}^{\text{trunc}})^2} = \text{TEE} \equiv \sqrt{\frac{1}{I_1 I_2} \sum_{l=m+1}^r \sigma_l^2}. \quad (2.4)$$

SVD modes are customized to the particular database that is being described: for each $m \leq r$, the first m left and right SVD modes span two manifolds that are the linear manifolds of dimension m that give the best joint approximation of the columns and rows of the database A , respectively. And truncated SVD minimizes the left hand side of Equation (2.4), meaning that SVD is an *optimal decomposition* in the least-squares sense. Note that truncated SVD yields a very good approximation of the matrix when the neglected singular values are small, which occurs when appropriate redundancies exist among the rows and columns of the matrix A . In addition to RMSE, as defined in (2.4), maximum errors (MaxE) are also convenient and defined as

$$\text{MaxE} = \max_{i_1 i_2} |A_{i_1 i_2} - A_{i_1 i_2}^{\text{trunc}}|. \quad (2.5)$$

Truncation means retaining only the most energetic (measuring energy with the Frobenius norm defined as $\|A\| = \sqrt{\sum_{i,j} A_{ij}^2}$) modes, and the neglected modes are orthogonal to the retained modes, which means that highly oscillatory, noisy errors are filtered out by truncation. In general, such automatic filtering effect depends on both the magnitude and structure of errors.

POD seems to have been first obtained by Pearson [41] in 1901. The POD of a system of m vectors, $v_1, \dots, v_m$, is calculated considering the eigenvalue problem

$$\sum_l R_{kl} \alpha_j^l = \sigma_j^2 \alpha_j^k, \text{ with } R_{ij} = \langle v_i, v_j \rangle, \quad (2.6)$$

where the rank of the symmetric, positive semidefinite matrix $\mathbf{R}$ (covariance matrix) equals the rank of vector system, r . The *inner product* $\langle \cdot, \cdot \rangle$ can be customized. The scalars $\sigma_i > 0$ (*singular values*) are ordered as $\sigma_1 \geq \dots \geq \sigma_r > 0$

and the eigenvectors

$$U_i = \sum_k \alpha_i^k v_k \quad (2.7)$$

(known as *POD modes*) are selected as orthonormal, which means that

$$\sum_l \alpha_i^l \alpha_j^l = \delta_{ij} / (\sigma_i \sigma_j), \quad \sum_i \sigma_i^2 \alpha_i^l \alpha_i^k = \delta_{lk}, \quad (2.8)$$

where δ is the Krönecker delta. Thus,

$$v_k = \sum_i \sigma_i^2 \alpha_i^k U_i, \quad (2.9)$$

as obtained multiplying (2.7) by $\sigma_i^2 \alpha_i^k$, adding in k , and invoking (2.8). The POD-modes are such that, for each $m < r$ the manifold spanned by the first m modes is the m -dimensional manifold that provides the best root mean square (RMS) approximation of the system of vectors. The truncation error is

$$\sum_k \|v_k - \sum_{i=1}^m \langle v_k, U_i \rangle U_i\|^2 = \sum_{i=m+1}^r \sigma_i^2. \quad (2.10)$$

If the standard inner product is used, then the POD modes are readily seen to coincide with the right-SVD-modes of the matrix whose columns are the vectors $v_1, \dots, v_m$. Thus, computational efficiency of POD calculations is also improved applying a QR decomposition, as explained above in connection with SVD.

It is to be noted that SVD is closely related to *proper orthogonal decomposition* (POD). In particular, the left SVD modes (namely, the orthonormal columns of the matrix V in (2.1)) are the *POD modes* of the columns of the matrix A . Thus, the MATLAB command ‘svd’ is also an efficient means to calculate POD.

2.2 HOSVD

For the sake of clarity, we consider third-order tensors, but the extension to higher order tensors is straightforward. Instead, the extension from of SVD to third order tensors is not straightforward. In other words, the extension of SVD to third order tensors, $A_{i_1 i_2 i_3} = \sum_{l=1}^r \sigma_l U_{i_1}^l V_{i_2}^l W_{i_3}^l$ (cf. Equation (2.1)), which is known as the *canonical decomposition* [26], is closely related to the computation of the rank of the tensor, which is a subtle matter [26, 42]. Thus, instead of the canonical decomposition, some more flexible decompositions are considered in which σ depends on three indexes. For a $I_1 \times I_2 \times I_3$ -tensor, these decompositions are of the form

$$A_{i_1 i_2 i_3} = \sum_{i'_1=1}^{r_1} \sum_{i'_2=1}^{r_2} \sum_{i'_3=1}^{r_3} \sigma_{i'_1 i'_2 i'_3} U_{i_1}^{i'_1} V_{i_2}^{i'_2} W_{i_3}^{i'_3}, \quad (2.11)$$

and can be performed in various ways, depending on how the *core tensor* σ and the three sets of *modes*, $\{U_{i_1}^1, \dots, U_{i_1}^{r_1}\}$, $\{V_{i_2}^1, \dots, V_{i_2}^{r_2}\}$, and $\{W_{i_3}^1, \dots, W_{i_3}^{r_3}\}$, are defined. HOSVD was invented by Tucker [28] and has been more recently popularized by de Lathauwer et al. [29]. In this decomposition, the modes along the various directions of the tensor are the POD modes of the associated tensor *fibres* (namely, the vectors that are obtained fixing two of the indexes of the tensor, see Figure 2.1).

FIGURE 2.1: The fibres of a third order tensor.

In other words, the three sets of modes are the (orthonormal) eigenvectors associated with the strictly positive eigenvalues of the positive definite, symmetric matrices B^1 , B^2 , and B^3 , defined as

$$B_{i_1 i_1^*}^1 = \sum_{i_2, i_3} A_{i_1 i_2 i_3} A_{i_1^* i_2 i_3}^*, \quad B_{i_2 i_2^*}^2 = \sum_{i_1, i_3} A_{i_1 i_2 i_3} A_{i_1 i_2^* i_3}^*, \quad B_{i_3 i_3^*}^3 = \sum_{i_1, i_2} A_{i_1 i_2 i_3} A_{i_1 i_2 i_3^*}^*. \quad (2.12)$$

Namely, *HOSVD-modes* (selected with norm one) are eigenvectors of

$$\sum_{i_1^*=1}^{I_1} B_{i_1 i_1^*}^1 U_{i_1^*}^{i_1'} = \alpha_{i_1'}^2 U_{i_1}^{i_1'}, \quad \sum_{i_2^*=1}^{I_2} B_{i_2 i_2^*}^2 V_{i_2^*}^{i_2'} = \beta_{i_2'}^2 V_{i_2}^{i_2'}, \quad \sum_{i_3^*=1}^{I_3} B_{i_3 i_3^*}^3 W_{i_3^*}^{i_3'} = \gamma_{i_3'}^2 W_{i_3}^{i_3'} \quad (2.13)$$

for $i_1' = 1, \dots, r_1$, $i_2' = 1, \dots, r_2$, and $i_3' = 1, \dots, r_3$, where r_1 , r_2 , and r_3 are the ranks of the matrices B^1 , B^2 , and B^3 , and the square roots of the eigenvalues, $\alpha_{i_1'} > 0$, $\beta_{i_2'} > 0$, and $\gamma_{i_3'} > 0$ are known as HOSVD-singular values. The core tensor σ is calculated from

$$\sigma_{i_1' i_2' i_3'} = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \sum_{i_3=1}^{I_3} A_{i_1 i_2 i_3} U_{i_1}^{i_1'} V_{i_2}^{i_2'} W_{i_3}^{i_3'}. \quad (2.14)$$

as obtained multiplying (2.11) by $U_{i_1}^{i_1'} V_{i_2}^{i_2'} W_{i_3}^{i_3'}$, adding in i_1 , i_2 , and i_3 , and recalling that the HOSVD-modes are orthonormal. Substitution of Equation (2.14) into Equation (2.11) provides an exact representation of the tensor A . The core tensor and the HOSVD modes are very efficiently calculated using the MATLAB routine 'hosvd' (provided with the MATLAB toolbox TP Tool), which is able to treat tensors of any order.

As in standard SVD, if the elements of A show redundancies along the three dimensions, then the singular values along the three directions decay very fast,

and truncating the expansion in Equation (2.11) to $m_1 \leq r_1$, $m_2 \leq r_2$, and $m_3 \leq r_3$ terms, as

$$A_{i_1 i_2 i_3} \simeq A_{i_1 i_2 i_3}^{\text{trunc.}} \equiv \sum_{i'_1=1}^{m_1} \sum_{i'_2=1}^{m_2} \sum_{i'_3=1}^{m_3} \sigma_{i'_1 i'_2 i'_3} U_{i_1}^{i'_1} V_{i_2}^{i'_2} W_{i_3}^{i'_3}, \quad (2.15)$$

may still give a good approximation. In fact, the RMS *truncation error*, defined in terms of the root mean square norm

$$\|A\|_{\text{RMS}} = \sqrt{\frac{1}{I_1 I_2 I_3} \sum_{i_1, i_2, i_3} A_{i_1 i_2 i_3}^2}, \quad (2.16)$$

can be estimated/bounded in terms of an a priori *truncation error estimate* (TEE), as

$$\|A - A^{\text{trunc.}}\|_{\text{RMS}} \leq \text{TEE} \equiv \sqrt{\frac{1}{I_1 I_2 I_3} \left[\sum_{i'_1=m_1+1}^{r_1} (\sigma_{i'_1}^1)^2 + \sum_{i'_2=m_2+1}^{r_2} (\sigma_{i'_2}^2)^2 + \sum_{i'_3=m_3+1}^{r_3} (\sigma_{i'_3}^3)^2 \right]}, \quad (2.17)$$

where $\sigma_{i'_1}^1$, $\sigma_{i'_2}^2$, and $\sigma_{i'_3}^3$ are the POD-singular values of the fibers of the tensor along the i_1 , i_2 , and i_3 directions, respectively. Note that TEE is an upper bound, while its counterpart for the standard SVD, in Equation (2.4), is exact. Equation (2.17) confirms that, as anticipated, the truncation RMSE is small provided that the neglected singular values are small compared to the retained ones. In the sequel, the truncation errors will be measured in terms of both the RMSE and MaxE defined as

$$\text{RMSE} = \|A - A^{\text{trunc.}}\|_{\text{RMS}}, \quad \text{MaxE} = \max_{i_1 i_2 i_3} |A_{i_1 i_2 i_3} - A_{i_1 i_2 i_3}^{\text{trunc.}}|, \quad (2.18)$$

Some remarks on the comparison between SVD and HOSVD, and their connection with POD, are now in order:

- Standard SVD is a variant of POD. In fact, the left and right SVD-modes of (2.1) can be seen as the POD modes associated with the columns and rows of the matrix A , respectively. This is because left modes and right modes are eigenvectors of $AA^\top$ and $A^\top A$, which are the covariant matrices of the columns and rows. The components of the right modes can be seen as the coefficients in the projection of the columns of the matrix onto the left modes. Namely $V_{i_2}^m = \alpha_m^{-1} \sum_{i_1} A_{i_1 i_2} U_{i_1}^m$.
- HOSVD is a genuine extension to tensors of standard SVD, which is not obvious because the 2D restriction of (2.11),

$$A_{i_1 i_2} = \sum_{i'_1=1}^r \sum_{i'_2=1}^r \sigma_{i'_1 i'_2} U_{i_1}^{i'_1} V_{i_2}^{i'_2}, \quad (2.19)$$

does not coincide in principle with (2.1). But the modes $U_{i_1}^{i'_1}$ and $V_{i_2}^{i'_2}$ are the eigenvectors of the matrices B^1 and B^2 , defined ignoring the third index in (2.12). These are readily seen to be $B^1 = AA^\top$ and $B^2 = A^\top A$; namely, these two matrices coincide with the matrices that provide the left and right modes in standard SVD. The core tensor σ (which is now a 'core matrix') is calculated from the 2D counterpart of (2.14), and is readily seen to be diagonal, with its non-zero elements equal to the singular values provided by standard SVD. Thus, (2.19) coincides with the standard SVD decomposition in Equation (2.1), as stated.

- The calculation of the HOSVD modes and the core tensor are performed in two steps. (i) The three sets of HOSVD modes in (2.11) are the POD modes resulting from the fibers of the tensor in the three indexes, and (ii) the core tensor results from projecting the whole tensor onto the set of tensor products of the HOSVD modes. Since step (i) involves a standard POD, computational efficiency is greatly improved applying a QR decomposition, as explained above. The calculation of HOSVD-modes and the core tensor are both mathematically well posed and computationally inexpensive (compared with other decompositions), which is especially relevant when the decomposition is to be applied a large number of times, as it will happen in the iterative methods that will be presented in this paper. This is a quite important difference with canonical decompositions and also with other Tucker decompositions based on error minimization [26], which require solving a nonlinear minimization problem that is not always well posed and is quite computationally expensive.
- HOSVD is not optimal, but it is nearly optimal since (i) HOSVD modes are POD modes of the fibers of the tensor and thus they provide best RMS approximations of the fibers of the tensor; and (ii) once the modes have been calculated, the orthogonal projection implicit in (2.14) means that the core tensor optimizes the projection in the sense of mean squares.
- If the right hand side of (2.17) is small, then the resulting HOSVD truncated expansion is quite close to the approximation that would be obtained minimizing the RMS difference between the truncated expansion and the original tensor. Such optimization process is quite computationally expensive, since it leads to a coupled, nonlinear problem [26, 30] to determine the HOSVD-modes and the core tensor simultaneously.
- The third order tensor can be treated via standard SVD, isolating one of the indexes (say, the first index). The associated SVD is

$$A_{i_1 i_2 i_3} = \sum_{l=1}^{r_1} \sigma_l^1 U_{i_1}^l V_{i_2 i_3}^l, \quad (2.20)$$

where the left modes $\{U_{i_1}^1, \dots, U_{i_1}^{r_1}\}$ and the singular values $\sigma_1^1, \dots, \sigma_{r_1}^1$ are still given by the POD of the fibres along the index i_1 , while the right modes $\{V_{i_2 i_3}^1, \dots, V_{i_2 i_3}^{r_1}\}$ are given by $V_{i_2 i_3}^l = (\sigma_l^1)^{-1} \sum_i A_{i i_2 i_3} U_{i_1}^l$. But this

decomposition only takes into account redundancies in the first index i and thus yields a less efficient truncation than HOSVD: truncating to $m_1 < r_1$ terms in Equation (2.20), the required storage capacity is $m_1 + m_1(I_1 + I_2I_3)$, which (assuming that $m_1 \ll I_1$, $m_2 \ll I_2$, $m_3 \ll I_3$, and $I_1 \sim I_2 \sim I_3$) scales as $m_1I_2I_3$. Instead, truncation to (m_1, m_2, m_3) -terms in the HOSVD (see Equation (2.15)) requires a storage capacity of $m_1m_2m_3 + m_1I_1 + m_2I_2 + m_3I_3$, which is much smaller than $m_1I_2I_3$. In fact, taking into account that HOSVD is orthogonal, the required storage capacity, which will be referred to as *truncated information* (TI), is

$$\text{TI} = m_1m_2m_3 + \frac{(2I_1 - m_1 - 1)m_1 + (2I_2 - m_2 - 1)m_2 + (2I_3 - m_3 - 1)m_3}{2}. \quad (2.21)$$

Thus, HOSVD is much more efficient than standard SVD in multidimensional databases to identify the relevant information as it is able to capture the redundancies along all dimensions of the tensor. This is important because identifying the relevant information is the main ingredient in the database error filtering method developed below.

Chapter 3

Filtering method and its validation

3.1 Cleaning two-dimensional databases

For the sake of clarity we first develop the method in terms of SVD, for two-dimensional databases. Another reason for such approach is that this is the smallest dimension that can be considered and there exist many databases that are only two-dimensional, such as a pressure distribution over the wing for a single flight condition.

3.1.1 The improved gappy reconstruction SVD method

Now, the object of *gappy reconstruction methods* is to reconstruct a part of the database using the remaining data. The missing data may either be lost data or data that are difficult or impossible to calculate. After the pioneering work by Yates [43], who reconstructed gappy data using least squares, a variety of gappy reconstruction methods has been developed. Here, we concentrate on a SVD-based iterative gappy reconstruction method for two-dimensional databases introduced by Everson and Sirovich [23]. When applied to a gappy matrix A , the iterative *Everson-Sirovich* (hereafter, *ES*) *method* relies on truncated SVD, retaining a fixed number m of modes. The iterations $\tilde{A}^s$ are calculated as follows:

- *First step*: An initial guess, A^0 , is considered.
- *s-th step* (for $s = 2, 3, \dots$): A modified matrix $\hat{A}^s$ is defined as $\hat{A}^s = \tilde{A}^{s-1}$ at gappy points and $\hat{A}^s = A$ at non-gappy points. $\tilde{A}^s$ is obtained by applying truncated SVD, retaining m modes, to the modified matrix $\hat{A}^s$.

A good *stopping condition* for the ES method consists in terminating the iteration when either

$$(a) \text{ EG}_s < \text{CF} \cdot \text{TEE}_s \quad \text{or} \quad (b) \text{ } s > s_{\min}, \quad \text{EG}_s > \text{EG}_{s-1}, \text{ and } \text{EG}_{s-1} > \text{EG}_{s-2}. \quad (3.1)$$

Here, the parameters CF and $s_{\min}$ are tunable, TEE_s is the error estimate defined in Equation (2.4) for the truncated SVD applied at step s , and EG is the RMS difference between the former and the present reconstructions in the gappy

region at step s , defined as

$$EG_s = \sqrt{\frac{1}{|S^{gappy}|} \sum_{(i_1 i_2) \in S^{gappy}} |\hat{A}_{i_1 i_2}^s - \hat{A}_{i_1 i_2}^{s-1}|^2}, \quad (3.2)$$

where S^{gappy} is the set of indexes corresponding to the gappy points and $|S^{gappy}|$ is the total number of gappy points. In other words, the inner iteration is terminated when either:

- (a) The RMS difference (labelled as EG_s) between the reconstructions in the gappy part of the database in two consecutive iterations is a fraction (measured by CF) of the error estimate TEE. This is consistent with the fact that it would be useless to achieve convergence beyond the precision of the SVD reconstruction. The convergence factor CF must be somewhat small and is tunable.
- (b) The quantity EG_s reaches a minimum as s increases. Stopping the iteration at this minimum prevents divergence as s increases further, which would occur in some cases. The minimum number of iterations in (3.2), s_{min} , is tunable and avoids a possible irregular behaviour of the iterative process in the initial iterations. Again, the performance of the method is fairly insensitive to s_{min} .

This method is a very intuitive SVD-based gappy reconstruction method, but its robustness is limited: the method may not converge and the reconstruction may depend on both the initial guess and the number of retained POD-modes, m . The robustness of the ES method was increased independently, using the same idea, by Beckers and Rixen [24] (hereafter, BR) and Venturi and Karniadakis (hereafter, VK) [25] introducing a *second iteration*, in which the number of retained modes, m , is increased one by one:

- *First step*: The above mentioned ES method is applied with just one mode, using as initial condition a guess of the gappy data.
- *m-th step* (for $m = 2, 3, \dots$): The ES method is applied retaining m modes, taking as initial condition the converged result retaining $m - 1$ modes.

This method will be labelled as the ES-BR-VK method and was extensively tested by Gunes et al. [44] to provide very good results for clean databases. Nevertheless, as recognized in [25], for slightly noisy databases (exhibiting a small noise), the method produces a non-monotone behavior as the number of retained modes, m , increases. In other words, as m increases, the reconstruction error first decreases and then increases. The optimal number of retained modes will be estimated in this Thesis by using the following *mode selection estimate*

(MSE)

$$\text{MSE} = \frac{\sqrt{\sum_{m'=1}^m (\sigma_m^{m'})^2 \|U_m^{m'} - U_{m-1}^{m'}\|^2}}{\sqrt{\sum_{m'=1}^m (\sigma_m^{m'})^2}} + \frac{\sqrt{\sum_{m'=1}^m (\sigma_m^{m'})^2 \|V_m^{m'} - V_{m-1}^{m'}\|^2}}{\sqrt{\sum_{m'=1}^m (\sigma_m^{m'})^2}} \quad (3.3)$$

in the interval

$$N^{\min} \leq m \leq N^{\max}, \quad (3.4)$$

where $N^{\min}$ and $N^{\max}$ are tunable. Here, for $m' = 1, \dots, m$, $\sigma_m^{m'}$ are the singular values in the truncated SVD applied at the m -th step, and the vectors $U_m^{m'}$ and $V_m^{m'}$ are the associated left and right SVD modes, respectively; the norm $\|\cdot\|$ is the usual Euclidean norm.

The idea behind the MSE defined in (3.3) is that if noise is small and relatively uncorrelated to the clean information, then the most energetic modes are associated with true redundancies of the clean database, while noise produces the less energetic modes. Thus, if only the most energetic 'clean' modes are retained, then the MSE and the error of the gappy-SVD reconstruction will both decrease as the number of retained modes increases. On the other hand, if too many modes are retained, then the redundancies associated with the noise will come into play (in the reconstruction at non-gappy points that is performed at the inner iteration) and contaminate all retained modes, promoting an increase in both the MSE and the error of the gappy-SVD reconstruction. The crossover between these two situations will occur around the minimum of MSE, which is expected to give a good estimate of the 'optimal' numbers of modes to be retained. Restricting from below the interval (3.4) (through $N^{\min}$) avoids the irregular behaviour when the number of retained modes is too small; the restriction from above (through $N^{\max}$) avoids a possible spurious decrease of the MSE when a sufficient number of modes are retained as to also capture well the redundancies associated with noise.

The ES-BR-VK method, selecting m according to the estimate (3.3), will be labelled below as the *improved gappy-SVD method* and the reconstruction of the entire domain will be labelled as the *improved gappy virtual reconstruction*. The method will be seen in Section 3.4 to provide very good results in noisy databases.

3.1.2 The error cleaning SVD method

Now, the idea is to use the improved gappy-SVD method developed in the last subsection to both identifying error locations (as those positions in the database where the reconstruction by the method shows the largest errors) and reconstructing the associated data. Those error locations are used as gappy points, and evolve dynamically along the iterations of the improved gappy-SVD method. In other words, at each iteration step of the inner iteration of

the ES-BR-VK method (retaining a fixed number of SVD modes, m), the gappy points for the next iteration step are identified as those values of (i_1, i_2) such that

$$|\tilde{A}_{i_1 i_2}^{s-1} - A_{i_1 i_2}| > \text{FF} \cdot \max_{i_1 \leq I_1, i_2 \leq I_2} |\tilde{A}_{i_1 i_2}^{s-1} - A_{i_1 i_2}|, \quad (3.5)$$

where $\tilde{A}^{s-1}$ is the improved gappy virtual reconstruction (i.e. the whole domain, in this case a matrix, reconstructed using the improved gappy-SVD method) obtained in the previous inner iteration step and A is the original (contaminated) database; the factor FF is tunable. Now the method is allowed to run until the number of retained modes $m \leq N_1^{\max}$ and the gappy points identified at the end of the inner iterations, retaining the 'optimal' number of modes (greater or equal to $N_1^{\max}$) in the outer iteration loop (using the MSE defined in Equation (3.3)), are identified as the contaminated points, and the associated gappy reconstruction, as the final cleaned database. The minimum and maximum retained modes, $N_1^{\max}$ and $N_1^{\min}$, are set *a priori* and their selection is explained below. This method will be labelled as the *basic cleaning-SVD method*.

The method works fairly well but may not identify all gappy points. Thus, the whole (doubly) iterative method described above is *iterated* (namely, applied again and again) retaining an increasing number of modes to identify new contaminated positions. In the new applications of the method, labelled with index k , the method proceeds as explained above, except for two differences:

- i. Those points that were identified as contaminated in the previous application of the method (corresponding to an index set S_{cont}^{k-1}) are kept as gappy points from the outset. The new contaminated points at each iteration in the new application of the method are those such that (cf Equation (3.5))

$$(i_1, i_2) \notin S_{\text{cont}}^{k-1} \quad \text{and} \quad |\tilde{A}_{i_1 i_2}^{s-1} - A_{i_1 i_2}| > \text{FF} \cdot \max_{(i_1, i_2) \notin S_{\text{cont}}^{k-1}} |\tilde{A}_{i_1 i_2}^{s-1} - A_{i_1 i_2}|. \quad (3.6)$$

- ii. The optimal number of modes in the k -th application of the method (using the MSE defined in (3.3)), N_n^{opt} , is required to be such that

$$\max\{N_{k-1}^{\text{opt}} - N^-, N^{\min}\} \leq N_n^{\text{opt}} \leq \min\{N_{k-1}^{\text{opt}} + N^+, N^{\max}\}, \quad (3.7)$$

where the minimum and maximum allowable numbers of modes, $N^{\min}$ and $N^{\max}$, are tunable, as are the shifts N^- and N^+ . Equation (3.7) means that the optimal number of modes is not allowed to decrease or increase too much from one application of the method to another. This condition was also imposed in the first application of the method by setting $N_1^{\max} = N_1^{\min} + N^+$ and $N_1^{\min} = N^{\min}$.

In the subsequent applications of the basic cleaning-SVD method the RMSE of the reconstruction in the untreated part of the database, $\text{RMSE}_{\text{clean}}^k$, is defined as

$$\text{RMSE}_{\text{clean}}^k = \|A_{i_1 i_2} - \tilde{A}_{i_1 i_2}^k\|_{\text{RMS}} \quad \text{for } (i_1, i_2) \in S_{\text{clean}}^k \quad (3.8)$$

where $\tilde{A}^k$ is the improved gappy virtual reconstruction matrix calculated in the k -th application of the method, first decreases (as more and more truly contaminated points are identified) and then increases (as the method begins identifying as erroneous points that are not, and uses these points as gappy points). Thus, the basic method is applied again and again until the first value of k such that

$$\text{RMSE}_{\text{clean}}^{k+1} > \text{RMSE}_{\text{clean}}^k. \quad (3.9)$$

Additionally, as the method advances the reconstructed matrices become more and more similar. This means that MSE_k defined for k -th iteration

$$\text{MSE}_k = \frac{\sqrt{\sum_{m'} (\sigma_k^{m'})^2 \|\mathbf{U}_k^{m'} - \mathbf{U}_{k-1}^{m'}\|^2}}{\sqrt{\sum_{m'} (\sigma_k^{m'})^2}} + \frac{\sqrt{\sum_{m'} (\sigma_k^{m'})^2 \|\mathbf{V}_k^{m'} - \mathbf{V}_{k-1}^{m'}\|^2}}{\sqrt{\sum_{m'} (\sigma_k^{m'})^2}} \quad (3.10)$$

has to decrease and the method continues until

$$\text{MSE}_{k+1} > \text{MSE}_k. \quad (3.11)$$

is satisfied. This means that when either Equation (3.9) or (3.11) is reached, the k -th application is considered the final.

The resulting method will be called the *cleaning-SVD* method. Note that each iteration of the method involves an application of the basic cleaning-SVD method, which is itself (doubly) iterative.

3.2 Cleaning high-dimensional databases

For higher dimensional databases, the method proceeds as in the 2D case, except that SVD must be substituted by HOSVD, which is briefly described in the next subsection, and that the various ingredients must be modified accordingly.

3.2.1 The improved gappy reconstruction HOSVD method

This method is the natural extension of its counterpart described in Section 3.1.1 for 2D databases, substituting the matrix $A_{i_1 i_2}$ by the tensor $A_{i_1 i_2 i_3}$ and SVD by HOSVD. On the one hand, for each set of numbers of selected modes, the extension of the ES method retaining (m_1, m_2, m_3) modes along the three directions, proceeds iteratively as:

- *First step*: An initial guess, A^0 , is considered.
- *s-th step* (for $s = 2, 3, \dots$): A modified tensor $\hat{A}^s$ is defined as $\hat{A}^s = \tilde{A}^{s-1}$ at gappy points and $\hat{A}^s = A$ at non-gappy points. $\tilde{A}^s$ is obtained by applying truncated HOSVD, retaining (m_1, m_2, m_3) modes, to the modified tensor $\hat{A}^s$.

The stopping condition defined in Equation (3.1) is also used here. In other words, the inner iteration is terminated either

$$(a) \text{ EG}_s < \text{CF} \cdot \text{TEE}_s \quad \text{or} \quad (b) \text{ } s > s_{\min}, \quad \text{EG}_s > \text{EG}_{s-1}, \text{ and } \text{EG}_{s-1} > \text{EG}_{s-2} \quad (3.12)$$

holds, where $\text{CF} < 1$ is tunable and EG defined in Equation (3.2) is replaced by

$$\text{EG}_s = \sqrt{\frac{1}{|S^{\text{gappy}}|} \sum_{(i_1 i_2 i_3) \in S^{\text{gappy}}} |\hat{A}_{i_1 i_2 i_3}^s - \hat{A}_{i_1 i_2 i_3}^{s-1}|^2}. \quad (3.13)$$

The *outer iteration*, in which the total number of retained modes, $m = m_1 + m_2 + m_3$, is increased one by one in either direction, is also extended as:

- *First step*: The above mentioned ES method is applied with $m_1 = m_2 = m_3 = 1$, using as initial condition a guess of the gappy data.
- *m-th step* (for $m = 2, 3, \dots$): The ES method is applied retaining m modes, taking as initial condition the converged result retaining $m - 1$ modes. Now, since there are three types of HOSVD-modes (one per index), three possible modes can be selected to be added for the next iteration. A convenient selection is that mode promoting the largest decrease in TEE, which is that exhibiting a largest singular value (see Equation (2.17)). Thus, if the numbers of modes in the current inner iteration in the directions i_1 , i_2 , and i_3 are m_1 , m_2 , and m_3 , respectively, then the maximum among the HOSVD singular values $\sigma_{m_1+1}^1$, $\sigma_{m_2+1}^2$, and $\sigma_{m_3+1}^3$ is calculated and the associated HOSVD mode is selected for the next inner iteration.

The MSE is now defined as (cf (3.3))

$$\begin{aligned} \text{MSE} = & \frac{\sqrt{\sum_{m'_1=1}^{m_1} (\sigma_{m'_1}^1)^2} \|U_m^{m'_1} - U_{m-1}^{m'_1}\|^2}{\sqrt{\sum_{m'_1=1}^{m_1} (\sigma_{m'_1}^1)^2}} + \frac{\sqrt{\sum_{m'_2=1}^{m_2} (\sigma_{m'_2}^2)^2} \|V_m^{m'_2} - V_{m-1}^{m'_2}\|^2}{\sqrt{\sum_{m'_2=1}^{m_2} (\sigma_{m'_2}^2)^2}} \\ & + \frac{\sqrt{\sum_{m'_3=1}^{m_3} (\sigma_{m'_3}^3)^2} \|W_m^{m'_3} - W_{m-1}^{m'_3}\|^2}{\sqrt{\sum_{m'_3=1}^{m_3} (\sigma_{m'_3}^3)^2}}, \end{aligned} \quad (3.14)$$

where (using the same notation as in §2.2), $\sigma_{m'_1}^1$, $\sigma_{m'_2}^2$, and $\sigma_{m'_3}^3$ are the HOSVD-singular values associated with the fibres along the directions i_1 , i_2 , and i_3 , respectively, and $\|\cdot\|$ denotes the usual vector (Euclidean) norm. The modes numbers m_1 , m_2 , and m_3 are those retained in the former (outer) iteration, and the subscripts *old* and *new* denote the HOSVD modes in the former and current (outer) iterations, respectively.

From now on this method will be called the *improved gappy-HOSVD* method and the reconstruction of the entire domain will be labelled as, as in the caso of improved gappy-SVD method, the *improved gappy virtual reconstruction*.

3.2.2 The error cleaning HOSVD method

Again, this method is the natural extension of its counterpart in Section 3.1.2. Specifically, contaminated positions are identified using the counterpart of (3.5), namely selecting those database positions (i_1, i_2, i_3) such that

$$|\tilde{A}_{i_1 i_2 i_3}^{s-1} - A_{i_1 i_2 i_3}| > \text{FF} \cdot \max_{i_1 \leq I_1, i_2 \leq I_2, i_3 \leq I_3} |\tilde{A}_{i_1 i_2 i_3}^{s-1} - A_{i_1 i_2 i_3}|, \quad (3.15)$$

where $\tilde{A}^{s-1}$ is the improved gappy virtual reconstruction (i.e. the whole domain, in this case a tensor, reconstructed using the improved gappy-HOSVD method) obtained in the former inner iteration step and A is the original (contaminated) database; the factor FF is tunable. As in the two-dimensional case, the gappy points identified at the end of the inner iterations, retaining the 'optimal' number of modes (using the MSE defined in (3.14)), are identified as the contaminated points, and the associated gappy reconstruction, as the final cleaned database. The 'optimal' number of retained modes has to be greater or equal and lesser or equal to $N_1^{\min}$ and $N_1^{\max}$, respectively, which are set *a priori* and their selection is explained below. This method will be labelled as the *basic cleaning-HOSVD method*.

As for matrices, the method may not identify all gappy points. Thus, the whole (doubly) iterative method described above is *iterated* (namely, applied again and again) retaining an increasing number of modes to identify new contaminated positions. In the new applications of the method, labelled with index k , it proceeds as explained above, except for two differences:

- i. Those points that were already identified as contaminated in the previous application of the method (corresponding to an index set S_{cont}^{k-1}) are kept as gappy points from the outset. The new contaminated points at each iteration in the new application of the method are those such that (cf (3.6))

$$(i_1, i_2, i_3) \notin S_{\text{cont}}^{k-1} \quad \text{and} \quad |\tilde{A}_{i_1 i_2 i_3}^{s-1} - A_{i_1 i_2 i_3}| > \text{FF} \cdot \max_{(i_1, i_2, i_3) \notin S_{\text{cont}}^{k-1}} |\tilde{A}_{i_1 i_2 i_3}^{s-1} - A_{i_1 i_2 i_3}|. \quad (3.16)$$

- ii. The optimal number of modes in the k -th application of the method (using the MSE defined in (3.14)), N_n^{opt} , is required to be such that

$$\max\{N_{k-1}^{\text{opt}} - N^-, N^{\min}\} \leq N_n^{\text{opt}} \leq \min\{N_{k-1}^{\text{opt}} + N^+, N^{\max}\}, \quad (3.17)$$

where, as in the two-dimensional case, the absolute minimum and maximum allowable numbers of modes, $N^{\min}$ and $N^{\max}$, and the shifts N^- and N^+ are tunable. As before, this condition was also imposed in the first application of the method by setting $N_1^{\max} = N_1^{\min} + N^+$ and $N_1^{\min} = N^{\min}$.

As in the 2D case, the subsequent applications of the basic cleaning-HOSVD method, labelled with the index k , are applied again and again until the first value of k such that

$$\text{RMSE}_{\text{clean}}^{k+1} > \text{RMSE}_{\text{clean}}^k, \quad (3.18)$$

where $\text{RMSE}_{\text{clean}}^k$ is the RMSE of the reconstruction in the clean part of the database at the k -th iteration defined by

$$\text{RMSE}_{\text{clean}}^k = \|A_{i_1 i_2 i_3} - \tilde{A}_{i_1 i_2 i_3}^k\|_{\text{RMS}} \quad \text{for } (i_1, i_2, i_3) \in S_{\text{clean}}^k, \quad (3.19)$$

where $\tilde{A}^k$ is the improved gappy virtual reconstruction tensor calculated in the k -th application of the method. Similarly to the 2D case, as the method advances, the difference between cleaned tensors at each iterations will become smaller and smaller. Hence, defining a global MSE_k in the k -th iteration as

$$\begin{aligned} \text{MSE}_k = & \frac{\sqrt{\sum_{m'_1} (\sigma_k^{m'_1})^2 \|U_k^{m'_1} - U_{k-1}^{m'_1}\|^2}}{\sqrt{\sum_{m'_1} (\sigma_k^{m'_1})^2}} + \frac{\sqrt{\sum_{m'_2} (\sigma_k^{m'_2})^2 \|V_k^{m'_2} - V_{k-1}^{m'_2}\|^2}}{\sqrt{\sum_{m'_2} (\sigma_k^{m'_2})^2}} \\ & + \frac{\sqrt{\sum_{m'_3} (\sigma_k^{m'_3})^2 \|W_k^{m'_3} - W_{k-1}^{m'_3}\|^2}}{\sqrt{\sum_{m'_3} (\sigma_k^{m'_3})^2}}, \end{aligned} \quad (3.20)$$

another stop condition can be added stopping the method when

$$\text{MSE}_{k+1} > \text{MSE}_k \quad (3.21)$$

is satisfied. This means that when either Equation (3.19) or (3.21) is reached, the k -th application is considered the final.

The resulting method will be called the *cleaning-HOSVD* method.

3.3 Toy model database

The error method developed above will be first validated with a two and three-dimensional database defined by a three-variate transcendental function

$$\begin{aligned} f(x, y, z) = & x^2 [\sin(5\pi y + 3\log(x^3 + y^2 + z + \pi^2)) - 1]^2 - 4x^2 y^3 (1 - z)^{3/2} \\ & + (x + z - 1)(2y - z) \cos[30(x + z)] \log(6 + x^2 y^2 + z^3) \end{aligned} \quad (3.22)$$

defined on a cubic domain $0 \leq x, y, z \leq 1$. The section $z = 0$ of this function will define the bi-variate function used to construct the two-dimensional toy model database. A linear transformation is applied to f such that the resulting rescaled function varies between 0 and 1 in the cubic domain for both the 2D and 3D database. This transformation facilitates the interpretation of errors. Denoting the transformed function also as f , the tensor A is defined as

$$A_{i_1 i_2} = f(x_{i_1}, y_{i_2}), \quad \text{with } x_{i_1} = \frac{i_1 - 1}{I_1 - 1}, \quad y_{i_2} = \frac{i_2 - 1}{I_2 - 1} \quad (3.23)$$

for the 2D case and

$$A_{i_1 i_2 i_3} = f(x_{i_1}, y_{i_2}, z_{i_3}), \quad \text{with } x_{i_1} = \frac{i_1 - 1}{I_1 - 1}, y_{i_2} = \frac{i_2 - 1}{I_2 - 1}, z_{i_3} = \frac{i_3 - 1}{I_3 - 1} \quad (3.24)$$

for the 3D case. We consider *dense and coarse toy model databases*, with $I_1 = I_2 = 50$ and $I_1 = I_2 = 20$, respectively, for the 2D model and with $I_1 = I_2 = I_3 = 50$ and $I_1 = I_2 = I_3 = 20$, respectively, for the 3D model. The 2D and 3D databases are shown in Figure 3.1 and 3.2 respectively, where for the 3D database only the $z = 0, 1/2, 1$ sections are plotted.

FIGURE 3.1: The dense (left) and coarse (right) two-dimensional toy model databases.

FIGURE 3.2: The sections $z = 0$ (left), $1/2$ (middle), and 1 (right) of the dense (top) and coarse (bottom) toy model databases.

As can be seen f is oscillatory in the two directions and does not exhibit any simple regularity, except for being defined through sums, products, and functional composition of the sine, cosine, powers, and log functions. In particular, the 2D 20^2 mesh is quite coarse, especially in the x direction, where the function exhibits five oscillations, which means that the discretized database contains only four points per period. All these make the toy model database quite demanding (in fact, much more demanding than the usual databases in practical situations), except of course for the fact that the toy model database is 'exact', free from errors.

Lets now analyse the SVD and HOSVD decomposition for the 2D and 3D database respectively. SVD singular values and the three sets of HOSVD singular values associated with the dense and coarse toy model tensors are given in

FIGURE 3.3: SVD applied to the toy model matrices defined in Equation (3.22) and (3.23), plotting the SVD singular values (SVs) against the modes m for the dense (left) and coarse (right) 2D toy model matrices.

FIGURE 3.4: HOSVD applied to the toy model tensors defined in Equation (3.22) and (3.24), plotting the HOSVD singular values (HOSVs) associated with the indexes i_1 (solid lines), i_2 (dashed lines), and i_3 (dot-dashed lines) for the dense (left) and coarse (right) 3D toy model tensors.

Figure 3.3 and 3.4 respectively. Note that the singular values decay spectrally in both the dense and coarse databases, which is due to the redundancies along the two and three directions (for 2D and 3D databases respectively) imposed by the implicit structure of the database imposed by the function defined by Equation (3.22). Thus, truncating the SVD or HOSVD of these databases to an appropriate number of modes still gives a good approximation of the database. With respect to the 3D database, this means that, for a reasonable accuracy (say, the truncated RMSE $\sim 10^{-3}$), the truncated information (as defined in Equation (2.21)) is much smaller than the database size, which is essential for the good performance of the cleaning-HOSVD method. Also note that for both databases the singular value distributions are quite similar in the dense and coarse meshes, which means that the redundancies along their directions are well captured in the coarsest mesh.

3.4 Performance of the method on the toy model

In the following subsections, we consider several types of $O(1)$ -errors. These examples have been chosen to illustrate the main difference between standard methods and the present method, which is able to localize the contaminated error points in spite of the fact that the error is $O(1)$. Standard methods, instead, would not be able to filter the errors out without a knowledge of the nature of the errors (which are not needed at all with the present method).

The subsections will be ordered by the type of error applied and within each subsection the proper functioning of the method will be validated using the 2D and 3D databases considering both dense and coarse meshes. In all applications to the 2D database, the tunable parameters of the method appearing in Equations (3.1), (3.5) and (3.7), have been calibrated to be

$$CF = 0.1, \quad s_{\min} = 4, \quad FF = 0.5, \quad N_{\min} = 3, \quad N_{\max} = 15, \quad N^- = 1, \quad N^+ = 3. \quad (3.25)$$

Analogically for the 3D database, the tunable parameters of the method, appearing in Equations (3.12), (3.15) and (3.17), have been selected as

$$CF = 0.1, \quad s_{\min} = 4, \quad FF = 0.3, \quad N^{\min} = 15, \quad N^{\max} = 50, \quad N^- = 1, \quad N^+ = 10. \quad (3.26)$$

As already mentioned in Chapter 1, all calculations in this Thesis are performed using MATLAB on a desktop PC, with a 3.4 GHz processor. Several applications involve randomness, which means that the results (slightly) vary from one run to another.

3.4.1 Random errors at randomly chosen database positions

2D database

To begin with, we contaminate the dense toy model database with uniformly distributed random noise of average size 0.5 at 30% randomly chosen positions in the database. The cleaning-SVD method involves 19 iterations (until the conditions defined in Equation (3.9) and (3.11) hold) displayed in Table 3.1. As can be seen, Equation (3.5) subsequently determines the 750 contaminated points, without including points that are not contaminated, meaning that the performance of the method is optimal in this sense.

The method determines very quickly those points exhibiting the largest errors (in the first few iterations) retaining only 5 SVD modes. At the 9-th iteration, 701 contaminated points are determined retaining 8 SVD modes. The last 10 iterations determine the remaining 49 contaminated points more slowly. As expected, the errors (both the RMSE and MaxE) in the reconstruction of the

Iter.	CDEP	IDEP	m	$\text{MaxE}_{\text{cont}}$	$\text{RMSE}_{\text{cont}}$	T (sec)
1	202	0	4	$1.76 \cdot 10^{-1}$	$5.20 \cdot 10^{-2}$	0.1
2	288	0	5	$2.03 \cdot 10^{-1}$	$4.87 \cdot 10^{-2}$	0.0
3	336	0	4	$1.34 \cdot 10^{-1}$	$3.94 \cdot 10^{-2}$	0.1
4	469	0	4	$7.10 \cdot 10^{-2}$	$2.10 \cdot 10^{-2}$	0.0
5	544	0	5	$1.33 \cdot 10^{-1}$	$1.72 \cdot 10^{-2}$	0.1
6	608	0	4	$4.42 \cdot 10^{-2}$	$8.65 \cdot 10^{-3}$	0.1
7	656	0	5	$4.71 \cdot 10^{-2}$	$4.87 \cdot 10^{-3}$	0.0
8	681	0	5	$1.83 \cdot 10^{-2}$	$2.86 \cdot 10^{-3}$	0.1
9	701	0	8	$4.26 \cdot 10^{-2}$	$4.62 \cdot 10^{-3}$	0.1
10	710	0	7	$5.91 \cdot 10^{-2}$	$3.55 \cdot 10^{-3}$	0.1
11	727	0	6	$1.21 \cdot 10^{-2}$	$9.70 \cdot 10^{-4}$	0.1
12	735	0	9	$2.34 \cdot 10^{-2}$	$1.15 \cdot 10^{-3}$	0.1
13	736	0	12	$2.16 \cdot 10^{-2}$	$1.24 \cdot 10^{-3}$	0.2
14	741	0	11	$7.43 \cdot 10^{-3}$	$3.43 \cdot 10^{-4}$	0.3
15	744	0	14	$2.89 \cdot 10^{-3}$	$1.73 \cdot 10^{-4}$	0.3
16	747	0	15	$2.32 \cdot 10^{-4}$	$1.47 \cdot 10^{-5}$	0.4
17	748	0	15	$9.30 \cdot 10^{-4}$	$4.76 \cdot 10^{-5}$	0.6
18	749	0	15	$1.71 \cdot 10^{-4}$	$7.33 \cdot 10^{-6}$	0.5
19	750	0	15	$9.13 \cdot 10^{-8}$	$5.48 \cdot 10^{-9}$	0.6

TABLE 3.1: The 19 iterations of the cleaning-SVD method applied to the dense toy model database contaminated with a positive, random noise of 0.5 size at 30% randomly chosen points (namely, 750 points): numbers of correctly and incorrectly determined error points, CDEP and IDEP, respectively, number of retained modes, m , reconstruction errors defined in Equation (2.4) and (2.5), and required CPU time T (in seconds).

contaminated points steadily decrease along the iterations, reaching very small values in the last iteration. This is remarkable, noting that the error size was in the range $0 < \text{error} < 0.5$ and was positive (no zero mean). Besides, the method is quite fast, as seen in the last column. The whole process only takes a few CPU seconds.

The very good performance of the method is further illustrated in Figure 3.5, where it can be seen how the method identifies and corrects the contaminated positions, beginning with those showing the largest errors (in the first few iterations) and ending with those positions where the errors are smallest, which obviously are the most difficult to identify. In fact, it is not until the largest errors have been corrected (in the first 7 iterations) that the smallest errors show up (in the remaining iterations). In other words, the method works in a very natural and robust way.

Now, in order to illustrate how each individual gappy reconstruction method (at the iterations considered in Figure 3.5) works, Figure 3.6 shows the evolution with the number of retained modes of both the errors in the contaminated part of the database and the mode selection estimate MSE defined in Equation

FIGURE 3.5: The contaminated database (top, left) and the outcomes of the first two iterations considered in Table 3.1 (top, center and right) and iterations 5-7 (bottom); the outcomes of the iterations 8-13 are increasingly good, and the outcomes of the 14-th and subsequent iterations are plot-indistinguishable from the clean database.

(3.3). Note that the MSE selects well the optimal number of modes in each iteration. The last plot in this figure shows the evolution of the reconstruction errors along 20 iterations of the method and shows that the selected number of applications, 19, is the correct one since a further 20-th iteration would increase the reconstruction error. In other words, the stopping condition (3.9) and (3.11) work well.

Additional tests (not included for 2D cases for the sake of brevity but done for 3D cases in Chapter 5) have shown that, as expected, decreasing the number of contaminated points the performance of method improves, namely the required number of iterations and retained modes both decrease, as does the error of the final reconstruction. On the other hand, if the number of contaminated points increases, the performance of the method worsens. In fact, beyond a 35% contaminated points, the method identifies as contaminated points that are not, and the reconstruction errors increase. However, the reconstruction of correct points through the gappy-SVD method is still good as far as the number of incorrectly determined points is limited. Beyond 40% contamination, the method fails to clean the contaminated database. This is because the identified SVD modes in the gappy-reconstruction part of the method may be based in contaminated points, which subsequently are identified as correct points by the method.

Results for the coarse database (omitted for the sake of brevity) show that the performance of the cleaning-SVD method is worse than for the dense database, as expected, but still fairly good taking into account that the errors are $O(1)$. Namely, 16 % randomly located error positions are filtered out in 12 iterations, retaining 15 SVD modes and showing $\text{RMSE} \sim 10^{-6}$ and $\text{MaxE} \sim 10^{-5}$ in the last

FIGURE 3.6: From left to right, downwards. First five plots: the performance of the gappy reconstruction method in the iterations #1, #2, #5, #6, and #7 of the cleaning-SVD method considered in Fig3.5, plotting the RMSE (dashed black line) and MaxE (solid black line) in the contaminated part of the database and the MSE (solid blue line). The minimum and maximum allowable numbers of retained modes considered in Equation (3.7) are indicated with vertical dashed blue lines and the optimum number of retained modes is indicated with a vertical blue line. Last plot: evolution of RMSE and MaxE, $\text{RMSE}_{\text{clean}}$ (solid red line) defined in Equation (3.9) and MSE (solid blue line) defined in Equation (3.11) along the 19 iterations in Table 3.1.

iteration.

3D database

As in the two-dimensional case, we first consider the case in which errors are random, both in magnitude and in location on the database. The case of random errors of size 0.5 at 65% randomly located database positions (namely, 81250 positions) is considered in Table 3.2. Note that the number of contaminated positions more than doubles its counterpart for the two-dimensional case, see Table 3.1. Comparison with Table 3.1 shows that the performance of the method is qualitatively similar to its counterpart in the two-dimensional database, namely:

- The method correctly identifies more and more error points along the various iterations.
- All points identified as error points are indeed error points.

Iter.	CDEP	IDEP	m_1	m_2	m_3	m	$\text{MaxE}_{\text{cont}}$	$\text{RMSE}_{\text{cont}}$	T (sec)
1	47635	0	5	7	6	18	$1.25 \cdot 10^{-1}$	$4.58 \cdot 10^{-2}$	3.6
2	66973	0	6	10	7	23	$4.41 \cdot 10^{-2}$	$1.15 \cdot 10^{-2}$	3.7
3	75785	0	5	9	9	23	$1.06 \cdot 10^{-2}$	$2.16 \cdot 10^{-3}$	5.0
4	79322	0	7	10	9	26	$8.32 \cdot 10^{-3}$	$4.32 \cdot 10^{-4}$	5.6
5	80507	0	7	10	10	27	$1.76 \cdot 10^{-3}$	$9.96 \cdot 10^{-5}$	6.9
6	80992	0	8	10	12	30	$3.67 \cdot 10^{-4}$	$2.38 \cdot 10^{-5}$	7.2
7	81168	0	8	11	14	33	$7.90 \cdot 10^{-5}$	$5.86 \cdot 10^{-6}$	8.1
8	81225	0	9	14	14	37	$6.99 \cdot 10^{-5}$	$1.30 \cdot 10^{-6}$	8.9
9	81238	0	11	16	17	44	$2.14 \cdot 10^{-5}$	$3.46 \cdot 10^{-7}$	10.2
10	81249	0	12	19	17	48	$7.44 \cdot 10^{-6}$	$4.84 \cdot 10^{-8}$	12.7
11	81250	0	12	19	19	50	$5.23 \cdot 10^{-7}$	$4.93 \cdot 10^{-9}$	12.8

TABLE 3.2: Counterpart of Table 3.1 (giving the numbers of correctly and incorrectly determined error points, CDEP and IDEP, respectively, numbers of retained modes along the three database dimensions, the total number of retained modes $m = m_1 + m_2 + m_3$, the reconstruction errors defined in Equation (2.18), and required CPU time T , in seconds) along the various iterations for the three-dimensional toy model database, contaminated with random errors of size 0.5 at 65% randomly chosen positions.

- The reconstruction errors steadily decrease along the iterations, reaching very small values at the last iteration, when all error points have been identified.

However, as anticipated, a comparison with Table 3.1 shows that the performance of the method is quantitatively much better than in the two-dimensional case. In other words, the allowable percentage of error points is much larger and the required number of iterations much smaller.

The counterpart of Figure 3.5 shows a qualitatively similar behaviour and is omitted (illustrating the iterations requires more space here due to the three-dimensional character of the database). The counterpart Figure 3.6 is given in Figure 3.7. As can be seen, the plotted errors behave quite consistently, and the MSE selects very well the optimal number of retained modes.

As in the two-dimensional case, if the number of error positions is increased further, the method identifies points that are correct as error positions, and hence the reconstruction errors worsens. Increasing the number of error position too much, beyond 70%, the method identifies clean points as erroneous points, and beyond 75%, the method diverges. Also as in the two-dimensional case, complete results for the coarse database (omitted for the sake of brevity) have shown that the performance of the cleaning-SVD method is slightly worse than for the dense database, but still fairly good. Namely, 40% randomly located error positions are filtered out in 9 iterations, retaining 50 SVD modes and showing $\text{RMSE} = 6.12 \cdot 10^{-7}$ and $\text{MaxE} = 2.54 \cdot 10^{-5}$ in the last iteration.

FIGURE 3.7: From left to right, downwards. First five plots: the performance of the gappy reconstruction method in the first five iterations of the cleaning-HOSVD method considered in Table 3.2, plotting the RMSE (dashed black line) and MaxE (solid black line) in the contaminated part of the database and the MSE (solid blue line). The minimum and maximum allowable numbers of retained modes considered in Equation (3.17) are indicated with vertical dashed blue lines and the optimum number of retained modes is indicated with a vertical blue line. Last plot: evolution of RMSE and MaxE, $\text{RMSE}_{\text{clean}}$ (solid red line) defined in Equation (3.19) and MSE (solid blue line) defined in Equation (3.21) along the 11 iterations in Table 3.2

3.4.2 Systematic errors errors at randomly chosen database positions

2D database

Let us now introduce systematic errors by just multiplying by 1.5 the database values at 22% randomly located positions. The counterparts of Table 3.1 and Figure 3.5 are given in Table 3.3 and Figure 3.8, respectively.

As can be seen, the performance of the method is similar to that in the former case but, for the optimal performance of the method, the amount of contaminated positions is smaller. This is because systematic errors are more difficult to handle.

As expected, if the number of contaminated positions increases, the performance of the method worsens and, beyond 30% contamination, the method does not converge. On the other hand, as also expected, if the number of contaminated positions is increased beyond 78%, the method identifies as contaminated positions the clean positions of the database and converges to the

Iter.	CDEP	IDEP	m	$\text{MaxE}_{\text{cont}}$	$\text{RMSE}_{\text{cont}}$	T (sec)
1	109	0	5	$1.09 \cdot 10^{-1}$	$3.84 \cdot 10^{-2}$	0.1
2	251	0	4	$9.47 \cdot 10^{-2}$	$2.69 \cdot 10^{-2}$	0.0
3	468	0	4	$8.91 \cdot 10^{-2}$	$1.29 \cdot 10^{-2}$	0.1
4	518	0	4	$4.44 \cdot 10^{-2}$	$8.55 \cdot 10^{-3}$	0.0
5	533	0	7	$5.96 \cdot 10^{-2}$	$5.96 \cdot 10^{-3}$	0.0
6	544	0	10	$1.54 \cdot 10^{-2}$	$1.23 \cdot 10^{-3}$	0.2
7	547	0	13	$3.00 \cdot 10^{-3}$	$2.08 \cdot 10^{-4}$	0.4
8	549	0	15	$2.76 \cdot 10^{-3}$	$2.63 \cdot 10^{-4}$	0.5
9	550	0	15	$7.46 \cdot 10^{-7}$	$3.50 \cdot 10^{-8}$	0.4

TABLE 3.3: Counterpart of Table 3.1 for the contaminated database obtained by multiplying by 1.5 the database elements at 22% randomly located points (namely, 550 points) in the dense toy model database.

FIGURE 3.8: From left to right, downwards: the contaminated database and the outcomes of the first five iterations considered in Table 3.3; the outcomes of the remaining iterations are plot-indistinguishable from the clean database.

database 1.5A.

For the coarse database, the performance of the cleaning-SVD method is slightly worse but still quite good. Namely, 10% systematic, randomly located error positions are filtered out in 6 iterations, retaining 15 SVD modes and showing $\text{RMSE} \sim 10^{-6}$ and $\text{MaxE} \sim 10^{-5}$ in the last iteration.

3D database

Let us now introduce systematic errors by just multiplying by 1.5 the database values at 40% randomly located positions.

Iter.	CDEP	IDEP	m_1	m_2	m_3	m	$\text{MaxE}_{\text{cont}}$	$\text{RMSE}_{\text{cont}}$	T (sec)
1	11420	0	9	4	5	18	$1.39 \cdot 10^{-1}$	$3.13 \cdot 10^{-2}$	4.7
2	49686	0	7	5	11	23	$3.39 \cdot 10^{-2}$	$1.96 \cdot 10^{-3}$	5.2
3	49994	0	13	9	11	33	$1.58 \cdot 10^{-3}$	$1.66 \cdot 10^{-5}$	6.1
4	49998	0	16	10	16	42	$3.88 \cdot 10^{-3}$	$2.07 \cdot 10^{-5}$	10.5
5	50000	0	19	13	18	50	$4.47 \cdot 10^{-9}$	$9.01 \cdot 10^{-11}$	12.0

TABLE 3.4: Counterpart of Table 3.3 for the three-dimensional toy model database, contaminated by multiplying by 1.5 the database elements at 40% randomly chosen positions, namely 50000 positions.

Table 3.4 shows that the number of error points is much larger than in the two-dimensional case and the number of required iterations is smaller. And, as in the previous subsection, the method identifies very well the error positions.

As in the former subsection, if the number of error positions is increased further, the method identifies as error positions points that are correct, and the reconstruction errors worsens. Increasing the number of error position too much, beyond 60%, the method converges to the database obtained by multiplying by 1.5 all database elements, which was to be expected and illustrates well the underlying essence of the method.

As in the previous case, for the coarse database, the performance of the cleaning-HOSVD method is slightly worse but still quite good. Namely, 20% systematic, randomly located error positions are filtered out in 4 iterations, retaining 50 HOSVD modes and showing $\text{RMSE} = 5.28 \cdot 10^{-9}$ and $\text{MaxE} = 9.78 \cdot 10^{-8}$ in the last iteration.

3.4.3 Random errors at concentrated database positions

2D database

The database is contaminated now with a positive, random error of 0.5 maximum size in a centred 0.2-diameter circular cylinder, resulting in 76 erroneous points. This is a more challenging situation as clean information is not homogeneously spread along the database. Moreover, there is no clean data in the affected part. Table 3.5, which is a counterpart of Table 3.1, shows 7 iterations in which the method correctly identifies all the 76 points.

As in the previous cases, the same type of error was applied to the coarse database. Namely, a positive, random noise of 0.5 maximum size in a centred 0.2-diameter circle was added to the clean database. All the 12 errors were cleaned in 4 iterations, retaining 14 SVD and showing $\text{RMSE} \sim 10^{-9}$ and $\text{MaxE} \sim 10^{-9}$ in the last iteration.

Iter.	CDEP	IDEP	m	$\text{MaxE}_{\text{cont}}$	$\text{RMSE}_{\text{cont}}$	T (sec)
1	33	0	3	$5.90 \cdot 10^{-2}$	$3.60 \cdot 10^{-2}$	0.1
2	52	0	4	$3.37 \cdot 10^{-2}$	$1.57 \cdot 10^{-2}$	0.1
3	66	0	7	$8.05 \cdot 10^{-2}$	$2.43 \cdot 10^{-2}$	0.1
4	74	0	9	$2.90 \cdot 10^{-2}$	$4.93 \cdot 10^{-3}$	0.2
5	76	0	12	$2.58 \cdot 10^{-9}$	$1.09 \cdot 10^{-9}$	0.3

TABLE 3.5: Counterpart of Table 3.1 for the database contaminated with a positive, random noise of 0.5 maximum size in a centred 0.2-diameter circle

FIGURE 3.9: From left to right, downwards: the contaminated database and the outcomes of the five iterations considered in Table 3.5.

3D database

Similarly to the previous section, the database elements are now contaminated with a positive, random error of 0.5 maximum size in a black-box, centred 0.315-diameter circular cylinder ($x^2 + y^2 \leq 0.1575^2, 0 \leq z \leq 1$), which contains 9400 points. Results are given in Table 3.6, where it can be seen that the method identifies all error points in eight iterations.

As in the previous cases, for the coarse database, the performance of the cleaning-HOSVD method is slightly worse but still quite good. Namely, random, positive errors located in a 0.2-diameter circular cylinder, which is smaller than in the previous case, are filtered out in 6 iterations, retaining 50 HOSVD modes and showing $\text{RMSE} = 7.04 \cdot 10^{-10}$ and $\text{MaxE} = 3.66 \cdot 10^{-9}$ in the last iteration.

Iter.	CDEP	IDEP	m_1	m_2	m_3	m	$\text{MaxE}_{\text{cont}}$	$\text{RMSE}_{\text{cont}}$	T (sec)
1	6054	0	6	4	5	15	$6.03 \cdot 10^{-2}$	$2.87 \cdot 10^{-2}$	5.2
2	8173	0	7	4	9	20	$8.43 \cdot 10^{-3}$	$4.06 \cdot 10^{-3}$	6.4
3	9009	0	7	5	9	21	$2.78 \cdot 10^{-3}$	$7.96 \cdot 10^{-4}$	8.3
4	9282	0	10	7	10	27	$7.22 \cdot 10^{-4}$	$1.75 \cdot 10^{-4}$	10.6
5	9371	0	12	8	10	30	$1.53 \cdot 10^{-4}$	$2.32 \cdot 10^{-5}$	17.6
6	9394	0	13	8	13	34	$1.97 \cdot 10^{-5}$	$3.99 \cdot 10^{-6}$	16.7
7	9399	0	17	11	16	44	$1.02 \cdot 10^{-4}$	$4.67 \cdot 10^{-6}$	15.9
8	9400	0	19	13	18	50	$7.20 \cdot 10^{-7}$	$8.12 \cdot 10^{-8}$	30.4

TABLE 3.6: Counterpart of Table 3.5 for the three-dimensional toy model database, with a positive, random noise of 0.5 in a centered 0.315-diameter circular cylinder.

3.4.4 Systematic errors at concentrated database positions

2D database

Let us now multiply by 1.5 the database elements in a centred 0.16-diameter circular box, which contains 52 points. This is the worst possible situation because both the error location and magnitude are systematic, meaning that they exhibit their own redundancies. Moreover, the redundancies associated with the errors are very close to the clean database redundancies, the only difference is the factor 1.5.

Iter.	CDEP	IDEP	m	$\text{MaxE}_{\text{cont}}$	$\text{RMSE}_{\text{cont}}$	T (sec)
1	52	0	6	$2.71 \cdot 10^{-5}$	$1.70 \cdot 10^{-5}$	0.1
2	52	5	9	$6.98 \cdot 10^{-8}$	$3.53 \cdot 10^{-8}$	0.1
3	52	7	12	$5.70 \cdot 10^{-9}$	$1.20 \cdot 10^{-9}$	0.3
4	52	8	15	$1.41 \cdot 10^{-8}$	$3.29 \cdot 10^{-9}$	0.4

TABLE 3.7: Counterpart of Table 3.1 for systematically contaminated points in the circular box.

FIGURE 3.10: From left to right: the contaminated database and the outcome of the first iteration considered in Table 3.7; the outcomes of the remaining iterations are plot-indistinguishable from the clean database.

Table 3.7 shows that the method localizes all contaminated points in just one iteration. In the remaining iterations, the method reconstructs several clean points, but does this quite well, namely the reconstruction errors are quite small.

As in the previous cases, for the coarse database, the performance of the cleaning-SVD method is slightly worse but still quite good. Namely, systematic errors located in a 0.2-diameter circular box are filtered out in 4 iterations, retaining 14 SVD modes and showing $\text{RMSE} \sim 10^{-9}$ and $\text{MaxE} \sim 5 \cdot 10^{-9}$ in the last iteration.

3D database

Now, we multiply by two those database elements in a black-box, centred 0.32-diameter circular cylinder ($x^2 + y^2 \leq 0.16^2, 0 \leq z \leq 1$), which contains 9600 points. The results are given in Table 3.8, where it can be seen that the method identifies all error points in the first iteration. The remaining iterations just improve the accuracy by retaining a larger number of HOSVD modes.

Iter.	CDEP	IDEP	m_1	m_2	m_3	m	$\text{MaxE}_{\text{cont}}$	$\text{RMSE}_{\text{cont}}$	T (sec)
1	9600	0	6	9	10	25	$4.71 \cdot 10^{-4}$	$7.00 \cdot 10^{-5}$	3.4
2	9600	33	8	13	13	34	$2.70 \cdot 10^{-5}$	$4.03 \cdot 10^{-6}$	6.8
3	9600	73	11	17	16	44	$2.39 \cdot 10^{-6}$	$2.65 \cdot 10^{-7}$	13.5
4	9600	125	12	18	17	47	$4.16 \cdot 10^{-6}$	$3.98 \cdot 10^{-7}$	18.1
5	9600	144	13	18	19	50	$4.72 \cdot 10^{-6}$	$4.66 \cdot 10^{-7}$	18.3
6	9600	182	13	18	19	50	$4.47 \cdot 10^{-6}$	$4.49 \cdot 10^{-7}$	19.0

TABLE 3.8: Counterpart of Table 3.7 for the three-dimensional toy model database, contaminated by multiplying by 1.5 the database elements in a centered 0.32-diameter circular cylinder.

As in the previous cases, for the coarse database, the performance of the cleaning-SVD method is slightly worse but still quite good. Namely, systematic errors located in a 0.30-diameter circular cylinder are filtered out in 5 iterations, retaining 18 SVD modes and showing $\text{RMSE} = 3.96 \cdot 10^{-6}$ and $\text{MaxE} = 6.92 \cdot 10^{-5}$ in the last iteration.

3.5 Conclusions

Summarizing the applications above:

- The cleaning SVD and HOSVD methods are able to identify and clean noisy databases, with either random or systematic $O(1)$ noise, either spread along the database or concentrated in a localized region.

- The method works in a fairly natural and robust way in all considered cases.
- As expected, systematic errors are more difficult to handle than random errors. The present method is able to filter systematic errors that would be not accessible to standard error filtering methods.

The performance of the method in the three-dimensional toy model database is qualitatively similar to its counterpart, using SVD, for the two-dimensional database. However, the performance is quantitatively better than in the two-dimensional case, namely the allowed number of contaminated positions is larger and the required number of iterations is smaller. This is because the three-dimensional database exhibits a larger number of redundancies along the three dimensions, which are taken advantage of by the method. If, instead, as explained at the end of Section 2.2, the tensor were reduced to a matrix, treating together two of the indexes, the performance of the method would be quite similar to its counterpart for the two-dimensional case considered in the last section.

Chapter 4

Application to a CFD-generated aerodynamic database

The cleaning-HOSVD method developed and illustrated in the previous section is now applied to two aerodynamic databases obtained through CFD analysis. In the analytical database the data patterns were governed directly by algebraic mathematical formulas. In this case, those patterns are governed by the physics of the problem, namely the Navier-Stokes equations that describe the characteristics of the flow. Analogically, the same type of errors will be applied in order to compare both databases in terms of the method functioning and reflect some of the difficulties expected in engineering applications.

4.1 Model definition

The CFD aerodynamic database contains the pressure coefficient in a structured mesh along surface of a swept wing for 12 equispaced values of the angle of attack in the range

$$-5^\circ \leq \alpha \leq 6^\circ. \quad (4.1)$$

The wing geometry is was modelled using an example taken from [45]. Specifically, the wing exhibits the following properties (see Figure 4.1-left): taper ratio=0.4, aspect ratio=9, washout angle=2°, quarter chord sweep angle=0°, and cross section aerofoil=NACA 64-210. The pressure distributions correspond to the steady, incompressible, turbulent flow around a swept wing, with a Reynolds number based on the mean aerodynamic chord $Re = 4.4 \cdot 10^6$. The CFD calculations are generated using OpenFOAM as CFD solver and Pointwise as mesh generator. Specifically:

- The superficial mesh is structured and constructed along a the curvilinear system, in terms of the coordinates s and y along the chord and the span directions, respectively. The mesh exhibits 300×450 grid points in the chord/span directions, which are concentrated in the leading edge zone, where the pressure coefficient is quite steep; see Figure 4.1-right, which also shows that the mesh is fairly fine.

FIGURE 4.1: The CFD-generated aerodynamic database. Left: a three-dimensional view of the wing. Right: The pressure coefficient in the CFD mesh for $\alpha = -5^\circ$, considering only one half of the mesh points in the s -direction and one fourth in the y -direction. See Figure 4.2 for the flow visualization around the root and the tip.

- In order to model turbulence the Spalart-Allmaras one-equation model [39, 46], implemented in OpenFOAM, was used. The choice was made based on two main reasons. Firstly, due to its simplicity, resources efficiency and reliability is one of the mostly used turbulence models in aeronautical engineering. It is suitable for solving relatively simple external/internal problems, although it performs poorly for separated flows [47]. It is used together with Wall Functions [48] to approximate the flow near the wall and hence reduce the computation time. Secondly, more sophisticated two-equation turbulence models such as the $k-\omega$ model suffer from so called stagnation point anomaly [49]. In the case of wing analysis, this would cause erroneous pressure values of $O(1)$ in the proximity of the leading edge that could have greatly hindered the performance of the developed method.
- Concerning the volumetric mesh, the boundary layer is extruded from the superficial mesh (using hexahedral cells) keeping the parameter y^+ in the range $30 < y^+ < 300$, defined by $y^+ = u_* d / \nu$ (where u_* is the friction velocity, d is the distance from the wall and ν is the kinematic viscosity), for the first cell layer. This assures that the first cell is in the log-law region inner layer which allows to apply wall functions [50]. As can be seen in Figure 4.2, it is not an easy task and there is a small region on the wing where the value of y^+ falls below 30. Finding the correct distribution on d across the whole surface of the wing is an iterative task and is not easily accomplished. To have a general idea of characteristics of the boundary layer, XFOIL 2D viscous aerofoil analysis program [51] was used to get an initial value, which was then updated at every run of the CFD analysis. The remaining part of the mesh is unstructured, using tetrahedral cells, keeping the mesh fairly fine in the proximity of the wing and gradually

lowering the mesh-density towards the boundaries. The wake zone is also refined. The total number of cells is $10.9 \cdot 10^6$.

- The artificial boundaries of the computational domain are placed far enough to simulate free flight. The calculation time is reduced by using a symmetry plane. The CFD analysis are performed using simpleFoam solver with second order schemes for spatial discretization [52]. They are terminated when both the variable residuals and aerodynamic coefficients reach a stable value.
- Finally, in order to check mesh convergence the grid convergence index (GCI) was calculated based on the calculated lift force [53]. The calculated value of 0.26% means very good grid convergence.

The full database giving the pressure coefficient at the wing surface consists in a third order tensor whose size along the chord/span/angle-of-attack directions is $300 \times 450 \times 12$. Applying the cleaning-HOSVD method to this database yields very good results that are omitted for the sake of brevity. Instead, we consider two coarser databases that somewhat mimic typical experimental databases resulting from wind tunnel tests, in which the number of mesh points is limited. The data in these databases is extracted from the original database, considering a smaller number of mesh points. Specifically, the sizes of these reduced-size databases are

- **Database #1:** $100 \times 20 \times 12$.
- **Database #2:** $25 \times 5 \times 12$.

Now the counterparts of Figure 4.1-right for these two databases are given in Figure 4.4, which illustrate the selected sub-meshes. Note that the second database is quite coarse. In the applications below to these aerodynamic databases, the tunable parameters of the cleaning-HOSVD method are

$$CF = 0.1, \quad s_{\min} = 4, \quad FF = 0.4, \quad N^{\min} = 5, \quad N^{\max} = 25, \quad N^- = 1, \quad N^+ = 5. \quad (4.2)$$

Note that, to emphasize robustness, the tunable parameters are maintained in all applications below and, moreover, are very similar to those selected for the toy model databases in the previous sections.

As in the toy model databases, we test the performance of the cleaning-HOSVD method, contaminating the aerodynamic databases in various fashions. In these applications, we keep in mind typical experimental wind tunnel databases, in which errors resulting from ill-functioning of sensors (or groups of sensors) are $O(1)$. The number of error positions in these experimental database is usually smaller than their counterparts considered below.

FIGURE 4.2: Distribution of $y+$ parameter (blue and green line for -5° and 6° angle of attack respectively) and the first cell centroid distance from the wall d (red line) along the equidistant s_i coordinate. Top, middle and bottom correspond to y coordinate equal to 0, 2.2 and 4.4m respectively. The coordinates $s_i < 150$, $s_i = 150$ and $s_i > 150$ correspond to the upper surface, leading edge and lower surface respectively.

4.2 Error filtering

4.2.1 Random errors at randomly located positions

To begin with, we contaminate the aerodynamic database #1 with uniformly distributed, positive random noise of size 0.5 at 40% randomly located points (namely, 9600 points). The performance of the cleaning-HOSVD method is

FIGURE 4.3: CFD flow visualizations for $\alpha = -5^\circ$. Top: flow velocity for the wing root (right) and wing tip (left) with corresponding streamlines. Bottom: C_P distribution for the wing root (left) and wing tip (right) with the corresponding isobars. Flow separation at the trailing edge can be clearly visible for the suction side of the tip.

FIGURE 4.4: Counterparts of Figure 4.1-right for the databases #1 (left) and #2 (right).

summarized in table 4.1, which shows the outcomes of the 7 iterations allowed by condition (3.19) and (3.21).

As can be seen, the performance of the method is as good as in the toy model databases. In particular, the method identifies and reconstructs 3774 contaminated points in the first iteration, 8610 points in the fourth iteration, and 9463 points in the last iteration. The remaining $9600 - 9463 = 137$ points exhibit errors that are smaller than $6.1 \cdot 10^{-3}$ and cannot be identified with the considered

Iter	CDEP	IDEP	m_1	m_2	m_3	m	$\text{MaxE}_{\text{cont}}$	$\text{RMSE}_{\text{cont}}$	T (sec)
1	3774	0	3	1	6	10	$2.46 \cdot 10^{-1}$	$5.07 \cdot 10^{-2}$	0.7
2	6181	0	3	3	6	12	$9.76 \cdot 10^{-2}$	$2.24 \cdot 10^{-2}$	0.9
3	7708	0	5	3	8	16	$4.05 \cdot 10^{-2}$	$8.34 \cdot 10^{-3}$	1.2
4	8610	0	5	4	10	19	$2.30 \cdot 10^{-2}$	$4.08 \cdot 10^{-3}$	1.6
5	9123	0	8	5	10	23	$1.56 \cdot 10^{-2}$	$1.85 \cdot 10^{-3}$	1.9
6	9320	0	10	6	8	24	$1.21 \cdot 10^{-2}$	$1.10 \cdot 10^{-3}$	1.6
7	9463	0	12	6	7	25	$6.10 \cdot 10^{-3}$	$8.43 \cdot 10^{-4}$	1.5

TABLE 4.1: Application of the cleaning HOSVD method to the aerodynamic database #1 contaminated with a positive, random noise of 0.5 size at 40% randomly chosen points: outcomes (namely, numbers of correctly and incorrectly determined error points, CDEP and IDEP, respectively, number of retained modes m , reconstruction errors defined in Equation (2.18), and required CPU time T (in seconds).

retained modes. The number of retained modes overall increases along the various iterations, but such increase is not necessarily monotonous since, e.g., the number of retained modes in the third dimension (angle of attack) first increases (to 10, in the 4-th iteration) and then decreases (to 7 in the 7-th iteration).

The evolution of the reconstructed database along the iterations is illustrated in Figure 4.5, which shows that this evolution is fairly similar to its counterparts for the toy model database.

FIGURE 4.5: From left to right, downwards: the $\alpha = -5^\circ$ -section of the contaminated database (top, left) and the outcomes of the first five iterations considered in Table 4.1; the outcomes of the remaining iterations are plot-indistinguishable from the clean database.

Now, in order to illustrate how each individual gappy reconstruction method

(at the iterations considered in Figure 4.5) works, Figure 4.6 shows the evolution with the number of retained modes of both the errors in the contaminated part of the database and MSE defined in Equation (3.14). This was previously done for the toy model in Section 3.4.1 but is worth repeating as the nature of the databases is different now. Once again, MSE selects well the optimal number of modes in each iteration. The last plot in this figure shows the evolution of the reconstruction errors along 8 iterations of the method and shows that the selected number of iterations, 7, is the correct since a further 8-th iteration would increase the reconstruction error. In fact, the 8-th iteration does not identify any new erroneous points, but despite it selects 131 clean points which drastically increases the reconstruction errors. This can be clearly seen in Figure 4.6. In other words, the stopping condition (3.19) and (3.21) work well.

Iter	CDEP	IDEP	m_1	m_2	m_3	m	$\text{MaxE}_{\text{cont}}$	$\text{RMSE}_{\text{cont}}$	T (sec)
1	191	0	3	4	3	10	$9.69 \cdot 10^{-2}$	$2.86 \cdot 10^{-2}$	0.2
2	273	0	5	5	4	14	$7.54 \cdot 10^{-2}$	$1.49 \cdot 10^{-2}$	0.2
3	316	0	7	5	6	18	$6.84 \cdot 10^{-2}$	$1.11 \cdot 10^{-2}$	0.2
4	337	0	10	5	7	22	$8.68 \cdot 10^{-2}$	$9.88 \cdot 10^{-3}$	0.3
5	362	0	11	5	8	24	$4.49 \cdot 10^{-2}$	$6.29 \cdot 10^{-3}$	0.5
6	369	0	12	5	8	25	$3.66 \cdot 10^{-2}$	$3.67 \cdot 10^{-3}$	0.5
7	373	1	12	5	7	24	$1.65 \cdot 10^{-2}$	$2.15 \cdot 10^{-3}$	0.4
8	375	4	12	5	6	23	$1.53 \cdot 10^{-2}$	$2.09 \cdot 10^{-3}$	0.5
9	375	7	13	5	7	25	$1.36 \cdot 10^{-2}$	$2.09 \cdot 10^{-3}$	0.4
10	375	14	13	5	7	25	$1.45 \cdot 10^{-2}$	$2.38 \cdot 10^{-3}$	0.5
11	375	16	13	5	7	25	$1.46 \cdot 10^{-2}$	$2.73 \cdot 10^{-3}$	0.4

TABLE 4.2: Counterpart of table 4.1 for the aerodynamic database # 2, but considering 25% randomly contaminated positions (namely, 450 positions).

The coarser aerodynamic database #2 is considered in Table 4.2, which shows that, as expected, the performance of the method is worse than in the denser database #1. In particular, the number of iterations and the reconstruction errors are both larger than in Table 4.1. In addition, 16 clean positions are wrongly identified as contaminated, which did not occur in Table 4.1, and 125 contaminated points are not identified. In spite of this, the final reconstructed database (after the 11 iterations allowed by the stopping conditions (3.19) and (3.21)) is plot indistinguishable from its clean counterpart.

4.2.2 Systematic errors at randomly located positions

The case systematic errors (resulting from multiplying by 1.5 the database elements) at randomly chosen positions for the aerodynamic databases # 1 and #2 is considered in Table 4.3 and 4.4, respectively. For the denser database #1 (Table 4.3), 6541 of the 8400 contaminated points are identified in the first iteration,

FIGURE 4.6: From left to right, downwards. First five plots: the performance of the gappy reconstruction method in the first five iterations of the cleaning-HOSVD method considered in Table 4.1, plotting the RMSE (dashed black line) and MaxE (solid black line) in the contaminated part of the database and the MSE (solid blue line). The minimum and maximum allowable numbers of retained modes considered in Equation (3.17) are indicated with vertical dashed blue lines and the optimum number of retained modes is indicated with a vertical blue line. Last plot: evolution of RMSE and MaxE, $\text{RMSE}_{\text{clean}}$ (solid red line) defined in Equation (3.19) and MSE (solid blue line) defined in Equation (3.21) along the 11 iterations in Table 4.1.

Iter	CDEP	IDEP	m_1	m_2	m_3	m	$\text{MaxE}_{\text{cont}}$	$\text{RMSE}_{\text{cont}}$	T (sec)
1	6541	0	3	1	5	9	$2.54 \cdot 10^{-1}$	$2.66 \cdot 10^{-2}$	0.8
2	8118	0	5	4	5	14	$6.91 \cdot 10^{-2}$	$6.28 \cdot 10^{-3}$	1.2
3	8309	0	5	5	5	15	$3.35 \cdot 10^{-2}$	$3.80 \cdot 10^{-3}$	1.4
4	8351	0	6	5	4	15	$1.86 \cdot 10^{-2}$	$2.68 \cdot 10^{-3}$	1.4
5	8392	0	10	5	5	20	$8.89 \cdot 10^{-3}$	$1.16 \cdot 10^{-3}$	1.5
6	8396	0	12	6	7	25	$7.30 \cdot 10^{-3}$	$8.04 \cdot 10^{-4}$	1.9

TABLE 4.3: Counterpart of table 4.1 for the aerodynamic database # 1, contaminated by multiplying by 1.5 the pressure coefficient at 35% randomly located database positions.

while in the 6-th iteration 8396 points are identified and corrected. The remaining 4 points are not identified because they show an error that is smaller than $7.3 \cdot 10^{-3}$. The counterpart of Figure 4.5 is omitted for the sake of brevity. We just point out that the final database (after 6 iterations) is plot-indistinguishable from the clean database.

FIGURE 4.7: From left to right, downwards: the $\alpha = -5^\circ$ -section of the contaminated database (top, left) and the outcomes of the first five iterations considered in Table 4.3; the outcome of the remaining iteration is plot-indistinguishable from the clean database.

For the coarser aerodynamic database #2 (Table 4.4), the performance of the method is slightly worse, but still fairly good.

Iter	CDEP	IDEP	m_1	m_2	m_3	m	$\text{MaxE}_{\text{cont}}$	$\text{RMSE}_{\text{cont}}$	T (sec)
1	396	1	4	3	3	10	$3.81 \cdot 10^{-1}$	$2.80 \cdot 10^{-2}$	0.2
2	433	1	5	4	4	13	$5.08 \cdot 10^{-2}$	$9.84 \cdot 10^{-3}$	0.2
3	443	1	7	5	5	17	$9.25 \cdot 10^{-2}$	$9.62 \cdot 10^{-3}$	0.3
4	448	1	9	5	6	20	$2.37 \cdot 10^{-2}$	$3.59 \cdot 10^{-3}$	0.4
5	450	1	13	5	7	25	$1.72 \cdot 10^{-2}$	$2.07 \cdot 10^{-3}$	0.4
6	450	4	13	5	7	25	$1.33 \cdot 10^{-2}$	$1.99 \cdot 10^{-3}$	0.5

TABLE 4.4: Counterpart of Table 4.3, with systematic errors at 30% randomly located positions.

4.2.3 Random errors at concentrated positions

Let us now consider a case in which the clean database is contaminated with random, positive errors of 0.5 maximum magnitude which are concentrated in the centre of the domain forming an elliptical cylinder spanning the angle of attack direction. As before, this error will be applied to the aerodynamic databases #1 and #2. This is a more demanding situation as the error points are located in an near the leading edge, where steep gradients are present. Specifically, for the aerodynamic database #1, we consider the elliptic cylinder $s^2/0.0034^2 + (y - 2.2)^2/0.34^2 = 1$, $-5^\circ \leq \alpha \leq 6^\circ$, which contains 444 of the 24000 points in the database.

Iter	CDEP	IDEP	m_1	m_2	m_3	m	$\text{MaxE}_{\text{cont}}$	$\text{RMSE}_{\text{cont}}$	T (sec)
1	236	0	3	3	3	9	$7.95 \cdot 10^{-2}$	$3.05 \cdot 10^{-2}$	0.6
2	342	0	5	3	3	11	$3.51 \cdot 10^{-2}$	$1.11 \cdot 10^{-2}$	1.3
3	404	0	7	4	4	15	$1.68 \cdot 10^{-2}$	$3.90 \cdot 10^{-3}$	1.2
4	423	73	7	4	3	14	$5.97 \cdot 10^{-2}$	$1.61 \cdot 10^{-2}$	1.6
5	438	88	9	5	5	19	$5.63 \cdot 10^{-2}$	$8.75 \cdot 10^{-3}$	1.8
6	444	110	10	6	6	22	$5.98 \cdot 10^{-2}$	$9.40 \cdot 10^{-3}$	2.4
7	444	142	13	6	6	25	$6.19 \cdot 10^{-2}$	$8.68 \cdot 10^{-3}$	2.2

TABLE 4.5: Counterpart of Table 4.1 for random, positive errors located in an elliptic cylinder.

The outcomes of 7 iterations allowed by conditions (3.19) and (3.21) in the cleaning-HOSVD method are given in Table 4.5, which shows that the 444 contaminated points are identified and corrected. The method also considers 142 clean points to be erroneous. However, the reconstruction RMSE and MaxE are $8.68 \cdot 10^{-3}$ and $6.19 \cdot 10^{-2}$ respectively, which is acceptable.

The contaminated database and its reconstructions in the first five iterations are illustrated in Figure 4.8; the reconstructions in the subsequent two iterations are plot-indistinguishable from the clean counterparts. As can be seen, the method very quickly identifies and reconstructs the error positions.

FIGURE 4.8: From left to right, downwards: the $\alpha = -5^\circ$ -section of the contaminated database (top, left) and the outcomes of the first five) iterations considered in Table 4.5; the outcomes of the remaining two iterations are plot-indistinguishable from the clean database.

The counterpart of Table 4.5 for the coarser aerodynamic database #2, considering the elongated elliptical cylinder $s^2/0.0032^2 + (y - 2.2)^2/0.32^2 = 1$, $-5^\circ \leq \alpha \leq 6^\circ$, which contains 71 of the 1500 points in the database, is given in Table 4.6. The performance of the method is comparable with the denser database #1, since the errors are concentrated in a slightly smaller ellipse but

the method requires one iteration less and recognises only 4 clean points as erroneous. The reconstruction errors are of the same order, being slightly smaller in the case of coarse database #2.

Iter	CDEP	IDEP	m_1	m_2	m_3	m	$\text{MaxE}_{\text{cont}}$	$\text{RMSE}_{\text{cont}}$	T (sec)
1	42	0	3	3	3	9	$5.99 \cdot 10^{-2}$	$3.32 \cdot 10^{-2}$	0.2
2	60	0	5	3	3	11	$4.07 \cdot 10^{-2}$	$1.32 \cdot 10^{-2}$	0.3
3	67	1	7	4	4	15	$5.48 \cdot 10^{-2}$	$8.30 \cdot 10^{-3}$	0.3
4	68	1	9	5	5	19	$4.48 \cdot 10^{-2}$	$8.37 \cdot 10^{-3}$	0.4
5	71	3	10	5	6	21	$4.51 \cdot 10^{-2}$	$6.88 \cdot 10^{-3}$	0.5
6	71	4	13	5	7	25	$3.62 \cdot 10^{-2}$	$6.57 \cdot 10^{-3}$	0.6

TABLE 4.6: Counterpart of Table 4.5 for the coarser database #2.

4.2.4 Systematic errors at concentrated positions

Let us now consider the most demanding case in which the errors are systematic and concentrated in the aerodynamic databases #1 and #2. To this end, we address the most demanding situation in which the error points are located in an elongated region near the leading edge, where steep gradients are present. Specifically, for the aerodynamic database #1, we consider an elliptic cylinder $s^2/0.0035^2 + (y - 2.2)^2/0.35^2 = 1$, $-5^\circ \leq \alpha \leq 6^\circ$, which contains 708 of the 24000 points in the database, and multiply by 1.5 the database elements in this region.

Iter	CDEP	IDEP	m_1	m_2	m_3	m	$\text{MaxE}_{\text{cont}}$	$\text{RMSE}_{\text{cont}}$	T (sec)
1	606	0	3	3	3	9	$8.46 \cdot 10^{-2}$	$2.55 \cdot 10^{-2}$	0.5
2	683	0	5	3	3	11	$4.20 \cdot 10^{-2}$	$1.11 \cdot 10^{-2}$	0.7
3	701	0	7	4	4	15	$1.68 \cdot 10^{-2}$	$4.13 \cdot 10^{-3}$	0.8
4	708	0	8	5	5	18	$1.26 \cdot 10^{-2}$	$2.65 \cdot 10^{-3}$	0.9

TABLE 4.7: Counterpart of Table 4.1 for systematic errors located in an elliptic cylinder.

The outcomes of the 4 iterations allowed by conditions (3.19) and (3.21) in the cleaning-HOSVD method are given in Table 4.7, which shows that the 708 contaminated points are identified and corrected.

The reconstructions in the first two iterations are illustrated in Figure 4.9; the reconstructions in the subsequent two iterations are plot-indistinguishable from the clean counterparts. As can be seen, the method very quickly identifies and reconstructs the error positions.

The counterpart of Table 4.7 for the coarser aerodynamic database #2, considering the elongated elliptical cylinder $s^2/0.008^2 + (y - 2.2)^2/0.8^2 = 1$, $-5^\circ \leq \alpha \leq 6^\circ$, which contains 108 of the 1500 points in the database, is given in Table 4.8. Note that the performance of the method is worse than in the denser database # 1,

FIGURE 4.9: The $\alpha = -5^\circ$ -section of the contaminated database (left) and the outcomes of the first (middle) and second (right) iterations considered in Table 4.7.

Iter	CDEP	IDEP	m_1	m_2	m_3	m	$\text{MaxE}_{\text{cont}}$	$\text{RMSE}_{\text{cont}}$	T (sec)
1	83	0	3	2	3	8	$9.99 \cdot 10^{-2}$	$3.40 \cdot 10^{-2}$	0.2
2	96	0	4	3	3	10	$1.04 \cdot 10^{-1}$	$3.23 \cdot 10^{-2}$	0.3
3	102	0	7	5	3	15	$1.48 \cdot 10^{-1}$	$3.27 \cdot 10^{-2}$	0.3
4	104	0	10	5	5	20	$1.12 \cdot 10^{-1}$	$3.15 \cdot 10^{-2}$	0.4
5	106	0	12	5	6	23	$5.82 \cdot 10^{-2}$	$1.35 \cdot 10^{-2}$	0.4
6	107	0	13	5	7	25	$3.99 \cdot 10^{-2}$	$1.31 \cdot 10^{-2}$	0.3

TABLE 4.8: Counterpart of Table 4.7 for the coarser database #2.

since the method requires more iterations, fails to identify one error point, and exhibits a larger reconstruction error.

In the applications above, we considered the worst possible situation, in which the error points were precisely at the leading edge. If, instead, the error points are either in the pressure or suction sides of the wing, where the pressure distribution behaves more smoothly, the performance of the method improves. For instance, in the aerodynamic database # 1 the elliptical cylinder $(s + 0.5)^2/0.24^2 + (y - 2.2)^2/1.2^2 = 1, -5^\circ \leq \alpha \leq 6^\circ$ contains 1008 points (namely, a significant part of the pressure side). If the database values are multiplied by 1.5 in these positions, the cleaning-HOSVD method identifies the 1008 points in 4 iterations, with a final reconstruction errors $\text{MaxE}_{\text{cont}} = 1.25 \cdot 10^{-2}$ and $\text{RMSE}_{\text{cont}} = 1.55 \cdot 10^{-3}$; the method also identifies 182 points as contaminated. Similarly, contaminating the coarser aerodynamic database #2 in the elliptical cylinder $(s + 0.5)^2/0.24^2 + (y - 2.2)^2/1.4^2 = 1, -5^\circ \leq \alpha \leq 6^\circ$, which contains 60 erroneous points, the method identifies these 60 points (and another 32 clean points) in 2 iterations, and produces a final reconstruction with $\text{MaxE}_{\text{cont}} = 1.92 \cdot 10^{-2}$ and $\text{RMSE}_{\text{cont}} = 7.54 \cdot 10^{-3}$.

Chapter 5

Method performance analysis

So far in Chapter 3 and 4 the functioning of the method was illustrated and tested on single cases where two types of both gappyness and error were present:

- a) random, positive errors (non-zero mean) randomly distributed in the database;
- b) random, positive errors (non-zero mean) concentrated in a circle or ellipse, for toy model or CFD database respectively, with its centre located in the centre of the database;
- c) systematic errors randomly distributed in the database;
- d) systematic errors concentrated in a circle or ellipse, for toy model or CFD database respectively, with its centre located in the centre of the database,

Clearly, the performance of the filtering method will depend greatly on the randomness of the error magnitude and its distribution. More specifically, the case described in a) suffers from double randomness, error magnitude randomness and gappyness randomness, b) suffers from error magnitude randomness only and finally c) suffers from gappyness randomness. Case d) does not suffer from any randomness as the error is correlated with the database and the gappyness constrained by size parameter.

Consequently, for the cases where there is a factor of randomness, it is worthwhile to analyse a sample to see trends in the performance of the cleaning method. The findings of such analysis, including statistical parameters, are described in Section 5.1 where the error magnitude is fixed to the same value as in Chapter 3 and 4, but the gappyness size is varied.

Another interesting study is to quantify the size of error the method can detect. Once again, since this type of study greatly depends on randomness, similarly to Section 5.1, several cases will be run to obtain mean values and other useful statistical parameters. The findings will be described in Section 5.2. In this case the gappyness will be fixed to a level where the method performs well for $O(1)$ size errors, basing on the results from Section 5.1, and the error magnitude will be varied.

For both analyses the method will be applied to the 3D toy model database (dense and coarse) and CFD database (dense and coarse). The 2D toy model database will be omitted for the sake of brevity.

5.1 Randomness influence for varying contamination

For this analysis, the tunable parameters that appear in Equations (3.1), (3.5) and (3.7), have been maintained and are equal to the ones defined in Equation (3.26) and (4.2) for the 3D toy model and CFD database respectively. For each combination of error and contamination, the analysis was run 100 times for both databases in order to obtain a sufficiently large sample of results. The maximum number of allowed applications of the cleaning-HOSVD method, labelled k , has been set to 21 due to computational power restrictions. As will be seen later, this value is never reached.

5.1.1 Random errors at randomly chosen database positions

Toy model database

Similarly to Section 3.4, the dense toy model database is contaminated with positive, random error of maximum magnitude 0.5 at randomly chosen database points. The level of contamination has been varied between 32.5% to 90.0% in steps of 2.5%. The results are presented in Table 5.1.

In Section 3.4 the toy dense model was contaminated with positive, random error of maximum magnitude 0.5 at 65% of points, which was motivated by the fact, that for higher levels of contamination, the method can fail. It is interesting to see that for a 65% contamination level, the method has 25% chance that it will identify all the error points, without identifying any clean points as erroneous (see Table 5.1). For the sake of simplicity, in line with the nomenclature established in Section 3.4, correctly identified error points will be abbreviated to CDEP for correctly determined error points and clean points mistakenly taken as error point will be referred to as IDEP for incorrectly determined error points. From now on this type of convergence will be called *absolute convergence*. The probability might seem low; however it must be noted that the criterion is very rigorous. Considering that in reality a clean database to compare the cleaning results with is not available, from an engineering point of view a good convergence will happen when most of the erroneous points are identified and corrected. Some clean points can also be mistakenly taken as error points but under the condition that the reconstruction errors are satisfactory.

For the purpose of this performance analysis, the method is considered to perform satisfactory if 90% error points are detected and reconstruction MaxE and

%cont	%CD _m	%CD _{std}	%ID _m	%ID _{std}	k _m	k _{std}	%C ₁	%C ₂
32.5	100.0	$3.5 \cdot 10^{-4}$	0.0	$1.3 \cdot 10^{-3}$	9.7	$7.3 \cdot 10^{-1}$	94	100
35.0	100.0	$3.2 \cdot 10^{-4}$	0.0	$1.7 \cdot 10^{-3}$	9.8	$7.4 \cdot 10^{-1}$	91	100
37.5	100.0	$3.0 \cdot 10^{-4}$	0.0	$3.1 \cdot 10^{-3}$	10.0	$8.0 \cdot 10^{-1}$	82	100
40.0	100.0	$2.8 \cdot 10^{-4}$	0.0	$3.1 \cdot 10^{-3}$	10.1	$7.8 \cdot 10^{-1}$	81	100
42.5	100.0	$2.6 \cdot 10^{-4}$	0.0	$2.8 \cdot 10^{-3}$	10.2	$7.7 \cdot 10^{-1}$	81	100
45.0	100.0	$3.0 \cdot 10^{-4}$	0.0	$2.2 \cdot 10^{-3}$	10.2	$7.0 \cdot 10^{-1}$	80	100
47.5	100.0	$2.9 \cdot 10^{-4}$	0.0	$1.8 \cdot 10^{-3}$	10.3	$7.6 \cdot 10^{-1}$	79	100
50.0	100.0	$6.1 \cdot 10^{-4}$	0.0	$2.8 \cdot 10^{-3}$	10.5	$7.7 \cdot 10^{-1}$	69	100
52.5	100.0	$4.9 \cdot 10^{-4}$	0.0	$2.5 \cdot 10^{-3}$	10.6	$7.2 \cdot 10^{-1}$	68	100
55.0	100.0	$5.1 \cdot 10^{-4}$	0.0	$3.2 \cdot 10^{-3}$	10.8	$9.1 \cdot 10^{-1}$	57	100
57.5	100.0	$1.1 \cdot 10^{-3}$	0.0	$4.0 \cdot 10^{-3}$	10.8	$8.8 \cdot 10^{-1}$	54	100
60.0	100.0	$1.4 \cdot 10^{-3}$	0.0	$3.4 \cdot 10^{-3}$	10.8	$8.0 \cdot 10^{-1}$	49	100
62.5	99.9	$3.6 \cdot 10^{-1}$	0.0	$3.6 \cdot 10^{-3}$	10.7	$1.5 \cdot 10^0$	35	100
65.0	99.7	$7.5 \cdot 10^{-1}$	0.0	$3.3 \cdot 10^{-3}$	9.6	$2.6 \cdot 10^0$	25	100
67.5	99.2	$1.2 \cdot 10^0$	0.0	$1.3 \cdot 10^{-1}$	8.2	$3.1 \cdot 10^0$	16	98
70.0	97.9	$2.1 \cdot 10^0$	0.0	$2.0 \cdot 10^{-1}$	6.1	$3.0 \cdot 10^0$	3	97
72.5	96.1	$2.5 \cdot 10^0$	0.1	$3.6 \cdot 10^{-1}$	4.4	$1.4 \cdot 10^0$	0	87
75.0	94.8	$2.7 \cdot 10^0$	0.2	$4.0 \cdot 10^{-1}$	4.1	$7.7 \cdot 10^{-1}$	0	73
77.5	92.7	$4.0 \cdot 10^0$	0.2	$3.2 \cdot 10^{-1}$	4.1	$8.6 \cdot 10^{-1}$	0	65
80.0	89.5	$5.5 \cdot 10^0$	0.9	$7.7 \cdot 10^{-1}$	5.2	$1.1 \cdot 10^0$	0	31
82.5	82.4	$8.0 \cdot 10^0$	4.2	$1.3 \cdot 10^0$	5.3	$1.8 \cdot 10^0$	0	0
85.0	73.6	$7.2 \cdot 10^0$	6.8	$1.4 \cdot 10^0$	4.2	$1.1 \cdot 10^0$	0	0
87.5	70.1	$5.3 \cdot 10^0$	9.2	$1.2 \cdot 10^0$	3.9	$9.2 \cdot 10^{-1}$	0	0
90.0	65.9	$5.4 \cdot 10^0$	8.6	$6.5 \cdot 10^{-1}$	3.3	$7.5 \cdot 10^{-1}$	0	0

TABLE 5.1: Results of the dense 3D toy model database contaminated with positive, random error of maximum magnitude 0.5 at randomly chosen database points for varying contamination percentage. The meaning of the column headers is: %cont- contamination percentage; %CD_m- mean percentage of correctly detected error points (%CDEP) normalised with the total number of error points; %CD_{std}- standard deviation of %CDEP; %ID_m- mean percentage of incorrectly detected error points (%IDEP) normalised with the total number of error points; %ID_{std}- standard deviation of %IDEP; k_m- mean number of application of the method, k; k_{std}- standard deviation of k; %C₁- convergence percentage when the method correctly identifies all the error points without including any clean points (*absolute convergence*); %C₂- convergence percentage when the method identifies at least 90% of error points and the reconstruction MaxE and RMSE defined in Equation (2.18) are of order lower than 0.1 and 0.01 respectively (*relative convergence*).

RMSE, as defined in Equation (2.18), are lower than $O(-1)$ and $O(-2)$ respectively. From now on this type of convergence will be called *relative convergence*. From Table 5.1 one can see that the relative convergence is highly probable until 70.0% contamination level and then it decreases below 90% probability. The

limit when the relative convergence probability decreases below 90% from now on will be call *convergence limit*.

FIGURE 5.1: Complement of Table 5.1. Left: $\%CDEP_m$ with the corresponding $\%CDEP_{std}$ bars (thick black line) and $\%IDEP_m$ with the corresponding $\%IDEP_{std}$ bars (thin black line). Right: mean MaxE and RMSE of the reconstructed part (solid and dashed line respectively), as defined in Equation (2.18).

It is also interesting to note that on average the method achieves a 90% of CDEP for contamination levels as high as 77.5% (4% standard deviation). This can be clearly seen in Figure 5.1-left. However, the stop conditions defined in Equation (3.19) and (3.21) terminate the method on average in 4.1 method iterations which means that the reconstruction does not retain sufficiently many modes for MaxE and RMSE to be small (see Figure 5.1-right). At the same time the method achieves a very low percentage of IDEP (only 0.2% on average). This proves that the method is very robust in terms of error identification and would benefit from retaining more modes in the final database reconstruction.

The counterpart for the coarse database of Table 5.1 and Figure 5.1 is shown in Table A.1 and Figure A.1 respectively. In Section 3.4 the error contamination was set to 40% for the coarse database. The corresponding absolute convergence probability is 40%. In this case the convergence limit is also 50% contamination level for an average of 8.3 method application. It is worth noting that the method is capable of achieving above 90% of CDEP for contamination level up to 60%. However, if you compare Figure 5.1-right with Figure A.1-right, it is clear that the reconstruction errors MaxE and RMSE are much higher in this case and they stop fulfilling the convergence limit criterion at 55.7% error concentration. It is interesting to observe that the iteration number needed for convergence is similar to the dense database case. Additionally, the number of retained modes is similar and all the parameters defined in Equation (3.1), (3.5) and (3.7) are the same for both mesh densities meaning that the method depends weakly on the mesh density as it mainly depends on the patterns formed within it, whether they are due to mathematical formulas or physical phenomena.

CFD database

Chapter 4 is dedicated to applying the method to the CFD database. More specifically, Section 4.2 presents an analysis of the method functioning. Similarly to the toy model application, it starts with contaminating the dense CFD database with positive, random errors of 0.5 maximum magnitude at randomly distributed points. The error concentration for the mentioned analysis is 40% and the method succeeded in finding 9463 errors out of 9600 in total, in other words 98.6% CDEP, without taking any IDEP. In the analysis below, the error concentration was varied from 5.0 to 62.5% at 2.5% steps and the results are presented in Table 5.2.

%cont	%CD _m	%CD _{std}	%ID _m	%ID _{std}	k _m	k _{std}	%C ₁	%C ₂
5.0	98.6	$4.4 \cdot 10^{-1}$	2.2	$2.2 \cdot 10^0$	6.0	$9.9 \cdot 10^{-1}$	0	100
7.5	98.5	$4.4 \cdot 10^{-1}$	1.3	$2.2 \cdot 10^0$	5.8	$1.1 \cdot 10^0$	0	100
10.0	98.5	$5.0 \cdot 10^{-1}$	1.4	$2.8 \cdot 10^0$	6.0	$1.3 \cdot 10^0$	0	100
12.5	98.5	$5.2 \cdot 10^{-1}$	0.9	$1.5 \cdot 10^0$	6.1	$1.4 \cdot 10^0$	0	100
15.0	98.8	$5.5 \cdot 10^{-1}$	1.9	$1.9 \cdot 10^0$	7.3	$1.7 \cdot 10^0$	0	100
17.5	98.9	$4.6 \cdot 10^{-1}$	1.7	$1.6 \cdot 10^0$	7.7	$1.5 \cdot 10^0$	0	100
20.0	98.9	$4.5 \cdot 10^{-1}$	1.4	$9.5 \cdot 10^{-1}$	7.8	$1.4 \cdot 10^0$	0	100
22.5	98.9	$3.8 \cdot 10^{-1}$	1.1	$6.4 \cdot 10^{-1}$	8.0	$1.3 \cdot 10^0$	0	100
25.0	98.9	$4.0 \cdot 10^{-1}$	1.1	$6.1 \cdot 10^{-1}$	8.2	$1.5 \cdot 10^0$	0	100
27.5	98.9	$3.2 \cdot 10^{-1}$	0.9	$4.2 \cdot 10^{-1}$	8.3	$1.4 \cdot 10^0$	0	100
30.0	98.9	$3.2 \cdot 10^{-1}$	0.8	$4.2 \cdot 10^{-1}$	8.4	$1.2 \cdot 10^0$	0	100
32.5	98.8	$7.0 \cdot 10^{-1}$	0.6	$2.3 \cdot 10^{-1}$	8.0	$9.9 \cdot 10^{-1}$	0	100
35.0	98.5	$1.6 \cdot 10^0$	0.6	$2.5 \cdot 10^{-1}$	8.2	$1.2 \cdot 10^0$	0	99
37.5	98.2	$2.1 \cdot 10^0$	0.6	$3.5 \cdot 10^{-1}$	8.3	$1.6 \cdot 10^0$	0	97
40.0	95.9	$6.2 \cdot 10^0$	0.9	$2.0 \cdot 10^0$	7.2	$2.1 \cdot 10^0$	0	87
42.5	91.6	$8.7 \cdot 10^0$	2.7	$3.5 \cdot 10^0$	6.3	$2.3 \cdot 10^0$	0	68
45.0	83.5	$1.1 \cdot 10^1$	5.1	$3.3 \cdot 10^0$	5.0	$2.0 \cdot 10^0$	0	33
47.5	79.6	$9.3 \cdot 10^0$	6.0	$3.1 \cdot 10^0$	4.3	$1.5 \cdot 10^0$	0	15
50.0	76.3	$8.4 \cdot 10^0$	6.4	$2.4 \cdot 10^0$	4.2	$1.4 \cdot 10^0$	0	1
52.5	74.5	$6.3 \cdot 10^0$	7.0	$1.4 \cdot 10^0$	3.8	$9.9 \cdot 10^{-1}$	0	0
55.0	72.3	$6.2 \cdot 10^0$	6.6	$1.3 \cdot 10^0$	3.6	$9.2 \cdot 10^{-1}$	0	0
57.5	71.3	$6.4 \cdot 10^0$	6.7	$1.5 \cdot 10^0$	3.5	$1.1 \cdot 10^0$	0	0
60.0	70.3	$7.1 \cdot 10^0$	6.4	$1.3 \cdot 10^0$	3.4	$1.1 \cdot 10^0$	0	0
62.5	68.9	$7.1 \cdot 10^0$	5.8	$1.1 \cdot 10^0$	3.2	$7.6 \cdot 10^{-1}$	0	0

TABLE 5.2: Results of the dense CFD database contaminated with positive, random error of maximum magnitude 0.5 at randomly chosen database points for varying contamination percentage. For column headers refer to Table 5.1.

For 40% error concentration the mean probability of identifying all the error points is 95.9%. This error concentration is already below the convergence limit, as the chance of relative convergence is 87%. It is interesting to see that the method never converges absolutely. One of the reasons is that it is incapable

of identifying all the error points. The other is that the chance of taking IDEP oscillates about 1%, which also inhibits absolute convergence. This is typical for engineering databases, either obtained by numerical simulation or experimentally, where there is some intrinsic error present that in certain regions of the database can be of the order of synthetically added errors. This can be seen in Figure 5.2-right, where the reconstruction MaxE and RMSE for the range satisfying the relative convergence condition is 10^{-2} and 10^{-3} respectively. Nevertheless, from Table 5.2 it is clear that the method performs very well up to an error concentration level of 37.5%, requiring on average only 8.3 iterations to finish.

FIGURE 5.2: Complement of Table 5.2. Left: $\%CDEP_m$ with the corresponding $\%CDEP_{std}$ bars (thick black line) and $\%IDEP_m$ with the corresponding $\%IDEP_{std}$ bars (thin black line). Right: mean MaxE and RMSE of the reconstructed part (solid and dashed line respectively), as defined in Equation (2.18).

The counterpart for the coarse database of Table 5.2 and Figure 5.2 is shown in Table A.2 and Figure A.2 respectively. In Section 4.2 the error contamination was set to 25% for the coarse database. From Table A.2 one can see that absolute convergence is impossible, which is not a surprise as the dense CFD database has 0% probability of absolute convergence. However, the relative convergence has a 69% chance of occurring. For this case, the convergence limit is 20% which, as expected, is lower than for the case of the dense database (convergence limit equal to 37.5%). This is mainly due to high reconstruction errors (see Figure A.2-left), in particular RMSE, which do not satisfy the threshold established for the relative convergence.

5.1.2 Systematic errors at randomly chosen database positions

Toy model database

In this case the dense toy model database is contaminated with systematic errors obtained by multiplying the database by 1.5 at randomly chosen database

points, analogically to the analysis presented in Section 3.4. The level of contamination has been varied between 32.5% to 90.0% every 2.5%. The results are presented in Table 5.3.

%error	%CD _m	%CD _{std}	%ID _m	%ID _{std}	iter _m	iter _{std}	%C ₁	%C ₂
32.5	100.0	$1.1 \cdot 10^{-3}$	0.0	$1.2 \cdot 10^{-2}$	5.2	$4.3 \cdot 10^{-1}$	16	100
35.0	100.0	$1.0 \cdot 10^{-3}$	0.0	$1.2 \cdot 10^{-2}$	5.2	$4.1 \cdot 10^{-1}$	14	100
37.5	100.0	$1.0 \cdot 10^{-3}$	0.0	$1.6 \cdot 10^{-2}$	5.2	$3.9 \cdot 10^{-1}$	9	100
40.0	100.0	$9.7 \cdot 10^{-4}$	0.0	$4.9 \cdot 10^{-2}$	5.2	$4.5 \cdot 10^{-1}$	4	100
42.5	100.0	$1.3 \cdot 10^{-3}$	0.1	$8.0 \cdot 10^{-2}$	5.4	$6.4 \cdot 10^{-1}$	0	100
45.0	100.0	$2.2 \cdot 10^{-1}$	0.2	$3.2 \cdot 10^{-1}$	5.8	$8.5 \cdot 10^{-1}$	0	100
47.5	98.8	$4.2 \cdot 10^0$	2.6	$1.9 \cdot 10^0$	5.7	$1.9 \cdot 10^0$	0	55
50.0	84.0	$1.2 \cdot 10^1$	17.5	$5.9 \cdot 10^0$	5.6	$1.9 \cdot 10^0$	0	0
52.5	42.7	$9.0 \cdot 10^0$	54.5	$9.1 \cdot 10^0$	5.5	$1.7 \cdot 10^0$	0	0
55.0	17.7	$3.8 \cdot 10^0$	68.4	$3.2 \cdot 10^0$	5.3	$1.5 \cdot 10^0$	0	0
57.5	9.4	$3.5 \cdot 10^0$	68.1	$2.7 \cdot 10^0$	5.2	$1.3 \cdot 10^0$	0	0
60.0	5.1	$2.3 \cdot 10^0$	64.2	$1.3 \cdot 10^0$	5.6	$1.1 \cdot 10^0$	0	0
62.5	3.0	$1.7 \cdot 10^0$	58.8	$7.7 \cdot 10^{-1}$	5.2	$9.5 \cdot 10^{-1}$	0	0
65.0	1.9	$1.1 \cdot 10^0$	53.2	$4.2 \cdot 10^{-1}$	4.8	$9.9 \cdot 10^{-1}$	0	0
67.5	1.1	$7.4 \cdot 10^{-1}$	47.8	$2.4 \cdot 10^{-1}$	4.5	$8.7 \cdot 10^{-1}$	0	0
70.0	0.5	$6.0 \cdot 10^{-1}$	42.7	$1.6 \cdot 10^{-1}$	4.8	$9.5 \cdot 10^{-1}$	0	0
72.5	0.1	$2.3 \cdot 10^{-1}$	37.9	$5.3 \cdot 10^{-2}$	5.2	$6.9 \cdot 10^{-1}$	0	0
75.0	0.0	$9.4 \cdot 10^{-2}$	33.3	$3.1 \cdot 10^{-2}$	5.3	$6.0 \cdot 10^{-1}$	0	0
77.5	0.0	$5.3 \cdot 10^{-3}$	29.0	$4.6 \cdot 10^{-4}$	5.3	$5.6 \cdot 10^{-1}$	0	0
80.0	0.0	$6.0 \cdot 10^{-3}$	25.0	$4.2 \cdot 10^{-4}$	5.3	$6.1 \cdot 10^{-1}$	0	0
82.5	0.0	$6.1 \cdot 10^{-3}$	21.2	$3.9 \cdot 10^{-4}$	5.4	$6.7 \cdot 10^{-1}$	0	0
85.0	0.0	$6.4 \cdot 10^{-3}$	17.6	$3.5 \cdot 10^{-4}$	5.2	$7.0 \cdot 10^{-1}$	0	0
87.5	0.0	$6.8 \cdot 10^{-3}$	14.3	$3.1 \cdot 10^{-4}$	5.0	$7.6 \cdot 10^{-1}$	0	0
90.0	0.0	$7.9 \cdot 10^{-3}$	11.1	$2.6 \cdot 10^{-4}$	4.8	$7.3 \cdot 10^{-1}$	0	0

TABLE 5.3: Results of the dense 3D toy model database contaminated with a systematic error obtained by multiplying the database by 1.5 at randomly chosen database points for varying contamination percentage. For column headers please refer to Table 5.1.

In Section 3.4 the method was shown to converge absolutely for error concentration of 40%. However, from Table 5.3 it results that absolute convergence has a 4% of being accomplished. On the other hand, the percentage of CDEP and IDEP is 100% and 0.0% respectively. Analysing thoroughly the results revealed that the method identifies on average 17.4 IDEP which, as the total number of errors corresponding to 40% contamination is 50,000, becomes 0.0% IDEP. Additionally, the chance of identifying no IDEP is 6%. Nevertheless, the convergence limit in this case is 45% error concentration which is reasonable as the errors are correlated with the database. Hence, at 50% error concentration the method should behave erratically and converge at one time to the correct, clean database and at the other to the erroneous one. From Figure 5.3-left it

can be clearly seen, as the IDEP start to build up from just before 50%. They never reach 100% because they are normalized by the total number of erroneous points making IDEP tends towards 0 as the contamination level approaches 100%. This phenomenon is accompanied by a sharp rise of reconstruction errors just after 40% contamination. It is interesting to note that the number of method iterations does not vary much with respect to the contamination level and is about 5.

FIGURE 5.3: Complement of Table 5.3. Left: $\%CDEP_m$ with the corresponding $\%CDEP_{std}$ bars (thick black line) and $\%IDEP_m$ with the corresponding $\%IDEP_{std}$ bars (thin black line). Right: mean MaxE and RMSE of the reconstructed part (solid and dashed line respectively), as defined in Equation (2.18).

The counterpart for the coarse database of Table 5.3 and Figure 5.3 is shown in Table A.3 and Figure A.3 respectively. In Section 3.4 the error contamination was set to 20% for the coarse database. Table A.3 shows that for this error contamination the absolute convergence has a 8% chance of occurring. At the same time the maximum probability of absolute convergence is 12.5% for 17.5%. The table also shows that the method is 100% correct in identifying all the error points up to 37.5% contamination level and the reconstruction errors are considered low until about 40% contamination level (see Figure A.3-right). Consequently, the probability of relative convergence is high and the convergence limit is 37.5%. It is interesting to note that the method needs on average about 5 iterations to converge and it makes relatively little mistakes (below 1% IDEP until convergence limit).

CFD database

Section 4.2 tested the dense CFD database with systematic errors resulting from multiplying the database by 1.5 at randomly distributed points for 35% error concentration. As expected, this is slightly lower than for the random, positive error, as it is more difficult to clean correlated errors. The method in this case identified 8,396 errors, out of 8,400 in total, in other words 100.0% CDEP, without taking any IDEP. For the purpose of this more detailed analysis, the error

concentration was varied from 5.0 to 62.5% at 2.5% steps and the results are presented in Table 5.4.

%cont	%CD _m	%CD _{std}	%ID _m	%ID _{std}	k _m	k _{std}	%C ₁	%C ₂
5.0	99.9	$8.9 \cdot 10^{-2}$	1.8	$2.7 \cdot 10^0$	5.1	$9.2 \cdot 10^{-1}$	0	100
7.5	99.9	$7.5 \cdot 10^{-2}$	1.2	$2.0 \cdot 10^0$	5.3	$1.0 \cdot 10^0$	0	100
10.0	99.9	$6.3 \cdot 10^{-2}$	1.1	$2.7 \cdot 10^0$	5.4	$1.3 \cdot 10^0$	0	100
12.5	99.9	$6.4 \cdot 10^{-2}$	0.5	$1.1 \cdot 10^0$	5.2	$9.7 \cdot 10^{-1}$	0	100
15.0	99.9	$6.1 \cdot 10^{-2}$	0.4	$9.5 \cdot 10^{-1}$	5.4	$1.0 \cdot 10^0$	0	100
17.5	99.9	$6.4 \cdot 10^{-2}$	0.4	$8.9 \cdot 10^{-1}$	5.5	$1.1 \cdot 10^0$	0	100
20.0	99.9	$6.2 \cdot 10^{-2}$	0.3	$5.8 \cdot 10^{-1}$	5.5	$9.4 \cdot 10^{-1}$	0	100
22.5	99.9	$6.0 \cdot 10^{-2}$	0.2	$3.4 \cdot 10^{-1}$	5.5	$9.6 \cdot 10^{-1}$	0	100
25.0	99.9	$5.6 \cdot 10^{-2}$	0.2	$3.6 \cdot 10^{-1}$	6.0	$1.2 \cdot 10^0$	0	100
27.5	99.9	$5.5 \cdot 10^{-2}$	0.3	$4.2 \cdot 10^{-1}$	6.3	$1.4 \cdot 10^0$	0	100
30.0	99.9	$2.9 \cdot 10^{-1}$	0.4	$5.2 \cdot 10^{-1}$	6.7	$1.4 \cdot 10^0$	0	96
32.5	99.9	$3.3 \cdot 10^{-1}$	0.6	$8.0 \cdot 10^{-1}$	6.9	$1.3 \cdot 10^0$	0	87
35.0	99.6	$1.1 \cdot 10^0$	1.1	$1.2 \cdot 10^0$	6.9	$1.9 \cdot 10^0$	0	68
37.5	98.8	$1.9 \cdot 10^0$	2.4	$1.8 \cdot 10^0$	6.7	$2.6 \cdot 10^0$	0	38
40.0	96.7	$2.6 \cdot 10^0$	3.3	$2.2 \cdot 10^0$	5.4	$3.1 \cdot 10^0$	0	15
42.5	94.0	$3.0 \cdot 10^0$	6.1	$3.8 \cdot 10^0$	4.0	$2.2 \cdot 10^0$	0	0
45.0	91.2	$5.1 \cdot 10^0$	9.7	$4.4 \cdot 10^0$	3.4	$1.4 \cdot 10^0$	0	0
47.5	81.8	$1.2 \cdot 10^1$	13.1	$5.4 \cdot 10^0$	2.6	$8.1 \cdot 10^{-1}$	0	0
50.0	68.2	$1.4 \cdot 10^1$	19.9	$7.1 \cdot 10^0$	2.4	$9.2 \cdot 10^{-1}$	0	0
52.5	53.5	$1.3 \cdot 10^1$	25.8	$7.8 \cdot 10^0$	2.1	$7.0 \cdot 10^{-1}$	0	0
55.0	42.1	$1.3 \cdot 10^1$	28.4	$1.0 \cdot 10^1$	2.0	$6.0 \cdot 10^{-1}$	0	0
57.5	33.3	$1.4 \cdot 10^1$	26.1	$1.3 \cdot 10^1$	1.8	$4.9 \cdot 10^{-1}$	0	0
60.0	22.7	$1.6 \cdot 10^1$	21.2	$1.5 \cdot 10^1$	1.8	$9.0 \cdot 10^{-1}$	0	0
62.5	11.8	$1.4 \cdot 10^1$	16.9	$1.5 \cdot 10^1$	1.9	$1.2 \cdot 10^0$	0	0

TABLE 5.4: Results of the dense CFD database contaminated with a systematic error obtained by multiplying the database by 1.5 at randomly chosen database points for varying contamination percentage. For column headers refer to Table 5.1.

Error concentration of 35% corresponds to 68% probability of relative convergence, which is once again below the convergence limit (30%), which is primarily set by the reconstruction errors rapidly increasing at 27.5% error concentration (see Figure 5.4-right). If it was not for the errors, the method would perform well until 45% error concentration (see Figure 5.4-left). It is interesting to note that the convergence limit decreased by 7.5% with regard to the convergence limit of the dense CFD database contaminated with random, positive errors. This once again proves that the method performs worse when the error is correlated with database. The method also needs less iterations in order to converge.

The counterpart for the coarse database of Table 5.4 and Figure 5.4 is shown in Table A.4 and Figure A.4 respectively. In Section 4.2 the error contamination was set to 30% for the coarse database which is higher than the 25% set

FIGURE 5.4: Complement of Table 5.4. Left: $\%CDEP_m$ with the corresponding $\%CDEP_{std}$ bars (thick black line) and $\%IDEP_m$ with the corresponding $\%IDEP_{std}$ bars (thin black line). Right: mean MaxE and RMSE of the reconstructed part (solid and dashed line respectively), as defined in Equation (2.18).

for the positive, random error. 25% contamination error corresponds to 27% probability of relative convergence (see Table A.2), as once can see that absolute convergence is impossible. In this case the limit of convergence is 20%, lower than the 30% for the dense case. Again, this is mainly due to high reconstruction errors (see Figure A.4-left), in particular RMSE, which do not satisfy the threshold established for the relative convergence.

5.1.3 Random errors at concentrated database positions

Toy model database

Let us analyse now positive, random errors of 0.5 maximum magnitude concentrated in a circular cylinder spanning the z -direction for a varying diameter size, similarly to Section 3.4. The diameter has been varied between 0.325 to 0.900 every 0.025. The results are presented in Table 5.5.

In Section 3.4 the diameter value was chosen to be 0.315 which corresponds to about 60% absolute level convergence. However, the base circle can be expanded to 0.425 to reach the convergence limit. Beyond this point the method begins to fail as it is incapable of reaching 90% CDEP and the reconstruction errors rise sharply to $O(1)$ reaching a plateau (see Figure 5.5-right). The average iteration number also decreases from 9.2 to 6.4. 0.425 diameter corresponds to 13.3% contamination level meaning that concentrating errors without leaving any clean points in between them drastically worsens the performance of the cleaning method.

d	%CD _m	%CD _{std}	%ID _m	%ID _{std}	iter _m	iter _{std}	%C ₁	%C ₂
0.275	100	0	0.1	$2.1 \cdot 10^{-1}$	8.6	$5.8 \cdot 10^{-1}$	68	100
0.300	100.0	$1.7 \cdot 10^{-3}$	0.2	$2.3 \cdot 10^{-1}$	9.0	$7.7 \cdot 10^{-1}$	60	100
0.325	100.0	$1.6 \cdot 10^{-3}$	0.0	$6.5 \cdot 10^{-2}$	8.8	$5.0 \cdot 10^{-1}$	61	100
0.350	100.0	$8.6 \cdot 10^{-4}$	0.1	$4.2 \cdot 10^{-2}$	9.3	$5.7 \cdot 10^{-1}$	35	100
0.375	100.0	$7.8 \cdot 10^{-4}$	0.3	$4.6 \cdot 10^{-1}$	9.3	$6.8 \cdot 10^{-1}$	52	100
0.400	100.0	$1.3 \cdot 10^{-3}$	0.3	$3.9 \cdot 10^{-1}$	9.2	$4.6 \cdot 10^{-1}$	49	100
0.425	100.0	$5.8 \cdot 10^{-3}$	0.3	$3.2 \cdot 10^{-1}$	9.2	$5.6 \cdot 10^{-1}$	48	100
0.450	88.1	$1.0 \cdot 10^1$	0.1	$1.9 \cdot 10^{-1}$	6.4	$3.1 \cdot 10^0$	1	40
0.475	81.0	$5.5 \cdot 10^0$	0.0	$6.3 \cdot 10^{-2}$	4.4	$1.5 \cdot 10^0$	0	4
0.500	79.9	$5.1 \cdot 10^0$	0.0	$6.7 \cdot 10^{-2}$	4.1	$1.3 \cdot 10^0$	0	1
0.525	77.9	$4.8 \cdot 10^0$	0.0	$6.1 \cdot 10^{-2}$	4.1	$1.3 \cdot 10^0$	0	0
0.550	78.2	$5.0 \cdot 10^0$	0.0	$3.0 \cdot 10^{-2}$	4.2	$1.3 \cdot 10^0$	0	0
0.575	78.0	$5.2 \cdot 10^0$	0.0	$5.5 \cdot 10^{-2}$	4.2	$1.4 \cdot 10^0$	0	0
0.600	76.9	$5.2 \cdot 10^0$	0.0	$1.6 \cdot 10^{-1}$	3.8	$1.2 \cdot 10^0$	0	0
0.625	78.0	$5.3 \cdot 10^0$	0.0	$2.0 \cdot 10^{-2}$	4.0	$1.2 \cdot 10^0$	0	0
0.650	78.8	$6.0 \cdot 10^0$	0	0	3.8	$1.3 \cdot 10^0$	0	0
0.675	80.8	$5.9 \cdot 10^0$	0.0	$1.5 \cdot 10^{-1}$	3.8	$9.1 \cdot 10^{-1}$	0	0
0.700	82.9	$5.2 \cdot 10^0$	0.0	$1.5 \cdot 10^{-1}$	3.6	$7.9 \cdot 10^{-1}$	0	0
0.725	83.3	$4.1 \cdot 10^0$	0.0	$1.3 \cdot 10^{-1}$	3.5	$8.0 \cdot 10^{-1}$	0	0
0.750	82.8	$3.7 \cdot 10^0$	0.1	$1.9 \cdot 10^{-1}$	3.8	$1.0 \cdot 10^0$	0	0
0.775	82.8	$3.1 \cdot 10^0$	0.1	$2.0 \cdot 10^{-1}$	4.2	$1.2 \cdot 10^0$	0	0
0.800	83.0	$2.7 \cdot 10^0$	0.1	$2.1 \cdot 10^{-1}$	4.1	$9.5 \cdot 10^{-1}$	0	0
0.825	82.4	$2.5 \cdot 10^0$	0.1	$2.1 \cdot 10^{-1}$	4.1	$1.1 \cdot 10^0$	0	0
0.850	79.2	$3.7 \cdot 10^0$	0.2	$2.1 \cdot 10^{-1}$	4.3	$1.1 \cdot 10^0$	0	0

TABLE 5.5: Results of the dense 3D toy model database contaminated with positive, random error of maximum magnitude 0.5 in a circular cylinder spanning the z -direction for varying diameter d . For column headers please refer to Table 5.1.

The counterpart for the coarse database of Table 5.5 and Figure 5.5 is shown in Table A.5 and Figure A.5 respectively. Note that while the diameter is increased in steps of 0.025, the error concentration may not always increase as the mesh is too coarse. In Section 3.4 the circle diameter was set to 0.2 for the coarse database, which was considered an optimal performance of the method, slightly smaller than for the dense database. This diameter has 0% chance of absolute convergence (due to IDEP) but a 84% probability of relative convergence. In this case the behaviour of the method is not monotonous, i.e. the probability of convergence does not decrease with the increase of the diameter. However, the probability of identifying all CDEP is monotonous and falls below 90% when the diameter of the circle is 0.425. Until this value the relative convergence is highly probable, reaching 97%, the reconstruction errors being still small (see Figure A.5).

FIGURE 5.5: Complement of Table 5.5. Left: %CDEP_m with the corresponding %CDEP_{std} bars (thick black line) and %IDEP_m with the corresponding %IDEP_{std} bars (thin black line). Right: mean MaxE and RMSE of the reconstructed part (solid and dashed line respectively), as defined in Equation (2.18).

CFD database

In Section 4.2 positive random errors of 0.5 maximum magnitude were applied to an elliptical cylinder with the base defined by $s^2/0.0034^2 + (y-2.2)^2/0.34^2 = 1$, spanning the whole database in the angle of attack. In this case the method correctly identifies all the erroneous points, but at the same time it considers 142 points as erroneous. For the purpose of this analysis, we will consider a elliptical cylinder of the same shape given by $s^2/a^2 + (y-2.2)^2/b^2 = 1$, where $a = b/100$ and b is varied from 0.125 to 0.625 at 0.025 steps.

This case is very demanding as the ellipse is located in the centre of the domain, just at the leading edge of the wing where the pressure gradients are the highest. As a result, the method does not behave monotonously as in the previous examples, but has a clear optimum in the parameter b . When b is small, the errors are concentrated in a very demanding region and the method is incapable of performing well. The probability of detecting IDEP is very high (up to 67.9%) as there are very few error points and the performance is more erratic which is reflected by high %CDEP_{std}. Indeed, for b between 0.125 and 0.275 the chance of relative convergence is below 90%, but with incrementing tendency. As the ellipse grows, it places errors in less demanding regions, where the method can perform much better. For b between 0.300 and 0.475 the method achieves relative convergence of above 90% thus fulfilling the criteria of convergence limit. The behaviour is also more stable if %CDEP_{std} is low. From $b = 0.5$ and onwards the reconstruction errors begin to build up and the probability of relative convergence decay. In light of this analysis, the value of $b = 0.34$ used in Section 4.2 has been chosen correctly to demonstrate the functioning of the filtering method. It is also worth mentioning that the probability of CDEP is always above 90%.

b	$\%CD_m$	$\%CD_{std}$	$\%ID_m$	$\%ID_{std}$	k_m	k_{std}	$\%C_1$	$\%C_2$
0.125	99.1	$1.2 \cdot 10^0$	67.9	$6.1 \cdot 10^1$	7.2	$1.4 \cdot 10^0$	0	54
0.150	98.1	$3.6 \cdot 10^0$	46.6	$3.0 \cdot 10^1$	7.1	$1.8 \cdot 10^0$	0	30
0.175	98.6	$2.0 \cdot 10^0$	40.4	$2.4 \cdot 10^1$	7.0	$1.6 \cdot 10^0$	0	36
0.200	97.7	$4.1 \cdot 10^0$	40.3	$3.2 \cdot 10^1$	6.8	$2.1 \cdot 10^0$	0	45
0.225	97.6	$4.5 \cdot 10^0$	31.8	$2.2 \cdot 10^1$	6.8	$1.8 \cdot 10^0$	0	69
0.250	98.0	$3.6 \cdot 10^0$	37.1	$2.4 \cdot 10^1$	7.0	$1.9 \cdot 10^0$	0	85
0.275	98.0	$3.6 \cdot 10^0$	37.1	$2.4 \cdot 10^1$	7.0	$1.9 \cdot 10^0$	0	85
0.300	98.7	$1.8 \cdot 10^0$	35.2	$2.1 \cdot 10^1$	7.2	$1.7 \cdot 10^0$	0	99
0.325	98.7	$1.8 \cdot 10^0$	35.2	$2.1 \cdot 10^1$	7.2	$1.7 \cdot 10^0$	0	99
0.350	98.8	$9.6 \cdot 10^{-1}$	20.7	$1.7 \cdot 10^1$	7.3	$1.5 \cdot 10^0$	0	100
0.375	98.7	$8.7 \cdot 10^{-1}$	13.4	$1.5 \cdot 10^1$	7.1	$1.3 \cdot 10^0$	0	100
0.400	98.6	$1.1 \cdot 10^0$	11.4	$1.3 \cdot 10^1$	7.2	$1.2 \cdot 10^0$	0	100
0.425	98.2	$3.1 \cdot 10^0$	8.8	$1.2 \cdot 10^1$	7.4	$1.3 \cdot 10^0$	0	98
0.450	97.9	$3.2 \cdot 10^0$	7.2	$9.6 \cdot 10^0$	7.5	$1.6 \cdot 10^0$	0	96
0.475	97.8	$3.2 \cdot 10^0$	6.7	$8.8 \cdot 10^0$	7.8	$1.6 \cdot 10^0$	0	92
0.500	96.6	$6.2 \cdot 10^0$	9.0	$1.0 \cdot 10^1$	7.9	$1.8 \cdot 10^0$	0	86
0.525	96.7	$6.0 \cdot 10^0$	7.5	$9.8 \cdot 10^0$	7.8	$1.8 \cdot 10^0$	0	88
0.550	96.1	$6.8 \cdot 10^0$	10.6	$1.2 \cdot 10^1$	8.1	$2.0 \cdot 10^0$	0	85
0.575	93.9	$8.6 \cdot 10^0$	7.1	$8.2 \cdot 10^0$	8.0	$1.7 \cdot 10^0$	0	71
0.600	96.6	$5.5 \cdot 10^0$	7.6	$7.6 \cdot 10^0$	8.6	$1.6 \cdot 10^0$	0	81
0.625	95.0	$8.1 \cdot 10^0$	7.4	$1.2 \cdot 10^1$	8.1	$1.7 \cdot 10^0$	0	76

TABLE 5.6: Results of the dense CFD database contaminated with positive, random error of maximum magnitude 0.5 in an elliptical cylinder spanning the z -direction for varying parameter b . For column headers refer to Table 5.1.

FIGURE 5.6: Complement of Table 5.6. Left: $\%CDEP_m$ with the corresponding $\%CDEP_{std}$ bars (thick black line) and $\%IDEP_m$ with the corresponding $\%IDEP_{std}$ bars (thin black line). Right: mean MaxE and RMSE of the reconstructed part (solid and dashed line respectively), as defined in Equation (2.18).

The counterpart for the coarse database of Table 5.6 and Figure 5.6 is shown in Table A.6 and Figure A.6 respectively. Since the performance of the method degrades with the decreasing mesh density, this is the worst case presented in this section for the CFD database. The parameter b was set to 0.32 in Section 4.2, only slightly lower than for the dense database case. This corresponds to 50% probability of reaching relative convergence. However, in comparison to the dense model, in this case the method is always below the convergence limit. As it can be seen from Figure A.6-left, this is once again due to high reconstruction errors.

5.2 Randomness influence for varying error magnitude

For this analysis, the tunable parameters that appear in Equations (3.1), (3.5) and (3.7), have been maintained and are equal to the ones defined in Equation (3.26) and (4.2) for the 3D toy model and CFD database respectively. For each combination of error and contamination, the analysis was run 100 times for both databases in order to obtain a sufficiently big sample of results. The number of allowed applications of the cleaning-HOSVD method, labelled k , has been set to 21 due to computational power restrictions. As will be seen later, this value is never reached.

5.2.1 Random errors at randomly chosen database positions

Toy model database

To begin with, the dense toy model database is contaminated with positive, random error of varying maximum magnitude at 10% randomly chosen database points. The error maximum magnitude has been varied between $1.0 \cdot 10^{-4}$ and $0.5 \cdot 10^0$. The results are presented in Table 5.7.

Analysing data from Table 5.7 it can be seen that, as expected, for high error magnitude (from $6.6 \cdot 10^{-3}$ to 0.5) the method is capable of cleaning all the erroneous points, reaching absolute and relative convergence probability of more than 80 and 100% respectively. It is also reflected in Figure 5.7-left where the $\%CDEP_m$ and $\%IDEP_m$ is 100 and 0% respectively. To further confirm the correct functioning of the method, the reconstruction errors shown in Figure 5.7-right are below 10^{-8} up to the error magnitude $6.6 \cdot 10^{-3}$. The situation abruptly changes as the magnitude is further diminished since for magnitude $5.3 \cdot 10^{-3}$ the probability for both absolute and relative convergence is 0%. This is due to the fact that the method is incapable of detecting new errors is terminated by the stop conditions defined in Equation (3.19) and/or (3.21) just after performing two iterations. The method then only considers the first iteration as

err	%CD _m	%CD _{std}	%ID _m	%ID _{std}	k _m	k _{std}	%C ₁	%C ₂
$1.0 \cdot 10^{-4}$	0	0	0	0	1	0	0	0
$1.0 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$1.2 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$1.5 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$1.9 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$2.3 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$2.8 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$3.5 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$4.3 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$5.3 \cdot 10^{-3}$	1.6	$1.5 \cdot 10^{-1}$	0	0	1	0	0	0
$6.6 \cdot 10^{-3}$	100.0	$2.8 \cdot 10^{-3}$	0.0	$6.6 \cdot 10^{-3}$	9.2	$6.8 \cdot 10^{-1}$	84	100
$8.1 \cdot 10^{-3}$	100.0	$2.7 \cdot 10^{-3}$	0.0	$6.6 \cdot 10^{-3}$	9.1	$6.7 \cdot 10^{-1}$	85	100
$1.0 \cdot 10^{-2}$	100.0	$2.6 \cdot 10^{-3}$	0.0	$6.6 \cdot 10^{-3}$	8.8	$6.9 \cdot 10^{-1}$	86	100
$1.3 \cdot 10^{-2}$	100.0	$2.3 \cdot 10^{-3}$	0.0	$6.6 \cdot 10^{-3}$	8.6	$6.6 \cdot 10^{-1}$	89	100
$1.7 \cdot 10^{-2}$	100.0	$2.1 \cdot 10^{-3}$	0.0	$6.6 \cdot 10^{-3}$	8.4	$6.7 \cdot 10^{-1}$	91	100
$2.0 \cdot 10^{-2}$	100.0	$1.8 \cdot 10^{-3}$	0.0	$6.6 \cdot 10^{-3}$	8.4	$6.8 \cdot 10^{-1}$	93	100
$4.0 \cdot 10^{-2}$	100.0	$1.6 \cdot 10^{-3}$	0.0	$6.6 \cdot 10^{-3}$	8.4	$6.8 \cdot 10^{-1}$	94	100
$6.0 \cdot 10^{-2}$	100.0	$1.4 \cdot 10^{-3}$	0.0	$6.6 \cdot 10^{-3}$	8.4	$7.0 \cdot 10^{-1}$	95	100
$8.0 \cdot 10^{-2}$	100.0	$1.4 \cdot 10^{-3}$	0.0	$6.6 \cdot 10^{-3}$	8.4	$7.1 \cdot 10^{-1}$	95	100
$1.0 \cdot 10^{-1}$	100.0	$1.4 \cdot 10^{-3}$	0.0	$6.6 \cdot 10^{-3}$	8.4	$7.1 \cdot 10^{-1}$	95	100
$2.0 \cdot 10^{-1}$	100.0	$8.0 \cdot 10^{-4}$	0.0	$9.7 \cdot 10^{-3}$	8.4	$7.3 \cdot 10^{-1}$	96	100
$3.0 \cdot 10^{-1}$	100	0	0.0	$1.6 \cdot 10^{-2}$	8.5	$6.7 \cdot 10^{-1}$	93	100
$4.0 \cdot 10^{-1}$	100	0	0.0	$2.4 \cdot 10^{-2}$	8.5	$6.6 \cdot 10^{-1}$	92	100
$5.0 \cdot 10^{-1}$	100	0	0.0	$3.0 \cdot 10^{-2}$	8.6	$6.4 \cdot 10^{-1}$	87	100

TABLE 5.7: Results of the dense 3D toy model database contaminated with positive, random error of varying maximum magnitude at 10% randomly chosen database points. The meaning of the column headers is: err- maximum magnitude; %CD_m- mean percentage of correctly detected error points (%CDEP) normalised with the total number of error points; %CD_{std}- standard deviation of %CDEP; %ID_m- mean percentage of incorrectly detected error points (%IDEF) normalised with the total number of error points; %ID_{std}- standard deviation of %IDEF; k_m- mean number of application of the method, k; k_{std}- standard deviation of k; %C₁- convergence percentage when the method correctly identifies all the error points without including any clean points (*absolute convergence*); %C₂- convergence percentage when the method identifies at least 90% of error points and the reconstruction MaxE and RMSE defined in Equation (2.18) are of order lower than $O(-1)$ and $O(-2)$ respectively (*relative convergence*).

the converged one but since it was incapable of detecting any points in both applications, it is stopped. Changing the FF parameter or the number of retained modes in the first application could solve this problem. However, testing this hypothesis is beyond the scope of this Thesis. Nevertheless, since the method

was designed to filter errors of $O(1)$, it is considered a rather good achievement to be able to filter errors as small as $6.6 \cdot 10^{-3}$, without changing any adjustment to the parameters discussed in Chapter 3.

FIGURE 5.7: Complement of Table 5.7. Left: $\%CDEP_m$ (thick black line) and $\%IDEP_m$ (thin black line). Right: mean MaxE and RMSE of the reconstructed part (solid and dashed line respectively), as defined in Equation (2.18).

The counterpart for the coarse database of Table 5.7 and Figure 5.7 is shown in Table B.1 and Figure B.1 respectively. Similarly to the dense database, the method performs very well filtering errors of high magnitude. In this case however the absolute convergence is lower as it never reaches 80% (see Table B.1). This was to be expected though as it has already been said in Chapter 3 that the method performs worse for coarser grids. Nevertheless, up to an error magnitude of $6.6 \cdot 10^{-3}$ the method reaches 100% relative convergence and it almost always identifies 100% erroneous points (see Figure B.1-left) while both reconstruction errors are well below 10^{-3} (see Figure B.1-right). Although slightly worse, especially in terms of absolute convergence, the performance of the method for varying error magnitude does not strongly depend on the grid definition.

CFD database

In this case the CFD dense database was contaminated with positive, random error of varying maximum magnitude at 10% randomly chosen database points. Analogically to the toy database, the error maximum magnitude has been varied between $1.0 \cdot 10^{-4}$ and $0.5 \cdot 10^0$. The results are presented in Table 5.8.

In comparison to the toy model distribution with same type of error, the method performs worse as the minimum error magnitude, which it is capable of filtering, is $4.0 \cdot 10^{-2}$. For this value the method has 55% probability of relative convergence (absolute convergence already had no chance of occurring for 0.5

err	%CD _m	%CD _{std}	%ID _m	%ID _{std}	k _m	k _{std}	%C ₁	%C ₂
$1.0 \cdot 10^{-4}$	1.6	$4.4 \cdot 10^{-1}$	14.5	$3.4 \cdot 10^0$	5.1	$9.6 \cdot 10^{-1}$	0	0
$1.0 \cdot 10^{-3}$	1.7	$5.7 \cdot 10^{-1}$	15.1	$3.9 \cdot 10^0$	5.1	$1.1 \cdot 10^0$	0	0
$1.0 \cdot 10^{-2}$	19.9	$2.7 \cdot 10^1$	13.6	$4.0 \cdot 10^0$	4.0	$1.7 \cdot 10^0$	0	0
$2.0 \cdot 10^{-2}$	23.0	$3.5 \cdot 10^1$	11.1	$4.6 \cdot 10^0$	2.5	$2.6 \cdot 10^0$	0	0
$4.0 \cdot 10^{-2}$	69.5	$3.7 \cdot 10^1$	12.7	$5.7 \cdot 10^0$	5.2	$2.8 \cdot 10^0$	0	55
$6.0 \cdot 10^{-2}$	92.9	$9.1 \cdot 10^0$	9.7	$3.8 \cdot 10^0$	6.7	$1.4 \cdot 10^0$	0	94
$8.0 \cdot 10^{-2}$	95.2	$1.5 \cdot 10^0$	7.3	$5.8 \cdot 10^0$	7.0	$1.3 \cdot 10^0$	0	99
$1.0 \cdot 10^{-1}$	96.2	$1.2 \cdot 10^0$	6.9	$4.6 \cdot 10^0$	7.2	$1.3 \cdot 10^0$	0	100
$2.0 \cdot 10^{-1}$	97.8	$8.9 \cdot 10^{-1}$	5.7	$4.5 \cdot 10^0$	7.5	$1.5 \cdot 10^0$	0	100
$3.0 \cdot 10^{-1}$	97.8	$7.1 \cdot 10^{-1}$	1.4	$1.8 \cdot 10^0$	5.8	$1.2 \cdot 10^0$	0	100
$4.0 \cdot 10^{-1}$	98.3	$6.0 \cdot 10^{-1}$	1.4	$2.4 \cdot 10^0$	5.8	$1.4 \cdot 10^0$	0	100
$5.0 \cdot 10^{-1}$	98.5	$5.0 \cdot 10^{-1}$	1.4	$2.8 \cdot 10^0$	6.0	$1.3 \cdot 10^0$	0	100

TABLE 5.8: Results of the dense CFD database contaminated with positive, random error of varying maximum magnitude at 10% randomly chosen database points. For column headers please refer to Table 5.7.

error magnitude, see Section 5.1). Nevertheless, for this error magnitude the mean standard deviation the CDEP of 37% means that the method behaves erratic and hence it is better to assume that the method functions well for a slightly greater error magnitude of $6.0 \cdot 10^{-2}$. In this case the probability of relative convergence is 94% which is also the convergence limit and the mean standard deviation is 9%. When looking at Figure 5.8-left, this point is clearly marked as the method manages to find on average 92.9% of erroneous points and additionally taking 9.7% clean points as erroneous. The reconstruction errors shown in Figure 5.8-right are also low and fulfil the relative convergence criteria described above.

FIGURE 5.8: Complement of Table 5.8. Left: %CDEP_m (thick black line) and %IDEP_m (thin black line). Right: mean MaxE and RMSE of the reconstructed part (solid and dashed line respectively), as defined in Equation (2.18).

The counterpart for the coarse database of Table 5.8 and Figure 5.8 is shown in Table B.2 and Figure B.2 respectively. As seen in for the toy model database, the method performs worse also for the CFD coarse database. The limit of convergence is for the error magnitude of $2.0 \cdot 10^{-1}$ as the method achieves 96% relative convergence. For this value the method is able to identify a sufficient percentage of erroneous points (see Figure B.2-left) and also the reconstruction errors were sufficiently low (see Figure B.2-right). Afterwards, both errors reach a plateau of about 0.1 which is unacceptably high.

5.2.2 Systematic errors at randomly chosen database positions

Toy model database

Similarly to the previous Section 5.1, where the contamination level was varied, the toy model dense database is contaminated with systematic error of varying magnitude at 10% random database positions. The results are presented in Table 5.7. In this case the parameter error means the factor by which the database is multiplied and the result is summed with the original database, i.e. an err of $3.0 \cdot 10^{-1}$ means that the database point is replaced by 1.3 of its value.

Once again it has been shown that systematic errors at randomly placed database points functions worse than for random error. The former is efficiently filtered for $1.0 \cdot 10^{-2}$ (see Table 5.9) whereas the latter for $6.6 \cdot 10^{-3}$ (see Table 5.7). Once again the relative convergence decreases from 100% to 0% over $1.9 \cdot 10^{-3}$ magnitude span. This happens due to both incapability of finding erroneous points (see Figure 5.9-left) and rapidly increasing reconstruction errors (see Figure 5.9-right). However, both reconstruction errors are still below the level required for relative convergence. In general, although it performs worse for the systematic error type, the functioning is still very good as it is able to clean errors one order of magnitude smaller.

The counterpart for the coarse database of Table 5.9 and Figure 5.9 is shown in Table B.3 and Figure B.3 respectively. The method performs similarly to the dense database case. In this case the relative convergence begins to decay for error magnitude of $1.0 \cdot 10^{-2}$, reaching a probability of 62%. For this value the average percentage of CDEP is already 65.7% (see Figure B.3-left), which is very low and adding to the fact that the standard deviation is high means that the method behaviour is not stable. Although the reconstruction errors increase rapidly (see Figure B.3-right), they are still below the relative convergence criteria. The convergence limit is at error magnitude of $1.3 \cdot 10^{-2}$. In terms of absolute convergence, the probability is very low and reaches the maximum probability of 15% corresponding to the same error level.

err	%CD _m	%CD _{std}	%ID _m	%ID _{std}	k _m	k _{std}	%C ₁	%C ₂
$1.0 \cdot 10^{-4}$	0	0	0	0	1	0	0	0
$1.0 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$1.2 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$1.5 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$1.9 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$2.3 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$2.8 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$3.5 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$4.3 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$5.3 \cdot 10^{-3}$	0.0	$7.0 \cdot 10^{-3}$	0.0	0	1.0	0	0	0
$6.6 \cdot 10^{-3}$	0.2	$5.4 \cdot 10^{-2}$	0.0	0	1.0	0	0	0
$8.1 \cdot 10^{-3}$	3.7	$1.8 \cdot 10^{-1}$	0.0	0	1.0	0	0	0
$1.0 \cdot 10^{-2}$	100.0	$2.3 \cdot 10^{-3}$	0.1	$6.8 \cdot 10^{-2}$	4.6	$7.7 \cdot 10^{-1}$	32	100
$1.3 \cdot 10^{-2}$	100.0	$2.3 \cdot 10^{-3}$	0.1	$6.5 \cdot 10^{-2}$	4.7	$8.1 \cdot 10^{-1}$	22	100
$1.7 \cdot 10^{-2}$	100.0	$2.3 \cdot 10^{-3}$	0.1	$6.7 \cdot 10^{-2}$	4.6	$6.3 \cdot 10^{-1}$	19	100
$2.0 \cdot 10^{-2}$	100.0	$2.3 \cdot 10^{-3}$	0.1	$6.7 \cdot 10^{-2}$	4.5	$6.4 \cdot 10^{-1}$	19	100
$4.0 \cdot 10^{-2}$	100.0	$2.3 \cdot 10^{-3}$	0.1	$6.6 \cdot 10^{-2}$	4.7	$6.7 \cdot 10^{-1}$	13	100
$6.0 \cdot 10^{-2}$	100.0	$2.3 \cdot 10^{-3}$	0.1	$6.7 \cdot 10^{-2}$	4.7	$6.5 \cdot 10^{-1}$	15	100
$8.0 \cdot 10^{-2}$	100.0	$2.3 \cdot 10^{-3}$	0.1	$9.9 \cdot 10^{-2}$	4.8	$7.4 \cdot 10^{-1}$	11	100
$1.0 \cdot 10^{-1}$	100.0	$2.3 \cdot 10^{-3}$	0.1	$8.0 \cdot 10^{-2}$	5.0	$7.6 \cdot 10^{-1}$	10	100
$2.0 \cdot 10^{-1}$	100.0	$2.3 \cdot 10^{-3}$	0.1	$7.0 \cdot 10^{-2}$	4.9	$7.5 \cdot 10^{-1}$	12	100
$3.0 \cdot 10^{-1}$	100.0	$2.3 \cdot 10^{-3}$	0.1	$9.4 \cdot 10^{-2}$	4.7	$7.3 \cdot 10^{-1}$	14	100
$4.0 \cdot 10^{-1}$	100.0	$2.3 \cdot 10^{-3}$	0.1	$7.2 \cdot 10^{-2}$	4.7	$7.5 \cdot 10^{-1}$	16	100
$5.0 \cdot 10^{-1}$	100.0	$2.3 \cdot 10^{-3}$	0.1	$7.2 \cdot 10^{-2}$	4.7	$7.1 \cdot 10^{-1}$	18	100

TABLE 5.9: Results of the dense 3D toy model database contaminated with systematic error of varying maximum magnitude at 10% randomly chosen database points. For column headers please refer to Table 5.7.

CFD database

This section describes the application of systematic error of varying maximum magnitude to 10% randomly distributed locations. Similarly to the case of the toy model database described above, the parameter err means the factor by which the database is multiplied and the result is summed with the original database. The results are summarised in Table 5.10.

The systematic error type is filtered with a similar degree of efficiency. The method performs well up to error magnitude of $6.0 \cdot 10^{-2}$, although for $8.0 \cdot 10^{-2}$ the method's function does not fit well to the trend, i.e. for higher values the relative convergence is 100% and for lower ones it is 99%. It could be assumed that for $8.0 \cdot 10^{-2}$ the probability of relative convergence should be in between of those two values. However, the probability is 83% and the method behaves quite erratically, as the average CDEP is 85.8% with an average standard deviation of 31% (see Table 5.10). This is quite high. This is reflected in Figure 5.10

FIGURE 5.9: Complement of Table 5.9. Left: $\%CDEP_m$ (thick black line) and $\%IDEP_m$ (thin black line). Right: mean MaxE and RMSE of the reconstructed part (solid and dashed line respectively), as defined in Equation (2.18).

err	$\%CD_m$	$\%CD_{std}$	$\%ID_m$	$\%ID_{std}$	k_m	k_{std}	$\%C_1$	$\%C_2$
$1.0 \cdot 10^{-4}$	1.6	$4.9 \cdot 10^{-1}$	14.4	$3.9 \cdot 10^0$	5.0	$1.1 \cdot 10^0$	0	0
$1.0 \cdot 10^{-3}$	1.6	$5.4 \cdot 10^{-1}$	14.4	$3.7 \cdot 10^0$	4.8	$1.2 \cdot 10^0$	0	0
$1.0 \cdot 10^{-2}$	5.1	$1.0 \cdot 10^1$	11.8	$2.3 \cdot 10^0$	3.0	$9.2 \cdot 10^{-1}$	0	1
$2.0 \cdot 10^{-2}$	20.0	$3.5 \cdot 10^1$	10.8	$3.7 \cdot 10^0$	2.0	$2.1 \cdot 10^0$	0	16
$4.0 \cdot 10^{-2}$	36.7	$3.8 \cdot 10^1$	9.1	$3.1 \cdot 10^0$	2.2	$2.1 \cdot 10^0$	0	27
$6.0 \cdot 10^{-2}$	98.8	$9.2 \cdot 10^0$	9.0	$3.3 \cdot 10^0$	5.9	$1.1 \cdot 10^0$	0	99
$8.0 \cdot 10^{-2}$	85.8	$3.1 \cdot 10^1$	6.5	$3.3 \cdot 10^0$	5.3	$2.1 \cdot 10^0$	0	83
$1.0 \cdot 10^{-1}$	99.7	$1.4 \cdot 10^{-1}$	3.8	$2.2 \cdot 10^0$	5.8	$9.7 \cdot 10^{-1}$	0	100
$2.0 \cdot 10^{-1}$	99.8	$1.2 \cdot 10^{-1}$	3.8	$3.5 \cdot 10^0$	6.4	$1.4 \cdot 10^0$	0	100
$3.0 \cdot 10^{-1}$	99.9	$1.2 \cdot 10^{-1}$	0.8	$1.3 \cdot 10^0$	5.1	$1.1 \cdot 10^0$	0	100
$4.0 \cdot 10^{-1}$	99.9	$9.3 \cdot 10^{-2}$	1.0	$1.7 \cdot 10^0$	5.3	$1.2 \cdot 10^0$	0	100
$5.0 \cdot 10^{-1}$	99.9	$6.3 \cdot 10^{-2}$	1.1	$2.7 \cdot 10^0$	5.4	$1.3 \cdot 10^0$	0	100

TABLE 5.10: Results of the CFD dense database contaminated with systematic error of varying maximum magnitude at 10% randomly chosen database points. For column headers please refer to Table 5.7.

as there is a clearly visible drop in average percentage of CDEP and a spike in the reconstruction errors. After an error magnitude of $6.0 \cdot 10^{-2}$ the decrease in the relative convergence is not as sudden as for the toy model. Additionally, then the method performs well and it requires below 2 iterations to find all the erroneous points.

The counterpart for the coarse database of Table 5.10 and Figure 5.10 is shown in Table B.4 and Figure B.4 respectively. Not surprisingly, the method performs worse for the coarse database. The convergence limit is for $2.0 \cdot 10^{-1}$. Afterwards the performance gradually decreases until the probability is 1% for $2.0 \cdot 10^{-2}$. For $8.0 \cdot 10^{-2}$ error magnitude and smaller, the reconstruction errors both reach

FIGURE 5.10: Complement of Table 5.10. Left: $\%CDEP_m$ (thick black line) and $\%IDEP_m$ (thin black line). Right: mean MaxE and RMSE of the reconstructed part (solid and dashed line respectively), as defined in Equation (2.18).

the same order $O(-1)$ (see Figure B.4). However, the average percentage of CDEP is too low for the method to perform well.

5.2.3 Random errors at concentrated database positions

Toy model database

In this case a positive, random error was distributed in a 0.1-diameter cylinder spanning the z -direction. As before, the maximum magnitude of the error was varied and the results are presented in Table 5.11.

One conclusion that can be drawn from Table 5.11 is that the method never reaches absolute convergence. Furthermore, the previous Section 5.1 reports that the method reaches 68% probability of absolute convergence for cylinder diameter 0.275 (see Table 5.5), with an increasing tendency for smaller cylinder diameters. This is very counter-intuitive; however it is not sensible to extrapolate such tendencies for much smaller circle diameters such as the one considered in this Section 3.4. The reason why the absolute convergence is zero is the fact that the method takes a few IDEP in every single run, violating the definition of absolute convergence. However, from an engineering point of view, correcting a few clean points with virtually no reconstruction error can be considered a satisfactory result. In fact, up to $1.0 \cdot 10^{-2}$ the method has almost 100% probability of identifying all the error points and reconstructing the missing data with small reconstruction errors. This is reflected in the fact that the relative convergence has almost 100% probability of occurring up to $1.0 \cdot 10^{-2}$ error magnitude. This value coincides with the convergence limit as the method is suddenly incapable of identifying any points.

err	%CD _m	%CD _{std}	%ID _m	%ID _{std}	k _m	k _{std}	%C ₁	%C ₂
$1.0 \cdot 10^{-4}$	0	0	0	0	1	0	0	0
$1.0 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$1.2 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$1.5 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$1.9 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$2.3 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$2.8 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$3.5 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$4.3 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$5.3 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$6.6 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$8.1 \cdot 10^{-3}$	1.0	$10.0 \cdot 10^0$	0.0	$5.0 \cdot 10^{-2}$	1.1	$6.0 \cdot 10^{-1}$	0	1
$1.0 \cdot 10^{-2}$	99.0	$9.8 \cdot 10^0$	0.5	$5.0 \cdot 10^{-2}$	7.5	$9.5 \cdot 10^{-1}$	0	99
$1.3 \cdot 10^{-2}$	100	0	0.5	$3.5 \cdot 10^{-16}$	7.3	$6.4 \cdot 10^{-1}$	0	100
$1.7 \cdot 10^{-2}$	100	0	0.5	$3.5 \cdot 10^{-16}$	7.1	$6.7 \cdot 10^{-1}$	0	100
$2.0 \cdot 10^{-2}$	100	0	0.5	$3.5 \cdot 10^{-16}$	7.1	$6.5 \cdot 10^{-1}$	0	100
$4.0 \cdot 10^{-2}$	100	0	0.5	$3.5 \cdot 10^{-16}$	7.1	$6.5 \cdot 10^{-1}$	0	100
$6.0 \cdot 10^{-2}$	98.5	$7.0 \cdot 10^0$	0.5	$1.1 \cdot 10^{-1}$	6.9	$1.2 \cdot 10^0$	0	95
$8.0 \cdot 10^{-2}$	99.5	$3.9 \cdot 10^0$	0.5	$7.0 \cdot 10^{-2}$	7.0	$9.7 \cdot 10^{-1}$	0	98
$1.0 \cdot 10^{-1}$	100	0	0.5	$1.2 \cdot 10^{-1}$	7.0	$6.3 \cdot 10^{-1}$	0	100
$2.0 \cdot 10^{-1}$	100	0	0.6	$3.0 \cdot 10^{-1}$	7.1	$5.3 \cdot 10^{-1}$	0	100
$3.0 \cdot 10^{-1}$	100	0	0.8	$7.2 \cdot 10^{-1}$	7.2	$4.9 \cdot 10^{-1}$	0	100
$4.0 \cdot 10^{-1}$	100	0	0.6	$3.3 \cdot 10^{-1}$	7.2	$4.7 \cdot 10^{-1}$	0	100
$5.0 \cdot 10^{-1}$	100	0	0.6	$3.1 \cdot 10^{-1}$	7.2	$5.2 \cdot 10^{-1}$	0	100

TABLE 5.11: Results of the dense 3D toy model database contaminated with positive, random error of varying maximum magnitude concentrated in a cylinder of 0.1 diameter. For column headers please refer to Table 5.7.

The counterpart for the coarse database of Table 5.11 and Figure 5.11 is shown in Table B.5 and Figure B.5 respectively. In this case for large error maximum magnitudes absolute convergence is possible. Nevertheless, it is more interesting to analyse the general performance of the method by looking at the relative performance. As has been seen before, the filtering method performs worse for coarse databases as for positive, random errors concentrated in a circular cylinder the relative convergence is more erratic as the method is not capable of always identifying all the erroneous points (see Figure B.5-left) and, furthermore, it does not reconstruct the data with high accuracy (see Figure B.5-right). Consequently, the convergence limit is not easy to establish. A good estimate is $8.0 \cdot 10^{-2}$.

FIGURE 5.11: Complement of Table 5.11. Left: $\%CDEP_m$ (thick black line) and $\%IDEP_m$ (thin black line). Right: mean MaxE and RMSE of the reconstructed part (solid and dashed line respectively), as defined in Equation (2.18).

CFD database

Similarly to the toy model, the CFD dense database is contaminated with positive, random error in an elliptical cylinder, where the error maximum magnitude is varied and the parameter b , which controls the size of the ellipse, is fixed to 0.35. The results are presented in Table 5.12.

err	$\%CD_m$	$\%CD_{std}$	$\%ID_m$	$\%ID_{std}$	k_m	k_{std}	$\%C_1$	$\%C_2$
$1.0 \cdot 10^{-4}$	1.0	$9.8 \cdot 10^{-1}$	59.1	$1.2 \cdot 10^1$	5.3	$8.6 \cdot 10^{-1}$	0	0
$1.0 \cdot 10^{-3}$	0.7	$1.0 \cdot 10^0$	55.1	$1.2 \cdot 10^1$	5.2	$9.7 \cdot 10^{-1}$	0	0
$1.0 \cdot 10^{-2}$	21.0	$2.2 \cdot 10^1$	64.8	$2.1 \cdot 10^1$	5.2	$2.0 \cdot 10^0$	0	0
$2.0 \cdot 10^{-2}$	29.1	$3.0 \cdot 10^1$	51.8	$1.9 \cdot 10^1$	4.1	$2.9 \cdot 10^0$	0	0
$4.0 \cdot 10^{-2}$	38.5	$3.6 \cdot 10^1$	54.3	$1.5 \cdot 10^1$	3.5	$2.8 \cdot 10^0$	0	2
$6.0 \cdot 10^{-2}$	91.4	$5.3 \cdot 10^0$	76.1	$1.6 \cdot 10^1$	6.8	$1.5 \cdot 10^0$	0	69
$8.0 \cdot 10^{-2}$	92.8	$8.5 \cdot 10^0$	55.7	$1.4 \cdot 10^1$	6.0	$1.4 \cdot 10^0$	0	83
$1.0 \cdot 10^{-1}$	94.8	$1.2 \cdot 10^1$	42.9	$1.9 \cdot 10^1$	6.8	$1.3 \cdot 10^0$	0	95
$2.0 \cdot 10^{-1}$	97.1	$3.3 \cdot 10^0$	26.2	$2.0 \cdot 10^1$	7.0	$1.5 \cdot 10^0$	0	97
$3.0 \cdot 10^{-1}$	97.3	$3.5 \cdot 10^0$	22.6	$1.8 \cdot 10^1$	7.2	$1.8 \cdot 10^0$	0	91
$4.0 \cdot 10^{-1}$	98.4	$2.5 \cdot 10^0$	22.5	$1.6 \cdot 10^1$	8.0	$1.6 \cdot 10^0$	0	97
$5.0 \cdot 10^{-1}$	98.8	$9.6 \cdot 10^{-1}$	20.7	$1.7 \cdot 10^1$	7.3	$1.5 \cdot 10^0$	0	100

TABLE 5.12: Results of the CFD dense database contaminated with positive, random error of varying maximum magnitude concentrated in an elliptical cylinder of 0.35 b parameter. For column headers please refer to Table 5.7.

Once again, the probability for absolute convergence is zero as there are always some clean points being mistaken as erroneous ones. However, the probability for relative convergence is high up to $1.0 \cdot 10^{-1}$ error maximum magnitude which also sets the convergence limit. It is interesting to see that, in contrary to

the toy model, the probability of relative convergence decreases steadily. It is a consequence of the fact that the method is capable of detecting less and less erroneous points. This gradual change can be appreciated in Figure 5.12-left. In terms of reconstruction errors, MaxE is always below the value set for relative convergence (10^{-1}), but RMSE is smaller than 10^{-2} at high values of error maximum magnitude and for values below $6.0 \cdot 10^{-2}$ it becomes higher (see Figure 5.12-right).

FIGURE 5.12: Complement of Table 5.12. Left: %CDEP_m (thick black line) and %IDEP_m (thin black line). Right: mean MaxE and RMSE of the reconstructed part (solid and dashed line respectively), as defined in Equation (2.18).

The counterpart for the coarse database of Table 5.12 and Figure 5.12 is shown in Table B.6 and Figure B.6 respectively. The method clearly performs worse as it is incapable of reaching the convergence limit, despite the fact that until $2.0 \cdot 10^{-1}$ it is capable of identifying on average above 94.5% of points. However, the reconstruction errors are very high and which hinders reaching high probability of relative convergence. As shown before, this is due to the poor resolution of the mesh.

Chapter 6

Automatic tuning of method parameters

Previous chapters demonstrated that both SVD- and HOSVD-filtering methods work well and do not require *a priori* knowledge about the error. However, both methods have a number of tunable parameters which need to be pre-set in order to guarantee correct functioning. For the 2D or 3D cases those parameters appear in Equations (3.12), (3.15) and (3.17) or Equations (3.1), (3.5) and (3.7) respectively, and they are:

$$CF, \quad s_{\min}, \quad FF, \quad N^{\min}, \quad N^{\max}, \quad N^-, \quad N^+. \quad (6.1)$$

From Equation (6.1), CF and $s_{\min}$ are related to the gappy reconstruction method and their influence on the reconstruction errors have been proven insignificant [35, 36]. Therefore, for now on $CF = 0.1$ and $s_{\min} = 4$, are set analogically to previous analyses. FF quantifies how aggressive the method is in finding errors. On the other hand $N^{\min}$, $N^{\max}$, N^- and N^+ depend on the size and nature of the clean database and so far have been optimised for best performance of the method using clean databases as a reference. In real life, however, clean databases are almost never available and fixing those parameters should be done automatically. Section 6.1 will describe a method based on MSE, defined for 2D and 3D in Equation (3.10) and (3.20) respectively, which is able to automatically choose the number of modes to be retained and hence *a priori* setting of $N^{\min}$, $N^{\max}$, N^- and N^+ is no longer needed. For the sake of brevity, the method is described for 3D databases. Extrapolating it to 2D matrices or higher order tensors is straight forward.

6.1 Mode number selection using MSE

Sections 3.4 and 4.2 have shown that the first iteration of the filtering method is very important as it has a strong influence on further applications. This is due to the fact that the information about detected erroneous points, whether correct or not, are stored only after each iteration of the method. Therefore,

the first step is to determine local $N^{\min}$ and $N^{\max}$ for the first iteration of the method.

It has been observed that in order for HOSVD-filtering method to function correctly at least 2 modes per direction have to be retained, i.e. $m = 2N$ where N is the tensor order. This guarantees that the method will retain enough information to correctly detect erroneous points and that the reconstruction errors will be sufficiently small. However, this is not always sufficient due to the fact that the optimal number modes used for reconstruction is chosen according to the minimum value of MSE defined in Equation (3.14). In some cases MSE is very low for very few modes and provokes the method to detect many clean points as erroneous and thus significantly affecting the final results of the filtering method.

Applying the method once, treating such application as a pilot trial, can provide useful information regarding the nature of the method. A useful parameter, which is known regardless of any knowledge about the error-free database, is the root mean square error in the clean part defined as:

$$\text{RMSE}_{\text{clean}}^m = \|A_{i_1 i_2 i_3} - \tilde{A}_{i_1 i_2 i_3}^m\|_{\text{RMS}} \quad \text{for } (i_1, i_2, i_3) \in S_{\text{clean}}^m, \quad (6.2)$$

where A is the original tensor and $\tilde{A}^m$ is the reconstruction corresponding to m retained modes, namely the improved gappy virtual reconstruction tensor. S_{clean}^m is the index set corresponding to the clean database position, i.e. the points not treated by the method. S_{clean}^m is chosen according to Equation (3.15) and since at the first iteration the method has not identified any errors, S_{clean}^m changes whenever m is incremented. $\text{RMSE}_{\text{clean}}^m$ describes how well the tensor is reconstructed for all the points that the method considers clean in the first iteration. It has been observed that the minimum number of retained modes coincides approximately when the derivative of $\text{RMSE}_{\text{clean}}^m$ with respect to m is smallest. Hence the minimum number of modes is defined by

$$N_1^{\min} = m \quad \text{when} \quad \min_m (\text{RMSE}_{\text{clean}}^m - \text{RMSE}_{\text{clean}}^{m-1}) \quad \text{for } m = 2 \cdot N, \dots, \alpha \cdot M, \quad (6.3)$$

where M is the total number of modes, N is the tensor order and α prevents retaining all the modes and is set to 0.5 after some preliminary trials. $N_1^{\min}$ is then normalised with M , i.e. $f_1^{\min} = N_1^{\min}/M$. $N_1^{\min}$ and $N_1^{\max}$ can be then defined as

$$N_1^{\min} = f_1^{\min} \cdot M \quad \text{and} \quad N_1^{\max} = f_1^{\max} \cdot M, \quad \text{where } f_1^{\max} = f_1^{\min} + \Delta f, \quad (6.4)$$

where Δf is the normalised difference between maximum and minimum retained modes. It has been shown that in the first application of the method the minimum number is crucial and retaining more modes does not affect the accuracy, hence $\Delta f = 0.1$ is sufficient. Now the method can be applied for the first time using $N_{\min}$ and $N_{\max}$ defined in Equation (6.4).

Once the method has been run for the first time, N_{min} and N_{max} have to be incremented in order to increase the reconstruction precision. By analysing bottom right plot of Figure 3.7 and for toy model and CFD database respectively, a linear behaviour of $\log_{10} \text{MSE}_k$, defined in Equation (3.20), with respect to the number of method iterations k , can be observed. Also the optimum retained modes N_k^{opt} increases linearly with respect to k . Since MSE_k can be calculated without knowing the clean database, N_k^{min} can be calculated using

$$\begin{aligned} f_k^{min} &= a^{min} \log_{10} \text{MSE}_{k-1} + b^{min} \\ \text{where } a^{min} &= \frac{f_1^{min} - f^{min}}{\log_{10} \text{MSE}^{max} - \log_{10} \text{MSE}^{min}} \quad \text{and} \\ b^{min} &= f_1^{min} - a^{min} \cdot \log_{10} \text{MSE}^{max}, \end{aligned} \quad (6.5)$$

where f_k^{max} is the normalized N_k^{min} for the k -th application. Analogically, N_k^{max} can be calculated using

$$\begin{aligned} f_k^{max} &= a^{max} \log_{10} \text{MSE}_{k-1} + b^{max} \\ \text{where } a^{max} &= \frac{f_1^{max} - f^{max}}{\log_{10} \text{MSE}^{max} - \log_{10} \text{MSE}^{min}} \quad \text{and} \\ b^{max} &= f_1^{max} - a^{max} \cdot \log_{10} \text{MSE}^{max}. \end{aligned} \quad (6.6)$$

Equation (6.5) and (6.5) require setting 4 new parameters, namely f^{min} , f^{max} , MSE^{min} and MSE^{max} . f^{min} and f^{max} are bounds for the normalized minimum and maximum number of retained modes such that

$$N_k^{min} \leq f^{min} M \quad \text{and} \quad N_k^{max} \leq f^{max} M. \quad (6.7)$$

MSE^{min} and MSE^{max} quantify the reconstruction difference between two tensors of consecutive method applications defined in Equation (3.20). This defines a new mode number selection method from now on called *MSE automatic mode selection method*. The mode selection method defined in Chapter 3, Equation (3.7) and (3.17) for SVD-filtering and HOSVD-filtering respectively, will be called *manual mode selection method*.

Although the MSE mode selection method introduces 4 parameters, not including α and Δf from Equation (6.3) and (6.4) respectively, which is the same number of parameters needed using the manual mode selection method, f^{min} , f^{max} , MSE^{min} and MSE^{max} are virtually independent of the nature of the tensor. This statement is later verified in Section 6.2.

6.2 Verification

The purpose of this section is to verify whether the method of selecting the number of modes for the HOSVD-filtering proposed in Section 6.1 is robust and accurate. For this purpose, the toy model and CFD databases are going to be

contaminated with the same error magnitude and distribution as in the analyses performed in Chapter 5. The same approach applies to the parameters which are kept the same. Specifically, FF parameter is fixed to 0.3 and 0.4 for toy model and CFD database respectively. The MSE mode selection method parameters for all the databases were selected after an initial exploration study and are

$$f^{min} = 0.2, f^{max} = 0.5, \text{MSE}^{min} = 10^{-9} \text{ and } \text{MSE}^{max} = 10^1. \quad (6.8)$$

6.2.1 Random errors at randomly chosen database positions

Toy model database

The first major conclusion that one can draw from comparing Table 5.1 with 6.1 is that for the manual method the absolute convergence is much more probable. The MSE automatic mode selection method has a relatively low probability of absolute convergence reaching a maximum value of 16% at 42.5% contamination level. In comparison, the manual method descends monotonously from 94% probability at 32.5% error contamination to 16% at 67.5%. However, both methods are capable of finding above 90% of all errors up to error concentration levels of 77.5%. Additionally, both methods exhibit the same pattern for both finding erroneous points and mistakingly taking clean points as errors (see Figure 6.1-left). The reconstruction errors are also very similar (see Figure 6.1-right) where the MSE automatic method achieves slightly worse results. Finally, the convergence limit is 70.0% and 72.5% for the manual and MSE automatic mode selection method respectively. Therefore, from an engineering point of view, where over 90% of erroneous points need to be found and repaired with a high degree of accuracy, both methods perform equally. However, as said before, the manual method requires *a priori* knowledge about the clean database whereas MSE uses the same parameters for any database.

The counterpart for the coarse database of Table 6.1 and Figure 6.1 is shown in Table C.1 and Figure C.1 respectively. The method using automatic MSE automatic mode selection identifies well the erroneous points as it is capable of identifying on average 90% and above of them up to error contamination level of 57.5%. At the same time it identifies wrongly on average about 5% of clean points as erroneous. Due to this fact the probability of absolute convergence is very low. The relative convergence reaches a probability of 80% for low error contamination levels and hence does not ever reach convergence limit. When comparing this mode selection method to the manual one, both perform almost the same in terms of erroneous points identification. However, the automatic mode selection performs worse when it comes to reconstructing missing information, especially for low error contamination levels. This is probably due to the fact that it retains less modes to reconstruct the database.

%cont	%CD _m	%CD _{std}	%ID _m	%ID _{std}	k _m	k _{std}	%C ₁	%C ₂
32.5	100.0	$3.5 \cdot 10^{-4}$	0.0	$3.6 \cdot 10^{-3}$	10.7	$7.1 \cdot 10^{-1}$	4	100
35.0	100.0	$2.3 \cdot 10^{-4}$	0.0	$4.2 \cdot 10^{-3}$	10.8	$8.1 \cdot 10^{-1}$	8	100
37.5	100.0	$2.1 \cdot 10^{-4}$	0.0	$3.3 \cdot 10^{-3}$	10.9	$7.7 \cdot 10^{-1}$	7	100
40.0	100.0	$2.0 \cdot 10^{-4}$	0.0	$4.2 \cdot 10^{-3}$	10.9	$7.8 \cdot 10^{-1}$	10	100
42.5	100.0	$1.9 \cdot 10^{-4}$	0.0	$3.2 \cdot 10^{-3}$	10.9	$8.4 \cdot 10^{-1}$	16	100
45.0	100.0	$2.5 \cdot 10^{-4}$	0.0	$3.4 \cdot 10^{-3}$	11.0	$8.6 \cdot 10^{-1}$	14	100
47.5	100.0	$2.9 \cdot 10^{-4}$	0.0	$2.8 \cdot 10^{-3}$	11.1	$8.2 \cdot 10^{-1}$	15	100
50.0	100.0	$6.6 \cdot 10^{-4}$	0.0	$3.9 \cdot 10^{-3}$	11.2	$8.4 \cdot 10^{-1}$	10	100
52.5	100.0	$4.5 \cdot 10^{-4}$	0.0	$3.3 \cdot 10^{-3}$	11.3	$7.7 \cdot 10^{-1}$	10	100
55.0	100.0	$7.0 \cdot 10^{-4}$	0.0	$4.7 \cdot 10^{-3}$	11.4	$8.1 \cdot 10^{-1}$	9	100
57.5	100.0	$1.1 \cdot 10^{-3}$	0.0	$3.7 \cdot 10^{-3}$	11.4	$8.6 \cdot 10^{-1}$	9	100
60.0	100.0	$1.4 \cdot 10^{-3}$	0.0	$4.0 \cdot 10^{-3}$	11.5	$8.3 \cdot 10^{-1}$	13	100
62.5	99.9	$3.6 \cdot 10^{-1}$	0.0	$4.6 \cdot 10^{-3}$	11.2	$1.6 \cdot 10^0$	9	100
65.0	99.7	$7.5 \cdot 10^{-1}$	0.0	$5.9 \cdot 10^{-3}$	10.1	$2.9 \cdot 10^0$	6	100
67.5	99.1	$1.6 \cdot 10^0$	0.0	$5.9 \cdot 10^{-3}$	8.5	$3.5 \cdot 10^0$	3	100
70.0	97.8	$2.2 \cdot 10^0$	0.0	$2.0 \cdot 10^{-1}$	6.2	$3.2 \cdot 10^0$	0	98
72.5	96.1	$2.6 \cdot 10^0$	0.1	$3.3 \cdot 10^{-1}$	4.4	$1.6 \cdot 10^0$	0	90
75.0	94.5	$2.8 \cdot 10^0$	0.1	$3.6 \cdot 10^{-1}$	4.0	$7.8 \cdot 10^{-1}$	0	78
77.5	92.4	$4.4 \cdot 10^0$	0.2	$3.2 \cdot 10^{-1}$	4.1	$8.4 \cdot 10^{-1}$	0	63
80.0	88.7	$6.6 \cdot 10^0$	1.1	$1.3 \cdot 10^0$	4.8	$1.2 \cdot 10^0$	0	31
82.5	81.7	$8.1 \cdot 10^0$	6.4	$4.2 \cdot 10^0$	5.6	$2.2 \cdot 10^0$	0	0
85.0	76.0	$7.1 \cdot 10^0$	10.4	$3.3 \cdot 10^0$	5.6	$2.0 \cdot 10^0$	0	0
87.5	72.9	$5.1 \cdot 10^0$	10.9	$1.4 \cdot 10^0$	5.3	$1.8 \cdot 10^0$	0	0
90.0	70.1	$5.3 \cdot 10^0$	9.3	$5.4 \cdot 10^{-1}$	5.0	$1.7 \cdot 10^0$	0	0

TABLE 6.1: Results of the dense 3D toy model database contaminated with positive, random error of maximum magnitude 0.5 at randomly chosen database points for varying contamination percentage. Filtering was performed using MSE automatic mode selection. For column headers please refer to Table 5.1.

CFD database

A similar situation occurs for the CFD dense database, which results for the manual and MSE automatic method can be appreciated in Table 5.2 and 6.2 respectively. Neither of those methods achieves absolute convergence, which is not surprising for engineering databases, especially ones that experience very steep gradients. However, in this case the MSE automatic method performs better than the manual one. Firstly, it manages to identify on average more erroneous points and commits fewer mistakes (see Figure 6.2-left). For instance, it detects on average above 90% of erroneous points up to error concentration level of 47.5% whereas for the manual method it is 42.5%. The reconstruction errors are also always lower for the MSE automatic method (see Figure 6.2-right). Finally, the convergence limit for the manual and MSE automatic method is

FIGURE 6.1: Complement of Table 6.1. Manual and MSE automatic mode selection is shown in black and red lines respectively. Left: $\%CDEP_m$ with the corresponding $\%CDEP_{std}$ bars (thick line) and $\%IDEF_m$ with the corresponding $\%IDEF_{std}$ bars (thin line). Right: mean MaxE and RMSE of the reconstructed part (solid and dashed line respectively), as defined in Equation (2.18).

37.5% and 40.0% which indicates a slightly better performance of the MSE automatic method. Therefore, not only the MSE mode selection method does not need any information about the clean database, but it also performs better than the manual method.

FIGURE 6.2: Complement of Table 6.2. Manual and MSE automatic mode selection is shown in black and red lines respectively. Left: $\%CDEP_m$ with the corresponding $\%CDEP_{std}$ bars (thick line) and $\%IDEF_m$ with the corresponding $\%IDEF_{std}$ bars (thin line). Right: mean MaxE and RMSE of the reconstructed part (solid and dashed line respectively), as defined in Equation (2.18).

The counterpart for the coarse database of Table 6.2 and Figure 6.2 is shown in Table C.2 and Figure C.2 respectively. The MSE automatic mode selection method performs better than the manual one as it identifies more erroneous

%cont	%CD _m	%CD _{std}	%ID _m	%ID _{std}	k _m	k _{std}	%C ₁	%C ₂
5.0	99.1	$4.8 \cdot 10^{-1}$	1.8	$1.1 \cdot 10^0$	6.7	$1.1 \cdot 10^0$	0	100
7.5	99.2	$3.8 \cdot 10^{-1}$	1.1	$6.7 \cdot 10^{-1}$	7.0	$1.1 \cdot 10^0$	0	100
10.0	99.1	$3.9 \cdot 10^{-1}$	0.8	$5.6 \cdot 10^{-1}$	7.1	$1.1 \cdot 10^0$	0	100
12.5	99.1	$5.3 \cdot 10^{-1}$	0.6	$3.7 \cdot 10^{-1}$	7.2	$1.2 \cdot 10^0$	0	100
15.0	99.1	$6.9 \cdot 10^{-1}$	0.6	$3.0 \cdot 10^{-1}$	7.6	$1.2 \cdot 10^0$	0	100
17.5	99.0	$7.6 \cdot 10^{-1}$	0.5	$3.8 \cdot 10^{-1}$	7.7	$1.4 \cdot 10^0$	0	100
20.0	98.9	$7.2 \cdot 10^{-1}$	0.4	$2.9 \cdot 10^{-1}$	7.4	$1.3 \cdot 10^0$	0	100
22.5	99.1	$4.7 \cdot 10^{-1}$	0.5	$3.5 \cdot 10^{-1}$	8.3	$1.4 \cdot 10^0$	0	100
25.0	99.0	$5.6 \cdot 10^{-1}$	0.4	$3.0 \cdot 10^{-1}$	8.0	$1.4 \cdot 10^0$	0	100
27.5	99.0	$6.5 \cdot 10^{-1}$	0.4	$3.5 \cdot 10^{-1}$	8.3	$1.6 \cdot 10^0$	0	100
30.0	99.0	$5.4 \cdot 10^{-1}$	0.4	$2.4 \cdot 10^{-1}$	8.6	$1.5 \cdot 10^0$	0	100
32.5	98.9	$9.1 \cdot 10^{-1}$	0.4	$2.3 \cdot 10^{-1}$	8.8	$1.6 \cdot 10^0$	0	100
35.0	98.8	$1.5 \cdot 10^0$	0.4	$2.2 \cdot 10^{-1}$	8.7	$1.7 \cdot 10^0$	0	99
37.5	98.5	$2.0 \cdot 10^0$	0.4	$2.4 \cdot 10^{-1}$	8.6	$2.0 \cdot 10^0$	0	98
40.0	97.2	$3.3 \cdot 10^0$	0.3	$4.1 \cdot 10^{-1}$	7.3	$2.3 \cdot 10^0$	0	92
42.5	95.4	$3.6 \cdot 10^0$	0.2	$3.6 \cdot 10^{-1}$	6.0	$1.9 \cdot 10^0$	0	87
45.0	93.4	$4.2 \cdot 10^0$	0.2	$4.0 \cdot 10^{-1}$	5.1	$1.2 \cdot 10^0$	0	79
47.5	90.8	$4.8 \cdot 10^0$	0.3	$9.5 \cdot 10^{-1}$	4.5	$1.1 \cdot 10^0$	0	60
50.0	87.9	$5.3 \cdot 10^0$	0.5	$1.3 \cdot 10^0$	4.2	$8.2 \cdot 10^{-1}$	0	32
52.5	82.8	$7.7 \cdot 10^0$	1.7	$2.6 \cdot 10^0$	4.0	$6.1 \cdot 10^{-1}$	0	9
55.0	79.9	$6.8 \cdot 10^0$	2.3	$2.8 \cdot 10^0$	3.9	$8.3 \cdot 10^{-1}$	0	0
57.5	76.0	$9.0 \cdot 10^0$	3.1	$3.1 \cdot 10^0$	4.0	$1.2 \cdot 10^0$	0	0
60.0	73.2	$7.4 \cdot 10^0$	3.7	$2.8 \cdot 10^0$	4.1	$1.1 \cdot 10^0$	0	0
62.5	70.4	$9.4 \cdot 10^0$	3.9	$2.4 \cdot 10^0$	4.1	$1.4 \cdot 10^0$	0	0

TABLE 6.2: Results of the dense CFD database contaminated with positive, random error of maximum magnitude 0.5 at randomly chosen database points for varying contamination percentage. Filtering was performed using MSE automatic mode selection. For column headers refer to Table 5.1.

points, especially for high error concentration levels. It also commits less mistakes as it identifies less IDEP, especially for high error concentration levels. If it goes for the reconstruction errors, they are higher in the case of MSE automatic mode selection method for low error concentration but above 20% they become lower than to ones committed by the manual method. The convergence limit of this method is 20%, equal to the one for the manual method, but also decreases with the decrease of the error concentration.

6.2.2 Systematic errors at randomly chosen database positions

Toy model database

Similarly to the positive, random error, using the MSE automatic method to choose the minimum and maximum number of retained modes results in zero probability for absolute convergence as shown in Table 6.3. However, using manual mode selection method, the highest probability of absolute convergence is 16% at 32.5% error contamination (see Table 5.3). In this case what is more interesting is the relative convergence and the limit of convergence. For the MSE and manual method the convergence limit is for 45.0% error contamination for both of them as the probability of relative convergence is above 90%. This is due to the fact that the %CDEP curves shown in Figure 6.3-left for both methods are virtually identical and, although for the MSE automatic method the reconstruction errors are slightly higher at the beginning (until about 40% error concentration), they then closely resemble those of the manual method (see Figure 6.3-right). As for the number of iterations that both methods need to converge, the MSE automatic method requires on average one iteration more. As for the toy model dense database, both mode selection methods perform similarly with a slight advantage on the manual method side.

FIGURE 6.3: Complement of Table 6.3. Manual and MSE automatic mode selection is shown in black and red lines respectively. Left: %CDEP_m with the corresponding %CDEP_{std} bars (thick line) and %IDEP_m with the corresponding %IDEP_{std} bars (thin line). Right: mean MaxE and RMSE of the reconstructed part (solid and dashed line respectively), as defined in Equation (2.18).

The counterpart for the coarse database of Table 6.3 and Figure 6.3 is shown in Table C.3 and Figure C.3 respectively. The method once again is very efficient at identifying erroneous points as it is capable of more than 90% of them for error contamination level up to 37.5%. Nevertheless, it underperforms when it comes to database reconstruction. In this case the difference between the two methods is very significant as the automatic MSE mode selection reconstruction

%cont	%CD _m	%CD _{std}	%ID _m	%ID _{std}	k _m	k _{std}	%C ₁	%C ₂
32.5	100.0	$1.1 \cdot 10^{-3}$	0.0	$1.1 \cdot 10^{-1}$	6.3	$6.0 \cdot 10^{-1}$	0	100
35.0	100.0	$1.0 \cdot 10^{-3}$	0.0	$6.0 \cdot 10^{-2}$	6.4	$5.9 \cdot 10^{-1}$	0	100
37.5	100.0	$1.0 \cdot 10^{-3}$	0.0	$1.7 \cdot 10^{-2}$	6.3	$5.1 \cdot 10^{-1}$	0	100
40.0	100.0	$9.7 \cdot 10^{-4}$	0.0	$4.9 \cdot 10^{-2}$	6.5	$5.8 \cdot 10^{-1}$	0	100
42.5	100.0	$9.3 \cdot 10^{-4}$	0.1	$8.2 \cdot 10^{-2}$	6.7	$6.5 \cdot 10^{-1}$	0	100
45.0	100.0	$2.8 \cdot 10^{-1}$	0.2	$2.6 \cdot 10^{-1}$	6.7	$9.6 \cdot 10^{-1}$	0	99
47.5	99.2	$1.4 \cdot 10^0$	2.5	$1.8 \cdot 10^0$	5.9	$2.0 \cdot 10^0$	0	62
50.0	85.6	$1.4 \cdot 10^1$	15.4	$6.6 \cdot 10^0$	5.7	$2.6 \cdot 10^0$	0	1
52.5	42.5	$1.4 \cdot 10^1$	49.3	$1.7 \cdot 10^1$	5.4	$2.5 \cdot 10^0$	0	0
55.0	17.4	$4.1 \cdot 10^0$	67.8	$7.5 \cdot 10^0$	5.7	$2.2 \cdot 10^0$	0	0
57.5	9.4	$3.5 \cdot 10^0$	68.1	$2.7 \cdot 10^0$	5.4	$1.7 \cdot 10^0$	0	0
60.0	5.1	$2.3 \cdot 10^0$	64.2	$1.3 \cdot 10^0$	5.8	$1.3 \cdot 10^0$	0	0
62.5	3.0	$1.7 \cdot 10^0$	58.7	$1.7 \cdot 10^0$	5.4	$1.4 \cdot 10^0$	0	0
65.0	1.9	$1.1 \cdot 10^0$	53.1	$9.4 \cdot 10^{-1}$	5.0	$1.2 \cdot 10^0$	0	0
67.5	1.1	$7.4 \cdot 10^{-1}$	47.6	$1.3 \cdot 10^0$	5.0	$1.1 \cdot 10^0$	0	0
70.0	0.5	$5.9 \cdot 10^{-1}$	42.7	$1.7 \cdot 10^{-1}$	5.2	$1.2 \cdot 10^0$	0	0
72.5	0.1	$2.3 \cdot 10^{-1}$	37.9	$5.8 \cdot 10^{-2}$	6.0	$8.3 \cdot 10^{-1}$	0	0
75.0	0.0	$9.4 \cdot 10^{-2}$	33.3	$3.1 \cdot 10^{-2}$	6.0	$7.0 \cdot 10^{-1}$	0	0
77.5	0.0	$4.9 \cdot 10^{-3}$	29.0	$4.6 \cdot 10^{-4}$	6.1	$5.7 \cdot 10^{-1}$	0	0
80.0	0.0	$5.3 \cdot 10^{-3}$	25.0	$4.2 \cdot 10^{-4}$	6.2	$5.1 \cdot 10^{-1}$	0	0
82.5	0.0	$5.5 \cdot 10^{-3}$	21.2	$3.9 \cdot 10^{-4}$	6.2	$5.2 \cdot 10^{-1}$	0	0
85.0	0.0	$5.3 \cdot 10^{-3}$	17.6	$3.5 \cdot 10^{-4}$	6.1	$5.6 \cdot 10^{-1}$	0	0
87.5	0.0	$1.2 \cdot 10^{-2}$	14.3	$3.1 \cdot 10^{-4}$	6.0	$6.8 \cdot 10^{-1}$	0	0
90.0	0.0	$7.1 \cdot 10^{-3}$	11.1	$2.7 \cdot 10^{-4}$	6.0	$8.5 \cdot 10^{-1}$	0	0

TABLE 6.3: Results of the dense 3D toy model database contaminated with a systematic error obtained by multiplying the database by 1.5 at randomly chosen database points for varying contamination percentage. Filtering was performed using MSE automatic mode selection. For column headers refer to Table 5.1.

errors are too big to satisfy the relative convergence criteria. For this reason the method performs poorly for this kind of mesh density and error characteristics.

CFD database

Table 5.4 and 6.4 show the filtering analysis for the dense CFD database contaminated with systematic errors resulting from multiplying the database by 1.5 at random positions using the manual and MSE automatic mode selection method respectively. The analysis was performed for a range of error contamination percentage (5.0 to 62.5 at 2.5% steps). The convergence limit for both methods is 30.0% error concentration and surprisingly in this case the probability of achieving absolute convergence for the MSE mode selection method is 70% at 5% error concentration, whereas it is always zero for the manual method. If one compares the manual and MSE automatic mode selection method in terms

of CDEP percentage (see Figure 6.4-left), it can be seen that the MSE automatic method performs exactly the same until about 40% error concentration and then it becomes worse in comparison to the manual selection method. At the same time the reconstruction errors presented in Figure 6.4-right are nearly the same for both methods. In summary, the performance of both methods can be considered the same.

%cont	%CD _m	%CD _{std}	%ID _m	%ID _{std}	k _m	k _{std}	%C ₁	%C ₂
5.0	99.9	$1.6 \cdot 10^{-1}$	1.2	$1.0 \cdot 10^0$	5.4	$1.4 \cdot 10^0$	70	99
7.5	99.7	$1.6 \cdot 10^0$	0.8	$6.8 \cdot 10^{-1}$	5.6	$1.4 \cdot 10^0$	1	96
10.0	99.9	$1.1 \cdot 10^{-1}$	0.6	$5.2 \cdot 10^{-1}$	6.1	$1.1 \cdot 10^0$	1	100
12.5	99.9	$4.1 \cdot 10^{-1}$	0.4	$3.4 \cdot 10^{-1}$	6.1	$1.3 \cdot 10^0$	0	98
15.0	99.9	$6.8 \cdot 10^{-2}$	0.3	$2.5 \cdot 10^{-1}$	6.4	$1.3 \cdot 10^0$	0	100
17.5	99.9	$2.8 \cdot 10^{-1}$	0.3	$2.2 \cdot 10^{-1}$	6.5	$1.2 \cdot 10^0$	0	98
20.0	99.9	$1.1 \cdot 10^{-1}$	0.3	$1.9 \cdot 10^{-1}$	6.7	$1.4 \cdot 10^0$	0	100
22.5	99.8	$6.3 \cdot 10^{-1}$	0.2	$1.7 \cdot 10^{-1}$	6.8	$1.5 \cdot 10^0$	0	98
25.0	99.9	$1.6 \cdot 10^{-1}$	0.2	$1.2 \cdot 10^{-1}$	6.8	$1.4 \cdot 10^0$	0	99
27.5	99.9	$1.9 \cdot 10^{-1}$	0.1	$1.1 \cdot 10^{-1}$	6.7	$1.4 \cdot 10^0$	0	98
30.0	99.8	$2.8 \cdot 10^{-1}$	0.2	$3.9 \cdot 10^{-1}$	6.8	$1.5 \cdot 10^0$	0	94
32.5	99.7	$6.8 \cdot 10^{-1}$	0.6	$1.2 \cdot 10^0$	6.6	$1.9 \cdot 10^0$	0	77
35.0	99.2	$1.4 \cdot 10^0$	2.3	$2.7 \cdot 10^0$	6.4	$2.0 \cdot 10^0$	0	33
37.5	97.4	$6.0 \cdot 10^0$	3.7	$3.9 \cdot 10^0$	5.8	$2.7 \cdot 10^0$	0	19
40.0	96.0	$2.5 \cdot 10^0$	6.2	$5.1 \cdot 10^0$	4.0	$2.5 \cdot 10^0$	0	6
42.5	90.9	$1.0 \cdot 10^1$	4.8	$3.1 \cdot 10^0$	3.6	$1.6 \cdot 10^0$	0	0
45.0	83.5	$1.6 \cdot 10^1$	6.3	$3.4 \cdot 10^0$	3.1	$9.8 \cdot 10^{-1}$	0	0
47.5	68.6	$2.5 \cdot 10^1$	9.6	$4.3 \cdot 10^0$	2.5	$8.2 \cdot 10^{-1}$	0	0
50.0	54.2	$2.1 \cdot 10^1$	15.6	$7.5 \cdot 10^0$	2.4	$1.1 \cdot 10^0$	0	0
52.5	39.7	$1.7 \cdot 10^1$	22.3	$1.2 \cdot 10^1$	2.4	$10.0 \cdot 10^{-1}$	0	0
55.0	26.4	$1.3 \cdot 10^1$	31.3	$1.4 \cdot 10^1$	2.6	$1.3 \cdot 10^0$	0	0
57.5	12.2	$8.7 \cdot 10^0$	47.3	$1.7 \cdot 10^1$	3.5	$1.6 \cdot 10^0$	0	0
60.0	4.8	$2.4 \cdot 10^0$	58.2	$5.9 \cdot 10^0$	4.3	$1.7 \cdot 10^0$	0	0
62.5	1.8	$1.4 \cdot 10^0$	56.7	$2.6 \cdot 10^0$	4.3	$1.6 \cdot 10^0$	0	0

TABLE 6.4: Results of the dense CFD database contaminated with a systematic error obtained by multiplying the database by 1.5 at randomly chosen database points for varying contamination percentage. Filtering was performed using MSE automatic mode selection. For column headers refer to Table 5.1.

The counterpart for the coarse database of Table 6.4 and Figure 6.4 is shown in Table C.4 and Figure C.4 respectively. In this case the MSE automatic mode selection method performs similar to the manual one. For low error concentration levels it identifies the same percentage of erroneous points. At the same time it commits slightly higher reconstruction errors. For higher concentrations, it performs less, but this decrease in performance is not significant. The reconstruction errors are in this case lower than of the manual method. The convergence limit in this case is 20%, similar to the manual method.

FIGURE 6.4: Complement of Table 6.4. Manual and MSE automatic mode selection is shown in black and red lines respectively. Left: $\%CDEP_m$ with the corresponding $\%CDEP_{std}$ bars (thick line) and $\%IDEP_m$ with the corresponding $\%IDEP_{std}$ bars (thin line). Right: mean MaxE and RMSE of the reconstructed part (solid and dashed line respectively), as defined in Equation (2.18).

6.2.3 Random errors at concentrated database positions

Toy model database

Table 5.5 and 6.5 show the filtering analysis for the dense toy model database contaminated with positive, random errors of maximum magnitude of 0.5 in a circular shaped cylinder using the manual and MSE automatic mode selection method respectively. The analysis was performed for a range of cylinder base circle diameters (0.275 to 0.850 at 0.025 steps). Once again, the MSE automatic method underperforms if the absolute convergence is considered. However, the convergence limit for both methods are the same and occur for a 0.425 cylinder base circle diameter. When comparing the percentage of CDEP shown in Figure 6.5-left it can be deduced that both methods perform exactly the same. In terms of the reconstruction errors, the MSE automatic method is two orders of magnitude above the manual method, still within the convergence limit conditions though, and at 0.450 diameter equals the performance of the manual method (see Figure 6.5). In the end, both methods perform similarly.

The counterpart for the coarse database of Table 6.5 and Figure 6.5 is shown in Table C.5 and Figure C.5 respectively. In this case the method identifies well the erroneous points (on average 90% of CDEP up circle diameter of 0.3), a result better than for manual method, but at the same time it incorrectly identifies clean points as erroneous ones, especially for small circle diameters (up to diameter 0.15). What is more, the standard deviation of the latter is very high (above 10% up to circle diameter of 0.475), meaning that the method behaves quite erratically. Nevertheless, the database reconstruction errors are smaller for MSE automatic than manual method for small circle diameters which means

d	%CD _m	%CD _{std}	%ID _m	%ID _{std}	k _m	k _{std}	%C ₁	%C ₂
0.275	100.0	$6.2 \cdot 10^{-2}$	0.1	$5.4 \cdot 10^{-2}$	10.1	$1.4 \cdot 10^0$	2	100
0.300	100.0	$8.2 \cdot 10^{-2}$	0.0	$1.8 \cdot 10^{-2}$	9.7	$1.7 \cdot 10^0$	2	100
0.325	100.0	$1.1 \cdot 10^{-1}$	0.0	$1.9 \cdot 10^{-2}$	9.9	$1.7 \cdot 10^0$	1	100
0.350	100.0	$10.0 \cdot 10^{-2}$	0.0	$1.6 \cdot 10^{-2}$	11.1	$1.8 \cdot 10^0$	0	100
0.375	100.0	$1.1 \cdot 10^{-1}$	0.0	$1.1 \cdot 10^{-2}$	10.4	$1.6 \cdot 10^0$	0	100
0.400	100.0	$6.5 \cdot 10^{-2}$	0.0	$7.7 \cdot 10^{-2}$	10.7	$1.5 \cdot 10^0$	2	100
0.425	100.0	$8.0 \cdot 10^{-2}$	0.1	$2.3 \cdot 10^{-1}$	12.4	$2.7 \cdot 10^0$	2	100
0.450	88.5	$1.5 \cdot 10^1$	0.1	$2.1 \cdot 10^{-1}$	7.5	$4.3 \cdot 10^0$	13	57
0.475	79.5	$9.3 \cdot 10^0$	0.0	$1.7 \cdot 10^{-1}$	4.2	$3.1 \cdot 10^0$	0	12
0.500	79.0	$7.4 \cdot 10^0$	0.0	$2.4 \cdot 10^{-1}$	3.8	$2.4 \cdot 10^0$	0	5
0.525	76.8	$5.0 \cdot 10^0$	0.0	$5.5 \cdot 10^{-2}$	3.4	$9.7 \cdot 10^{-1}$	0	0
0.550	77.3	$5.0 \cdot 10^0$	0.0	$3.0 \cdot 10^{-2}$	3.6	$1.1 \cdot 10^0$	0	0
0.575	75.7	$9.5 \cdot 10^0$	0.0	$5.5 \cdot 10^{-2}$	3.5	$1.1 \cdot 10^0$	0	0
0.600	76.4	$5.1 \cdot 10^0$	0.0	$1.6 \cdot 10^{-1}$	3.5	$1.0 \cdot 10^0$	0	0
0.625	77.4	$5.7 \cdot 10^0$	0.0	$1.3 \cdot 10^{-2}$	3.6	$1.1 \cdot 10^0$	0	0
0.650	78.4	$6.1 \cdot 10^0$	0	0	3.5	$1.0 \cdot 10^0$	0	0
0.675	80.5	$6.6 \cdot 10^0$	0.0	$1.1 \cdot 10^{-1}$	3.6	$9.8 \cdot 10^{-1}$	0	0
0.700	82.7	$5.4 \cdot 10^0$	0.0	$1.1 \cdot 10^{-1}$	3.5	$7.7 \cdot 10^{-1}$	0	0
0.725	82.9	$4.7 \cdot 10^0$	0.0	$1.0 \cdot 10^{-1}$	3.5	$8.3 \cdot 10^{-1}$	0	0
0.750	82.6	$3.7 \cdot 10^0$	0.1	$2.1 \cdot 10^{-1}$	3.8	$9.0 \cdot 10^{-1}$	0	0
0.775	82.1	$3.1 \cdot 10^0$	0.0	$1.7 \cdot 10^{-1}$	4.0	$9.4 \cdot 10^{-1}$	0	0
0.800	82.6	$2.9 \cdot 10^0$	0.1	$1.9 \cdot 10^{-1}$	4.0	$8.2 \cdot 10^{-1}$	0	0
0.825	82.2	$3.0 \cdot 10^0$	0.1	$2.0 \cdot 10^{-1}$	4.0	$9.6 \cdot 10^{-1}$	0	0
0.850	78.9	$3.7 \cdot 10^0$	0.2	$1.9 \cdot 10^{-1}$	4.0	$9.3 \cdot 10^{-1}$	0	0

TABLE 6.5: Results of the dense 3D toy model database contaminated with positive, random error of maximum magnitude 0.5 in a circular cylinder spanning the z -direction for varying diameter d . Filtering was performed using MSE automatic mode selection.

For column headers refer to Table 5.1.

that although it identifies many clean points as erroneous ones, it reconstructs them well.

CFD database

So far the filtering methods using manual and MSE automatic mode selection methods give very similar results. The case when the CFD dense database is contaminated with random, positive error of 0.5 maximum magnitude is different and can be clearly seen in Table 5.6 and 6.6 for manual and MSE automatic method respectively. Whereas for the manual mode selection method convergence limit was achieved for values of b parameter from 0.300 to 0.475, the MSE automatic method does not manage to reach it at all. When the percentage of CDEP is analysed (see Figure 6.6-left), the MSE is always below the performance of the manual method. In fact, it is above 90% twice for b equal to 0.175

FIGURE 6.5: Complement of Table 6.5. Manual and MSE automatic mode selection is shown in black and red lines respectively. Left: $\%CDEP_m$ with the corresponding $\%CDEP_{std}$ bars (thick line) and $\%IDEP_m$ with the corresponding $\%IDEP_{std}$ bars (thin line). Right: mean MaxE and RMSE of the reconstructed part (solid and dashed line respectively), as defined in Equation (2.18).

and 0.200. The situation does not improve when the reconstruction errors in Figure 6.6-left are analysed. The filtering method using MSE automatic mode selection commits an error that is always higher than the manual method. The reason behind those two phenomena are unknown at the time of writing this Thesis and should be investigated. Nevertheless, considering the fact that the MSE mode selection method uses the same parameters for both the toy model and CFD databases, regardless of their size, its performance can be still considered good.

FIGURE 6.6: Complement of Table 6.6. Manual and MSE automatic mode selection is shown in black and red lines respectively. Left: $\%CDEP_m$ with the corresponding $\%CDEP_{std}$ bars (thick line) and $\%IDEP_m$ with the corresponding $\%IDEP_{std}$ bars (thin line). Right: mean MaxE and RMSE of the reconstructed part (solid and dashed line respectively), as defined in Equation (2.18).

b	%CD _{m}	%CD _{std}	%ID _{m}	%ID _{std}	k_m	k_{std}	%C ₁	%C ₂
0.125	85.3	$2.6 \cdot 10^1$	7.9	$1.1 \cdot 10^1$	4.8	$2.1 \cdot 10^0$	0	74
0.150	86.9	$1.4 \cdot 10^1$	5.4	$8.6 \cdot 10^0$	4.5	$2.3 \cdot 10^0$	0	55
0.175	90.7	$1.5 \cdot 10^1$	6.7	$8.9 \cdot 10^0$	5.1	$2.2 \cdot 10^0$	0	71
0.200	91.7	$1.6 \cdot 10^1$	6.8	$1.1 \cdot 10^1$	5.1	$2.0 \cdot 10^0$	0	79
0.225	89.4	$1.7 \cdot 10^1$	7.3	$1.2 \cdot 10^1$	4.9	$2.2 \cdot 10^0$	0	71
0.250	89.2	$2.0 \cdot 10^1$	6.3	$10.0 \cdot 10^0$	4.7	$2.0 \cdot 10^0$	0	76
0.275	89.2	$2.0 \cdot 10^1$	6.3	$10.0 \cdot 10^0$	4.7	$2.0 \cdot 10^0$	0	76
0.300	86.2	$2.6 \cdot 10^1$	6.5	$1.0 \cdot 10^1$	4.8	$2.0 \cdot 10^0$	0	71
0.325	86.2	$2.6 \cdot 10^1$	6.5	$1.0 \cdot 10^1$	4.8	$2.0 \cdot 10^0$	0	71
0.350	85.8	$1.9 \cdot 10^1$	4.7	$6.4 \cdot 10^0$	5.7	$2.1 \cdot 10^0$	0	64
0.375	83.1	$2.2 \cdot 10^1$	3.4	$5.1 \cdot 10^0$	5.5	$2.1 \cdot 10^0$	0	55
0.400	86.8	$1.6 \cdot 10^1$	3.1	$5.0 \cdot 10^0$	5.8	$1.8 \cdot 10^0$	0	60
0.425	77.1	$2.1 \cdot 10^1$	0.8	$2.2 \cdot 10^0$	5.2	$1.8 \cdot 10^0$	0	28
0.450	73.7	$2.0 \cdot 10^1$	0.4	$8.3 \cdot 10^{-1}$	5.1	$1.7 \cdot 10^0$	0	23
0.475	78.2	$1.4 \cdot 10^1$	0.4	$8.7 \cdot 10^{-1}$	5.5	$1.6 \cdot 10^0$	0	21
0.500	73.2	$1.3 \cdot 10^1$	0.2	$5.5 \cdot 10^{-1}$	4.9	$1.4 \cdot 10^0$	0	9
0.525	73.9	$1.7 \cdot 10^1$	0.5	$1.9 \cdot 10^0$	4.8	$1.7 \cdot 10^0$	0	14
0.550	73.2	$1.5 \cdot 10^1$	0.1	$4.9 \cdot 10^{-1}$	5.1	$1.6 \cdot 10^0$	0	10
0.575	73.7	$1.7 \cdot 10^1$	0.1	$2.3 \cdot 10^{-1}$	5.2	$1.7 \cdot 10^0$	0	70
0.600	71.0	$2.0 \cdot 10^1$	0.2	$7.7 \cdot 10^{-1}$	5.1	$1.8 \cdot 10^0$	0	12
0.625	75.3	$1.7 \cdot 10^1$	0.4	$1.2 \cdot 10^0$	5.6	$1.8 \cdot 10^0$	0	11

TABLE 6.6: Results of the dense CFD database contaminated with positive, random error of maximum magnitude 0.5 in a elliptical cylinder spanning the z-direction for varying parameter b . Filtering was performed using MSE automatic mode selection. For column headers refer to Table 5.1.

The counterpart for the coarse database of Table 6.6 and Figure 6.6 is shown in Table C.6 and Figure C.6 respectively. In this case the automatic MSE mode selection method performs. As in the case of manual mode selection method, the error filtering method does not perform well for the coarse CFD database with random, positive error concentrated in an elliptical cylinder. The method functioning is erratic, both for the percentage of CDEP and IDEP, which is indicated by large standard deviations. The reconstruction errors are also high. Consequently, the probability of both absolute and relative convergence are virtually zero.

Chapter 7

Conclusions and Future Work

An error filtering method based on SVD-like tools has been developed that is able to identify and correct $O(1)$ erroneous data in multidimensional databases. The method does not require any knowledge of the nature of the error and is fairly general. The only requirement is that the errors only affect a part of the database. In other words, a significant part of the database must be clean or almost clean. This occurs in several applications of industrial interest.

For the sake of clarity, the method has been first developed for two-dimensional databases, using a convenient triple iteration of truncated SVD, inspired by the seminal gappy-POD method due to Sirovich. The performance of method, called the cleaning-SVD method, was illustrated in a two-dimensional toy model database contaminated with various kinds (i.e., spread and concentrated, random and systematic) of $O(1)$ errors. The application to these contaminated databases (see Figures 3.5 and 3.8) showed that the method identifies the erroneous positions (and reconstructs the data) in a very natural way, beginning with those positions exhibiting the largest errors and then proceeding with smaller and smaller contaminated positions. This is the expected performance, taking into account the underlying ideas, namely at each iteration of the method, the erroneous points are identified as those points where the difference between the actual database values and the gappy-SVD reconstruction is largest.

After consolidating the two-dimensional version of the method, the extension to three-dimensional databases relied on the same basic ideas as in the two-dimensional case, except for:

- Replacing standard SVD by HOSVD everywhere.
- Using some additional ingredients that were necessary because HOSVD is not a straightforward extension of SVD. This is due to well known subtleties associated with the definition of the rank of a higher than three order tensor.

The method was called the cleaning-HOSVD method and illustrated in a three-dimensional toy model database. The method was also applied to an aerodynamic database (dense and coarse) prepared by analysing a wing in a steady, turbulent flow using the CFD solver OpenFOAM. Again, the performance of

the method was seen to be consistent with its very essence, since the error positions were iteratively identified and corrected, beginning with those exhibiting the largest errors and proceeding with positions showing smaller errors, which were obviously more difficult to identify; see Figures 4.6 and 4.6.

On the other hand, it should be stressed that HOSVD is able to capture redundancies along all database dimensions and the performance of the method was better for three-dimensional than for two-dimensional databases. Similarly, the straightforward extension to N -dimensional databases, with $N > 3$, is able to capture more and more redundancies as N increases, which is very promising as a means to strongly alleviate the curse of dimensionality [54].

Finally, in order to prove the tendencies described in the application of the method to both toy and aerodynamic model and to expand the analysis, the cleaning-HOSVD method was applied to both of the models (3D type only) of both types mesh density while varying either the degree of error distribution or the error maximum magnitude. In the case of randomly distributed errors, the error percentage concentration was varied while for the cylindrical or the elliptic base, the parameter defining the size was varied. For the error magnitude, the maximum error magnitude and the multiplication factor was varied for positive random and systematic errors respectively. This allowed to define the limits of performance of the cleaning-HOSVD, both in terms of gappyness degree and error magnitude. It has been found that the error concentration used in Section 3.4.1 and 4.2 were chosen in such a way that they just reached the limiting values of the method's correct functioning. Any increase in the error concentration would cause the method to diverge. On the other hand the error maximum magnitude has shown that the cleaning-HOSVD method is capable of cleaning relative errors smaller than 10^{-1} . In the case of the dense toy model database the relative errors can be as small as 10^{-3} and the method is still able to perform well. For the CFD dense database the error maximum magnitude has to be higher than or equal to 10^{-2} .

The cleaning-SVD and cleaning-HOSVD methods developed in this Thesis were designed with some tunable parameters. As anticipated in Chapter 1, the methods are quite robust, namely they are fairly insensitive to the values of the tunable parameters. This has been stressed by choosing the tunable parameter values as common to each application type, namely a common set of parameter values for all applications to the toy model databases and a common set for all applications to the aerodynamic databases. Nevertheless, the necessity of choosing a set of tunable parameters *a priori* makes the method less robust and versatile, especially the number of absolute minimum and maximum modes that can be retained. Those values depend on the nature of the database and their tuning is not straightforward. In order to overcome this limitation, a MSE automatic mode selection has been devised to first analyse the erroneous database and adapt to it the retained modes range. The method is still in its development phase but the results described in Chapter 6 are very promising and offer possibilities to further increase the robustness of the cleaning-SVD and cleaning-HOSVD methods. For instance in all the cases of the dense 3D toy

model and CFD databases the MSE automatic mode selection has performed similarly or better than the manual selection. In the case of the coarse mesh, the application to the toy model database resulted less favourable but in the case of the CFD database both mode selection methods performed evenly. However, it has to be stressed that the MSE automatic mode selection makes the cleaning-HOSVD method dependent only on the FF parameter, which describes how aggressively should the method look for errors. This improvement means that the method can be applied to databases without any prior knowledge about neither their characterising nor the type and magnitude of error distribution, having only to adjust the FF parameter.

The applications of the method have been made keeping in mind experimental databases. However, treating actual experimental databases and testing the method performance on them is beyond the scope of this thesis and left as the object of future research.

Despite the fact that the cleaning-SVD and cleaning-HOSVD methods are in a very mature development stage, there is room for further improvement.

- It is interesting to implement a modification that would reconstruct the database retaining at the end of filtering process with a high percentage of modes (e.g 99%). This number is to be quantified but assuming that the method has cleaned all the errors, increasing the number of retained modes would decrease reconstruction errors.
- Instead of using artificial errors, it would be wise to test the CFD database with experimental errors. For instance, simulating badly placed static pressure tappings (not perpendicular to the surface, etc.), similar to an investigation done by R. Shaw [55].
- An interesting but laborious test would be to run the method for both gapyness level and error magnitude, i.e. checking all the possible combinations. This would show whether gapyness level and error magnitude are mutually independent or not if it comes for influencing the performance of the method.
- Chapter 6 proposed a method of reducing some parameters, namely the number of retained modes N^{min} and N^{max} , that need to be set *a priori* in order for the filtering method to function properly. Nevertheless, it has not addressed a possibility of automatically choosing a correct value of the FF parameter. Therefore, it is reasonable to first analyse the way this parameter influences the performance of the filtering method, analogously to Chapter 5, and then propose an algorithm to automatically choose it. One way would be to implement a dynamically assigned FF value so that the method could change the FF value as it progresses, i.e. decrease it as more and more errors are identified and repaired and hence detect smaller and smaller errors.
- Finally, it would be worthwhile to check the method with other types of databases. For instance the cleaning-HOSVD should be tried on databases

exhibiting steep gradients such as the ones found in transonic wing tests. In general, the underlying mathematical methods developed by Sirovich are known to perform poorly in such conditions, especially experimental ones, as described in [12, 16]. Certain improvement can be made to those underlying method however such modifications are beyond the scope of this Thesis and its line of research. Nevertheless, applying the cleaning method to new databases, especially experimental ones and not necessary of aeronautical origin, would allow to check the functioning of the MSE automatic mode selection, validating its applicability to any type of database contaminated with unknown errors.

Finally, the cleaning-SVD or -HOSVD principle can be adjusted to identify the critical points in the database which are most valuable when it comes to database reconstruction. Therefore, a sampling method can be designed that would allow, for instance, to minimise the number of experimental runs needed to reconstruct the complete domain by applying the methodology presented in [35, 36]. This method would iteratively analyse the database showing the highest errors between two consecutive reconstructions and thus identifying them as the crucial points. The development of such method is beyond the scope of this Thesis. Nevertheless, its application to real engineering problems is evident and makes it worth developing.

Appendix A

Chapter 5, Section 5.1 coarse databases tables and figures

%cont	%CD _m	%CD _{std}	%ID _m	%ID _{std}	k _m	k _{std}	%C ₁	%C ₂
32.5	100.0	$8.5 \cdot 10^{-2}$	0.0	$7.3 \cdot 10^{-2}$	8.4	$7.1 \cdot 10^{-1}$	46	100
35.0	99.9	$4.9 \cdot 10^{-1}$	0.1	$7.9 \cdot 10^{-2}$	8.7	$9.5 \cdot 10^{-1}$	44	99
37.5	99.8	$7.7 \cdot 10^{-1}$	0.1	$1.3 \cdot 10^{-1}$	8.6	$1.2 \cdot 10^0$	41	100
40.0	99.8	$8.6 \cdot 10^{-1}$	0.1	$1.5 \cdot 10^{-1}$	8.7	$1.1 \cdot 10^0$	40	100
42.5	99.5	$1.2 \cdot 10^0$	0.1	$1.9 \cdot 10^{-1}$	8.4	$1.5 \cdot 10^0$	29	95
45.0	99.3	$1.5 \cdot 10^0$	0.1	$2.2 \cdot 10^{-1}$	8.5	$1.9 \cdot 10^0$	25	92
47.5	99.0	$1.7 \cdot 10^0$	0.2	$2.0 \cdot 10^{-1}$	8.2	$2.0 \cdot 10^0$	15	92
50.0	98.8	$1.7 \cdot 10^0$	0.2	$3.9 \cdot 10^{-1}$	8.3	$2.1 \cdot 10^0$	9	91
52.5	98.3	$2.1 \cdot 10^0$	0.3	$3.0 \cdot 10^{-1}$	8.2	$2.3 \cdot 10^0$	1	88
55.0	95.9	$4.7 \cdot 10^0$	0.5	$4.3 \cdot 10^{-1}$	7.2	$2.4 \cdot 10^0$	1	67
57.5	94.1	$4.8 \cdot 10^0$	0.8	$6.2 \cdot 10^{-1}$	6.7	$2.1 \cdot 10^0$	0	54
60.0	90.1	$7.4 \cdot 10^0$	1.4	$7.5 \cdot 10^{-1}$	6.0	$1.7 \cdot 10^0$	0	28
62.5	86.5	$6.7 \cdot 10^0$	2.5	$1.2 \cdot 10^0$	5.5	$1.3 \cdot 10^0$	0	2
65.0	82.8	$7.5 \cdot 10^0$	3.6	$1.3 \cdot 10^0$	5.4	$1.4 \cdot 10^0$	0	1
67.5	79.8	$7.8 \cdot 10^0$	5.1	$1.6 \cdot 10^0$	5.2	$1.3 \cdot 10^0$	0	0
70.0	76.3	$8.4 \cdot 10^0$	6.1	$1.4 \cdot 10^0$	4.8	$1.2 \cdot 10^0$	0	0
72.5	75.9	$6.1 \cdot 10^0$	7.3	$1.6 \cdot 10^0$	4.8	$9.3 \cdot 10^{-1}$	0	0
75.0	72.7	$7.1 \cdot 10^0$	8.7	$1.8 \cdot 10^0$	4.7	$1.1 \cdot 10^0$	0	0
77.5	68.0	$6.1 \cdot 10^0$	10.1	$1.9 \cdot 10^0$	4.6	$1.0 \cdot 10^0$	0	0
80.0	64.4	$4.8 \cdot 10^0$	11.1	$1.5 \cdot 10^0$	4.5	$8.5 \cdot 10^{-1}$	0	0
82.5	62.3	$5.2 \cdot 10^0$	11.4	$1.4 \cdot 10^0$	4.4	$1.1 \cdot 10^0$	0	0
85.0	59.9	$5.1 \cdot 10^0$	10.8	$1.0 \cdot 10^0$	4.2	$9.6 \cdot 10^{-1}$	0	0
87.5	59.6	$4.3 \cdot 10^0$	9.9	$6.8 \cdot 10^{-1}$	4.0	$9.2 \cdot 10^{-1}$	0	0
90.0	57.4	$5.0 \cdot 10^0$	8.2	$6.1 \cdot 10^{-1}$	3.9	$8.6 \cdot 10^{-1}$	0	0

TABLE A.1: Counterpart of Table 5.1 for the 3D toy model coarse database contaminated with positive, random error at random database positions.

FIGURE A.1: Counterpart of Figure 5.1 for the 3D toy model coarse database contaminated with positive, random errors at random database positions.

%cont	%CD _m	%CD _{std}	%ID _m	%ID _{std}	k _m	k _{std}	%C ₁	%C ₂
5.0	98.5	$1.7 \cdot 10^0$	8.6	$8.6 \cdot 10^0$	5.9	$1.2 \cdot 10^0$	0	100
7.5	98.4	$1.4 \cdot 10^0$	6.0	$5.0 \cdot 10^0$	6.2	$1.2 \cdot 10^0$	0	99
10.0	98.3	$1.2 \cdot 10^0$	4.3	$2.8 \cdot 10^0$	6.4	$1.2 \cdot 10^0$	0	100
12.5	98.3	$1.1 \cdot 10^0$	3.3	$2.1 \cdot 10^0$	6.5	$1.1 \cdot 10^0$	0	100
15.0	98.3	$1.1 \cdot 10^0$	3.7	$3.0 \cdot 10^0$	6.8	$1.2 \cdot 10^0$	0	98
17.5	97.9	$2.8 \cdot 10^0$	3.2	$3.1 \cdot 10^0$	6.7	$1.2 \cdot 10^0$	0	96
20.0	97.8	$3.0 \cdot 10^0$	4.0	$6.6 \cdot 10^0$	7.1	$1.3 \cdot 10^0$	0	94
22.5	96.4	$5.4 \cdot 10^0$	4.1	$5.7 \cdot 10^0$	7.0	$1.5 \cdot 10^0$	0	84
25.0	94.4	$7.6 \cdot 10^0$	5.9	$7.3 \cdot 10^0$	6.9	$1.8 \cdot 10^0$	0	69
27.5	91.9	$9.6 \cdot 10^0$	6.8	$7.2 \cdot 10^0$	6.6	$2.0 \cdot 10^0$	0	52
30.0	86.2	$1.0 \cdot 10^1$	10.0	$7.5 \cdot 10^0$	5.6	$2.2 \cdot 10^0$	0	27
32.5	81.5	$9.9 \cdot 10^0$	12.5	$8.1 \cdot 10^0$	4.7	$1.8 \cdot 10^0$	0	13
35.0	78.3	$9.9 \cdot 10^0$	12.8	$6.4 \cdot 10^0$	4.4	$1.6 \cdot 10^0$	0	1
37.5	74.7	$7.6 \cdot 10^0$	14.5	$6.0 \cdot 10^0$	3.8	$1.3 \cdot 10^0$	0	1
40.0	71.7	$1.1 \cdot 10^1$	14.8	$5.7 \cdot 10^0$	3.6	$1.2 \cdot 10^0$	0	0
42.5	68.9	$1.2 \cdot 10^1$	15.4	$5.8 \cdot 10^0$	3.4	$9.6 \cdot 10^{-1}$	0	0
45.0	66.5	$1.1 \cdot 10^1$	14.6	$4.9 \cdot 10^0$	3.4	$9.1 \cdot 10^{-1}$	0	0
47.5	59.9	$1.8 \cdot 10^1$	13.7	$4.5 \cdot 10^0$	3.1	$1.1 \cdot 10^0$	0	0
50.0	55.7	$1.9 \cdot 10^1$	15.1	$5.2 \cdot 10^0$	3.0	$1.0 \cdot 10^0$	0	0
52.5	53.7	$1.8 \cdot 10^1$	14.2	$4.6 \cdot 10^0$	2.9	$10.0 \cdot 10^{-1}$	0	0
55.0	49.9	$1.9 \cdot 10^1$	13.7	$4.9 \cdot 10^0$	2.8	$1.1 \cdot 10^0$	0	0
57.5	47.5	$1.9 \cdot 10^1$	12.5	$4.5 \cdot 10^0$	2.8	$1.1 \cdot 10^0$	0	0
60.0	42.2	$2.1 \cdot 10^1$	11.2	$4.5 \cdot 10^0$	2.5	$1.4 \cdot 10^0$	0	0
62.5	36.4	$2.2 \cdot 10^1$	9.7	$4.1 \cdot 10^0$	2.2	$1.2 \cdot 10^0$	0	0

TABLE A.2: Counterpart of Table 5.2 for the CFD coarse database contaminated with positive, random error at random database positions.

FIGURE A.2: Counterpart of Figure 5.2 for the CFD coarse database contaminated with positive, random errors at random database positions.

FIGURE A.3: Counterpart of Figure 5.3 for the 3D toy model coarse database contaminated with systematic error at random database positions.

FIGURE A.4: Counterpart of Figure 5.4 for the CFD coarse database contaminated with systematic error at random database positions.

%cont	%CD _m	%CD _{std}	%ID _m	%ID _{std}	k _m	k _{std}	%C ₁	%C ₂
5.0	100.0	$4.9 \cdot 10^{-2}$	1.2	$9.9 \cdot 10^{-1}$	4.2	$4.5 \cdot 10^{-1}$	0	100
7.5	100.0	$3.7 \cdot 10^{-2}$	0.7	$4.7 \cdot 10^{-1}$	4.2	$4.6 \cdot 10^{-1}$	1	100
10.0	100.0	$3.4 \cdot 10^{-2}$	0.4	$2.2 \cdot 10^{-1}$	4.2	$4.4 \cdot 10^{-1}$	1	100
12.5	100.0	$3.0 \cdot 10^{-2}$	0.3	$2.4 \cdot 10^{-1}$	4.3	$4.9 \cdot 10^{-1}$	3	100
15.0	100.0	$2.7 \cdot 10^{-2}$	0.3	$2.4 \cdot 10^{-1}$	4.3	$4.6 \cdot 10^{-1}$	9	100
17.5	100.0	$2.5 \cdot 10^{-2}$	0.2	$1.8 \cdot 10^{-1}$	4.5	$6.6 \cdot 10^{-1}$	12	100
20.0	100.0	$2.2 \cdot 10^{-2}$	0.2	$1.3 \cdot 10^{-1}$	4.6	$6.7 \cdot 10^{-1}$	8	100
22.5	100.0	$2.5 \cdot 10^{-2}$	0.2	$1.1 \cdot 10^{-1}$	4.9	$6.7 \cdot 10^{-1}$	2	99
25.0	100.0	$1.8 \cdot 10^{-2}$	0.2	$1.4 \cdot 10^{-1}$	4.9	$5.6 \cdot 10^{-1}$	3	100
27.5	100.0	$1.8 \cdot 10^{-2}$	0.3	$1.6 \cdot 10^{-1}$	5.0	$5.1 \cdot 10^{-1}$	1	100
30.0	100.0	$1.6 \cdot 10^{-2}$	0.3	$2.5 \cdot 10^{-1}$	5.2	$5.9 \cdot 10^{-1}$	0	100
32.5	100.0	$1.7 \cdot 10^{-2}$	0.5	$3.7 \cdot 10^{-1}$	5.3	$5.9 \cdot 10^{-1}$	0	100
35.0	100.0	$3.5 \cdot 10^{-2}$	0.9	$7.3 \cdot 10^{-1}$	5.4	$6.7 \cdot 10^{-1}$	0	97
37.5	100.0	$2.4 \cdot 10^{-2}$	2.2	$1.2 \cdot 10^0$	5.8	$7.9 \cdot 10^{-1}$	0	95
40.0	92.8	$2.2 \cdot 10^1$	4.1	$1.9 \cdot 10^0$	5.2	$1.4 \cdot 10^0$	0	72
42.5	88.7	$2.5 \cdot 10^1$	7.5	$3.6 \cdot 10^0$	5.1	$1.7 \cdot 10^0$	0	42
45.0	76.7	$3.1 \cdot 10^1$	11.9	$6.1 \cdot 10^0$	4.6	$2.0 \cdot 10^0$	0	11
47.5	62.8	$3.1 \cdot 10^1$	17.3	$1.0 \cdot 10^1$	4.2	$2.1 \cdot 10^0$	0	0
50.0	53.0	$2.8 \cdot 10^1$	23.5	$1.4 \cdot 10^1$	4.1	$2.2 \cdot 10^0$	0	0
52.5	45.1	$2.2 \cdot 10^1$	29.8	$1.6 \cdot 10^1$	4.3	$2.1 \cdot 10^0$	0	0
55.0	36.1	$1.7 \cdot 10^1$	34.8	$1.8 \cdot 10^1$	4.3	$2.1 \cdot 10^0$	0	0
57.5	28.7	$1.2 \cdot 10^1$	41.8	$1.9 \cdot 10^1$	4.7	$2.2 \cdot 10^0$	0	0
60.0	19.9	$6.8 \cdot 10^0$	46.4	$1.5 \cdot 10^1$	4.4	$1.6 \cdot 10^0$	0	0
62.5	12.7	$4.3 \cdot 10^0$	49.9	$1.2 \cdot 10^1$	4.7	$1.5 \cdot 10^0$	0	0
65.0	7.3	$3.2 \cdot 10^0$	50.0	$7.7 \cdot 10^0$	4.8	$1.4 \cdot 10^0$	0	0
67.5	4.3	$2.5 \cdot 10^0$	47.2	$7.8 \cdot 10^{-1}$	4.7	$1.1 \cdot 10^0$	0	0
70.0	2.4	$2.0 \cdot 10^0$	42.3	$5.3 \cdot 10^{-1}$	4.4	$8.6 \cdot 10^{-1}$	0	0
72.5	1.1	$1.4 \cdot 10^0$	37.7	$3.1 \cdot 10^{-1}$	4.4	$7.7 \cdot 10^{-1}$	0	0
75.0	0.3	$7.3 \cdot 10^{-1}$	33.3	$1.7 \cdot 10^{-1}$	4.6	$7.0 \cdot 10^{-1}$	0	0
77.5	0.1	$3.2 \cdot 10^{-1}$	29.0	$8.5 \cdot 10^{-2}$	4.5	$6.4 \cdot 10^{-1}$	0	0
80.0	0.0	$3.0 \cdot 10^{-2}$	25.0	$6.5 \cdot 10^{-3}$	4.4	$6.2 \cdot 10^{-1}$	0	0
82.5	0.0	$2.8 \cdot 10^{-2}$	21.2	$6.0 \cdot 10^{-3}$	4.4	$5.2 \cdot 10^{-1}$	0	0
85.0	0.0	$2.7 \cdot 10^{-2}$	17.6	$5.4 \cdot 10^{-3}$	4.3	$5.7 \cdot 10^{-1}$	0	0
87.5	0.0	$2.7 \cdot 10^{-2}$	14.3	$5.0 \cdot 10^{-3}$	4.3	$4.8 \cdot 10^{-1}$	0	0
90.0	0.0	$3.0 \cdot 10^{-2}$	11.1	$4.7 \cdot 10^{-3}$	4.2	$3.9 \cdot 10^{-1}$	0	0

TABLE A.3: Counterpart of Table 5.3 for the 3D toy model coarse database contaminated with systematic error at random database positions.

%cont	%CD _m	%CD _{std}	%ID _m	%ID _{std}	k _m	k _{std}	%C ₁	%C ₂
5.0	99.9	$3.2 \cdot 10^{-1}$	7.9	$7.3 \cdot 10^0$	4.8	$1.0 \cdot 10^0$	0	98
7.5	99.9	$2.4 \cdot 10^{-1}$	4.4	$3.9 \cdot 10^0$	4.8	$1.0 \cdot 10^0$	0	100
10.0	99.9	$3.2 \cdot 10^{-1}$	3.2	$2.9 \cdot 10^0$	5.1	$1.1 \cdot 10^0$	1	99
12.5	99.9	$2.4 \cdot 10^{-1}$	2.6	$2.6 \cdot 10^0$	5.2	$1.0 \cdot 10^0$	0	99
15.0	99.7	$1.0 \cdot 10^0$	2.7	$7.6 \cdot 10^0$	5.3	$1.1 \cdot 10^0$	0	97
17.5	99.8	$3.0 \cdot 10^{-1}$	1.7	$1.6 \cdot 10^0$	5.3	$1.2 \cdot 10^0$	2	98
20.0	99.8	$5.0 \cdot 10^{-1}$	2.1	$4.7 \cdot 10^0$	5.5	$9.1 \cdot 10^{-1}$	1	97
22.5	99.4	$1.4 \cdot 10^0$	4.2	$8.8 \cdot 10^0$	5.6	$1.4 \cdot 10^0$	0	87
25.0	98.6	$2.2 \cdot 10^0$	8.9	$1.2 \cdot 10^1$	5.2	$1.5 \cdot 10^0$	1	60
27.5	96.1	$7.0 \cdot 10^0$	18.9	$1.1 \cdot 10^1$	4.6	$1.7 \cdot 10^0$	0	19
30.0	95.6	$2.8 \cdot 10^0$	20.9	$7.8 \cdot 10^0$	3.9	$1.3 \cdot 10^0$	0	6
32.5	92.4	$1.2 \cdot 10^1$	20.4	$6.2 \cdot 10^0$	3.5	$1.2 \cdot 10^0$	0	0
35.0	90.8	$1.3 \cdot 10^1$	20.3	$5.1 \cdot 10^0$	3.4	$1.2 \cdot 10^0$	0	0
37.5	86.5	$1.6 \cdot 10^1$	20.1	$6.2 \cdot 10^0$	3.0	$1.1 \cdot 10^0$	0	0
40.0	85.9	$1.2 \cdot 10^1$	20.8	$6.2 \cdot 10^0$	3.0	$9.5 \cdot 10^{-1}$	0	0
42.5	80.1	$1.5 \cdot 10^1$	23.3	$7.8 \cdot 10^0$	3.0	$1.1 \cdot 10^0$	0	0
45.0	73.2	$1.3 \cdot 10^1$	25.3	$7.0 \cdot 10^0$	2.8	$8.0 \cdot 10^{-1}$	0	0
47.5	60.1	$1.7 \cdot 10^1$	25.9	$9.5 \cdot 10^0$	2.5	$9.9 \cdot 10^{-1}$	0	0
50.0	53.8	$1.6 \cdot 10^1$	27.9	$1.0 \cdot 10^1$	2.7	$1.2 \cdot 10^0$	0	0
52.5	45.4	$1.4 \cdot 10^1$	27.0	$1.1 \cdot 10^1$	2.4	$1.0 \cdot 10^0$	0	0
55.0	39.3	$1.3 \cdot 10^1$	26.1	$1.2 \cdot 10^1$	2.2	$9.0 \cdot 10^{-1}$	0	0
57.5	34.8	$1.3 \cdot 10^1$	26.7	$1.3 \cdot 10^1$	2.3	$1.2 \cdot 10^0$	0	0
60.0	31.8	$1.3 \cdot 10^1$	26.5	$1.3 \cdot 10^1$	2.4	$1.2 \cdot 10^0$	0	0
62.5	25.6	$1.2 \cdot 10^1$	22.6	$1.4 \cdot 10^1$	2.1	$1.1 \cdot 10^0$	0	0

TABLE A.4: Counterpart of Table 5.4 for the the CFD coarse database contaminated with systematic error at random database positions.

FIGURE A.5: Counterpart of Figure 5.5 for the dense 3D toy model database contaminated with positive, random error of maximum magnitude 0.5 at concentrated database points.

d	%CD _m	%CD _{std}	%ID _m	%ID _{std}	k _m	k _{std}	%C ₁	%C ₂
0.075	94.7	$1.5 \cdot 10^1$	2.9	$2.9 \cdot 10^0$	6.0	$1.6 \cdot 10^0$	0	86
0.100	94.7	$1.5 \cdot 10^1$	2.9	$2.9 \cdot 10^0$	6.0	$1.6 \cdot 10^0$	0	86
0.125	94.7	$1.5 \cdot 10^1$	2.9	$2.9 \cdot 10^0$	6.0	$1.6 \cdot 10^0$	0	86
0.150	94.7	$1.5 \cdot 10^1$	2.9	$2.9 \cdot 10^0$	6.0	$1.6 \cdot 10^0$	0	86
0.175	93.7	$1.9 \cdot 10^1$	0.8	$7.0 \cdot 10^{-1}$	6.8	$2.0 \cdot 10^0$	0	84
0.200	93.7	$1.9 \cdot 10^1$	0.8	$7.0 \cdot 10^{-1}$	6.8	$2.0 \cdot 10^0$	0	84
0.225	96.2	$8.9 \cdot 10^0$	0.5	$6.9 \cdot 10^{-1}$	6.7	$1.8 \cdot 10^0$	27	80
0.250	96.2	$8.9 \cdot 10^0$	0.5	$6.9 \cdot 10^{-1}$	6.7	$1.8 \cdot 10^0$	27	80
0.275	99.3	$4.3 \cdot 10^0$	0.3	$3.1 \cdot 10^{-1}$	7.5	$1.6 \cdot 10^0$	39	97
0.300	99.3	$4.3 \cdot 10^0$	0.3	$3.1 \cdot 10^{-1}$	7.5	$1.6 \cdot 10^0$	39	97
0.325	97.3	$8.3 \cdot 10^0$	0.3	$2.1 \cdot 10^{-1}$	6.9	$1.6 \cdot 10^0$	1	89
0.350	97.3	$8.3 \cdot 10^0$	0.3	$2.1 \cdot 10^{-1}$	6.9	$1.6 \cdot 10^0$	1	89
0.375	91.8	$1.3 \cdot 10^1$	0.2	$1.0 \cdot 10^0$	6.0	$1.9 \cdot 10^0$	42	68
0.400	91.8	$1.3 \cdot 10^1$	0.2	$1.0 \cdot 10^0$	6.0	$1.9 \cdot 10^0$	42	68
0.425	78.1	$9.8 \cdot 10^0$	0.5	$2.0 \cdot 10^0$	4.0	$1.4 \cdot 10^0$	4	9
0.450	78.1	$9.8 \cdot 10^0$	0.5	$2.0 \cdot 10^0$	4.0	$1.4 \cdot 10^0$	4	9
0.475	72.8	$6.3 \cdot 10^0$	1.9	$5.4 \cdot 10^0$	3.2	$1.1 \cdot 10^0$	0	0
0.500	74.3	$5.7 \cdot 10^0$	2.1	$4.2 \cdot 10^0$	3.4	$8.9 \cdot 10^{-1}$	0	0
0.525	74.1	$8.1 \cdot 10^0$	1.9	$4.1 \cdot 10^0$	3.4	$1.1 \cdot 10^0$	0	0
0.550	74.2	$5.6 \cdot 10^0$	2.1	$3.9 \cdot 10^0$	3.5	$1.0 \cdot 10^0$	0	0
0.575	74.2	$5.6 \cdot 10^0$	2.1	$3.9 \cdot 10^0$	3.5	$1.0 \cdot 10^0$	0	0
0.600	74.9	$6.4 \cdot 10^0$	2.6	$4.1 \cdot 10^0$	3.5	$9.0 \cdot 10^{-1}$	0	0
0.625	76.6	$7.2 \cdot 10^0$	1.2	$2.1 \cdot 10^0$	3.6	$9.3 \cdot 10^{-1}$	0	0
0.650	76.5	$6.6 \cdot 10^0$	1.3	$2.5 \cdot 10^0$	3.7	$9.9 \cdot 10^{-1}$	0	0
0.675	76.5	$6.0 \cdot 10^0$	1.0	$2.4 \cdot 10^0$	3.7	$8.7 \cdot 10^{-1}$	0	0
0.700	75.6	$6.1 \cdot 10^0$	1.5	$2.2 \cdot 10^0$	4.2	$9.6 \cdot 10^{-1}$	0	0
0.725	73.5	$7.1 \cdot 10^0$	2.0	$2.3 \cdot 10^0$	4.2	$1.2 \cdot 10^0$	0	0
0.750	72.6	$6.4 \cdot 10^0$	1.5	$2.0 \cdot 10^0$	4.4	$1.2 \cdot 10^0$	0	0
0.775	72.6	$6.4 \cdot 10^0$	1.5	$2.0 \cdot 10^0$	4.4	$1.2 \cdot 10^0$	0	0
0.800	67.6	$8.1 \cdot 10^0$	1.7	$2.1 \cdot 10^0$	4.3	$1.1 \cdot 10^0$	0	0
0.825	69.5	$8.3 \cdot 10^0$	2.4	$2.6 \cdot 10^0$	4.3	$1.1 \cdot 10^0$	0	0
0.850	68.4	$6.7 \cdot 10^0$	3.1	$2.3 \cdot 10^0$	4.3	$9.5 \cdot 10^{-1}$	0	0

TABLE A.5: Counterpart of Table 5.5 for the 3D toy model coarse database contaminated with positive, random error of maximum magnitude 0.5 in a circular cylinder spanning the z -direction for varying diameter d .

b	$\%CD_m$	$\%CD_{std}$	$\%ID_m$	$\%ID_{std}$	k_m	k_{std}	$\%C_1$	$\%C_2$
0.050	96.3	$7.0 \cdot 10^0$	36.5	$5.4 \cdot 10^1$	5.9	$1.8 \cdot 10^0$	0	38
0.075	96.3	$7.0 \cdot 10^0$	36.5	$5.4 \cdot 10^1$	5.9	$1.8 \cdot 10^0$	0	38
0.100	94.5	$8.1 \cdot 10^0$	26.4	$2.9 \cdot 10^1$	5.8	$1.9 \cdot 10^0$	0	56
0.125	94.5	$8.1 \cdot 10^0$	26.4	$2.9 \cdot 10^1$	5.8	$1.9 \cdot 10^0$	0	56
0.150	94.5	$8.1 \cdot 10^0$	26.4	$2.9 \cdot 10^1$	5.8	$1.9 \cdot 10^0$	0	56
0.175	92.0	$1.1 \cdot 10^1$	41.8	$7.1 \cdot 10^1$	5.6	$1.7 \cdot 10^0$	0	55
0.200	92.0	$1.1 \cdot 10^1$	41.8	$7.1 \cdot 10^1$	5.6	$1.7 \cdot 10^0$	0	55
0.225	92.0	$1.1 \cdot 10^1$	41.8	$7.1 \cdot 10^1$	5.6	$1.7 \cdot 10^0$	0	55
0.250	92.0	$1.1 \cdot 10^1$	41.8	$7.1 \cdot 10^1$	5.6	$1.7 \cdot 10^0$	0	55
0.275	90.7	$1.2 \cdot 10^1$	39.3	$6.3 \cdot 10^1$	5.5	$1.9 \cdot 10^0$	0	50
0.300	90.7	$1.2 \cdot 10^1$	39.3	$6.3 \cdot 10^1$	5.5	$1.9 \cdot 10^0$	0	50
0.325	90.7	$1.2 \cdot 10^1$	39.3	$6.3 \cdot 10^1$	5.5	$1.9 \cdot 10^0$	0	50
0.350	85.7	$1.4 \cdot 10^1$	48.8	$6.5 \cdot 10^1$	5.6	$2.2 \cdot 10^0$	0	31
0.375	85.7	$1.4 \cdot 10^1$	48.8	$6.5 \cdot 10^1$	5.6	$2.2 \cdot 10^0$	0	31
0.400	85.7	$1.4 \cdot 10^1$	48.8	$6.5 \cdot 10^1$	5.6	$2.2 \cdot 10^0$	0	31
0.425	85.7	$1.4 \cdot 10^1$	48.8	$6.5 \cdot 10^1$	5.6	$2.2 \cdot 10^0$	0	31
0.450	85.7	$1.4 \cdot 10^1$	48.8	$6.5 \cdot 10^1$	5.6	$2.2 \cdot 10^0$	0	31
0.475	80.0	$2.0 \cdot 10^1$	44.4	$6.3 \cdot 10^1$	5.1	$2.5 \cdot 10^0$	0	27
0.500	80.0	$2.0 \cdot 10^1$	44.4	$6.3 \cdot 10^1$	5.1	$2.5 \cdot 10^0$	0	27
0.525	80.0	$2.0 \cdot 10^1$	44.4	$6.3 \cdot 10^1$	5.1	$2.5 \cdot 10^0$	0	27
0.550	80.0	$2.0 \cdot 10^1$	44.4	$6.3 \cdot 10^1$	5.1	$2.5 \cdot 10^0$	0	27
0.575	80.0	$2.0 \cdot 10^1$	44.4	$6.3 \cdot 10^1$	5.1	$2.5 \cdot 10^0$	0	27
0.600	80.0	$2.0 \cdot 10^1$	44.4	$6.3 \cdot 10^1$	5.1	$2.5 \cdot 10^0$	0	27
0.625	80.0	$2.0 \cdot 10^1$	44.4	$6.3 \cdot 10^1$	5.1	$2.5 \cdot 10^0$	0	27

TABLE A.6: Counterpart of Table 5.6 for the CFD coarse database contaminated with positive, random error at elliptical cylinder position.

FIGURE A.6: Counterpart of Figure 5.6 for the CFD coarse database contaminated with positive, random errors at elliptical cylinder position.

Appendix B

Chapter 5, Section 5.2 coarse databases tables and figures

err	%CD _m	%CD _{std}	%ID _m	%ID _{std}	k _m	k _{std}	%C ₁	%C ₂
$1.0 \cdot 10^{-4}$	0	0	0	0	1	0	0	0
$1.0 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$1.2 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$1.5 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$1.9 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$2.3 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$2.8 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$3.5 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$4.3 \cdot 10^{-3}$	0.0	$1.2 \cdot 10^{-2}$	0	0	1	0	0	0
$5.3 \cdot 10^{-3}$	1.4	$4.7 \cdot 10^{-1}$	0	0	1	0	0	0
$6.6 \cdot 10^{-3}$	99.9	$2.8 \cdot 10^{-1}$	0.1	$1.7 \cdot 10^{-1}$	8.2	$9.7 \cdot 10^{-1}$	50	100
$8.1 \cdot 10^{-3}$	100.0	$8.8 \cdot 10^{-2}$	0.1	$1.3 \cdot 10^{-1}$	8.0	$9.7 \cdot 10^{-1}$	53	100
$1.0 \cdot 10^{-2}$	100.0	$5.6 \cdot 10^{-2}$	0.1	$1.3 \cdot 10^{-1}$	7.8	$8.4 \cdot 10^{-1}$	61	100
$1.3 \cdot 10^{-2}$	100.0	$4.1 \cdot 10^{-2}$	0.1	$1.2 \cdot 10^{-1}$	7.4	$7.4 \cdot 10^{-1}$	65	100
$1.7 \cdot 10^{-2}$	100.0	$4.5 \cdot 10^{-2}$	0.1	$1.0 \cdot 10^{-1}$	7.2	$7.6 \cdot 10^{-1}$	70	100
$2.0 \cdot 10^{-2}$	100.0	$4.2 \cdot 10^{-2}$	0.0	$9.6 \cdot 10^{-2}$	7.2	$7.4 \cdot 10^{-1}$	72	100
$4.0 \cdot 10^{-2}$	100.0	$3.0 \cdot 10^{-2}$	0.1	$1.2 \cdot 10^{-1}$	7.0	$7.0 \cdot 10^{-1}$	69	100
$6.0 \cdot 10^{-2}$	100.0	$3.5 \cdot 10^{-2}$	0.1	$1.4 \cdot 10^{-1}$	7.0	$6.7 \cdot 10^{-1}$	69	100
$8.0 \cdot 10^{-2}$	100.0	$1.8 \cdot 10^{-2}$	0.1	$1.4 \cdot 10^{-1}$	7.0	$7.0 \cdot 10^{-1}$	70	100
$1.0 \cdot 10^{-1}$	100.0	$1.8 \cdot 10^{-2}$	0.1	$1.3 \cdot 10^{-1}$	7.0	$7.2 \cdot 10^{-1}$	63	100
$2.0 \cdot 10^{-1}$	100.0	$1.2 \cdot 10^{-2}$	0.1	$1.3 \cdot 10^{-1}$	7.0	$5.3 \cdot 10^{-1}$	62	100
$3.0 \cdot 10^{-1}$	100.0	$1.2 \cdot 10^{-2}$	0.1	$1.4 \cdot 10^{-1}$	7.0	$6.0 \cdot 10^{-1}$	58	100
$4.0 \cdot 10^{-1}$	100.0	$1.2 \cdot 10^{-2}$	0.1	$1.4 \cdot 10^{-1}$	7.2	$5.8 \cdot 10^{-1}$	44	100
$5.0 \cdot 10^{-1}$	100	0	0.2	$1.5 \cdot 10^{-1}$	7.3	$5.9 \cdot 10^{-1}$	34	100

TABLE B.1: Counterpart of Table 5.7 for the 3D toy model coarse database contaminated with positive, random error of varying error magnitude at 10% random database positions.

FIGURE B.1: Counterpart of Figure 5.7 for the 3D toy model coarse database contaminated with positive, random errors of varying magnitude at 10% random database positions.

err	%CD _m	%CD _{std}	%ID _m	%ID _{std}	k _m	k _{std}	%C ₁	%C ₂
$1.0 \cdot 10^{-4}$	0.9	$7.8 \cdot 10^{-1}$	7.7	$7.8 \cdot 10^{-1}$	1	0	0	0
$1.0 \cdot 10^{-3}$	1.1	$1.4 \cdot 10^0$	8.6	$4.6 \cdot 10^0$	1.2	$9.1 \cdot 10^{-1}$	0	0
$1.0 \cdot 10^{-2}$	15.2	$2.2 \cdot 10^1$	29.3	$2.8 \cdot 10^1$	2.8	$2.5 \cdot 10^0$	0	0
$2.0 \cdot 10^{-2}$	34.0	$3.3 \cdot 10^1$	35.5	$3.0 \cdot 10^1$	3.7	$2.6 \cdot 10^0$	0	0
$4.0 \cdot 10^{-2}$	30.2	$3.6 \cdot 10^1$	31.5	$3.3 \cdot 10^1$	3.0	$2.5 \cdot 10^0$	0	1
$6.0 \cdot 10^{-2}$	55.7	$4.0 \cdot 10^1$	31.2	$3.1 \cdot 10^1$	4.2	$2.4 \cdot 10^0$	0	25
$8.0 \cdot 10^{-2}$	74.3	$3.3 \cdot 10^1$	27.9	$3.0 \cdot 10^1$	5.5	$2.1 \cdot 10^0$	0	49
$1.0 \cdot 10^{-1}$	80.0	$3.1 \cdot 10^1$	24.2	$3.1 \cdot 10^1$	5.8	$1.9 \cdot 10^0$	0	71
$2.0 \cdot 10^{-1}$	96.3	$2.2 \cdot 10^0$	5.6	$5.0 \cdot 10^0$	6.0	$1.2 \cdot 10^0$	0	96
$3.0 \cdot 10^{-1}$	97.2	$3.0 \cdot 10^0$	5.6	$9.5 \cdot 10^0$	6.2	$1.2 \cdot 10^0$	0	98
$4.0 \cdot 10^{-1}$	98.0	$1.9 \cdot 10^0$	4.7	$3.4 \cdot 10^0$	6.3	$1.2 \cdot 10^0$	0	99
$5.0 \cdot 10^{-1}$	98.3	$1.2 \cdot 10^0$	4.3	$2.8 \cdot 10^0$	6.4	$1.2 \cdot 10^0$	0	100

TABLE B.2: Counterpart of Table 5.8 for the CFD coarse database contaminated with positive, random error of varying error magnitude at 10% random database positions.

FIGURE B.2: Counterpart of Figure 5.8 for the CFD coarse database contaminated with positive, random errors of varying magnitude at 10% random database positions.

err	%CD _m	%CD _{std}	%ID _m	%ID _{std}	k _m	k _{std}	%C ₁	%C ₂
$1.0 \cdot 10^{-4}$	0	0	0	0	1	0	0	0
$1.0 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$1.2 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$1.5 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$1.9 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$2.3 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$2.8 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$3.5 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$4.3 \cdot 10^{-3}$	0	0	0	0	1	0	0	0
$5.3 \cdot 10^{-3}$	0.0	$6.2 \cdot 10^{-2}$	0	0	1	0	0	0
$6.6 \cdot 10^{-3}$	0.3	$1.9 \cdot 10^{-1}$	0	0	1	0	0	0
$8.1 \cdot 10^{-3}$	2.5	$5.1 \cdot 10^{-1}$	0.0	$1.2 \cdot 10^{-2}$	1.0	0	0	0
$1.0 \cdot 10^{-2}$	65.7	$4.4 \cdot 10^1$	0.2	$2.5 \cdot 10^{-1}$	3.3	$1.9 \cdot 10^0$	7	62
$1.3 \cdot 10^{-2}$	100.0	$3.4 \cdot 10^{-2}$	0.3	$1.9 \cdot 10^{-1}$	4.5	$6.1 \cdot 10^{-1}$	14	100
$1.7 \cdot 10^{-2}$	100.0	$3.4 \cdot 10^{-2}$	0.3	$1.8 \cdot 10^{-1}$	4.3	$5.3 \cdot 10^{-1}$	12	100
$2.0 \cdot 10^{-2}$	100.0	$3.4 \cdot 10^{-2}$	0.3	$2.5 \cdot 10^{-1}$	4.3	$4.6 \cdot 10^{-1}$	11	100
$4.0 \cdot 10^{-2}$	100.0	$3.4 \cdot 10^{-2}$	0.4	$3.0 \cdot 10^{-1}$	4.3	$4.8 \cdot 10^{-1}$	3	100
$6.0 \cdot 10^{-2}$	100.0	$3.4 \cdot 10^{-2}$	0.4	$3.1 \cdot 10^{-1}$	4.3	$5.0 \cdot 10^{-1}$	3	100
$8.0 \cdot 10^{-2}$	100.0	$3.4 \cdot 10^{-2}$	0.4	$2.9 \cdot 10^{-1}$	4.3	$4.8 \cdot 10^{-1}$	2	100
$1.0 \cdot 10^{-1}$	100.0	$3.4 \cdot 10^{-2}$	0.4	$2.7 \cdot 10^{-1}$	4.3	$4.5 \cdot 10^{-1}$	3	100
$2.0 \cdot 10^{-1}$	100.0	$3.4 \cdot 10^{-2}$	0.4	$2.5 \cdot 10^{-1}$	4.2	$4.6 \cdot 10^{-1}$	3	100
$3.0 \cdot 10^{-1}$	100.0	$3.4 \cdot 10^{-2}$	0.4	$2.5 \cdot 10^{-1}$	4.2	$4.3 \cdot 10^{-1}$	2	100
$4.0 \cdot 10^{-1}$	100.0	$3.4 \cdot 10^{-2}$	0.4	$2.4 \cdot 10^{-1}$	4.2	$4.2 \cdot 10^{-1}$	2	100
$5.0 \cdot 10^{-1}$	100.0	$3.4 \cdot 10^{-2}$	0.4	$2.2 \cdot 10^{-1}$	4.2	$4.4 \cdot 10^{-1}$	1	100

TABLE B.3: Counterpart of Table 5.9 for the 3D toy model coarse database contaminated with systematic errors of varying magnitude at 10% random database positions.

FIGURE B.3: Counterpart of Figure 5.9 for the 3D toy model coarse database contaminated with systematic errors of varying magnitude at 10% random database positions.

err	%CD _m	%CD _{std}	%ID _m	%ID _{std}	k _m	k _{std}	%C ₁	%C ₂
$1.0 \cdot 10^{-4}$	0.9	$7.8 \cdot 10^{-1}$	7.7	$7.8 \cdot 10^{-1}$	1	0	0	0
$1.0 \cdot 10^{-3}$	1.0	$8.8 \cdot 10^{-1}$	7.9	$9.0 \cdot 10^{-1}$	1.1	$3.4 \cdot 10^{-1}$	0	0
$1.0 \cdot 10^{-2}$	15.9	$2.8 \cdot 10^1$	25.7	$2.8 \cdot 10^1$	2.3	$2.0 \cdot 10^0$	0	1
$2.0 \cdot 10^{-2}$	45.9	$4.5 \cdot 10^1$	31.8	$2.3 \cdot 10^1$	3.5	$2.5 \cdot 10^0$	0	24
$4.0 \cdot 10^{-2}$	32.4	$3.9 \cdot 10^1$	32.2	$3.3 \cdot 10^1$	2.4	$1.7 \cdot 10^0$	0	20
$6.0 \cdot 10^{-2}$	61.3	$4.4 \cdot 10^1$	28.4	$3.1 \cdot 10^1$	4.0	$2.3 \cdot 10^0$	0	55
$8.0 \cdot 10^{-2}$	71.9	$3.9 \cdot 10^1$	33.2	$3.7 \cdot 10^1$	4.4	$1.8 \cdot 10^0$	0	58
$1.0 \cdot 10^{-1}$	89.4	$2.8 \cdot 10^1$	15.0	$2.1 \cdot 10^1$	4.9	$1.5 \cdot 10^0$	0	85
$2.0 \cdot 10^{-1}$	99.7	$5.5 \cdot 10^{-1}$	4.3	$5.0 \cdot 10^0$	4.8	$1.0 \cdot 10^0$	1	99
$3.0 \cdot 10^{-1}$	99.8	$4.4 \cdot 10^{-1}$	3.6	$4.3 \cdot 10^0$	4.7	$1.1 \cdot 10^0$	1	99
$4.0 \cdot 10^{-1}$	99.9	$3.3 \cdot 10^{-1}$	3.3	$3.2 \cdot 10^0$	4.8	$1.0 \cdot 10^0$	1	100
$5.0 \cdot 10^{-1}$	99.9	$3.2 \cdot 10^{-1}$	3.2	$2.9 \cdot 10^0$	5.1	$1.1 \cdot 10^0$	1	99

TABLE B.4: Counterpart of Table 5.10 for the CFD coarse database contaminated with systematic error of varying error magnitude at 10% random database positions.

FIGURE B.4: Counterpart of Figure 5.10 for the CFD coarse database contaminated with systematic errors of varying magnitude at 10% random database positions.

err	%CD _m	%CD _{std}	%ID _m	%ID _{std}	k _m	k _{std}	%C ₁	%C ₂
1.0 · 10 ⁻⁴	0.0	0	0.0	0	1.0	0	0	0
1.0 · 10 ⁻³	0.0	0	0.0	0	1.0	0	0	0
1.2 · 10 ⁻³	0.0	0	0.0	0	1.0	0	0	0
1.5 · 10 ⁻³	0.0	0	0.0	0	1.0	0	0	0
1.9 · 10 ⁻³	0.0	0	0.0	0	1.0	0	0	0
2.3 · 10 ⁻³	0.0	0	0.0	0	1.0	0	0	0
2.8 · 10 ⁻³	0.0	0	0.0	0	1.0	0	0	0
3.5 · 10 ⁻³	0.0	0	0.0	0	1.0	0	0	0
4.3 · 10 ⁻³	0.0	0	0.0	0	1.0	0	0	0
5.3 · 10 ⁻³	0.0	0	0.0	0	1.0	0	0	0
6.6 · 10 ⁻³	0.2	1.3 · 10 ⁰	0.0	0	1.0	0	0	0
8.1 · 10 ⁻³	18.6	3.6 · 10 ¹	0.2	7.5 · 10 ⁻¹	1.9	2.1 · 10 ⁰	0	12
1.0 · 10 ⁻²	79.2	3.8 · 10 ¹	1.1	1.4 · 10 ⁰	4.5	2.0 · 10 ⁰	36	76
1.3 · 10 ⁻²	91.4	1.9 · 10 ¹	2.1	1.8 · 10 ⁰	5.6	1.8 · 10 ⁰	0	79
1.7 · 10 ⁻²	90.0	2.0 · 10 ¹	2.0	1.8 · 10 ⁰	5.5	1.7 · 10 ⁰	0	76
2.0 · 10 ⁻²	99.1	4.8 · 10 ⁰	2.0	1.5 · 10 ⁰	5.2	7.6 · 10 ⁻¹	27	97
4.0 · 10 ⁻²	87.6	1.8 · 10 ¹	1.4	1.6 · 10 ⁰	4.2	1.5 · 10 ⁰	19	65
6.0 · 10 ⁻²	94.7	1.4 · 10 ¹	1.5	1.5 · 10 ⁰	4.7	1.1 · 10 ⁰	32	84
8.0 · 10 ⁻²	97.2	1.1 · 10 ¹	1.7	1.5 · 10 ⁰	5.0	8.3 · 10 ⁻¹	33	92
1.0 · 10 ⁻¹	98.5	6.9 · 10 ⁰	1.8	1.6 · 10 ⁰	5.0	8.9 · 10 ⁻¹	35	94
2.0 · 10 ⁻¹	98.6	3.9 · 10 ⁰	1.6	1.9 · 10 ⁰	5.0	9.5 · 10 ⁻¹	39	92
3.0 · 10 ⁻¹	98.3	5.0 · 10 ⁰	1.5	1.8 · 10 ⁰	4.9	1.0 · 10 ⁰	43	88
4.0 · 10 ⁻¹	99.4	2.4 · 10 ⁰	2.3	2.0 · 10 ⁰	5.3	8.8 · 10 ⁻¹	31	96
5.0 · 10 ⁻¹	99.7	1.8 · 10 ⁰	3.0	2.1 · 10 ⁰	5.6	6.9 · 10 ⁻¹	18	98

TABLE B.5: Counterpart of Table 5.11 for the 3D toy model coarse database contaminated with random, positive errors of varying magnitude in a circular cylinder of 0.1 diameter.

FIGURE B.5: Counterpart of Figure 5.11 for the 3D toy model coarse database contaminated with random, positive errors of varying magnitude in a circular cylinder of 0.1 diameter

err	%CD _m	%CD _{std}	%ID _m	%ID _{std}	k _m	k _{std}	%C ₁	%C ₂
$1.0 \cdot 10^{-4}$	0.0	0	27.1	$3.3 \cdot 10^{-14}$	1.0	0	0	0
$1.0 \cdot 10^{-3}$	0.0	0	27.1	$3.3 \cdot 10^{-14}$	1.0	0	0	0
$1.0 \cdot 10^{-2}$	7.9	$1.3 \cdot 10^1$	112.2	$8.1 \cdot 10^1$	2.7	$1.5 \cdot 10^0$	0	0
$2.0 \cdot 10^{-2}$	16.9	$1.9 \cdot 10^1$	110.3	$8.6 \cdot 10^1$	2.7	$1.8 \cdot 10^0$	0	0
$4.0 \cdot 10^{-2}$	23.4	$2.9 \cdot 10^1$	80.0	$7.3 \cdot 10^1$	2.5	$2.0 \cdot 10^0$	0	1
$6.0 \cdot 10^{-2}$	28.6	$3.5 \cdot 10^1$	81.8	$7.5 \cdot 10^1$	2.4	$2.0 \cdot 10^0$	0	1
$8.0 \cdot 10^{-2}$	32.5	$3.9 \cdot 10^1$	72.1	$7.0 \cdot 10^1$	2.5	$2.1 \cdot 10^0$	0	4
$1.0 \cdot 10^{-1}$	68.7	$3.1 \cdot 10^1$	68.1	$5.0 \cdot 10^1$	4.4	$2.2 \cdot 10^0$	0	22
$2.0 \cdot 10^{-1}$	91.8	$1.3 \cdot 10^1$	40.1	$3.9 \cdot 10^1$	5.8	$1.5 \cdot 10^0$	0	47
$3.0 \cdot 10^{-1}$	94.6	$9.3 \cdot 10^0$	30.4	$2.1 \cdot 10^1$	6.1	$1.5 \cdot 10^0$	0	49
$4.0 \cdot 10^{-1}$	92.3	$1.2 \cdot 10^1$	27.4	$3.5 \cdot 10^1$	5.8	$1.9 \cdot 10^0$	0	41
$5.0 \cdot 10^{-1}$	94.5	$8.1 \cdot 10^0$	26.4	$2.9 \cdot 10^1$	5.8	$1.9 \cdot 10^0$	0	56

TABLE B.6: Counterpart of Table 5.12 for the CFD coarse database contaminated with positive, random error of varying error magnitude in an elliptic cylinder of 0.35 b parameter.

FIGURE B.6: Counterpart of Figure 5.12 for the CFD coarse database contaminated with positive, random error of varying error magnitude in an elliptic cylinder of 0.35 b parameter.

Appendix C

Chapter 6 coarse databases tables and figures

%cont	%CD _m	%CD _{std}	%ID _m	%ID _{std}	k _m	k _{std}	%C ₁	%C ₂
32.5	97.8	$4.6 \cdot 10^0$	3.0	$8.0 \cdot 10^0$	7.3	$2.4 \cdot 10^0$	0	80
35.0	97.9	$4.7 \cdot 10^0$	1.5	$4.6 \cdot 10^0$	7.5	$2.5 \cdot 10^0$	0	82
37.5	96.9	$6.6 \cdot 10^0$	1.7	$5.3 \cdot 10^0$	7.2	$2.6 \cdot 10^0$	2	81
40.0	96.8	$6.3 \cdot 10^0$	1.7	$4.8 \cdot 10^0$	7.2	$2.8 \cdot 10^0$	2	80
42.5	97.9	$4.5 \cdot 10^0$	1.2	$3.7 \cdot 10^0$	7.3	$2.3 \cdot 10^0$	0	83
45.0	96.3	$6.2 \cdot 10^0$	1.7	$4.5 \cdot 10^0$	6.9	$2.7 \cdot 10^0$	0	76
47.5	95.9	$7.4 \cdot 10^0$	1.1	$3.0 \cdot 10^0$	6.7	$2.8 \cdot 10^0$	1	77
50.0	95.5	$5.8 \cdot 10^0$	1.4	$3.4 \cdot 10^0$	6.3	$2.7 \cdot 10^0$	0	67
52.5	94.3	$6.6 \cdot 10^0$	1.9	$3.4 \cdot 10^0$	5.7	$2.3 \cdot 10^0$	0	54
55.0	92.5	$6.3 \cdot 10^0$	2.9	$4.0 \cdot 10^0$	5.2	$2.0 \cdot 10^0$	0	36
57.5	91.1	$6.1 \cdot 10^0$	3.5	$3.5 \cdot 10^0$	5.1	$1.9 \cdot 10^0$	0	26
60.0	89.4	$5.4 \cdot 10^0$	5.0	$3.3 \cdot 10^0$	4.9	$1.5 \cdot 10^0$	0	6
62.5	87.4	$4.6 \cdot 10^0$	5.4	$2.6 \cdot 10^0$	4.5	$9.9 \cdot 10^{-1}$	0	1
65.0	84.2	$5.0 \cdot 10^0$	5.8	$1.7 \cdot 10^0$	4.0	$8.8 \cdot 10^{-1}$	0	0
67.5	81.7	$4.7 \cdot 10^0$	5.8	$1.2 \cdot 10^0$	3.7	$8.0 \cdot 10^{-1}$	0	0
70.0	79.2	$6.3 \cdot 10^0$	6.0	$1.3 \cdot 10^0$	3.6	$8.2 \cdot 10^{-1}$	0	0
72.5	77.1	$5.5 \cdot 10^0$	6.5	$1.6 \cdot 10^0$	3.7	$8.2 \cdot 10^{-1}$	0	0
75.0	73.0	$6.1 \cdot 10^0$	7.1	$1.6 \cdot 10^0$	3.5	$7.8 \cdot 10^{-1}$	0	0
77.5	69.4	$5.6 \cdot 10^0$	8.6	$2.1 \cdot 10^0$	3.5	$7.6 \cdot 10^{-1}$	0	0
80.0	65.8	$5.0 \cdot 10^0$	10.1	$1.9 \cdot 10^0$	3.3	$7.0 \cdot 10^{-1}$	0	0
82.5	64.0	$4.7 \cdot 10^0$	10.8	$1.7 \cdot 10^0$	3.4	$8.3 \cdot 10^{-1}$	0	0
85.0	61.4	$5.3 \cdot 10^0$	10.3	$1.4 \cdot 10^0$	3.3	$7.1 \cdot 10^{-1}$	0	0
87.5	58.9	$4.8 \cdot 10^0$	9.5	$9.8 \cdot 10^{-1}$	3.2	$6.2 \cdot 10^{-1}$	0	0
90.0	58.2	$4.5 \cdot 10^0$	8.2	$6.1 \cdot 10^{-1}$	3.2	$6.2 \cdot 10^{-1}$	0	0

TABLE C.1: Counterpart of Table 6.1 for the 3D toy model coarse database contaminated with positive, random error at random database positions (MSE automatic mode selection method).

FIGURE C.1: Counterpart of Figure 6.1 for the 3D toy model coarse database contaminated with positive, random errors at random database positions (MSE automatic mode selection method).

%cont	%CD _m	%CD _{std}	%ID _m	%ID _{std}	k _m	k _{std}	%C ₁	%C ₂
5.0	98.1	$2.0 \cdot 10^0$	32.2	$2.6 \cdot 10^1$	6.3	$1.7 \cdot 10^0$	0	22
7.5	98.5	$1.3 \cdot 10^0$	23.7	$2.0 \cdot 10^1$	6.6	$1.7 \cdot 10^0$	0	42
10.0	98.4	$1.3 \cdot 10^0$	16.4	$1.2 \cdot 10^1$	6.6	$1.6 \cdot 10^0$	0	62
12.5	98.5	$9.8 \cdot 10^{-1}$	16.1	$1.2 \cdot 10^1$	7.3	$1.8 \cdot 10^0$	0	71
15.0	98.4	$1.5 \cdot 10^0$	13.2	$8.1 \cdot 10^0$	7.4	$1.6 \cdot 10^0$	0	84
17.5	98.5	$9.4 \cdot 10^{-1}$	10.3	$7.0 \cdot 10^0$	7.5	$1.5 \cdot 10^0$	0	93
20.0	98.4	$1.0 \cdot 10^0$	8.2	$4.7 \cdot 10^0$	7.6	$1.5 \cdot 10^0$	0	92
22.5	97.8	$2.6 \cdot 10^0$	6.6	$3.4 \cdot 10^0$	7.5	$1.6 \cdot 10^0$	0	86
25.0	97.4	$2.8 \cdot 10^0$	5.6	$3.3 \cdot 10^0$	7.4	$1.9 \cdot 10^0$	0	82
27.5	95.7	$5.5 \cdot 10^0$	4.5	$2.7 \cdot 10^0$	6.8	$1.9 \cdot 10^0$	0	73
30.0	92.4	$9.4 \cdot 10^0$	4.5	$3.7 \cdot 10^0$	6.3	$2.1 \cdot 10^0$	0	55
32.5	90.4	$8.3 \cdot 10^0$	5.3	$4.9 \cdot 10^0$	6.0	$2.0 \cdot 10^0$	0	40
35.0	85.6	$1.0 \cdot 10^1$	5.1	$3.6 \cdot 10^0$	5.2	$1.8 \cdot 10^0$	0	17
37.5	80.2	$1.0 \cdot 10^1$	6.7	$4.5 \cdot 10^0$	4.3	$1.6 \cdot 10^0$	0	7
40.0	76.7	$8.8 \cdot 10^0$	9.0	$4.8 \cdot 10^0$	4.2	$1.7 \cdot 10^0$	0	3
42.5	73.1	$9.0 \cdot 10^0$	9.9	$4.5 \cdot 10^0$	3.7	$1.2 \cdot 10^0$	0	0
45.0	71.0	$7.7 \cdot 10^0$	10.4	$3.6 \cdot 10^0$	3.5	$9.5 \cdot 10^{-1}$	0	0
47.5	69.4	$8.1 \cdot 10^0$	11.1	$3.2 \cdot 10^0$	3.4	$9.1 \cdot 10^{-1}$	0	0
50.0	68.3	$8.3 \cdot 10^0$	10.3	$2.7 \cdot 10^0$	3.3	$8.3 \cdot 10^{-1}$	0	0
52.5	66.9	$7.3 \cdot 10^0$	10.4	$2.6 \cdot 10^0$	3.2	$6.9 \cdot 10^{-1}$	0	0
55.0	64.8	$9.1 \cdot 10^0$	10.0	$2.4 \cdot 10^0$	3.4	$8.2 \cdot 10^{-1}$	0	0
57.5	63.0	$1.1 \cdot 10^1$	9.6	$2.6 \cdot 10^0$	3.3	$8.1 \cdot 10^{-1}$	0	0
60.0	59.4	$1.2 \cdot 10^1$	8.9	$2.4 \cdot 10^0$	3.1	$1.0 \cdot 10^0$	0	0
62.5	57.0	$1.4 \cdot 10^1$	9.2	$3.0 \cdot 10^0$	3.3	$1.1 \cdot 10^0$	0	0

TABLE C.2: Counterpart of Table 6.2 for the CFD coarse database contaminated with positive, random error at random database positions (MSE automatic mode selection method).

FIGURE C.2: Counterpart of Figure 6.2 for the CFD coarse database contaminated with positive, random errors at random database positions (MSE automatic mode selection method).

FIGURE C.3: Counterpart of Figure 6.3 for the 3D toy model coarse database contaminated with systematic error at random database positions (MSE automatic mode selection method).

FIGURE C.4: Counterpart of Figure 6.4 for the CFD coarse database contaminated with systematic error at random database positions (MSE automatic mode selection method).

%cont	%CD _m	%CD _{std}	%ID _m	%ID _{std}	k _m	k _{std}	%C ₁	%C ₂
5.0	99.9	$3.6 \cdot 10^{-1}$	30.1	$2.4 \cdot 10^1$	4.3	$1.6 \cdot 10^0$	0	4
7.5	99.5	$1.5 \cdot 10^0$	22.5	$2.2 \cdot 10^1$	4.3	$1.8 \cdot 10^0$	0	7
10.0	99.6	$1.4 \cdot 10^0$	12.1	$1.4 \cdot 10^1$	4.3	$1.8 \cdot 10^0$	0	5
12.5	99.4	$1.5 \cdot 10^0$	10.6	$1.5 \cdot 10^1$	4.4	$2.1 \cdot 10^0$	0	4
15.0	99.6	$1.2 \cdot 10^0$	8.0	$1.1 \cdot 10^1$	4.8	$2.1 \cdot 10^0$	0	7
17.5	99.8	$9.3 \cdot 10^{-1}$	5.2	$5.3 \cdot 10^0$	4.8	$1.9 \cdot 10^0$	0	7
20.0	99.8	$8.8 \cdot 10^{-1}$	5.4	$6.2 \cdot 10^0$	5.3	$1.9 \cdot 10^0$	0	9
22.5	99.9	$5.0 \cdot 10^{-1}$	6.9	$1.0 \cdot 10^1$	5.7	$1.7 \cdot 10^0$	0	9
25.0	99.8	$5.1 \cdot 10^{-1}$	4.6	$7.4 \cdot 10^0$	5.2	$1.8 \cdot 10^0$	0	8
27.5	99.8	$4.8 \cdot 10^{-1}$	3.5	$5.4 \cdot 10^0$	5.4	$1.7 \cdot 10^0$	0	10
30.0	99.7	$1.0 \cdot 10^0$	4.5	$9.9 \cdot 10^0$	5.3	$1.9 \cdot 10^0$	0	9
32.5	98.9	$8.4 \cdot 10^0$	3.0	$5.3 \cdot 10^0$	5.4	$2.1 \cdot 10^0$	0	4
35.0	98.0	$1.2 \cdot 10^1$	4.3	$8.4 \cdot 10^0$	5.0	$1.9 \cdot 10^0$	0	7
37.5	92.1	$2.4 \cdot 10^1$	5.6	$7.6 \cdot 10^0$	4.9	$1.9 \cdot 10^0$	0	7
40.0	88.5	$2.9 \cdot 10^1$	7.1	$7.3 \cdot 10^0$	4.5	$1.9 \cdot 10^0$	0	9
42.5	85.7	$3.1 \cdot 10^1$	9.3	$7.5 \cdot 10^0$	4.1	$1.7 \cdot 10^0$	0	7
45.0	81.9	$3.2 \cdot 10^1$	13.4	$7.6 \cdot 10^0$	4.0	$1.7 \cdot 10^0$	0	9
47.5	65.8	$3.7 \cdot 10^1$	14.5	$9.6 \cdot 10^0$	3.3	$1.7 \cdot 10^0$	0	11
50.0	57.4	$3.4 \cdot 10^1$	20.7	$1.4 \cdot 10^1$	3.2	$1.7 \cdot 10^0$	0	9
52.5	46.4	$2.8 \cdot 10^1$	25.6	$1.7 \cdot 10^1$	3.0	$1.6 \cdot 10^0$	0	10
55.0	34.6	$1.9 \cdot 10^1$	30.2	$2.0 \cdot 10^1$	3.1	$1.7 \cdot 10^0$	0	9
57.5	27.8	$1.5 \cdot 10^1$	35.5	$2.2 \cdot 10^1$	3.0	$1.6 \cdot 10^0$	0	7
60.0	22.5	$9.6 \cdot 10^0$	43.7	$2.0 \cdot 10^1$	3.5	$1.5 \cdot 10^0$	0	7
62.5	15.7	$7.1 \cdot 10^0$	45.3	$1.8 \cdot 10^1$	3.4	$1.3 \cdot 10^0$	0	9
65.0	10.1	$5.2 \cdot 10^0$	48.6	$1.0 \cdot 10^1$	4.2	$1.6 \cdot 10^0$	0	9
67.5	6.8	$5.4 \cdot 10^0$	45.8	$6.8 \cdot 10^0$	4.2	$1.5 \cdot 10^0$	0	10
70.0	4.4	$4.6 \cdot 10^0$	42.1	$1.0 \cdot 10^0$	4.3	$1.6 \cdot 10^0$	0	10
72.5	2.1	$2.1 \cdot 10^0$	37.6	$5.4 \cdot 10^{-1}$	5.2	$1.9 \cdot 10^0$	0	8
75.0	1.5	$1.6 \cdot 10^0$	33.2	$3.2 \cdot 10^{-1}$	5.8	$1.7 \cdot 10^0$	0	11
77.5	1.2	$8.0 \cdot 10^{-1}$	29.0	$3.2 \cdot 10^{-1}$	5.8	$1.7 \cdot 10^0$	0	9
80.0	1.0	$8.7 \cdot 10^{-1}$	24.9	$2.5 \cdot 10^{-1}$	5.2	$1.9 \cdot 10^0$	0	9
82.5	1.3	$1.7 \cdot 10^0$	21.1	$3.1 \cdot 10^{-1}$	4.6	$2.1 \cdot 10^0$	0	7
85.0	1.2	$1.3 \cdot 10^0$	17.6	$1.5 \cdot 10^{-1}$	4.8	$1.9 \cdot 10^0$	0	7
87.5	1.9	$2.6 \cdot 10^0$	14.2	$1.8 \cdot 10^{-1}$	4.6	$2.3 \cdot 10^0$	0	9
90.0	2.0	$3.1 \cdot 10^0$	11.1	$9.7 \cdot 10^{-2}$	4.3	$1.8 \cdot 10^0$	0	6

TABLE C.3: Counterpart of Table 6.3 for the 3D toy model coarse database contaminated with systematic error at random database positions (MSE automatic mode selection method).

%cont	%CD _m	%CD _{std}	%ID _m	%ID _{std}	k _m	k _{std}	%C ₁	%C ₂
5.0	99.9	$3.2 \cdot 10^{-1}$	17.9	$2.0 \cdot 10^1$	4.8	$1.6 \cdot 10^0$	3	41
7.5	99.9	$2.8 \cdot 10^{-1}$	13.1	$1.4 \cdot 10^1$	5.1	$1.3 \cdot 10^0$	0	78
10.0	99.9	$3.2 \cdot 10^{-1}$	9.8	$9.5 \cdot 10^0$	5.3	$1.4 \cdot 10^0$	0	89
12.5	99.9	$2.6 \cdot 10^{-1}$	7.2	$6.4 \cdot 10^0$	5.3	$1.3 \cdot 10^0$	0	92
15.0	99.9	$3.5 \cdot 10^{-1}$	8.1	$7.9 \cdot 10^0$	6.0	$1.9 \cdot 10^0$	0	95
17.5	99.9	$3.0 \cdot 10^{-1}$	5.9	$5.1 \cdot 10^0$	5.9	$1.5 \cdot 10^0$	0	96
20.0	99.9	$2.8 \cdot 10^{-1}$	5.1	$4.1 \cdot 10^0$	6.1	$1.6 \cdot 10^0$	1	94
22.5	99.8	$6.8 \cdot 10^{-1}$	5.4	$4.4 \cdot 10^0$	6.3	$1.6 \cdot 10^0$	0	87
25.0	99.3	$1.6 \cdot 10^0$	4.8	$3.3 \cdot 10^0$	5.8	$1.8 \cdot 10^0$	0	76
27.5	98.3	$2.5 \cdot 10^0$	5.7	$5.0 \cdot 10^0$	5.3	$2.2 \cdot 10^0$	0	52
30.0	97.2	$2.8 \cdot 10^0$	6.5	$4.2 \cdot 10^0$	4.9	$2.1 \cdot 10^0$	0	30
32.5	95.4	$5.0 \cdot 10^0$	8.3	$5.8 \cdot 10^0$	4.5	$2.0 \cdot 10^0$	0	17
35.0	93.1	$5.1 \cdot 10^0$	9.8	$6.0 \cdot 10^0$	3.7	$1.6 \cdot 10^0$	0	2
37.5	87.7	$1.2 \cdot 10^1$	11.7	$6.0 \cdot 10^0$	3.3	$1.2 \cdot 10^0$	0	0
40.0	83.7	$1.2 \cdot 10^1$	13.8	$5.9 \cdot 10^0$	3.3	$1.1 \cdot 10^0$	0	0
42.5	74.5	$1.6 \cdot 10^1$	16.4	$6.5 \cdot 10^0$	3.2	$1.1 \cdot 10^0$	0	0
45.0	63.7	$2.1 \cdot 10^1$	18.2	$7.9 \cdot 10^0$	3.0	$1.2 \cdot 10^0$	0	0
47.5	52.5	$2.1 \cdot 10^1$	20.8	$9.9 \cdot 10^0$	3.0	$1.3 \cdot 10^0$	0	0
50.0	43.3	$1.9 \cdot 10^1$	22.8	$1.1 \cdot 10^1$	3.0	$1.4 \cdot 10^0$	0	0
52.5	33.6	$1.7 \cdot 10^1$	22.3	$1.3 \cdot 10^1$	2.8	$1.4 \cdot 10^0$	0	0
55.0	29.9	$1.6 \cdot 10^1$	23.7	$1.3 \cdot 10^1$	2.9	$1.5 \cdot 10^0$	0	0
57.5	21.2	$1.5 \cdot 10^1$	21.5	$1.5 \cdot 10^1$	2.5	$1.6 \cdot 10^0$	0	0
60.0	18.6	$1.2 \cdot 10^1$	26.2	$1.5 \cdot 10^1$	2.8	$1.6 \cdot 10^0$	0	0
62.5	13.8	$1.1 \cdot 10^1$	26.5	$1.6 \cdot 10^1$	2.7	$1.6 \cdot 10^0$	0	0

TABLE C.4: Counterpart of Table 6.4 for the the CFD coarse database contaminated with systematic error at random database positions (MSE automatic mode selection method).

FIGURE C.5: Counterpart of Figure 6.5 for the dense 3D toy model database contaminated with positive, random error of maximum magnitude 0.5 at concentrated database points (MSE automatic mode selection method).

d	%CD _m	%CD _{std}	%ID _m	%ID _{std}	k _m	k _{std}	%C ₁	%C ₂
0.075	98.9	$8.0 \cdot 10^0$	100.5	$7.5 \cdot 10^1$	5.8	$9.3 \cdot 10^{-1}$	0	11
0.100	98.9	$8.0 \cdot 10^0$	100.5	$7.5 \cdot 10^1$	5.8	$9.3 \cdot 10^{-1}$	0	7
0.125	98.9	$8.0 \cdot 10^0$	100.5	$7.5 \cdot 10^1$	5.8	$9.3 \cdot 10^{-1}$	0	11
0.150	98.9	$8.0 \cdot 10^0$	100.5	$7.5 \cdot 10^1$	5.8	$9.3 \cdot 10^{-1}$	0	13
0.175	98.8	$6.0 \cdot 10^0$	36.8	$2.7 \cdot 10^1$	6.2	$1.4 \cdot 10^0$	0	6
0.200	98.8	$6.0 \cdot 10^0$	36.8	$2.7 \cdot 10^1$	6.2	$1.4 \cdot 10^0$	0	6
0.225	99.9	$2.0 \cdot 10^{-1}$	32.9	$4.3 \cdot 10^1$	7.0	$1.2 \cdot 10^0$	0	8
0.250	99.9	$2.0 \cdot 10^{-1}$	32.9	$4.3 \cdot 10^1$	7.0	$1.2 \cdot 10^0$	0	7
0.275	95.3	$9.3 \cdot 10^0$	18.2	$3.2 \cdot 10^1$	5.4	$2.4 \cdot 10^0$	0	12
0.300	95.3	$9.3 \cdot 10^0$	18.2	$3.2 \cdot 10^1$	5.4	$2.4 \cdot 10^0$	0	12
0.325	90.7	$1.0 \cdot 10^1$	41.9	$5.1 \cdot 10^1$	4.2	$2.7 \cdot 10^0$	0	7
0.350	90.7	$1.0 \cdot 10^1$	41.9	$5.1 \cdot 10^1$	4.2	$2.7 \cdot 10^0$	0	9
0.375	87.2	$2.0 \cdot 10^1$	39.2	$5.0 \cdot 10^1$	5.2	$3.0 \cdot 10^0$	2	14
0.400	87.2	$2.0 \cdot 10^1$	39.2	$5.0 \cdot 10^1$	5.2	$3.0 \cdot 10^0$	2	10
0.425	61.7	$2.0 \cdot 10^1$	4.0	$1.1 \cdot 10^1$	2.4	$2.3 \cdot 10^0$	0	9
0.450	61.7	$2.0 \cdot 10^1$	4.0	$1.1 \cdot 10^1$	2.4	$2.3 \cdot 10^0$	0	11
0.475	66.8	$1.1 \cdot 10^1$	4.4	$1.0 \cdot 10^1$	2.2	$9.5 \cdot 10^{-1}$	0	5
0.500	69.8	$7.9 \cdot 10^0$	3.2	$5.3 \cdot 10^0$	2.5	$9.7 \cdot 10^{-1}$	0	9
0.525	71.2	$1.1 \cdot 10^1$	6.0	$1.1 \cdot 10^1$	2.7	$8.8 \cdot 10^{-1}$	0	12
0.550	72.0	$8.4 \cdot 10^0$	6.1	$8.6 \cdot 10^0$	2.8	$8.9 \cdot 10^{-1}$	0	5
0.575	72.0	$8.4 \cdot 10^0$	6.1	$8.6 \cdot 10^0$	2.8	$8.9 \cdot 10^{-1}$	0	9
0.600	73.4	$6.9 \cdot 10^0$	5.7	$7.8 \cdot 10^0$	3.0	$8.8 \cdot 10^{-1}$	0	10
0.625	76.1	$7.1 \cdot 10^0$	7.5	$9.3 \cdot 10^0$	3.2	$8.6 \cdot 10^{-1}$	0	5
0.650	72.9	$9.0 \cdot 10^0$	6.8	$7.9 \cdot 10^0$	3.1	$9.0 \cdot 10^{-1}$	0	10
0.675	73.6	$8.4 \cdot 10^0$	7.8	$9.3 \cdot 10^0$	3.1	$9.1 \cdot 10^{-1}$	0	8
0.700	74.3	$6.6 \cdot 10^0$	10.8	$8.7 \cdot 10^0$	3.4	$8.9 \cdot 10^{-1}$	0	4
0.725	73.9	$6.6 \cdot 10^0$	10.1	$7.2 \cdot 10^0$	3.3	$7.7 \cdot 10^{-1}$	0	13
0.750	70.9	$8.3 \cdot 10^0$	9.9	$5.4 \cdot 10^0$	3.1	$9.1 \cdot 10^{-1}$	0	11
0.775	70.9	$8.3 \cdot 10^0$	9.9	$5.4 \cdot 10^0$	3.1	$9.1 \cdot 10^{-1}$	0	5
0.800	72.7	$5.6 \cdot 10^0$	9.7	$4.3 \cdot 10^0$	3.3	$8.4 \cdot 10^{-1}$	0	12
0.825	72.7	$7.0 \cdot 10^0$	9.1	$4.4 \cdot 10^0$	3.4	$8.7 \cdot 10^{-1}$	0	9
0.850	70.3	$6.1 \cdot 10^0$	8.5	$3.2 \cdot 10^0$	3.3	$8.7 \cdot 10^{-1}$	0	4

TABLE C.5: Counterpart of Table 6.5 for the 3D toy model coarse database contaminated with positive, random error of maximum magnitude 0.5 in a circular cylinder spanning the z-direction for varying diameter d (MSE automatic mode selection method).

b	%CD _m	%CD _{std}	%ID _m	%ID _{std}	k _m	k _{std}	%C ₁	%C ₂
0.050	77.6	$3.1 \cdot 10^1$	48.4	$6.1 \cdot 10^1$	4.7	$2.7 \cdot 10^0$	0	1
0.075	77.6	$3.1 \cdot 10^1$	48.4	$6.1 \cdot 10^1$	4.7	$2.7 \cdot 10^0$	0	1
0.100	71.4	$3.5 \cdot 10^1$	24.8	$3.0 \cdot 10^1$	4.2	$2.4 \cdot 10^0$	0	0
0.125	71.4	$3.5 \cdot 10^1$	24.8	$3.0 \cdot 10^1$	4.2	$2.4 \cdot 10^0$	0	0
0.150	71.4	$3.5 \cdot 10^1$	24.8	$3.0 \cdot 10^1$	4.2	$2.4 \cdot 10^0$	0	0
0.175	81.2	$3.0 \cdot 10^1$	27.4	$2.7 \cdot 10^1$	5.0	$2.4 \cdot 10^0$	0	2
0.200	81.2	$3.0 \cdot 10^1$	27.4	$2.7 \cdot 10^1$	5.0	$2.4 \cdot 10^0$	0	2
0.225	81.2	$3.0 \cdot 10^1$	27.4	$2.7 \cdot 10^1$	5.0	$2.4 \cdot 10^0$	0	2
0.250	81.2	$3.0 \cdot 10^1$	27.4	$2.7 \cdot 10^1$	5.0	$2.4 \cdot 10^0$	0	2
0.275	85.8	$2.5 \cdot 10^1$	27.2	$2.7 \cdot 10^1$	5.6	$2.6 \cdot 10^0$	0	7
0.300	85.8	$2.5 \cdot 10^1$	27.2	$2.7 \cdot 10^1$	5.6	$2.6 \cdot 10^0$	0	7
0.325	85.8	$2.5 \cdot 10^1$	27.2	$2.7 \cdot 10^1$	5.6	$2.6 \cdot 10^0$	0	7
0.350	84.7	$2.4 \cdot 10^1$	24.5	$2.5 \cdot 10^1$	5.9	$2.8 \cdot 10^0$	0	2
0.375	84.7	$2.4 \cdot 10^1$	24.5	$2.5 \cdot 10^1$	5.9	$2.8 \cdot 10^0$	0	2
0.400	84.7	$2.4 \cdot 10^1$	24.5	$2.5 \cdot 10^1$	5.9	$2.8 \cdot 10^0$	0	2
0.425	84.7	$2.4 \cdot 10^1$	24.5	$2.5 \cdot 10^1$	5.9	$2.8 \cdot 10^0$	0	2
0.450	84.7	$2.4 \cdot 10^1$	24.5	$2.5 \cdot 10^1$	5.9	$2.8 \cdot 10^0$	0	2
0.475	74.5	$2.7 \cdot 10^1$	13.5	$2.0 \cdot 10^1$	4.8	$2.9 \cdot 10^0$	0	5
0.500	74.5	$2.7 \cdot 10^1$	13.5	$2.0 \cdot 10^1$	4.8	$2.9 \cdot 10^0$	0	5
0.525	74.5	$2.7 \cdot 10^1$	13.5	$2.0 \cdot 10^1$	4.8	$2.9 \cdot 10^0$	0	5
0.550	74.5	$2.7 \cdot 10^1$	13.5	$2.0 \cdot 10^1$	4.8	$2.9 \cdot 10^0$	0	5
0.575	74.5	$2.7 \cdot 10^1$	13.5	$2.0 \cdot 10^1$	4.8	$2.9 \cdot 10^0$	0	5
0.600	74.5	$2.7 \cdot 10^1$	13.5	$2.0 \cdot 10^1$	4.8	$2.9 \cdot 10^0$	0	5
0.625	74.5	$2.7 \cdot 10^1$	13.5	$2.0 \cdot 10^1$	4.8	$2.9 \cdot 10^0$	0	5

TABLE C.6: Counterpart of Table 6.6 for the CFD coarse database contaminated with positive, random error at elliptical cylinder position (MSE automatic mode selection method).

FIGURE C.6: Counterpart of Figure 6.6 for the CFD coarse database contaminated with positive, random errors at elliptical cylinder position (MSE automatic mode selection method).

Bibliography

- [1] Vasilescu, G., *Electronic Noise and Interfering Signals*, Springer-Verlag Berlin Heidelberg, 2005.
- [2] Roy, C. J., "Review of Discretization Error Estimators in Scientific Computing," *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, No. 126, Orlando, Florida, 4-7 January 2010.
- [3] Taylor, J., *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements*, University Science Books, 1997.
- [4] Oppenheim, A. and Schafer, R., *Digital Signal Processing*, Prentice Hall, 1975.
- [5] Orfanidis, S., *Introduction to Signal Processing*, Prentice Hall, 2010.
- [6] Kroonenberg, P., *Applied multiway data analysis*, Wiley-Interscience, 2008.
- [7] Acar, E. and Yener, B., "Unsupervised Multiway Data Analysis: A Literature Survey," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 21, No. 1, 2009, pp. 6 – 20.
- [8] Golub, G. H. and Van Loan, C. F., *Matrix computations*, Johns Hopkins University Press, 3rd ed., 1996.
- [9] Chatterjee, A., "An introduction to the proper orthogonal decomposition," *Current Science*, Vol. 78, 2000, pp. 808 – 817.
- [10] Bui-Thanh, T., *Proper Orthogonal Decomposition Extensions and Their Applications in Steady Aerodynamics*, Master's thesis, Singapore-MIT Alliance, 2003.
- [11] Bui-Thanh, T., Damodaran, M., and Willcox, K., "Aerodynamic Data Reconstruction and Inverse Design Using Proper Orthogonal Decomposition," *AIAA Journal*, Vol. 42, 2004, pp. 1505 – 1516.
- [12] Lorente, L. S., Vega, J. M., and Velazquez, A., "Generation of Aerodynamics Databases Using High-Order Singular Value Decomposition," *Journal of Aircraft*, Vol. 45, 2008, pp. 1779 – 1788.
- [13] LeGresley, P. A. and Alonso, J. J., "Investigation of Non-Linear Projection for POD Based Reduced Order Models for Aerodynamics," *39th AIAA Aerospace Sciences Meeting & Exhibit*, Vol. 2001-0926, 2001.

- [14] Alonso, D., Velazquez, A., and Vega, J. M., "Robust reduced order modeling of heat transfer in a back step flow," *International Journal of Heat and Mass Transfer*, Vol. 52, No. 5 - 6, 2009, pp. 1149 – 1157.
- [15] Alonso, D., Velazquez, A., and Vega, J. M., "A method to generate computationally efficient reduced order models," *Computer Methods in Applied Mechanics and Engineering*, Vol. 198, No. 33-36, 2009, pp. 2683 – 2691.
- [16] Alonso, D., Vega, J., Velazquez, J., and de Pablo, V., "Reduced order modeling of three dimensional external aerodynamic flows," *Journal of Aerospace Engineering*, Vol. 25, 2012, pp. 588 – 599.
- [17] Ma, X. and Karniadakis, G. E., "A low-dimensional model for simulating three-dimensional cylinder flow," *Journal of Fluid Mechanics*, Vol. 458, 2002, pp. 181 – 190.
- [18] Braconnier, T., Ferrier, M., Jouhaud, J., Montagnac, M., and Sagaut, P., "Towards an adaptive POD / SVD surrogate model for aeronautic design," *Computers and Fluids*, Vol. 40, No. 1, 2011, pp. 195 – 209.
- [19] Miled, Z. B., Huian, L., Bukhres, O. A., Bem, M., Jones, R., and Oppelt, R. J., "Data Compression in a Pharmaceutical Drug Candidate Database," *Informatica*, Vol. 27, No. 2, 2003, pp. 213 – 224.
- [20] del Castillo-Negrete, D., Hirshman, S. P., Spong, D. A., and D'Azevedo, E. F., "Compression of magnetohydrodynamic simulation data using singular value decomposition." *Journal of Computational Physics*, Vol. 222, 2007, pp. 265–286.
- [21] Sirovich, L. and Kirby, M., "Low-dimensional procedure for the characterization of human faces," *Journal of the Optical Society of America A*, Vol. 4, No. 3, 1987, pp. 519 – 524.
- [22] Wall, M., Rechtsteiner, A., and Rocha, L., *A Practical Approach to Microarray Data Analysis*, chap. Singular value decomposition and principal component analysis, Kulwer Academic Publishers, 2003, pp. 91–109.
- [23] Everson, R. and Sirovich, L., "Karhunen-Loeve procedure for gappy data," *Journal of the Optical Society of America A*, Vol. 12, 1995, pp. 1657–1664.
- [24] Beckers, J. and Rixen, M., "EOF calculations and data filling from incomplete oceanographic datasets," *Journal of Atmospheric and Oceanic Technology*, Vol. 20, No. 12, 2003, pp. 1839–1856.
- [25] Venturi, D. and Karniadakis, G., "Gappy data and reconstruction procedures for flow past a cylinder," *Journal of Fluid Mechanics*, Vol. 519, 2004, pp. 315–336.
- [26] Kolda, T. G. and Bader, B. W., "Tensor Decompositions and Applications," *SIAM Review*, Vol. 51, 2009, pp. 455–500.

- [27] Tucker, L. R., *Contributions to mathematical psychology*, chap. The extension of factor analysis to three-dimensional matrices, Holt, Rinehart and Winston, New York, 1964, pp. 110 – 127.
- [28] Tucker, L. R., “Some mathematical notes on three-mode factor analysis,” *Psychometrika*, Vol. 31, 1966, pp. 279–311.
- [29] de Lathauwer, L., de Moor, B., and Vandewalle, J., “A multilinear singular value decomposition,” *SIAM Journal on Matrix Analysis and Applications*, Vol. 21, 2000, pp. 1253–1278.
- [30] de Lathauwer, L., de Moor, B., and Vandewalle, J., “On the Best Rank-1 and Rank-(R1,R2,...,RN) Approximation of Higher-Order Tensors,” *SIAM Journal on Matrix Analysis and Applications*, Vol. 21, 2000, pp. 1324 – 1342.
- [31] Bergqvist, G. and Larsson, E., “The Higher-Order Singular Value Decomposition: Theory and an Application,” *IEEE Signal Processing Magazine*, Vol. 27, No. 3, 2010, pp. 151 – 154.
- [32] Lorente, L. S., Vega, J. M., and Velazquez, A., “Compression of aerodynamic databases using high-order singular value decomposition,” *Aerospace Science and Technology*, Vol. 14, 2010, pp. 168 – 177.
- [33] Benito, N., Arias, J., Velazquez, A., and Vega, J., “Real time performance improvement of engineering control units via Higher Order Singular Value Decomposition: Application to a SI engine,” *Control Engineering Practice*, Vol. 19, No. 11, 2011, pp. 1315–1327.
- [34] de Lucas, S., Vega, J. M., and Velazquez, A., “Aeronautic conceptual design optimization method based on high order singular value decomposition,” *AIAA Journal*, Vol. 49, 2011, pp. 2713 – 2725.
- [35] Moreno, A., Jarzabek, A., Perales, J., and Vega, J., “Aerodynamic database reconstruction via gappy high order singular value decomposition,” *Aerospace Science and Technology*, 2016.
- [36] Moreno López, A., *Aerodynamic database reconstruction via gappy high order singular value decomposition*, Ph.D. thesis, Universidad Politécnica de Madrid, Spain, 2016.
- [37] Feddersen, H., Navarra, A., and N., W. M., “Reduction of Model Systematic Error by Statistical Correction for Dynamical Seasonal Predictions,” *Journal of Climate*, Vol. 12, 1999, pp. 1974–1989.
- [38] Kang, I.-S. and Lee, J.-Y., “Potential Predictability of Summer Mean Precipitation in a Dynamical Seasonal Prediction System with Systematic Error Correction,” *Journal of Climate*, Vol. 17, 2004, pp. 834–844.
- [39] Spalart, P. and Allmaras, S., *One equation turbulence model for aerodynamic flows*, No. AIAA-92-0439, 1992.
- [40] Eckart, C. and Young, G., “The approximation of one matrix by another of lower rank,” *Psychometrika*, Vol. 1, 1936, pp. 211 – 218.

- [41] Pearson, K., "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, Vol. 2, No. 6, 1901, pp. 559 – 572.
- [42] da Silva, V. and Lim, L.-H., "Tensor Rank and the ill-Posedness of the Best Low-Rank Approximation Problem," *SIAM J. Matrix Analysis Applications*, Vol. 30, 2008, pp. 1084–1127.
- [43] Yates, F., "The analysis of replicated experiments when the field results are incomplete," *Empire Journal of Experimental Agriculture*, Vol. 1, 1933, pp. 129 – 142.
- [44] Gunes, H., Sirisup, S., and Karniadakis, G. E., "Gappy data: To Krig or not to Krig?" *Journal of Computational Physics*, Vol. 212, 2006, pp. 358–382.
- [45] Abbott, I. H. and von, D. A. E., *Theory of wing sections*, Inc. Dover Publications, 2nd ed., 1959, pp. 47–63.
- [46] Edwards, J. R. and Chandra, S., "Comparison of eddy viscosity-transport turbulence models for three-dimensional, shock-separated flowfields," *AIAA Journal*, Vol. 34, No. 4, 1996, pp. 756 – 763.
- [47] University of Genoa, Dipartimento di Ingegneria Civile, Chimica e Ambientale, *Introductory OpenFOAM® Course*, July 2014.
- [48] Wilcox, D., *Turbulence Modelling for CFD*, D C W Industries, 1998.
- [49] Moukalled, F., Mangani, L., and Darwish, M., *The Finite Volume Method in Computational Fluid Dynamics*, Springer International Publishing AG Switzerland, 2016.
- [50] Schlichting, H., *Boundary-Layer Theory*, McGraw-Hill Book Company, 1979.
- [51] Drela, M., "XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils," *Conference on Low Reynolds Number Airfoil Aerodynamics*, Springer Berlin Heidelberg, 1989, pp. 1 – 12.
- [52] Tannehill, J. C., Anderson, D. A., and Pletcher, R. H., *Computational fluid mechanics and heat transfer*, Taylor & Francis, 1997.
- [53] Roache, P., "Perspective: A Method for Uniform Reporting of Grid Refinement Studies," *Journal of Fluids Engineering*, Vol. 116, No. 3, 1994, pp. 405 – 413.
- [54] Bellman, R., *Dynamic Programming*, Princeton University Press, Princeton, NJ, USA, 1st ed., 1957.
- [55] Shaw, R., "The influence of hole dimensions on static pressure measurements," *Journal of Fluid Mechanics*, Vol. 7, 1960, pp. 550–564.