

Efficient description of shape perturbations

Problem presented by

Amir Kayani, Richard Burguete

Airbus

Executive Summary

Airbus wish to have efficient ways of describing perturbations of a manufactured aerofoil from its design shape. The typical kind of perturbations expected are waves, steps, and bumps, and automatic classification into the classes is desired. Various possible methods of analysis were proposed and studied in some detail, including projection onto suitable basis functions, wavelets, and radial basis functions. Other methods were studied in less detail, but with the aim of giving a digital signature of defects that could be used to classify them.

Version 1.2
December 7, 2012
iii+34 pages

Report author

David Allwright (Industrial Maths KTN)

Yujia Chen (University of Oxford)

Artur Gower (NUI Galway)

Kei Fong Lam (University of Warwick)

Contributors

Amine Chakhchoukh (University of Bath)

Stephen Cowley (University of Cambridge)

Alistair Forbes (NPL)

Alexander Guterman (Moscow State University)

Paul Hammerton (University of East Anglia)

Dewei Jia (University of Oxford)

Aleksander Klimek (Polish Academy of Sciences)

Michael McCarthy (NPL)

Parousia Rockstroh (Simon Fraser University)

Piotr Ziolo (Polish Academy of Sciences)

ESGI85 was jointly organised by

The University of East Anglia

The Knowledge Transfer Network for Industrial Mathematics

and was supported by

The Engineering and Physical Sciences Research Council

Contents

1	Introduction	1
2	Wing structure	1
3	Overview of the relevance of this problem	1
3.1	Laminar flow control	1
3.2	Sampling rates	3
3.3	Measurement process	3
4	Geometric approaches	3
4.1	Unwrapping	3
5	Projection onto basis functions	6
5.1	Introduction	6
5.2	The Minimal Projection Method	9
5.3	Example of Application	15
6	Wavelets	18
6.1	Introduction	18
6.2	Application	21
6.3	Compressed sensing-style approach	25
7	Radial Basis Functions	26
7.1	Transforming the Problem to Interpolation or Approximation	26
7.2	Solving the Interpolation Problem by RBFs	27
7.3	Numerical Example: Scattered Data on the unit square	28
7.4	Possible Future Work	29
8	Other approaches	30
8.1	Global geometric invariants	30
8.2	Defect signature approach	31
8.3	Two-stage method	32
9	Conclusions	33
	Bibliography	33

1 Introduction

The background and problem definition were given in the circulated document [1]. The problem was presented by Amir Kayani on Monday 16th April, and the presentation slides are available at [2]. In essence the problem takes 2 forms. One is to compare point cloud data with a nominal profile and describe the perturbation in as concise a way as possible, as consisting of certain bumps, steps and waves. In the second form, there is no nominal profile, and the aim is to fit a smooth surface through the point cloud data, and then again describe the residual perturbation of the data from that surface in a concise way. During the Study Group, we had very helpful teleconferences with Airbus with David Belford (Manufacturing), Norman Wood and Paul Phillips (Aerodynamics), and Richard Burguete (Structural Mechanics).

2 Wing structure

It is helpful to have in mind the general structure of a wing and of its construction. The structure is built on the front and rear spars, which are long spanwise beams. Joining these at fairly regular intervals are the ribs, which are chordwise beams joining the 2 spars. The outer skin of the upper surface has stringers (spanwise stiffeners) attached to its underside, and is attached to the ribs and spars at fixing points (which may be of various kinds depending whether the structure is metal or composite). Often the outer skin on the upper surface is formed from two parts: one from the front spar back to the trailing edge, and another (the D-nose) forming the leading edge which has its own internal stiffener structure (partly visible on Slide 12 of [2]). There is therefore a join at the front spar between these 2 parts, and this is one place where a step may occur.

3 Overview of the relevance of this problem

3.1 Laminar flow control

(3.1.1) Airbus wish to have a characterization of the deviation of a wing from its ideal shape in order to assess whether the wing meets the design requirements. These in turn depend critically on the extent of the chord over which the boundary layer remains laminar. Anything that affects the transition of the boundary layer to turbulence is important. Things that affect that transition are both static and dynamic. The static features are shape perturbations from ideal, and in particular

- (a) roughness on the aerofoil surfaces (even roughness of the order of $1\ \mu$ could be significant for this),
- (b) discontinuities of curvature in the surface.

The main dynamic mechanisms that can initiate the transition to turbulence are believed to include at least:

-
- (a) free stream turbulence (though this is expected to be small at cruise altitudes),
 - (b) noise and vibration: this includes not only the passive elastic response of the wing to the engine noise and structural forcing, but also flutter, *i.e.* effects involving the aeroelastic feedback loop in which the elastic structural deformation affects the aerodynamic flow.

The interaction of the perturbations to wing shape with these unsteady terms may initiate cross-flow instabilities (on a swept wing), and Tollmien-Schlichting (TS) waves, and either of these will generally lead to the transition to turbulence in the boundary layer. The processes by which small amplitude disturbances enter the boundary layer are called receptivity, and the subsequent processes by which they grow are both linear and nonlinear. There is a current (2011–2016) major EPSRC-funded research programme entitled “LFC-UK: Development of Underpinning Technology for Laminar Flow Control” at Imperial College, and the last of the 6 major hurdles it aims to address is “How can we quantify the manufacturing tolerances, such as say surface waviness or bumps, needed to maintain laminar flow?” Clearly it is important for Airbus to follow and participate in the progress of this research programme. For TS waves, the receptivity of the boundary layer is determined by the local Fourier transform of the deviation of the surface from ideal. At different parts of the profile, different parts of the Fourier spectrum are important.

- (3.1.2) There is also research being carried out as part of the Clean Sky Joint Undertaking, [5], and the Topic Description document gives full details of its aims in this area.
- (3.1.3) A further cause of surface features in practice is the build-up of squashed insects on the aerofoils. There have apparently been various attempts to design insect-proof surfaces to avoid this problem.
- (3.1.4) The wing is not a rigid body. In particular it is subject to at least
 - (a) structural vibration,
 - (b) thermal stresses,
 - (c) the weight of the wing and fuel and aerodynamic loading.

The first of these has already been mentioned as a possible contributory effect to the transition to turbulence. Thermal stresses were mentioned by Airbus in a teleconference as a contributory effect to twist of the wing. The weight and aerodynamic loading will produce large length-scale deformations such as bending and twisting of the wing. Such deformations are then likely to alter the amplitudes of any waviness structures within the wing perturbation. Since the wing is designed to be light for the load it supports, and these structures are made up of very thin panels, the possibility of out-of-surface deformation of the wing panels has to be

considered. In some designs this is avoided: in others, some panels are designed with the view that at high loads some panels may begin to deform in a direction normal to the surface (in the sense that the compressive load is taken by other panels). These structural considerations are one of the reasons why waviness deviations are required to be characterized.

3.2 Sampling rates

- (3.2.1) Airbus are open to the fact that measurements to detect features of different in-surface wavelengths require different in-surface sampling rates. It is accepted that some areas of the wing may need to be scanned at different rates from others.

3.3 Measurement process

- (3.3.1) The measurement process can be carried out in various ways. One method used is by an optical method that covers the surface by snapshots, each of about $400\text{ mm} \times 400\text{ mm}$. Each snapshot produces a set of (x, y, z) points with some approximate regularity to it, but with irregularity of detail that is visible on closer inspection. The successive snapshots, and the movements of the scanner, are then combined together to produce an (x, y, z) point cloud over the whole surface.
- (3.3.2) The example point cloud dataset provided by Airbus is that illustrated in Slide 16 of [2]. It consists of 1514039 (x, y, z) points, covering Area 6 of a particular scan (but not that shown in Slides 10–11 of [2]). The points in the (x, y) -plane are shown in Figure 1, and the subsection $-500 < x < 490$, $-500 < y < 490$ is shown in Figure 2 to illustrate the way the points are approximately regularly spaced, but with visible departures from exact regularity.

4 Geometric approaches

In this section we shall describe some of the different geometric approaches to the problem of decomposing the shape perturbation into bumps, edges, and waves.

4.1 Unwrapping

- (4.1.1) Some approaches to analysing surface deviations can work directly on the curved surface itself, but for most of our work we assume that the deviations can be regarded as a function $w(u, v)$ of 2 variables, giving the perpendicular deviation w of the surface from its ideal position, as a function of two in-surface coordinates (u, v) .

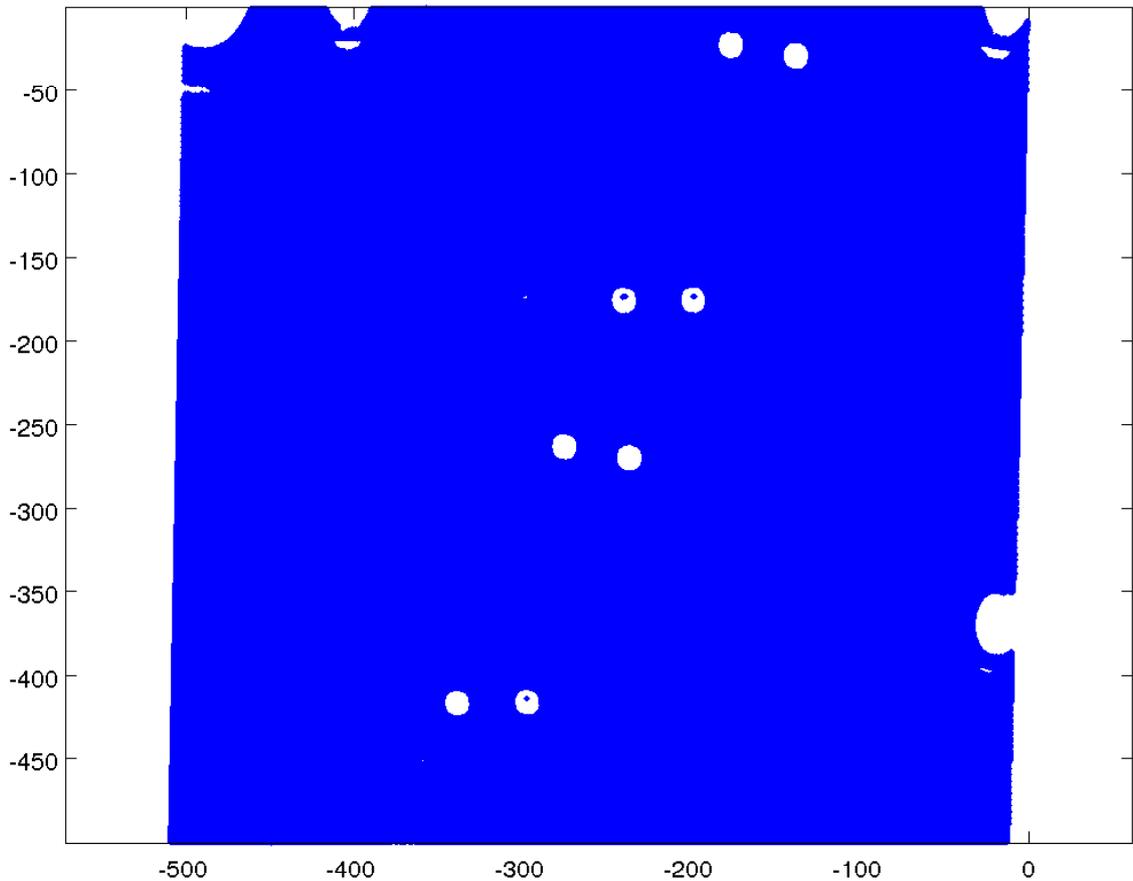


Figure 1: Region of the (x, y) -plane covered by the point cloud data

(4.1.2) To show how this could (in principle) be computed, suppose that the design shape of the wing is a function $\phi(u, v)$ taking a certain region of the (u, v) plane into the 3D shape in (x, y, z) -space. At the point $\phi(u, v)$ on the surface, there is a unit normal $\mathbf{n}(u, v)$, and we now extend the mapping ϕ to take (u, v, w) into (x, y, z) -space by

$$(x, y, z) = \phi_{\text{ext}}(u, v, w) = \phi(u, v) + w\mathbf{n}(u, v). \quad (1)$$

Provided that $|w|$ is less than the radius of curvature of the surface everywhere, this map remains 1-1, and we assume that the point cloud measurements are within the region that is covered in this 1-1 way. (Any failure of this would either indicate some gross artefact on the surface, or some defect in the measurement system.) The point cloud data can then be “unwrapped” back to (u, v, w) -space by ϕ_{ext}^{-1} to provide a point cloud where we are looking for deviations of w from zero, rather than deviations of (x, y, z) from the design surface defined by ϕ .

(4.1.3) An equivalent way to define this inverse mapping is that (u, v) are chosen

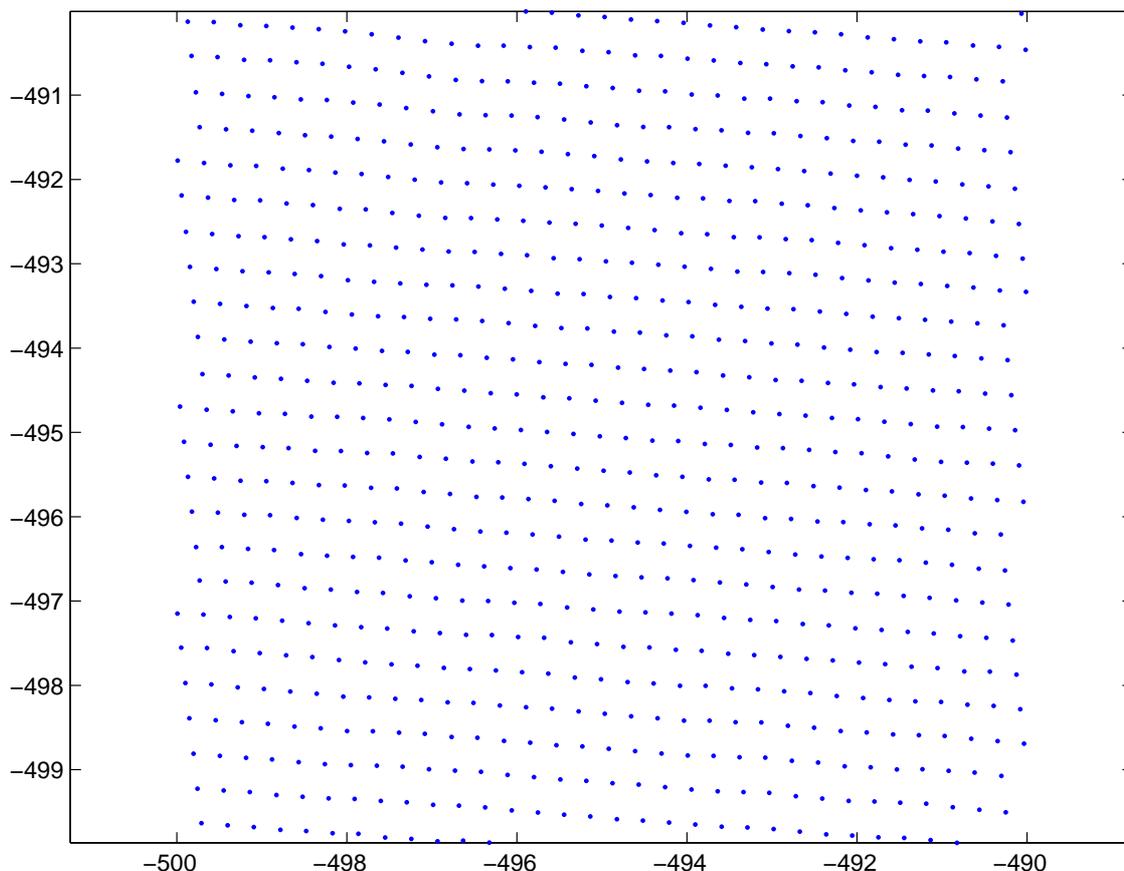


Figure 2: Enlarged view of the region $-500 < x < -490, -500 < y < -490$ showing the approximate regularity of the point cloud.

to be the parameters of the point $\phi(u, v)$ on the surface that is closest to (x, y, z) , and then w is defined by (1).

- (4.1.4) Not having the details of the CAD packages used, we do not know how easy or difficult this unwrapping might be in practice. If the surface is given analytically then finding the parameters of the closest point to (x, y, z) is a minimization. But after this has been done for one point and then u, v, w found by (1), we would expect that subsequent points can be processed as small perturbations, solving (1) by (say) Newton iteration using (u, v, w) from a nearby (x, y, z) point as starting values.
- (4.1.5) If instead there is a CAD description of the surface, then it presumably consists internally of some patchwise polynomial parameterization of the surface. So it would be necessary to put together the local parameters of each patch into a global (u, v) system first, and then proceed roughly as stated above, but with perhaps extra care near the boundaries and corners of the patches.

5 Projection onto basis functions

5.1 Introduction

- (5.1.1) This method was applied to the data after the unwrapping process, or inverse map of the design intent. The basic idea is to build a set of vectors, \mathcal{B} , or fault vectors, that represent the faults we wish to find, then “project” the unwrapped data onto these fault vectors in \mathcal{B} . Then from looking at the coefficients of this “projection” we choose only a few of the most important vectors. Finally we re-project the original unwrapped data onto these important vectors from \mathcal{B} . Hopefully at this point we will have a good representation of the wing along with the information of what and where are the faults.
- (5.1.2) Now we will carefully explain the steps of this method for a slice of data, a 1D curve. It is possible that to apply the above to the full 2D surface wing could be infeasible: however if describing the faults on an arbitrary slice is well resolved then it should be rather straightforward to join these slices to describe the total wing Faults.

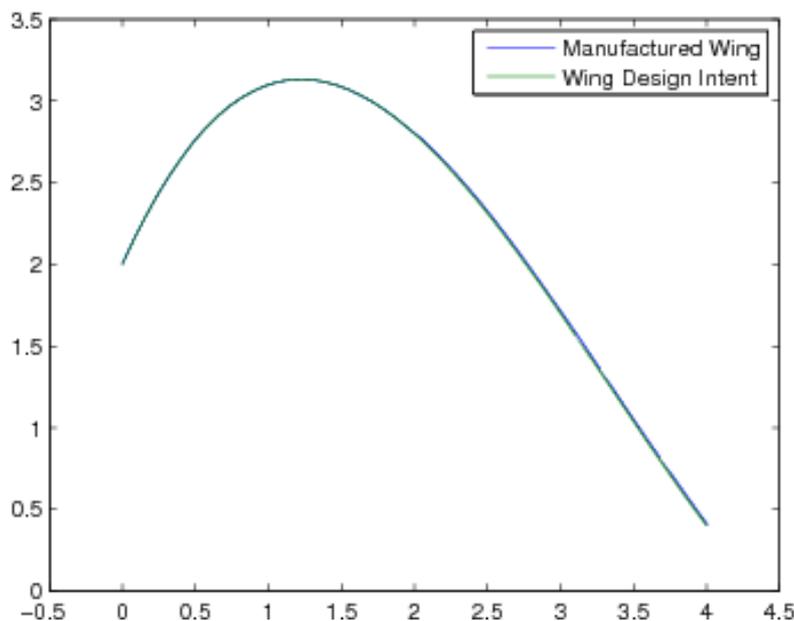


Figure 3: Example: wing design intent and the measurement of the manufactured wing: they are almost indistinguishable.

- (5.1.3) To design these fault vectors we need to first recognize what the faults will look like after the unwrapping. If the wing was produced with no flaws whatsoever the unwrapped data slice, to be referred to as data from now on, would be a straight line. For example if two plates were joined together

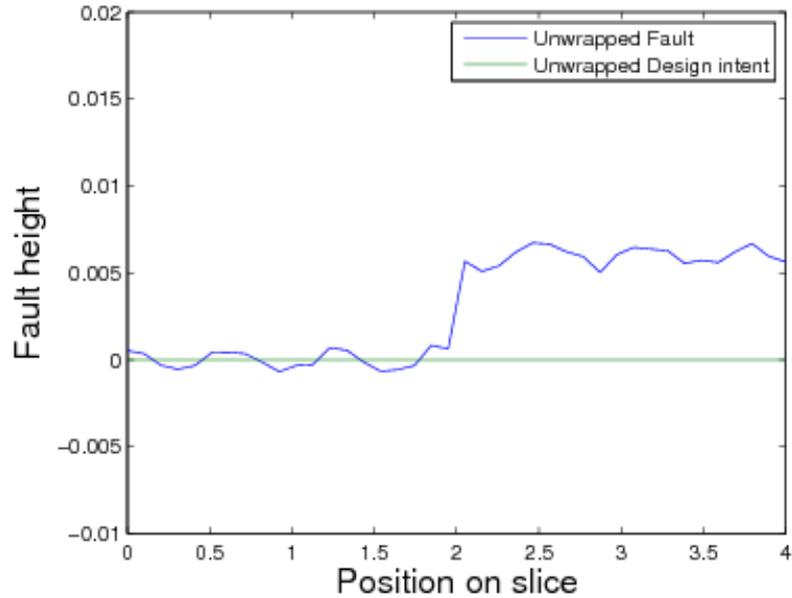


Figure 4: The corresponding unwrapped design intent and the measured wing.

with a slight error, but both were angled correctly, then the unwrapping of this data would produce a step as in Figure 4 below.

(5.1.4) The fluctuations from an exact step are due to errors in the measurement, which at times can be on the same scale as the fault we wish to find. To locate this step we construct a discrete basis for all possible steps and then project the data onto these steps. An example of a few of these basis elements can be seen in Figure 5.

(5.1.5) Mathematically speaking, each discrete step will be represented by \mathbf{b}_0^n , a vector in \mathbb{R}^N (where N is the number of points in the data slice) defined by

$$b_0^n(m) := \begin{cases} c_0 & \text{if } m \geq n, \\ 0 & \text{if } m < n, \end{cases} \quad (2)$$

where c_0 is a constant that will be chosen later. Let f represent the data. Then we wish to find α_0^n 's to solve the problem:

$$\min_{\alpha_0^n} F(\alpha_0^1, \alpha_0^2, \dots, \alpha_0^N), \quad (3)$$

where $F = \|f - \sum_{n=0}^N \alpha_0^n \mathbf{b}_0^n\|$ and $N =$ number of data points.

(5.1.6) This is accomplished by finding the critical point, *i.e.* finding the α_0^n 's such

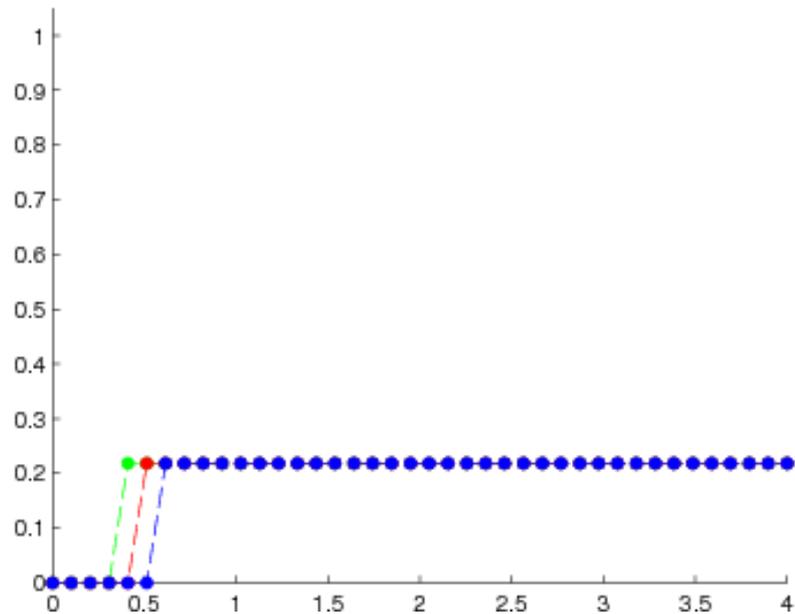


Figure 5: Step vectors with different starting positions.

that

$$\frac{\partial F}{\partial \alpha_0^n} = 0, \text{ for every } n \text{ from } 1 \text{ to } N. \quad (4)$$

Then we say that $\sum_n \alpha_0^n \mathbf{b}_0^n$ is the projection of f onto the step vectors.

(5.1.7) See Figure 6 for an example of the α_0^n 's distribution for the fault in Figure 4. Clearly in this case the most relevant coefficient is $\alpha_0^{N/2} \approx 0.027$. This corresponds to a step that starts at $x = 2$, so we choose only this coefficient and plot $\alpha_0^{N/2} \mathbf{b}_0^{N/2}$ with the original fault resulting in Figure 7.

(5.1.8) Note that, for this example, if we were to remove the errors of the measurement and compare the real fault with the projection of the measured fault they would be almost identical. That is, by taking only the most relevant coefficients of the projection we somewhat ignore the measurement errors.

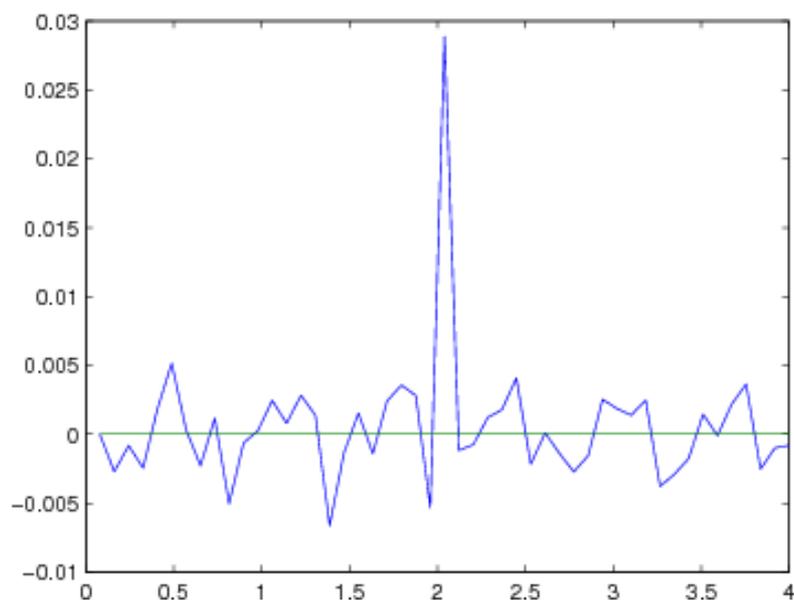


Figure 6: The value of the vector α_0^n . The x -axis is labelled with the step position corresponding to n .

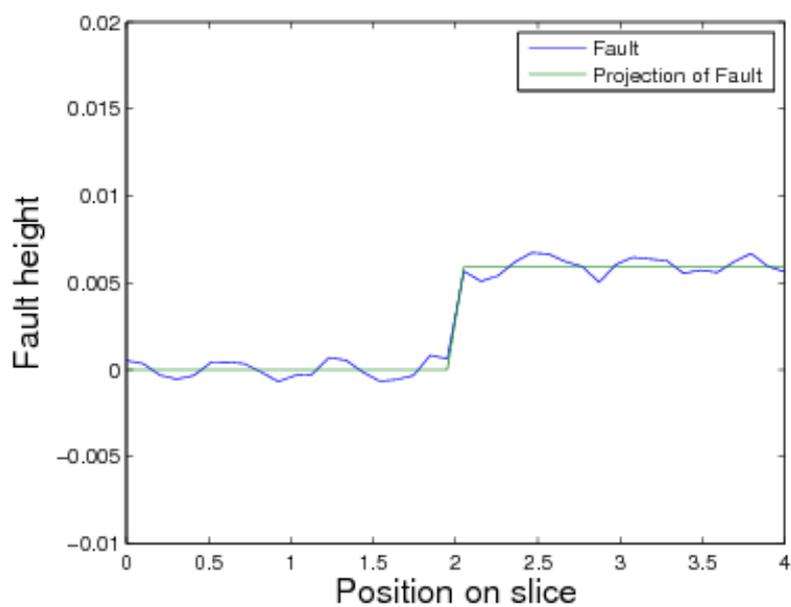


Figure 7: The projection of the measured fault compared with the measured fault itself.

5.2 The Minimal Projection Method

(5.2.1) To begin describing this method we present all the basis vectors which will be used in this projection method. To do so, let the slice be composed of

the N points $\{(x_1, f(x_1)), \dots, (x_N, f(x_N))\} = (\mathbf{x}, f(\mathbf{x}))$. Then the order- k fault vector starting at x_n is defined by

$$\mathbf{b}_k^n(m) := \begin{cases} c_k(x_m - x_n)^k & \text{if } m \geq n, \\ 0 & \text{if } m < n, \end{cases} \quad (5)$$

where c_k is such that $\|\mathbf{b}_k^{N/2}\| = 1$.

- (5.2.2) Later on we shall see that c_k must be a constant so as to accurately locate the position of the faults. It will be a result of all the fault vectors of order- k having the same growth from their “starting” point: for example the step vectors will have the same height. For the record, c_k will be the typical L_2 norm of the vector

$$\mathbf{c}_k(m) := \begin{cases} (x_m - x_{N/2})^k & \text{if } m \geq N/2, \\ 0 & \text{if } m < N/2. \end{cases} \quad (6)$$

- (5.2.3) Given these vectors, one may deduce from the introduction the following method: project the data onto step vectors, then choose only the most relevant coefficients β_0^n 's, store them and let $p_0 \leftarrow \sum_n \beta_0^n \mathbf{b}_0^n$, then remove this projection from the data $f \leftarrow f - p_0$. Now again project this remainder data f onto the fault vectors of order-1, the line basis, choose only the most relevant coefficients β_1^n 's, store them and so on.

- (5.2.4) At the end of this process all you need do is consult the coefficients β_k^n 's to find what faults are present and their location. The major flaw in this method lies in attempting to define carefully “the most relevant coefficient”, for the \mathbf{b}_k^n 's are not linearly independent¹, so the same fault will be clearly detected by different order fault vectors. It would then be unclear what criteria we could use to determine if a coefficient α_k^n is relevant enough.

- (5.2.5) Another crucial feature is that we want the least number of coefficients to characterize the faults. To summarize, we want to represent the data in terms of all these basis vectors while choosing only the most relevant coefficients. Mathematically speaking, we want to find α_k^n 's such that,

$$\min_{\alpha_k^n} F(\alpha_0^1, \alpha_0^2, \dots, \alpha_K^N),$$

$$\text{where } F(\alpha_0^1, \alpha_0^2, \dots, \alpha_K^N) = \left\| \sum_{n=1}^N \sum_{k=0}^K \alpha_k^n \mathbf{b}_k^n - f \right\|^2 + \omega \left\| \sum_{n=1}^N \sum_{k=0}^K \alpha_k^n \right\|^2$$

where ω is a weight that can be chosen. This is, in effect, a Tychonov regularization of the problem, with ω as the regularization parameter.

¹If we use basis vectors for $k = 0, 1, 2, 3, 4$, that would give us $5N$ fault vectors to represent an N -dimensional space.

(5.2.6) To minimize, we equate

$$\frac{\partial F}{\partial \alpha_i^j} = 0$$

for every appropriate i and j , resulting in

$$\begin{aligned} \frac{\partial F}{\partial \alpha_i^j} = \langle \mathbf{b}_i^j, \sum_{n=1}^N \sum_{k=0}^K \alpha_k^n \mathbf{b}_k^n - f \rangle + \omega \alpha_i^j = 0 \implies \\ \sum_{n=1}^N \sum_{k=0}^K (\omega \delta_{ik} \delta^{jn} + \langle \mathbf{b}_i^j, \mathbf{b}_k^n \rangle) \alpha_k^n = \langle \mathbf{b}_i^j, f \rangle, \end{aligned} \quad (7)$$

where $\delta_{ik} = 1$ if $i = k$ and zero otherwise, $\delta^{jn} = 1$ if $j = n$ and zero otherwise. If we denote

$$\boldsymbol{\alpha} = (\alpha_0^1, \alpha_0^2, \dots, \alpha_0^N, \alpha_1^1, \alpha_1^2, \dots, \alpha_1^N, \alpha_2^1, \dots, \alpha_K^N)^T,$$

then equation (7) can be rewritten as a matrix equation,

$$(\mathbf{B}^T \mathbf{B} + \omega \mathbf{I}) \boldsymbol{\alpha} = \mathbf{B}^T f, \quad (8)$$

where,

$$\mathbf{B} = (\mathbf{b}_0^1 \quad \mathbf{b}_0^2 \quad \dots \quad \mathbf{b}_0^N \quad \mathbf{b}_1^1 \quad \mathbf{b}_1^2 \quad \dots \quad \mathbf{b}_1^N \quad \mathbf{b}_2^1 \quad \dots \quad \mathbf{b}_K^N), \quad (9)$$

and \mathbf{I} is a KN -dimensional identity matrix. Note that if $\omega > 0$ then $\mathbf{B}^T \mathbf{B} + \omega \mathbf{I}$ is invertible since $\mathbf{B}^T \mathbf{B}$ is certainly positive semi-definite. Hence,

$$\boldsymbol{\alpha} = (\mathbf{B}^T \mathbf{B} + \omega \mathbf{I})^{-1} \mathbf{B}^T f. \quad (10)$$

(5.2.7) If \mathbf{B} 's columns were composed only of fault vectors of order- k and $\omega = 0$, then the above formula would give the coefficients for the projection of the data onto the order- k fault vectors.

(5.2.8) Let us examine the distribution of the α_i^j 's for the data given in Figure 8. Using equation (10) we can find $\boldsymbol{\alpha}$, shown in Figure 9.

Note that the the same fault vectors perceive the same events. Now we choose the most representative coefficients to be the largest local maximum of $|\boldsymbol{\alpha}|$, the absolute value of the components of the coefficient vector $\boldsymbol{\alpha}$. This is done separately for the coefficients of each fault vector order. Then we project the data onto the fault vectors of these coefficients. Note, if c_k 's in equations (5) were not constants then the critical points in Figure 9 would be dislocated and would not represent the position of the fault. The result, which we call the reconstructed fault is shown in Figure 10 below.

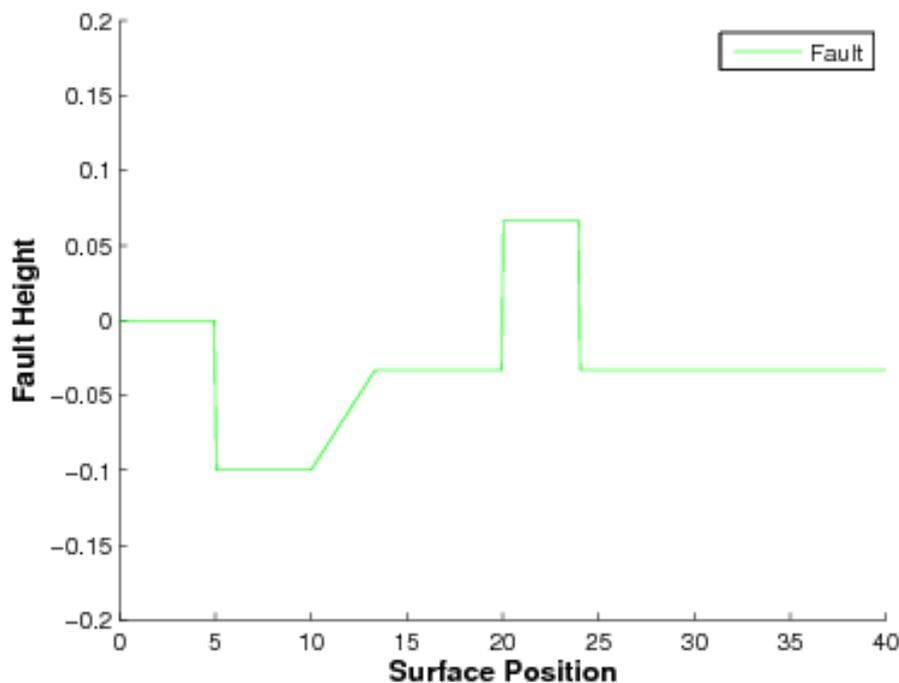


Figure 8: A Fault example.

(5.2.9) When measurement errors are present it is necessary to use higher order fault vectors, defined in equations (5), to obtain better results. Figure 11 shows an example of a reconstructed fault where the data include measurement errors. When dealing with measurement errors possibly the best procedure would be to smooth the data, though we can apply the method directly to the data.

(5.2.10) For the minimal projection method presented below let,

K be the largest order fault vector defined by equations (5),

M_k be the number of representative coefficients for the fault vectors of order- k ,

$$|\alpha_k| = (|\alpha_k^1|, |\alpha_k^2|, \dots, |\alpha_k^N|),$$

then the method is illustrated by the algorithm below:

Minimal Projection Method

1. Choose K .
 2. Choose each M_k for k from 1 to K .
 3. For data f use equation (10) to obtain α .
 4. Let β_k be the M_k largest local maximum of $|\alpha_k|$.
 5. Let Pf be the Projection of f onto the fault vectors of all the β_k 's.
 6. Check that Pf accurately approximates f , if not increase the M_k 's that need more representatives and go back to item 4.
-

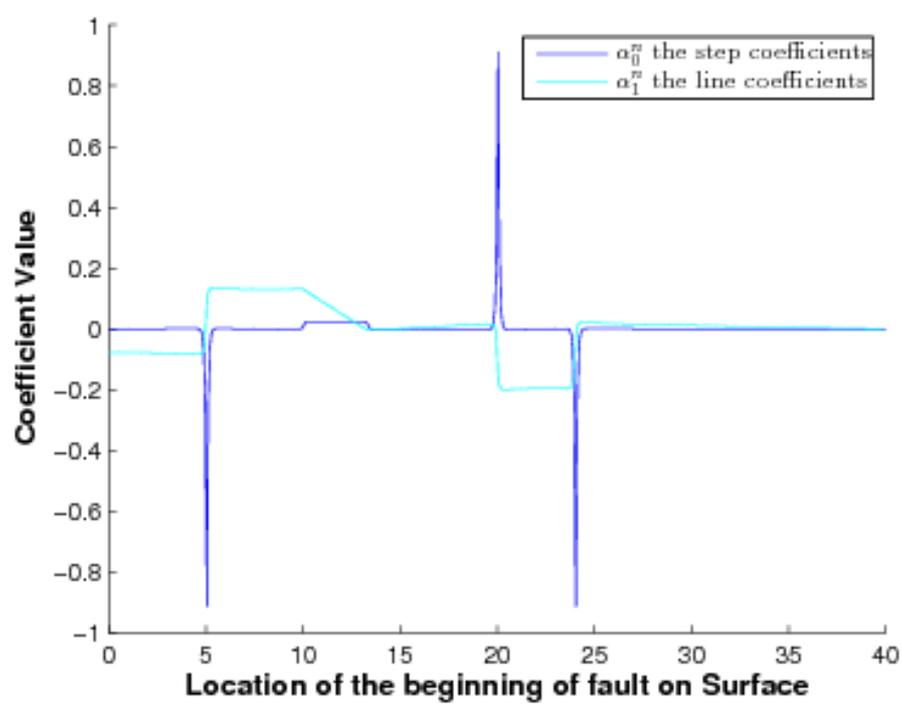


Figure 9: Step and line fault coefficients. The line coefficients have been multiplied by 20 to make them visible on the same scale as the step coefficients.

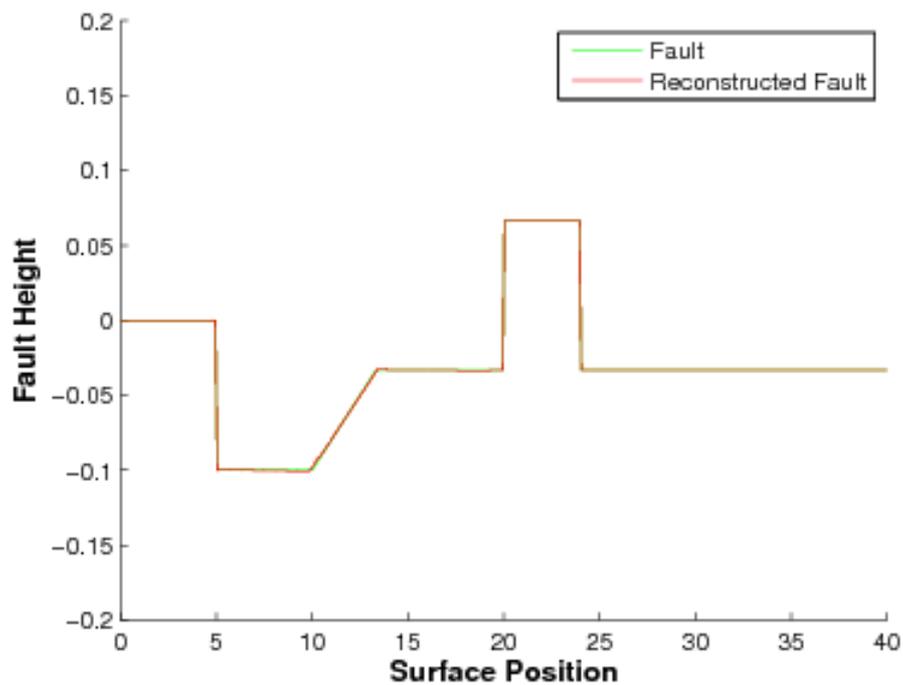


Figure 10: Fault and reconstructed fault.

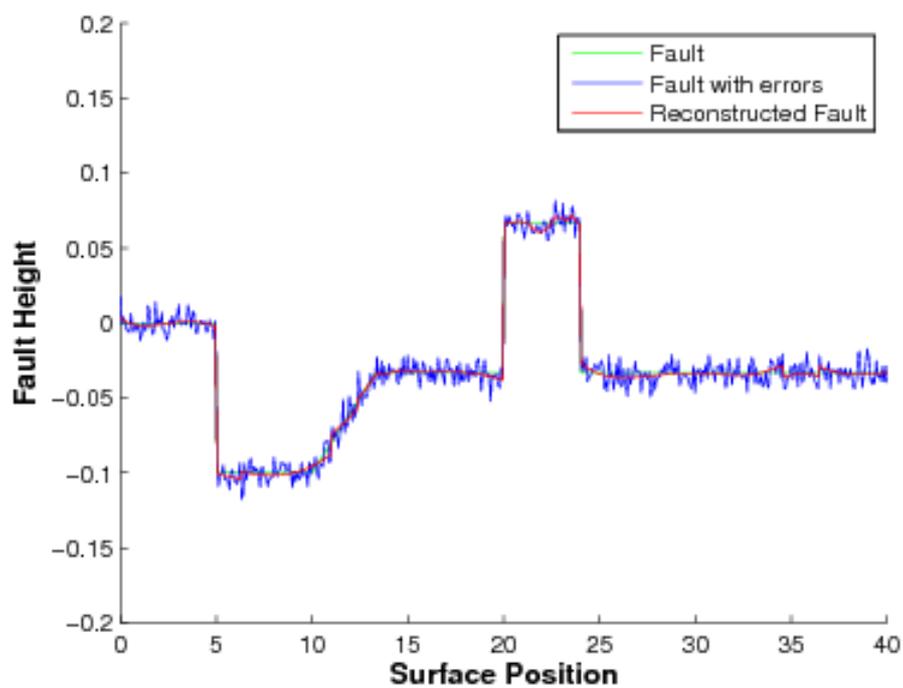


Figure 11: Fault with measurement errors and reconstructed fault.

5.3 Example of Application

- (5.3.1) With the help of Dr. Richard Burguete we designed some synthetic data that is closer to the scales and possible events that occur on a slice of measured data from a wing. The chosen data imitates a bolt head, a wave and a misaligned plate see Figure 12. If we consider the extension of this slice to be 40 cm then: the bolt head protrudes 0.04 mm above the plate, the wave's amplitude is 0.04 mm and the plates are misaligned by 0.027 mm. We have used only 1500 points on the slice so that there are 10 measured points on the bolt's head.

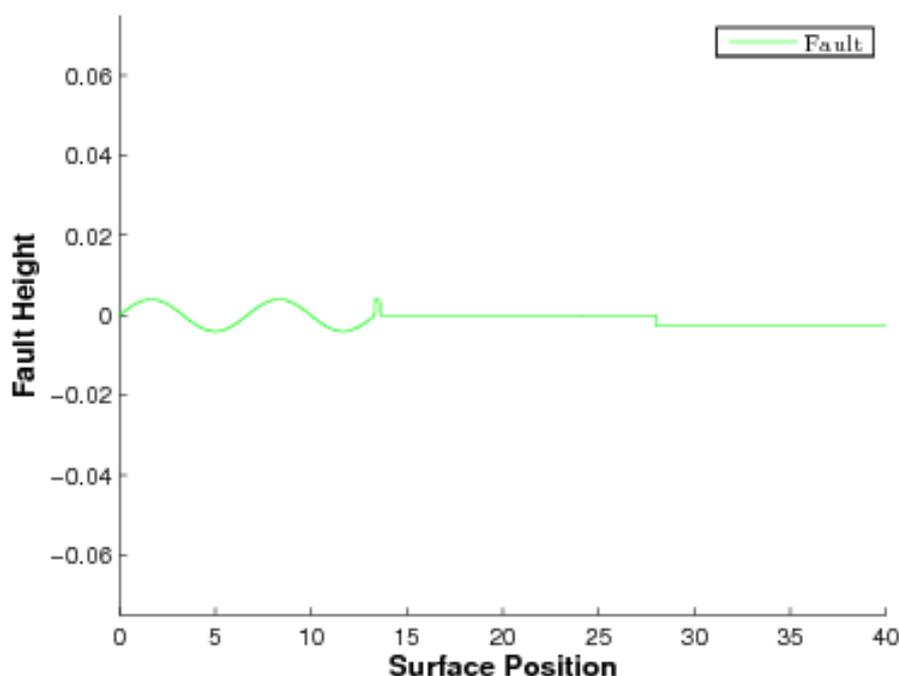


Figure 12: Synthetic data with a bolt head, a wave and a misalignment of two plates.

- (5.3.2) We have modelled the measurement errors by Gaussian white noise with an average amplitude equal to a third of the bolt head height. Now we apply the minimal projection method to this data, that contains the measurement errors, using fault vectors up to order-5, defined by equations (5), and choose the 5 most representative coefficients for each order of fault vector, i.e. $M_0 = M_1 = M_2 = M_3 = M_4 = M_5 = 5$.
- (5.3.3) We compare the output of the method (reconstructed data), the data and the original fault in Figure 13 and onwards.

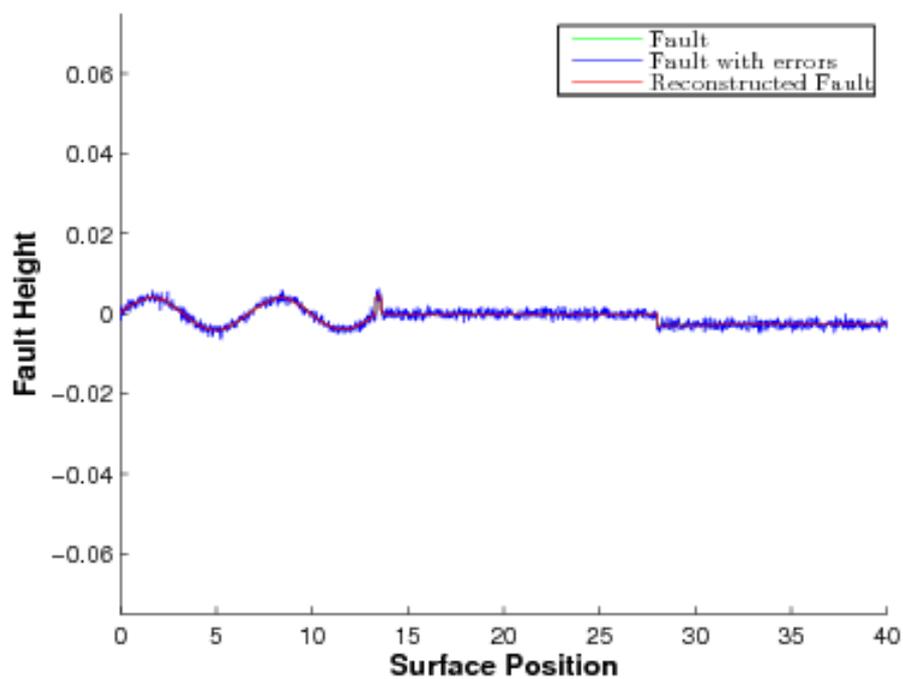


Figure 13: The green line is the fault, the blue is the measured data and red the output of the method.

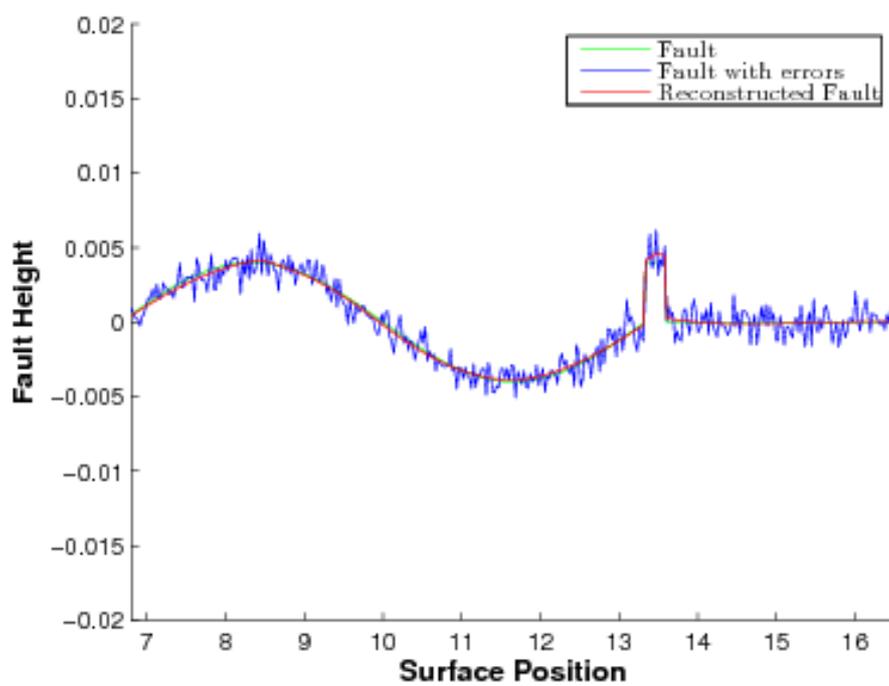


Figure 14: Close-up of wave.

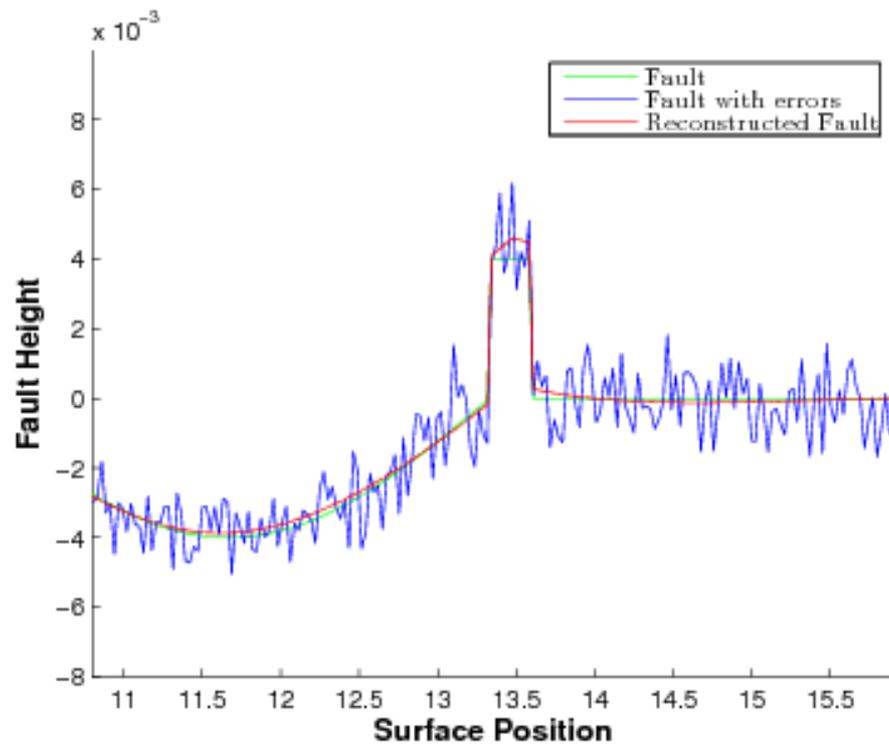


Figure 15: Close-up of bolt.

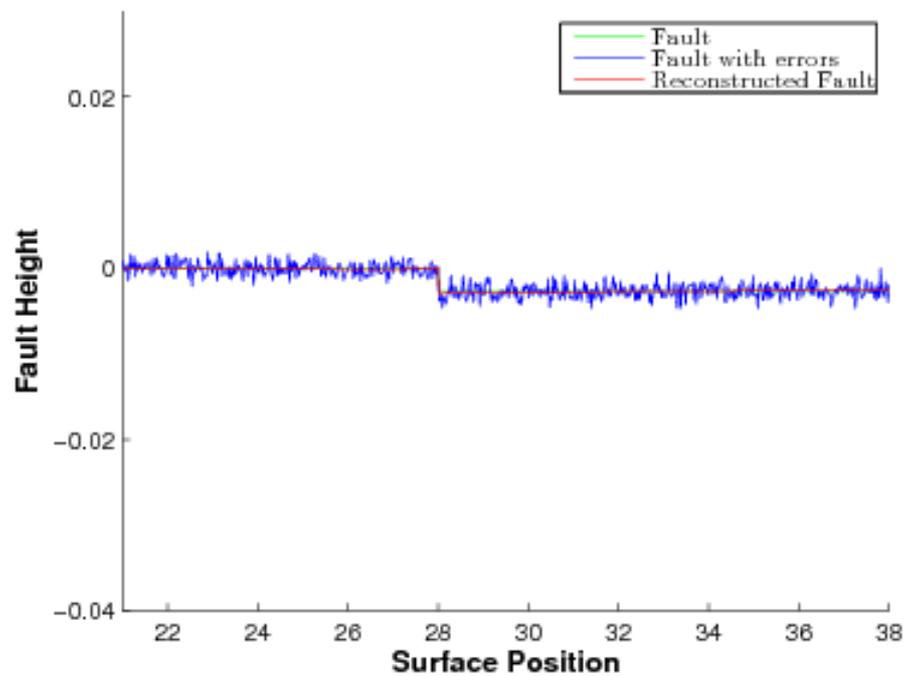


Figure 16: The misalignment.

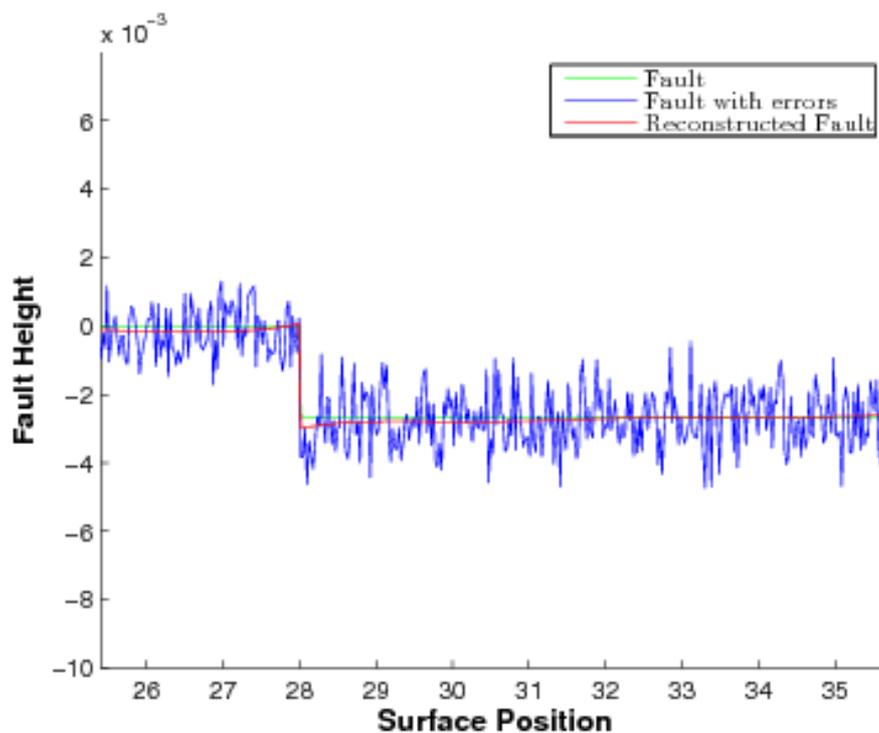


Figure 17: Close-up of the misalignment.

- (5.3.4) We feel that this minimal projection method works very well and we predict it would have better results if the measured data was to be treated to smooth these error measurements. Another possible method is to project the data onto the fault vector of different orders, one-by-one, record all the local minimums and maximums of the coefficients α_n^k 's called β , then finally project the data onto the fault vectors of all coefficients in β .

6 Wavelets

6.1 Introduction

- (6.1.1) The Fourier transform is a powerful tool in providing information about the frequencies of a signal, but it is not able to return the time interval where these frequency components exist. Hence, the Fourier transform is not well suited to analyse a signal that has time-varying frequency content. (In the context of the Airbus problem, we are of course thinking of a function of 2-dimensional space rather than a function of time, but we shall use the time-frequency terminology here and leave the reader to make the analogy with space-wavenumber.)
- (6.1.2) The short-term Fourier transform tries to remedy this problem by dividing

the signal into small segments, where these segments are assumed to be stationary, then introducing a window function with width equal to the length of the segment. One can take the Fourier transform of the convolution between the signal and the window function to obtain the frequency components of that segment. The window is then shifted along and the previous step is repeated.

(6.1.3) Due to the Heisenberg Uncertainty Principle, one cannot know the exact time-frequency representation of the signal. The problem with the short-term Fourier transform is that the width of the window function is fixed, which results in poorer frequency resolution, *i.e.*, we only gain information about a band of frequencies that exist. Wider windows might achieve better frequency resolution, but they may violate the stationarity condition we assume that the signal possesses. This can be summarised as the following dilemma:

- a window of infinite length, which is equivalent to the usual Fourier transform, gives perfect frequency resolution, but no time resolution;
- narrower windows give better time resolution but poorer frequency resolution.

(6.1.4) Wavelets and Multiresolution analysis analyse the signal at different frequencies with different resolutions, in particular they obtain

- good time resolution and poor frequency resolution at high frequencies;
- poor time resolution and good frequency resolution at low frequencies;

(6.1.5) The wavelet transform is developed as an alternative to the short-term Fourier transform to overcome the resolution problem. The wavelet analysis is done in a similar fashion to the short-term Fourier transform analysis, in the sense that the signal is convolved with a function and the wavelet transform is calculated separately for different segments of the signal. The two main differences between the wavelet transform and the short-term Fourier transform are

- the Fourier transform of the convolution is not taken;
- the width of the window is changed as the transform is computed for every spectral component.

(6.1.6) Starting off with a fundamental function ψ , which is often called the “mother wavelet”, the wavelet transform involves taking the convolution of the signal S and translated/scaled versions of the mother wavelet ψ . “Translated” relates to shifting the window through the signal, which corresponds to the time information and “Scaled” corresponds to the level of detail. High scales (or low frequencies) correspond to a non-detailed

global view of the signal and low scales (or high frequencies) correspond to a detailed view of the signal.

- (6.1.7) At the most basic level, the signal S is passed through two complementary filters and emerges as two signals: The high frequency components are encompassed in the **D**etails and the low frequencies are summarised in the **A**pproximation. However, in passing through these filters we gain twice as many sample points. Thus the next step is to downsample and throw away every second data point. This is illustrated in Figure 1.

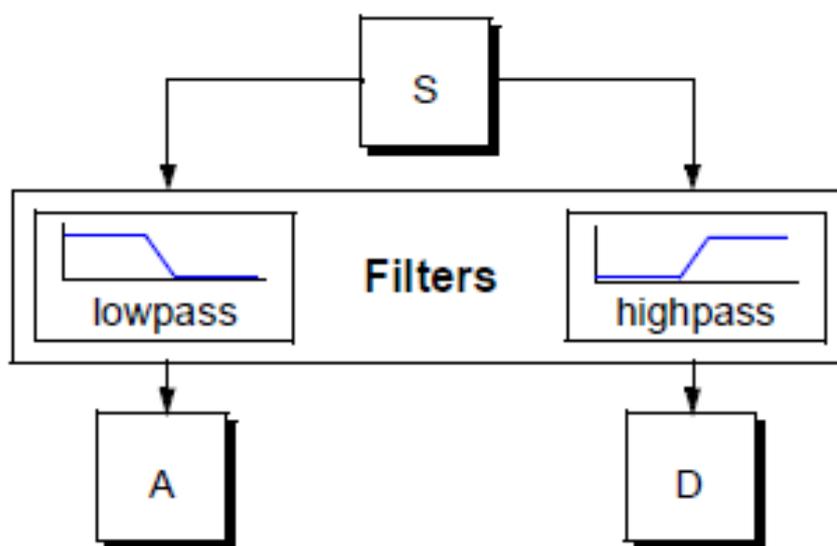


Figure 1: Wavelet Decomposition Diagram

- (6.1.8) The filters used in the above decomposition usually arise from a family of wavelet functions. Two common wavelet families are the Haar wavelet (Figure 2) and the Daubechies family (Figure 3).

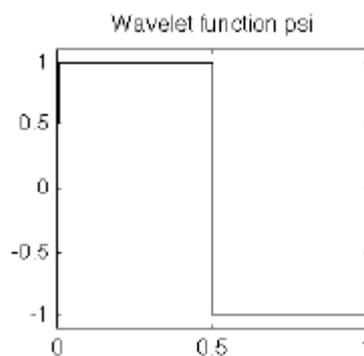


Figure 2: Haar wavelet

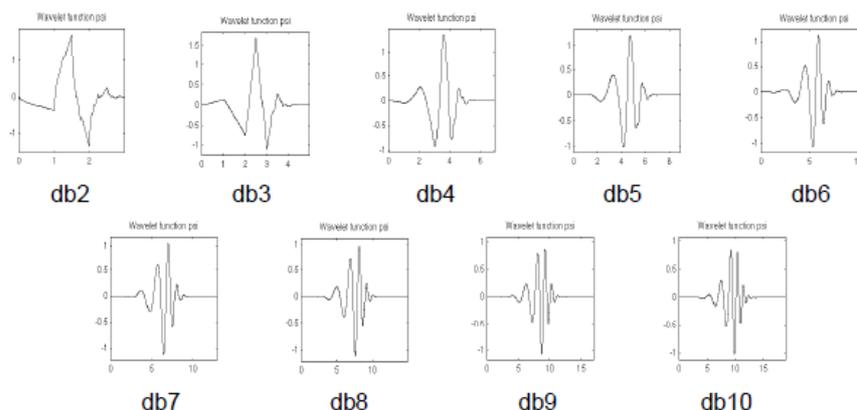


Figure 3: Daubechies family of wavelets

- (6.1.9) In passing the signal through a highpass filter, the Detail part of the decomposition only contains the upper half of frequency band of the original signal, hence the frequency resolution is doubled, but the downsampling procedure halves the time resolution.
- (6.1.10) One can repeat the decomposition on the Approximation segment of the signal to build a multi-level analysis (Figure 4).

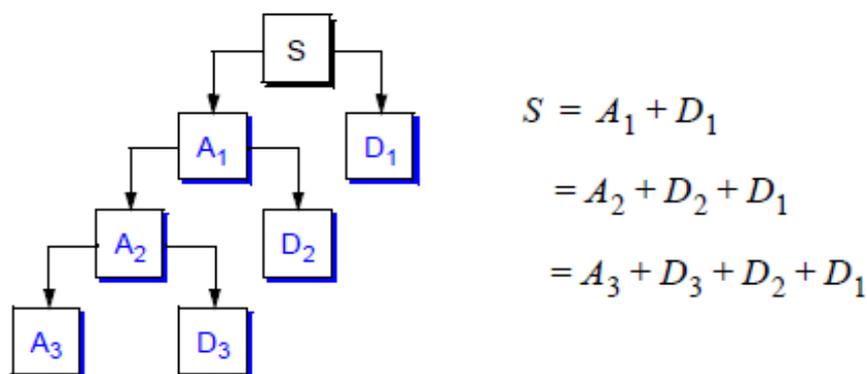


Figure 4: Multi-level analysis

6.2 Application

- (6.2.1) Our first investigation into the application of wavelets to Airbus' problem is to distinguish the various types of defects that can arise during manufacture. We test the method on an artificial data set that consists of steps, jumps in the gradient and in curvature. The data set is shown in Figure 5 and it consists of steps at $x = -0.75, x = 0.5$, jumps in the gradient at $x = -0.25, x = 0$ and a jump in the curvature at $x = 0.25$. (We include this jump in curvature because of the importance of such jumps for the

aerodynamics as we mentioned earlier.) We use the **wavemenu** package available in Matlab to analyse this data set with a 10-level decomposition using the Daubechies-8 wavelet. Since we are only interested in the location of defects, which compose the high frequency part of the signal, we focus our attention on the Details (D1-D10).

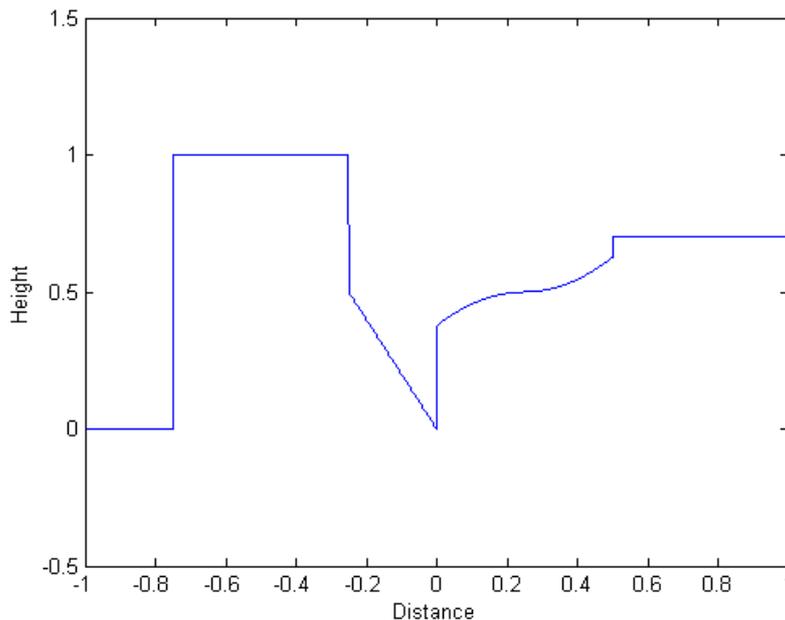


Figure 5: Artificial data for defect identification

- (6.2.2) The result is shown in Figure 6 and we observe that the steps are the most prominent, followed by the jump in gradient. At first glance it appears that the high frequency components are not able to pick out the jump in curvature, but in Figure 7 we see that the jump in curvature has a much smaller size $\sim 10^{-5}$ compared to the jump in gradient $\sim 10^{-3}$ and the steps $\sim 10^{-1}$.
- (6.2.3) This is mainly because the defect in curvature is smoother than either of the first two kinds of defect and hence will have a less prominent signature than the other two. It is currently unknown if there is a robust method of identifying curvature defects in an arbitrary slice of data. One way to spot them is to systematically identify and remove the largest signature from the data, then apply the same wavelet analysis to find and remove the next largest signature, and so on. Eventually the signature of the curvature defects will show up once all other prominent defects are identified and removed.
- (6.2.4) Finally we note that the size of the defect has a positive effect on the size of the signature, as observed in Figure 6 where the smaller step at $x = 0.5$

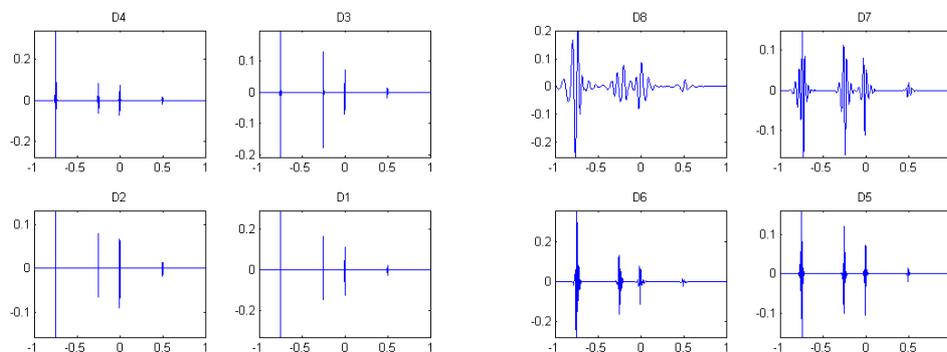


Figure 6: Wavelet decomposition of artificial data

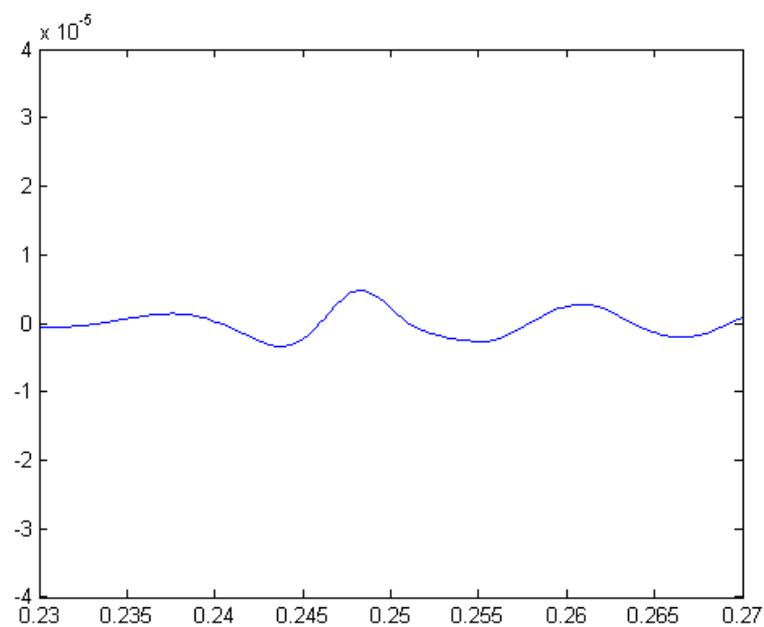


Figure 7: Zoomed-in view of D5

has a smaller signature than the larger step at $x = -0.75$.

- (6.2.5) Next we apply the wavelet method to a piece of data that has steps at $x = -1.5$, $x = -0.1548$, and $x = 0.5$, plotted in Figure 8. The step at $x = -1.5$ is 5 times smaller than the steps at $x = -0.1548$ and $x = 0.5$. We introduce a certain level of noise into the data and observe how well the wavelet method is suited to picking up these defects in the presence of noise (which would correspond to measurement noise in Airbus' application). We recover the spatial locations of these steps using 2-level decomposition with the Daubechies-2 wavelet in the absence of noise (not shown). Let h denote the height of the step at $x = -1.5$: our specification of noise level will be in terms of h , *i.e.* a noise level of $0.1h$ indicates that the data fluctuates with magnitude equal to 10% of h .

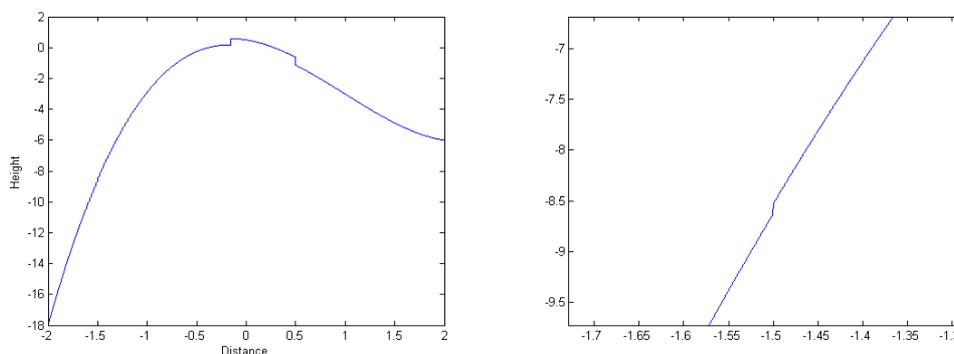


Figure 8: Artificial data to test robustness of wavelet method in presence of noise

- (6.2.6) We observed that with a noise level of $0.3h$, we cannot differentiate between the noise and the first step at $x = -1.5$ using 2-level Daubechies-2 wavelet decomposition, neither does increasing the number of levels (with Daubechies-2) nor using a higher Daubechies wavelet (with 2-levels) have any significant effect in picking out this step. Only when increasing the number of levels *and* using a higher Daubechies wavelet (6-level with Daubechies-6 decomposition) do we pick out the location of this step (Figure 9 at D6). If we increase the noise to $0.5h$ then the step cannot be differentiated from the noise even with the highest level of decomposition and the highest Daubechies wavelet available in Matlab.
- (6.2.7) To summarise
- The wavelet transform provides an alternative approach to analyse data containing various kinds of defects. It is able to pick out steps, jumps in gradient and jumps in curvature in the absence of noise.
 - The signature of the defects depends on their smoothness and their sizes. Steps and jumps in gradient can be picked out much more

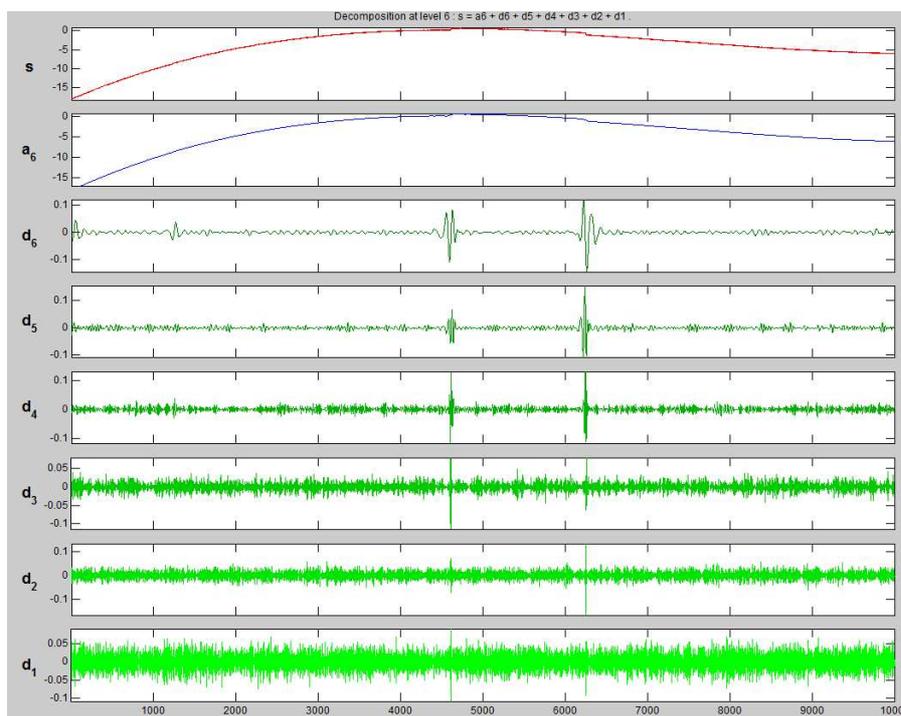


Figure 9: 6-level Daubechies-6 decomposition of data with $0.3h$ levels of noise

easily than jumps in curvature. Bigger steps can be picked out more easily than smaller steps.

- There is currently no robust method to identify all types of defect in an arbitrary piece of data. From these initial tests, the jumps in gradient will have a signature of order 10^{-5} , the jumps in gradient will have a signature of order 10^{-3} and the steps will have signature of order 10^{-1} .
- In the presence of sufficiently low levels of noise, the wavelet method is able to pick out the location of the steps.

We refer to [6], [7], [9] and [8] for a concise introduction to the theory and applications of wavelets. For surface defect detection by wavelet analysis, we refer the reader to [3].

6.3 Compressed sensing-style approach

- (6.3.1) Compressed sensing is based on the idea that a function can often be approximately expressed in terms of far fewer parameters than would be needed to specify the whole function, *i.e.* that the function has a representation that is *sparse* when expressed in the right form. In compressed sensing this idea is used to measure an image with fewer measurements than the full image, and one of the uses of this is to provide a compressed form of the image that will, for instance, use less bandwidth in transmission.

7 Radial Basis Functions

7.1 Transforming the Problem to Interpolation or Approximation

- (7.1.1) As pointed out by Wendland in the first chapter of his book *Scattered Data Approximation* [10], surface reconstruction, especially the *implicit* reconstruction of a surface through point cloud data, has been one of the main motivations driving the development of Radial Basis Functions (RBFs). In this section, we will first regard the original problem from Airbus as a problem of surface reconstruction from point cloud data, then transform it to an interpolation or approximation problem, and finally discuss how the transformed problem can be solved by RBFs.
- (7.1.2) The description of a surface can be either *explicit* or *implicit*. An explicit (*i.e.* parameterized) surface can be represented as the graph of a function $f : \Omega \rightarrow \mathbb{R}$ defined on some region $\Omega \subseteq \mathbb{R}^d$, where d is usually 2. In our problem, if we divide the wing surface into different patches, then on each patch we want to find a function f (*i.e.* a parameterization) to match the given point cloud data. Thus the original problem reduces to a 2-D interpolation or approximation problem on an irregular mesh: given data sites $\{(x_j, y_j)\} \in \mathbb{R}^2, j = 1, 2, \dots, N$ (which are the x, y -coordinates of the data points), and the corresponding data values $\{z_j\} \in \mathbb{R}, j = 1, 2, \dots, N$ (which are the z -coordinates of the data points), we want to find a function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ which interpolates the data: $f(x_j, y_j) = z_j, j = 1, 2, \dots, N$, or approximates the data: $f(x_j, y_j) \approx z_j$. The latter case is particularly important if the data contain noise, and we could specify a certain norm or semi-norm to specify how the approximation is to be quantified.
- (7.1.3) Before introducing the concept of RBFs and how they deal with the above problem, we shall also formulate the problem of *implicit* reconstruction of a orientable surface. Instead of dividing the surface into several patches and finding a parameterization for each patch of the surface, the implicit approach tries to describe the surface \mathcal{S} as the zero-level set of a single function F , *i.e.* $\mathcal{S} = \{x \in \Omega : F(x) = 0\}$. The function F could give a measure of how far away a point is from the surface, and changing F describes the deformation of the surface. Also, the way to generate meshes from this implicit representation has been well discussed. (See [11], [12], and the related software found on their website.)
- (7.1.4) Returning to our problem, what we have is the point cloud data $\mathbf{x}_j \in \mathbb{R}^3, j = 1, 2, \dots, N$, and what we want is an approximate function $s : \mathbb{R}^3 \rightarrow \mathbb{R}$ satisfying $s(\mathbf{x}_j) = 0$ for all \mathbf{x}_j . Obviously this condition alone does not suffice to determine an accurate approximation to the surface, since the zero function satisfies them. One common remedy for this problem is to add additional *off-surface* points; and one way to do this is to compute

the approximate surface normal at each point from a few of its nearest neighbours, and then for each point and its (approximated) surface normal, add two off-surface points, one outside the surface, and the other inside the surface, and then specify the s -values at these two off-surface points to be the signed distance function (positive outside the surface and negative inside).

- (7.1.5) This idea of computing and orienting surface normals goes back to Hoppe [13], [14], and has been successfully combined with RBFs to deal with implicit surface reconstruction problems, see [15], [16].
- (7.1.6) After adding those off-surface points, we denote the set of all these data points by $\mathbf{y}_j \in \mathbb{R}^3, j = 1, 2, \dots, 3N$, and the problem is to find $s : \mathbb{R}^3 \rightarrow \mathbb{R}$ satisfying $s(\mathbf{y}_j) = d_j$ for all \mathbf{y}_j , where d_j is the corresponding signed distance, positive outside the surface, negative inside, and zero on the surface. In other words, the implicit reconstruction of the surface also turns out to be an interpolation or approximation problem through point cloud data, except that these scattered points lie in 3-D.
- (7.1.7) Both the explicit and implicit reconstruction problems can be dealt with by radial basis functions in the same framework since they each reduce to interpolation or approximation problems through scattered data points.

7.2 Solving the Interpolation Problem by RBFs

- (7.2.1) A radial basis function is a function only dependent on the distance from a certain central point, *i.e.* of the form $\phi(\|\mathbf{x} - \mathbf{c}\|)$; here $\mathbf{c} \in \mathbb{R}^d$ is a known central point, $\mathbf{x} \in \mathbb{R}^d$ is the variable of the function, $\|\cdot\|$ is the 2-norm in \mathbb{R}^d , and ϕ is a function from \mathbb{R} to \mathbb{R} . By the definition of the radial basis function, the univariate function $\phi : \mathbb{R} \rightarrow \mathbb{R}$ actually determines the multivariate basis function provided that we have specified the central point. Generally there are two types of radial basis functions, or two kinds of ϕ . One is compactly supported, which means ϕ is non-zero only in a closed interval and the resulting basis function is non-zero only on a closed ball in \mathbb{R}^d . An example would be

$$\phi(x) = (1 - x)_+ = \begin{cases} 1 - x & \text{for } 1 - x \geq 0; \\ 0 & \text{for } 1 - x < 0. \end{cases} \quad (11)$$

The other type of RBF has global support, *i.e.* ϕ could be non-zero on the whole of \mathbb{R} , for example (see [16]) the thin-plate spline $\phi(r) = r^2 \log(r)$, and the biharmonic splines $\phi(r) = r$. Both types of RBFs have their practical uses in surface reconstruction: [16] uses globally supported RBFs, while [15] uses compactly supported RBFs to reconstruct surfaces. As we shall see, compactly supported RBFs will result a sparse linear system which is desirable for fast solvers, and in the numerical tests in the

following subsection we shall use compactly supported RBFs; Figure 10 shows a few examples of compactly supported RBFs in 2D with different smoothness.

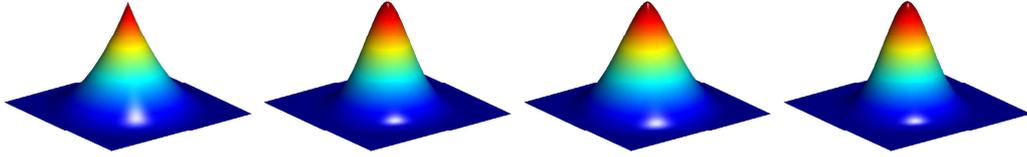


Figure 10: Compactly supported RBFs of different smoothness

- (7.2.2) Suppose we have chosen a suitable ϕ and thus the radial basis functions, and we want to interpolate a function at n data points $\mathbf{c}_j, j = 1, 2, \dots, n$ by a linear combination of n radial basis functions centred at these points. The resulting interpolated function thus becomes:

$$f(\mathbf{x}) = \sum_{j=1}^n w_j \phi(\|\mathbf{x} - \mathbf{c}_j\|), \quad (12)$$

where w_j is the weight of the radial basis function positioned at point \mathbf{c}_j . Then the question reduces to finding w_j , such that

$$f(\mathbf{c}_i) = \sum_{j=1}^n w_j \phi(\|\mathbf{c}_i - \mathbf{c}_j\|) = d_i, \quad i = 1, 2, \dots, n, \quad (13)$$

where d_i are the data values at points \mathbf{c}_i .

- (7.2.3) Putting the above equations into matrix form and defining $\phi_{ij} = \phi(\|\mathbf{c}_i - \mathbf{c}_j\|)$, we obtain the following linear system:

$$\begin{pmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1n} \\ \phi_{21} & \phi_{22} & \cdots & \phi_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{n1} & \phi_{n2} & \cdots & \phi_{nn} \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{pmatrix} \quad (14)$$

- (7.2.4) If we choose the basis functions wisely (*e.g.* see Wendland[10]), the matrix of the linear system (14) is symmetric and positive definite, and thus has a unique solution. Also, if ϕ is compactly-supported, the matrix is sparse and thus the equations can be solved effectively by direct sparse solvers or iterative solvers.

7.3 Numerical Example: Scattered Data on the unit square

- (7.3.1) In this subsection we give an example for the explicit reconstruction by RBFs, when the scattered data lie in 2-D. The 4 test functions (constructed by Shengxin Zhu) shown in Figure 11 are investigated by radial

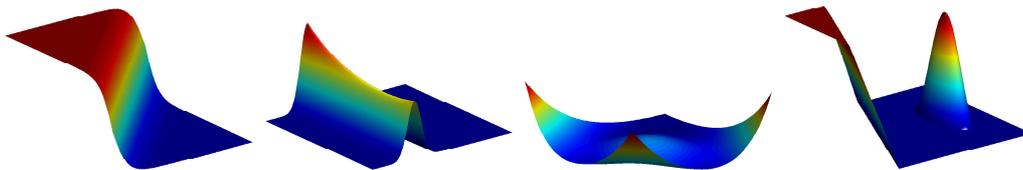


Figure 11: 4 test functions for interpolation from irregularly-spaced data values

basis function interpolation. Each test function could be regarded as a possible shape on a certain patch of the aircraft surface.

- (7.3.2) The fourth test function proves to be the hardest to recover, so we show the results only for that function. Figure 12 shows the numerical reconstruction of test function 4 from 2000 scattered data points in the unit square and the pointwise error.

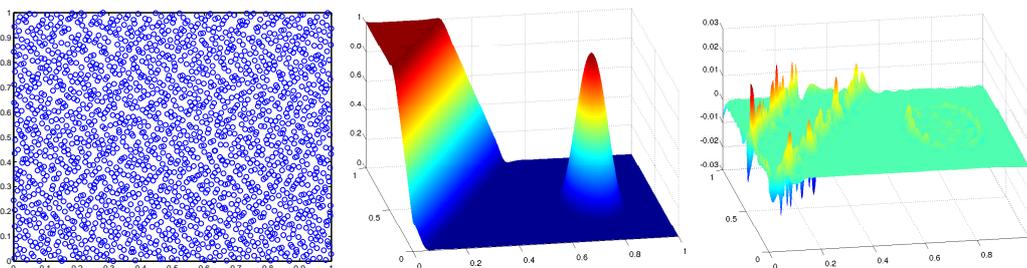


Figure 12: 2000 data points: reconstructed function; pointwise error

7.4 Possible Future Work

- (7.4.1) There are at least two possible improvements to the above approach. The first one is to reduce of the number of radial basis functions. Since each radial basis function corresponds to exactly one centred point, the total number of basis functions we have used above is equal to the number of data points given. However, in practice we would wish to represent the surface by fewer basis functions, in order to have an efficient description of the surface features. In [16, section 5], a greedy algorithm is introduced to reduce the number of RBF centres. The algorithm iteratively chooses a subset from the interpolation centres and computes RBFs to fit the data of that subset until the desired fitting accuracy is achieved.
- (7.4.2) Another issue is to deal with the noisy data from real measurement. As mentioned in the beginning of the section, if the data contains noise we might want to find a function to *approximate* instead of *interpolating* the data. In general the approximation should both be close to the original data and have some smoothness. Carr ([16] and [17]) suggests two ways

to deal with noise. One is to look for a function s minimizing some semi-norm, *i.e.* to solve the following problem:

$$\min_s \quad \rho \|s(\mathbf{x})\|_{H^2}^2 + \frac{1}{N} \sum_{i=1}^N (s(\mathbf{x}_i) - d_i)^2, \quad (15)$$

where in 2D, $\|s(\mathbf{x})\|_{H^2}^2 = \int_{\mathbb{R}^2} (\frac{\partial^2 s}{\partial x^2})^2 + (\frac{\partial^2 s}{\partial y^2})^2 + 2(\frac{\partial^2 s}{\partial x \partial y})^2 d\mathbf{x}$, and ρ is a regularization parameter balancing how smooth the approximation function is and how close it is to the data.

(7.4.3) Another approach they have used is low-pass filtering by convolution.

(7.4.4) To conclude, radial basis functions provide a well-investigated approach to deal with surface reconstruction problems, and might be useful here for our problem.

8 Other approaches

8.1 Global geometric invariants

(8.1.1) *Global* geometric invariants of a surface, such as its area or total curvature, characterise it in ways that are different from those that arise from studying its *local* properties. Some of these global invariants arise from studying the spectrum of the Laplace-Beltrami operator Δ_g on the surface. If it has eigenvalues

$$\lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \dots \quad (16)$$

and if ξ_j is a complete set of corresponding normalized eigenfunctions, so $\Delta_g \xi_j = \lambda_j \xi_j$, then the fundamental kernel (Green's function) of the heat equation on the surface is

$$K(t, x, y) = \sum_0^{\infty} \exp(-\lambda_j t) \xi_j(x) \xi_j(y), \quad (17)$$

and the trace of this is

$$Z(t) = \int K(t, x, x) dx \sim (4\pi t)^{-1} \sum_0^{\infty} a_k t^k, \quad (18)$$

and these coefficients a_k are a sequence of geometric invariants of the surface. For instance, a_0 will be the area, and a_1 is related to the curvature.

(8.1.2) If these invariants are known for the ideal form of a patch of the surface, then calculating these invariants for the actual produced surface would give new measures of the departure of the actual surface from the ideal. These measures of departure would be quite different from those studied

elsewhere in this report, in that they are not looking at the local structure but at its global properties over an area. It is possible therefore that they would detect deviations of different kinds from those that could be detected by the other methods in the report.

8.2 Defect signature approach

- (8.2.1) Our aim here is to give a digital description of possible defects on a surface, and we shall do this in the 1-D case. So we assume that we are given the departures v_i of the actual shape from the ideal shape at an ordered sequence of positions x_i , as illustrated in Figure 13. From these we con-

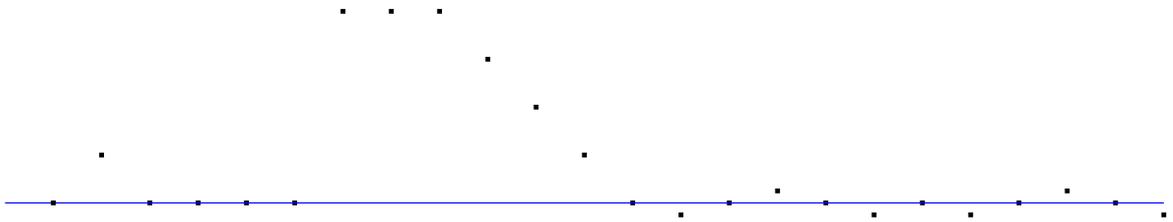


Figure 13: Schematic of deviations of surface from ideal, in one dimension.

struct finite difference approximations v'_i and v''_i to the first and second derivatives of v at x_i ,

$$v'_i = \frac{1}{2} \left(\frac{v_{i+1} - v_i}{x_{i+1} - x_i} + \frac{v_i - v_{i-1}}{x_i - x_{i-1}} \right), \quad v''_i = \frac{1}{2} \left(\frac{v'_{i+1} - v'_i}{x_{i+1} - x_i} + \frac{v'_i - v'_{i-1}}{x_i - x_{i-1}} \right). \quad (19)$$

- (8.2.2) If there were no defect, then we would expect that v would be subject to only measurement noise, and so we could hypothesize that v is distributed as a Gaussian with mean 0 and standard deviation σ , $v \sim G(0, \sigma^2)$. Then v' and v'' will also be Gaussian, with mean 0, and variances σ_1^2 , σ_2^2 . Suppose we choose suitable estimates of σ , σ_1 , and σ_2 , and divide v , v' , v'' by these to produce standardised versions that should be standard Gaussians. Of course, they will not be independent, but we indicate the general idea.

- (8.2.3) Then we can form defect measures such as

$$d_i = \sqrt{v_i^2 + v_i'^2 + v_i''^2}, \quad d_i = v_i^2 + v_i'^2 + v_i''^2, \quad d_i = |v_i| + |v_i'| + |v_i''|, \quad (20)$$

and we can let F_d be the cumulative distribution of d . Then large values of d tend to imply the presence of a defect, and so for a statistical way of estimating defects we could say that if $F_d(d_i) \geq 1 - \alpha$ then the point at index i belongs to a defect, at significance level $1 - \alpha$.

- (8.2.4) Then in any region of x -space, we can form various characteristics of the shape. For instance we might form the area deviation $a = \sum v_i$, the

variation $var = \sum |v_i|$, the maximal slope $s = \max |v'_i|$, and the maximal curvature $c = \max |v''_i|$. But there could be many other characteristics also formed from the data. Then for a defect we would form the vector (a, var, s, c) and then convert that into a vector of \pm signs: $+$ if the null hypothesis is rejected, $-$ if it is confirmed/accepted.

- (8.2.5) Some examples are shown in Figure 14, along with the kind of defect signature they might generate.

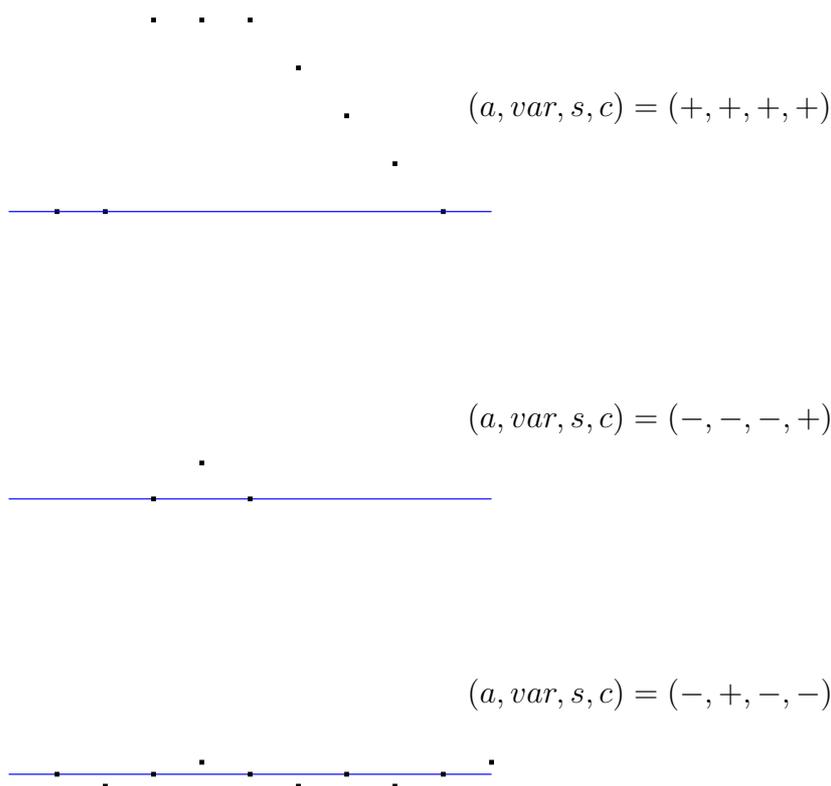


Figure 14: Schematic of deviations and the kind of defect signature they might generate.

- (8.2.6) In two dimensions, it would be necessary to consider other measures naturally, appropriate to 2 dimensions, but the same kind of methods would be used, and would again lead to a digital signature of each defect.

8.3 Two-stage method

- (8.3.1) On the small length-scale, what is wanted is the Fourier transform of the disturbances. So, over the area of $400\text{ mm} \times 400\text{ mm}$ that is covered by a single snapshot, there is enough information to extract the small length-scale information, and this can be done without combining that snapshot with any others.

- (8.3.2) One way of doing this would be to apply an interpolation procedure to the irregularly spaced data over perhaps a 50 mm or 100 mm square, to extract the high wavenumber components, which are of interest for laminar flow control. Then, discarding those high wavenumber components, we have the information that is needed for detecting waviness. Since we have an interpolant, we have a smooth function for the low wavenumber component, and when we come to assemble these over a larger area for the wave-extraction problem, we can evaluate these smooth interpolants at a suitable regularly-spaced grid of points, to make the wave-extraction problem easier.

9 Conclusions

We have outlined some of the possible approaches to this problem from the mathematical point of view. In particular, we have given some detail of the methods of projection onto suitable basis functions, and of wavelet methods, and of radial basis function methods. Further, we have suggested a method that could lead to a digital signature of each defect that could be used to classify it.

Bibliography

- [1] Efficient Geometrical Description of Perturbations to Design Shapes. Airbus Operations Ltd, Problem Statement. Available via <https://connect.innovateuk.org/web/study-group/esgi85/> Airbus, Problem brief.
- [2] Presentation Slides, 16-Apr-2012. Available via <https://connect.innovateuk.org/web/study-group/esgi85> Airbus, Problem presentation (slides).
- [3] Surface description and defect detection by wavelet analysis. Lars Rosenboom, Thomas Kreis, Werner Jüptner. *Measurement Science and Technology* **22** 045102 (2011).
- [4] Wavelets on irregular point sets. Ingrid Daubechies, Igor Guskov, Peter Schröder, Wim Sweldens. *Phil.Trans.Roy.Sec.A* **357**,2397–2413 (1999).
- [5] Clean Sky Joint Undertaking. Topic Nr. JTI-CS-2010-5-SFWA-03-005. Local deformation measurements on the outer wing in flight.
- [6] L. Chun-Lin. A tutorial of the wavelet transform (2010).
- [7] R.Polikar. The wavelet tutorial (1996).
- [8] M.Misiti, Y.Misiti, G.Oppenheim, J.Poggi. Wavelet toolbox for the use of MATLAB, User's Guide Version 1.

-
- [9] A.Graps. An introduction to wavelets. Computational science and Engineering, IEEE, 2(2):50–61,(1995).
- [10] Wendland, Holger. Scattered data approximation. Cambridge Monographs on Applied and Computational Mathematics, 17. Cambridge University Press, Cambridge, 2005. x+336 pp. ISBN: 978-0521-84335-5; 0-521-84335-9 MR2131724 (2006i:41002)
- [11] T.K.Dey and J.A.Levine. Delaunay meshing of isosurfaces. Proc. Shape Modeling International (2007), 241–250.
- [12] T.K.Dey, G.Li and T.Ray. Polygonal surface remeshing with Delaunay refinement. Proc. 14th Intl. Meshing Roundtable., 2005, 343–361.
- [13] H.Hoppe. Surface Reconstruction from Unorganized Points. Ph.D. thesis, University of Washington, 1994.
- [14] H.Hoppe, T.DeRose, T.Duchamp, J.McDonald, and W.Stuetzle. Surface reconstruction from unorganized points. In Computer Graphics, Proc SIGGRAPH’92, 26:71–78, 1992
- [15] B.Morse, T.S.Yoo, P.Rheingans, D.T.Chen, and K.R.Subramanian. Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions. In 2001 International Conf. on Shape Modeling and Applications (SMI 2001), Los Alamitos, CA,89–98. IEEE Computer Society Press, 2001.
- [16] J.C.Carr, R.K.Beatson, J.B.Cherrie, T.J.Mitchell, W.R.Fright, B.C.McCallum, and T.R.Evans. Reconstruction and representation of 3D objects with radial basis functions. In Computer Graphics Proceedings, Annual Conference Series, pp.67–76. Addison Wesley, 2001.
- [17] J.C.Carr, R.K.Beatson, B.C.McCallum, W.R.Fright, T.J.McLennan, and T.J.Mitchell. Smooth surface reconstruction from noisy range data. In Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia (Graphite 2003), pages 119–126, 2003.