MASTER

Research of projected implicit reconstruction in optimisation

van Etten, A.J.A.

Award date:
1996

Link to publication
Report of a graduation project carried out in the group Dynamics at the Eindhoven University of Technology. This project is part of a collaboration with Sowerby Research Centre at British Aerospace in Bristol (Great Britain).

Supervisors : Dr. ir. A.J.G. Schoofs  
             : Ir. M.H. van Houten

Professor : Prof. dr. ir. D.H. van Campen
Abstract

The analysis of aerodynamic designs with Computational Fluid Dynamic (CFD) tools is very time consuming. With the introduction of the Projected Implicit Reconstruction (PIR) method, fast approximation of the design space became possible. The PIR method needs one CFD analysis to evaluate another design. Perturbations are made in one or more design variables from the CFD analysis until the required design site is reached. Depending on the design variables which are variated, two to eight-hundred PIR steps can be made in one CFD run. The accuracy of the PIR results depends mainly on the total change in the design variables. A slight improvement can be made by making smaller perturbations.

An advantage of the PIR method is that results are also available for design points situated between the CFD analysis and the investigated design. Based on this characteristic of the PIR code, a new type of line search is developed for use in optimisation. This line search makes small steps along the line until the optimum on that line is reached. Convergence of the optimisation process is ensured with a 'standard' line search near the optimum.

The line search with small steps is tested in a Sequential Quadratic Programming routine in Matlab and in an optimisation routine in Fortran that uses penalty functions. The line search with small steps in combination with the PIR method is most effective with that last optimisation routine. This is explained by the fact that optimisation with exterior penalty functions allows intermediate solutions in the infeasible domain.

The PIR method can also be used to make function fits on the aerodynamic coefficients (the lift, drag and pitching moment coefficients, and area of an aerofoil). The accuracy of these functions fits depends on the function model and on the choice of experimental design. The experimental design determines which points in the design space should be analysed for the function fit. To ensure that the function fits are correct, an analysis of variance should be made. Finally, some function fitting methods are compared using CFD and PIR results from a 2-D aerofoil. The function fits on the lift and drag coefficient are smooth functions with little interaction between the two design variables (Mach number and angle of attack). Therefore, complex function models are not necessary and the choice of experimental design becomes less important.
6.6.1 Axial function fits ............................................. 31
6.6.2 Diagonal function fits ........................................... 33
6.6.3 Summary .......................................................... 34

7 Conclusions and recommendations 35

Bibliography 36

A Accuracy of axial function fits 37
B Accuracy of diagonal function fits 40
Nomenclature

The list below gives the used symbols and abbreviations in this report.

Latin symbols:

- $A_{ij}$: function model coefficients
- $a$: the speed of sound
- $B$: approximation to the Hessian of the Lagrangian function
- $C_d$: drag coefficient
- $C_l$: lift coefficient
- $D$: drag
- $F$: objective function
- $\mathbf{F}$: $x$-component inviscid flux vector
- $\mathbf{f}$: function components vector
- $f_i$: $i$-th function component
- $\mathbf{G}$: $y$-component inviscid flux vector
- $\mathbf{G}_i$: inequality constraint vector
- $H$: enthalpy
- $\mathbf{H}_j$: equality constraint vector
- $k$: iteration number
- $k$: number of function model parameters
- $L$: lift
- $L$: auxiliary objective function
- $\mathbf{l}$: line search vector
- $l$: order of the function model
- $M$: Mach number
- $M_p$: pitching moment
- $m_a$: number of active constraints
- $m_g$: number of inequality constraints
- $m_h$: number of equality constraints
- $N$: number of evaluated design points
- $n$: number of design variables
- $n_P$: number of PIR analyses
- $n_s$: number of small steps
- $p$: pressure
- $\mathbf{Q}$: conserved field variable vector
- $\mathbf{R}$: residual flux vector
- $r$: penalty value
- $\mathbf{s}$: search direction vector
- $t$: time
- $u$: $x$-component of velocity
- $v$: $y$-component of velocity
Velocity function components matrix
design vector
i-th design variable
axial grid coordinate
function calculation vector
function expectation vector
normal grid coordinate
function calculation
function model
complete vector of (x,y) grid points

Greek characters:
\( \alpha \) angle of attack
\( \alpha_i \) step size
\( \beta \) function parameters vector
\( \beta \) parameters estimate vector
\( \beta_i \) i-th function parameter
\( \delta \) change in
\( \delta \) step length
\( \delta_i \) parameter for the SQP method
\( \delta_j \) parameter for the SQP method
\( \Delta \) change in
\( \Delta \alpha \) size of small step
\( \gamma \) ratio of specific heats
\( \lambda \) Lagrange multipliers vector
\( \lambda_i \) j-th element of the Lagrange multipliers vector
\( \rho \) density
\( \phi \) auxiliary function

Other symbols:
\( \infty \) free stream conditions
\( \infty \) infinity
* steady state value

Abbreviations:
CFD Computational Fluid Dynamics
PIR Projected Implicit Reconstruction
SQP Sequential Quadratic Programming
SSE Error Sums of Squares
SSM Model Sums of Squares
SS_{yy} Total Sums of Squares
ANOVA Analysis of Variance
Chapter 1

Introduction

For the development of aerodynamic products a good approximation of the form and structure of the associated design space is essential. With the introduction of powerful computers the investigation of this design space has moved from using experimental tools towards the use of Computational Fluid Dynamics (CFD) tools.

A disadvantage of exploring the design space with CFD tools is the required computational time and associated cost. Topliss et al. (1996) developed an alternative approach to investigate the design space, named Projected Implicit Reconstruction (PIR). With the PIR method design points can be evaluated based on one CFD analysis in the design space. From this CFD analysis perturbations are made in one or more design variables until the required design point is reached. With the initial data of the CFD analysis the steady state flow field at that new design point is constructed. This steady state flow field determines the values of several aerodynamic coefficients (lift, drag, pitching moment, etc.). However, evaluation with the PIR method is not faultless. Therefore, careful use of the PIR method is necessary. Figure 1.1 shows the aerodynamic forces and moments on an aerofoil. $L$ is the lift force, $D$ the drag force and $M_p$ the pitching moment. The Mach number is determined by the free stream velocity $V$ and the speed of sound $a$: $M = V / a$. The angle of attack $\alpha$ is the angle between the chord line and the free stream velocity vector.

Figure 1.1: Aerodynamic forces and moments on an aerofoil.

Figure 1.2: Surface curve and shape variables for an aerofoil.
With the introduction of the PIR method, a great reduction in time can be made in the optimisation of aerodynamic designs. This constrained optimisation problem is defined as to find that design point $\mathbf{x}$ where the function

$$F(\mathbf{x})$$

has a minimum, subjected to the constraints:

$$G_i(\mathbf{x}) \leq \mathbf{c} \quad i = 1, \ldots, m_g$$

$$H_j(\mathbf{x}) = \mathbf{d} \quad j = 1, \ldots, m_h$$

Mostly $F$ is the drag coefficient, $H_j$ the lift coefficient, pitching moment and area of the aerofoil and $G_i$ the bounds on the design variables. The design variables are the free stream Mach number $M_{\infty}$, the angle of attack on the aerofoil $\alpha$ (Figure 1.1) and the shape variables. The shape variables control the surface curve of the aerofoil (with spline interpolation), see Figure 1.2.

The optimisation of aerodynamic designs can even be made faster if function fits can be constructed for the aerodynamic coefficients depending on the design variables. To analytically fit each aerodynamic coefficient to a function which covers the whole design space, PIR evaluations are made at specific design points.

In chapter two a summary is given of the PIR method. In chapter three a special characteristic of the PIR method is used to develop a new type of line search. Most of the optimisation routines consists of finding a suitable search direction and performing a line search along this direction. After finding the minimum on the line, the optimisation is restarted. The new line search is tested with a Sequential Quadratic Programming optimisation routine in chapter four and with an optimisation routine that uses penalty functions in chapter five. These tests are not made with a CFD code and the PIR method but simulated with a test function.

In chapter six the function fits are discussed. The important aspects of function fitting are which function to choose and fit this function, which design points have to be evaluated to fit the function on (experimental design), and the accuracy of the function fit on these design points. The accuracy of the function fits can be investigated with an analysis of variance. These aspects of function fitting are further investigated with CFD and PIR results from a 2-D aerofoil. Function fits are made for the lift and drag coefficients. Only the Mach number and angle of attack are changed, so the area of the aerofoil remains constant.
Chapter 2

Projected Implicit Reconstruction

Topliss et al. (1996) have developed a rapid design space approximation for two dimensional transonic aerofoil design, named Projected Implicit Reconstruction. First the aerodynamic equations of the aerofoil are discussed, followed by the equations concerning the PIR method.

2.1 Aerodynamic equations

The equations used for describing the isenthalpic, inviscid flow around an aerofoil are the unsteady two dimensional Euler equations along with the relevant boundary conditions on the aerofoil surface.

Cartesian space coordinates $x$ and $y$ are used to describe the domain. The local static density and pressure are represented by $\rho$ and $p$, the vector components of the local velocity by $u$ and $v$. The Euler equations are here

$$\frac{\partial Q}{\partial t} = R(Q),$$

where $Q$ is a vector of the conserved field variables $(\rho, \rho u, \rho v)^T$, $t$ is the time, and $R$ is the residual flux vector. In the steady state situation eq. 2.1 reduces to

$$R(Q^*) = \{0\},$$

where $^*$ denotes the steady state solution. The residual flux vector $R$ consists of gradients in the flux which are evaluated along $x$ and $y$. $R$ is written as

$$R(Q) = -\frac{\partial F(Q)}{\partial x} - \frac{\partial G(Q)}{\partial y}. \quad (2.3)$$

$F$ and $G$ are the inviscid fluxes which are composed of convective and pressure terms:

$$F = \begin{pmatrix} \rho u \\ \rho u^2 \\ \rho uv \end{pmatrix} + \begin{pmatrix} 0 \\ p \\ 0 \end{pmatrix} \quad (2.4)$$
and

\[ G = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ p \end{pmatrix}. \] (2.5)

Furthermore, \( p \) is known by using the ideal gas law:

\[ p = \left[ \frac{H_\infty - \frac{1}{2}(u^2 + v^2)}{\gamma \rho} \right] \] (2.6)

with \( \gamma \) the constant ratio of heats, and \( H_\infty \) the total enthalpy of the system.

Equations 2.1 and 2.6 are solved with Computational Fluid Dynamic (CFD) tools which provide the converged flow field for the state variables at a point in the design space. This design point is defined by the free stream Mach number \( M_\infty \), the angle of attack on the aerofoil \( \alpha \), and the shape variables (the position of the vertices which describes the sectional shape of the aerofoil), see Figures 1.1 and 1.2. This flow field is then used as starting solution in the Projected Implicit Reconstruction method.

### 2.2 Equations belonging to the PIR method

The PIR method is consistent with CFD methods because the same flow equations and boundary conditions are satisfied. However, the formulation of the equations of motion are different. In the PIR method, perturbations are made in the design variables to evaluate certain responses of the system. This is known as the ‘quasi-analytic’ method (Taylor III et al., 1992).

The considered design variables \( x \) of the transonic aerofoil (Figure 1.1) are the angle of attack, the Mach number and the shape variables which define the aerofoil surface. \( Z \) is the representation of the complete vector of \((z,y)\) grid coordinates.

Equation (2.2) can be expressed explicitly as a function of the mesh \( Z(x) \) and the design variables \( x \). In the steady state this relation is

\[ R(\mathbf{Q}^*(x), Z(x), x) = \{0\} \] (2.7)

The values for the components of the residual vector \( R \) can be calculated in each grid cell through knowledge of the discrete steady state flow field \( \mathbf{Q}^* \). Differentiation of eq. 2.7 to the design variables \( x \), leads to a linear system of algebraic equations

\[ - \left[ \frac{\partial R}{\partial \mathbf{Q}^*} \right] \left[ \frac{\partial \mathbf{Q}^*}{\partial x} \right] = \left[ \frac{\partial R}{\partial Z} \right] \left[ \frac{\partial Z}{\partial x} \right] + \left[ \frac{\partial R}{\partial x} \right]. \] (2.8)

The approach of the design space is based on this equation of the exact derivative of the discrete residual vector. This formulation is known as the ‘quasi-analytic’ method and holds for the flow in each grid cell. Therefore, eq. 2.8 describes a set of linear algebraic sensitivity equations. The formulation used in the PIR method is

\[ - \left[ \frac{\partial R}{\partial \mathbf{Q}^*} \right] [\Delta \mathbf{Q}^*] = \left[ \frac{\partial R}{\partial Z} \right] [\Delta Z] + [\Delta R]. \] (2.9)
A perturbation in the design vector $x$ produces the changes $\Delta Q^*$, $\Delta Z$ and $\Delta R$. Eq. 2.9 is used instead of eq. 2.8 because some exact representations are known and these can be more accurate than using finite difference or the differential forms.

The linear eq. 2.9 is solved for values of matrix $[\Delta Q^*]$. The flow field can be constructed at the new design point if these variations are known. The changes of the flow field due to the perturbation of the design vector $\delta x$ are combined with the steady state flow field at the design point $x$, to produce a new flow field at the design point $(x + \delta x)$:

$$Q^*(x + \delta x) = Q^*(x) + \{\Delta Q^*\}. \tag{2.10}$$

To conclude, one CFD analysis provides the converged flow field solution for a point in the design space. To calculate other points in the design space, perturbations in one or more values of the design variables are made. The PIR method iteratively reconstructs the flow field as it marches towards the required position.

The PIR method shows a good level of agreement with CFD tools. When angle of attack or Mach number are changed, PIR runs around 800 steps in the time taken to perform one CFD run, and around 200 PIR steps for changes in the shape variables. If larger perturbations are made, the accuracy of the PIR method decreases, especially for large perturbations in the shape variables. Furthermore, the PIR method is less accurate for changes in the Mach number $M_\infty$ than for changes in the angle of attack $\alpha$. 
Chapter 3

Line search with small steps

An unconstrained optimisation problem is defined as to find that design point $\mathbf{x}$ where the function

$$F(\mathbf{x})$$

has a minimum. General optimisation techniques consist of finding a suitable search direction $\mathbf{s}$ and performing a line search along this search direction. For the $k$-th iteration the line search is formulated as:

$$l(\alpha) = \mathbf{x}^{(k)} + \alpha \mathbf{s}^{(k)}.$$  

The value $\alpha$ has to be found at which $F(l(\alpha))$ has a minimum. Usually, the line search is performed by applying a safeguarded polynomial interpolation, a golden section search or a combination of these. For instance the golden section method (Vanderplaats, 1984) uses lower and upper bounds on $\mathbf{x}$: $\mathbf{x}_l$ and $\mathbf{x}_u$, see Figure 3.1, where $\mathbf{x}_i = \mathbf{x}^{(k)} + \alpha_i \mathbf{s}^{(k)}$.

![Figure 3.1: Line search using the golden section method, o are the first five points evaluated during the golden section search.](image1)

![Figure 3.2: Evaluation of $\mathbf{x}_4$ using the PIR method, $\Delta$ are the points evaluated with the PIR method.](image2)

10
\( \alpha_l \) and \( \alpha_u \) are known, \( \alpha_1 \) and \( \alpha_2 \) have to comply to:

\[
\begin{align*}
\alpha_u - \alpha_2 &= \alpha_1 - \alpha_l \\
\frac{\alpha_1 - \alpha_l}{\alpha_u - \alpha_l} &= \frac{\alpha_2 - \alpha_1}{\alpha_u - \alpha_1}
\end{align*}
\tag{3.3}
\]

After evaluation of \( x_1 \) and \( x_2 \) the new lower and upper bounds (\( x_l = x_1, x_u = x_u \)) and a point in this interval (\( x_2 \)) are known. \( \alpha_3 \) is calculated with eq. 3.3 and 3.4. Again new lower and upper bounds (\( x_l = x_l, x_u = x_u \)) can be made. This procedure is repeated until the minimum is found: \( x^{(k+1)} = x^{(k)} + \alpha^{(k)} s^{(k)} \).

If the PIR method is used to evaluate, for instance \( x_4 \) (see Figure 3.2), small perturbations are made in the design variables. While making these steps the minimum will be crossed at \( x_3 \), and the minimum found with only making the steps is \( x_2 \). This becomes the line search with small steps: from a design point \( x^{(k)} \), \( n_s \) steps \( \Delta \alpha \) are taken along the search direction \( s^{(k)} \):

\[
x_{n_s} = x^{(k)} + n_s \Delta \alpha s^{(k)}, \quad n_s = 1, 2, 3, \ldots
\tag{3.5}
\]

until the objective function \( F \) no longer decreases:

\[
F(x_{n_s}) > F(x_{n_s-1}).
\tag{3.6}
\]

The new design point becomes

\[
x^{(k+1)} = x^{(k)} + (n_s - 1) \Delta \alpha s^{(k)}.
\tag{3.7}
\]

The size of the step length \( |\delta| = |\Delta \alpha s^{(k)}| \) is taken constant during the entire optimisation process. This step size cannot be chosen too large, because the PIR method is less accurate for larger perturbations (Topliss et al., 1996). On the other hand, the more evaluations with the PIR method, the larger the error. So there is also a limit, \( n_{s,\text{max}} \), on the number of small steps \( \Delta \alpha \) necessary. For instance, \( n_s \) can exceed \( n_{s,\text{max}} \) if \( F \) is almost constant along \( s^{(k)} \).

Another problem arises if the minimum \( x^{(k+1)} \) is very close to the start point of the line search \( x^{(k)} \), which certainly will occur if \( x^{(k)} \) is close to the global optimum. The small step \( \Delta \alpha \) is often too large to guarantee the stop conditions in the optimisation routine because one of the stop conditions is always the change in the design variables. To solve this problem, the standard line search is used if the number of small steps \( n_s \) to cross the minimum on the line is less than three. Information derived from the line search with small steps is useful for the standard line search: the minimum on the line is situated between the starting point \( x^{(k)} \) and \( x^{(k)} + n_s \Delta s^{(k)} \).

This type of line search can deal with constraints just as other line searches. One of the methods used to perform an optimisation with constraints is the method of Lagrange multipliers (Haftka et al., 1992). An optimisation problem with an objective function plus \( m_a \) active constraints is equivalent to an unconstrained problem with an auxiliary function:

\[
L(x, \lambda) = F(x) + \sum_{i=1}^{m_a} \lambda_i G_i,
\tag{3.8}
\]

where \( \lambda_i \) are the Lagrange multipliers and \( G_i \) the constraints. The Sequential Quadratic Programming method, discussed in chapter four, is a projected Lagrangian method. Another
method to deal with constraints, is the use of penalty functions, this is described in chapter five.

The total optimisation including the line search with small steps, is illustrated schematically in Figure 3.3. If the number of small steps $n_s$ is less than three the standard line search will be performed from the same design point $x(k)$. $n_s$ is set to zero before each line search. Figure 3.3 is a very global description, the calculation of the search direction $s(k)$, does not depend only on the design point, but can also depend on the last search direction $s(k-1)$ and the constraints. However, calculation of the search direction is different for each optimisation routine and therefore not shown in Figure 3.3.
Chapter 4

Modification of the SQP optimisation routine

The Sequential Quadratic Programming (SQP) method (Powell, 1987 and Haftka et al., 1992) is one of the most robust methods for optimisation of constrained functions. The objective function (eq. 1.1) is approximated with a quadratic function, the constraints (eq. 1.2 and 1.3) with a linear function. At the k-th iteration step, the search direction \( s^{(k)} \) is determined by solving the following quadratic programming problem:

\[
\phi(s) = F(x) + \nabla F(x) + \frac{1}{2} s^T B(x, \lambda) s,
\]

with

\[
\nabla G_i(x) s + \delta_i G_i \leq c \quad \text{(4.2)}
\]

\[
\nabla H_j(x) s + \delta_j H_j = d \quad \text{(4.3)}
\]

where matrix \( B \) is a positive definite approximation to the Hessian of the Lagrangian function, parameters \( \delta_i, \delta_j \approx 0.9 \div 0.95 \) are used to avoid inconsistence in the linearised constraints.

4.1 Implementation of line search with small steps in a SQP optimisation routine of Matlab

The SQP method is used in the Matlab routine CONSTR.M (Grace, 1990). The standard line search in CONSTR.M uses a merit function (Han, 1977). In this line search not the objective function is minimised, but a function like eq. 3.8. This standard line search has been modified to make comparison with the line search with small steps possible: the first step in the line search has a step size \( |\alpha s^{(k)}| \) of one. This modification prevents a big step size if \( |s^k| \) is very large. The line search with small steps (eq. 3.5 and 3.6) is introduced into the SQP routine, see Chapter 3.

As a test function, Rosenbrocks function

\[
F = 100(x_2 - x_1)^2 + (1 - x_1)^2
\]

(4.4)
Modification of the SQP optimisation routine is used. To give an estimate of the reduction in CPU time of the optimisation with PIR against the optimisation with CFD tools, evaluation of the function with PIR is assumed to be 100 times faster than with CFD. However, in the above function it is not possible to speak of CFD or PIR evaluations. Therefore, slow, exact (CFD) evaluations are compared with fast, not exact (PIR) ones. The reduction is defined as

$$\text{Reduction} = \frac{\# \text{slow}_\text{standard} - (\# \text{slow} + 0.01 \times \# \text{fast}_\text{small steps})}{\# \text{slow}_\text{standard}} \times 100\%.$$  \hspace{1cm} (4.5)

The standard line search only contains slow evaluations, \# slow standard denotes the total number of evaluations to reach the optimum. The line search with small steps consist of slow and fast evaluations, \# slow small steps and \# fast small steps denotes the number of slow and fast evaluations to reach the optimum.

4.2 Comparison of standard line search and line search with small steps

The test function (eq. 4.4) is minimised from start point \(x_0 = (-2,1)\). There are no constraints. The assumption is made that the derivatives are not available and must be calculated with a finite difference approximation. At first the error made with fast evaluations is not taken into account. The followed search path with the standard line search and with the line search with small steps (step size \(|\delta| = 0.05\)) are plotted in Figure 4.1 and 4.2.

Two methods can be applied in the optimisation routine:

- There is only one slow, exact analysis made (method A)
- After each line search a new slow, exact analysis is made (method B)
Modification of the SQP optimisation routine

Method A naturally gives more reduction, however if method B is used a more accurate solution can be expected. The optimisation using the standard line search reaches the optimum (1,1) with 148 slow function evaluations. The optimisation with the line search with small steps requires for method A one slow and 191 fast function evaluations, method B used 23 slow and 191 fast calculations. This is no coincidence, because there is no error on the fast evaluations.

4.3 Influence of the error

When the fast (PIR) method is used an error is made on the function evaluation. To investigate the effect of such an error on the optimisation process, the test function (eq. 4.4) is expanded with an error:

\[ F_{\text{error}} = F + (\text{dist} \times \text{err}). \]  

In \( F_{\text{error}} \), the error size \( \text{dist} \times \text{err} \) depends on the distance from the last, exact calculation \( \text{dist} \) and on an error factor \( \text{err} \), \( \text{err} \) is chosen so that the \( \text{dist} \times \text{err} \) has the same dimension as \( F \).

![Figure 4.3: no error, err = 0: Reduction depending on the step size \( \delta \) and the method (A and B).](image1)

![Figure 4.4: err = 0.05: Reduction depending on the step size \( \delta \) and the method (A, B and C).](image2)

The reduction in the optimisation cost, using fast evaluations and the line search with small steps, is plotted in Figure 4.3 and 4.4 for several step sizes \( \delta \). Both have starting point \( x_0 = (-2, 1) \), in Figure 4.3 there is no error in the fast calculations, while in Figure 4.4 the error factor \( \text{err} \) is equal to 0.05. An extra method C is introduced to increase the accuracy of method A: after reaching the optimum a new CFD calculation is made in that optimum and the optimisation routine is restarted from the optimum design point.

For several starting points the reduction is calculated for step sizes 0.02 and 0.05 with error factor 0.05. The reduction achieved by methods A, B and C are shown in Table 4.1. Method B gives lesser reduction than the other methods, as expected. Furthermore, the reduction with method B is more dependent on the step size, see also Figure 4.4. This is
Modification of the SQP optimisation routine

Starting point | Step size | Method A | Method B | Method C
---|---|---|---|---
(-2, 1) | 0.02 | 97.28 | 80.23 | 96.34
 | 0.05 | 98.07 | 82.35 | 97.09
(-1, -1) | 0.02 | 96.81 | 81.99 | 93.16
 | 0.05 | 97.43 | 77.55 | 93.76
(2, -1) | 0.02 | 93.93 | 65.39 | 91.43
 | 0.05 | 95.09 | 63.20 | 89.94
(0, 2) | 0.02 | 93.93 | 65.39 | 91.43
 | 0.05 | 95.09 | 63.20 | 89.94

Table 4.1: Reduction (%) depending on starting point, step size and method. Error factor is 0.05, no constraints.

explained by the fact that a different step size leads to a slightly different search path, which also variates the number of line searches. And with method B an exact, but slow calculation is made before each line search. The difference in reduction between method B and C strongly depends on the starting point. All three methods find the optimum round (1,1). However, method A should not be used because of its inaccurate calculation of the objective function.

4.4 Constraints

Constraints are added to the optimisation problem to investigate their influence on the optimisation process. The effect of the line search with small steps is investigated for inequality and equality constraints.

4.4.1 Inequality constraints

As an example Rosenbrocks function (eq. 4.4) had to comply to the following inequality constraint:

\[ x_2 \geq 0.5 \] (4.7)

From several starting points this optimisation problem is solved. Optimisation with the standard line search found two optima. A local optimum at \((x_1, x_2) = (0.70, 0.5)\) and the global optimum at \((1,1)\). Optimisation with the line search with small steps also found these two optima. If the optimisation is started from a feasible solution, the line search with small steps found the same (local or global) optimum as the standard line search. Table 4.2 shows the reduction for optimisation with the equality constraint for two feasible starting points. Methods B and C are used with an error factor of 0.05. However, if the starting point is infeasible the optimisation with the line search with small steps can converge to the other optimum than optimisation with the standard line search. Whether or not it converges to the other optimum depends on the step size, because the step size controls the followed search path. It also depends on the maximum number of steps \(n_{s,max}\) (Chapter 3), this only has an effect on the search path if the number of steps \(n_s\) exceeds \(n_{s,max}\). The search path of the
Modification of the SQP optimisation routine

optimisation with the inequality constraint is plotted in Figure 4.5 from starting points (-2,1) and (-1,-1), step size 0.05 and error factor 0.05.

<table>
<thead>
<tr>
<th>Starting point</th>
<th>Step size</th>
<th>Method B</th>
<th>Method C</th>
<th>Optimum point</th>
</tr>
</thead>
<tbody>
<tr>
<td>(-2, 1)</td>
<td>0.01</td>
<td>61.62</td>
<td>84.28</td>
<td>(-0.70,0.5)</td>
</tr>
<tr>
<td></td>
<td>0.02</td>
<td>67.67</td>
<td>84.90</td>
<td>(-0.70,0.5)</td>
</tr>
<tr>
<td></td>
<td>0.03</td>
<td>61.45</td>
<td>87.76</td>
<td>(-0.70,0.5)</td>
</tr>
<tr>
<td>(0, 2)</td>
<td>0.01</td>
<td>91.40</td>
<td>94.18</td>
<td>(1,1)</td>
</tr>
<tr>
<td></td>
<td>0.02</td>
<td>92.22</td>
<td>95.32</td>
<td>(1,1)</td>
</tr>
<tr>
<td></td>
<td>0.03</td>
<td>93.90</td>
<td>93.90</td>
<td>(1,1)</td>
</tr>
</tbody>
</table>

Table 4.2: Reduction (%) depending on starting point, step size and method. Error factor is 0.05 and with inequality constraint.

Figure 4.5: Followed search path from (-2,1) and (-1,-1) due to inequality constraint, with step size \( \delta = 0.05 \).

Figure 4.6: Followed search path from (-2,1) and (2,-1) due to equality constraint, with step size \( \delta = 0.05 \).

4.4.2 Equality constraints

Again Rosenbrocks function (eq. 4.1) is used. The inequality constraint is replaced by a equality constraint:

\[
x_2 = x_1^2 + x_1 - 0.5
\]  

(4.8)

The followed search path for starting points (-2,1) and (2,-1) and error factor zero is plotted in Figure 4.6. The search path is completely determined by the constraint. The optimum is found at (0.51,0.26) for both starting points.

The line search with small steps does not find a better point on the line due to the constraint. This is explained by the fact that the influence of the constraint is bigger than the influence of the objective function (eq. 3.8). If the standard line search is used, with an initial step size equal to the small step, the minimum on the line is found very close to the starting point, see Figure 4.6. The influence of the constraint can be decreased by enlarging
the step size, hence the change in the objective function can increase. So if there are equality constraints present in the optimisation problem it is advised to perform line searches with the standard line search and step size one or bigger. This step size $|\alpha^k|$ should be controlled so that every point can be evaluated with a limited number of PIR evaluations.

4.5 Summary

Optimisation with PIR evaluations and line search with small steps, can lead to a reduction in CPU time for Sequential Quadratic Programming compared with the standard line search. The error made with PIR is not negligible, however its influence can be minimised if after each line search a new CFD analysis is made or if after a complete optimisation run at the found optimum design point a new CFD analysis is made and the optimisation from this point is restarted. If inequality constraints are present, optimisation with the line search with small steps can converge to a local optimum just as optimisation with the standard line search. If there are equality constraints, especially non-linear, the line search with small steps is not effective. In this case the standard line search should be applied.
Chapter 5

Optimisation using penalty functions

Besides the SQP method there are more methods which deal with constrained optimisation. One of those is the Penalty Function Method (Haftka et al., 1992). There are two sorts of penalty function methods: exterior and interior. The exterior penalty function method only applies penalties in the infeasible area of the optimisation domain, this has the effect that the design moves into the infeasible domain. The interior penalty function method produces only feasible designs. The disadvantage of this is that the interior penalty function methods needs a feasible starting point and that it can not deal with equality constraints.

In an exterior penalty function method, the penalty is mostly proportional to the square of a violation. So in the optimisation problem the objective function (eq. 1.1) and constraints (eq. 1.2 and 1.3) are replaced by:

\[ \phi(x, r) = F(x) + r \sum_{j=1}^{m_h} H_j^2(x) + r \sum_{i=1}^{m_a} G_i^2(x) \]  

(5.1)

with

\[ r = r_1, r_2, \ldots, r_j \rightarrow \infty \]  

(5.2)

The inequality constraints \( G \) are treated differently from the equality constraints \( H \). Penalties are only applied for the \( m_a \) active inequality constraints.

5.1 Implementation of line search with small steps in a penalty function method in Fortran

The optimisation routine MINFUN (Dijkstra, 1989) is written for Fortran 77 and uses an exterior penalty function method to find the optimum. The search direction is determined by the variable-metric method with a updating formula by Broyden, Fletcher and Shanno, which is also described by Gill et al. (1981).

The standard line search in this routine uses polynomial interpolation. The line search with small steps is added to the optimisation routine and its influence is investigated. The
reduction is defined the same as in section 4.1, eq. 4.5. Before each line search a slow, but exact (CFD) analysis is made (method B in section 4.2). The error made with the fast (PIR) evaluations is therefore neglected. Again Rosenbrocks function (eq. 4.4) is used as a test function.

If the function is unconstrained the optimum is found at (1,1). The reduction achieved from two starting points (-2,1) and (2,-1) is tabulated in Table 5.1. It shows that the number of line searches has a big influence on the reduction. The standard line search needs 263 evaluations from (-2,1) and 240 from (2,-1)

<table>
<thead>
<tr>
<th>Starting point</th>
<th>Step size</th>
<th>Reduction</th>
<th>Line searches</th>
</tr>
</thead>
<tbody>
<tr>
<td>(-2, 1)</td>
<td>0.01</td>
<td>84.18</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>0.02</td>
<td>87.57</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>0.03</td>
<td>88.13</td>
<td>28</td>
</tr>
<tr>
<td>(2,-1)</td>
<td>0.01</td>
<td>86.99</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>0.02</td>
<td>91.23</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>0.03</td>
<td>93.11</td>
<td>14</td>
</tr>
</tbody>
</table>

*Table 5.1: Reduction (%) and number of line searches depending on starting point and step size. No constraints.*

Introduction of the inequality constraint \( x_2 \geq 0.5 \) in the optimisation with exterior penalty functions can have the same effect as with the SQP method in section 4.4. Depending on the step size, the optimisation process with the line search with small steps can converge to a different optimum than optimisation with the standard line search. A remarkable difference between the two optimisation routines is the followed search path (see Figure 4.5 and 5.1). There are two possible solutions, a local optimum at (-0.70,0.5) and the global optimum at (1,1). For two (infeasible) starting points, (-1,-1) and (2,-1), the search paths are plotted in Figure 5.1.

If there is an equality constraint, optimisation with the exterior penalty function method is quite different than with the SQP method. A great advantage is that the design points during the optimisation process can be infeasible with the exterior penalty function method. In contradiction to the SQP method, see Figure 4.6, the line search with small steps works in the exterior penalty function method. For starting points (-2,1) and (2,-1) the search paths of the optimisation with the equality constraint \( x_2 = x_1^2 + x_1 - 0.5 \) are plotted in Figure 5.2. The optimum is found at (0.51,0.27). The standard line search made 306 function evaluations from (2,-1) and 323 from (2,-1). The reduction from these two points is written in Table 5.2.
Optimisation using penalty functions

### 5.2 Summary

Optimisation with exterior penalty functions mostly needs more function analyses than optimisation with sequential quadratic programming. However, the use of line search with small steps has more effect, especially when there are constraints present in the optimisation problem. This can be explained by the fact that the exterior penalty function method allows infeasible design points during the optimisation process. Special notice should be given to convergence criteria. The optimum can be approximated in a few line searches. However, convergence needs more line searches for an accurate optimum. Avoiding expensive calculations (CFD) in this part of the optimisation can lead to a greater reduction.
Chapter 6

Approximation of aerodynamic functions

With the introduction of the PIR method the design space can be analysed much faster than with CFD tools. However, PIR must be applied for each point of interest. Therefore Topliss et al. (1996) developed a strategy to provide information for each point in the design space by using the results of the PIR method on several design points. With these results function fits are made for each aerodynamic coefficient (lift, drag, pitching moment, etc.), covering the entire design space. These functions can be used directly for optimisation. Expensive calculations of design points is no longer necessary. The required time to come to an optimal design point is almost completely determined by the number of CFD and PIR analyses to make a good function fit.

First the function fitting method used by Topliss et al. (1996) is described. Subsequently more methods for function fitting are discussed. In these methods three aspects are involved which interact with each other. The first aspect is the function on which the calculated aerodynamic coefficients are fitted. The second one is the choice of the experimental design. The experimental design determines how many design points have to be investigated and the position of these points in the design space. The third one is the accuracy of the function fit. This can be investigated with an analysis of variance.

6.1 Function approximation

The PIR method needs one CFD analysis as a starting point. This analysis is made at the centre of the design space. A number (depending on the number of design variables) of PIR analyses is made at specific points in the design space (see Figure 6.1 for two design variables $x_1$ and $x_2$). The CFD analysis is placed at the centre: $x_i = 0$; "-1" denotes the lower bound of a design variable, and "1" the upper bound.

Topliss et al. (1996) used the information of the CFD analysis ($\square$), the PIR analyses nearest to the CFD analysis ($\circ$ with dashed lines) and the PIR analyses on the boundaries ($\bullet$) to fit the functions for each aerodynamic coefficient, see Figure 6.1. For instance, if there
are $n$ design variables, $3 \times 2 \times n$ PIR analyses have been performed (3 analyses in each direction), and the data from $4 \times n$ of these are taken. With the initial CFD analysis, there are $4 \times n + 1$ points which provide information. The functions fitted on these points have been formulated as:

$$\hat{y}(x) = A_0 + \sum_{i=1}^{n} \sum_{j=1}^{4} A_{ij} x_i^j$$  \hspace{1cm} (6.1)$$

where $n$ is the number of design variables, $A_{ij}$ the coefficients to compute and $x_i^j$ the terms of the polynomial for each design variable. The extra $2 \times n$ points, which results are not used, are taken to ensure the accuracy in the PIR calculations at the boundary-points. This is done because larger steps lead to a bigger error in the PIR method.

If smaller perturbations are necessary to ensure the accuracy, more PIR analyses have to be made. Still only the results of $4 \times n$ design points are taken. There are $n_p$ analyses made from the CFD to the boundary points, so data is available for $n_p \times 2 \times n + 1$ points. For instance if $n_p = 12$, $24 \times n$ PIR analyses are made. However, the results of $20 \times n$ of these are discarded.

Optimisation with the function approximation is faster than with the line search with small steps. The advantage of the line search with small steps is that this method does not need accurate function approximations. Figure 6.2 shows these two ways for optimisation. Both need at least one CFD analysis. The number of PIR analyses is larger if the line search with small steps is applied. However, if the number of design variables $n$ increases, the number of PIR evaluations needed for the functions approximation increases. This increase depends on how many PIR analyses $n_p$ have to be made to ensure the accuracy of the PIR results at the boundary-points. The analyses are only made on the axes, so the total number of PIR analyses is $n_p \times 2 \times n$. 
6.2 Improvements in the function fitting method

The function fitting method described above can be improved in several ways. At first it is not safe to fit a fourth-order polynomial on only five points. Although this function fit is exact on all these five points, the error on other points can be quite large. It is better to fit the function on almost every available point, leaving some points to check the accuracy of the function fit. A frequently used method for making function fits is called the method of least squares, this method is described in the model fitting section.

Another important aspect of function fitting is the choice of an experimental design. The experimental design determines which design points should be calculated to make a good fit. If the design points are located only on the axes, the possible functions to fit are limited, and mixture of terms \((x_1x_2)\) is not possible.

At the end the accuracy of the function fit must be investigated, this can be done with an analysis of variance. An advantage of this method is that it shows the significance of the parameters in the function model. So it determines whether or not the function model can be simplified or should be extended.

6.3 Model fitting

In general any possible function model can be written as:

\[
y = \beta_1 f_1(x) + \ldots + \beta_k f_k(x) + \epsilon = f^T(x)\beta + \epsilon
\]  

(6.2)

where \(f_i, \ i = 1, \ldots, k\) can be any function depending on \(x\), i.e. \(x_1, x_2, x_1x_2, x_1^2x_2, e^{x_1}, \text{etc}\). The column \(\beta\) contains the unknown parameters \(\beta_i\), and \(\epsilon\) is the error between the calculated value and the value predicted by the model.

The values for the parameters \(\beta_i\) are determined by comparing the calculation \(y_j\) and the model expectation \(\hat{y}_j\) of a design point \(x_j, \ j = 1, \ldots, N\). \(N\) is the number of evaluated design points. Two columns can be constructed, a calculation column \(y\) and an expectation column \(\hat{y}\). These two columns can be written as:

\[
y = X\beta + \epsilon
\]

\[
\hat{y} = X\beta
\]

(6.3)

where \(\epsilon\) is the column of errors, \(X\) depends on \(f(x)\): \(X = [f(x_1), f(x_2), \ldots, f(x_N)]^T\) and \(x_1, x_2, \ldots, x_N\) are the evaluated design points.

An estimate for \(\beta\), \(\hat{\beta}\) can be derived from the normal equations \(X^TX\beta = X^Ty\):

\[
\hat{\beta} = (X^TX)^{-1}X^Ty
\]

(6.4)

In general not \((X^TX)^{-1}\) is determined, but the normal equations themselves are solved, for instance with the LU-decomposition combined with the QR-decomposition for the matrix \(X\). This leads to more accurate numerical results.
6.4 Experimental design

A very important aspect of model fitting is which design points should be evaluated for the function fit, also called experimental design. Choices have to be made about the number of design points and their place in the design space. Several methods of experimental design are discussed by Box and Draper (1987). Full factorial, fractional factorial, Latin hypercube, and D-optimal designs are discussed here.

If the PIR method is used to analyse the design points, one should consider its characteristics. There is always one CFD analysis necessary, this analyses is best placed at the centre of the design space. Furthermore, there are extra points evaluated with the PIR method.

In the experimental design problem, the number of levels in each design variable is very important, i.e. \( x_1 \) has three levels if design points are evaluated for three different values of \( x_1 \). The number of levels determines the possible function models. For instance, to fit a polynomial of the \( l \)-th order in a particular variable, there are at least \( l + 1 \) levels required for that variable.

6.4.1 Full factorial designs

The place of the design points in a full factorial design are shown in Figure 6.3 for a two-dimensional problem. Factorial designs are extensively discussed by Box et al. (1978). In a full factorial design the design points are placed on all possible combination of levels of the design variables.

![Figure 6.3: Full factorial design: ● are the design points to be evaluated.](image)

![Figure 6.4: Full factorial design with PIR: ● are the design points to be evaluated, □ the CFD analysis and ○ the extra PIR evaluations.](image)

The effect of evaluation of these design points with the PIR method is shown in Figure 6.4. There is a CFD analysis at the centre and extra points are 'generated' with the evaluation of the required design points. These points create extra sub-levels. With these sub-levels more complex models can be made.
A disadvantage of full factorial design is that the number of analysed design points grows rapidly with the number of design variables. If 3 levels are investigated, $3^n$ design points are required for $n$ design variables, see Figure 6.3 for $n = 2$. If the PIR method generates three extra points for one analysed design point, there are in total $3^n + 3 \times (3^n - 1)$ points which provide information for the function fit, see Figure 6.4 for $n = 2$.

### 6.4.2 Fractional factorial designs

A full factorial design becomes very large if the number of design variables or the number of levels increases. Mostly the function model is still accurate enough if a fraction of the factorial design is investigated. Which points to choose from a full factorial design is discussed by Box et al. (1978) for a two level factorial design. The best half fractions for a two level design are shown in Figure 6.5 for $n = 3$, the two levels for each design variable are the low-level "-" and the high-level "+", these levels are also the bounds on the design space. The best half-fractions are summarised in Table 6.1 for $n = 3$ and $n = 4$. Half fractions A and B are determined by multiplying the low-level "-" or high-level "+", for each design variable. This is done for each possible combination, half fraction A is formed with the positive results, B with the negative ones. For instance, multiplying each variable for $(x_1 x_2 x_3) = (- - +)$ gives a positive result, so this design point belongs to half fraction A. The two half fractions A and

<table>
<thead>
<tr>
<th>A</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n = 3$</td>
<td>$n = 4$</td>
</tr>
<tr>
<td>$x_1$</td>
<td>$x_1$</td>
</tr>
<tr>
<td>$x_2$</td>
<td>$x_2$</td>
</tr>
<tr>
<td>$x_3$</td>
<td>$x_3$</td>
</tr>
<tr>
<td>$x_4$</td>
<td>$x_4$</td>
</tr>
<tr>
<td>$-$</td>
<td>$-$</td>
</tr>
<tr>
<td>$-$</td>
<td>$-$</td>
</tr>
<tr>
<td>$+$</td>
<td>$+$</td>
</tr>
<tr>
<td>$+$</td>
<td>$+$</td>
</tr>
<tr>
<td>$+$</td>
<td>$+$</td>
</tr>
<tr>
<td>$-$</td>
<td>$-$</td>
</tr>
<tr>
<td>$+$</td>
<td>$+$</td>
</tr>
<tr>
<td>$+$</td>
<td>$+$</td>
</tr>
<tr>
<td>$-$</td>
<td>$-$</td>
</tr>
</tbody>
</table>

Figure 6.5: Best half fractions for $n = 3$.

Table 6.1: Best half fractions for $n = 3$ and $n = 4$. 
B together form a complete factorial design.

Again if these design points are evaluated with the PIR method, extra points are generated. The experimental design used by Topliss et al. (1996), see Figure 6.1, looks like a half fractional design with three levels, but it is not. The number of required design points to increases exponential with the number of design variables in a half fractional design (\(\frac{1}{2} \times 2^n\) for a two level half fractional design). However, in the experimental design used by Topliss et al. (1996) this increase is linear, see section 6.1.

6.4.3 Latin hypercube designs

In the Latin hypercube design (van Houten, 1995) only one design point has to be evaluated for each level of a design variable, see Figure 6.6. The number of design points to be evaluated for each design variable is therefore equal to the number of levels. Compared with the factorial design, the Latin hypercube has an advantage if a design variable is not significant for the model function. In Figure 6.6 this is shown if variable \(x_2\) is not important for the function model: five levels remain for \(x_1\). If this is done in a full factorial design (Figure 6.3) only three levels remain for \(x_1\). If the PIR method is used, one design point is situated at the centre of the design space: the CFD analysis. This, and the extra PIR steps are shown in Figure 6.7.

![Figure 6.6: Latin hypercube design: • are the design points to be evaluated.](image1)

![Figure 6.7: Latin hypercube design with PIR: • are the design points to be evaluated, □ the CFD analysis and ○ the extra PIR evaluations.](image2)

Figure 6.6 compared with 6.4 shows that the initial advantage of the Latin hypercube design compared with the factorial design disappears partly if the PIR method is used. Although the PIR method creates less (sub)levels with the factorial design than with the Latin hypercube design, the (sub)levels are better placed in the factorial design. In Figure 6.7 12
(sub)levels remain for $x_1$ if $x_2$ is not important. If this is done in Figure 6.4 9 (sub)levels remain for $x_1$. Although the initial advantage decreases, another advantage of the Latin hypercube design appears if the PIR method is used. The design points to be evaluated are located closer to the CFD analyses: less PIR analyses are necessary and the PIR code is more accurate.

6.4.4 D-optimal designs

The objective of the D-optimal design is to determine those $N$ points from a set of candidate points which minimise the variances of the estimated parameters. The candidate points can be part of another experimental design, i.e. a full factorial design. The variance of the estimated parameters can be minimised by minimising $\text{det}(X^TX)^{-1}$ (section 6.3). In all the earlier methods the number of design points to be evaluated grows rapidly with the number of design variables. The optimal design tries to give a maximum amount of information for a minimum of design points. A disadvantage of the D-optimal design is that the function model must be known prior to the choice of design points (in contrast to the other methods). However, the advantage is that if extra design points are needed, these points are placed optimally by taken into account the first evaluated design points. The D-optimal design is further discussed by van Houten (1994).

If the PIR method is used in a D-optimal design $N$ points are evaluated. This results in data for $\sum_{i=1}^{N}(n_p)_i$ points. There is no guarantee that the variance of the estimated parameters calculated with these points is minimised. The same number of (other) design points can give less variance because the extra generated points with the PIR method are not located optimally. However, if the number of design variables increases one should consider the D-optimal design because this limits the required evaluations.

6.4.5 Summary

The PIR method is best used with a fractional factorial design or a Latin hypercube design. In the fractional factorial design the extra generated design points are most useful, the sub-levels created by the extra PIR analyses can be located better than in a Latin hypercube design and the number of design points to be evaluated are less than in a full factorial design. The advantage of the Latin hypercube design is that a great part of the design points is located closer to the CFD analysis, which results in less and more accurate PIR evaluations. In the D-optimal design the extra PIR analyses can not be placed optimally. However, with increasing number of design variables and knowledge of aerodynamic function models the D-optimal design is more useful than the other experimental designs.

Compared with Figure 6.1, it is better to evaluate the vertices of the design space rather than the end points on the axes. However, the number of vertices grows exponentially with the number of design variables ($2^n$), while the number of end points is linear with the number of design variables ($2 \times n$). A proposed experimental design is shown for two design variables in Figure 6.8, where 3 levels are evaluated and 6 sub-levels are created with the PIR method. This experimental design is derived from a full factorial design (Figure 6.4). The difference with the experimental design used by Topliss et al. (Figure 6.1) is the evaluation of design
6 Approximation of aerodynamic functions

Figure 6.8: Proposed design points for \( n = 2 \).

Table 6.2: Proposed design points for \( n = 2 \) and \( n = 3 \).

<table>
<thead>
<tr>
<th>( n = 2 )</th>
<th>( n = 3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 )</td>
<td>( x_1 )</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>

points on the diagonals instead of design points on the axes. Interaction between design variables is possible with the proposed experimental design. The number of sub-levels can be changed by variation of the number of PIR analyses. Table 6.2 shows the CFD analysis and the proposed design points to be evaluated for \( n = 2 \) and \( n = 3 \). The CFD analysis \((x_i = 0)\) is placed at the centre, "-1" denotes the lower bound of a design variable \( x_i \) and "1" the higher bound, respectively. A disadvantage of this experimental design is that the number or required evaluations grows exponentially with the number of design variables.

6.5 Analysis of variance

One way to check the accuracy of the function model is with an analysis of variance. This can be presented in an ANOVA table (Chatfield, 1983), see Table 6.3: \( N \) is the number of design points used for the fit, \( k \) the number of parameters in the function model and D.F. represents the degree of freedom. The difference between the calculation \( y_i \) and mean

\[
\bar{y} = \frac{1}{N} \sum_{i=1}^{N} y_i
\]

is splitted up into two components:

\[
y_i - \bar{y} = (y_i - \bar{y}) + (\bar{y} - \bar{y})
\]

This is also possible for the sum of squares:

\[
SS_{yy} = \sum_{i=1}^{N} (y_i - \bar{y})^2 = \sum_{i=1}^{N} (y_i - \bar{y})^2 + \sum_{i=1}^{N} (\bar{y}_i - \bar{y})^2
\]
6 Approximation of aerodynamic functions

<table>
<thead>
<tr>
<th>Source</th>
<th>Sum of Squares</th>
<th>D.F.</th>
<th>Mean Square</th>
<th>$F$</th>
<th>$R$-squared</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>$SS_M$</td>
<td>$k - 1$</td>
<td>$MS_M = SSM/(k - 1)$</td>
<td>$MSM/MSE$</td>
<td>$SSM/SS_{yy}$</td>
</tr>
<tr>
<td>Error</td>
<td>$SSE$</td>
<td>$N - k$</td>
<td>$MSE = SSE/(N - k)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>$SS_{yy}$</td>
<td>$N - 1$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6.3: ANOVA: Analysis of Variance.

The first term,

$$SSE = \sum_{i=1}^{N}(y_i - \bar{y}_i)^2$$

is called Error Sums of Squares. This is in fact a coincidence term. The second one,

$$SS_M = \sum_{i=1}^{N}(\hat{y}_i - \bar{y})^2$$

is the Model Sum of Squares, this term depends on the model. $F$ represents the goodness of fit. If $F$ is significant larger than 1, the model fits the data correctly. $R$-squared $= SS_M/SS_{yy}$ gives the part of the total sum of squares depending on the model: the higher the value the better the model.

6.6 Aerodynamic function fits

Several function models and choices of experimental designs are compared using PIR analyses on a 2-D aerofoil design (RAE2882). The design variables are the free stream Mach number $M$ and the angle of attack $\alpha$ (Figure 1.1). The Mach number lies between 0.7 and 0.8 (subsonic), while the angle of attack is bounded by -10 and 10 degrees. The CFD analyses is placed in the middle of this design space: Mach number is equal to 0.75 and the angle of attack is 0 degrees. The shape variables (Figure 1.2) do not change, so the area of the aerofoil is constant.

Function fits are made for the lift and the drag coefficient ($C_l$ and $C_d$). At first, the PIR analyses on which the fits are made are placed on the axes (Figure 6.1). Secondly the PIR method is used to evaluate the vertices of the design space (Figure 6.8). The absolute and relative accuracy of the PIR method is shown in Table 6.4 for the end points. The number of PIR evaluations needed to calculate an end point differs for each one. Difficulties arise with the calculation of the drag coefficient at (0.70,0). Although the increment here is chosen smaller than in the other cases the drag coefficient becomes negative (a negative drag is not
Table 6.4: Absolute and relative accuracy of PIR analyses at specific boundary points.

### Possible. The relative error is therefore bigger than 100% (not shown in Table 6.4). However, the absolute error differs not that much from the other end points. Problems also arise with the evaluation of (0.80,10), the increment here must be taken ten times smaller. This is not done to ensure the accuracy of the PIR results, but to ensure that the PIR method works. The design space chosen here is quite large, it should be decreased to increase the accuracy of the PIR method.

One CFD run takes five to six hours and one PIR step around twenty-five seconds. There is only a slight difference in the run-time of the different PIR iterations (changing the angle of attack, the Mach number or both together). So in one CFD run 700 to 850 PIR iterations can be made. This means a great reduction in computing time, however the evaluation of (0.80,10) took around 500 steps, so the reduction here is only 40%, while the evaluation of the other points gives a reduction between 90 and 95%.

### 6.6.1 Axial function fits

First, function fits are made if the PIR evaluations are placed on the axes. The influence of the order of the function model is also investigated. The used design points are only located on the axes, so mixture of terms is not possible. The used function models can be described by

\[
\hat{y}(x) = A_0 + \sum_{i=1}^{n} \sum_{j=1}^{l} A_{ij} x_i^j,
\]

with \(n\) the number of design variables, \(l\) the order of the function and \(A, A_{ij}\) the model parameters. First, all the available data from the PIR evaluations on the axes is used to determine the model parameters. Figure 6.9 shows a third order function fit of the lift coefficient and Figure 6.10 a fourth order function fit of the drag coefficient as function of the Mach number and angle of attack. In Appendix A (Table A.1 to A.4), the accuracy of the function fits is investigated, this is done for several function orders. Function values are compared with PIR results and an analysis of variance is given. To check the accuracy of the fit one should only look at the difference at points on the axes. For the lift coefficient...
6 Approximation of aerodynamic functions

$F$ and $R$ values), it covers the entire design space better (the errors made on the axes with the diagonal fits are smaller than the errors made on the diagonals with the axial fits).

6.6.3 Summary

The function fits of the lift and drag coefficient are smooth functions, and there is not much interaction between the design variables $M$ and $\alpha$. Therefore, the extra information given by the diagonal fits is small. The diagonal fit has a disadvantage if the number of design variables increases. The number of design points to be evaluated increases exponential with the number of design variables and not linear as with the axial fit. Furthermore, evaluation of the vertices with the PIR method is more complex than the evaluation of the end points on the axes (only one design variable has to be varied for the evaluation of an end point).

If axial fits are made, the functions are not valid for the entire design space. Figure 6.15 and 6.16 show the part of the design space (with two design variables $x_1$ and $x_2$) for which the axial and diagonal function fits are valid.

![Figure 6.15: Valid design space for axial function fits.](image1)

![Figure 6.16: Valid design space for diagonal function fits.](image2)

In this stage, where the functions are smooth, it is best to make axial function fits on many design points. If these functions are used for optimisation, the accuracy should be tested with a CFD analysis at the found optimum. If the results are not accurate enough, fits can be made based on that CFD analysis and the optimisation can be restarted.

The accuracy of the function fits can be improved, if more CFD analyses are used for the fit. The PIR evaluations are made on the axes, so these CFD analyses should be placed on the diagonals, for instance in the vertices of the design space. The results of these CFD analyses can then be used in the function fit to increase its accuracy, they are also an indication whether the function fit is good enough. To further improve the accuracy one can introduce weight-factors on the data. The CFD results are more accurate than the PIR ones, so the CFD data can be given a higher weight-factor.
Chapter 7

Conclusions and recommendations

The optimisation of aerodynamic designs becomes much faster with the use of the Projected Implicit Reconstruction method. Based on this PIR method, a new type of line search is successfully developed, the line search with small steps. The line search is a part of the optimisation routine. Other important parts of the optimisation routine are the calculation of the search direction, the influence of the constraints, and the convergence criteria. The line search with small steps is most effective if the optimisation process allows movement of intermediate designs in the infeasible domain. The use of exterior penalty functions is therefore better in this case than the use of a Sequential Quadratic Programming routine, especially if there are equality constraints present. Special notice should be given to the convergence criteria. These have a great influence in the computational time if the line search with small steps is used in combination with the PIR method.

Functions for the aerodynamic coefficients (lift, drag and pitching moment) simplify the optimisation of aerodynamic designs. These functions are made by fitting results from CFD and PIR analyses. The number of data points must be large enough to ensure the accuracy of the fit. If only the Mach number $M$ and angle of attack $\alpha$ are changed, the functions for the lift and drag coefficient are smooth and there is little interaction between $M$ and $\alpha$. So the design points on which the functions are fitted can be placed on the axes. This choice of experimental design (determination of the location of design points) makes PIR analyses easier, because only one design variable has to be varied at a time. However, if interaction between design variables is present the function fits can be inaccurate. To increase the accuracy there are extra CFD analyses necessary. Another option to increase the accuracy is to change the experimental design. Good experimental designs which make use of the PIR method are the proposed experimental design (section 6.4.5) and the Latin hypercube design. A D-optimal design should be used if the number of design variables increases.

Finally, the accuracy of the aerodynamic functions depends on the accuracy of the PIR method. So the design space that can be investigated is limited by the PIR method. Therefore, more CFD analyses are required to investigate a larger design space.
Bibliography


Appendix A

Accuracy of axial function fits

The function fits of the lift $C_l$ and drag coefficient $C_d$ are based on design points which are placed on the axes. Second, third and fourth order models are investigated.

<table>
<thead>
<tr>
<th>Order</th>
<th>Source</th>
<th>Sum of Squares</th>
<th>D.F.</th>
<th>Mean Square</th>
<th>$F$</th>
<th>R-squared</th>
</tr>
</thead>
<tbody>
<tr>
<td>Second</td>
<td>Model</td>
<td>77.18</td>
<td>4</td>
<td>19.30</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Error</td>
<td>0.423</td>
<td>216</td>
<td>0.002</td>
<td>9.86 $10^3$</td>
<td>0.9946</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>77.61</td>
<td>220</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Third</td>
<td>Model</td>
<td>77.59</td>
<td>6</td>
<td>12.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Error</td>
<td>0.012</td>
<td>214</td>
<td>5.38 $10^{-5}$</td>
<td>2.41 $10^5$</td>
<td>0.9999</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>77.61</td>
<td>220</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fourth</td>
<td>Model</td>
<td>77.60</td>
<td>8</td>
<td>9.70</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Error</td>
<td>0.011</td>
<td>212</td>
<td>5.36 $10^{-5}$</td>
<td>1.81 $10^5$</td>
<td>0.9999</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>77.61</td>
<td>220</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Table A.1: ANOVA: second, third and fourth order model fit on the lift coefficient $C_l$.*

<table>
<thead>
<tr>
<th>Order</th>
<th>Source</th>
<th>Sum of Squares</th>
<th>D.F.</th>
<th>Mean Square</th>
<th>$F$</th>
<th>R-squared</th>
</tr>
</thead>
<tbody>
<tr>
<td>Second</td>
<td>Model</td>
<td>1.710</td>
<td>4</td>
<td>0.428</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Error</td>
<td>0.007</td>
<td>216</td>
<td>3.20 $10^{-5}$</td>
<td>1.34 $10^3$</td>
<td>0.9960</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>1.717</td>
<td>220</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Third</td>
<td>Model</td>
<td>1.714</td>
<td>6</td>
<td>0.289</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Error</td>
<td>0.002</td>
<td>214</td>
<td>1.13 $10^{-5}$</td>
<td>2.52 $10^5$</td>
<td>0.9986</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>1.717</td>
<td>220</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fourth</td>
<td>Model</td>
<td>1.717$-4$</td>
<td>8</td>
<td>0.215</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Error</td>
<td>2.41$-6$</td>
<td>212</td>
<td>1.62 $10^{-6}$</td>
<td>1.85 $10^5$</td>
<td>0.9999</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>1.717</td>
<td>220</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Table A.2: ANOVA: second, third and fourth order model fit on the drag coefficient $C_d$.***
Table A.1 and A.2 show the analysis of variance for the second, third and fourth order model fits on the lift coefficient $C_l$ and the drag coefficient $C_d$. 221 design points are used for the function fits.

The accuracy of the function fits can also be investigated at the centre and at the endpoints on the axes. This shows the accuracy of the fit on the design points. Whether or not the function model is valid for the entire design space is investigated by looking at the results at the end points on the diagonals. Table A.3 shows the accuracy of the functions for the lift coefficient $C_l$, Table A.4 for the drag coefficient $C_d$. The results of the function fits are compared with the PIR results (Table 6.4), because the function fits are based on the PIR evaluations and not on the CFD analyses.

<table>
<thead>
<tr>
<th>$M$</th>
<th>$\alpha$</th>
<th>Second</th>
<th>error (abs &amp; %)</th>
<th>Third</th>
<th>error (abs &amp; %)</th>
<th>Fourth</th>
<th>error (abs &amp; %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.75</td>
<td>0</td>
<td>0.5056</td>
<td>-0.0321 -6.77</td>
<td>0.4799</td>
<td>-0.0064 -1.35</td>
<td>0.4807</td>
<td>-0.0072 -1.52</td>
</tr>
<tr>
<td>0.75</td>
<td>-10</td>
<td>-1.2790</td>
<td>0.1970 -18.20</td>
<td>-1.0715</td>
<td>-0.0105 0.97</td>
<td>-1.0684</td>
<td>-0.0136 1.26</td>
</tr>
<tr>
<td>0.75</td>
<td>10</td>
<td>1.8751</td>
<td>-0.1012 -5.71</td>
<td>1.7488</td>
<td>0.0251 1.42</td>
<td>1.7506</td>
<td>0.0233 1.31</td>
</tr>
<tr>
<td>0.70</td>
<td>0</td>
<td>0.3841</td>
<td>0.0181 4.49</td>
<td>0.4023</td>
<td>-0.0001 -0.03</td>
<td>0.4047</td>
<td>-0.0025 -0.61</td>
</tr>
<tr>
<td>0.80</td>
<td>0</td>
<td>0.5926</td>
<td>0.0146 2.40</td>
<td>0.6070</td>
<td>0.0002 0.03</td>
<td>0.6094</td>
<td>-0.0022 -0.35</td>
</tr>
<tr>
<td>0.70</td>
<td>-10</td>
<td>-1.4004</td>
<td>0.3648 -35.23</td>
<td>-1.1491</td>
<td>0.1135 -10.69</td>
<td>-1.1444</td>
<td>0.1088 -10.51</td>
</tr>
<tr>
<td>0.70</td>
<td>10</td>
<td>1.7537</td>
<td>-0.0160 -0.92</td>
<td>1.6712</td>
<td>0.0665 3.83</td>
<td>1.6746</td>
<td>0.0631 3.63</td>
</tr>
<tr>
<td>0.80</td>
<td>-10</td>
<td>-1.1919</td>
<td>0.0723 -6.46</td>
<td>-0.9444</td>
<td>-0.1752 15.65</td>
<td>-0.9397</td>
<td>-0.1799 16.07</td>
</tr>
<tr>
<td>0.80</td>
<td>10</td>
<td>1.9622</td>
<td>-0.2233 -12.84</td>
<td>1.8759</td>
<td>-0.1370 -7.88</td>
<td>1.8792</td>
<td>-0.1403 -8.07</td>
</tr>
</tbody>
</table>

Table A.3: Absolute and relative accuracy of the function fit for a complete second, third and fourth order function model on the lift coefficient $C_l$ at specific boundary points, only design point on the axes are used for the fit.

<table>
<thead>
<tr>
<th>$M$</th>
<th>$\alpha$</th>
<th>Second</th>
<th>error (abs &amp; %)</th>
<th>Third</th>
<th>error (abs &amp; %)</th>
<th>Fourth</th>
<th>error (abs &amp; %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.75</td>
<td>0</td>
<td>0.0072</td>
<td>-0.0036 -100.86</td>
<td>0.0046</td>
<td>-0.0010 -26.77</td>
<td>0.0028</td>
<td>0.0008 23.11</td>
</tr>
<tr>
<td>0.75</td>
<td>-10</td>
<td>0.1967</td>
<td>0.0046 2.28</td>
<td>0.2183</td>
<td>-0.0170 -8.43</td>
<td>0.2000</td>
<td>0.0013 0.67</td>
</tr>
<tr>
<td>0.75</td>
<td>10</td>
<td>0.3303</td>
<td>-0.0214 -6.93</td>
<td>0.3172</td>
<td>-0.0083 -2.68</td>
<td>0.3081</td>
<td>0.0008 0.25</td>
</tr>
<tr>
<td>0.70</td>
<td>0</td>
<td>-0.0006</td>
<td>0.0004 -220.57</td>
<td>-0.0006</td>
<td>0.0004 -194.84</td>
<td>-0.0007</td>
<td>0.0005 -266.73</td>
</tr>
<tr>
<td>0.80</td>
<td>0</td>
<td>0.0251</td>
<td>0.0035 12.26</td>
<td>0.0284</td>
<td>0.0002 0.61</td>
<td>0.0283</td>
<td>0.0003 1.11</td>
</tr>
<tr>
<td>0.70</td>
<td>-10</td>
<td>0.1888</td>
<td>-0.0331 -21.29</td>
<td>0.2131</td>
<td>-0.0574 -36.87</td>
<td>0.1965</td>
<td>-0.0408 -26.18</td>
</tr>
<tr>
<td>0.70</td>
<td>10</td>
<td>0.3224</td>
<td>-0.0720 -28.77</td>
<td>0.3120</td>
<td>-0.0616 -24.61</td>
<td>0.3047</td>
<td>-0.0542 -21.65</td>
</tr>
<tr>
<td>0.80</td>
<td>-10</td>
<td>0.2146</td>
<td>0.0379 15.02</td>
<td>0.2421</td>
<td>0.0104 4.11</td>
<td>0.2255</td>
<td>0.0270 10.70</td>
</tr>
<tr>
<td>0.80</td>
<td>10</td>
<td>0.3482</td>
<td>0.0058 1.65</td>
<td>0.3410</td>
<td>0.0130 3.66</td>
<td>0.3336</td>
<td>0.0204 5.75</td>
</tr>
</tbody>
</table>

Table A.4: Absolute and relative accuracy of the function fit for a complete second, third and fourth order function model on the drag coefficient $C_d$ at specific boundary points, only design points on the axes are used for the fit.
To investigate the fourth order function fits on only nine design points, the fit is compared with the other design points. Table A.5 is not a real ANOVA table. The difference is that not only the nine design points are used to calculate the ANOVA table, but also the other available design points. If only the nine points were used, the results would be perfect: all the Squares zero, $F = \infty$ and $R = 1$.

<table>
<thead>
<tr>
<th>Function</th>
<th>Source</th>
<th>Sum of Squares</th>
<th>D.F.</th>
<th>Mean Square</th>
<th>$F$</th>
<th>$R$-squared</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_l$</td>
<td>Model</td>
<td>83.18</td>
<td>8</td>
<td>10.40</td>
<td>5.34 $10^3$</td>
<td>0.9951</td>
</tr>
<tr>
<td></td>
<td>Error</td>
<td>0.413</td>
<td>212</td>
<td>1.94 $10^{-3}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>83.60</td>
<td>220</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C_d$</td>
<td>Model</td>
<td>1.654</td>
<td>8</td>
<td>0.207</td>
<td>1.10 $10^5$</td>
<td>0.9976</td>
</tr>
<tr>
<td></td>
<td>Error</td>
<td>0.004</td>
<td>212</td>
<td>1.87 $10^{-5}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>1.658</td>
<td>220</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table A.5: ANOVA: fourth order model fits on the lift $C_l$ and the drag coefficient $C_d$, only nine points used.
Appendix B

Accuracy of diagonal function fits

The function fits on the lift $C_l$ and drag coefficient $C_d$ are based on design points which are placed on the diagonals (Figure 6.8).

<table>
<thead>
<tr>
<th>Order</th>
<th>Source</th>
<th>Sum of Squares</th>
<th>D.F.</th>
<th>Mean Square</th>
<th>$F$</th>
<th>R-squared</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Model</td>
<td>320.23</td>
<td>5</td>
<td>64.05</td>
<td>1.44$10^4$</td>
<td>0.9911</td>
</tr>
<tr>
<td></td>
<td>Error</td>
<td>2.87</td>
<td>645</td>
<td>0.0044</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>323.10</td>
<td>650</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Model</td>
<td>322.88</td>
<td>9</td>
<td>35.88</td>
<td>1.05$10^5$</td>
<td>0.9993</td>
</tr>
<tr>
<td></td>
<td>Error</td>
<td>0.22</td>
<td>64</td>
<td>3.42$10^{-4}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>323.10</td>
<td>650</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Model</td>
<td>322.91</td>
<td>14</td>
<td>23.07</td>
<td>7.90$10^4$</td>
<td>0.9994</td>
</tr>
<tr>
<td></td>
<td>Error</td>
<td>0.19</td>
<td>636</td>
<td>2.92$10^{-4}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>323.10</td>
<td>650</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table B.1: ANOVA: second, third and fourth order model fit on the lift coefficient $C_l$.

<table>
<thead>
<tr>
<th>Order</th>
<th>Source</th>
<th>Sum of Squares</th>
<th>D.F.</th>
<th>Mean Square</th>
<th>$F$</th>
<th>R-squared</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Model</td>
<td>7.266</td>
<td>5</td>
<td>1.453</td>
<td>9.70$10^{-5}$</td>
<td>1.50$10^4$</td>
</tr>
<tr>
<td></td>
<td>Error</td>
<td>0.063</td>
<td>645</td>
<td>9.70$10^{-5}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>7.329</td>
<td>650</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Model</td>
<td>7.300</td>
<td>9</td>
<td>0.811</td>
<td>4.41$10^{-5}$</td>
<td>1.84$10^4$</td>
</tr>
<tr>
<td></td>
<td>Error</td>
<td>0.028</td>
<td>641</td>
<td>4.41$10^{-5}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>7.329</td>
<td>650</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Model</td>
<td>7.325</td>
<td>14</td>
<td>0.523</td>
<td>6.13$10^{-6}$</td>
<td>8.54$10^4$</td>
</tr>
<tr>
<td></td>
<td>Error</td>
<td>0.004</td>
<td>636</td>
<td>6.13$10^{-6}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>7.329</td>
<td>650</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table B.2: ANOVA: second, third and fourth order model fit on the drag coefficient $C_d$. 40
Table B.1 and B.2 show the analysis of variance for the second, third and fourth order model fits on the lift coefficient \( C_l \) and the drag coefficient \( C_d \). 651 design points are used for the function fits.

Table B.3 and B.4 show the accuracy for the lift \( C_l \) and drag coefficient \( C_d \) at the centre, at the end points on the axes and on the end points on the diagonals. The results from the function models are compared with the PIR results from Table 6.4 because the functions are based on the PIR evaluations. The accuracy at the centre and end points on the diagonals shows the goodness of fit. The accuracy at the end points on the axes shows whether or not the function is valid for the entire design space.

<table>
<thead>
<tr>
<th>( M )</th>
<th>( \alpha )</th>
<th>Second</th>
<th>error (abs &amp; %)</th>
<th>Third</th>
<th>error (abs &amp; %)</th>
<th>Fourth</th>
<th>error (abs &amp; %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.75</td>
<td>0</td>
<td>0.5587</td>
<td>-0.0862</td>
<td>-17.99</td>
<td>0.4953</td>
<td>-0.0218</td>
<td>-4.59</td>
</tr>
<tr>
<td>0.75</td>
<td>-10</td>
<td>-1.3202</td>
<td>0.2382</td>
<td>-22.02</td>
<td>-1.0775</td>
<td>-0.0045</td>
<td>4.42</td>
</tr>
<tr>
<td>0.75</td>
<td>10</td>
<td>1.8114</td>
<td>-0.0375</td>
<td>-2.11</td>
<td>1.7018</td>
<td>0.0721</td>
<td>4.07</td>
</tr>
<tr>
<td>0.70</td>
<td>0</td>
<td>0.5447</td>
<td>-0.1425</td>
<td>-35.44</td>
<td>0.4347</td>
<td>-0.0325</td>
<td>-8.08</td>
</tr>
<tr>
<td>0.80</td>
<td>0</td>
<td>0.5733</td>
<td>0.0339</td>
<td>5.58</td>
<td>0.5643</td>
<td>0.0429</td>
<td>7.06</td>
</tr>
<tr>
<td>0.70</td>
<td>-10</td>
<td>-1.3208</td>
<td>0.2852</td>
<td>-27.54</td>
<td>-0.9795</td>
<td>-0.0561</td>
<td>5.42</td>
</tr>
<tr>
<td>0.70</td>
<td>10</td>
<td>1.7841</td>
<td>-0.0464</td>
<td>-2.67</td>
<td>1.6936</td>
<td>0.0441</td>
<td>2.54</td>
</tr>
<tr>
<td>0.80</td>
<td>-10</td>
<td>-1.3189</td>
<td>0.1993</td>
<td>-17.80</td>
<td>-1.1480</td>
<td>0.0284</td>
<td>2.53</td>
</tr>
<tr>
<td>0.80</td>
<td>10</td>
<td>1.8394</td>
<td>-0.1005</td>
<td>-5.78</td>
<td>1.6995</td>
<td>0.0394</td>
<td>2.27</td>
</tr>
</tbody>
</table>

Table B.3: Absolute and relative accuracy of the function fit for a complete second, third and fourth order function model on the lift coefficient \( C_l \) at specific boundary points, only design points on the diagonals are used for the fit.

<table>
<thead>
<tr>
<th>( M )</th>
<th>( \alpha )</th>
<th>Second</th>
<th>error (abs &amp; %)</th>
<th>Third</th>
<th>error (abs &amp; %)</th>
<th>Fourth</th>
<th>error (abs &amp; %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.75</td>
<td>0</td>
<td>0.0155</td>
<td>0.0069</td>
<td>3.42</td>
<td>0.2272</td>
<td>-0.0259</td>
<td>-12.87</td>
</tr>
<tr>
<td>0.75</td>
<td>-10</td>
<td>0.1944</td>
<td>-0.0079</td>
<td>-219.89</td>
<td>0.0047</td>
<td>-0.0011</td>
<td>-29.57</td>
</tr>
<tr>
<td>0.75</td>
<td>10</td>
<td>0.3234</td>
<td>-0.0145</td>
<td>-4.69</td>
<td>0.2986</td>
<td>0.0103</td>
<td>3.35</td>
</tr>
<tr>
<td>0.70</td>
<td>0</td>
<td>-0.0358</td>
<td>0.0356</td>
<td>\</td>
<td>-0.0389</td>
<td>0.0396</td>
<td>\</td>
</tr>
<tr>
<td>0.80</td>
<td>0</td>
<td>0.0615</td>
<td>-0.0356</td>
<td>-115.17</td>
<td>0.0591</td>
<td>-0.0305</td>
<td>-106.65</td>
</tr>
<tr>
<td>0.70</td>
<td>-10</td>
<td>0.1528</td>
<td>0.0029</td>
<td>1.89</td>
<td>0.1896</td>
<td>-0.0339</td>
<td>-21.79</td>
</tr>
<tr>
<td>0.70</td>
<td>10</td>
<td>0.2705</td>
<td>-0.0201</td>
<td>-8.03</td>
<td>0.2546</td>
<td>-0.0042</td>
<td>-1.68</td>
</tr>
<tr>
<td>0.80</td>
<td>-10</td>
<td>0.2388</td>
<td>0.0137</td>
<td>5.41</td>
<td>0.2634</td>
<td>-0.0109</td>
<td>-4.33</td>
</tr>
<tr>
<td>0.80</td>
<td>10</td>
<td>0.3790</td>
<td>-0.0250</td>
<td>-7.07</td>
<td>0.3639</td>
<td>-0.0099</td>
<td>-2.80</td>
</tr>
</tbody>
</table>

Table B.4: Absolute and relative accuracy of the function fit for a complete second, third and fourth order function model on the drag coefficient \( C_d \) at specific boundary points, only design points on the diagonals are used for the fit.
the absolute and relative error are small, while the fourth order function model is not much better than the third order function model. The relative errors in the drag coefficient can become large but the absolute errors stay small. This can be explained by the fact that the fitting method only looks at the absolute differences between the model expectations and the calculations. The fourth order function model for the drag coefficient is an improvement over the third order function model. This can be seen with an analysis of variance, $F$ and $R$ increase with the order in contrast to the lift coefficient where there is almost no difference between the third and fourth order model.

If a function of the $l$-th order is fitted on $l + 1$ points, this fit is exact. However the fit on other design points can be inaccurate. In Figure 6.1 the experimental design is shown for a fourth order function model with two design variables, only nine (necessary) design points are used. Figure 6.11 shows the difference of the fourth order function fit for the lift coefficient on all the points and the fit on only 9 points, the same is done for the drag coefficient in
Approximation of aerodynamic functions

Figure 6.12. With an analysis of variance it can be shown that the fit on all the points is better. $F$ and $R$ are both larger than with the fits on only nine points. For the fourth order function fit on the lift coefficient these values are: $F_{all} = 1.81 \times 10^5$, $F_{nine} = 5.34 \times 10^3$, $R_{all} = 0.9999$ and $R_{nine} = 0.9976$. Other values of the analysis of variance of the fit on nine points are given in Appendix A (Table A.5 and A.6).

Although it seems that the fit on only nine points is not that bad, it is better to fit the function on more points. The reason that the difference between the fits is not that great can be explained by the 'smoothness' of the functions, great distortions do not occur in the functions for the lift and drag coefficient.

6.6.2 Diagonal function fits

With the proposed design in section 6.4.5 the PIR analyses are placed on the diagonals of the design space. This location of the design points makes mixture of terms possible. A complete $l$-th order model is now equation 6.7 expanded with these extra terms. For instance, a complete third order model with two design variables can be written as:

$$
\hat{y}(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_1 x_2 + \beta_5 x_2^2 + \beta_6 x_1^3 + \beta_7 x_1^2 x_2 + \beta_8 x_1 x_2^2 + \beta_9 x_2^3
$$

(6.8)

The number of parameters $\beta_i$ increases fast for a complete function model if the number of design variables or the order of the model increases. The lift coefficient is fitted with a complete third order model, the difference with the fit on the axes is shown in Figure 6.13. The function model of the drag coefficient is a complete fourth order model, Figure 6.14 shows the difference of this fit with the fit on the axes. As expected, the difference between the fit on the axes and the fit on the diagonals is largest in the vertices of the design space. The accuracy of the function fits on the diagonals is given in Appendix B for several function orders. A complete third order model suffices again for the lift coefficient.

The function fits with the design points on the diagonals show a slight improvement over the fits with the design points on the axes. Although the fit on the diagonals is worser (lower