Summer 2005

Principal Component Regression for Construction of Wing Weight Estimation Models

Humberto Rocha
Old Dominion University

PRINCIPAL COMPONENT REGRESSION FOR CONSTRUCTION OF WING WEIGHT ESTIMATION MODELS

by

Humberto Rocha
B.S. September 1998, University of Coimbra, Portugal
M.S. March 2001, University of Coimbra, Portugal

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirement for the Degree of

DOCTOR OF PHILOSOPHY

COMPUTATIONAL AND APPLIED MATHEMATICS

OLD DOMINION UNIVERSITY
August 2005

Approved by:

John J. Swetits (Director)

Fang Q. Hu (Member)

Przemyslaw Bogacki (Member)

Dayanand N. Naik (Member)

Wu Li (Advisor)

ABSTRACT

PRINCIPAL COMPONENT REGRESSION FOR CONSTRUCTION OF WING WEIGHT ESTIMATION MODELS

Humberto Rocha
Old Dominion University, 2005
Director: Dr. John J. Swetits

The multivariate data fitting problem occurs frequently in many branches of science and engineering. It is very easy to fit a data set exactly by a mathematical model no matter how the data points are distributed. But building a response by using a limited number of poorly distributed data points is very unreliable, yet necessary in conceptual design process. This thesis documents the lessons learned from fitting the wing weight data of 41 subsonic transports by three types of interpolation methods – least polynomial interpolation, radial basis function interpolation, and Kriging interpolation. The objective of this thesis is to develop an automatic procedure of using this interpolation methods for construction of an approximation of the relationship between the actual wing weight and various key configuration parameters of wing by using actual wing weight data of 41 subsonic transports. The focus of the thesis is on four key technical issues in practical use of approximation methods: data generation and variable screening, fitting the data by a parametric function model, tuning intrinsic model parameters by using cross-validation, and verification of constructed approximation. One controversial topic is the assessment of the constructed approximations, which is of great importance to practitioners but depends too much on subjective judgment. Some formal approaches for the assessment will be proposed and analyzed. Even though the benefits of using principal component regression with cross validation are only demonstrated by the wing weight data fitting problem, the proposed methodology could have significant advantages in fitting other historical or hard-to-obtain data.

ACKNOWLEDGMENTS

It gives me great pleasure to thank Dr. Wu Li for being an excellent advisor. This thesis gives only a small indication of the profound influence his professional guidance and encouragement has had on me. I would particularly like to thank him for sharing his ideas and experience on mathematics and science in general.

I would like to thank all members of my committee for their time, comments, and support. In particular, I thank Professor John J. Swetits not only for all his attention but also for the privilege of assisting to the excellency of his teaching. I thank Professor Fang Q. Hu for how much I learned from the classes I took from him, in particular the learning of a new language for me, MATLAB. I thank Professor Przemyslaw Bogacki and Professor Dayanand N. Naik for their helpful comments and suggestions.

Engineering insights on wing weight estimations provided by Andrew Hahn and Sharon Padula at NASA Langley Research Center are greatly appreciated.

Many thanks for all my colleagues and friends at Old Dominion University. In particular, I thank Brian for the friendship, support, good advices, and love for soccer. I thank also my roommates and friends Gustavo, Nelson, and Kristine for understanding me, spoiling me, and making me gain a few pounds.

The love, care, and support of Isabel gave me the strength needed to go through all the obstacles of this long journey of three years apart. Thank you for everything. I thank also all the support and care of my family as well as the support given by all my friends in Portugal.

I would like to thank also Professor Luís Nunes Vicente and Professor Michael Wagner for their encouragement for me to study abroad. Without their help and support my studies in the United States would not have been possible.

Financial support from Portuguese Catholic University is gratefully acknowledged.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter I

INTRODUCTION

System analysis is a multidisciplinary, constrained, optimization process that tries to find a solution that best satisfies a set of requirements. For simple systems, this process can be relatively efficient (few function evaluations) and quick (short execution time). For very complex systems, such as aircraft design, the sheer number of analyses needed and the difficulty of the individual function evaluations can quickly make the large number of function evaluations needed impractical. Often, the data generated in the function evaluation are poorly behaved, in terms of both execution completion and the smoothness of the results, which then causes the efficiency and robustness of the optimization to suffer.

In the past, when only simple analytic approaches severely limited the problems that could be analyzed, data collected from tests were regressed with the aid of engineering theories to form the semi-empirical handbook methods familiar to designers. These regressions had many advantages, such as low input detail requirements as well as the ability to embody many detailed and difficult-to-assess considerations into an average state-of-the-art and very low calculation requirements. In fact, the requirements were so low that they could be, and were, calculated by hand (see [1]). These regressions also had many drawbacks, such as limited ranges of applicability and a strong dependency on the database from which they were regressed. This meant that new and unusual concepts often became unanalyzable, which required new tests to expand the database. These tests were time consuming, difficult, and expensive.

Attempts to improve the applicability of system analysis has centered around high fidelity numerical calculation because the calculation can generate similar data with much less time and cost for some disciplines (such as fluid dynamics) than the traditional tests (such as wind tunnel tests). High fidelity calculation involves running computer simulation code to generate numerical solutions that accurately approximate the true system responses, which could be validated by experiments. However, some high fidelity calculations still can not be included inside of the optimization loop because they are much too costly, or require human intervention to complete, or are still inadequate for the task.

This dissertation follows the style of *American Institute of Aeronautics and Astronautics.*

What is needed is a more general, robust, and rigorous regression method that takes multivariate data, generated from any source, and creates a fitting function that can be evaluated quickly, robustly, and accurately. The regression process should automatically identify key parameters that the result strongly depends on, would preferably not depend on theoretical knowledge of the physics, be able to handle sparse data sets, be able to handle poorly behaved data sets, and would be easy to implement.

A typical, well understood regression problem was chosen to try out some new methods to see how well they apply and to see how well they compare to traditional regression methods. The problem chosen was the determination of a wing weight estimation model given a database of actual aircraft wing configurations.

This thesis develops an automatic procedure of using interpolation methods for construction of an approximation of the relationship between the actual wing weight and various key configuration parameters of wing by using actual wing weight data of 41 subsonic transports.

I.1 CONTENTS OF THE THESIS

A standard approximation procedure can usually be decomposed into four steps [2]: (i) data generation and variable screening, (ii) fitting the data by a parametric function model, (iii) tuning intrinsic model parameters by using cross-validation, and (iv) verification of the constructed approximation. The focus of the thesis is on these four key technical issues in practical use of approximation methods. One controversial topic is the assessment of the constructed approximations, which is of great importance to practitioners but depends too much on subjective judgment. Some formal approaches for the assessment will be proposed and analyzed.

Data generation in the approximation procedure is mainly for selection of data sites of the input vector when the corresponding response is calculated by a computer simulation code, and is not needed for wing weight data fitting that uses only historical data. The rest of the topics will be discussed in this thesis. The following chapters describe an automatic procedure for construction of wing weight estimation models based on general approximation methods, whereas a user may receive warnings by the automatic procedure and exercise some control options to avoid potential failures of the procedure.

The thesis is organized as follows. In the next chapter we briefly describe the

wing weight data fitting problem. Chapter III is devoted to principal component analysis. Variable screening is explored in chapter IV. In chapter V we describe radial basis function interpolation methods. Model parameter tuning and other data fitting methods are presented in chapters VI and VII, respectively. Comparison of constructed approximations is given in chapter VIII. In the last chapter we have the conclusions. For definitions of wing configuration parameters, see the book by Raymer [1].

I.2 NOTATION

A	aspect ratio of wing, i.e., b^2/s
b	wingspan
c_m	mean chord of wing, i.e., s/b
c_r	root chord of wing at fuselage intersection
c_t	tip chord of wing
f	theoretical wing weight function
g	approximation of f
n	number of input variables
N	number of data points
s	plan area of wing
t_r	thickness of airfoil at fuselage intersection
t_t	thickness of airfoil at wingtip
t_r/c_r	thickness/chord ratio of airfoil at fuselage intersection
t_t/c_t	thickness/chord ratio of airfoil at wingtip
$[t/c]_m$	average thickness/chord ratio, i.e., $(t_r/c_r + t_t/c_t)/2$
$\mathbf{x}$	column vector of input variables $x_1, \ldots, x_n$
x_i	the ith component of column vector $\mathbf{x}$
x_i^j	the ith component of column vector $\mathbf{x}^j$
w	actual wing weight
$\bar{w}$	estimated or calculated wing weight

w_{to}	gross takeoff weight of aircraft
$\mathbf{1}$	column vector of ones
λ	taper ratio of wing, i.e., c_t/c_r
Λ	wing sweep angle in radian
μ	ultimate load
σ_i	estimated standard deviation of variable x_i
φ	radial basis function (RBF)

Subscripts and Superscripts

i	index for ith component of vector
j	index for data point
k	index for iteration or iterate
T	transpose of vector or matrix

Chapter II

WING WEIGHT DATA FITTING PROBLEM

II.1 INTRODUCTION

For system analysis of conceptual design of aircraft, one important task is to resize a conceptual aircraft for a mission analysis. To conduct a mission analysis of a resized aircraft, system analysts must estimate the gross takeoff weight w_{to} of the aircraft. Specifically, one commonly resized component of aircraft is wing. As a result, system analysts need a relationship between the wing weight w and sizing parameters of wing (such as s, b, λ, and Λ). Ardema et al. [3] describe a variety of methods to construct weight estimation of transport aircraft – from empirical regression to classical plate theory. In particular, they show how to use the beam theory structural analysis for fuselage and wing structural weight estimation. Linear and power regression methods are used by Ardema et al. (see [3, pp. 18–24]) to adjust the estimated structural weight to the actual structural weight for eight subsonic transports.

The objective of this thesis is to develop an automatic procedure of using interpolation methods for construction of an approximation of the relationship between the actual wing weight and various key configuration parameters of wing by using actual wing weight data of 41 subsonic transports. Such a procedure is called an empirical approach by Ardema et al [3]. However, system analysts usually reject the idea of using a general regression or approximation model for wing weight estimation, because a general model lacks any engineering insight and usually leads to non-physical weight estimation formula that gives negative wing weight or exhibits non-monotonicity of wing weight versus some key configuration parameter such as s. The best practices in empirical regression for wing weight estimate are based on heuristic regression models that incorporate some engineering understanding of the weight relationship. For example, the two best empirical regression models for the given wing weight data of 41 subsonic transports are the geometry model:

$$\bar{w} = \alpha_1 \Big[\mu^{\alpha_2} (0.01b)^{\alpha_3} \left(10^{-3}s\right)^{\alpha_4} (t_r)^{\alpha_5} (0.1c_r)^{\alpha_6} (\cos\Lambda)^{\alpha_7} (0.1c_t)^{\alpha_8} \left(10^{-5}w_{\text{to}}\right)^{\alpha_9} \Big] \tag{II.1}$$

and the ratio model:

$$\bar{w} = \bar{\alpha}_1 \Big[\mu^{\bar{\alpha}_2} A^{\bar{\alpha}_3} s^{\bar{\alpha}_4} ([t/c]_m)^{\bar{\alpha}_5} (\cos\Lambda)^{\bar{\alpha}_6} (1+\lambda)^{\bar{\alpha}_7} \left(10^{-3}w_{\text{to}}\right)^{\bar{\alpha}_8} \Big], \tag{II.2}$$

where $\alpha_1, \ldots, \alpha_9$ or $\bar{\alpha}_1, \ldots, \bar{\alpha}_8$ are determined by least squares fitting of the data. The engineering intuition behind these wing weight models is that the wing weight is a monotone function with respect to each of the configuration parameters in the model and its range is from 0 to ∞. However, there are two limitations of these two wing weight models: (i) the models are based on system analysts' knowledge of subsonic transports and it is nontrivial to derive similar empirical regression models for other types of aircrafts, and (ii) the models are not flexible enough to fit the wing weight data for the 41 subsonic transports. Fig. 1 shows more than 10% errors in the wing weight estimation by the best fit of each of these two engineering wing weight models. Moreover, later on, we will see that these models do not exhibit the expected weight growth trends with respect to changes of some key configuration parameters (such as b).

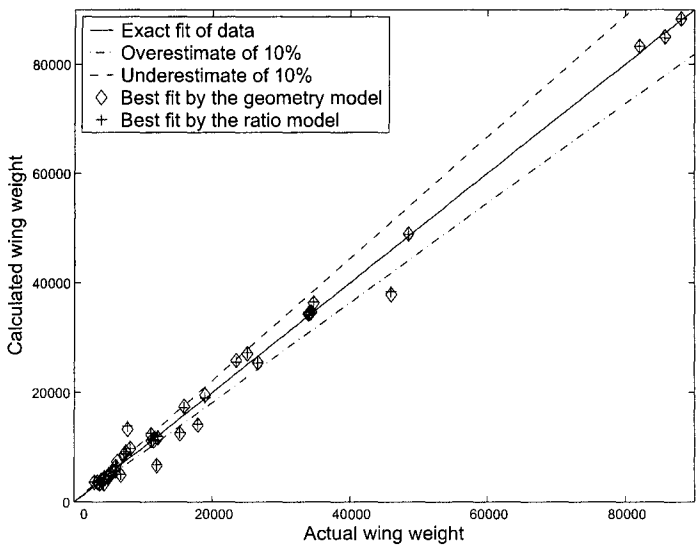

Figure 1: Regression errors of wing weight fitting by the geometry model and the ratio model.

Fitting the wing weight data by either the geometry model or the ratio model is a nonlinear least squares problem that may have many local optimal solutions. The best fitting depends on the initial choice of the regression parameters $\alpha_1, \ldots, \alpha_9$ or $\bar{\alpha}_1, \ldots, \bar{\alpha}_8$. Fig. 1 shows the best fitting computed by the nonlinear optimization code **lsqnonlin** in MATLAB. The maximums of relative fitting errors are 56.44%

and 57.55% for the geometry model and the ratio model, respectively. There is no decisive advantage of one model over the other if the goodness-of-fit is the decision criterion. It is important to provide analysts with useful information about the prediction behaviors of these wing weight models (instead of fitting errors), that helps the analysts choose an appropriate prediction model based on the characteristics of configuration design study at hand.

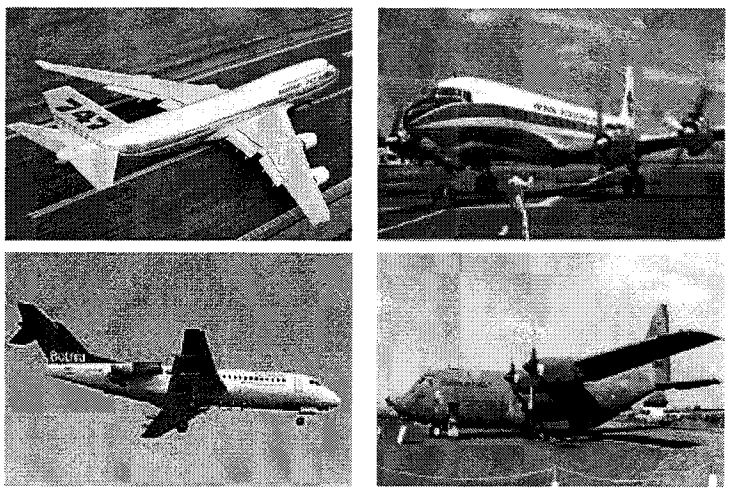

Figure 2: A variety of subsonic transports in the data set.

There are many studies [3, 4] on building approximation models for weight estimation. So far, useful weight models, such as Eqs. (II.1) and (II.2), are mainly derived from knowledge and insight of experienced engineers, instead of rigorous principles of physics. In some cases, useful weight estimation models are considered proprietary information not to be shared with the public. Weight information of existing aircrafts is not necessarily available to the public. System analysts at NASA Ames Research Center were able to collect weight information of 41 subsonic transports including Boeing 747, Douglas DC-7C, Fokker F-28 twin engine jet liner, and Lockheed C-130B cargo aircraft (see Fig. 2). This set of weight data allows the current study of benefits and limitations of general approximation methods for building a wing weight estimation model.

Each wing weight data point consists of the actual wing weight w and relevant key configuration parameters: $A, b, c_m, c_r, c_t, s, [t/c]_m, t_r, t_r/c_r, t_t, t_t/c_t, w_{\text{to}}, \lambda, \Lambda$, and

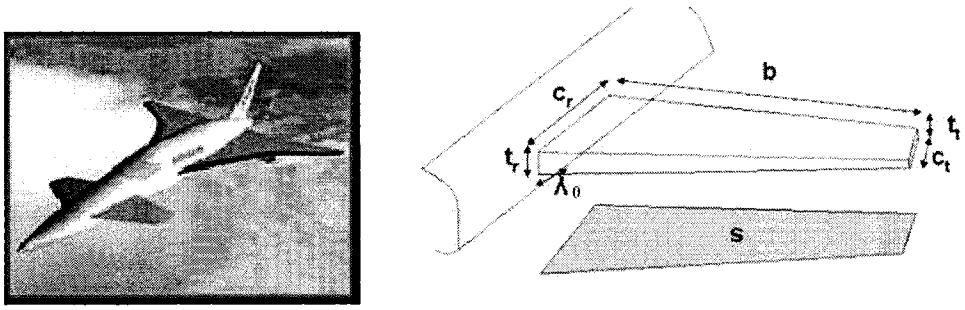

Figure 3: Airplane wing geometry parameters.

μ. These parameters can be regrouped in three categories: (1) wing geometry parameters including chord length at root (c_r), chord length at tip (c_t), span (b), reference area (s), thickness at root (t_r), thickness at tip (t_t), and sweep angle ($\Lambda = |90 - \Lambda_0|$); (2) wing aerodynamics parameters including taper ratio ($\lambda = c_t/c_r$), aspect ratio ($A = b^2/s$), mean chord of wing ($c_m = s/b$) , thickness-to-chord ratio at root (t_r/c_r), and thickness-to-chord ratio at tip (t_t/c_t); and (3) wing structure parameters including gross takeoff weight of aircraft (w_{to}) and wing load factor (μ). Fig. 3 shows the wing geometry parameters for a trapezoidal approximation of the actual wing. A detailed explanation of wing configuration parameters can be found in Raymer's book [1].

The goal is to construct a weight estimation model $\bar{w} \approx w$, where $\bar{w}$ is a function of all or a subset of the configuration parameters. Notice that some configuration parameters are related, e.g., $t_t/c_t + t_r/c_r = 2[t/c]_m$ and $A = b^2/s$. There are many different ways to select a set of independent configuration parameters, such as the two sets of parameters in Eqs. (II.1) and (II.2). Even though replacing b by A in Eq. (II.1) yields a mathematically identical model with appropriate choices of model parameters, such a replacement will lead to a completely different regression model if a general approximation model is used. For example, if a general quadratic polynomial P_2 is used as a regression model, then $P_2(A, s) = a_0 + a_1 A + a_2 s + a_3 A^2 + a_4 As + a_5 s^2$ and $P_2(b, s) = \bar{a}_0 + \bar{a}_1 b + \bar{a}_2 s + \bar{a}_3 b^2 + \bar{a}_4 bs + \bar{a}_5 s^2$ are two completely different regression models no matter what are the values of a_i and $\bar{a}_i$. Similarly, each choice of configuration parameters as input variables of a general approximation

model (such as polynomials, radial basis functions, and Kriging models) will lead to a new regression model. As a consequence, one must exercise caution in determination of which configuration parameters should be used as the input variables.

Recently, Li and Padula [2] did a survey of approximation methods that might be useful for conceptual design of complex systems. However, the survey didn't give any specific example on how the approximation methods can be used in conceptual design. In this thesis, the wing weight approximation problem is used to show feasibility of using approximation methods in construction of a wing weight estimation formula for conceptual design of subsonic transports. In particular, this thesis documents the lessons learned from fitting the 41 wing weight data points by three types of interpolation methods – least polynomial interpolation, radial basis function (RBF) interpolation, and Kriging interpolation.

II.2 CHALLENGES IN WING WEIGHT DATA FITTING

The wing weight data set was collected over a period of time and will be expanded when new weight statements of subsonic transports become available. Such dynamic characteristics of the data set requires a relatively easy way to generate a wing weight estimation model to capture the trend in the updated data set, when Eqs. (II.1) and (II.2) become inadequate as estimation models.

A data fitting model can only be as good as the data in representation of a mathematical relationship of the data attributes. While expert knowledge could be extremely helpful in choosing a practical regression model, it is important to eliminate unjustifiable subjective decisions when the data is fitted by general approximation or regression models.

Because the wing weight data is reliable, only interpolation methods will be considered for the data fitting. In other words, each wing weight estimation model will reproduce the actual wing weight for 41 subsonic transports. However, such an exact fit of the data has no useful purpose for the sizing of wing in conceptual design phase. System analysts are mostly interested in whether an estimation model captures the weight growth trends "correctly" between and beyond the known data points. There is no physics-based criterion for verification of a correct solution; instead, expert opinions determine whether a mathematical solution is useful in practice. Nevertheless, strategies of avoiding unjustifiable subjective decisions will be discussed.

II.3 MODELING OF WING WEIGHT DATA FITTING PROBLEM

Before getting into details of the steps of the approximation procedure, notations to describe the data interpolation are needed. Let $f(\mathbf{x})$ be the true response to a given input vector $\mathbf{x}$ (of n components) such that the value of f is only known at a set of N input vectors $\mathbf{x} = \mathbf{x}^1, \ldots, \mathbf{x}^N$, i.e., only $f_k = f(\mathbf{x}^k)$ $(k = 1, \ldots, N)$ are known. An interpolation model $g(\mathbf{x}) = \sum_{j=1}^{N} \alpha_j \varphi_j(\mathbf{x})$ is used as an approximation of $f(\mathbf{x})$, where α_j are the coefficients to be determined by interpolation conditions $g(\mathbf{x}^k) = f_k$ or $\sum_{j=1}^{N} \alpha_j \varphi_j(\mathbf{x}^k) = f_k$ $(k = 1, \ldots, N)$, and $\varphi_1, \ldots, \varphi_N$ are the basis functions depending on the choice of interpolation methods. The coefficient matrix of the linear equations $\sum_{j=1}^{N} \alpha_j \varphi_j(\mathbf{x}^k) = f_k$ $(k = 1, \ldots, N)$ is called the interpolation matrix.

For multivariate data fitting problems, it is not easy to decide what should be potential input variables. For the wing weight approximation problem, the input variables are usually the sizing parameters required in conceptual design of aircraft. The regression models (II.1) and (II.2) indicate system analysts' preference of input variables. One objective of the weight data fitting study is to understand whether analysts' choice is justifiable or can be reproduced by a variable screening method, which identifies the input variables that have a significant influence on the response. To avoid missing any important input variable, all the configuration parameters of wing (including the ratios) will be included as potential input variables of $g(\mathbf{x})$, i.e., $\mathbf{x}$ is a vector of 15 configuration parameters: $A, b, c_m, c_r, c_t, s, [t/c]_m, t_r, t_r/c_r, t_t, t_t/c_t, w_{\text{to}}, \lambda, \Lambda$, and μ. Note that the engineering insight of using $\cos(\Lambda)$ instead of Λ for wing weight estimation is intentionally ignored in this study to see whether general approximation methods without engineering insight is capable of generating a useful wing weight estimation formula. The "ratios" are included as potential input variables because they are native configuration parameters of wing.

The first step of the approximation procedure is to find out which of the 15 variables are important for a wing weight estimation model and whether there is any collinearity of the input variables. This step tries to reduce the dimension of the input space of the data fitting problem. Two approaches will be used for dimension reduction of the input space: principal component analysis (PCA) and variable screening. Both approaches require preprocessing of the data.

II.4 PREPROCESSING OF DATA

The given wing weight data of 41 subsonic transports had some errors. For example, some recorded taper ratio is not the same as c_t/c_r and some recorded thickness-to-chord ratio at root is not the same as t_r/c_r. A careful examination of the data file reveals the nine source input attributes of the data: $A, c_r, c_t, s, t_r, t_t/c_t, w_{\text{to}}, \Lambda$, and μ. Based on these nine attributes, the remaining six data attributes are uniquely determined by the mathematical relationships among the attributes. For example,

$$b = \sqrt{A \cdot s}, \quad \lambda = c_t/c_r, \quad \text{and} \quad c_m = s/b.$$

A standard data normalization approach is to scale each component x_i by an estimation of its standard deviation σ_i calculated from the data:

$$\sigma_i = \sqrt{\frac{\sum_{j=1}^{N}(x_i^j - \text{ave}(x_i))^2}{N-1}}, \quad \text{with} \quad \text{ave}(x_i) = \frac{1}{N}\sum_{j=1}^{N} x_i^j.$$

Index	Variable	Min	Max	Mean (ave(x_i))	Deviation (σ_i)
1	A	0.3	12.4	8.86	2.3
2	b	26.11	222.7	122.1	40.22
3	c_m	7.78	86.17	16.26	12.83
4	c_r	11.15	54.39	22.2	10.66
5	c_t	3.62	16.16	7.35	2.96
6	s	542.5	8,200	2,019	1,589
7	t_r	1.56	9.75	3.42	1.45
8	t_t	0.34	1.65	0.8	0.25
9	t_r/c_r	0.11	0.22	0.16	0.03
10	t_t/c_t	0.06	0.17	0.12	0.03
11	$[t/c]_m$	0.08	0.18	0.13	0.03
12	w_{to}	26,000	800,000	163,806	175,787
13	λ	0.20	0.61	0.35	0.1
14	Λ	0	55	11.91	15.83
15	μ	3.75	5.3	4.06	0.46

Table 1: Estimated means and standard deviations of configuration variables.

Note that system analysts need to be warned of small values of σ_i, say, less than ten percent of the mean value ave(x_i). There are two reasons for such a small value of σ_i: (i) the actual range of the variable x_i is about the same magnitude as σ_i or

(ii) there is not enough data to model the change of the response with respect to x_i. Expert knowledge can be used to determine which case it is. For example, in Table 1, the smallest ratio of $\sigma_i/\text{ave}(x_i)$ is the ultimate load μ, which is usually determined by FAA regulation and has a range from 3.75 to 5.3 for subsonic transports with estimated standard deviation of 0.46 and mean of 4.06. A warning will help analysts to discover that among 41 subsonic transports, 26 of them have the same ultimate load of 3.75, i.e., there is not much variation in μ for the given data set. Therefore, any relationship between the ultimate load and the wing weight based on the given data set might be questionable. In this study, the estimated standard deviation of μ is accepted for scaling, which may inflate the significance of μ in both the PCA and variable screening analysis.

Scaling each data attribute by its estimated standard deviation also helps the initial formulation of the approximation problem. To illustrate how scaling affects the problem formulation in practice, the cubic RBF interpolation is used as an example. The cubic RBF model is defined by

$$g(\mathbf{x}) = \sum_{j=1}^{N} \alpha_j \varphi(\|\mathbf{x} - \mathbf{x}^j\|),$$

where $\varphi(\|\mathbf{x}-\mathbf{x}^j\|)$ represents $\varphi_j(\mathbf{x})$ in the interpolation model, $\varphi(t) = t^3$ is the cubic RBF, and $\|\mathbf{x} - \mathbf{x}^j\|$ is a parameterized distance between $\mathbf{x}$ and $\mathbf{x}^j$ defined as

$$\|\mathbf{x} - \mathbf{x}^j\| = \sqrt{\sum_{i=1}^{n} |\theta_i| \left(\frac{x_i - x_i^j}{\sigma_i}\right)^2}. \tag{II.3}$$

The scalars $\theta_1, \ldots, \theta_n$ in Eq. (II.3) are the model tuning parameters that will be determined by a cross-validation method for the best prediction model of the given data. Mathematically, one could rewrite $\|\mathbf{x} - \mathbf{x}^j\|$ as

$$\|\mathbf{x} - \mathbf{x}^j\| = \sqrt{\sum_{i=1}^{n} |\bar{\theta}_i| \left(x_i - x_i^j\right)^2}, \quad \text{with } \bar{\theta}_i = \frac{\theta_i}{\sigma_i^2}. \tag{II.4}$$

In practice, starting without any scaling (i.e., $\bar{\theta}_i = 1$ in Eq. (II.4)) may lead to ill-conditioning of the interpolation problem even though the mathematical theory [5, 6] guarantees the existence of a unique cubic RBF interpolant for any given data points $(\mathbf{x}^1, f_1), \ldots, (\mathbf{x}^N, f_N)$. For eight input variables given in (II.1) and forty-one data points, the condition number of the unscaled interpolation matrix is 4.4×10^{14},

while the scaled interpolation matrix (corresponding to $\theta_i = 1$ in Eq. (II.3)) has a condition number of 1.1×10^5. The purpose of using two sets of scaling parameters in Eq. (II.3) is to allow a nondimensional initial choice of $\theta_i = 1$.

Chapter III

PRINCIPAL COMPONENT REGRESSION

III.1 INTRODUCTION

For a limited number of historical or measurement data points in a high-dimensional input space, a principal component analysis (PCA) is recommended to check any collinearity of the input attributes of the data points. Assume that there exists in fact collinearity among the input vectors $\mathbf{x}^1, \ldots, \mathbf{x}^N$ in $\mathbb{R}^n$. Then the input vectors are scattered around a r-dimensional subspace of $\mathbb{R}^n$ spanned by a set of orthogonal vectors $\mathbf{u}^1, \ldots, \mathbf{u}^{\bar{r}}$ with $\bar{r} < n$. These orthogonal vectors $\mathbf{u}^1, \ldots, \mathbf{u}^{\bar{r}}$ can be generated by PCA and will be called the *feature vectors* for the vectors $\mathbf{x}^1, \ldots, \mathbf{x}^N$. The main applications of PCA are: (i) reduce the number of variables; (ii) detect structure in the relationships between variables; and (iii) transform correlated variables into uncorrelated ones. In other words, PCA is applied as a data reduction or structure detection and correction method.

The importance of using PCA for irregularly distributed input vectors was discussed in [2, section 3.6] where we can find the following example that illustrates the importance of using PCA: all the input vectors fall in a straight line that is not parallel to any coordinate axis, e.g., $\mathbf{x}^1, \ldots, \mathbf{x}^N$ are distributed along a line with a unit direction $\mathbf{u}^1$. Thus, $\mathbf{x}^j = \alpha_j \mathbf{u}^1$ for some scalar α_j with $j = 1, \ldots, N$. For this worst case example, variable screening methods will not work because the collinear input vectors suggest the rate of the changes in the response with respect to the corresponding changes in each of the components of the input vector is the same when all the components of $\mathbf{u}^1$ are equal. Therefore, all input variables are equally important. In this case, the only information given by the data points is how the response changes when the input vector changes along the line with the direction vector $\mathbf{u}^1$. Thus, any meaningful approximation should only capture the trend of the response in the feature direction $\mathbf{u}^1$. That can be accomplished by reducing the input space to one-dimensional feature space ($\bar{r} = 1$) and solving the corresponding fitting problem in that space to construct an approximation $\hat{f}(\alpha)$. The relationship $\alpha = \mathbf{x}^T \mathbf{u}^1$ allows us to recover the corresponding approximation $f(\mathbf{x}) \approx \hat{f}(\mathbf{x}^T \mathbf{u}^1)$ in the original input space from the constructed approximation in the feature space. Fig. 4 shows a similar example in $\mathbb{R}^2$ on how PCA can be used to construct approximations to

capture the data trends in the first feature direction.

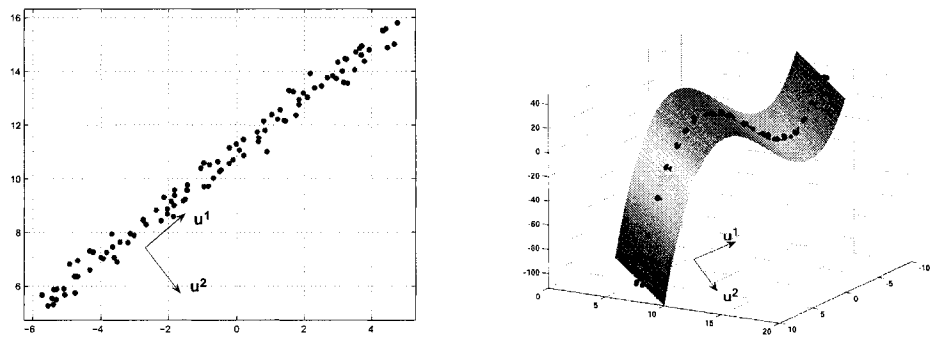

Figure 4: Capture data trend in the first feature direction of historical and measurement data.

The dimension reduction illustrated in the previous examples works for any collinear or nearly collinear data distribution in the input space. By applying PCA to the input vectors $\mathbf{x}^1, \ldots, \mathbf{x}^N$, we can treat the response as a function defined on the feature space $\mathbb{R}^{\bar{r}}$ with a reduced dimension $\bar{r}$ ($< n$) and then solve the approximation problem by fitting the transformed data in the feature space. We can then recover the corresponding approximation found in the feature space to the original input space. Regression methods based on PCA are called principal component regression (PCR).

III.2 PCA OF WING WEIGHT DATA

In this section we discuss the PCA of the wing weight data.

The wing data of each subsonic transport have 15 configuration parameters: $A, b, c_m, c_r, c_t, s, [t/c]_m, t_r, t_r/c_r, t_t, t_t/c_t, w_{\text{to}}, \lambda, \Lambda$, and μ. Because $[t/c]_m = (t_r/c_r + t_t/c_t)/2$, the three configuration parameters $[t/c]_m, t_r/c_r$, and t_t/c_t are linearly dependent. The redundant parameter $[t/c]_m$ is considered as an input variable due to analysts' preference of using $[t/c]_m$ as an input variable instead of t_r/c_r and t_t/c_t (see Eq. (II.2)).

The PCA of the 41 subsonic transport wing data is done as follows. First, relabel the 15 parameters as variables $x_1, \ldots, x_n$ ($n = 15$) and the 41 wing configurations as $\mathbf{x}^1, \ldots, \mathbf{x}^N$ ($N = 41$). Scale each variable by its estimated standard deviation:

$\hat{x}_i^j = x_i^j/\sigma_i$. (Note that different orders of magnitude of the components of $\mathbf{x}^1, \ldots, \mathbf{x}^N$ may render the PCA of $\mathbf{x}^1, \ldots, \mathbf{x}^N$ useless for collinearity analysis.)

Next calculate the covariance matrix $\mathbf{C}$ of the scaled input vectors $\hat{\mathbf{x}}^1, \ldots, \hat{\mathbf{x}}^N$:

$$\mathbf{C} = \frac{1}{N-1} \sum_{j=1}^{N} \left[\hat{\mathbf{x}}^j - \text{ave}(\hat{\mathbf{x}})\right] \left[\hat{\mathbf{x}}^j - \text{ave}(\hat{\mathbf{x}})\right]^T, \tag{III.1}$$

where $\text{ave}(\hat{\mathbf{x}}) = \frac{1}{N} \sum_{j=1}^{N} \hat{\mathbf{x}}^j$. Then the following spectral decomposition of $\mathbf{C}$ can be used to analyze the collinearity of the input parameters: $\mathbf{C} = \sum_{j=1}^{n} \gamma_j \mathbf{u}^j (\mathbf{u}^j)^T$, where $\gamma_1 \geq \gamma_2 \geq \cdots \geq \gamma_n \geq 0$ are the eigenvalues of $\mathbf{C}$, and $\mathbf{u}^1, \ldots, \mathbf{u}^n$ are the corresponding unit eigenvectors.

For the 15 wing configuration parameters of the 41 subsonic transports, the eigenvalues of $\mathbf{C}$ are 7.952, 2.757, 1.349, 0.861, 0.676, 0.565, 0.341, 0.187, 0.157, 0.096, 0.031, 0.019, 0.005, 0.003, and 0.000. The last eigenvalue of 0 means linear dependence of the 15 configuration parameters due to the linear relationship $[t/c]_m = (t_r/c_r + t_t/c_t)/2$. In fact, the three components of $\mathbf{u}^{15}$ corresponding to $[t/c]_m, t_r/c_r, t_t/c_t$ are -0.45, -0.43, and 0.78, while the remaining components of $\mathbf{u}^{15}$ are zero (accurate up to two significant digits). The next two smallest eigenvalues 0.003 and 0.005 also indicate nearly collinear relationships among the 15 configuration parameters because of specific locations of the 41 input vectors in the 15-dimensional space. Such data-specific relationships are most likely to disappear when new wing weight data is added to the existing data set. However, one should exercise caution when using a weight prediction of a wing configuration represented by $\hat{\mathbf{x}}$ with relatively large absolute values of $(\hat{\mathbf{x}} - \text{ave}(\hat{\mathbf{x}}))^T \mathbf{u}^{13}$ or $(\hat{\mathbf{x}} - \text{ave}(\hat{\mathbf{x}}))^T \mathbf{u}^{14}$, because the data do not have much information on how the wing weight changes in terms of these two quantities.

If t_r/c_r and t_t/c_t are excluded from the list of wing configuration parameters, then the 13 eigenvalues of the corresponding $\mathbf{C}$ are 7.17, 2.17, 1.33, 0.758, 0.631, 0.312, 0.244, 0.179, 0.123, 0.047, 0.021, 0.016, and 0.004. It also suggests that one nearly collinear relationship could be used to reduce the dimension of the input space to 12, which is the same conclusion from the previous PCA.

If the PCA is applied to analyze the collinearity of the configuration parameters in the two engineering wing weight models, then the smallest eigenvalue of $\mathbf{C}$ is 0.146 for configuration parameters in model (II.2). However, for configuration parameters in model (II.1), the two smallest eigenvalues of $\mathbf{C}$ are 0.082 and 0.019, which indicates

a higher level of collinearity among the input variables in model (II.1) than that in model (II.2).

III.3 PROJECTION OF DATA TO REDUCED FEATURE SPACES

We call the unit vector $\mathbf{u}^j$, the jth feature vector of the sample data set $\hat{\mathbf{x}}^1, \ldots, \hat{\mathbf{x}}^N$ and the scalar $v_j = \hat{\mathbf{x}}^T \mathbf{u}^j$ the jth principal component of $\hat{\mathbf{x}}$. The number of positive eigenvalues of $\mathbf{C}$, $\bar{r}$, is the degree of freedom in the sampled input set and we can write each input vector $\hat{\mathbf{x}}^k$ as a linear combination of the $\bar{r}$ feature vectors $\mathbf{u}^1, \ldots, \mathbf{u}^{\bar{r}}$:

$$\hat{\mathbf{x}}^k = \text{ave}(\hat{\mathbf{x}}) + \sum_{j=1}^{\bar{r}} \left[(\hat{\mathbf{x}}^k - \text{ave}(\hat{\mathbf{x}}))^T \mathbf{u}^j \right] \mathbf{u}^j. \tag{III.2}$$

The value of each eigenvalue of $\mathbf{C}$, γ_j, indicates the significance the jth principal component of $\hat{\mathbf{x}}$ in representing the variance in the sampled input set. Therefore, the principal component analysis (PCA) is simply an ordination technique for describing the variation in a multivariate data set. The first axis (the first principal component) describes the most significant direction of variance in the sampled input set; the second describes the second most significant direction of variance in the sampled input set, and so forth, with each direction orthogonal to the preceding ones.

A standard dimension reduction technique is to use the r most significant feature vectors corresponding to the r largest eigenvalues of $\mathbf{C}$ for an approximate representation of the input vectors $\mathbf{x}^k$ instead of the exact representation formula (III.2). The approximation is based on the projection from the input space to the r-dimensional feature space:

$$P(\mathbf{x}) = \text{ave}(\mathbf{x}) + \sum_{j=1}^{r} [(\mathbf{x} - \text{ave}(\mathbf{x}))^T \mathbf{u}^j] \mathbf{u}^j. \tag{III.3}$$

If $r = \bar{r}$, then $P(\mathbf{x}^k) = \mathbf{x}^k$ for $k = 1, \ldots, N$. If $r < \bar{r}$, then the difference between (III.2) and (III.3), i.e., the difference between $\mathbf{x}^k$ and $P(\mathbf{x}^k)$, increases as the maximum value of γ_j for $j > r$ increases. If γ_j for $j > r$ are small, we can consider $\mathbf{x}^k \approx P(\mathbf{x}^k)$ as a vector in the reduced r-dimensional feature space without much loss of accuracy.

III.4 PCR FOR WING WEIGHT APPROXIMATION

PCR for wing weight approximation is a process of constructing wing weight estimation models by fitting the data in reduced feature spaces.

Suppose that the first r ($< n$) principal components $(\hat{\mathbf{x}} - \text{ave}(\hat{\mathbf{x}}))^T\mathbf{u}^1, \ldots, (\hat{\mathbf{x}} - \text{ave}(\hat{\mathbf{x}}))^T\mathbf{u}^r$ are chosen as the significant input variables. Let $\mathbf{v}^j$ be the vector with r components $v_1^j, \ldots, v_r^j$ defined by

$$v_i^j = (\hat{\mathbf{x}}^j - \text{ave}(\hat{\mathbf{x}}))^T\mathbf{u}^i \quad \text{for } i = 1, \ldots, r.$$

Then the reformulated problem of fitting N data points $(\mathbf{v}^1, f_1), \ldots, (\mathbf{v}^N, f_N)$ can be solved by using any regression method or the interpolation methods described in Chapters V and VII. If the solution of the reformulated problem is $\hat{g}(\mathbf{v})$, then the corresponding fitting of the data points in the $\mathbf{x}$-space is the following:

$$\bar{w} = \hat{g}\Big((\hat{\mathbf{x}} - \text{ave}(\hat{\mathbf{x}}))^T\mathbf{u}^1, \ldots, (\hat{\mathbf{x}} - \text{ave}(\hat{\mathbf{x}}))^T\mathbf{u}^r\Big), \tag{III.4}$$

where $\hat{x}_i = x_i/\sigma_i$ are scaled variables.

There are two main advantages for fitting the data in reduced feature spaces: (i) $\mathbf{v}$ has fewer components than $\mathbf{x}$ which reduces the impact of curse of dimensionality [7]; (ii) significant uncorrelated variances of the input vectors in the directions $\mathbf{u}^1, \ldots, \mathbf{u}^r$ ensure that sufficient data information are available for modeling of the relationship between the response and independent variables $v_1, \ldots, v_r$.

The wing weight estimation model (III.4) has two sources of approximation errors: (i) errors due to dimension reduction, i.e., errors due to approximate representation (III.3) of $\mathbf{x}$, and (ii) errors inherited from fitting $\hat{g}(\mathbf{v})$ to the data in the feature space. However, we will see later that PCR can produce much better approximations of the response than directly fitting the data in the original space because the fitting process is customized for the given data.

If γ_r is much greater than 0 and $\gamma_{r+1} = \ldots = \gamma_n = 0$, then all the existing input vectors have the same jth principal component for $j = r+1, \ldots, n$, i.e., $(\hat{\mathbf{x}}^1)^T\mathbf{u}^j = \ldots = (\hat{\mathbf{x}}^N)^T\mathbf{u}^j$ for $j = r+1, \ldots, n$. In this simple case, the existing input vectors spread out in an r-dimensional subspace and the data do not provide enough information for building any approximation model with more than r degree of freedom in the input. In the simplest case when $n = 2$ and all the existing input vectors have the same value of the x_1-coordinate, any nonconstant relationship between an approximation model and the input variable x_1 is unjustified and should be avoided, which is exactly the purpose of the PCR process described above.

In real world, it is not easy to choose r. For example, based on the PCA of all 15 configuration parameters, one could choose $r = 10$ or $r = 12$. The key issue is

whether one wants to extract a functional relationship between the change of the input vector $\hat{\mathbf{x}}$ along a direction $\mathbf{u}^j$ and the response $f(\hat{\mathbf{x}})$ even if the existing input vectors $\hat{\mathbf{x}}^1, \ldots, \hat{\mathbf{x}}^N$ have a small variation along the direction $\mathbf{u}^j$. Analysts can view the distribution of the eigenvalues of $\mathbf{C}$ and make a few plausible choices of r, build approximations for the different values of r, and use a systematic evaluation process to down select the approximation that is most appropriate for the analysis task at hand.

Chapter IV

APPLICATION OF VARIABLE SCREENING METHODS

IV.1 INTRODUCTION

A standard variable screening process identifies a subset of the input variables $x_1, \ldots, x_n$, denoted by $\bar{\mathbf{x}}$, that have significant influences on the response $f(\mathbf{x})$. In other words, if the change of $f(\mathbf{x})$ with respect to a variable x_i is negligible, then eliminate x_i from the input vector. It is worth pointing out that variable screening should be applied in the r-dimensional feature space if the dimension of the input space has been reduced by the PCA, where the purpose of variable screening is to identify variables in $\mathbf{v}$ that have significant influences on the response. For the wing weight approximation problem, any variable screening method (such as ANOVA) that requires the values of the response for specific input vectors is not applicable. The main effects estimate (MEE) method, proposed by Tu and Jones [8], generally requires a uniform distribution of the existing input vectors in a rectangular domain of the input space, while the forward or backward variable selection method is mainly to determine the explanatory power of input variables of linear regression models (such as polynomial models) that are independent of the data distribution. Rech *et al.* [9] proposed a variable selection technique based on polynomial approximations of the nonlinear regression model, but this method only works well when the response can be approximated by a low-degree polynomial. Moreover, their rule-of-the-thumb is that there are at least about four times as many observations as the number of the coefficients in polynomials. For forty one wing weight data points and the geometry model (II.1) with eight variables, this rule-of-the-thumb leads to a poor linear polynomial approximation of the geometry model. Therefore, these variable screening methods cannot be applied to identify important input variables for wing weight estimation models based on the historical wing weight data.

However, the two-dimensional plots of the response with respect to the input variables are always helpful to gain an intuitive understanding of whether a particular input variable is important for modeling the response. Fig. 5 indicates that the wing weight w has some functional relationship with respect to each of the 15 configuration parameters except μ. Note that significant wing weight changes only occur at the same μ value of 3.75, suggesting that other configuration parameters determine the

wing weight instead of μ. In fact, for civil transports, μ is usually determined by FAA regulation and is not a design variable.

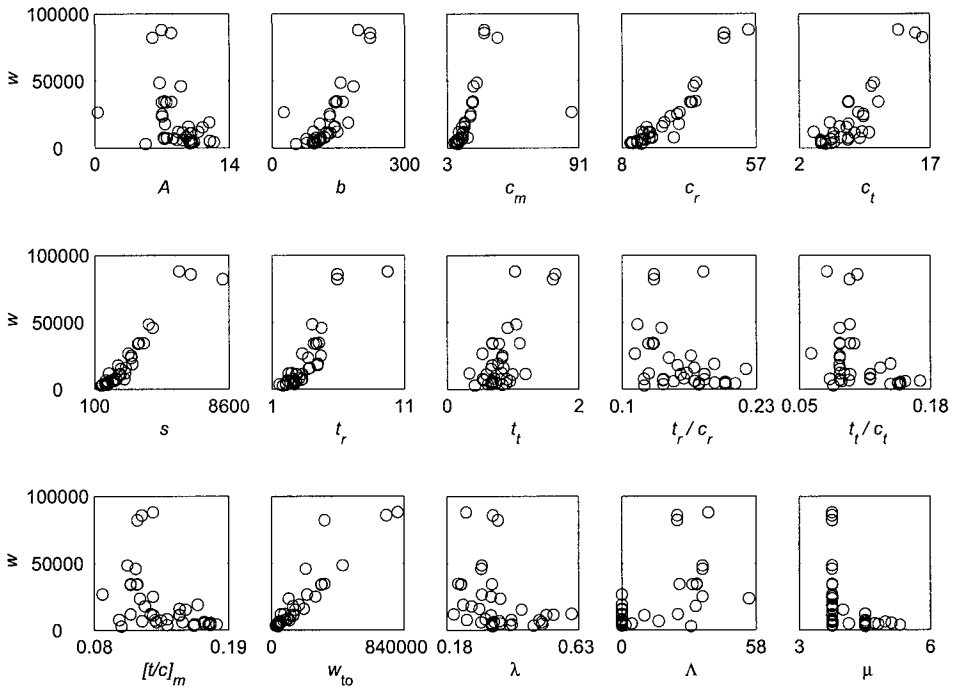

Figure 5: Two-dimensional plots of wing weight versus configuration parameters.

If we assume that the forward and backward variable selection methods are valid for variable screening in nonlinear models, then they can be formally applied for variable screening in wing weight data fitting by the geometry model. In general, under this assumption, the forward and backward variable selection methods can be formally applied for variable screening if the wing weight data is fitted by a regression model that is independent of data distribution. For demonstration purpose, these two methods are applied to check which input variable in the geometry model (II.1) is insignificant for wing weight estimation.

IV.2 FORWARD SCREENING

For convenience, relabel the eight variables in Eq. (II.1) as $x_1, \ldots, x_n$ (with $n = 8$). If the sample coefficient of determination (R^2) is used to measure the proportion of

the total variation in $f_1, \ldots, f_N$ explained by a given model, then the corresponding forward variable selection procedure can be described as follows.

Forward Variable Selection

1. Let $g_i(x_i)$ be the best fit of the simplified geometry model representing the relationship between the ith input variable in Eq. (II.1) and w, i.e., $g_i(x_i)$ is in the form of the univariate model obtained by setting the exponents of the terms not involving x_i to zero in Eq. (II.1).

2. Compute the sample coefficients of determination (R_i^2) for g_i:

$$R_i^2 = 1 - \frac{\sum_{j=1}^{N} \left(f_j - g_i(x_i^j)\right)^2}{\sum_{j=1}^{N} (f_j - \text{ave}(f))^2},$$

   where $\text{ave}(f) = \frac{1}{N}\sum_{j=1}^{N} f_j$. The quantity R_i^2 is the proportion of the total variation in $f_1, \ldots, f_N$ explained by the simplified geometry model and it can be used as a metric for ranking the significance of x_i in variation of the response.

3. If a simplified geometry model of k input variables is desirable, then the input variables corresponding to the k largest R_i^2 shall be selected as the significant input variables.

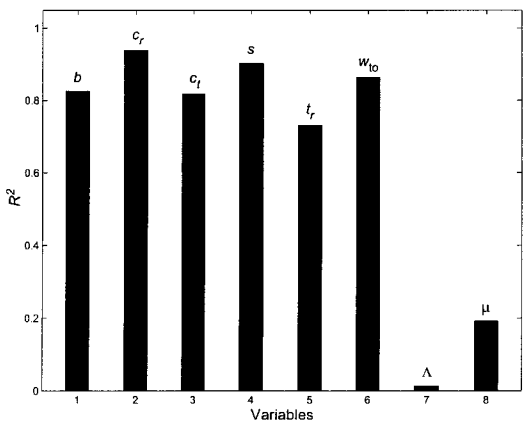

Figure 6: Forward variable selection for wing weight approximation by the geometry model.

This variable selection procedure is designed for linear regression problems, while the wing weight approximation by the geometry model is a nonlinear regression problem whose solution is very sensitive to the initial guess of the optimal solution. The nonlinear optimization code **lsqnonlin** in MATLAB was used to solve the nonlinear least squares problems and the results given in Fig. 6 represent the best solutions found by trying a few educated guesses (see MATLAB code in section A.1.1).

Fig. 6 shows the results of applying the forward variable selection to the wing weight data fitting problem by using the geometry model. The result suggests that μ is an insignificant variable in the geometry model for wing weight estimation, which is consistent with the observation from Fig. 5. However, it is puzzling to see that the sweep angle Λ is identified as the least significant variable. Later, the backward variable selection will contradict the conclusion of insignificance of Λ implied by the forward variable selection procedure.

IV.3 BACKWARD SCREENING

The backward variable selection procedure can be described as follows.

Backward Variable Selection

1. Let $g(\mathbf{x})$ be the best fit of the wing weight data by the geometry model (II.1).

2. Let $g_{-i}(\mathbf{x}_{-i})$ be the best fit of the simplified geometry model obtained by setting the exponent of the term involving x_i to zero in Eq. (II.1).

3. Compute the adjusted sample coefficients of determination ($\bar{R}^2$ and $\bar{R}^2_{-i}$) for g and g_{-i} $(i = 1, \ldots, n)$:

$$\bar{R}^2 = 1 - \frac{(N-1)\sum_{j=1}^{N}\left(f_j - g(\mathbf{x}^j)\right)^2}{(N-n)\sum_{j=1}^{N}(f_j - \text{ave}(f))^2}$$

and

$$\bar{R}^2_{-i} = 1 - \frac{(N-1)\sum_{j=1}^{N}\left(f_j - g_{-i}(\mathbf{x}^j_{-i})\right)^2}{(N-n+1)\sum_{j=1}^{N}(f_j - \text{ave}(f))^2}.$$

4. If the difference in adjusted sample coefficients of determination $\Delta\bar{R}^2_{-i} = \bar{R}^2 - \bar{R}^2_{-i}$ is nonpositive for some i, then the corresponding variable x_i could be removed from the input vector and the simplified geometry model would have $(n-1)$ variables.

5. Repeat the process with the simplified geometry model until the number of input variables becomes desirable or all ΔR^2_{-i} are positive.

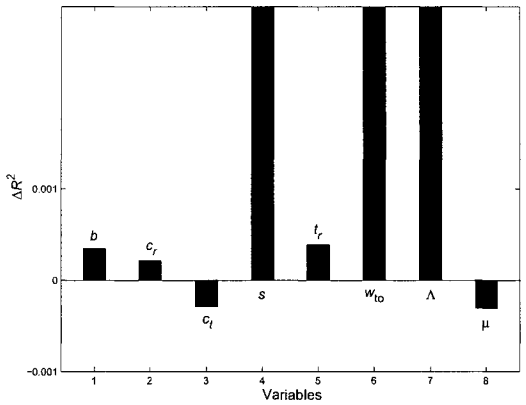

Figure 7: First iteration of backward variable selection for the geometry model.

Fig. 7 shows ΔR^2_{-i} from the first iteration of the backward variable selection for the geometry model and Fig. 8 shows ΔR^2_{-i} from the second iteration (where c_t was already removed). In the third iteration, all ΔR^2_{-i} are positive, thus the iteration process is terminated.

Again the nonlinear optimization code **lsqnonlin** in MATLAB was used to solve the nonlinear least squares problems and the results given in Figs. 7 and 8 represent the best solutions found by trying a few educated guesses (see MATLAB code in section A.1.2).

IV.4 CONCLUSION

Variable screening for nonlinear regression is a challenging problem if there are only a few historical or measured data points available. If the nonlinear regression model is independent of data distribution, one could formally use the forward and backward variable selection methods for variable screening, even though these methods were developed for variable screening of linear regression models. Together with the two-dimensional plots, these screening methods could provide some insight on significance of input variables.

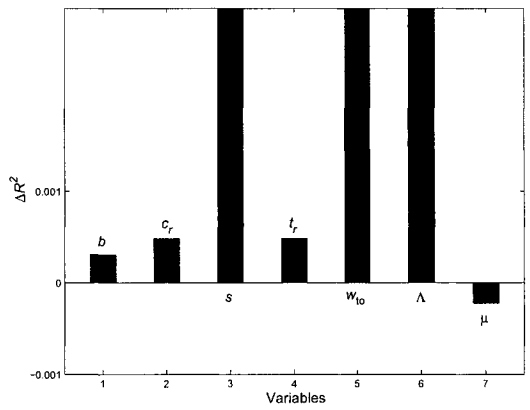

Figure 8: Second iteration of backward variable selection for the geometry model.

It is heuristic to use the adjusted R^2 in variable selection for nonlinear models. Further study is needed to understand the merit of this approach. However, for wing weight approximation by the geometry model, the consistency of the forward and backward variable selections in identifying the ultimate load μ as an insignificant variable is encouraging, because the values of μ for civil transports are mandated by FAA, not a design variable determined by engineers. The reason of including μ as a design variable is a legacy inherited from military aircraft weight estimation practices, where μ is a common parameter in the design tradeoff and there is considerably greater variation in the values of μ.

Chapter V

RADIAL BASIS FUNCTION INTERPOLATION

V.1 INTRODUCTION

For numerical approximation of multivariate functions, radial basis functions (RBFs) are very useful. For any finite data set in any finite dimensional space, one can construct an interpolation of the data by using RBFs. There is a wide range of applications where RBF interpolation methods can be successfully applied (see [10]). One interesting application of RBF interpolation is in medical imaging for skull defect repair [11].

In the following sections, we will formulate RBF interpolation problems, discuss the solvability of RBF interpolation problems, and introduce two related interpolation methods (Kriging and Gaussian process).

V.2 RBF INTERPOLATION PROBLEMS

Let $f(\mathbf{x})$ be a real-valued function of the input vector $\mathbf{x}$ defined on a subset Ω of $\mathbb{R}^n$ such that the value of f is given at N input vectors $\mathbf{x}^j$, $j = 1, \ldots, N$. Let $f_j \approx f(\mathbf{x}^j)$, $j = 1, \ldots, N$. For wing weight data fitting, f_j and $f(\mathbf{x}^j)$ represent the documented and actual wing weight for the jth transport in the data. In this case, f_j is almost the same as $f(\mathbf{x}^j)$. Therefore, it is desirable to construct a wing weight estimation model $g(\mathbf{x})$ such that $g(\mathbf{x}^j) = f_j$ for $j = 1, \ldots, N$. The interpolation requirement can be satisfied by RBF interpolation.

Interpolation functions generated from a RBF $\varphi(t)$ can be represented in the following form:

$$g(\mathbf{x}) = \sum_{j=1}^{N} \alpha_j \varphi(\|\mathbf{x} - \mathbf{x}^j\|), \tag{V.1}$$

where $\|\mathbf{x} - \mathbf{x}^j\|$ denotes the parameterized distance between $\mathbf{x}$ and $\mathbf{x}^j$ defined as

$$\|\mathbf{x} - \mathbf{x}^j\| = \sqrt{\sum_{i=1}^{n} |\bar{\theta}_i| \left(x_i - x_i^j\right)^2}, \tag{V.2}$$

and $\bar{\theta}_i$ are positive numbers.

The most important examples of RBF [5, 12, 13] are multiquadric $\varphi(t) = \sqrt{1 + t^2}$, thin plate spline $\varphi(t) = t^2 \ln t$, cubic spline $\varphi(t) = t^3$, and Gaussian $\varphi(t) = \exp(-t^2)$

(see Fig. 9). These RBFs can be used to model linear, almost quadratic, and cubic growth rates, as well as exponential decay, of the response for trend predictions.

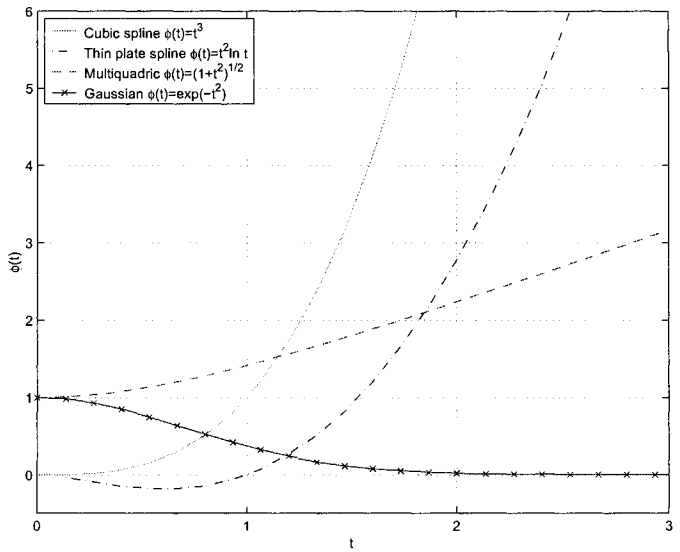

Figure 9: Graphs of radial basis functions.

For fixed positive parameters $\bar{\theta}_i$, the coefficients $\alpha_1, \ldots, \alpha_N$ in (V.1) can be calculated by solving the following linear system of interpolation equations:

$$\sum_{j=1}^{N} \alpha_j \varphi(||\mathbf{x}^k - \mathbf{x}^j||) = f_k, \quad \text{for } k = 1, \ldots, N. \tag{V.3}$$

One can rewrite (V.3) in matrix form as

$$\Phi \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{pmatrix}, \tag{V.4}$$

where Φ is the interpolation matrix defined as

$$\Phi = \begin{pmatrix} \varphi(||\mathbf{x}^1 - \mathbf{x}^1||) & \varphi(||\mathbf{x}^1 - \mathbf{x}^2||) & \ldots & \varphi(||\mathbf{x}^1 - \mathbf{x}^N||) \\ \varphi(||\mathbf{x}^2 - \mathbf{x}^1||) & \varphi(||\mathbf{x}^2 - \mathbf{x}^2||) & \ldots & \varphi(||\mathbf{x}^2 - \mathbf{x}^N||) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi(||\mathbf{x}^N - \mathbf{x}^1||) & \varphi(||\mathbf{x}^N - \mathbf{x}^2||) & \ldots & \varphi(||\mathbf{x}^N - \mathbf{x}^N||) \end{pmatrix}. \tag{V.5}$$

For multiquadric and Gaussian RBFs, a unique interpolant is guaranteed (i.e., Φ is a nonsingular matrix) even if the input vectors $\mathbf{x}^j$ are few and poorly distributed, provided only that the input vectors are all different when $N > 1$. But for cubic and thin plate spline RBFs, Φ might be singular. See [5] for an example of singular Φ when $\varphi(t) = t^3$. If $\varphi(t) = t^2 \ln t$, we can easily find an example where the interpolation matrix Φ is singular for a nontrivial set of distinct points $\mathbf{x}^1, \ldots, \mathbf{x}^N$. For example, let $\mathbf{x}^2, \ldots, \mathbf{x}^N$ be any different points on the sphere centered at $\mathbf{x}^1$ with radius 1. For this set of points, the first row and column of Φ consist of zeros, which implies the singularity of Φ.

A practical approach for constructing cubic and thin plate spline RBF interpolants is to add low-degree polynomials to interpolation functions in (V.1) and formulate an interpolation problem with constraints. That is, let $p(x) = \sum_{j=1}^{M} \beta_j p_j(\mathbf{x})$, where $p_1, \ldots, p_M$ form a basis of algebraic polynomials in $\mathbb{R}^n$ with degree at most m. Then interpolation functions are of the following form:

$$g(\mathbf{x}) = p(\mathbf{x}) + \sum_{j=1}^{N} \alpha_j \varphi(||\mathbf{x} - \mathbf{x}^j||). \tag{V.6}$$

The M extra degrees of freedom in $g(\mathbf{x})$ can be eliminated by forcing the following M constraints:

$$\sum_{j=1}^{N} \alpha_j p_k(\mathbf{x}^j) = 0 \quad \text{for } k = 1, \ldots, M, \tag{V.7}$$

which has the following matrix form:

$$\mathbf{P} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{pmatrix} = 0,$$

where

$$\mathbf{P} = \begin{pmatrix} p_1(\mathbf{x}^1) & \ldots & p_1(\mathbf{x}^N) \\ \vdots & \ddots & \vdots \\ p_M(\mathbf{x}^1) & \ldots & p_M(\mathbf{x}^N) \end{pmatrix}.$$

The interpolation equations using $g(\mathbf{x})$ in Eq. (V.6) become

$$\sum_{j=1}^{N} \alpha_j \varphi(||\mathbf{x}^k - \mathbf{x}^j||) + \sum_{j=1}^{M} \beta_j p_j(\mathbf{x}^k) = f_k \quad \text{for } k = 1, \ldots, N. \tag{V.8}$$

Combining (V.7) and (V.8), we obtain the following matrix equation for the constrained RBF interpolation:

$$\begin{pmatrix} \Phi & \mathbf{P}^T \\ \mathbf{P} & 0 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_N \\ \beta_1 \\ \vdots \\ \beta_M \end{pmatrix} = \begin{pmatrix} f_1 \\ \vdots \\ f_N \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{V.9}$$

V.3 SOLVABILITY OF RBF INTERPOLATION PROBLEMS

In this section, we study the solvability of the linear systems (V.4) and (V.9). Proofs of some known results are included for a better understanding of why these linear systems are solvable. The key results on solvability of RBF interpolations related to the four RBFs shown in Fig. 9 are the following:

- if $\varphi(t) = \sqrt{1+t^2}$ or $\varphi(t) = \exp(-t^2)$, then Eq. (V.4) is always solvable;
- if $\varphi(t) = t^3$ or $\varphi(t) = t^2 \ln t$, then Eq. (V.9) for $m = 2$ is solvable provided that the input vectors $\mathbf{x}^1, \ldots, \mathbf{x}^N$ do not fall into the zero set of a nonconstant quadratic polynomial.

The proofs of the above statements are based on mathematical concepts called (conditionally) positive definiteness and their Schoenberg-Micchelli characterizations as described by Schaback and Wendland [14].

First we give the definitions of (conditionally) positive definiteness of functions defined on $\mathbb{R}^n$.

Definition V.1 *Suppose that $H(\mathbf{x})$ is a real-valued function on $\mathbb{R}^n$ and $H(-\mathbf{x}) = H(\mathbf{x})$. Then*

- *$H(\mathbf{x})$ is said to be positive definite on $\mathbb{R}^n$, if*

$$\sum_{j,k=1}^{N} \alpha_j \alpha_k H(\mathbf{x}^k - \mathbf{x}^j) > 0 \text{ whenever } \sum_{i=1}^{N} |\alpha_i| > 0 \text{ and } \mathbf{x}^j \neq \mathbf{x}^k \text{ for } j \neq k; \tag{V.10}$$

- *$H(\mathbf{x})$ is said to be conditionally positive definite of order m on $\mathbb{R}^n$, if Eq. (V.10) holds for any vector $(\alpha_1, \ldots, \alpha_N)^T \in \mathbb{R}^N$ satisfying (V.7).*

For a RBF $\varphi(t)$, there is a corresponding $H(\mathbf{x}) = \varphi(\|\mathbf{x}\|)$. Note that if $\varphi(\|\mathbf{x}\|)$ is positive definite, then the RBF interpolation matrix Φ defined in Eq. (V.5) is positive definite, which implies the nonsingularity of Φ and the solvability of the RBF interpolation problem (V.4). In the case of $\varphi(\|\mathbf{x}\|)$ being conditionally positive definite of order m, the constrained RBF interpolation problem (V.9) is a linear system with a nonsingular coefficient matrix and has a unique solution too.

To prove the positive definiteness of Φ for $\varphi(t) = \exp(-t^2)$, we need the following characterization of positive definiteness of $H(\mathbf{x}) = \varphi(\|\mathbf{x}\|)$ by Schoenberg [15].

Theorem V.1 *(Schoenberg) Suppose that $\varphi(t) \geq 0$ is a nonconstant continuous function for $t \geq 0$. Then $H(\mathbf{x}) = \varphi(\|\mathbf{x}\|)$ is positive definite on $\mathbb{R}^n$ for every positive integer n if and only if*

$$(-1)^k \frac{d^k}{dt^k}\left[\varphi\left(\sqrt{t}\right)\right] \geq 0 \quad \text{for } t > 0 \text{ and } k = 1, 2, \ldots, \tag{V.11}$$

where $\frac{d^k}{dt^k}\left[\varphi\left(\sqrt{t}\right)\right]$ denotes the kth derivative of $\varphi\left(\sqrt{t}\right)$ with respect to t.

The nonnegative and nonconstant continuous function $\varphi(\sqrt{t})$ satisfying Eq. (V.11) is also called a completely monotone function by Schoenberg [15]. The solvability of RBF interpolation problem (V.4) for Gaussian RBF follows immediately from Theorem V.1.

Theorem V.2 *Suppose that $\mathbf{x}^1, \ldots, \mathbf{x}^N$ are distinct points in $\mathbb{R}^n$ and $\varphi(t) = \exp(-t^2)$ (Gaussian RBF). Then the RBF interpolation matrix Φ defined in Eq. (V.5) is positive definite and the RBF interpolation problem (V.4) is always solvable.*

Proof: Obviously, $\varphi(t)$ is a nonnegative and nonconstant continuous function for $t \geq 0$. Moreover,

$$(-1)^k \frac{d^k}{dt^k}\left[\varphi\left(\sqrt{t}\right)\right] = (-1)^k \frac{d^k}{dt^k}\Big[\exp(-t)\Big] = \exp(-t) > 0.$$

Therefore, by Theorem V.1, Φ is positive definite; hence, the interpolation problem (V.4) always has a unique solution. □

However, for $\varphi(t) = \sqrt{1+t^2}$ (multiquadric RBF), $\varphi(\|\mathbf{x}\|)$ is not a positive definite function. In fact,

$$-\frac{d}{dt}\left[\varphi\left(\sqrt{t}\right)\right] = -\frac{d}{dt}\left[\sqrt{1+t}\right] = -\frac{1}{2\sqrt{1+t}} < 0 \quad \text{for } t > 0,$$

which, along with Theorem V.1, implies that Φ is not always positive definite. However, Φ is always nonsingular if $\mathbf{x}^1, \ldots, \mathbf{x}^N$ are distinct. To understand why Φ is nonsingular for $\varphi(t) = \sqrt{1+t^2}$ (which was first introduced by Hardy [16] in topography applications), we reproduce the proof by Powell [5] here.

Theorem V.3 *Suppose that $\mathbf{x}^1, \ldots, \mathbf{x}^N$ are distinct points in $\mathbb{R}^n$ and $\varphi(t) = \sqrt{1+t^2}$ (multiquadric RBF). Then the RBF interpolation matrix Φ defined in Eq. (V.5) is nonsingular (with $(N-1)$ negative eigenvalues and one positive eigenvalue) and the RBF interpolation problem (V.4) is always solvable.*

Proof: The change of variables $s = z\tau$ establishes the identity

$$\sqrt{z} = \kappa \int_0^\infty [1 - \exp(-z\tau)]\, \tau^{-\frac{3}{2}} d\tau, \tag{V.12}$$

where

$$\kappa = \left(\int_0^\infty [1 - \exp(-s)]\, s^{-\frac{3}{2}} ds \right)^{-1} > 0.$$

Let $\alpha_1, \ldots, \alpha_n$ be such that $\sum_{j=1}^N \alpha_j = 0$ and $\sum_{j=1}^N |\alpha_j| > 0$. Then, for $H(\mathbf{x}) = \varphi(\|\mathbf{x}\|) = \sqrt{1+\|\mathbf{x}\|^2}$, we have

$$\begin{aligned}
\sum_{j,k=1}^N \alpha_j \alpha_k H(\mathbf{x}^k - \mathbf{x}^j) &= \sum_{j,k=1}^N \alpha_j \alpha_k \sqrt{1 + \|\mathbf{x}^k - \mathbf{x}^j\|^2} \\
&= \kappa \int_0^\infty \sum_{j,k=1}^N \alpha_j \alpha_k \left(1 - \exp\Big(- \big[1 + \|\mathbf{x}^k - \mathbf{x}^j\|^2\big]\tau \Big) \right) \tau^{-\frac{3}{2}} d\tau \\
&= -\kappa \int_0^\infty \left[\sum_{j,k=1}^N \alpha_j \alpha_k \exp\Big(- \tau \|\mathbf{x}^k - \mathbf{x}^j\|^2 \Big) \right] \exp(-\tau)\, \tau^{-\frac{3}{2}} d\tau < 0,
\end{aligned} \tag{V.13}$$

where the second equality follows from Eq. (V.12), the third equality depends on $\sum_{j=1}^N \alpha_j = 0$, and the last inequality is based on the positiveness of $\sum_{j,k=1}^N \alpha_j \alpha_k \exp\big(- \tau \|\mathbf{x}^k - \mathbf{x}^j\|^2 \big)$ for $\tau > 0$ (which is a consequence of Theorem V.2 about the positive definiteness of the Gaussian RBF interpolation matrix for the scaled input vectors $\sqrt{\tau}\mathbf{x}^1, \ldots, \sqrt{\tau}\mathbf{x}^N$).

If the multiquadric RBF interpolation matrix Φ, whose (k,j) entry is $H(\mathbf{x}^k - \mathbf{x}^j)$, has two nonnegative eigenvalues with eigenvectors $\mathbf{v}^1$ and $\mathbf{v}^2$, then there is a nonzero vector of the form $\mathbf{v} = a_1\mathbf{v}^1 + a_2\mathbf{v}^2$ whose components sum to zero. Because Φ is a symmetric matrix, we have

$$\mathbf{v}^T \Phi \mathbf{v} = a_1^2 (\mathbf{v}^1)^T \Phi \mathbf{v}^1 + a_2^2 (\mathbf{v}^2)^T \Phi \mathbf{v}^2 \geq 0. \tag{V.14}$$

However, if we use $\alpha_1, \ldots, \alpha_N$ to denote the components of $\mathbf{v}$, then $\sum_{j=1}^{N} \alpha_j = 0$ and $\sum_{j=1}^{N} |\alpha_j| > 0$. By Eq. (V.13), $\mathbf{v}^T \Phi \mathbf{v} = \sum_{j,k=1}^{N} \alpha_j \alpha_k H(\mathbf{x}^k - \mathbf{x}^j) < 0$, which is a contradiction to Eq. (V.14). This contradiction proves that Φ has at least $(N-1)$ negative eigenvalues. Moreover, because the trace of Φ (the sum of the diagonal elements) is $N > 0$, which is the same as the sum of all eigenvalues, Φ also has a positive eigenvalue. Hence, Φ has exactly $(N-1)$ negative eigenvalues and one positive eigenvalue. As a consequence, all the eigenvalues of Φ are nonzero and Φ is nonsingular; so the multiquadric RBF interpolation problem (V.4) always has a unique solution. □

For cubic spline and thin plate spline RBFs, it is not easy to determine the solvability of the RBF interpolation problem (V.4) based on distributions of the input vectors $\mathbf{x}^1, \ldots, \mathbf{x}^N$. However, using the concept of conditionally positive definiteness, one can solve the constrained RBF interpolation problem (V.9) for almost all non-trivial distributions of $\mathbf{x}^1, \ldots, \mathbf{x}^N$. The theory is based on following generalization of the sufficient part of Theorem V.1 by Micchelli [6].

Theorem V.4 *(Micchelli) Suppose that $\varphi(t)$ is a continuous function for $t \geq 0$ and*

$$(-1)^k \frac{d^k}{dt^k}\left[\varphi\left(\sqrt{t}\right)\right] \geq 0 \quad \text{for } t > 0 \text{ and } k = m, m+1, \ldots, \tag{V.15}$$

then $\varphi(\|\mathbf{x}\|)$ is conditionally positive definite of order m on $\mathbb{R}^n$ for every positive integer n.

The utility of the conditional positive definiteness concept is shown in the next theorem [14].

Theorem V.5 *Suppose that $\varphi(t)$ is a continuous function of $t \geq 0$ and $\varphi(\|\mathbf{x}\|)$ is conditionally positive definite of order m on $\mathbb{R}^n$. Let $\mathbf{x}^1, \ldots, \mathbf{x}^N$ be distinct points in $\mathbb{R}^n$ that do not fall into the zero set of a nonconstant polynomial $p(\mathbf{x})$ of degree at most m (i.e., $p(\mathbf{x})$ is identical to 0 if $p(\mathbf{x})$ is a polynomial of degree at most m and $p(\mathbf{x}^j) = 0$ for $j = 1, \ldots, N$). Then the constrained RBF interpolation problem (V.9) has a unique solution.*

Proof: Note that Eq. (V.9) has a unique solution if and only if the coefficient matrix in the linear system (V.9) is nonsingular. We prove Theorem V.5 by contradiction.

If the coefficient matrix is singular, then there exist $\alpha_1, \ldots, \alpha_N, \beta_1, \ldots, \beta_M$, not all zeros, such that Eq. (V.7) holds and

$$\sum_{j=1}^{N} \alpha_j \varphi(||\mathbf{x}^k - \mathbf{x}^j||) + \sum_{j=1}^{M} \beta_j p_j(\mathbf{x}^k) = 0 \quad \text{for } k = 1, \ldots, N. \tag{V.16}$$

Multiplying the kth equation in Eq. (V.16) by α_k for $k = 1, \ldots, N$ and adding the scaled equations together, we obtain

$$\sum_{k=1}^{N}\sum_{j=1}^{N} \alpha_j \alpha_k \varphi(||\mathbf{x}^k - \mathbf{x}^j||) + \sum_{k=1}^{N}\sum_{j=1}^{M} \alpha_k \beta_j p_j(\mathbf{x}^k) = 0. \tag{V.17}$$

However, by (V.7), the second term in Eq. (V.17) is zero, so we have

$$\sum_{k=1}^{N}\sum_{j=1}^{N} \alpha_j \alpha_k \varphi(||\mathbf{x}^k - \mathbf{x}^j||) = 0. \tag{V.18}$$

Because $\alpha_1, \ldots, \alpha_N$ satisfy (V.7), by the conditional positive definiteness of $\varphi(\|\mathbf{x}\|)$, Eq. (V.18) forces $\alpha_1 = \ldots = \alpha_N = 0$. Thus, Eq. (V.16) implies that $\mathbf{x}^1, \ldots, \mathbf{x}^N$ are zeros of the polynomial $\sum_{j=1}^{M} \beta_j p_j(\mathbf{x})$. By the assumption about $\mathbf{x}^1, \ldots, \mathbf{x}^N$ given in Theorem V.5, $\sum_{j=1}^{M} \beta_j p_j(\mathbf{x}) = 0$ for all $\mathbf{x}$, which implies $\beta_1 = \ldots = \beta_M = 0$ because $p_1(\mathbf{x}), \ldots, p_M(\mathbf{x})$ form a basis for polynomials of degree at most m. The conclusion that $\alpha_1 = \ldots = \alpha_N = \beta_1 = \ldots = \beta_M = 0$ contradicts the assumption that they are not all zeros. The contradiction proves the nonsingularity of the coefficient matrix of the linear system (V.9), thus, the constrained RBF interpolation problem (V.9) has a unique solution. □

Remark. (i) Note that $\varphi(t) = \exp(-t^2)$ obviously satisfies Eq. (V.15) for any nonnegative integer m, so $\exp(-\|\mathbf{x}\|^2)$ is conditionally positive definite of order m for every $m \geq 0$ (see Theorem V.4). By Theorem V.5, for any nonnegative integer m, the constrained Gaussian RBF interpolation problem (V.9) has a unique solution if $\mathbf{x}^1, \ldots, \mathbf{x}^N$ do not fall into the zero set of a nonconstant polynomial of degree at most m. In particular, if $\mathbf{x}^1, \ldots, \mathbf{x}^N$ are distinct, the constrained Gaussian RBF interpolation problem (V.9) with $m = 0$ has a unique solution.

(ii) For RBF $\varphi(t) = -\sqrt{1+t^2}$, we have

$$(-1)^k \frac{d^k}{dt^k}\left[\varphi\left(\sqrt{t}\right)\right] = (-1)^k \frac{d^k}{dt^k}\left[-\sqrt{1+t}\right] = \frac{(2k-2)!}{2^{2k-1}(k-1)!}(1+t)^{\frac{1-2k}{2}} > 0$$

for $t > 0$ and any positive integer k. Here $(k-1)! = 1 \cdot 2 \cdots (k-1)$ is the factorial notation. By Theorem V.4, $\varphi(\|\mathbf{x}\|)$ is conditionally positive definite of order m for

$m \geq 1$. By Theorem V.5, if $m \geq 1$ and $\mathbf{x}^1, \ldots, \mathbf{x}^N$ do not fall into the zero set of a nonconstant polynomial of degree at most m, then the constrained RBF interpolation problem (V.9) has a unique solution for $\varphi(t) = -\sqrt{1+t^2}$, which is equivalent to say that the constrained multiquadric RBF interpolation problem (V.9) has a unique solution.

Micchelli's theorem on conditionally positive definite functions allows us to prove the conditional positive definiteness of cubic and thin plate RBFs, and solvability of the related constrained RBF interpolation problems.

Theorem V.6 *Let $m \geq 2$. Then, for cubic RBF $\varphi(t) = t^3$ or thin plate RBF $\varphi(t) = t^2 \ln t$, the constrained RBF interpolation problem (V.9) has a unique solution if $\mathbf{x}^1, \ldots, \mathbf{x}^N$ do not fall into the zero set of a nonconstant polynomial of degree at most m.*

Proof: For $t > 0$ and $k = 2, 3, \ldots$, we have either

$$(-1)^k \frac{d^k}{dt^k}\left[\varphi(\sqrt{t})\right] = (-1)^k \frac{d^k}{dt^k}\left[t^{\frac{3}{2}}\right] = \frac{3}{2} \cdot \frac{(2k-4)!}{2^{2k-3}(k-2)!} t^{\frac{3-2k}{2}} > 0,$$

or

$$(-1)^k \frac{d^k}{dt^k}\left[\varphi(\sqrt{t})\right] = (-1)^k \frac{d^k}{dt^k}\left[\frac{1}{2} t \ln t\right] = \frac{(k-2)!}{2}\, t^{1-k} > 0 \quad \text{for } t > 0.$$

By Theorem V.4 and $m \geq 2$, $\varphi(\|\mathbf{x}\|)$ is conditionally positive definite of order m. The solvability of Eq. (V.9) follows from Theorem V.5. □

The above argument can be easily modified to prove theorems on solvability of the constrained RBF interpolation problem (V.9) for $\varphi(t) = t^\kappa$ with any positive κ that is not an even integer and $\varphi(t) = t^{2l} \ln t$ with positive integer l.

V.4 KRIGING VERSUS GAUSSIAN RBF INTERPOLATION

Kriging is an interpolation method named after a South African mining engineer D. G. Krige who developed the technique in an attempt to more accurately predict ore reserves. Over the past several decades Kriging has become a fundamental tool in many fields [17, 18].

In ordinary Kriging, the estimation $g(\mathbf{x})$ of an unknown function value $f(\mathbf{x})$ is done by using a weighted average of the known function values $f(\mathbf{x}^1), \ldots, f(\mathbf{x}^N)$:

$$g(\mathbf{x}) = \sum_{j=1}^{N} \tau_j f(\mathbf{x}^j) \quad \text{with} \quad \sum_{j=1}^{N} \tau_j = 1. \tag{V.19}$$

The function value $f(\mathbf{x})$ is assumed to be a realization of an intrinsic random function with the semivariance

$$\varphi(\|\Delta\mathbf{x}\|) = E\Big(|f(\mathbf{x}+\Delta\mathbf{x}) - f(\mathbf{x})|^2\Big), \tag{V.20}$$

where $E(\cdot)$ denotes the expected value of a random function and $\varphi(t)$ is a decreasing function of $t \geq 0$. Semivariance is a measure of the degree of spatial dependence between two function values and depends only on the distance $\|\Delta\mathbf{x}\|$ between the two input locations. A smaller distance yields a smaller semivariance and a larger distance results in a larger semivariance. The plot of the semivariances as a function of distance from a point is referred to as a semivariogram. The semivariance increases as the distance increases until at a certain distance away from a point the semivariance will equal the variance around the average value, and will therefore no longer increase, causing a flat region (whose height is called a sill) to occur on the semivariogram. The distance from the point of interest to where the flat region begins is termed the range or span of the regionalized input variable. Within this range, denoted by δ, locations are related to each other, and all known samples contained in this region, also referred to as the neighborhood, must be considered when estimating the function value at an unknown point of interest in the region. Two examples of semivariance are the spherical semivariance

$$\varphi(t) = \begin{cases} c_0 + c_1\left(1.5\frac{t}{\delta} - 0.5\left(\frac{t}{\delta}\right)^3\right) & \text{if } |t| \leq \delta \\ c_0 + c_1 & \text{if } |t| > \delta \end{cases}$$

and the exponential semivariance

$$\varphi(t) = \begin{cases} 0 & \text{if } |t| = 0 \\ c_0 + c_1\left(1 - \exp\left(\frac{-3|t|}{\delta}\right)\right) & \text{if } |t| > 0. \end{cases}$$

In both examples the sill is $c_0 + c_1$, where c_0 and c_1 are constants.

Using Eq. (V.19), one can calculate the variance of estimation error as follows:

$$E[(f(\mathbf{x}) - g(\mathbf{x}))^2] = E\left[\left(\sum_{j=1}^{N} \tau_j\Big(f(\mathbf{x}) - f(\mathbf{x}^j)\Big)\right)^2\right]$$

$$= \sum_{j=1}^{N}\sum_{k=1}^{N} \tau_j\tau_k E\left[\Big(f(\mathbf{x}) - f(\mathbf{x}^j)\Big)\Big(f(\mathbf{x}) - f(\mathbf{x}^k)\Big)\right]. \tag{V.21}$$

However,

$$\begin{aligned}
&\varphi(\|\mathbf{x}-\mathbf{x}^j\|)+\varphi(\|\mathbf{x}-\mathbf{x}^k\|)-\varphi(\|\mathbf{x}^k-\mathbf{x}^j\|)\\
&=E\left[\left(f(\mathbf{x})-f(\mathbf{x}^j)\right)^2\right]+E\left[\left(f(\mathbf{x})-f(\mathbf{x}^k)\right)^2\right]-E\left[\left(f(\mathbf{x}^j)-f(\mathbf{x}^k)\right)^2\right]\\
&=2E\left[\left(f(\mathbf{x})-f(\mathbf{x}^j)\right)\left(f(\mathbf{x})-f(\mathbf{x}^k)\right)\right]. \qquad \text{(V.22)}
\end{aligned}$$

Substituting Eq. (V.22) into Eq. (V.21) and simplifying the resulting expression by using $\sum_{j=1}^{N}\tau_j=1$, we obtain the following formula for the variance of estimation error:

$$E[(f(\mathbf{x})-g(\mathbf{x}))^2]=-\frac{1}{2}\sum_{j=1}^{N}\sum_{k=1}^{N}\tau_j\tau_k\varphi(\|\mathbf{x}^k-\mathbf{x}^j\|)+\sum_{j=1}^{N}\tau_j\varphi(\|\mathbf{x}-\mathbf{x}^j\|).$$

The ordinary Kriging estimate $g(\mathbf{x})$ of $f(\mathbf{x})$ is obtained by minimizing the variance of estimation error with the constraint on weights τ_j given in Eq. (V.19). The optimal solution of $\tau_1,\ldots,\tau_N$ is the solution to the following system of linear equations (i.e., the optimality conditions):

$$\begin{cases}
\displaystyle\sum_{j=1}^{N}\tau_j=1\\
\displaystyle\tau_0+\sum_{j=1}^{N}\tau_j\varphi(\|\mathbf{x}^k-\mathbf{x}^j\|)=\varphi(\|\mathbf{x}-\mathbf{x}^k\|) \quad \text{for } k=1,\ldots,N
\end{cases}$$

or

$$\begin{pmatrix}
0 & 1 & \cdots & 1\\
1 & \varphi(\|\mathbf{x}^1-\mathbf{x}^1\|) & \cdots & \varphi(\|\mathbf{x}^1-\mathbf{x}^N\|)\\
\vdots & \ddots & \vdots & \vdots\\
1 & \varphi(\|\mathbf{x}^N-\mathbf{x}^1\|) & \cdots & \varphi(\|\mathbf{x}^N-\mathbf{x}^N\|)
\end{pmatrix}\cdot\begin{pmatrix}\tau_0\\ \tau_1\\ \vdots\\ \tau_N\end{pmatrix}=\begin{pmatrix}1\\ \varphi(\|\mathbf{x}-\mathbf{x}^1\|)\\ \vdots\\ \varphi(\|\mathbf{x}-\mathbf{x}^N\|)\end{pmatrix}, \qquad \text{(V.23)}$$

where the parameter τ_0 is a Lagrange multiplier for the equality constraint of $\sum_{j=1}^{N}\tau_j=1$. The left-hand side of Eq. (V.23) describes the dissimilarities among data points while the right-hand side describes the dissimilarities between the data points and the estimation point.

It is well-known that ordinary Kriging estimation $g(\mathbf{x})$ is an interpolation of the data points $(\mathbf{x}^1,f(\mathbf{x}^1)),\ldots,(\mathbf{x}^N,f(\mathbf{x}^N))$, i.e., $g(\mathbf{x}^k)=f(\mathbf{x}^k)$ for $k=1,\ldots,N$. In fact, if $\mathbf{x}=\mathbf{x}^k$, then it is easy to verify that $\tau_j=0$ for $j\neq k$, $\tau_k=1$, and $\tau_0=0$ solve

Eq. (V.23). Once the inverse of the coefficient matrix of Eq. (V.23) is available, we can compute the weights $\tau_1, \ldots, \tau_N$ for Kriging estimation $g(\mathbf{x})$ by a matrix-vector multiplication.

Kriging interpolation is very similar to Gaussian RBF interpolation but the two interpolation methods are not the same. Li and Padula [2] proved that Kriging interpolant is the solution of the constrained Gaussian RBF interpolation problem (V.9) with $m = 0$. By Remark (i) after Theorem V.5, Kriging interpolation has a unique solution.

Li and Padula [2] also gave the following classical interpolation formulation of Kriging interpolation that shows the relationship between Gaussian RBF interpolation and Kriging interpolation. Let $\varphi(t) = \exp(-t^2)$ be the Gaussian RBF, Φ the interpolation matrix defined by Eq. (V.5), and $\vec{f}$ the column vector in the right-hand side of Eq. (V.4) (i.e., the jth component of $\vec{f}$ is f_j). Use $\vec{\psi}(\mathbf{x})$ to denote the column vector whose jth component is $\varphi(\|\mathbf{x} - \mathbf{x}^j\|)$ for $j = 1, \ldots, N$, and define

$$\vec{\zeta}(\mathbf{x}) = \begin{pmatrix} \zeta_1(\mathbf{x}) \\ \vdots \\ \zeta_N(\mathbf{x}) \end{pmatrix} = \vec{\psi}(\mathbf{x}) + \frac{1 - \mathbf{1}^T \Phi^{-1} \vec{\psi}(\mathbf{x})}{\mathbf{1}^T \Phi^{-1} \mathbf{1}} \mathbf{1},$$

where $\mathbf{1}$ denotes the column vector of ones. Then the Kriging interpolant can be written as a linear combination of $\zeta_j(\mathbf{x})$:

$$g(\mathbf{x}) = \sum_{j=1}^{N} \bar{\alpha}_j \zeta_j(\mathbf{x}),$$

where $\bar{\alpha}_1, \ldots, \bar{\alpha}_N$ satisfy the following interpolation conditions:

$$\sum_{j=1}^{N} \bar{\alpha}_j \zeta_j(\mathbf{x}^k) = f_k \quad \text{for } k = 1, \ldots, N. \tag{V.24}$$

In other words, the Kriging interpolation problem can be considered as a classical interpolation problem with the basis functions $\zeta_1(\mathbf{x}), \ldots, \zeta_N(\mathbf{x})$.

The system of linear equations (V.3) for Gaussian RBF interpolation has the same solution as Eq. (V.24) for Kriging interpolation because $\zeta_j(\mathbf{x}^k) = \varphi(\|\mathbf{x}^k - \mathbf{x}^j\|)$ for $j, k = 1, \ldots, N$. Thus, the Kriging interpolant is

$$(\Phi^{-1}\vec{f})^T \vec{\zeta}(\mathbf{x}) = (\Phi^{-1}\vec{f})^T \vec{\psi}(\mathbf{x}) + \left(\frac{1 - \mathbf{1}^T \Phi^{-1} \vec{\psi}(\mathbf{x})}{\mathbf{1}^T \Phi^{-1} \mathbf{1}} \right) (\Phi^{-1}\vec{f})^T \mathbf{1},$$

where the first term on the right-hand side is the Gaussian RBF interpolant.

Chapter VI

MODEL PARAMETER TUNING

VI.1 INTRODUCTION

Both Kriging and RBF interpolation models use the parameterized distance:

$$\|\mathbf{x}-\mathbf{x}^j\| = \sqrt{\sum_{i=1}^{n} |\theta_i| \left(\frac{x_i - x_i^j}{\sigma_i}\right)^2}, \tag{VI.1}$$

where σ_i is the estimated standard deviation of the ith component of $\mathbf{x}$ and $\theta_1, \ldots, \theta_n$ are scalars. Mathematically, one could pick any fixed set of $\theta_1, \ldots, \theta_n$ and construct the interpolation function for the given data. However, two different sets of $\theta_1, \ldots, \theta_n$ will lead to two interpolation models that behave very differently between the input vectors $\mathbf{x}^1, \ldots, \mathbf{x}^N$. Model parameter tuning for Kriging or RBF interpolation aims at finding a set of parameters $\theta_1, \ldots, \theta_n$ that results in the best prediction model of the unknown response based on the available data. Cross validation (CV) [8, 19, 20, 21] and maximum likelihood estimation [22, 23, 24, 25, 26] are two statistical methods for tuning the model parameters $\theta_1, \ldots, \theta_n$ for best prediction models.

CV can be used for general model parameter tuning, while maximum likelihood estimation can only be applied for density function parameter estimation. Both statistical methods are introduced in this chapter. RBF interpolation for wing weight estimation is used as an application of CV for model parameter tuning and Gaussian process [27, 28] for data fitting is used as an application of maximum likelihood estimation. In the last section, we introduce an automatic PCR procedure based on CV errors.

VI.2 MODEL PARAMETER TUNING BY CV

Because either Kriging or RBF interpolation method yields a fitting function $g(\mathbf{x})$ whose value at $\mathbf{x}^k$ is exactly f_k for $k = 1, \ldots, N$, other metrics instead of fitting errors must be used to determine which basis function $\varphi(t)$ and what scaling parameters θ_i are most appropriate to model the response function $f(\mathbf{x})$. A technique that can be used, but often impractical (and always expensive) is to obtain values of $f(\mathbf{x})$ at some additional data points $\mathbf{x}^{N+1}, \ldots, \mathbf{x}^{\bar{N}}$ and use the prediction errors $|g(\mathbf{x}^k) - f(\mathbf{x}^k)|$ for $k = N+1, \ldots, \bar{N}$ to assess the prediction accuracy of $g(\mathbf{x})$. The prediction accuracy

can be used as a criterion for choosing the best basis function $\varphi(t)$ and parameters θ_i.

Without additional sample points, CV [20, 8] was proposed to find $\varphi(t)$ and θ_i that lead to an approximate response model $g(\mathbf{x})$ with optimal prediction capability and proved to be effective [19, 21]. For wing weight data of 41 subsonic transports, only the leave-one-out CV procedure is applicable for RBF interpolation.

Leave-one-out Cross Validation for Kriging or RBF Interpolation:

- Fix a set of parameters $\theta_1, \ldots, \theta_n$.
- For $j = 1, \ldots, N$, construct the Kriging or RBF interpolant $g_{-j}(\mathbf{x})$ of the data points $(\mathbf{x}^k, f_k)$ for $1 \leq k \leq N, k \neq j$.
- Use the following CV root mean square error as the prediction error:

$$E^{CV}(\theta_1, \ldots, \theta_n) = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(g_{-j}(\mathbf{x}^j) - f_j)^2}. \tag{VI.2}$$

Remark. One could also use other forms of CV errors such as the CV average absolute error: $\frac{1}{N}\sum_{j=1}^{N}|g_{-j}(\mathbf{x}^j) - f(\mathbf{x}^j)|$.

In the case that each $\mathbf{x}^j$ has a close neighbor $\mathbf{x}^k$ $(k \neq j)$ in the space of input variables and $f(\mathbf{x})$ is a smooth function, E^{CV} is not a meaningful measure of the prediction accuracy of the fitting model because $g_{-j}(\mathbf{x}^k) = f_k$ implies $g_{-j}(\mathbf{x}^j) \approx f_j$ due to small values of $\|\mathbf{x}^k - \mathbf{x}^j\|$ and $|f_k - f_j|$. Therefore, if the leave-one-out CV error is used as a criterion for model parameter tuning, then analysts must be warned when undesirable clustering of $\mathbf{x}^j$ occurs.

Model parameter tuning by CV is to find $\theta_1, \ldots, \theta_n$ that minimize the CV error $E^{CV}(\theta_1, \ldots, \theta_n)$ so that the interpolation model has the highest prediction accuracy when measured by the CV error.

It is worth pointing out that it is difficult to minimize $E^{CV}(\theta_1, \ldots, \theta_n)$ numerically because $E^{CV}(\theta_1, \ldots, \theta_n)$ is a highly nonlinear and nonconvex function. One could make the model parameter tuning much easier by assuming $\theta_1 = \cdots = \theta_n$, which reduces the problem to unconstrained minimization of a univariate function (see [19]). This approach has the obvious benefit of dealing with a much easier optimization problem but the disadvantage of not using all different θ_i. Different θ_i allow the model parameter tuning to scale each variable x_i based on its significance in modeling the

variance in the response, thus, have the benefit of implicit variable screening built in the model parameter tuning.

VI.3 CV FOR PRINCIPAL COMPONENT REGRESSION

Following the notations for PCR in Chapter III, we can project the data to a reduced feature space by using the following formula:

$$v_i^j = (\hat{\mathbf{x}}^j - \text{ave}(\hat{\mathbf{x}}))^T \mathbf{u}^i \quad \text{for } 1 \le i \le r, 1 \le j \le N, \tag{VI.3}$$

where $\hat{x}_i = x_i/\sigma_i$, i.e., $\hat{\mathbf{x}}$ is a scaled version of $\mathbf{x}$. Let $\mathbf{v}^j$ be the reduced feature vector with r components v_i^j for $i = 1, \ldots, r$. Then the RBF interpolation problem in the reduced feature space can be formulated as follows:

$$\sum_{j=1}^{N} \alpha_j \varphi(\|\mathbf{v}^k - \mathbf{v}^j\|) = f_k \quad \text{for } k = 1, \ldots, N. \tag{VI.4}$$

The interpolation function $\hat{g}(\mathbf{v}) = \sum_{j=1}^{N} \alpha_j \varphi(\|\mathbf{v} - \mathbf{v}^j\|)$ can be used to construct a wing weight prediction model in the $\mathbf{x}$-space:

$$\bar{w} = \hat{g}\Big((\hat{\mathbf{x}} - \text{ave}(\hat{\mathbf{x}}))^T \mathbf{u}^1, \ldots, (\hat{\mathbf{x}} - \text{ave}(\hat{\mathbf{x}}))^T \mathbf{u}^r\Big). \tag{VI.5}$$

Because we use the components of the reduced feature vector as the input variables, Eq. (VI.1) has the following form:

$$\|\mathbf{v} - \mathbf{v}^j\| = \sqrt{\sum_{i=1}^{r} |\hat{\theta}_i| \left(\frac{v_i - v_i^j}{\hat{\sigma}_i}\right)^2}, \tag{VI.6}$$

where $\hat{\sigma}_i$ is the standard deviation of the ith components of $\mathbf{v}^1, \ldots, \mathbf{v}^N$.

The interpolation function $\hat{g}_{-j}(\mathbf{v}) = \sum_{i=1, i\neq j}^{N} \alpha_i \varphi(\|\mathbf{v} - \mathbf{v}^i\|)$ in the CV procedure can be obtained by solving the following interpolation equations:

$$\sum_{i=1, i\neq j}^{N} \alpha_i \varphi(\|\mathbf{v}^k - \mathbf{v}^i\|) = f_k \quad \text{for } 1 \le k \le N, k \neq j. \tag{VI.7}$$

Thus, the CV error is

$$E^{CV}(\hat{\theta}_1, \ldots, \hat{\theta}_r) = \sqrt{\sum_{j=1}^{N} \Big(g_{-j}(\mathbf{v}^j) - f_j\Big)^2}. \tag{VI.8}$$

Similarly, the CV errors for constrained RBF interpolation and Kriging interpolation can be computed in the reduced feature space or in the original $\mathbf{x}$-space.

VI.4 NUMERICAL RESULTS FOR MINIMIZATION OF CV ERROR

The multidimensional unconstrained nonlinear minimization method by Nelder and Mead [29] is a direct search method and one of the widely used methods to find a minimizer of a multivariate function. We use the MATLAB program **fminsearch**, an implementation of the Nelder-Mead multidimensional search algorithm, to minimize E^{CV}.

For the wing weight data fitting problem, we use both $n = 14$ (all configuration parameters listed in Table 1 except $[t/c]_m$ that is redundant) and $n = 8$ (for the input variables in Eq. (II.1)) as the dimension of the original $\mathbf{x}$ vector. If we do not use reduced feature spaces, then the CV error $E^{CV}(\theta_1, \ldots, \theta_n)$ in Eq. (VI.2) is minimized to find the best model parameters $\theta_1, \ldots, \theta_n$. If $\mathbf{x}^1, \ldots, \mathbf{x}^N$ are projected to the reduced feature space, then $E^{CV}(\hat{\theta}_1, \ldots, \hat{\theta}_r)$ in Eq. (VI.8) is minimized to find the best model parameters $\hat{\theta}_1, \ldots, \hat{\theta}_r$.

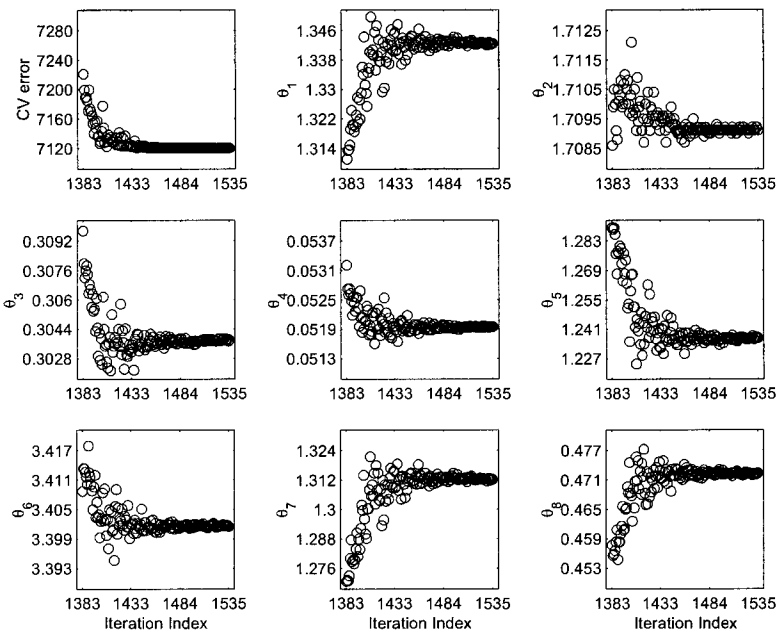

Figure 10: Typical convergence history for cross-validation error minimization.

MATLAB code **fminsearch** is very reliable for finding local optimal solutions. For $n = 8$ and $\theta_1 = \ldots = \theta_n = 1$ as the initial guess, a typical convergence history for the objective function and tuning parameters θ_i (when $n = 8$) is given in Fig. 10. The

local optimal solution generated by MATLAB code **fminsearch** for minimization of the CV error is very sensitive to the initial guess. Multiple initial guesses were used for searching a global minimizer of the CV error by **fminsearch**. However, there is no guarantee that the best solution among the calculated local optimal solutions is a global minimizer of the CV error. Tables 2 and 3 show the minimized CV errors for various interpolation models by using **fminsearch** with multiple initial guesses.

	Multiquadric CV Error	Thin Plate CV Error	Cubic CV Error	Gaussian CV Error	Kriging CV Error
$n = 14$	9162	88352	9180	20276	20151
$r = 14$	22694	12443	17605	56377	37187
$r = 13$	4724	16057	19703	49478	27091
$r = 12$	11858	15808	14749	28883	43594
$r = 11$	7384	44196	45941	34782	49113
$r = 10$	8074	50617	43110	70896	53259
$r = 9$	5731	39986	10711	33132	57457
$r = 8$	22266	37790	150800	50059	52972
$r = 7$	79859	8782	91118	33690	110560
$r = 6$	4888	43764	39554	60052	81911
$r = 5$	10462	28463	26460	57089	94579
$r = 4$	67649	118310	63386	231260	206000

Table 2: Minimized CV errors for the data set with fourteen input variables.

	Multiquadric CV Error	Thin Plate CV Error	Cubic CV Error	Gaussian CV Error	Kriging CV Error
$n = 8$	2697	18814	3396	21229	20579
$r = 8$	4321	7113	3930	34037	79362
$r = 7$	3065	16646	5505	7175000	543410
$r = 6$	3724	4372	4157	615560	445280
$r = 5$	5837	4583	5745	29257	16308
$r = 4$	4163	5589	9100	7757400	19034000

Table 3: Minimized CV errors for the data set with eight input variables.

In general, it is difficult to find global minimizers of nonconvex objective functions, which is not the subject of this thesis. Heuristic search methods like simulated annealing, tabu search, and genetic algorithms can be applied to find approximate solutions of global minimizers of the CV error.

Powell's **UOBYQA** algorithm ([30] or [31]) is a direct search algorithm for unconstrained minimization of a function of several variables. **UOBYQA** uses a smooth quadratic model of the objective function that is constructed by using computed objective function values to accelerate the convergence of iterates. The CV error was also minimized by using **UOBYQA**, but the values of the minimized CV error obtained by **fminsearch** are consistently lower than the ones obtained by **UOBYQA**.

VI.5 AUTOMATIC PCR PROCEDURE

The CV error of an interpolation model can be a useful and objective tool to help analysts decide which model is better. In Tables 2 and 3, we have the CV errors of various interpolation models. In particular, the first row of Table 2 or 3 has the CV errors for interpolation models in the **x**-space, while the other rows show the CV errors in the r-dimensional feature spaces. Instead of letting analysts make a few plausible choices of r, one could also use cross-validation errors to choose the best value for r for each given interpolation model.

Automatic PCR Procedure:

- Compute the eigenvalues of the covariance matrix $\mathbf{C}$ and let $\bar{r}$ be the number of positive eigenvalues.
- Choose an integer $r_{\min} < \bar{r}$ such that there is still a function relationship between the transformed input vectors in the $r_{\min}$-dimensional feature space and the response. For each integer r from $r_{\min}$ to $\bar{r}$, minimize the CV error for the given interpolation model in the r-dimensional feature space (see section VI.3).
- Choose r corresponding to the smallest value of the minimized CV errors and select the corresponding interpolant as $\hat{g}(\mathbf{v})$.
- Reconstruct the wing weight approximation in the **x**-space by using Eq. (VI.5).

For the wing weight data fitting problem, we have $\bar{r} = 14$ for $n = 14$ input variables and $\bar{r} = 8$ for $n = 8$ variables. Tables 2 and 3 have all the minimized CV errors for r from $r_{\min} = 4$ to $\bar{r} = n$. For each interpolation model, PRC generates 11 different interpolants in various feature spaces with r ranging from $r_{\min} = 4$ to

$n = 14$, which yield 11 different wing weight estimation models in the original **x**-space. The above automatic PCR procedure can help to reduce the PCR interpolants to one for each interpolation model. In the case of multiquadric RBF interpolation with $n = 8$ and $r_{\min} = 4$, the best PCR interpolant is generated by using $r = 7$. Later on, we shall see that this PCR interpolant is indeed the most desirable wing weight prediction model for subsonic transports.

VI.6 MAXIMUM LIKELIHOOD ESTIMATION

The maximum likelihood estimation is another widely used statistical method for model parameter tuning [25, 26]. The idea behind maximum likelihood estimation is to determine the parameters that maximize the probability (likelihood) of the sample data. From a statistical point of view, the method of maximum likelihood is considered to be more robust (with some exceptions) and yields estimators with good statistical properties. In other words, maximum likelihood estimation methods are versatile and applicable to many probability density models and to different types of data [22, 23, 24].

Let $\mathbf{y}$ be a random vector with probability density function

$$g(\mathbf{y}; \theta_1, \ldots, \theta_n),$$

where $\theta_1, \ldots, \theta_n$ are n unknown parameters that we want to estimate. Let $\mathbf{y}^1, \ldots, \mathbf{y}^N$ be the set of sample data points. Then the maximum likelihood estimator maximizes the likelihood function

$$L = L(\mathbf{y}^1, \ldots, \mathbf{y}^N; \theta_1, \ldots, \theta_n) = \prod_{j=1}^{N} g(\mathbf{y}^j; \theta_1, \ldots, \theta_n).$$

Equivalently, one can maximize the following log-likelihood function to obtain the maximum likelihood estimator:

$$\ln L = \sum_{j=1}^{N} \ln \Big(g(\mathbf{y}^j; \theta_1, \ldots, \theta_n) \Big).$$

The first-order optimality conditions for the maximum likelihood estimator are

$$\frac{\partial(\ln L)}{\partial \theta_i} = 0 \quad \text{for } i = 1, \ldots, n.$$

Except for a few cases where the maximum likelihood functions are simple, it is generally best to rely on high quality statistical software to obtain maximum likelihood

estimators. Maximum likelihood estimators might have some disadvantages such as being sensitive to the choice of starting values or heavily biased when the number of samples is small.

Gibbs and MacKay [27] used the maximum likelihood estimation method to construct a Gaussian process for fitting a set of sampled density function values $f_1 = f(\mathbf{x}^1), \ldots, f_N = f(\mathbf{x}^N)$. Without the noise model, the Gaussian covariance function used by Gibbs and MacKay is of the following form:

$$\hat{\varphi}(\|\mathbf{x} - \mathbf{x}^j\|) = \kappa_1 \exp\left(-\sum_{i=1}^{n} |\theta_i| \left(\frac{x_i - x_i^j}{\sigma_i}\right)^2\right) + \kappa_2, \tag{VI.9}$$

where κ_1 gives the overall vertical scale relative to the mean of the Gaussian process and κ_2 gives the vertical uncertainty. The corresponding covariance matrix for the sample data is

$$\hat{\Phi} = \begin{pmatrix} \hat{\varphi}(||\mathbf{x}^1 - \mathbf{x}^1||) & \hat{\varphi}(||\mathbf{x}^1 - \mathbf{x}^2||) & \ldots & \hat{\varphi}(||\mathbf{x}^1 - \mathbf{x}^N||) \\ \hat{\varphi}(||\mathbf{x}^2 - \mathbf{x}^1||) & \hat{\varphi}(||\mathbf{x}^2 - \mathbf{x}^2||) & \ldots & \hat{\varphi}(||\mathbf{x}^2 - \mathbf{x}^N||) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\varphi}(||\mathbf{x}^N - \mathbf{x}^1||) & \hat{\varphi}(||\mathbf{x}^N - \mathbf{x}^2||) & \ldots & \hat{\varphi}(||\mathbf{x}^N - \mathbf{x}^N||) \end{pmatrix}. \tag{VI.10}$$

The main result is that an unknown response at $\mathbf{x}$ is a Gaussian (or normal) distribution of $f(\mathbf{x})$ with the mean $g(\mathbf{x})$ and variance σ^2, where

$$g(\mathbf{x}) = \begin{pmatrix} f_1 \\ \vdots \\ f_N \end{pmatrix}^T \hat{\Phi}^{-1} \begin{pmatrix} \hat{\varphi}(\|\mathbf{x} - \mathbf{x}^1\|) \\ \vdots \\ \hat{\varphi}(\|\mathbf{x} - \mathbf{x}^N\| \end{pmatrix} \tag{VI.11}$$

and

$$\sigma^2 = \kappa_1 - \kappa_2 - \begin{pmatrix} \hat{\varphi}(\|\mathbf{x} - \mathbf{x}^1\|) \\ \vdots \\ \hat{\varphi}(\|\mathbf{x} - \mathbf{x}^N\| \end{pmatrix}^T \hat{\Phi}^{-1} \begin{pmatrix} \hat{\varphi}(\|\mathbf{x} - \mathbf{x}^1\|) \\ \vdots \\ \hat{\varphi}(\|\mathbf{x} - \mathbf{x}^N\| \end{pmatrix}. \tag{VI.12}$$

Therefore, the Gaussian process generates $g(\mathbf{x})$ as an estimate of $f(\mathbf{x})$ and uses σ^2 to quantify the variance in the generated estimate (i.e., a statistical error bound for the generated estimate).

The model parameters $\kappa_1, \kappa_2, \theta_1, \ldots, \theta_n$ are computed by maximizing the log-likelihood function $\ln(L)$ for the Gaussian process, where

$$\ln(L) = -\frac{1}{2}\ln\left(\det\left[\hat{\Phi}\right]\right) - \frac{1}{2}\begin{pmatrix} f_1 \\ \vdots \\ f_N \end{pmatrix}^T \hat{\Phi}^{-1} \begin{pmatrix} f_1 \\ \vdots \\ f_N \end{pmatrix} - \frac{N}{2}\ln(2\pi)$$

and $\det\left[\hat{\Phi}\right]$ denotes the determinant of $\hat{\Phi}$. See Ref. [27] for detailed analysis and implementation of using the maximum likelihood estimation for tuning the model parameters in the Gaussian process.

Note that if $\kappa_1 = 1$ and $\kappa_2 = 0$, then $\hat{\Phi}$ becomes the Gaussian RBF interpolation matrix. Therefore, one could also use Gibbs and MacKay's procedure to tune the model parameters for the Gaussian RBF interpolation. However, due to lack of positive definiteness of interpolation matrices of other RBFs, the maximum likelihood estimation is not applicable for model parameter tuning of non-Gaussian RBF interpolation.

Chapter VII

OTHER DATA FITTING METHODS

VII.1 INTRODUCTION

In this chapter, we give a brief discussion of the least polynomial interpolation of the data and the least squares fitting of the data.

Polynomial fitting of a given data set $\{(\mathbf{x}^1, f_1), \ldots, (\mathbf{x}^N, f_N)\}$ is the simplest and certainly the most widely used technique for data fitting. Polynomials owe this popularity to their simple structure, well understood algebraic properties, moderate flexibility of shapes, and computationally simple implementation. However, polynomials also have their limitations. For example, polynomials have poor extrapolatory properties, i.e., polynomials may provide good fits within the range of data, but they will frequently deteriorate rapidly outside the range of the data. High degree polynomials are notorious for unnecessary oscillations between data points. There is a tradeoff between the shape and degree of polynomials. To model data with a complicated structure, the degree of the polynomial must be high. However, a high degree may cause numerical instability during evaluation of the polynomials.

It is well-known that a multivariate polynomial interpolation of the data might not exist if polynomials of a fixed degree are used. There are several methods for finding a multivariate polynomial interpolation of the data [2]. The most promising method is the Least Polynomial Interpolation (LPI) method by de Boor and Ron [32, 33, 34, 35], which shall be presented in section VII.2; while least squares fitting methods are covered in section VII.3.

VII.2 LEAST POLYNOMIAL INTERPOLATION

As mentioned in the previous subsection, polynomial interpolants tend to be unnecessarily oscillatory when the degree of the polynomial is high. Therefore, it is of interest to obtain interpolants with degree as low as possible. The Least Polynomial Interpolation (LPI) by de Boor and Ron can be used to compute a polynomial interpolation of finitely many data points in a finite dimensional space. Given any finite set of data points, denoted by $\{(\mathbf{x}^1, f_1), (\mathbf{x}^2, f_2), \ldots, (\mathbf{x}^N, f_N)\}$, the goal is to determine a corresponding polynomial space with the least degree from which interpolation is

possible and can be determined uniquely.

We shall give a brief description on how the LPI interpolant is generated and list the most prominent properties as presented by de Boor and Ron in [35]. To distinguish the power of a scalar or vector from the superscript for a scalar or vector, we use the following form for a power term of $\mathbf{x}$:

$$\mathbf{x}^{\vec{\omega}} = (x_1)^{\omega_1}(x_2)^{\omega_2}\dots(x_n)^{\omega_n}, \tag{VII.1}$$

where $\vec{\omega}$ is a vector whose components are nonnegative integers $\omega_1, \dots, \omega_n$, and $(x_i)^{\omega_i}$ denotes x_i raised by the power of ω_i. The degree of $\mathbf{x}^{\vec{\omega}}$ is $|\vec{\omega}|$, the sum of the components of $\vec{\omega}$, i.e., $|\vec{\omega}| = \omega_1 + \omega_2 + \dots + \omega_n$. The degree coupled with the lexicographic order can be used to define the following order for $\mathbf{x}^{\vec{\omega}}$:

$\mathbf{x}^{\vec{\omega}} \prec \mathbf{x}^{\vec{\omega}'}$ if $|\vec{\omega}| < |\vec{\omega}'|$ or $|\vec{\omega}| = |\vec{\omega}'|$ with $\vec{\omega}$ preceding $\vec{\omega}'$ in the lexicographic order.

Using this order, we can arrange the power terms of $\mathbf{x}$ with $n = 2$ as follows:

$$1, \mathbf{x}^{(0,1)}, \mathbf{x}^{(1,0)}, \mathbf{x}^{(0,2)}, \mathbf{x}^{(1,1)}, \mathbf{x}^{(2,0)}, \mathbf{x}^{(0,3)}, \mathbf{x}^{(1,2)}, \mathbf{x}^{(2,1)}, \mathbf{x}^{(3,0)}, \dots$$

The LPI interpolant can be computed as follows. First, generate the Vandermonde matrix V for the data set, where the rows of V are indexed by the input vectors $\mathbf{x}^j$ for $j = 1, \dots, N$, and the columns of V correspond to the power terms of $\mathbf{x}$ in the order described above. That is, the entry in the jth row and the kth column of V is $(\mathbf{x}^j)^{\vec{\omega}}$ with ω being the kth element in the ordered sequence of the power terms of $\mathbf{x}$. If $n = 2$, $N = 4$, and $\mathbf{x}^1 = (0,0)^T$, $\mathbf{x}^2 = (1,2)^T$, $\mathbf{x}^3 = (2,4)^T$, $\mathbf{x}^4 = (2,2)^T$, then the entry at the 2nd row and 5th column of V is $(\mathbf{x}^2)^{(1,1)} = 2$. In fact, for this set of data, V has the following form:

$$V = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \dots \\ 1 & 2 & 1 & 4 & 2 & 1 & \dots \\ 1 & 4 & 2 & 16 & 8 & 4 & \dots \\ 1 & 2 & 2 & 4 & 4 & 4 & \dots \end{pmatrix}. \tag{VII.2}$$

Next follows a degree-based Gaussian elimination with pivoting. In the elimination process, all the terms with the same degree d are considered as a vector term. In other words, the Gaussian elimination is performed on the block matrix, whose (j,d) entry is a row vector with components $(\mathbf{x}^j)^{\vec{\omega}}$, where $|\vec{\omega}| = d$ and the components are arranged in the lexicographic order for $\vec{\omega}$. For $n = 2$, the $(j,2)$ entry in the block

matrix is the row vector $((\mathbf{x}^j)^{(0,2)}, (\mathbf{x}^j)^{(1,1)}, (\mathbf{x}^j)^{(2,0)})$. A special inner product is used for the vectors in the dth column of the block matrix:

$$\langle \mathbf{y}, \mathbf{z} \rangle_d = \sum_{\vec{\omega}=(\omega_1,\ldots,\omega_n),|\vec{\omega}|=d} \frac{y(\vec{\omega})z(\vec{\omega})}{\omega_1!\ldots\omega_n!},$$

where $y(\vec{\omega})$ or $z(\vec{\omega})$ denotes the component of $\mathbf{y}$ or $\mathbf{z}$ corresponding to the column with power index $\vec{\omega}$ in V. Starting with the block version of V as the working array $\mathbf{W}$, the degree-based Gaussian elimination with pivoting can be carried out as follows. At the jth step, we look for the smallest $d_j \geq d_{j-1}$ for which there is a nontrivial entry of the block matrix $\mathbf{W}$ in column d_j at or below row j. Then we find a largest such entry in terms of $\langle \cdot, \cdot \rangle_{d_j}$ and, if necessary, interchange its row with row j of $\mathbf{W}$ to bring it into the pivot position. For example, the first step of the degree-based Gaussian elimination for $\mathbf{W} = V$ in Eq. (VII.2) is the standard Gaussian elimination because the block column is the first column. The resulting block matrix is

$$\mathbf{W} = \begin{pmatrix} 1 & (0,0) & (0,0,0) & \ldots \\ 0 & (2,1) & (4,2,1) & \ldots \\ 0 & (4,2) & (16,8,4) & \ldots \\ 0 & (2,2) & (4,4,4) & \ldots \end{pmatrix}. \tag{VII.3}$$

The entry at the 3rd row and 2nd column of $\mathbf{W}$ is the largest in the 2nd column at or below the 2nd row, with $\langle (4,2), (4,2) \rangle_2 = 4 \cdot 4 + 2 \cdot 2 = 20$. Thus, the pivot rule exchanges the 2nd and 3rd rows of the working array $\mathbf{W}$ to yield the following updated $\mathbf{W}$:

$$\mathbf{W} = \begin{pmatrix} 1 & (0,0) & (0,0,0) & \ldots \\ 0 & (4,2) & (16,8,4) & \ldots \\ 0 & (2,1) & (4,2,1) & \ldots \\ 0 & (2,2) & (4,4,4) & \ldots \end{pmatrix}. \tag{VII.4}$$

The jth elimination step is to make the vector entries under the (j, d_j) entry of $\mathbf{W}$ orthogonal to the vector entry at row j and column d_j of $\mathbf{W}$ with respect to the inner product $\langle \cdot, \cdot \rangle_{d_j}$. For the block matrix in Eq. (VII.4), the elimination step yields the following matrix:

$$\mathbf{W} = \begin{pmatrix} 1 & (0,0) & (0,0,0) & \ldots \\ 0 & (4,2) & (16,8,4) & \ldots \\ 0 & (0,0) & (-4,-2,-1) & \ldots \\ 0 & \left(-\frac{2}{5}, \frac{4}{5}\right) & \left(-\frac{28}{5}, -\frac{4}{5}, \frac{8}{5}\right) & \ldots \end{pmatrix}. \tag{VII.5}$$

Then the next pivot step converts the block matrix in "row echelon form:"

$$\mathbf{W} = \begin{pmatrix} 1 & (0,0) & (0,0,0) & \dots \\ 0 & (4,2) & (16,8,4) & \dots \\ 0 & \left(-\frac{2}{5}, \frac{4}{5}\right) & \left(-\frac{28}{5}, -\frac{4}{5}, \frac{8}{5}\right) & \dots \\ 0 & (0,0) & (-4,-2,-1) & \dots \end{pmatrix}. \tag{VII.6}$$

Here the block matrix $\mathbf{W}$ is said to be in row echelon form if there is a nondecreasing sequence $d_1, d_2, \dots, d_N$ such that the vector entry at row j and column d_j is the first nonzero entry in row j and all the vectors in column d_j below row j are orthogonal to the vector entry at row j and column d_j (in terms of inner product $\langle \cdot, \cdot \rangle_{d_j}$).

The degree-based Gaussian elimination with pivoting leads to the factorization $\Gamma V = L\mathbf{W}$, where Γ is a permutation matrix, $\mathbf{W}$ is a row echelon block matrix, and L is a unit lower triangular matrix. For V in Eq. (VII.2), we have

$$\Gamma V = L\mathbf{W} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & \frac{3}{5} & 1 & 0 \\ 1 & \frac{1}{2} & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & (0,0) & (0,0,0) & \dots \\ 0 & (4,2) & (16,8,4) & \dots \\ 0 & \left(-\frac{2}{5}, \frac{4}{5}\right) & \left(-\frac{28}{5}, -\frac{4}{5}, \frac{8}{5}\right) & \dots \\ 0 & (0,0) & (-4,-2,-1) & \dots \end{pmatrix}. \tag{VII.7}$$

To obtain the basis functions for the interpolation polynomial space, we need a further factorization $\mathbf{W} = U\mathbf{G}$, where U is an upper triangular matrix obtained by a degree-based backward elimination, i.e., for each $j = N, N-1, \dots, 1$, we scale the vector $\mathbf{W}_{j,d_j}$ at row j and column d_j of the block matrix $\mathbf{W}$ by $\langle \mathbf{W}_{j,d_j}, \mathbf{W}_{j,d_j} \rangle_{d_j}$ and enforce the vector entries above the vector $\mathbf{W}_{j,d_j}$ to be orthogonal to $\mathbf{W}_{j,d_j}$ (in terms of inner product $\langle \cdot, \cdot \rangle_{d_j}$). For matrix $\mathbf{W}$ given in Eq. (VII.6), the degree-based backward elimination by de Boor and Ron generates an upper triangular matrix U:

$$U = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 20 & 0 & -50 \\ 0 & 0 & \frac{4}{5} & 12 \\ 0 & 0 & 0 & \frac{25}{2} \end{pmatrix} \tag{VII.8}$$

with the following factorization of $\mathbf{W}$:

$$\mathbf{W} = U\mathbf{G} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 20 & 0 & -50 \\ 0 & 0 & \frac{4}{5} & 12 \\ 0 & 0 & 0 & \frac{25}{2} \end{pmatrix} \begin{pmatrix} 1 & (0,0) & (0,0,0) & \dots \\ 0 & \left(\frac{1}{5}, \frac{1}{10}\right) & (0,0,0) & \dots \\ 0 & \left(-\frac{1}{2}, 1\right) & \left(-\frac{11}{5}, \frac{7}{5}, \frac{16}{5}\right) & \dots \\ 0 & (0,0) & \left(-\frac{8}{25}, -\frac{4}{25}, -\frac{2}{25}\right) & \dots \end{pmatrix}. \tag{VII.9}$$

Using matrices L and U, one can construct a set of basis functions $\psi_1(\mathbf{x}), \ldots, \psi_N(\mathbf{x})$ for the interpolation polynomial space and the LPI for the given data points. The LPI interpolant has the following form:

$$p(\mathbf{x}) = \sum_{j=1}^{N} \alpha_j \psi_j(\mathbf{x}),$$

where

$$\begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{pmatrix} = \mathrm{diag}(U)(LU)^{-1} \begin{pmatrix} f(\mathbf{x}^1) \\ f(\mathbf{x}^2) \\ \vdots \\ f(\mathbf{x}^N) \end{pmatrix}$$

and

$$\psi_j(\mathbf{x}) = \sum_{\vec{\omega}=(\omega_1,\ldots,\omega_N),|\vec{\omega}|=d_j} \frac{\mathbf{G}_j(\vec{\omega})}{\omega_1! \ldots \omega_N!} \mathbf{x}^{\vec{\omega}}. \tag{VII.10}$$

Here $\mathrm{diag}(U)$ denotes the diagonal matrix whose diagonal entries are the corresponding diagonal entries in U, $\mathbf{G}_j(\vec{\omega})$ is the component of the vector entry at row j and column d_j corresponding to the power index $\vec{\omega}$, and $\psi_1(\mathbf{x}), \ldots, \psi_N(\mathbf{x})$ form a set of basis functions for the polynomial interpolation space.

Corresponding to $\mathbf{G}$ in Eq. (VII.9), the basis functions are: $\psi_1(\mathbf{x}) = 1, \psi_2(\mathbf{x}) = \frac{1}{5}(\mathbf{x})^{(0,1)} + \frac{1}{10}(\mathbf{x})^{(1,0)}, \psi_3(\mathbf{x}) = -\frac{1}{2}(\mathbf{x})^{(0,1)} + (\mathbf{x})^{(1,0)}, \psi_4(\mathbf{x}) = -\frac{4}{25}(\mathbf{x})^{(0,2)} - \frac{4}{25}(\mathbf{x})^{(1,1)} - \frac{1}{25}(\mathbf{x})^{(2,0)}$. For more details on the degree-based Gauss elimination with pivoting, see Ref. [35].

The name of LPI comes from the following fact. Let $P_{\min}$ be the space generated by linear combinations of $\psi_1(\mathbf{x}), \ldots, \psi_N(\mathbf{x})$. For any subspace P^* of algebraic polynomials, if the system of interpolation equations, $p(\mathbf{x}^j) = f_j$ for $j = 1, \ldots, N$ with constraint $p \in P^*$, always has a solution no matter what values of $f_1, \ldots, f_N$ are, then the highest degree of polynomials in P^* is no less than the highest degree of polynomials in $P_{\min}$.

The interpolation polynomial space $P_{\min}$ also has many other interesting properties established by de Boor and Ron [35], such as

- **uniquely defined by** $\{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$: the basis functions are independent of how the input vectors are ordered;
- **translation invariance**: for any $p(\mathbf{x}) \in P_{\min}$ and any fixed point $\hat{\mathbf{x}} \in \mathbb{R}^n$, $p(\mathbf{x} + \hat{\mathbf{x}})$ is still a polynomial in $P_{\min}$;

- **scale invariance**: for any $p(\mathbf{x}) \in P_{\min}$ and any fixed scalar $\alpha \in \mathbb{R}$, $p(\alpha\mathbf{x})$ is still a polynomial in $P_{\min}$;
- **coordinate-system independence**: an affine transformation of variables affects the LPI polynomial space in a "reasonable" way;
- **monotonicity**: if $\{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$ is a subset of $\{\hat{\mathbf{x}}^1, \ldots, \hat{\mathbf{x}}^{\hat{N}}\}$, then $P_{\min}$ is a subset of the least polynomial interpolation space corresponding to $\{\hat{\mathbf{x}}^1, \ldots, \hat{\mathbf{x}}^{\hat{N}}\}$.

We use the MATLAB LPI code developed by de Boor and Ron [32, 33, 34, 35] to obtain the LPI of the wing weight data. The maximum degree of the polynomial basis functions for the LPI is 2 for the wing weight data interpolation. That is, LPI generates a quadratic polynomial interpolation of the weight data.

The LPI method can be very sensitive toward the location of the data points [2]. However, in their MATLAB LPI code, de Boor and Ron, use an optional tolerance parameter ($tol > 0$) that gives the method the ability to search for a set of basis functions such that the corresponding interpolation matrix has a better condition number than the interpolation matrix of the LPI, but the maximum degree of the basis functions may be higher than the LPI. It was recommended [2] to use $tol > 0$ to obtain a polynomial interpolant that is less sensitive to data locations, when the condition number of the interpolation matrix is too large. For the wing weight data fitting problem, the condition number of the interpolation matrix is small, therefore there is no justification for using polynomial interpolants of higher degree.

The relationship between the value of *tol* and the degree of the generated LPI is not straightforward. The appropriate *tol* value for obtaining a polynomial interpolant different from LPI depends on the given data points. For the wing weight data set, $tol = \frac{1}{100}$ leads to a cubic polynomial interpolant. If we continue to increase the value of *tol*, the maximum degree of the generated polynomial interpolant will also increase. For $tol < \frac{1}{100}$, the generated polynomial interpolant remains to be quadratic.

VII.3 LEAST SQUARES FITTING

While only interpolation methods are used for building wing weight estimation models in this thesis study, it is important to give a brief introduction of the least squares fitting methods [36].

The method of least squares assumes that the best-fit of a set of data $\{(\mathbf{x}^1, f_1), \ldots, (\mathbf{x}^N, f_N)\}$ by a given parametric model $g(\mathbf{x}; \alpha_1, \ldots, \alpha_M)$ is the one that

achieves the minimal sum of the deviations squared (least square error) from the data. The corresponding optimization problem can be formulated as follows:

$$\min_{\alpha_1,\ldots,\alpha_M} \sum_{j=1}^{N} \Big(g(\mathbf{x}^j;\alpha_1,\ldots,\alpha_M) - f_j\Big)^2. \tag{VII.11}$$

Least squares problems are classified as linear and nonlinear least squares problems depending on whether $g(\mathbf{x};\alpha_1,\ldots,\alpha_M)$ is a linear or nonlinear function of $\alpha_1,\ldots,\alpha_M$. For the polynomial least squares problem, $g(\mathbf{x};\alpha_1,\ldots,\alpha_M) = \sum_{j=1}^{M} \alpha_j p_j(\mathbf{x})$, where $p_1(\mathbf{x}),\ldots,p_M(\mathbf{x})$ form a basis of a subspace of all polynomials (such as the polynomials have degree at most m). In this case, Eq. (VII.11) is referred to as a linear least squares problem and its solution can be obtained by solving a system of linear equations [36]. For wing weight fitting by the geometry model,

$$g(\mathbf{x};\alpha_1,\ldots,\alpha_9) = \\ \alpha_1\Big[\mu^{\alpha_2}\,(0.01b)^{\alpha_3}\,\left(10^{-3}s\right)^{\alpha_4}\,(t_r)^{\alpha_5}\,(0.1c_r)^{\alpha_6}\,(\cos\Lambda)^{\alpha_7}\,(0.1c_t)^{\alpha_8}\,\left(10^{-5}w_{\text{to}}\right)^{\alpha_9}\Big],$$

where $n = 8$, $x_1 = \mu$, $x_2 = b$, $x_3 = s$, $x_4 = t_r$, $x_5 = c_r$, $x_6 = \Lambda$, $x_7 = c_t$, and $x_8 = w_{\text{to}}$. In this case, Eq. (VII.11) is referred to as a nonlinear least squares problem and its global optimal solution is difficult to compute.

Unlike linear least squares problems, whose estimates of the parameters can always be obtained analytically, nonlinear least squares problems require the use of iterative optimization procedures to compute the parameter estimates. The use of iterative procedures implies the need for initial guesses for the unknown parameters before the start of the optimization process. The initial guesses must be reasonably chosen, otherwise the optimization procedure may not converge or converge to a local minimum rather than the global minimum that defines the least squares estimates. We use the nonlinear optimization code **lsqnonlin** in MATLAB to solve the nonlinear least squares problem. For the wing weight data fitting by the geometry model or the ratio model, multiple initial guesses including the parameters values obtained by engineers and other random ones were used to search for a global optimal solution.

In Chapter IV, both forward and backward variable selection procedures use the nonlinear optimization code **lsqnonlin** in MATLAB to solve the nonlinear least squares problems. During that optimization processes, we also use multiple initial guesses to search for a global minimum (see MATLAB code in sections A.1.1 and A.1.2).

Chapter VIII

COMPARISON OF CONSTRUCTED APPROXIMATIONS

VIII.1 INTRODUCTION

For scarce and poorly distributed data (like wing weight data) in a high-dimensional space, it is difficult to find an appropriate approximation to model the data. But it is more problematic to verify if a constructed approximation has all the desirable properties that the underlying function should possess. Because no variable screening method is appropriate for identifying significant input variables for fitting historical data by interpolation methods, the numerical experiments are designed to help the user understand whether PCR could help to extract meaningful relationships between a general set of input variables and the output. Two candidate sets of input variables are used in the numerical experiments: (i) the set of eight input variables in Eq. (II.1) and (ii) the set of fourteen variables (all the data attributes in the wing weight data set listed in Table 1 except the mean thickness-to-chord ratio $[t/c]_m$). The first set is used to study whether a general approximation method could generate better wing weight estimation models than the knowledge-based engineering model (II.1); while the second set tests whether PCR is less effective if all possible input variables are used.

Note that PCR coupled with CV assigns various weights to the input variables in the interpolation model so that the resulting wing weight approximation model has the highest accuracy in predicting the trend in the data when measured by the leave-one-out CV procedure. In contrast to a classical approximation problem where the input variables are given and the regression model is known, PCR for wing weight data fitting generates hundreds of completely different approximation models by using different interpolation models and input spaces for data fitting. The numerical experiments for the wing weight fitting problem will focus on the benefits of using PCR (a data-driven approximation process) and the strategy of choosing the most appropriate approximation out of many constructed approximation models.

By examining all the interpolants obtained by the PCA procedure, in addition to the LPI and Gaussian process interpolant, we try to understand whether an appropriate number of features in the input space allows the PCR interpolant to capture the "physical trends" buried in the data correctly.

VIII.2 CHORD APPROXIMATION OF WING CONFIGURATION

Even in the case of eight variables, not all input variables can change independently. For example, if the span b is fixed and the reference area s is changed, then one must also change c_r and/or c_t appropriately to make the wing configuration feasible. In practice, there is no exact relationship among b, s, c_r, and c_t. For conceptual design of subsonic transports, it is acceptable to assume that

$$c_r + c_t = \gamma c_m = \gamma s/b, \tag{VIII.1}$$

where γ is the average of $(c_r + c_t)/c_m$ values for the known wing configurations. Eq. (VIII.1) will be used to approximate the dependency relationship among b, s, c_r, and c_t. In the case of eight variables, there is only one dependency relationship among the input variables. However, in the case of fourteen variables, there are six dependency relationships among the input variables and the maximum set of independent variables has eight variables. The extra degree of freedom for the wing configuration with fourteen variables is the thickness or the thickness-to-chord ratio at the wing tip, which was considered to be insignificant for the wing weight prediction by system analysts of subsonic transports. The constructed approximation will be converted to a function of the following variables: $b, s, t_r/c_r, w_{\text{to}}, \lambda, \Lambda, \mu$, and t_t/c_t (only for the 14-variable case). That is, for a constructed wing weight approximation

$$\bar{w} = g\left(b, c_r, s, \frac{t_r}{c_r}, w_{\text{to}}, \lambda, \Lambda, \mu\right)$$

or

$$\bar{w} = g\left(A, b, c_m, c_r, c_t, s, t_r, t_t, \frac{t_r}{c_r}, \frac{t_t}{c_t}, w_{\text{to}}, \lambda, \Lambda, \mu\right),$$

the final wing weight estimation formula for conceptual design is

$$\bar{w} = g\left(b, \frac{\gamma s}{b(1+\lambda)}, s, \frac{t_r}{c_r}, w_{\text{to}}, \lambda, \Lambda, \mu\right) \tag{VIII.2}$$

or

$$\bar{w} = g\left(\frac{b^2}{s}, b, \frac{s}{b}, \frac{\gamma s}{b(1+\lambda)}, \frac{\gamma\lambda s}{b(1+\lambda)}, s, \frac{\gamma s t_r}{(1+\lambda)bc_r}, \frac{\gamma\lambda s t_t}{(1+\lambda)bc_t}, \frac{t_r}{c_r}, \frac{t_t}{c_t}, w_{\text{to}}, \lambda, \Lambda, \mu\right). \tag{VIII.3}$$

The purpose of using the fourteen variables is to understand whether the PCR with CV is capable of identifying the weight growth trend in terms of the relevant input variables among the fourteen variables. For example, if t_t/c_t is truly insignificant for

wing weight estimation, then a useful PCR with CV should assign a very small weight to t_t/c_t in the constructed approximation model (VIII.3) so that the estimated wing weight is not sensitive to changes in t_t/c_t.

VIII.3 DESIRABLE PROPERTIES OF WEIGHT APPROXIMATION

In practice, the standard approach for validation of a constructed approximation $g(\mathbf{x})$ of $f(\mathbf{x})$ is to (randomly) generate function values $f(\mathbf{x}^{N+1}), \ldots, f(\mathbf{x}^{\bar{N}})$, where $\mathbf{x}^j$ are in the region of interest, and check the prediction errors $|f(\mathbf{x}^j) - g(\mathbf{x}^j)|$ for $j = N+1, \ldots, \bar{N}$. If the prediction errors are acceptable, then $g(\mathbf{x})$ is considered as a validated approximation of $f(\mathbf{x})$. Of course, a large $\bar{N}$ and an even distribution of $\mathbf{x}^j$ leads to a validation that is more reliable than in the case of a small $\bar{N}$ or an uneven distribution of $\mathbf{x}^j$. This process is similar to training and validation of neural networks. Unfortunately, for historical data, this validation process is not applicable due to data shortage.

An alternative approach is to use known knowledge of the true physical response $f(\mathbf{x})$ for validation of $g(\mathbf{x})$. For wing weight estimation, a desirable approximation should have the following properties: w is an increasing function with respect to each of b, s, λ, and Λ; and w is a decreasing function with respect to t_r/c_r. These properties are derived from simple engineering rules on the relationships between the wing weight and each of the five key configuration parameters. However, it is impossible to check the monotonicity of a multivariate function with respect to one input variable by using visual inspection of the two dimensional plots of the wing weight versus the specified input variable because there are infinitely many choices for the other variables.

As a compromise, we will inspect N plots of the wing weight versus one of the five key configuration parameters that pass through the N data points, respectively. More specifically, for each j, except one of the five key configuration parameters $b, s, t_r/c_r, \lambda$, and Λ, we substitute $b, s, t_r/c_r, w_{\text{to}}, \lambda, \Lambda, \mu$, and t_t/c_t by the corresponding values of the jth wing configuration parameters in either Eq. (VIII.2) or Eq. (VIII.3). The resulting function is a relationship between the wing weight and the unsubstituted configuration parameter, which has one of the following forms: $\bar{w} = h_{j,1}(b)$, $\bar{w} = h_{j,2}(s)$, $\bar{w} = h_{j,3}(t_r/c_r)$, $\bar{w} = h_{j,4}(\lambda)$, or $\bar{w} = h_{j,5}(\Lambda)$. Here the index j in $h_{j,i}$ indicates that the resulting function depends on the jth wing configuration and the second index in $h_{j,i}$ indicates the dependence of the resulting function on the

unsubstituted configuration parameter. Then the desirable properties of an approximation are the following: $h_{j,1}(\cdot), h_{j,2}(\cdot), h_{j,4}(\cdot)$, and $h_{j,5}(\cdot)$ are increasing functions, while $h_{j,3}(\cdot)$ is a decreasing function. One rationale behind using the selected two dimensional plots for visual inspection is that if these plots show the desired properties of the approximation, then the approximation gives the correct trend predictions when system analysts start conceptual design with an existing configuration as the baseline.

However, it is difficult to find approximations with these desirable properties, perhaps due to insufficient information on weight trends in the data. For example, the trends in the plots of $h_{j,i}(\cdot)$ for the geometry model are the following: w increases as s or Λ increases, and w decreases as b or t_r/c_r or λ increases. That is, the trends are not desirable in terms of change of the wing weight versus b or λ.

VIII.4 IMPACTS OF PROBLEM FORMULATION

One problem with approximations generated by using chord approximation formula

$$c_r = \frac{\gamma s}{b(1+\lambda)}$$

after the data fitting is that the constructed approximation is not an interpolation due to errors in the substitution formula (VIII.1). Moreover, the substitution errors generally lead to approximations with less desirable two dimensional trends. See Fig. 11 for typical plots of the wing weight versus span, where the curves represent the approximations generated by applying the chord substitution before and after multiquadric PCR fitting

For the remainder of the thesis, the chord substitution formula

$$c_r = \frac{\gamma s}{b(1+\lambda)}$$

is applied to the interpolation models before using CV optimization to construct the approximation. That is, the components of $\mathbf{x}^j$ are the corresponding values of the algebraic expressions in either Eq. (VIII.2) or Eq. (VIII.3) for construction of wing weight estimation models.

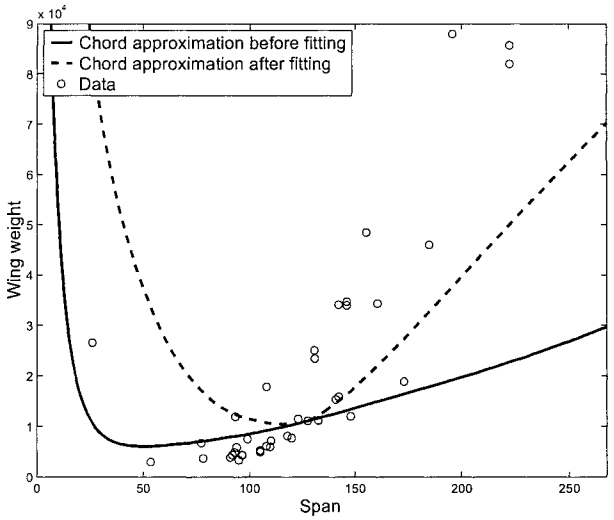

Figure 11: Differences between using chord approximation formula (VIII.1) before and after multiquadric PCR fitting.

VIII.5 COMPARISON OF INTERPOLATION MODELS

The CV error of a constructed approximation can be useful to help analysts decide which approximation is better. In Tables 2 and 3, the CV errors are for approximations obtained when the chord approximation formula $c_r = \frac{\gamma s}{b(1+\lambda)}$ is used after the data fitting. In Tables 4 and 5 we have the CV errors of all approximations obtained when the chord approximation formula $c_r = \frac{\gamma s}{b(1+\lambda)}$ is used before the data fitting. Again the row of $n = 8$ or $n = 14$ includes the CV errors for interpolation in the original $\mathbf{x}$-space (not in a feature space).

For wing weight data fitting with a fixed type of interpolation models, the "best" approximation model among all the approximations generated by PCR (for a range of r) usually corresponds to the smallest value of minimized CV errors. Here the criterion for best approximation is by a subjective judgment of overall desirable trends of the approximation by inspecting the five types of two dimensional plots for all forty-one baseline configurations. Note that this visual inspection is very time consuming because over 200 plots have to be inspected for each constructed approximation. Therefore, it makes sense to use the automatic PCR procedure in section VI.5 for

	Multiquadric CV Error	Thin Plate CV Error	Cubic CV Error	Gaussian CV Error	Kriging CV Error
$n = 14$	18585	2518	2898	1129900	6284700
$r = 14$	12755	3243	4208	70151	74722
$r = 13$	10711	3575	4069	42780	54585
$r = 12$	12540	2887	6395	72993	107630
$r = 11$	34515	3593	7711	73861	78001
$r = 10$	14235	3952	5780	50194	83175
$r = 9$	4653	3412	10209	394330	522840
$r = 8$	9497	14378	11512	43615	213940
$r = 7$	12425	5271	24341	24782	362550
$r = 6$	7242	10325	29184	98669	44993
$r = 5$	11686	92148	60937	28986	98159
$r = 4$	17609	25974	144140	17919	37089

Table 4: CV errors for the set of fourteen variables using the chord approximation formula before the data fitting.

	Multiquadric CV Error	Thin Plate CV Error	Cubic CV Error	Gaussian CV Error	Kriging CV Error
$n = 8$	5581	2870	6567	260310	27002
$r = 8$	5581	3772	5463	835360	43241
$r = 7$	5417	2961	7262	2214100	1084100
$r = 6$	5698	2506	5090	245960	178160
$r = 5$	5566	3852	8108	20800000	52056000
$r = 4$	5560	2864	19528	470920	12252000

Table 5: CV errors for the set of eight variables using the chord approximation formula before the data fitting.

choosing the best PCR for any fixed interpolation model. For example, the best PCR multiquadric approximation for the eight-variable case is constructed with $r = 7$ (see Table 5). Later on, we shall see that this approximation has the overall best trends for wing weight prediction.

Because the difference between the smallest and the largest wing weight values is considerable, one must verify whether the CV error value E^{CV} is dominated by the prediction errors at $\mathbf{x}^j$ with large wing weight values. In other words, we should compute the relative CV prediction errors for each data point, defined as

Airplane	Prediction	Rel. Error	Airplane	Prediction	Rel. Error
1049G	11932	3.8%	DC-7C	10623	-4.5%
C-46A	4274	-36.4%	F-28	5046	-28.7%
DC8F-54	34963	2.5%	B. 747	76456	-14.1%
DC9-30	11764	-1.5%	VC10-1101	33986	0.2%
VC10-1151	35565	2.5%	G-159	4399	18.8%
C-8A	5958	30.9%	AC-1	6151	59.5%
C-124A	19371	2.3%	KC-97E	15506	1.2%
C-123J	5879	-2.0%	XC-120	5327	-13.7%
C-119H	11894	-1.4%	C. 440	8057	46.9%
C. 340	4983	-6.9%	R4Y-2	5327	6.4%
C-131E	5174	3.9%	Con.110	3200	-9.2%
Con.T-29D	4147	-5.3%	Martin 404	5118	4.6%
V. V. 800	4878	-18.9%	DC-6B	8492	4.6%
B. 727	15741	-12.7%	CL44-D4-1	15411	-2.9%
C-130B	10908	-2.2%	Electra	8690	14.8%
B. 720	24341	3.6%	Jetstar	5866	66.7%
C5-A	77204	-6.1%	L-1011	35511	-25.7%
C-135A	29220	15.1%	22(880)	7280	5.6%
30A(990A)	26585	-0.1%	C-141A	35980	4.7%
C-5A	82296	-4.1%	F-27	6150	36.2%
DC10-10	48536	0.1%			

Table 6: Relative errors for the leave-one-out CV of multiquadric PCR with $r = 7$.

$(g_{-j}(\mathbf{x}^j) - f_j)/g_{-j}(\mathbf{x}^j)$ in Table 6, and check if the CV optimization attempts to minimize the prediction errors $(g_{-j}(\mathbf{x}^j) - f_j)^2$ for large f_j. One indicator for such a biased minimization of E^{CV} is that the relative errors for large $g_{-j}(\mathbf{x}^j)$ (or f_j) are smaller than those for small $g_{-j}(\mathbf{x}^j)$. But we don't see such a biased minimization of E^{CV} in Table 6, which shows the relative errors $(g_{-j}(\mathbf{x}^j) - f_j)/g_{-j}(\mathbf{x}^j)$ for $j = 1, \ldots, N$ when CV is applied to multiquadric PCR with $r = 7$. In fact, the big difference between the smallest and the largest wing weight values does not lead the CV optimization to minimize the prediction errors at $\mathbf{x}^j$ for large f_j. For PCR with Gaussian RBF, the relative errors are fairly uniform along the data points.

Fig. 12 shows the leave-one-out CV predictions $g_{-j}(\mathbf{x}^j)$ $(j = 1, \ldots, N)$ for the multiquadric PCR with $r = 7$ and the geometry model (II.1). It is clear that the multiquadric PCR with $r = 7$ has much more accurate predictions than the geometry

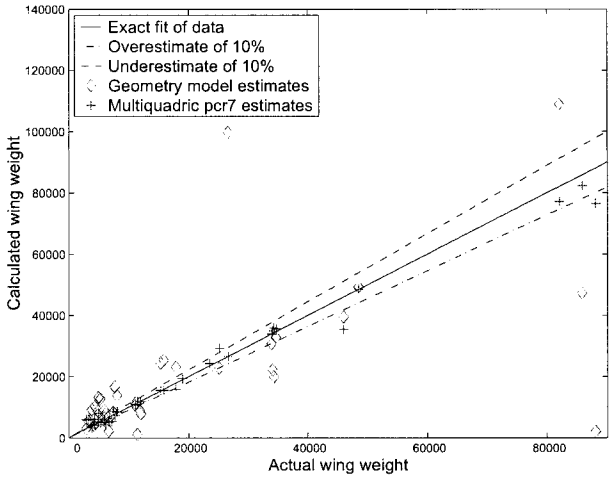

Figure 12: Leave-one out CV errors for geometry model and multiquadric PCR with $r = 7$ for the set of eight variables.

model (II.1). The main advantage of PCR with CV is its ability to explore the data and to tune the model for trend prediction.

Figs. 13 and 14 show the relative and absolute CV error distributions, respectively, at the data points for the multiquadric PCR with $r = 7$. For the relative error we divide the intervals $[0, 1]$ into 20 subintervals of length 0.05, and then plot the bar chart for the frequency of the CV errors fall into each subinterval. That is, each bar in Fig. 13 is the number of j such that $|g_{-j}(\mathbf{x}^j) - f_j|/g_{-j}(\mathbf{x}^j)$ is in the subinterval represented by its midpoint on the horizontal axis. For the absolute CV error the procedure is similar, only now we divide the interval from 0 to the maximum absolute error (about 12300) into 20 subintervals of length 615, and then plot the bar chart of the frequency of the CV errors fall into each subinterval.

For the wing weight fitting problem, Kriging and Gaussian RBF interpolants tend to create unnecessary oscillations between data points in comparison to the Gaussian process method – Tpros [27]. Nonetheless, all three Gaussian based methods present an exponential decay near the end of the data range, as expected. The reason is that the basis functions $\varphi(\|\mathbf{x} - \mathbf{x}^j\|)$ decreases at the exponential rate as $\|\mathbf{x} - \mathbf{x}^j\|$ increases (or the configuration is moving away from the existing configurations). LPI is extremely sensitive with respect to data points and LPI of the forty-one wing data

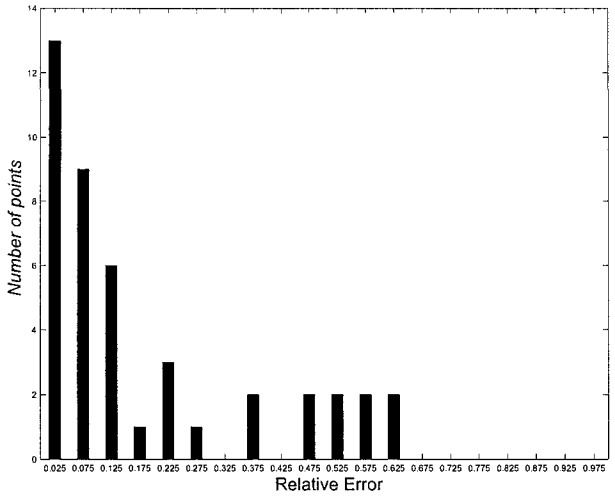

Figure 13: Relative CV error distribution at the data points.

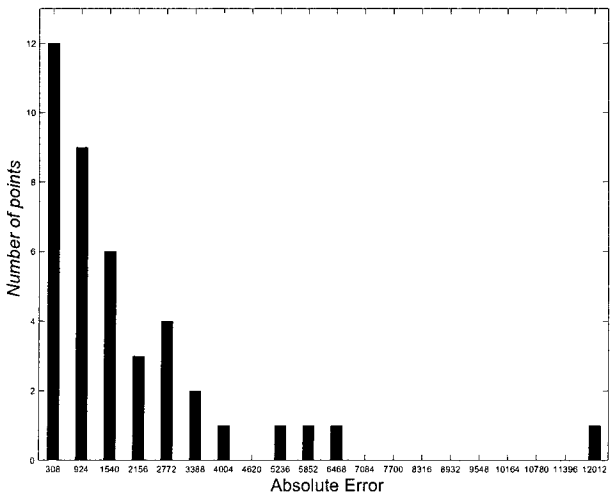

Figure 14: Absolute CV error distribution at the data points.

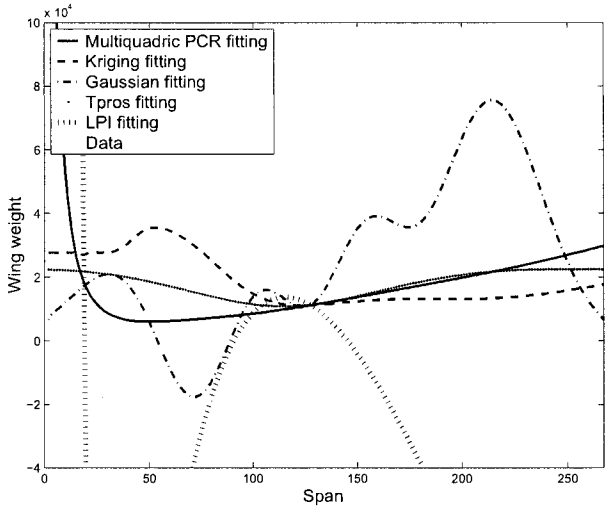

Figure 15: Wing weight versus span of constructed approximations.

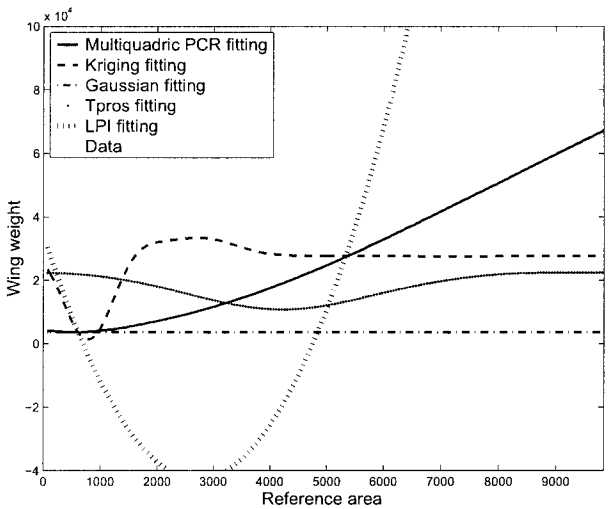

Figure 16: Wing weight versus reference area of constructed approximations.

points is very oscillatory. In comparison, an appropriate choice (multiquadric) of RBFs leads to nonoscillatory trend predictions of the wing weight. Figs. 15 and 16 show typical two-dimensional plots of the wing weight versus the reference area and span for various wing weight approximations.

VIII.6 BENEFITS OF PRINCIPAL COMPONENT REGRESSION

One major problem in fitting historical data is overfitting, i.e., unreliable minor trend changes might lead to undesirable characteristics (such as oscillations) in the approximation. The following five plots, along with Figs. 15 and 16, show the relationships between the wing weight and each of the five configuration parameters b, s, t_r/c_r, λ, and Λ. Points from Fig. 5 are added to the plots to give an indication of the scatter in the data. And a curve for the geometry model is added to indicate an approximation generated by systems analysts.

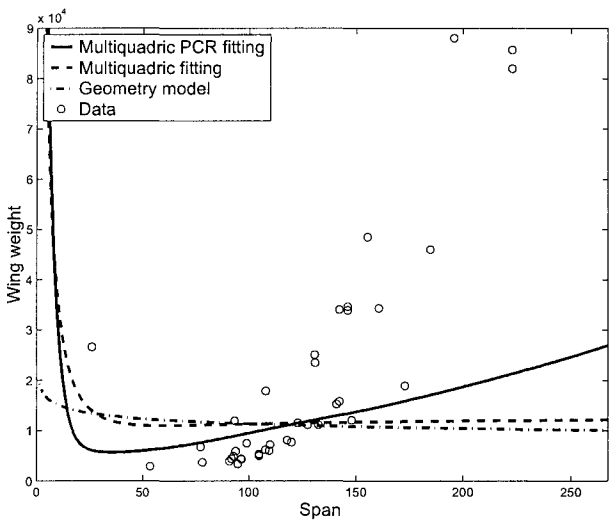

Figure 17: Differences between multiquadric PCR fitting and multiquadric fitting for span versus wing weight.

Almost all the two-dimensional plots for the multiquadric PCR fitting exhibit the desirable properties specified in section VIII.3, at least in a neighborhood of the baseline data point. In few cases, the wing weight is not a decreasing function of t_r/c_r. In many cases, the wing weight is not an increasing function of λ.

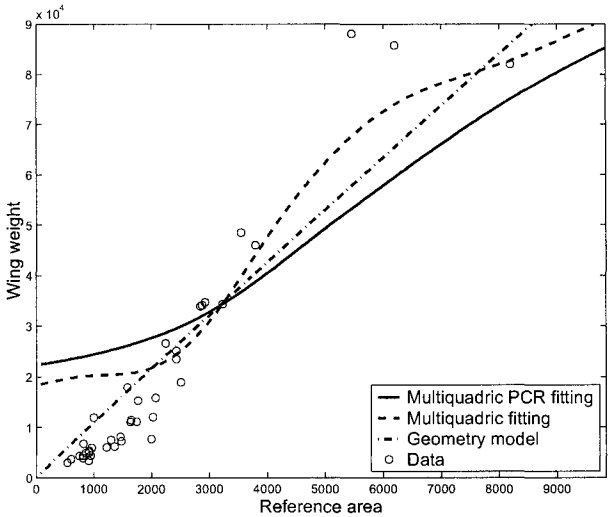

Figure 18: Differences between multiquadric PCR fitting and multiquadric fitting for reference area versus wing weight.

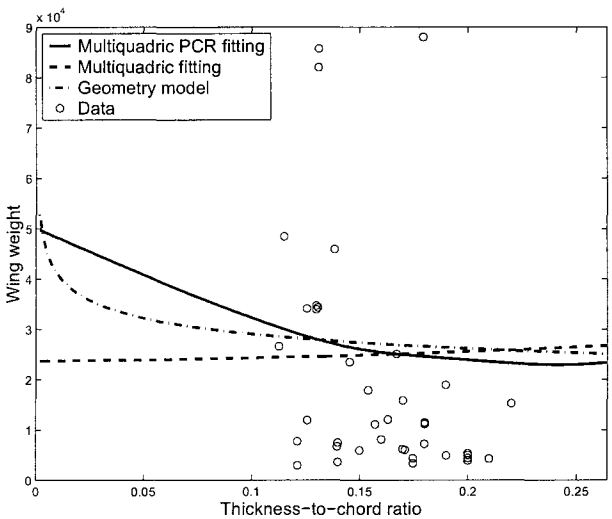

Figure 19: Differences between multiquadric PCR fitting and multiquadric fitting for thickness-to-chord ratio versus wing weight.

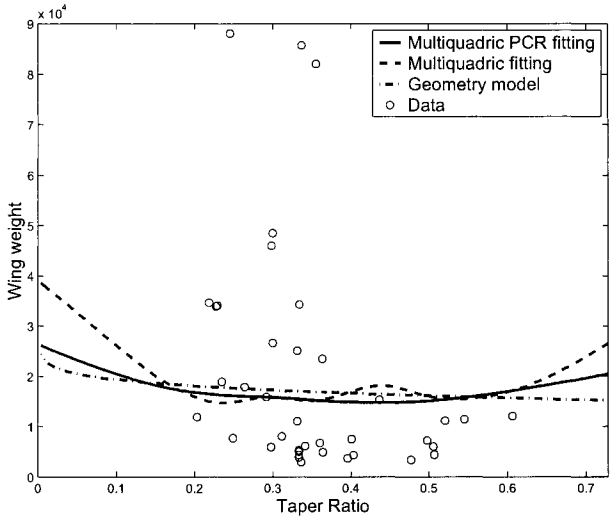

Figure 20: Differences between multiquadric PCR fitting and multiquadric fitting for taper ratio versus wing weight.

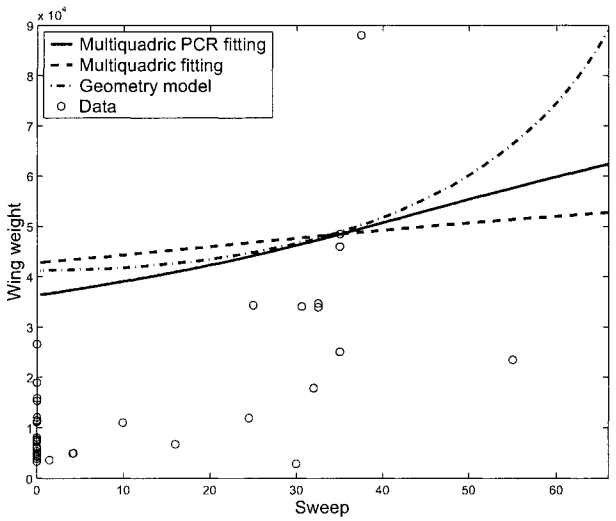

Figure 21: Differences between multiquadric PCR fitting and multiquadric fitting for sweep angle versus wing weight.

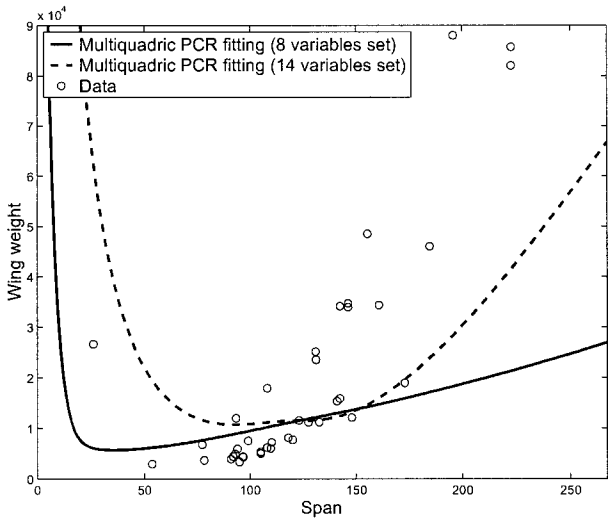

Figure 22: Wing weight versus span for multiquadric PCR fittings corresponding to $n = 8$ and $n = 14$.

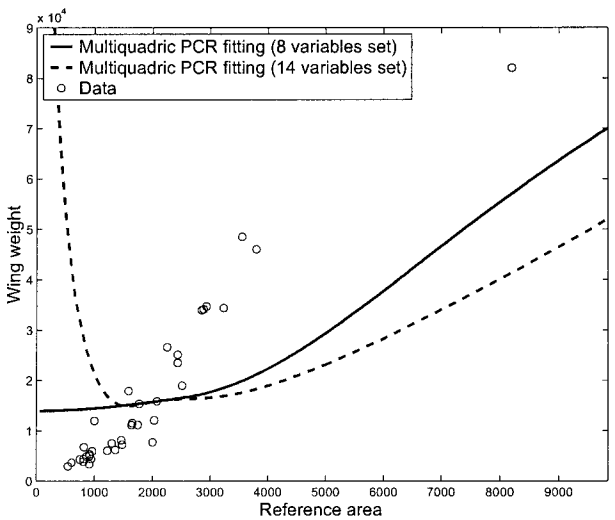

Figure 23: Wing weight versus reference area for multiquadric PCR fittings corresponding to $n = 8$ and $n = 14$.

The fourteen variables set is tested to understand whether the PCR with CV is capable of identifying the weight growth trend in terms of the relevant input variables among the fourteen variables. However, wing weight prediction trends shown in the two-dimensional plots are not as desirable as the multiquadric PCR with $r = 7$ for the eight variables set (see Figs. 22 and 23). Examination of the model parameters $\hat{\theta}_i$ in the fourteen variable case shows that the PCR with CV did not assign small values of $\hat{\theta}_i$ to any principal component. In other words, all principal components play important roles in construction of the approximation models. The implication is that expert knowledge of the underlying physical problem is essential for the wing weight approximation problem. One can not blindly build a meaningful approximation model of the response without a deep understanding of the underlying physical problem. On the other hand, the more desirable prediction behaviors of the multiquadric PCR with $r = 7$ when compared with the geometry model attests the benefit of coupling expert knowledge and intelligent approximation methods.

VIII.7 NUMERICAL ESTIMATION OF PREDICTION ERRORS

There is no theoretical foundation to choose the best wing weight approximation selected by using the desirable properties of wing weight approximations, because the desirable properties are based on simple engineering rules that are not applicable to all the possible values of the specified input variable. For example, if s and λ are fixed, then c_r approaches infinity as b goes to zero. There is a limitation on how small the span b can be before the configuration becomes unrealistic. Thus, one basic question is whether it is possible to provide some quantitative estimate on how accurate a wing weight approximation is. If Gaussian process or Kriging is used to construct the wing weight approximation, then each weight estimate is treated as the mean value of the unknown wing weight function and the associated standard deviation can be used as a quantitative estimate of the prediction error [28, 17, 18].

For nonstatistical fitting models such as RBF interpolation models, there is no quantitative estimate of prediction errors without information on $f(\mathbf{x})$. However, with a set of plausible approximations of the wing weight, one can use the differences among the plausible approximations as quantitative measures of prediction errors. For example, if two interpolants $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$ are plausible estimations of the wing weight by visual inspection, then the difference $|g_1(\mathbf{x}) - g_2(\mathbf{x})|$ provides a numerical estimation on how different the wing weight prediction could be due to a subjective

choice of the wing weight approximation. This idea leads to the following numerical estimation of wing weight prediction errors.

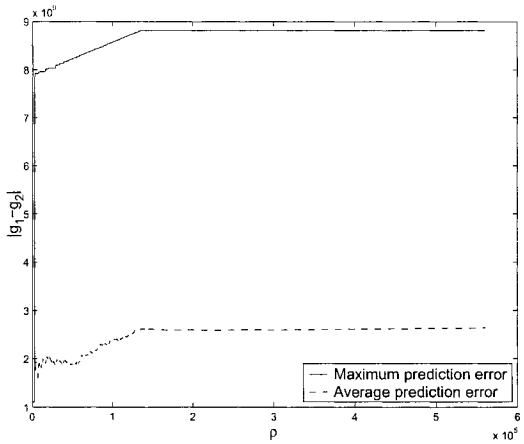

Figure 24: Maximum and average prediction errors based on the geometry model and the multiquadric PCR fitting over the concentric balls centered at the average of $\mathbf{x}^1, \ldots, \mathbf{x}^N$.

Numerical Estimation of Wing Weight Prediction Errors

- Generate $(\bar{N} - N)$ random points $\mathbf{x}^k$ $(k = N+1, \ldots, \bar{N})$ in the convex hull of the existing input vectors $\{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$, i.e., each $\mathbf{x}^k = \sum_{j=1}^{N} \epsilon_{jk} \mathbf{x}^j$ with $\epsilon_{jk} \geq 0$ and $\sum_{j=1}^{N} \epsilon_{jk} = 1$.

- Compute the maximum and average prediction errors as follows:

$$\max_{N+1 \leq k \leq \bar{N}, \mathbf{x}^k \in \Omega} |g_1(\mathbf{x}^k) - g_2(\mathbf{x}^k)|$$

and

$$\frac{1}{N_\Omega} \sum_{N+1 \leq k \leq \bar{N}, \mathbf{x}^k \in \Omega} |g_1(\mathbf{x}^k) - g_2(\mathbf{x}^k)|,$$

where $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$ are two acceptable approximations, Ω is a region of interest, and N_Ω is the number of $\mathbf{x}^{N+1}, \ldots, \mathbf{x}^{\bar{N}}$ in Ω.

However, there is no standard method for generating uniformly distributed random points in the convex hull of finitely many data points. One can use MATLAB

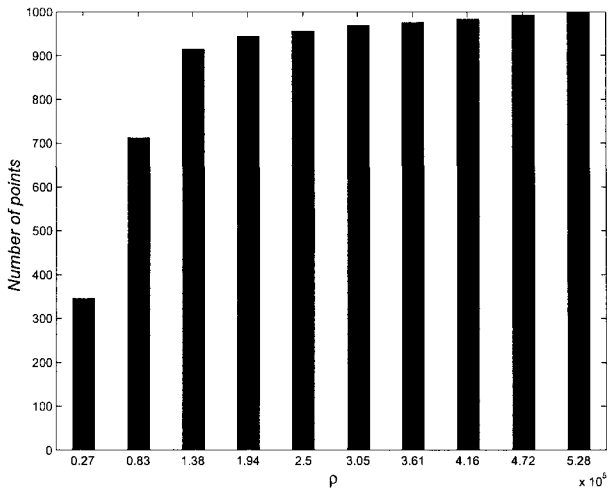

Figure 25: Cumulative frequency distribution of randomly generated data points $\mathbf{x}^{N+1}, \ldots, \mathbf{x}^{\bar{N}}$ over the concentric balls centered at the average of $\mathbf{x}^1, \ldots, \mathbf{x}^N$.

code **qhull** to construct convex hulls, Delaunay triangulations, halfspace intersections at a point, Voronoi diagrams, and other geometry configurations. MATLAB code **convhulln** uses **qhull** to determine the convex hull of N data points in $\mathbb{R}^n$. The solution generated by **convhulln** is an $l \times n$ matrix, where l is the number of the facets of the convex hull. Each row vector contains the indices of the data points that define a facet of the convex hull. For generation of convex hulls, **qhull** works well if n is small, say $n < 8$. In general, the size of the output and execution time grows in order of $N^{n/2}$. For example, to build a convex hull of 1000 points in $\mathbb{R}^{16}$, the number of facets of the convex hull might be of the order of 10^{24}. In the case of $N = 41$ and $n = 8$, the number of facets could be on the order of 2.8×10^6. For a randomly generated data point $\mathbf{x}^{N+j}$, it is nontrivial to check whether $\mathbf{x}^{N+j}$ is inside the convex hull generated by $\mathbf{x}^1, \ldots, \mathbf{x}^N$. Therefore, it is impractical to check if a point is inside the convex hull by using the facets of the convex hull. Moreover, if we generate uniformly distributed random points in a box containing $\mathbf{x}^1, \ldots, \mathbf{x}^N$, then almost all the points will be outside of the convex hull of $\mathbf{x}^1, \ldots, \mathbf{x}^N$ when the convex hull is contained in a lower dimensional subspace of $\mathbb{R}^n$.

An alternative is to generate random convex combinations of $\mathbf{x}^1, \ldots, \mathbf{x}^N$ directly.

That is, generate a random point inside the convex hull of $\mathbf{x}^1, \ldots, \mathbf{x}^N$ as follows:

$$\mathbf{x}^{N+j} = \frac{\epsilon_{j1}\mathbf{x}^1 + \epsilon_{j2}\mathbf{x}^2 + \ldots + \epsilon_{jN}\mathbf{x}^N}{\epsilon_{j1} + \epsilon_{j2} + \ldots + \epsilon_{jN}},$$

where $\epsilon_{j1}, \ldots, \epsilon_{jN}$ are randomly generated nonnegative numbers between 0 and 1. However, it is unclear whether $\mathbf{x}^{N+1}, \mathbf{x}^{N+2}, \ldots, \mathbf{x}^{\bar{N}}$ follow the uniform distribution as $\bar{N}$ approaches infinity.

In Fig. 24, $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$ are the geometry model and the multiquadric PCR fitting, respectively, and the regions of interest are the balls of radius ρ centered at the average of $\mathbf{x}^1, \ldots, \mathbf{x}^N$, denoted by ave($\mathbf{x}$). It is always true that the maximum prediction error is a nondecreasing function of ρ. However, the general increasing trend of average prediction error, as ρ increases, indicates that the difference between the geometry model and the multiquadric PCR fitting tends to become larger as the input vector moves away from ave($\mathbf{x}$). Fig. 25 shows the cumulative frequency distribution of $\mathbf{x}^{N+1}, \mathbf{x}^{N+2}, \ldots, \mathbf{x}^{\bar{N}}$ over the concentric balls, which indicates that most points are clustered around ave($\mathbf{x}$).

Chapter IX

CONCLUSIONS

It is very easy to fit a data set exactly by numerous methods no matter how the data points are distributed, but approximate responses are drastically different between the data points, even if they are almost identical at the data points. Principal Component Regression with cross-validation incorporates data mining into standard approximation processes so that the resulting approximation is less likely to overfit the data or to make predictions based on insufficient data information.

Polynomial based methods (e.g., LPI) are more sensitive to data change than radial basis function based methods. Among radial basis functions methods, Gaussian RBF methods (as well as Kriging), are more likely to perform poorly due to the exponential rate of decreasing of the Gaussian RBF as the distances between points increase. Gaussian process (Tpros) shows less oscillation than Gaussian RBF interpolation but has the same exponential decay trend outside the range of the data.

Variable screening could be a powerful tool for reducing the dimension of the input space for approximate responses if applicable. For real world problems of historical or measurement data fitting, most of the commonly used variable screening methods cannot be used and the few options left should be used with caution. Most of the times the only way to screen, or at least to identify, the possible least important variables is to simply analyze the data carefully.

A systematic principal component analysis procedure, that identifies the collinear or nearly collinear variables, is a powerful tool that customizes the corresponding regression method for the feature variables considered.

The assessment of the constructed approximations is of great importance. The physical properties of the problem, if well established, can be used as objective criteria to determine the ability of an approximation model to capture correctly the "physical trends" buried in the data. Cross-validation is much more than a simple parameter estimation tool. It can be used also as a tool to evaluate the performance of an approximation model as well as to indicate locations where addition of new data points would improve the prediction accuracy of the approximation model.

In general, the results obtained are quite satisfactory. We were able to obtain a general approximation that is more accurate and has more desirable properties than the best empirical response available. The biggest advantage is not the better

performance of the principal component regression in fitting the wing weight data of 41 subsonic transports, but the applicability of the principal component regression to general historical or measurement data fitting.

One area for improvement is the cross-validation error optimization. Cross-validation becomes more effective as the obtained optimal solution moves closer to a global minimum of the cross-validation error. Nonetheless, global optimization in a high dimensional space is beyond the scope of this thesis.

Other tasks for future work are the development of new variable screening procedures, improvement of the methods when the number of data points and the number of variables are close, and to understand better how the approximate responses can be affected by poor (or collinear) distribution of data points.

REFERENCES

[1] Raymer, D., *Aircraft Design: A Conceptual Approach*, Third Edition, AIAA, Reston, Virginia, 1999.

[2] Li, W., and Padula, S., "Approximation Methods for Conceptual Design of Complex Systems," *Approximation XI*, C. Chui, M. Neamtu, and L. Schumaker (eds.), Nashboro Press, Brentwood, TN, 2005, pp. 241–278.

[3] Ardema, M., Chambers, M., Patron, A., Hahn, A., Miura, H., and Moore, M., "Analytical Fuselage and Wing Weight Estimation of Transport Aircraft," NASA Technical Memorandum 110392, 1996.

[4] Keane, A., "Wing Optimization Using Design of Experiments, Response Surface, and Data Fusion Methods," *Journal of Aircraft*, Vol. 40, 2003, pp. 741–750.

[5] Powell, M.J.D., "Radial basis function methods for interpolation to functions of many variables," *HERMIS: International Journal of Computer Maths & Its Applications*, 2002, pp. 1–23.

[6] Micchelli, C., "Interpolation of scattered data: distance matrices and conditionally positive definite functions," *Constructive Approximation*, 1986, pp. 11–22.

[7] Friedman, J., and Stuetzle, W., "Projection pursuit regression," *Journal of the Amer. Stat. Assoc.*, 1981, pp. 817–823.

[8] Tu, J., and Jones, D.R., "Variable Screening in metamodel design by cross-validated moving least squares method," *44th AIAA*, 2003, Norfolk, Virginia.

[9] Rech, G., Teräsvirta, T., and Tschernig, R., "A Simple Variable Selection Technique for Nonlinear Models," *Commun. Statist. – Theory Meth.*, Vol. 30, No. 6, 2001, pp. 1227–1241.

[10] Buhmann, M., *Radial Basis Functions: Theory and Implementations*, Cambridge University Press, Cambridge, UK, 2003.

[11] Carr, J., Fright, W., and Beatson, R., "Surface interpolation with radial basis functions for medical imaging," *IEEE Transactions on Medical Imaging*, 1997, pp. 96–107.

[12] Powell, M.J.D., "Recent research at Cambridge on radial basis functions," *New Developments in Approximation Theory*, Internat. Ser. Numer. Math., Birkhauser, Basel, 1999, pp. 215–232.

[13] Baxter, C., "The interpolation theory of radial basis functions," Ph.D. Thesis, Cambridge University, Department of Applied Mathematics & Theoretical Physics (DAMTP), 2002.

[14] Schaback, R., and Wendland, H., "Characterization and Construction of Radial Basis Functions," N. Dyn, D. Leviatan, D. Levin and A. Pinkus: Multivariate Approximation and Applications, Cambridge University Press, 2001, pp. 1–24

[15] Schoenberg, I.J., "Metric spaces and completely monotone functions," *Ann. of Math.*, 1938, pp. 811–841.

[16] Hardy, R., "Theory and applications of the multiquadric-biharmonic method (20 years of discovery 1968-1988)," *Computers and Mathematics with Applications*, 1990, Issues 8-9, pp. 163–208.

[17] van Beers, W., and Kleijnen, J., "Kriging for interpolation in random simulation," *Journal of Oper. Res. Soc.*, 2003, pp. 255–262.

[18] van Beers, W., and Kleijnen, J., "Kriging interpolation in simulation: a survey," *Proceedings of the 2004 Winter Simulation Conference*, R. Ingalls, M. Rossetti, J. Smith, and B. Peter (eds.), Institute of Electrical and Electronics Engineers, Piscataway, NJ, 2004.

[19] Tu, J.,"Cross-validated multivariate metamodeling methods for physics-based computer simulations," *Proceedings of the IMAC-XXI*, 2003, Kissimmee, Florida.

[20] Stone, M., "Cross-Validatory Choice and Assessment of Statistical Predictions," *Journal of Royal Statistical Society*, Vol. 36, 1974, pp. 111–147.

[21] Efron, B., and Tibshirani, R.J., *An introduction to the Bootstrap*, Chapman & Hall, 1993.

[22] Fan, J., and Li, R., "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the Amer. Stat. Assoc.*, 2001, pp. 1348–1360.

[23] Li, R., and Lin, D., "Data analysis in supersaturated designs," *Stat. & Prob. Letters*, 2002, pp. 135–144.

[24] Li, R., and Sudjianto, A., "Analysis of computer experiments using penalized likelihood Gaussian Kriging model," *Proceedings of 2003 American Society of Mechanical Engineers (ASME) International Design Automation Conference*, DETC2003/DAC–48758, 2003, Chicago, Illinois, USA.

[25] Fan, J., and Gijbels, I., *Local Polynomial Modeling and Its Applications*, Chapman and Hall, London, 1996.

[26] Mandel, J., "Use of the singular value decomposition in regression analysis," *American Statistician*, 1982, pp. 15–24.

[27] Gibbs, M.N., and MacKay, D.J.C., "Efficient implementation of Gaussian processes," submitted to *Statistics and Computing.*

[28] Gibbs, M.N., "Bayesian Gaussian Processes for Regression and Classification," Ph.D. Thesis, Cambridge University, Department of Physics, 1997.

[29] Nelder, J.A., and Mead, R., "A simplex method for function minimization," *Comput. Journal*, 1965, pp. 308–313.

[30] Powell, M.J.D., "UOBYQA: Unconstrained Optimization By Quadratic Approximation," Technical report No. DAMTP2000/14, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England, 2000.

[31] Powell, M.J.D., "UOBYQA: Unconstrained Optimization By Quadratic Approximation," *Mathematical Programming*, 2002, pp. 555–582.

[32] de Boor, C., and Ron, A., "On multivariate polynomial interpolation," *Constr. Approx.*, 1990, pp. 287–302.

[33] de Boor, C., and Ron, A., "The least solution for the polynomial interpolation problem," *Math. Z. 210*, 1992, pp. 347–378.

[34] de Boor, C., "Polynomial interpolation in several variables," *Studies in Computer Science*, R. de Millo and J. R. Rice (eds.), Plenum Press, New York, 1994, pp. 87–119.

[35] de Boor, C., and Ron, A., "Computational aspects of polynomial interpolation in several variables," *Math. Comp.*, 1992, pp. 705–727.

[36] Björck, A., *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.

Appendix A

MATLAB CODES

A.1 VARIABLE SCREENING

A.1.1 Forward screening

```
% Wing Weight Estimation for Matlab
% Copyright (C) 2005 W. Li and H. Rocha
%
% Revision history:
%
%  1-MAR-2005: First version, W. Li and H. Rocha
%
%
% This program is free software; you can redistribute it and/or
% modify it under the terms of the GNU General Public License as
% published by the Free Software Foundation; A copy of the GNU
% General Public License can be obtained from the Free Software
% Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.


% FS - Forward Screening
% The Forward Screening algorithm is as follows:
%
%       1. Let g_i be the best fit of the simplified geometry model
%          representing the relationship between the ith input
%          variable and the wing weight, i.e., g_i is the best of
%          the univariate model obtained by setting the exponents
%          of the terms not related to x_i as zero.
%       2. Compute the sample coefficients of determination R_i^2
%          for g_i, where R_i^2 is the proportion of the total
%          variation in f^1,...,f^N explained by the simplified
%          geometry model and can be used as a metric for ranking
%          the significance of x_i in variation of the response.
```

```
%       3. If a simplified geometry model of k input variables is
%          desirable, then the input variables corresponding to
%          the k largest R_i^2 shall be selected as the significant
%          input variables.
%
%
% The function lsqfun.m is used.
%

clear all; cla; clf;
approx_option=0; %1 for using c_r+c_t=gamma*s/b before PCR analysis
rawdata=csvread('wing_data_10.csv');
% the actual input variable names
ref_area=rawdata(1,:);
aspect_ratio=rawdata(2,:);
sweep=rawdata(3,:);
root_chord=rawdata(4,:);
tip_chord=rawdata(5,:);
root_thick=rawdata(6,:);
tip_thick_to_chord=rawdata(7,:);
ultimate=rawdata(8,:);
gross_weight=rawdata(9,:);
wing_weight=rawdata(10,:);
% the derived variable names
span=sqrt(ref_area.*aspect_ratio);
taper_ratio=tip_chord./root_chord;
root_thick_to_chord=root_thick./root_chord;
tip_thick=tip_thick_to_chord.*tip_chord;
mean_chord=ref_area./span;
% modify the data if c_r+c_t=gamma*s/b is used before PCR analysis
if approx_option==1
   gamma=mean((root_chord+tip_chord)./mean_chord);
   root_chord=gamma*ref_area./(span.*(1+taper_ratio));
   tip_chord=root_chord.*taper_ratio;
```

```
   tip_thick=tip_thick_to_chord.*tip_chord;
   root_thick=root_thick_to_chord.*root_chord;
end
% define f and B for PCR
f_init=wing_weight';
n=8; N=41;
B(1,:)=ref_area;
B(2,:)=span;
B(3,:)=sweep;
B(4,:)=root_chord;
B(5,:)=tip_chord;
B(6,:)=root_thick;
B(7,:)=ultimate;
B(8,:)=gross_weight;
%
R2=zeros(n,1);
fbar=0;
for i=1:N
    fbar=fbar+f_init(i)/N;
end
den=0;
for i=1:N
     den=den+(f_init(i)-fbar)^2;
end
%
DTR=pi/180;
coef=zeros(8,1);
coef(1)=0.001;
coef(2)=0.01;
coef(4)=0.1;
coef(5)=0.11;
coef(6)=1;
coef(7)=1;
coef(8)=0.00001;
```

```
xcoef(1)=0.8795;
xcoef(2)=-0.1008;
xcoef(3)=-0.7842;
xcoef(4)=0.3720;
xcoef(5)=-0.0274;
xcoef(6)=-0.1580;
xcoef(7)=0.1895;
xcoef(8)=0.3469;
xcoef(9)=5150.6253;
ub=[10000000,10000000];
lb=-ub;
fprintf('Fitting_Error  Init_Error  Variance\n');
for k=1:8
    fnew=f_init;
    B1=B(k,:);
    N1=N;
    x0(1)=xcoef(k);
    x0(2)=xcoef(9);
    x0(1)=1;
    x0(1)=-0.15;
    x0(2)=fbar;
    err0 = sum(lsqfun(x0,B1,fnew,N1,k,coef,DTR).^2);
    %main process
    tol = 0.00001;
    options=optimset('Display','off','TolFun',tol);
    if k==3
        [x,err]=lsqnonlin(@lsqfun,x0,lb,ub,options,B1,fnew,...
        N1,k,coef,DTR);
        g=zeros(N1,1);
        for j=1:N1
            g(j)=x(2)*(cos(B1(j)*DTR))^x(1);
        end
        num=0;
        for j=1:N1
```

```
            num=num+(fnew(j)-g(j))^2;
        end
        R2(k)=1-num/den;
    else
        [x,err]=lsqnonlin(@lsqfun,x0,lb,ub,options,B1,fnew,...
        N1,k,coef,DTR);
        g=zeros(N1,1);
        for j=1:N1
            g(j)=x(2)*(coef(k)*B1(j))^x(1);
        end
        num=0;
        for j=1:N1
            num=num+(fnew(j)-g(j))^2;
        end
        R2(k)=1-num/den;
    end
    fprintf('  %.3e    %.3e   %.3e\n',err,err0,den);
end
%plot the bar chart
var=zeros(n,1);
for i=1:n
    var(i)=i;
end
clear id;
for i=1:n
  switch i
  case 1, id(i)=2;
  case 2, id(i)=4;
  case 3, id(i)=5;
  case 4, id(i)=1;
  case 5, id(i)=6;
  case 6, id(i)=8;
  case 7, id(i)=3;
  case 8, id(i)=7;
```

```
  end
  RR(i)=R2(id(i));
end
R2=RR;
bar(var,R2,0.4,'k');
set(gca,'xlim',[0.4,8.6]);
set(gca,'ylim',[0,1.05]);
xlabel('\fontsize{14}\rm Variables');
ylabel('\fontsize{14}\it R^2');
for i=1:8
  yshift=0.04;
  switch i
  case 4, lab='\fontsize{14}\it s';
  case 1, lab='\fontsize{14}\it b';
  case 7, lab='\fontsize{14} \Lambda';
  case 2, lab='\fontsize{14}\it c_r';
  case 3, lab='\fontsize{14}\it c_t';
  case 5, lab='\fontsize{14}\it t_r';
  case 8, lab='\fontsize{14} \mu';
  case 6, lab='\fontsize{14}{\it w}_{to}';
  end
  if R2(i)<0
     yshift=-yshift;
  end
  if i==6
    text(i-0.25,R2(i)+yshift,lab)
  else
    text(i-0.15,R2(i)+yshift,lab)
  end
end
set(gca,'YTick',[0:0.2:1]);

% Wing Weight Estimation for Matlab
% Copyright (C) 2005 W. Li and H. Rocha
%
```

```
% Revision history:
%
%  1-MAR-2005: First version, W. Li and H. Rocha
%
%
% This program is free software; you can redistribute it and/or
% modify it under the terms of the GNU General Public License as
% published by the Free Software Foundation; A copy of the GNU
% General Public License can be obtained from the Free Software
% Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

function LSQFUN = lsqfun(x,B1,fnew,N1,k,coef,DTR)

%
% function LSQFUN = lsqfun(x,B1,fnew,N1,k,coef,DTR)
%
%
% Input Arguments:
%   x            - parameters values
%   B1           - a matrix with the data points
%   fnew         - the actual wing weight of each data point
%   N1           - number of data points
%   k            - index of g_i
%   coef         - variables scaling coefficients
%   DTR          - constant to tranform degrees to radians
%

    if k==3
        LSQFUN=zeros(N1,1);
        for j=1:N1
            LSQFUN(j)=fnew(j)-x(2)*(cos(B1(j)*DTR))^x(1);
        end
    else
        LSQFUN=zeros(N1,1);
```

```
        for j=1:N1
            LSQFUN(j)=fnew(j)-x(2)*(coef(k)*B1(j))^x(1);
        end
    end
```

A.1.2 Backward screening

```
% Wing Weight Estimation for Matlab
% Copyright (C) 2005 W. Li and H. Rocha
%
% Revision history:
%
%  1-MAR-2005: First version, W. Li and H. Rocha
%
%
% This program is free software; you can redistribute it and/or
% modify it under the terms of the GNU General Public License as
% published by the Free Software Foundation; A copy of the GNU
% General Public License can be obtained from the Free Software
% Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.


% BS - Backward Screening
% The Backward Screening algorithm is as follows:
%
%       1. Let g be the best fit of the wing weight data by the
%          geometry model.
%       2. Let g_i be the best fit of the simplified geometry model
%          obtained by setting the exponent of the term related to
%          x_i as zero.
%       3. Compute the adjusted sample coefficients of determination
%          R^2 and R_i^2 for g and g_i, i=1,...,n.
%       4. If the difference in adjusted sample coefficients of
%          determination Delta R_i= R^2-R_i^2 is nonpositive for
%          some i, then the corresponding variable x_i could be
```

```
%          removed from the input vector and the simplified
%          geometry model would have n-1 variables.
%       5. Repeat the process with the simplified geometry model
%          until the number of input variables becomes desirable
%          or all Delta R_i^2 is greater than 0.
%
%
% The function nlsqfun.m is used.
%
clear all; clf; cla;
approx_option=0; %1 for using c_r+c_t=gamma*s/b before PCR analysis
rawdata=csvread('wing_data_10.csv');
% the actual input variable names
ref_area=rawdata(1,:);
aspect_ratio=rawdata(2,:);
sweep=rawdata(3,:);
root_chord=rawdata(4,:);
tip_chord=rawdata(5,:);
root_thick=rawdata(6,:);
tip_thick_to_chord=rawdata(7,:);
ultimate=rawdata(8,:);
gross_weight=rawdata(9,:);
wing_weight=rawdata(10,:);
% the derived variable names
span=sqrt(ref_area.*aspect_ratio);
taper_ratio=tip_chord./root_chord;
root_thick_to_chord=root_thick./root_chord;
tip_thick=tip_thick_to_chord.*tip_chord;
mean_chord=ref_area./span;
% modify the data if c_r+c_t=gamma*s/b is used before PCR analysis
if approx_option==1
   gamma=mean((root_chord+tip_chord)./mean_chord);
   root_chord=gamma*ref_area./(span.*(1+taper_ratio));
   tip_chord=root_chord.*taper_ratio;
```

```
   tip_thick=tip_thick_to_chord.*tip_chord;
   root_thick=root_thick_to_chord.*root_chord;
end
% define f and B for PCR
f=wing_weight';N=41;
den=var(f)*(N-1);
DTR=pi/180;
n=8;
xcoef(1)=0.1895;    % ultimate
xcoef(2)=-0.1008;   % b
xcoef(3)=0.8795;    % s_p
xcoef(4)=-0.1580;   % t_r
xcoef(5)=0.3720;    % c_r
xcoef(6)=-0.7842;   % sweep
xcoef(7)=-0.0274;   % c_tip
xcoef(8)=0.3469;    % gross weight
xcoef(9)=5150.6253; % leading coefficient
B(1,:)=ultimate;
B(2,:)=0.01*span;
B(3,:)=0.001*ref_area;
B(4,:)=root_thick;
B(5,:)=0.1*root_chord;
B(6,:)=cos(DTR*sweep);
B(7,:)=0.1*tip_chord;
B(8,:)=0.00001*gross_weight;
warning on;
R2i=zeros(8,1);
fprintf('   Fitting_Error   Init_Error  Best_Fitting\n');
for k=0:8
    ub=1000000000;
    lb=-ub;
    id=[1:9];
    id=id(id~=k);
    x_init=xcoef(id)*1.25;
```

```
    options=optimset('Display','off','TolFun',1.0e-10);
    err0 = sum(nlsqfun(x_init,B,f,N,k,n).^2);
    [x,err]=lsqnonlin(@nlsqfun,x_init,lb,ub,options,B,f,N,k,n);
    if k==0
       err_best=err;
       R2=1-((N-1)/(N-n))*err/den;
    else
       R2i(k)=1-((N-1)/(N-n+1))*err/den;
    end
    fprintf('%d:  %.5e   %.5e  %.5e\n',k,err,err0,err_best);
end

%plot the bar chart
var=zeros(n,1);
for i=1:n
    var(i)=i;
end
R2i=R2-R2i;
clear id;
for i=1:n
  switch i
  case 1, id(i)=2;
  case 2, id(i)=5;
  case 3, id(i)=7;
  case 4, id(i)=3;
  case 5, id(i)=4;
  case 6, id(i)=8;
  case 7, id(i)=6;
  case 8, id(i)=1;
  end
  RR(i)=R2i(id(i));
end
R2i=RR;
bar(var,R2i,0.4,'k');
```

```
set(gca,'xlim',[0.4,8.6]);
xlabel('\fontsize{14}\rm Variables');
ylabel('\fontsize{14}{\it \Delta{R^2}}');
ymin=min(R2i); ymax=max(R2i);
delta=0.00015;
set(gca,'ylim',[ymin,ymax]);
set(gca,'ylim',[-0.001,0.003]);
for i=1:8
  if R2i(i)<0
    yshift=R2i(i)-delta;
  elseif R2i(i)<0.003
    yshift=R2i(i)+delta;
  else
    yshift=-delta;
  end
  switch i
  case 4, lab='\fontsize{14}\it s';
  case 1, lab='\fontsize{14}\it b';
  case 7, lab='\fontsize{14} \Lambda';
  case 2, lab='\fontsize{14}\it c_r';
  case 3, lab='\fontsize{14}\it c_t';
  case 5, lab='\fontsize{14}\it t_r';
  case 8, lab='\fontsize{14} \mu';
  case 6, lab='\fontsize{14}{\it w}_{to}';
  end
  if i==6
    text(i-0.25,yshift,lab)
  else
    text(i-0.15,yshift,lab)
  end
end
set(gca,'YTick',[-0.001:0.001:0.001],'YTickLabel',...
{'-0.001','    0',' 0.001'});

% Wing Weight Estimation for Matlab
```

```
% Copyright (C) 2005 W. Li and H. Rocha
%
% Revision history:
%
%  1-MAR-2005: First version, W. Li and H. Rocha
%
%
% This program is free software; you can redistribute it and/or
% modify it under the terms of the GNU General Public License as
% published by the Free Software Foundation; A copy of the GNU
% General Public License can be obtained from the Free Software
% Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

function NLSQFUN = nlsqfun(x,B,f,N,k,n)

%
% function NLSQFUN = nlsqfun(x,B,f,N,k,n)
%
%
% Input Arguments:
%   x            - parameters values
%   B            - a matrix with the data points
%   f            - the actual wing weight of each data point
%   N            - number of data points
%   k            - index of g_i (g_0 == g)
%   n            - number of variables
%

    NLSQFUN=zeros(N,1);
    for j=1:N
        if k==0
            tmp=x(n+1);
        else
            tmp=x(n);
```

```
        end
        for i=1:n
            if i ~= k
                if i>k & k>0
                    id=i-1;
                else
                    id=i;
                end
                tmp=tmp*B(i,j)^x(id);
            end
        end
        NLSQFUN(j)=f(j)-tmp;
    end
```

A.2 PCA CODES

A.2.1 PCA automatic procedure

```
% Wing Weight Estimation for Matlab
% Copyright (C) 2005 W. Li and H. Rocha
%
% Revision history:
%
%  1-MAR-2005: First version, W. Li and H. Rocha
%
%
% This program is free software; you can redistribute it and/or
% modify it under the terms of the GNU General Public License as
% published by the Free Software Foundation; A copy of the GNU
% General Public License can be obtained from the Free Software
% Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.


% PCR - Principal Component Regression
```

```
% The Automatic Principal Component Analysis algorithm is as
% follows:
%
%    1. Scale each variable by its estimated standard deviation.
%    2. Compute the covariance matrix C of the scaled input vectors
%       and remove all the variables corresponding to 0 eigenvalue.
%    3. Let the number of the remaining variables be n_org. For
%       n=n_org downto n = nmin, use RBF or Kriging interpolation,
%       to compute the corresponding n_org-nmin principal component
%       regression approximations.
%
%
% The function CV_error.m is used to compute the
% cross-validation error.
%

clear all;
nmin=3;            % the smallest dimension of feature space
approx_option=1;   % 1 for using c_r+c_t=gamma*s/b before PCR
approx_model=5;    % 1 for Multiquadrics, 2 for Thin Plate Splines,
                   % 3 for Cubic, 4 for Gaussian, and 5 for Kriging
var_number=14;     % number of variables used (either 8 or 14)
rawdata=csvread('wing_data_10.csv');
% the actual input variable names
ref_area=rawdata(1,:);
aspect_ratio=rawdata(2,:);
sweep=rawdata(3,:);
root_chord=rawdata(4,:);
tip_chord=rawdata(5,:);
root_thick=rawdata(6,:);
tip_thick_to_chord=rawdata(7,:);
ultimate=rawdata(8,:);
gross_weight=rawdata(9,:);
wing_weight=rawdata(10,:);
```

```
% the derived variable names
span=sqrt(ref_area.*aspect_ratio);
taper_ratio=tip_chord./root_chord;
root_thick_to_chord=root_thick./root_chord;
tip_thick=tip_thick_to_chord.*tip_chord;
mean_chord=ref_area./span;
% modify the data if c_r+c_t=gamma*s/b is used before PCR analysis
if approx_option==1
   gamma=mean((root_chord+tip_chord)./mean_chord);
   root_chord=gamma*ref_area./(span.*(1+taper_ratio));
   tip_chord=root_chord.*taper_ratio;
   tip_thick=tip_thick_to_chord.*tip_chord;
   root_thick=root_thick_to_chord.*root_chord;
end
% define f and B for PCR
f=wing_weight';
if var_number==8
  n=8;
  N=41;
  B(1,:)=span;
  B(2,:)=root_chord;
  B(3,:)=ref_area;
  B(4,:)=root_thick_to_chord;
  B(5,:)=gross_weight;
  B(6,:)=taper_ratio;
  B(7,:)=sweep;
  B(8,:)=ultimate;
else
  n=14;
  N=41;
  B(1,:)=aspect_ratio;
  B(2,:)=span;
  B(3,:)=mean_chord;
  B(4,:)=root_chord;
```

```
  B(5,:)=tip_chord;
  B(6,:)=ref_area;
  B(7,:)=root_thick;
  B(8,:)=tip_thick;
  B(9,:)=root_thick_to_chord;
  B(10,:)=tip_thick_to_chord;
  B(11,:)=gross_weight;
  B(12,:)=taper_ratio;
  B(13,:)=sweep;
  B(14,:)=ultimate;
end

%scaling
B2=B;
for i=1:n
    B2(i,:)=(B2(i,:)-mean(B2(i,:)))/std(B2(i,:));
end

%coordinate change
C=cov(B2');
[V,D]=eig(C);
fp=fopen(sprintf('CV_%d_%d_%d.txt',approx_option,...
approx_model,var_number),'w');
fprintf(fp,'Eigenvalues of Covariance Matrix:\n');
for i=1:n
   fprintf(fp,'%.5f    ',D(i,i));
   if (mod(i,5)==0)
      fprintf(fp,'\n');
   end
end
if (mod(n,5)~=0)
   fprintf(fp,'\n');
end
```

```
n_org=n;
options=optimset('Display','off','TolFun',1.0e-5);
% Data fitting in the input space
fprintf(fp,'Num_Variable   Initial_Objective   Final_Objective\n');

B1=B2;
prms=ones(n,1);
err0=CV_error(prms,n,N,f,B1,approx_model);
[params,emin]=fminsearch(@CV_error,prms,options,n,...
N,f,B1,approx_model);
if n<10
 fprintf(fp,'  %d (orig space)   %.4e          %.4e\n',n,err0,emin);
else
 fprintf(fp,' %d (orig space)   %.4e          %.4e\n',n,err0,emin);
end
save(sprintf('CV_%d_%d_%d_%d.mat',approx_option,approx_model,...
var_number,0),'f','B','params');

% transform data into the feature space:
% each column of B1 is a data point in the feature space
prms=ones(n,1);
for n=n_org:-1:nmin
  T=V(:,n_org-n+1:n_org)'; % transformation matrix
  B1=T*B2;
  %Cross-validation minimization process
  if n<n_org
     prms=prms(2:n+1);
  end
  err0=CV_error(prms,n,N,f,B1,approx_model);
  [params,emin]=fminsearch(@CV_error,prms,options,n,...
  N,f,B1,approx_model);
  if n<10
    fprintf(fp,'       %d           %.4e          %.4e\n',...
    n,err0,emin);
```

```
  else
    fprintf(fp,'      %d          %.4e          %.4e\n',...
    n,err0,emin);
  end
  save(sprintf('CV_%d_%d_%d_%d.mat',approx_option,...
  approx_model,var_number,n),'f','B','params','B1','T');
end
fclose(fp);
```

A.2.2 CV error function

```
% Wing Weight Estimation for Matlab
% Copyright (C) 2005 W. Li and H. Rocha
%
% Revision history:
%
%  1-MAR-2005: First version, W. Li and H. Rocha
%
%
% This program is free software; you can redistribute it and/or
% modify it under the terms of the GNU General Public License as
% published by the Free Software Foundation; A copy of the GNU
% General Public License can be obtained from the Free Software
% Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

function Ecv = CV_error(params,n,N,f,B,approx_model)

%
% function Ecv=CV_error(params,n,N,f,B,approx_model)
%
% Output Arguments:
%   Ecv          - the cross-validation error value
%
% Input Arguments:
%   params       - a vector with the scaling parameters
```

```
%   n             - number of variables
%   N             - number of data points
%   f             - the actual wing weight of each data point
%   B             - a matrix with the data points in the feature
%                   space coordinates
%   approx_model - regression model to use: 1 for Multiquadrics,
%                   2 for Thin Plate Splines, 3 for Cubic,
%                   4 for Gaussian, and 5 for Kriging
%

    %Iteration matrix
    phi1=zeros(N,N);
    for i=1:N
        for j=1:N
            sub=0;
            for k=1:n
                sub = sub + params(k)*(B(k,i)-B(k,j))^2;
            end
            if approx_model==1
                phi1(i,j)=sqrt(sub+1);
            elseif approx_model==2
                if sub < 1e-10
                    sub=sub+1e-10;
                end
                phi1(i,j)=sub*log(sqrt(sub));
            elseif approx_model==3
                phi1(i,j)=(sqrt(sub))^3;
            else phi1(i,j)=exp(-sub);
            end
        end
    end

    %Main process
    Ecv=0;
```

```
for i=1:N
    fi1=zeros(N,1);
    for j=1:N
        sub=0;
        for k=1:n
            sub = sub + params(k)*(B(k,i)-B(k,j))^2;
        end
        if approx_model==1
            fi1(j)=sqrt(sub+1);
        elseif approx_model==2
            if sub < 1e-10
                sub=sub+1e-10;
            end
            fi1(j)=sub*log(sqrt(sub));
        elseif approx_model==3
            fi1(j)=(sqrt(sub))^3;
        else fi1(j)=exp(-sub);
        end
    end
    f1=zeros(N-1,1);
    fi=zeros(N-1,1);
    phi=zeros(N-1,N-1);
    for l=1:i-1
        f1(l)=f(l);
        fi(l)=fi1(l);
        for q=1:i-1
            phi(l,q)=phi1(l,q);
        end
    end
    for l=1:i-1
        for q=i+1:N
            phi(l,q-1)=phi1(l,q);
        end
    end
```

```
        for l=i+1:N
            fi(l-1)=fi1(l);
            f1(l-1)=f(l);
            for q=i+1:N-1
                phi(l-1,q-1)=phi1(l,q);
            end
        end
        for l=i+1:N
            for q=1:i-1
                phi(l-1,q)=phi1(l,q);
            end
        end
        if rcond(phi) < 10^(-6)
            phi=pinv(phi);
        else
            phi=inv(phi);
        end
        if approx_model==5
            ID=ones(N-1,1);
            gi=fi+((1-ID'*phi*fi)/(ID'*phi*ID))*ID;
            g=(phi*f1)'*gi;
        else
            g=(phi*f1)'*fi;
        end
        Ecv=Ecv+(f(i)-g)^2;
    end
    Ecv=Ecv/N;
```

A.2.3 Basis function evaluation

```
% Wing Weight Estimation for Matlab
% Copyright (C) 2005 W. Li and H. Rocha
%
% Revision history:
```

```
%
%  1-MAR-2005: First version, W. Li and H. Rocha
%
%
% This program is free software; you can redistribute it and/or
% modify it under the terms of the GNU General Public License as
% published by the Free Software Foundation; A copy of the GNU
% General Public License can be obtained from the Free Software
% Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

function fval = basisfunction(x,params,N,n,f,B1,approx_model)

%
% function fval=basisfunction(x,params,N,n,f,B,approx_model)
%
% Output Arguments:
%   fval          - the function value
%
% Input Arguments:
%   x             - point to compute the function value
%   params        - a vector with the scaling parameters
%   n             - number of variables
%   N             - number of data points
%   f             - the actual wing weight of each data point
%   B             - a matrix with the data points in the feature
%                   space coordinates
%   approx_model - regression model to use: 1 for Multiquadrics,
%                   2 for Thin Plate Splines, 3 for Cubic,
%                   4 for Gaussian, and 5 for Kriging
%

    %Iteration matrix
    phi=zeros(N,N);
    for i=1:N
```

```
    for j=1:N
        sub=0;
        for k=1:n
            sub = sub + params(k)*(B1(k,i)-B1(k,j))^2;
        end
        if approx_model==1
            phi(i,j)=sqrt(sub+1);
        elseif approx_model==2
            if sub < 1e-10
                sub=sub+1e-10;
            end
            phi(i,j)=sub*log(sqrt(sub));
        elseif approx_model==3
            phi(i,j)=(sqrt(sub))^3;
        else phi(i,j)=exp(-sub);
        end
    end
end
if rcond(phi) < 10^(-6)
    phi=pinv(phi);
else
    phi=inv(phi);
end

%main process
fi=zeros(N,1);
for j=1:N
    sub=0;
    for k=1:n
        sub = sub + params(k)*(x(k)-B1(k,j))^2;
    end
    if approx_model==1
        fi(j)=sqrt(sub+1);
    elseif approx_model==2
```

```
            if sub < 1e-10
                sub=sub+1e-10;
            end
            fi(j)=sub*log(sqrt(sub));
        elseif approx_model==3
            fi(j)=(sqrt(sub))^3;
        else fi(j)=exp(-sub);
        end
    end
    if approx_model==5
        ID=ones(N,1);
        gi=fi+((1-ID'*phi*fi)/(ID'*phi*ID))*ID;
        fval=(phi*f)'*gi;
    else
        fval=(phi*f)'*fi;
    end
```

VITA

Humberto Rocha
Department of Computational And Applied Mathematics
Old Dominion University
Norfolk, VA 23529

- Education
  - Degree in Mathematics, University of Coimbra, Portugal, 1998.
  - Master in Applied Mathematics, University of Coimbra, Portugal, 2001.
- Presentations
  - Pattern search methods for molecular geometry problems, 17th International Symposium on Mathematical Programming, Atlanta, EUA, August of 2000.
  - Pattern search methods for user-provided points: Application to molecular geometry problems, OPTIMIZATION 2001, Aveiro, Portugal, July of 2001.
  - Data Fitting Methods for Construction of Wing Weight Estimation Models, International Conference on the Interactions between Wavelets and Splines, Athens, USA, May of 2005.
- Publications
  - **H. Rocha**, Pattern Search Methods for Molecular Geometry Problems, Master Thesis, University of Coimbra, 2001.
  - P. Alberto, F. Nogueira, **H. Rocha**, and L. N. Vicente, Pattern search methods for user-provided points, Proceedings of The 2001 International Conference on Computational Science, San Francisco, California, USA, Lecture Notes in Computer Science, Springer-Verlag, 2001.
  - P. Alberto, F. Nogueira, **H. Rocha**, and L. N. Vicente, Pattern search methods for user-provided points: Application to molecular geometry problems, SIAM Journal on Optimization, 14 (1216-1236) 2004.

Typeset using LaTeX.