Optimization Using Surrogate Models

— by the Space Mapping Technique

Jacob Søndergaard

Ph.D. Thesis
Informatics and Mathematical Modelling
Technical University of Denmark
Kgs. Lyngby 2003

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

IMM-PHD-2003-111
ISSN 0909-3192

Preface

This dissertation is submitted to Informatics and Mathematical Modelling at the Technical University of Denmark in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

The work has been supervised by Professor, dr.techn. Kaj Madsen, Associate Professor Hans Bruun Nielsen and Poul Erik Frandsen.

Kgs. Lyngby, January 31, 2003.

Jacob Søndergaard

Abstract

Surrogate modelling and optimization techniques are intended for engineering design in the case where an expensive physical model is involved.

This thesis provides a literature overview of the field of surrogate modelling and optimization. The space mapping technique is one such method for constructing and optimizing a surrogate model based on a cheap physical model. The space mapping surrogate is the cheap model composed with a parameter mapping, the so-called space mapping, connecting similar responses of the cheap and the expensive model.

The thesis presents a theoretical study of the space mapping technique. Theoretical results are derived which characterize the space mapping under some ideal conditions. If these conditions are met, the solutions provided by the original space mapping technique are minimizers of the expensive model. However, in practice we cannot expect that these ideal conditions are satisfied. So hybrid methods, combining the space mapping technique with classical optimization methods, should be used if convergence to high accuracy is wanted.

Approximation abilities of the space mapping surrogate are compared with those of a Taylor model of the expensive model. The space mapping surrogate has a lower approximation error for long steps. For short steps, however, the Taylor model of the expensive model is best, due to exact interpolation at the model origin.

Five algorithms for space mapping optimization are presented and the numerical performance is evaluated. Three of the algorithms are hybrid algorithms. Convergence of a class of hybrid space mapping algorithms is proved.

Resumé

Surrogat modellerings- og optimeringsteknikker er rettet mod "ingeniør design" i det tilfælde hvor en meget dyr fysisk model er involveret.

Afhandling indeholder et litteraturstudie omhandlende surrogat modellerings- og optimeringsteknikker. Space mapping teknikken er en sådan metode til optimering af en surrogat model, som er baseret på en billig fysisk model. Space mapping surrogatet består af den billige model sammensat med en parameter afbildning, den såkaldte space mapping, der forbinder samme respons fra den billige med den dyre model.

Afhandlingen beskriver teoretiske undersøgelser af space mapping teknikken. Der udledes teoretiske resultater, som karakteriserer space mappingen under nogle ideelle betingelser. Såfremt disse betingelser er opfyldt, vil de løsninger som space mapping teknikken finder, være løsninger til minimeringsproblemet for den dyre model. Det kan dog ikke forventes, at disse ideelle betingelser er opfyldt i praksis. Derfor bør hybrid metoder, som kombinerer space mapping teknikken med klassiske optimeringsmetoder, anvendes hvis konvergens til høj nøjagtighed ønskes.

Approksimationsegenskaberne af space mapping surrogatet sammenlignes med en Taylor model af den dyre model. Space mapping surrogatet har en lavere approksimationsfejl for store skridt. For korte skridt derimod, er Taylor modellen af den dyre model bedst, hvilket skyldes eksakt interpolation i udviklingspunktet.

Fem algoritmer for space mapping optimering præsenteres og deres numeriske egenskaber er afprøvet. Tre af algoritmerne er hybrid algoritmer. Konvergens af en klasse af space mapping hybrid algoritmer bevises.

Acknowledgments

I wish to thank my main supervisor Professor, dr.techn. Kaj Madsen for giving me the opportunity to engage in this study. Thank you for the encouragement and motivation that carried me through the study. I also thank my supervisors Associate Professor Hans Bruun Nielsen and Poul Erik Frandsen, Ticra Engineering Consultants, for their guidance. Special thanks to Hans who helped me resolve many mathematical puzzles and always has been available for exchange of ideas. On the same note I also wish to thank Ph.D. student Frank Pedersen for many rewarding discussions.

I want to thank for the hospitality shown to me by Professor John E. Dennis and his Ph.D. student Mark Abramson during my stay at Rice University. Special thanks to John who inspired me with his eager passion for unveiling mathematical questions in a constant search for the truth.

I also want to thank for the hospitality shown to me by Professor John W. Bandler and his Ph.D. students Mohamed Bakr, José Rayas-Sanchez, Mostafa Ismail during my stays at McMaster University. Special thanks to John for sharing his many captivating ideas and his devotion to the space mapping technique. Also special thanks to Mohamed for his friendliness and his involvement in this study.

I would like to thank my colleagues at IMM for making the last three years a pleasant experience.

Finally, I sincerely thank my dear girlfriend Janna and the rest of my family and friends for their constant support.

Contents

Chapter 1

Introduction

In engineering design it is often encountered that traditional optimization is not feasible because the model under investigation is too expensive to compute. Surrogate modelling techniques have been developed to address this important issue. Surrogate models are intended to take the place of the expensive model for the purpose of modelling or optimization of the latter. In optimization using surrogate models, a sequence of subproblems is solved in the search for the optimizer of the expensive model. In the optimization process, most of the model evaluations are performed with the surrogate model. The expensive model is only scarcely evaluated in order to re-calibrate the surrogate model.

The space mapping technique is one such method for constructing and optimizing a surrogate model. The technique relies on the existence of a cheaper model, modelling the same system as the expensive model under investigation. A space mapping surrogate model is the cheaper model composed with a parameter mapping, the so-called space mapping. The space mapping connects similar responses of the cheaper model and the expensive model. Here, responses are the output returned from a model provided a given set of parameters and state variables, which e.g. is a set of sample points in the frequency or time domain.

The basic formulation of the space mapping technique is not convergent, in the sense that in general it does not converge to an optimizer of the expensive

model. Therefore, provably convergent hybrid methods have been developed, combining the space mapping method with a classical optimization method.

This thesis concerns optimization of expensive functions using surrogate models. The focal point of the study is the space mapping technique. The thesis address five main areas: First, the thesis presents a literature overview of surrogate modelling and optimization. Second, the thesis provides a motivation and introduction to the space mapping technique. Third, the thesis provides a theoretical study of the space mapping technique. Fourth, the thesis presents space mapping optimization algorithms and numerical tests of these algorithms. Fifth, the thesis presents a convergence proof for a class of hybrid space mapping algorithms.

The five areas mentioned are covered in separate chapters of the thesis, as described in the following outline.

1.1 Outline of the Thesis

This thesis is divided into six chapters. Each chapter is intended to be self-contained, though the Chapters 4, 5 and 6 are easier conceivable, if the reader is familiar with Chapter 3.

Chapter 2 contains a literature overview of surrogate modelling and optimization. For a review specifically of space mapping methods, refer to [1, 3], two papers co-authored by this author.

Chapter 3 is an included paper [2], introducing and motivating space mapping methodology to the engineering and mathematical communities.

Chapter 4 treats theoretical aspects of space mapping.

Chapter 5 considers formulation of space mapping optimization algorithms and the numerical performance of these.

Chapter 6 is an included paper [4], formulating and proving convergence of hybrid space mapping algorithms.

Chapter 7 is a summary of the conclusions of the study.

Appendix A serves as a manual for a Matlab toolbox with space mapping optimization routines and test problems, developed as a part of the study.

Each chapter contains a separate list of references, a complete list of references is provided before the appendix. We remark that the notation is not uniform throughout the thesis, lists of symbols are provided for the Chapters 2, 4 and 5 right before the list of references in these chapters.

References

[1] M.H. Bakr, J.W. Bandler, K. Madsen, J. Søndergaard, *Review of the Space Mapping Approach to Engineering Optimization and Modelling*, Optimization and Engineering, vol. 1, no. 3, pp. 241–276, 2000.

[2] M.H. Bakr, J.W. Bandler, K. Madsen, J. Søndergaard, *An Introduction to the Space Mapping Technique*, Optimization and Engineering, vol. 2, no. 4, pp. 369–384, 2001.

[3] J.W. Bandler, Q. Cheng, S. Dakroury, A.S. Mohamed, M.H. Bakr, K. Madsen, J. Søndergaard, *Space Mapping: The State of the Art*, submitted, IEEE Trans. Microwave Theory Tech., 2004.

[4] K. Madsen, J. Søndergaard, *Convergence of a Hybrid Space Mapping Algorithm*, submitted, Optimization an Engineering, 2003.

Chapter 2

Surrogate Modelling & Optimization

This chapter provides an overview of literature concerned with surrogate modelling and optimization. The chapter is divided into four sections. Section 2.1 is an introduction to the terminology and the general aspects of the field. Section 2.2 concerns the so-called functional models which are generic, and non-specific to a given problem. Section 2.3 concerns the so-called physical models which are specific to a given problem. The last section, Section 2.4, is a presentation of algorithms for optimization using surrogate models.

2.1 Introduction to Surrogate Models

In the context of engineering modelling and optimization, a surrogate model is a mathematical or physical model which can take the place of an expensive model for the purpose of modelling or optimization of the latter. The expensive model may arise from numerical solution of large systems of e.g. integral or differential equations describing a physical system, or it could be an actual physical system. The surrogate model may be a simplification physically or numerically of the expensive model; or it could be a purely empirical construct, based on information obtained from sparse sampling of the expensive model.

In other words, surrogates are cheap approximations which are especially well suited for acting in place of expensive models in connection with modelling and optimization. A surrogate model is often so cheap that many repeated runs of the model may be conducted within the expense of running the expensive model once. In some cases the surrogate even provides a mathematically more tractable formulation, e.g. in the context of optimization, than that of the expensive model. Often a surrogate model is less accurate than the expensive model, as there tends to be a duality between expensive and cheaper models. Expensive models being of higher fidelity and cheaper model being of lower fidelity.

Surrogates are widely used for modelling and optimization in many engineering fields. In surrogate modelling the engineer can use the surrogate in extensive analysis of different configurations of the model parameters, and thereby gain insight into the workings of the physical system at low expense.

Similar to classical Taylor based optimization, the search for an optimizer in surrogate based optimization is most often conducted by an iterative method relying on sequentially generated surrogate models. In each iteration, the iterative method performs a search on a surrogate model only occasionally referencing the expensive model for validation and correction of the surrogate. A classical Taylor based method requires frequent validation and correction, and thus is too expensive for the problems that surrogate based optimization is intended for.

In the following we survey methods for generating surrogate models and methods for optimization using surrogates. We start by examining the terminology of surrogate modelling and optimization, thereafter we define a set of terms and a problem formulation, from which we will develop the presentation of the methods.

2.1.1 Surrogate Models in the Literature

Surrogate models and optimization methods using surrogate models are an active area of research. For example, in the combined field of mathematicians and engineers several conferences, schools and journal issues have been devoted to the subject in the last years, see e.g. [2, 9, 21, 43, 58, 59].

Unfortunately the wide adoption of surrogate based methods in the engineering community has lead to an ambiguous terminology in the literature. For example a frequently used term related to surrogate modelling is *metamodel*,

in fact metamodel is often considered a synonym for a surrogate model, see e.g. [32].

In case of surrogates for expensive *computer models*, i.e. numerical models, Myers and Montgomery [42] uses the term metamodel about "a somewhat simplified mathematical approximation of the function calculated by design analysis". In other words, the expensive computer model is replaced by a cheaper, possibly simpler, "model of the model", hence a metamodel.

Metamodels are characterized by the algebraic or empirical formulation, constructed independently of the underlying physics of the system, thus some form of calibration is needed in order to use them as surrogates.

Metamodels are commonly used as synonym for the popular *response surface models*. A response surface model is a result of a linear or nonlinear regression of (usually simple) algebraic functions with data. For this reason, they belong to a broader class of models called *regression models*. Response surface models are based on sampling in a chosen set of experimental design points, a process called *design of experiments* is used for choosing these points.

For the purpose of optimization, surrogates are often managed in a *model management framework*, see e.g. [1, 12, 24, 27, 33, 42, 55, 63]. A model management framework enforce conditions on the models such as adaption to the expensive model in a local region or in a more global sense. The most widely adopted of these frameworks is the *Response Surface Methodology* (RSM), see e.g. [42], which is a framework for optimization using sequentially generated response surface models.

Only few of the frameworks referenced above have a proven convergence property, and according to [1], some of these frameworks actually focus on convergence to the problem defined by the surrogate model, rather than the original problem.

Another term used in context of surrogate models is *variable-complexity modelling*, see e.g. [15]. This term covers cases where both the inexpensive model and the expensive model are based on the underlying physics of the system under consideration. The terms *variable physics* and *multi-fidelity physics* are then used to denote that within this system there exist a range of possible physical models.

The mathematical field of constructing approximations to expensive functions has been actively researched for several decades. So, other authors have presented reviews and other retrospective contributions, in which they have partitioned the field in different ways.

The review paper on "approximation concepts for optimum structural design" by Barthelemy and Haftka [10] groups approximations to expensive models into three categories: local, medium-range and global approximations, each of these subdivided again into functional and problem approximation methods.

Barthelemy and Haftka use the term *functional approximation* in cases where an alternative, explicit expression is sought for the objective function and/or the constraints of the problem. Further, they use the term *problem approximation* in cases where the focus is on replacing the original statement of the problem by one which is approximately equivalent but which is easier to solve.

In his thesis Serafini [50] suggested to divide engineering models into two classes: physical and functional models. Where *physical models* are based on numerical solution of governing equations of physical systems and embodies knowledge of the physical system at all points; and where the *functional models* are algebraic approximations of the solutions of the equations constructed without resort to knowledge of the physical systems. Hence, functional models are purely mathematical constructs and they embody knowledge of the behaviour of the function it is approximating, only at the points for which function values are given.

Serafini points out that his distinction between physical and functional models is not absolute. Specifically he uses Taylor models, i.e. truncated Taylor series, as example of hybrids between physical and functional models, since they are purely a mathematical construct, but at the same time they describe the same physical system as the governing equations.

In our presentation we will adopt Serafini's classification into *physical* and *functional* models. However, we suggest to interpret the terms a bit differently. In contrast to Serafini, we have no qualms in classifying Taylor models as "pure" functional models, as they may be constructed without knowledge of the actual governing equations, e.g. by finite difference approximations. Even in cases where the user supplies gradient or higher derivative information, Taylor models are still considered functional models, as the models themselves can be constructed independently of the underlying physics. Thereby we discern between the act of deriving a model and that of assigning parameter values to a model.

On a side note, other references, see e.g. [13, 42], refer to physical models as *mechanistic models*.

Torczon and Trosset [53] distinguish between *surrogate models* and *surrogate approximations*. Where a surrogate model is an auxiliary simulation that is less

physically faithful, but also less computationally expensive, than the expensive model it takes the place of. Further, a surrogate model exists independently of the expensive simulation, and can provide new information about the physical system of interest without requiring additional runs of the expensive model. On the other hand, surrogate approximations are algebraic summaries obtained from previous runs of the expensive model. These approximations are typically inexpensive to evaluate; they could e.g. use radial basis functions, kriging, neural networks, (low-order) polynomials, wavelets and splines. We will review the mentioned approximation methods later on in this presentation.

We will not adopt Torczon and Trosset's distinction between models and approximations, as these terms to a great extent are used as synonyms in most other references. In fact, we will interchangeably use both terms for the same meaning.

Recalling the variable-complexity concept presented above, we observe how physical models may be available in varying fidelity. In cases where several physical models, one being more expensive than another, are used in an optimization process the term *multi-fidelity models* is sometimes used.

Below we define the classification and the problem description we will use in this presentation of methods for surrogate modelling and optimization.

2.1.2 Our Approach

In our presentation we will distinguish between methods that are used to generate surrogate models, and methods (sometimes called model management frameworks) that search for the optimum using surrogate models.

We classify surrogate models in the two categories: *functional models* and *physical models*, defined as follows.

Functional models are models constructed without any particular knowledge of the physical system or governing equations. They are based on algebraic expressions and empirical data; in optimization context this data arise from the current iterate and possibly some points either visited before in the process or found by sampling the parameter space. Hence, functional models exist only in the context of sampled data obtained from a physical model.

Functional models are generic, and therefore applicable to a wide class of problems. Some functional models are also *interpolating approximations*, with

regard to that they — under certain conditions, and given enough data points — eventually will interpolate the underlying model of the data points in some sense. In practice functional models are often very cheap to evaluate.

The methods we consider, which can generate surrogates based on functional models are: radial basis functions, kriging, neural networks, (low-order) polynomials, wavelets and splines.

The surrogate optimization methods we consider, which can employ functional models as surrogates, are: response surface methodology, trust region approach and pattern search.

Physical models are models based on knowledge about the particular physical system in question. Determining a response from a physical model may e.g. involve numerical solution of differential or integral equations. But in the extreme case, a physical model could be actual measurements of the physical system. Ranges of physical models may exist for the same system, as in the concept of variable- or multi-fidelity physics. Physical models are not generic, as each of them is related to a specific physical system. Hence, reuse of physical models across different problems is rare. In practice physical models are often expensive to evaluate, except in cases where they are based on a significant simplification of the physical system.

The methods we consider, which can generate surrogates based on physical models, are: response correction, multipoint method and space mapping.

The surrogate optimization methods we consider, which can employ physically based surrogates, are: response surface methodology, trust region approach, pattern search and space mapping.

The physical and functional models present extremes. Physical models in the one extreme, where a great deal is known about the system, and functional models in the other extreme, where the only assumption is that the response is locally smooth.

In reality a physical system is not completely determined by governing equations, so the practical physical model may contain some empirical elements e.g. as parameters determined by regression to experimental data. Since such a model is strongly coupled to the underlying physics we would still call it a physical model.

Problem Definition

The optimization problem we consider, in surrogate optimization, involves a real valued function $f : \mathbb{R}^n \mapsto \mathbb{R}$, which represents an expensive model. We call f a response function of the expensive model, it is the objective function to be minimized. We will focus on the case where f stems from a physical model, though most of the optimization methods we consider can deal with the case where f stems from a functional model.

We seek a point $x \in \mathbb{R}^n$ which is a solution to the problem

$$\min_x f(x) \, . \tag{2.1}$$

We will denote an optimizer of this problem by x^*. We assume that f is bounded below and uniform continuously differentiable.

In surrogate optimization f is to be replaced by $s : \mathbb{R}^n \mapsto \mathbb{R}$, a surrogate model, in the search for an optimizer of f. The search for x^* is conducted by a method combining sequentially generated surrogates s_k, $k = 1, 2, ...$, and an iterative scheme that performs a search on s_k to obtain the next iterate. We will denote the iterates by x_k, $k = 1, 2, \ldots$. When we say that a method is convergent we imply that the sequence $\{x_k\}$ converges to x^*, hence $x_k \to x^*$ for $k \to \infty$.

The methods for generating functional models rely on pre-sampled data, in order to construct the model. We will assume that p sample points $t_i \in \mathbb{R}^n, i = 1, \ldots, p$, have been chosen and that f has been evaluated at these points. So, we have a pre-sampled data set (t_i, y_i), $i = 1, \ldots, p$, where $y_i = f(t_i)$. We briefly discuss statistical strategies for placing these sample points in Section 2.2.1.

The methods for generating physically based surrogates rely on the existence of one or more user provided cheap (lower-fidelity) physical models represented by the response functions $c_i : \mathbb{R}^n \mapsto \mathbb{R}$, $i = 1, \ldots, q$. A cheap physical model may sometimes by itself act as a surrogate for the expensive model.

In the parts of this presentation about *space mapping*, namely Section 2.3.3 and 2.4.4, we consider vector valued response functions, e.g. $f : \mathbb{R}^n \mapsto \mathbb{R}^m$, $m \geqslant n$. For this purpose we introduce a convex merit function $H : \mathbb{R}^m \mapsto \mathbb{R}$, usually a norm. So (2.1) becomes

$$\min_x H(f(x)) \, . \tag{2.2}$$

Such vector valued response functions arise e.g. in electrical engineering where signals are measured on a discrete frequency or time axis. Here, often the

design problem is defined as minimizing the residual between the signals and given design specifications in some norm, so that the residues constitute the response functions.

In the following we present some of the most popular approaches to surrogate modelling and optimization, and show how they relate. We aim to keep the exposition clear by using a simple consistent notation throughout. However, by committing to a simple notation we must accept that we at the same time cannot capture the details of the more specialized approaches presented in the literature.

2.2 Surrogates Based on Functional Models

There is a large number of methods for generating surrogates in the category of functional models. So, we have limited this presentation to an overview of the most commonly used approximations for expensive models, namely the regression models, radial functions and single point approximations.

The regression models cover the broadest and most widely used class of approximations. They are based on algebraic expressions, the so-called basis functions, that are fitted to the pre-sampled data. We will in particular deal with polynomials, response surface models (including methods for design of experiments) and wavelets.

The radial functions cover a class of approximations which are based on combinations of basis functions localized around the pre-sampled points. We will in particular deal with kriging (including DACE), radial basis functions, neural networks and a special class of splines.

The single point approximations is a special class of local models, that includes Taylor models (i.e. truncated Taylor series). These approximations are usually not very attractive for approximating expensive functions, due to their local nature. But certain types of approximations in this class have found use in the structural engineering community. We will in particular consider the reciprocal approximations, conservative approximations and posynomial approximations.

2.2.1 Regression Models

Regression is the process of fitting some regression model, represented by the function $s(x, \beta)$, to the pre-sampled data (t_i, y_i), $i = 1, \ldots, p$ of the response

function f from the expensive model. The parameters β solve the regression or data fitting problem

$$\min_{\hat{\beta}} \sum (y_i - s(t_i, \hat{\beta}))^2 \ . \tag{2.3}$$

Here the Euclidean norm is used, but other norms or merits may be used as well. Another widely used formulation, augmenting problem (2.3), is the weighted least-squares problem. For simplicity, we will only consider the unweighted problem.

A number of statistical methods have been developed to help determine if a regression model yields a good fit to the data. These methods include residual analysis, testing for lack of fit, analysis of variance (ANOVA). To overcome problems with lack of fit, statisticians rely on methods like transformation of the variables. See [13, 42] for a thorough treatment of these statistical methods. Further, in [11, 19, 33] practical usage of these ideas are illustrated.

The regression model may take many forms. We start by discussing the linear regression models, that is, the regression models where $s(x, \beta)$ is linear in β. We write $s(x, \beta) = \beta^T v(x)$, where $v : \mathbb{R}^n \mapsto \mathbb{R}^u$, is a vector with u basis functions $v_j(x)$, $j = 1, \ldots, u$.

Response Surface Models

The most frequently used basis functions in linear least-squares regression are low order polynomial regression models. Hence, basis functions of the form

$$v(x) = \begin{cases} 1 \\ x^{(i)} \\ x^{(i)}x^{(j)} \end{cases} \qquad i, j \in \{1, \ldots, n\}$$

yielding approximations like

$$s(x) = \begin{cases} \beta_0 + \beta_1^T x & \text{(i)} \\ \beta_0 + \beta_1^T x + \displaystyle\sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \beta_2^{(i,j)} x^{(i)} x^{(j)} & \text{(ii)} \\ \beta_0 + \beta_1^T x + x^T \beta_2 x & \text{(iii)} \end{cases} \tag{2.4}$$

where $\beta_k \subset \beta$, $k = 0, 1, 2$, The models are called (i) first-order model, (ii) first-order model with interaction terms and (iii) second-order, or quadratic, model.

It is obvious that the models in (2.4) are so simple that they generally will not interpolate the data, especially as it usually required that the regression problem is over-determined. In fact we may only expect these models to describe a global trend in the functional behaviour.

Exceptions, where the simple models may be adequate, are for instance in cases where the sampled response is very noisy, see [26] for an example with a noisy aircraft wing design problem, and in cases where the response is so smooth that even a linear model is a valid approximation for a large region of the design space.

More sophisticated regression models are often used, but choosing a well suited regression model for a particular problem requires specific knowledge about the expected behaviour of the system from which the data originate, or at least extensive analysis of the sampled data. We note that more sophisticated models may enlarge the region of acceptable approximation compared to the simple models presented above. However, a drawback of introducing more sophisticated regression models is that even though such a model may interpolate the given data, it is not necessarily good at describing the behaviour between known data points, or in extrapolating outside the region spanned by the data points. This fact shows, e.g. for higher order polynomial approximations.

In the statistics literature regression models as those in (2.4), in particular the quadratic model (iii), are associated with the term *response surface models*. A very popular optimization framework, employing response surface models, is the *response surface methodology*, which we present in Section 2.4.1.

First we will discuss some statistical strategies on how to choose the pre-sampled points.

Design of Experiments

The functional models we have considered so far are constructed on basis of pre-samples data (t_i, y_i), $i = 1, \ldots, p$, where $t_i \in \mathbb{R}^n$ are called data points or design sites, and $y_i = f(t_i)$ are the responses from the expensive model at the design sites. The process of determining where to place the design sites in the parameter space is in the statistical literature called *Design of Experiments* (DOE).

In design of experiments the effort is on laying out experiments, i.e. the actual placement of the t_i's, in certain optimal ways. D-optimality is a popular measure related to the determinant of the covariance matrix in question, but other measures exist. Frequently used experimental designs are factorial, central composite and Box-Behnken designs (see references below). Traditionally much of the focus in DOE has been on reducing noise in the experiment, by repeated runs with "blocking of factors". However, more and more experiments are conducted using computer models, and DOE for computer experiments need special attention since the computer models

- are deterministic, i.e. always return the same result when evaluated for the same parameters (the numerical noise is assumed to be negligible), so repeated runs are not needed,
- are unaffected in response by the ordering of the experiment, hence "experimental blocking" is not needed,
- often are defined over wide parameter ranges, and often in many parameters.

Regarding the last point, computer experimenters often seek to find a complex approximation extending over a wide range of the design variables, hence large regions need to be covered by the experimental design.

These characteristics call for a special type of experimental design called *space-filling* designs, which aim to exercise all the parameters over their entire ranges. McKay et al. [41] presented an intuitive approach, which has become very popular, called *Latin hypercube sampling* from which stochastic space-filling designs are easily obtained, even when a large number of design sites are required. There also exist deterministic experimental designs with space-filling properties. Experimental designs are covered thoroughly in [13, 30, 42], where [13, 42] have special focus on response surface methodology, the topic of Section 2.4.1. A review of methods for design of "optimal experiments" are given in [28].

Wavelets

Wavelets provide a large number of orthonormal basis functions that can be used in linear least-squares regression. The wavelet basis functions are available at different scales, and each of these basis functions has a localized response in the parameter space. Popular families of wavelet basis functions are

Haar, Daubechies and symmlet. We present here the discrete wavelet formulation in one dimension, $x \in \mathbb{R}$, and the particular simple Haar family of basis functions; our presentation is based on [29] and [61].

First consider the piecewise constant function (called the *father wavelet* or *scaling function*)

$$\hat{v}(x) = \begin{cases} 1 & \text{for } x \in [0,1] \\ 0 & \text{otherwise,} \end{cases}$$

and the functions

$$v_{j,k}(x) = 2^{j/2}\hat{v}(2^j x - k) \ , \quad k, j \in \mathbb{Z}_+ \ .$$

The functions $v_{0,k}$ form an orthonormal[1] basis for functions with jumps at the integers. Let the space spanned by this basis be denoted $\mathcal{V}_0$. Similarly the functions $v_{1,k}$ form an orthonormal basis for a space $\mathcal{V}_1 \supset \mathcal{V}_0$ of functions piecewise constant on intervals of length $\frac{1}{2}$. More generally we have $\cdots \supset \mathcal{V}_2 \supset \mathcal{V}_1 \supset \mathcal{V}_0$, where $\mathcal{V}_j$ is spanned by the 2^j functions $v_{j,k}$, $k = 0, 1, \ldots, 2^j$.

We might represent a function in $\mathcal{V}_{j+1}$ by a component in $\mathcal{V}_j$ plus the component in the orthogonal complement $\mathcal{W}_j$ of $\mathcal{V}_j$ to $\mathcal{V}_{j+1}$, written $\mathcal{V}_{j+1} = \mathcal{V}_j \oplus \mathcal{W}_j$.

From the *mother wavelet*

$$\hat{w}(x) = \hat{v}(2x) - \hat{v}(2x-1)$$

we can generate the functions

$$w_{j,k}(x) = 2^{j/2}\hat{w}(2^j x - k) \ , \quad k, j \in \mathbb{Z}_+ \ .$$

Then it can be shown [61] that the functions $w_{j,k}$ form an orthonormal basis for $\mathcal{W}_j$.

Notice that since these spaces are orthogonal, all the basis functions $v_{j,k}$ and $w_{j,k}$ are orthonormal. Now $\mathcal{V}_{j+1} = \mathcal{V}_j \oplus \mathcal{W}_j = \mathcal{V}_{j-1} \oplus \mathcal{W}_{j-1} \oplus \mathcal{W}_j = \cdots$, so we can make a representation of the form $\mathcal{V}_j = \mathcal{V}_0 \oplus \mathcal{W}_0 \oplus \mathcal{W}_1 \cdots \oplus \mathcal{W}_{j-1}$. Assume that we were to construct an interpolation at 2^j data points, so at most 2^j basis functions are needed for interpolation. We could use the 2^j functions in $\mathcal{V}_j$, or alternatively the $2^j - 1$ functions in $\mathcal{W}_0 \oplus \mathcal{W}_1 \cdots \oplus \mathcal{W}_{j-1}$ and one in $\mathcal{V}_0$.

If a non-interpolating approximation is needed, one can chose to use only a subset of the basis functions, e.g. approximation at level j starting at level i, $i < j$, gives the $2^j - 2^i$ basis functions in $\mathcal{V}_i \oplus \mathcal{W}_i \oplus \mathcal{W}_{i+1} \cdots \oplus \mathcal{W}_{j-1}$.

[1] Orthogonality is determined by the usual inner product.

If we assume that the parameters x of our problem (2.1) is scaled to the interval $[0, 1]$, the resulting wavelet approximation can be written as

$$s(x) = \beta^T \begin{pmatrix} v_{i,0}(x) \\ \vdots \\ v_{i,2^i}(x) \\ w_{i,0}(x) \\ \vdots \\ w_{i,2^i}(x) \\ \vdots \\ w_{j-1,0}(x) \\ \vdots \\ w_{j-1,2^{j-1}}(x) \end{pmatrix},$$

for the level j approximation starting at level i. The regression coefficients β are usually found by least-squares.

A common procedure is to apply a threshold to discard or filter out the smaller coefficients, and thereby reduce the number of basis functions in the final approximation. Choosing a good level of approximation is difficult in general, and out of the scope of this presentation.

We should note that in the context of engineering design, the Haar family of basis functions may not be the best suited, due to their non-differentiable form, hence one of the other more smooth families of wavelets should be chosen instead. The concept of wavelets is thoroughly dealt with in [61], a classical reference on wavelets is [20].

2.2.2 Radial Functions

A particular successful class of interpolation methods are based on a model of the form

$$s(x) = v(x)^T \beta + b(x)^T \gamma \,, \tag{2.5}$$

where v is a vector with basis functions (as above) and β is the solution to a generalized linear least-squares regression problem, which we will introduce later. The second part is a radial model, which consists of the function $b : \mathbb{R}^n \mapsto \mathbb{R}^p$ that is a vector of radial functions

$$b_j(x) = \phi(\|x - t_j\|) \,, \quad j = 1, \ldots, p \,, \tag{2.6}$$

and the coefficients $\gamma \in \mathbb{R}^p$.

A radial function ϕ depends only on the radial distance from the origin, or in our case distance to design points. We will use the notation $\phi_j(x) = \phi(\|x - t_j\|)$, $j = 1, \ldots, p$. Various distance measures are used in practice, but the most popular is the Euclidean norm. The nature of ϕ and γ will be more clear to the reader as we present some radial function approximation methods in the following.

Kriging

The method of *kriging* [40] is very popular in the geo-statistical community. The general approximation is $s(x) = \alpha(x)^T y$, where the function α is derived in the following, and y is the vector with pre-sampled values of f.

Kriging models decompose the function f into two parts. The first part is a linear regression model related to a global trend in the data, as the response surface models in Section 2.2.1. The second part is a function $z(x)$, being the deviation between f and the regression model. Hence the interpolation conditions for the kriging model are

$$\begin{aligned} f(t_i) &= y_i \\ &= s(t_i) \\ &= \alpha(t_i)^T y \\ &= v(t_i)^T \beta + z(t_i)\,, \end{aligned}$$

for $i = 1, \ldots, p$.

In statistical terms, see [49], z is a stochastic function, with mean $\mathrm{E}[z(x)] = 0$ and variance $\mathrm{E}[z(x)^2] = 1$, sampled along a suitable path. We consider z to be a residual function, which we will approximate using a radial model, i.e. the last term in (2.5).

Let V be the matrix where the ith column is the ith basis function v_i evaluated at the design sites, $v_i(t_j)$, $i = 1, \ldots, u$, $j = 1, \ldots, p$. Let Z be the vector containing the residuals at the sample points, $z(t_j)$, $j = 1, \ldots, p$.

For any x, we have

$$\begin{aligned} s(x) - f(x) &= \alpha(x)^T y - f(x) \\ &= \alpha(x)^T (V\beta + Z) - (v(x)^T \beta + z(x)) \\ &= \alpha(x)^T Z - z(x) + (V^T \alpha(x) - v(x))^T \beta\,. \end{aligned}$$

We can now determine the *mean squared error*, which is the expected approximation variance,

$$\begin{aligned} \mathrm{E}[(s(x) - f(x))^2] &= \mathrm{E}[(\alpha(x)^T Z - z(x))^2] \\ &= \mathrm{E}[z(x)^2 + \alpha(x)^T Z Z^T \alpha(x) - 2\alpha(x)^T Z z(x)] \\ &= \sigma^2 \left(1 + \alpha(x)^T C \alpha(x) - 2\alpha(x)^T b(x)\right) . \end{aligned} \tag{2.7}$$

Where σ^2 is the process variance, $C \in R^{p \times p}$ is a symmetric matrix where the (i,j)th element is $\phi(\|t_i - t_j\|)$, $i, j = 1, \ldots, p$. In statistical terms the elements of C describe the covariance between design sites.

Now we determine the function $\alpha(x)$, for fixed x, by the quadratic programming problem, minimizing the expected approximation variance (2.7),

$$\begin{array}{ll} \min\limits_{\alpha(x)} & \frac{1}{2}\alpha(x)^T C \alpha(x) - \alpha(x)^T b(x) \\ \text{s.t.} & V^T \alpha(x) = v(x) \ . \end{array} \tag{2.8}$$

Using this formulation to derive $\alpha(x)$, the approximation is called *kriging with a trend*. When the problem (2.8) is unconstrained, the approximation is called *simple kriging* and the solution is $\alpha(x) = C^{-1} b(x)$. When $v(x) = 1$ for all x, the approximation is called *ordinary kriging*.

The Lagrangian function corresponding to (2.8) is

$$L(\alpha(x), \mu) = \frac{1}{2}\alpha(x)^T C \alpha(x) - \alpha(x)^T b(x) + \mu^T (V^T \alpha(x) - v(x))$$

where μ are the Lagrange multipliers. The necessary conditions for an optimal solution are

$$\begin{aligned} \nabla_{\alpha(x)} L(\alpha(x), \mu) = C\alpha(x) - b(x) + \mu^T V^T &= 0 \\ \nabla_{\mu} L(\alpha(x), \mu) = V^T \alpha(x) - v(x) &= 0 \ . \end{aligned}$$

From these equations we may find the solution by solving a linear system,

$$\begin{bmatrix} C & V \\ V^T & 0 \end{bmatrix} \begin{bmatrix} \alpha(x) \\ \mu \end{bmatrix} = \begin{bmatrix} b(x) \\ v(x) \end{bmatrix} \tag{2.9}$$

$$\Rightarrow \quad \begin{bmatrix} \alpha(x) \\ \mu \end{bmatrix} = \begin{bmatrix} C^{-1}(1 - VUV^T C^{-1}) & C^{-1}VU \\ UV^T C^{-1} & -U \end{bmatrix} \begin{bmatrix} b(x) \\ v(x) \end{bmatrix} \ ,$$

where $U = (V^T C^{-1} V)^{-1}$, and assuming that C is symmetric positive definite. Hence we can write

$$\begin{bmatrix} \alpha(x) \\ \mu \end{bmatrix} = \begin{bmatrix} R & W \\ W^T & -U \end{bmatrix} \begin{bmatrix} b(x) \\ v(x) \end{bmatrix} ,$$

where R and W are defined by the solution above.

The kriging approximation then is

$$\begin{aligned} s(x) &= \alpha(x)^T y \\ &= b(x)^T R y + v(x)^T W^T y \\ &= b(x)^T \gamma + v(x)^T \beta , \end{aligned} \tag{2.10}$$

where

$$\begin{aligned} \beta &= (V^T C^{-T} V)^{-1} V^T C^{-T} y \\ \gamma &= C^{-T}(y - V\beta) \end{aligned}$$

are independent of x. Note that β is the generalized least-squares solution to the linear regression problem. Having this formulation, we need only calculate $v(x)$ and $b(x)$ and the sum of two dot products for every evaluation of $s(x)$.

We could have derived γ and β in another way, namely by considering the problem

$$\begin{aligned} \min_{\gamma} \quad & \tfrac{1}{2}\gamma^T C \gamma - \gamma^T y \\ \text{s.t.} \quad & V^T \gamma = 0 , \end{aligned} \tag{2.11}$$

We should note that we have not been able to motivate the quadratic problem (2.11) in the same way as (2.8), which minimizes the expected approximation variance. The corresponding Lagrangian function to (2.11) is

$$L(\gamma, \beta) = \frac{1}{2}\gamma^T C \gamma - \gamma^T y + \beta^T V^T \gamma$$

where β are the Lagrange multipliers. The necessary conditions for an optimal solution are

$$\begin{aligned} \nabla_\gamma L(\gamma, \beta) = C\gamma - y + \beta^T V^T &= 0 \\ \nabla_\beta L(\gamma, \beta) = V^T \gamma &= 0 . \end{aligned}$$

From these equations we may find the solution by solving a linear system,

$$\begin{bmatrix} C & V \\ V^T & 0 \end{bmatrix} \begin{bmatrix} \gamma \\ \beta \end{bmatrix} = \begin{bmatrix} y \\ 0 \end{bmatrix} \tag{2.12}$$

$$\Rightarrow \quad \begin{bmatrix} \gamma \\ \beta \end{bmatrix} = \begin{bmatrix} R & W \\ W^T & -U \end{bmatrix} \begin{bmatrix} y \\ 0 \end{bmatrix}$$

again assuming that C is symmetric positive definite; R, W and U defined as above. Deriving the formulation in this way is by some called *dual kriging*, the simple case with $V = 0$ is e.g. described in [48].

When v_i is at most first order polynomials, the constraint $V^T\gamma = 0$ in (2.11) corresponds to the requirement that $|s(x)| = \mathcal{O}(|x|)$ [46].

For certain choice of C, the kriging approach relates to approximation using natural cubic splines, [62] shows this relation in one dimension. The relation between the thin-plate spline formulation and kriging is shown in [25], and presented in Section 2.2.2 below.

First we will discuss how statisticians have used the kriging approach in approximation of computer based models.

DACE Design and Analysis of Computer Experiments (DACE), named after a seminal paper by Sacks et al. [49] in 1989, is a statistical framework for dealing with kriging approximations to (complex or expensive) computer models. Kriging, in particular the DACE framework, has gained wide acceptance in many engineering communities, e.g. in mechanical, aerospace and electrical engineering, as a method for approximating expensive functions, see e.g. [32, 33, 49, 50, 57].

In the DACE framework the kriging correlation model is often presented as a radial function of the form

$$\phi_j(x) = \prod_{i=1}^{n} \psi(\theta, |x^{(i)} - t_j^{(i)}|) \; . \tag{2.13}$$

Hence a product of radial functions or, in statistical terms, correlation functions, one for each coordinate direction.

Examples of frequently used kriging correlation functions are [49]

$$\psi(\theta, |x^{(i)} - t_j^{(i)}|) = \begin{cases} \exp(-\theta^{(i)}(x^{(i)} - t_j^{(i)})^2) & \text{(i)} \\ \max\{0,\ 1 - \theta^{(i)}|x^{(i)} - t_j^{(i)}|\} & \text{(ii)} \\ \max\{0,\ 1 - \theta^{(i)}(x^{(i)} - t_j^{(i)})^2 \\ \qquad\qquad + \theta^{(i+n)}|x^{(i)} - t_j^{(i)}|^3\} & \text{(iii)} \end{cases} . \tag{2.14}$$

Each of these are also available in an isotropic version, i.e. where θ is constant for all coordinate directions.

The DACE framework is implemented in a Matlab toolbox, see [36, 37]. This particular implementation takes great care in solving the system (2.9) in a safe numerical way. In many other implementations the matrix C is often naively inverted, and since it is often ill-conditioned, numerical errors are likely to dominate the results. Further, the implementation includes a method for fitting the radial functions to data, i.e. finding θ minimizing a certain merit, namely maximum likelihood estimation, i.e. least-squares when assuming a Gaussian process.

Often, when using maximum likelihood estimation, a Gaussian process is assumed, then it is vital for the approximation to determine basis functions v such that the residuals $y_i - V\beta$ follows the normal distribution, [49, 18, 36, 37] elaborate further on this subject.

From the viewpoint of Bayesian statistics the choice of correlation function corresponds to a Bayesian prior on the "shape" or "smoothness" of the function. In this view Kriging, and thereby DACE, is a Bayesian method, see e.g. [49, 18].

Radial Basis Function Approximations

The radial basis function approximation is as in (2.5), and is thus identical to kriging. However, in the literature there is a difference in the way the functions v and b are chosen. In radial basis function approximations v is a vector of polynomials of at most order n, and b is a vector of radial (basis) functions, using the Euclidean norm as distance measure. The coefficients β and γ are determined by the system (2.12) above.

Examples of commonly used radial basis functions

$$\phi_j(x) = \begin{cases} \|x - t_j\|_2 & \text{(i)} \\ \|x - t_j\|_2^3 & \text{(ii)} \\ \|x - t_j\|_2^2 \, \log \|x - t_j\|_2 & \text{(iii)} \\ \sqrt{\|x - t_j\|_2^2 + \theta^2} & \text{(iv)} \\ \exp(-\theta \|x - t_j\|_2^2) & \text{(v)} \end{cases} \tag{2.15}$$

where θ is a fixed constant, provided by the user. As in the DACE framework mentioned in the previous section, θ may be fitted to the data, provided a suitable merit function. The set of radial functions in (2.15) include both functions which grow with distance and functions which vanish with distance. Unlike the kriging correlation functions in (2.14) which all vanish with distance.

Some of these radial functions are related to certain Green's functions for partial differential equations. Specifically, the partial differential equations $L\phi_j(x) = \delta(x - t_j)$ for the operators $L = \nabla^2$ and $L = \nabla^4$. In Table 2.1 the Green's functions are presented for the one to three dimensional cases. The association with the radial functions in (2.15) is evident. For example the

	$L = \nabla^2$	$L = \nabla^4$
1D	$\lvert x - t_j \rvert$	$\lvert x - t_j \rvert (x - t_j)^2$
2D	$\log \|x - t_j\|_2$	$\|x - t_j\|_2^2 \log \|x - t_j\|_2$
3D	$\|x - t_j\|_2^{-1}$	$\|x - t_j\|_2$

Table 2.1: Green's functions $\phi_j(x)$, solutions to $L\phi_j(x) = \delta(x - t_j)$

Green's function $\phi_j(x) = \|x - t_j\|_2^2 \log \|x - t_j\|_2$ is that spline which solves the minimal surface problem for a thin plate, with a point load at t_j, hence its name *thin-plate spline*. Roach [47] provides a thorough treatment of Green's functions.

Guttmann [27] presents a method for global optimization using radial basis function approximations. Guttmann states that the two main advantages of radial basis function approximations are an available measure of so-called "bumpiness", and that uniqueness of an interpolant is achieved under very mild conditions on the location of the interpolation points. The measure of

bumpiness may be used in a merit function for determining the placement of succeeding design sites.

Note that, choosing the function (v) in (2.15) makes the radial basis function approximation the same as the DACE kriging approximation, with the isotropic version of the correlation function (i) in (2.14). However, the constant θ is provided by the user in the radial basis function case, and often found by maximum likelihood estimation in the DACE case, as described previously.

Powell [46] provides a thorough treatment of radial basis function approximations, and contains many references to other works.

Multivariate Adaptive Regression Splines

Another form of radial basis function approximations is the *multivariate adaptive regression splines*, see e.g. [29]. We will present the approximation here for the one dimensional case, $x \in \mathbb{R}$.

The approximation is almost as in (2.5). However, v is always the zero order polynomial and b is a vector function $b : \mathbb{R} \mapsto \mathbb{R}^q$, and $\gamma \in \mathbb{R}^q$. Usually $q < p$, hence the approximation is not necessarily interpolating the data. The elements of b, $b_i = \psi_i(x)$, $i = 1, \ldots, q$, are radial functions,

$$\psi_i(x) = \prod_{j \in \mathcal{M}_i} \phi_j(x) \,, \quad \mathcal{M}_i \subseteq \{1, \ldots, p\}$$

with

$$\phi_j(x) \in \{|x - t_j|_+, |t_j - x|_+\} \,, \quad j = 1, \ldots, p \,,$$

where $|\cdot|_+ = \max\{\cdot, 0\}$. So, each ψ_i is a linear radial function or a product of two or more such functions.

The reader should note the structural similarity with the DACE kriging approximations, using the radial function (ii) in (2.14). The tricky part is, as for other approximation methods, in constructing the functions ψ_i, $i = 1, \ldots, q$, one strategy is suggested in [29, p. 284].

Neural Networks

Neural networks are *nonlinear* regression models [29]. The most widely used class of neural network is known by the names *feed-forward network*, *single*

hidden layer back-propagation network and *single layer perceptron*. The approximation is

$$s(x) = \sigma(\beta_0 + \beta_1 x)^T \beta_2 + \beta_3$$

where the vector function $\sigma : \mathbb{R}^n \mapsto \mathbb{R}^u$ contains the so-called activation functions. The large number of constants $\beta_0 \in \mathbb{R}^n$, $\beta_1 \in \mathbb{R}^{n \times n}$, $\beta_2 \in \mathbb{R}^u$ and $\beta_3 \in \mathbb{R}$ are determined by nonlinear least-squares regression, as in (2.3), assuming that the data set is sufficiently large.

A popular choice of activation function is the *sigmoid* function

$$\sigma(z) = \frac{1}{1 + e^{-z}} .$$

Other popular choices of activation functions include the radial basis functions (2.15). The latter case is called a *radial basis function neural network*, and is exactly the same approximation as the formulation in (2.5), with v being the zero order polynomial. However, instead of the procedure derived in 2.2.2 for determining the model parameters, the model parameters of the radial basis function neural networks are determined by least-squares regression [31].

For the purpose of determining the β's by regression, i.e. solving (2.3), several well-known optimization algorithms has been reinvented in the neural network community. One of the first methods to re-appear, and probably still the most widely used, is the steepest-descent algorithm, which neural network advocates have named *back-propagation learning*. Of course, a range of standard optimization algorithms, see [23], can be used to solve the regression problem (2.3), and steepest-descent is not the most obvious in that respect.

There is no reason to believe that neural networks should be able to provide better approximations than other methods mentioned in this work. In fact, we should stress that the frequently used viewpoint of neural networks as convenient, magic, black-box approximations, may mislead the user into overlooking the possibility of posing ill-conditioned problems — where the number of parameters to be determined by regression (the β's) is larger than the provided data set — not to mention a related problem namely serious risk of over-fitting.

2.2.3 Single Point Models

We will now discuss a class of models that is based on information obtained from a single point, usually the current iterate x_k in an optimization proce-

dure. Taylor models, i.e. truncated or approximate Taylor series, are prominent members of this class. However, the low-order Taylor models, that are feasible to construct in practice, are only valid as approximations in a small region around x_k.

In the following we consider some alternative models, also based on information from the current iterate only. Under certain conditions they have a larger region of validity, than Taylor based models, which make them tractable as surrogates for expensive models.

Reciprocal and conservative approximations

Consider the approximation

$$s_k(x) = f(x_k) + \sum_{i=1}^{n} \nabla_{x^{(i)}} f(x_k)(x^{(i)} - x_k^{(i)})\phi_i(x^{(i)}, x_k^{(i)})$$

where $x = (x^{(1)}, \ldots, x^{(n)})^T$. We see that enforcing the first order requirement $s_k'(x_k) = f'(x_k)$ it follows that $\phi_i(x_k^{(i)}, x_k^{(i)}) = 1$. The choice $\phi_i(x^{(i)}, x_k^{(i)}) = 1$ yields the (linear) first order Taylor series approximation.

In structural optimization the alternative form $\phi(x^{(i)}, x_k^{(i)}) = x_k^{(i)}/x^{(i)}$ is often used, the approximation is then called a *reciprocal approximation* [10]. A significant class of constraints in structural engineering can in this way be transformed from nonlinear to linear equations (at the expense of introducing nonlinearity into the objective function). As the reciprocal approximation may become unbounded if any of the variables approach zero. An alternative is the *modified reciprocal approximation*, $\phi(x^{(i)}, x_k^{(i)}) = (x_k^{(i)} + c^{(i)})/(x^{(i)} + c^{(i)})$, where the values of $c^{(i)}$'s are typically small compared to representative values of the corresponding $x^{(i)}$'s. Another alternative is the *conservative approximation*, having

$$\phi(x^{(i)}, x_k^{(i)}) = \begin{cases} 1 & \text{if } x_k^{(i)} \nabla_{x^{(i)}} f(x_k) \geqslant 0 \\ x_k^{(i)}/x^{(i)} & \text{otherwise.} \end{cases} \tag{2.16}$$

Following [1] the conservative approximation has the attractive feature of leading to a convex programming problem and thus is amenable to solution by nonlinear programming techniques that take advantage of the dual problem, in [10] there are further references on this subject.

We note that the reciprocal and conservative approximations destroy the linearity of the approximation, and thus the possibility of directly use it with a sequential linear programming algorithm.

Posynomial approximation

The *posynomial approximation* of the form

$$s_k(x) = f(x_k) \prod_{i=1}^{n} \left(\frac{x^{(i)}}{x_k^{(i)}} \right)^{\alpha^{(i)}} ,$$

where

$$\alpha = \frac{f'(x_k)}{f(x_k)} ,$$

can be treated using geometric programming techniques, which actually requires this form. According to [1], geometric programming techniques will, under appropriate conditions and when applied to a posynomial approximation of the original problem, converge to a stationary point of the original problem (2.1).

2.2.4 Summary

In the last sections we have reviewed methods for generating functional models that can be used as surrogate models for expensive functions. We have presented the most commonly used of these methods, namely regression models, radial functions and single point models.

The regression models consist of regression functions and parameters. The parameters are found by fitting the regression functions to pre-sampled data using least-squares. The particular methods we have discussed are the response surface models, wavelets and neural networks. The problem of positioning the pre-sampled data has only briefly been covered in this presentation. In fact, it is a discipline in itself, often referred to in the literature as design of experiments.

The radial functions are used in a class of models called kriging models or radial basis function approximations. Here, a regression model is combined with a radial model, which often only has a localized effect. The radial model

consists of radial functions and parameters. In the presentation we derived how to simultaneously determine the parameters of the regression model and the radial model. We also briefly discussed a related method, namely the method of multivariate adaptive regression splines.

The single point models are a class of models valid only in a local region around a single point. We have presented the reciprocal approximation, the conservative approximation and the posynomial approximation. In the literature the valid area of these approximations are claimed to be wider compared to that of a corresponding Taylor model. Models of this type have gained popularity in the mechanical engineering community.

2.3 Surrogates Based on Physical Models

We will now turn the attention to surrogate models which are specific for the particular physical system in question. We will assume that besides the response function f from the expensive model, a user provided, cheap (lower-fidelity), physical model with the response function $c : \mathbb{R}^n \mapsto \mathbb{R}$ is available.

For some problems, the cheap model may itself act as a surrogate for the expensive model. However, we cannot in general expect that any given cheap model approximates the expensive model well. Often, the deviations between two physical models can be referred to problems with incorrect alignment of the responses or the parameter spaces.

In the case of incorrectly aligned response functions, we could apply a response correction g on c and obtain the surrogate $g(c(x))$. The response correction could e.g. be a simple scaling function. By imposing conditions on the form and behaviour of the response correction g, we can make the surrogate interpolate f and its gradients at given points. Two methods of this type is presented in the next section.

One method for performing response scaling, called the *multipoint method*, is presented in Section 2.3.2. Here, a number of cheap models, related to subsystems of the original physical system, are used as regression functions in a data fitting problem. The resulting regression parameters can be viewed as scaling parameters of the cheap models.

In the case of incorrectly aligned parameter spaces, we could apply a transformation of the cheap model parameter space, say a mapping P, and obtain the surrogate $c(P(x))$. By imposing conditions on this so-called *space mapping* P,

we can make the surrogate approximate f and its gradients at given points. Space mapping techniques are discussed in Section 2.3.3.

2.3.1 Response Correction

Scaling of response functions is a mean of correcting physical models which lack approximation abilities like interpolation. Consider the surrogate s which is a *response correction* $g : \mathbb{R} \mapsto \mathbb{R}$ of the response function c from the physical model, hence

$$s(x) = g(c(x)) \; . \tag{2.17}$$

Such a response correction g could be identified by imposing conditions on s, e.g. at a given point x_k the function values and the gradients match, i.e. $s(x_k) = f(x_k)$ and $s'(x_k) = f'(x_k)$. Two methods of this type are considered below, the first method, the β-correlation method, performs a simple scaling of c, the second method is more general and seek to approximate an assumed function g for which $s(x) = f(x)$.

The β-correlation Method

We now consider the case of response correction (2.17) where g is a simple scaling of c. Hence $g(c(x)) = a(x)c(x)$, where $a(x)$ is the scaling function. Such a response scaling method, called the *β-correlation method*, is presented in [16] as a generic approach to correcting a lower-fidelity model response by scaling. The method assumes the existence of a smooth function $a(x)$ for which

$$a(x)c(x) = f(x) \quad \text{and} \quad \nabla \left(a(x)c(x) \right) = f'(x) \, .$$

Taylor's theorem provides the following approximation,

$$g(c(x+h)) \simeq \left(a(x) + a'(x)^T \, h \right) c(x+h) \, . \tag{2.18}$$

Using (2.18), at a given point x_k, we obtain the approximation,

$$\begin{aligned} s_k(x) &= \left(a(x_k) + a'(x_k)^T (x - x_k) \right) c(x) \\ &= \left(\frac{f(x_k)}{c(x_k)} + \left(\frac{f'(x_k)c(x_k) - c'(x_k)f(x_k)}{(c(x_k))^2} \right)^T (x - x_k) \right) c(x) \; , \end{aligned}$$

which performs a scaling of c such that s_k and s'_k interpolate f and f' at x_k. If f' is too expensive to calculate, an approximation can be used [1], e.g. obtained using a secant method. Alternatively, the derivatives $a'(x_k)$ could be approximated directly by a secant method.

A shortcoming of the β-correlation method is that if c vanishes at a point x_k the scaling is undefined at this point. The more general method presented next mends this shortcoming.

General Response Correction

A more general approach to response correction (2.17) is to assume existence of a smooth function g for which

$$f(x) = g(c(x)) \,, \tag{2.19}$$

and then approximate this function. We will show how to make a first order approximation to g assuming knowledge of the first derivatives of f, thereafter we show how a secant method can be used instead, not requiring knowledge of f'.

Taylor's theorem provides the following approximation

$$g(c(x+h)) \simeq g(c(x)) + g'(c(x))[c(x+h) - c(x)] \,. \tag{2.20}$$

At a point x_k, applying the interpolation conditions $s(x_k) = f(x_k)$ and $s'(x_k) = f'(x_k)$ on (2.20), we obtain the surrogate

$$s_k(x) = f(x_k) + g'(c(x_k))[c(x) - c(x_k)] \,, \tag{2.21}$$

with $g'(c(x_k))c'(x_k) = f'(x_k)$. In practice we only expect (2.19) to hold approximately, we can then choose $g'(c(x_k))$ as the solution of the linear least-squares problem

$$\min_{a \in \mathbb{R}} \|a \cdot c'(x_k) - f'(x_k)\|_2 \,.$$

Similar to the β-correlation method described above, if $f'(x_k)$ is too expensive to calculate, an approximation, e.g. obtained using a secant method, can be used.

An alternative approach is to sequentially approximate the gradient $g'(c(x))$ by scalars $a_k \in \mathbb{R}$, $k = 0, 1, \ldots$, which, for a given sequence of points $\{x_k\}$, obey the secant condition

$$f(x_{k+1}) - f(x_k) = a_{k+1}[c(x_{k+1}) - c(x_k)] \,, \tag{2.22}$$

where $a_0 \in \mathbb{R}$ is a user provided initial approximation to $g'(c(x_0))$.

The surrogate becomes

$$s_k(x) = f(x_k) + a_k[c(x) - c(x_k)] \,, \tag{2.23}$$

where a_k is found using (2.22). The user has to provide an initial approximation a_0, the most obvious choice is to assume no scaling, hence let $a_0 = 1$.

The advantage of the surrogate in (2.23) over (2.21) is obvious, since for most practical problems we cannot rely on the expensive response derivatives $f'(x_k)$ being available.

2.3.2 The Multipoint Method

The *multipoint method* proposed by Toropov in [55], and further described in [54, 56], is a method for creating physics based regression functions, for systems that can be partitioned into individual subsystems.

The multipoint method constructs an approximation based on partitioning the physical system into, say q, individual subsystems, which again are described by empirical expressions or known analytical solutions c_i, $i = 1, \ldots, q$. The approximation in its simplest form is

$$s(x) = \beta^{(0)} + \beta^{(1)} c_1(x) + \ldots + \beta^{(q)} c_q(x) + \sum_{j \geqslant 1} \beta^{(j+q+1)} x_s^{(j)} \tag{2.24}$$

where c_i is based on the physics of the ith subsystem. The parameters β are model parameters, which are determined by least-squares regression using presampled data, as in (2.3). The vector x_s, is a subset of the design variables, $x_s \subseteq x$, that has a global influence on the physical system. The idea is that each subsystem may depend only on a subset of the design variables, and that only a few (global) variables are related with the system behaviour as a whole.

Instead of using a simple linear combination (2.24), which we can write as

$$s(x) = \beta^{(0)} + \sum_{j \geqslant 1} \beta^{(j)} v_j(x) \,,$$

Toropov suggested three possible nonlinear formulations: multiplication,

$$s(x) = \beta^{(0)} \prod_{j=1}^{q} v_j(x)^{\beta^{(j)}} \,,$$

the exponential of the linear combination,

$$s(x) = \exp\left(\beta^{(0)} + \sum_{j=1}^{q} \beta^{(j)} v_j(x)\right) ,$$

and a power function of the linear combination,

$$s(x) = \left(\beta^{(0)} + \sum_{j=1}^{q} \beta^{(j)} v_j(x)\right)^{\alpha} ,$$

for some α. Further, Toropov suggested to use a logarithmic transformation on the above formulations to make them linear, for easier calculation of the β's. In his work Toropov not only approximated the objective function, but also nonlinear constraint functions using his multipoint method.

2.3.3 The Space Mapping Method

The *space mapping method* is a method for aligning the parameter spaces of physical models. Here, the space mapping is a parameter transformation that makes the cheaper model c exhibit same behaviour as the expensive model f. The concept of space mapping was introduced in [7], and it is reviewed in [6] and [8].

Let the function $P : \mathbb{R}^n \mapsto \mathbb{R}^n$ be the space mapping, which transforms the parameter space of c in such a way that the composite function $c \circ P$, the so-called *space mapped model*, can act as a surrogate for f. Conditions imposed on P determine the nature of the alignment.

Theoretically, see Chapter 4, ideal conditions ensuring that the minimizer of $c \circ P$ is x^*, the minimizer of f, are that the space mapping relate x^* and the minimizer of c, and that P is a one-to-one mapping.

In practice there has not yet been proposed a formulation of the space mapping for which these theoretical conditions are always meet. So, most often the formulation of the space mapping is based on the approximation condition

$$f(x) \simeq c(P(x)) ,$$

which characterizes a formulation of the space mapping connecting similar responses,

$$P(x) = \arg\min_{\tilde{x}} \|c(\tilde{x}) - f(x)\|_2 . \tag{2.25}$$

The problem from which $P(x)$ is calculated, as e.g. (2.25), is referred to as the *parameter extraction problem*.

In the literature, see [6], the space mapping method is most commonly used for problems involving vector response functions $f : \mathbb{R}^n \mapsto \mathbb{R}^m$ and $c : \mathbb{R}^n \mapsto \mathbb{R}^m$, with $m \gg n$, rather than scalar response functions, $m = 1$. The objective function is then measured using a convex merit function H, usually a norm, see (2.2). Assuming the problem is defined by such vector functions, then the formulation in (2.25) is expected to have a unique solution, since it is over-determined following the assumption $m \gg n$.

Note that for the space mapping formulation in (2.25), we cannot expect that $c \circ P$ interpolates f, and we cannot be sure that the minimizers are the same. But, if the norm of the residual in (2.25) is small we can expect that the minimizer of $c \circ P$ is close to x^*, the minimizer of f, refer to Chapter 4.

Other formulations of P have been proposed mainly in an effort to ensure uniqueness in the parameter extraction problem, again we refer to Chapter 4.

Since the space mapping P is at least as expensive to evaluate as f, for all practical purposes P must be approximated by a model. In context of optimization, a linear approximation is often used. Nonlinear approximations of P have been introduced in cases where $c \circ P$ is used for modelling.

2.3.4 Summary

In the last sections we have reviewed three methods for generating surrogates for expensive functions based on physical models. The methods are based on manipulating a cheaper model (or several cheaper models), such that the manipulated response of the cheap model approximates the response of the expensive model. We summarize the three methods in the following.

The response correction methods try to either multiply the cheap model response with a correction function (the β-correlation method), or make a composed model of a correction function and the cheap model (general response correction).

The multipoint method uses a (often linear) combination of response functions from several cheap models. The parameters of this combined model are determined by least-squares regression using known, pre-sampled expensive model responses.

The space mapping method tries to align the parameter space of the cheap model with that of the expensive model. This is done by connecting similar responses. Most often, the space mapping method is used for problems where the response functions are vector valued, which enhance uniqueness when establishing the space mapping.

2.4 Optimization Using Surrogate Models

We now present algorithms for solving the problem in (2.1) using surrogate models. The purpose of the surrogate s_k is to take the place of the expensive response function f in sequential subproblems solved by the algorithms. The hope is that by allowing extensive use of the surrogates, the number of evaluations of f, needed in order to locate the optimizer x^*, can be vastly reduced.

First we present a non rigorous approach using response surface models, then we introduce two provably convergent algorithms, namely a trust region algorithm and a pattern search based algorithm. At last we present some methods based on the space mapping idea, one of them is provably convergent.

2.4.1 Response Surface Methodology

Response surface Methodology (RSM) is a procedure of sequential experimentation in an effort of optimizing a response function, which typically comes from a physical model. The origin of RSM is the seminal paper by Box and Wilson [14].

RSM covers the process of identifying and fitting from experimental data a response surface model. This process requires knowledge of design of experiments (DOE), regression modelling techniques (how to choose a good model) and optimization techniques.

It involves a sequence of experiments each of which determines a direction of better response value until a local optimum is reached. A response surface can be generated for a wide range of regression models, although the term *response surface design* is commonly used to refer to quadratic regression models, as (iii) in (2.4).

The statistical foundation used in many applications involving response surface methods makes it possible to determine confidence intervals on the approximations obtained by the regression. Further, several statistical methods

exist for doing analysis on the residuals, outliers and lack of fit. Tools such as variable transformation are frequently used, to enhance the fit, see e.g. [13, 42].

With special focus on computer experiments [49] suggested the simple one-shot approach: first sample the design space sufficiently dense and then construct a surrogate using kriging, finally apply an optimization method to obtain the solution. This approach is not very likely to yield a satisfactory result, except if the design space is sampled so dense that the approximation becomes very close to the true objective. However, the expensive true objective prohibits such an approach. Hence a sequential strategy, regularly refining the surrogate model, is more likely to be successful. Fortunately, most applications of response surface methods are actually sequential in nature.

In Algorithm 1 we summarize an optimization procedure suggested in [42]. We have the following comments about the algorithm:

Algorithm 1 Response Surface Methodology

Phase 0:

Select the important variables by a screening experiment and by analysis of variance

Phase 1:

Require: $k = 0$

while x_k is not near x^* **do**

Construct a first order model s_k, based on using data from an experiment on f using an experimental design in a small region of interest around x_k

Determine the direction of steepest descent, $h_{sd} = -s'(x_k)$

Find an $\alpha > 0$ which approximately solves $\min_\alpha f(x_k + \alpha h_{sd})$

Set $k = k + 1$

end while

Phase 2:

Require: x_k near x^*

Construct a second order model s_k covering the optimal region

Determine the solution by solving the quadratic problem $\min_x s_k(x)$

Phase 0 of the algorithm suggests initial screening experiments on the system in order to identify important variables by analysis of variance (ANOVA) The ANOVA decompose the system behaviour into main effects (contribution of individual variables to variation in the response) and interaction effects (contributions of combinations of variables to variation in the response). Variables

with little contribution to the variation in the response can (in a first run of the algorithm) be left out. Using this technique may be a good idea especially in cases where there is a large number of variables.

Reducing the number of parameters in this initial stage can be crucial to the performance of the optimization procedure to come, see e.g. [11] for a 31 variable problem that by a screening experiment identified 11 key variables that accounts for the most of the variation in the response; the optimization procedure was then applied to the reduced problem. Another example is in [15] where a screening experiment was used to reduce a 28 variable problem, from aerospace vehicle design, before an optimization procedure was applied.

In phase 1 of the algorithm, [42] suggests using methods to test for curvature and possible lack-of-fit, in order to determine if a near optimal point has been reached.

In phase 2 the iterate is assumed so close to the optimizer that a single step using a quadratic model will give us the optimizer.

In [63] a so-called Bayesian-validated statistical framework for optimization using surrogates is presented. The ideas are very similar to what we have just presented in Algorithm 1.

Clearly, Algorithm 1 is not rigorous, and it is most likely not meant to be. Considering the origin of response surface methodology, in statistics, the algorithm is more like a practical guide for people trying to optimize very noisy systems as e.g. parameters of a machine running in a production environment. So, for problems with deterministic response functions, the main focus of this work, the algorithm is not well-suited. Anyhow, the algorithm is close in spirit, apart for the screening experiment, to the rigorous trust region algorithm we present below.

2.4.2 Trust Region Approach

An algorithm with a model management framework based on trust region methodology is presented in [1]. This trust region algorithm solves (2.1), requiring that the surrogate interpolate and that the first order information match, hence

$$\begin{aligned} s_k(x_k) &= f(x_k) \\ s_k'(x_k) &= f'(x_k)\ . \end{aligned}$$

The requirement can be extended to include second order information as well, $s_k''(x_k) = f''(x_k)$, or even higher derivatives, but it is not used, as higher order information is generally not available in most practical problems. We should note that the proposed algorithm can be modified to use approximate gradient information, as e.g. secant approximations [17, 1]. A trust region method is presented in [24] for the case where only zero order interpolation is required.

An outline of the algorithm is presented here as Algorithm 2. The trust region size Λ is controlled using conventional updating strategies, see [17], based on the computed gain factor ρ_k.

Algorithm 2 Trust region algorithm

Require: $k = 0$, $x_0 \in \mathbb{R}^n$, $\Lambda_0 > 0$
  while not converged **do**
    Select model $s_k(x_k) = f(x_k)$ and $s_k'(x_k) = f'(x_k)$
    Solve approximately for $h = x - x_k$:

$$\begin{aligned} \underset{h}{\text{minimize}} \quad & s_k(x_k + h) \\ \text{subject to} \quad & \|h\| \leqslant \Lambda_k \end{aligned}$$

    Compute $\rho_k \equiv \dfrac{f(x_k + h) - f(x_k)}{s_k(x_k + h) - s_k(x_k)}$
    if $f(x_k) > f(x_k + h)$ **then**
      accept the step, $x_{k+1} = x_k + h_k$
    else
      reject the step, $x_{k+1} = x_k$
    end if
    Update Λ_k
    Set $k = k + 1$
  end while

The part of the algorithm (approximately) solving the trust region problem

$$\begin{aligned} &\min_h s_k(x_k + h) \\ &\text{s.t. } \|h\| \leqslant \Lambda_k \ , \end{aligned}$$

in the case where s_k is general nonlinear function, can be implemented using a standard nonlinear programming method, see e.g. [17, 44]. If s_k is simple, e.g. linear or quadratic, the problem is solved by standard linear or quadratic programming methods, assuming that the trust region is measured in the ℓ_∞-norm. When measuring the trust region in the the ℓ_2-norm an approach due to Moré and Sorensen can be used, see e.g. [44].

It is not required that the trust region problem is solved to high accuracy, a solution providing a decrease in s(x) that is some positive fraction of that provided by the so-called *Cauchy point* is sufficient to guarantee convergence, see [17] for a thorough discussion.

The global convergence of the algorithm can be proven by imposing mild conditions on f, [1] states that a sequence of iterates $\{x_k\}$ generated by Algorithm 2 is convergent and that a result similar to

$$\liminf_{k\to\infty} \|f'(x_k)\| = 0$$

can be proven.

2.4.3 Pattern Search Approach

The trust region algorithm presented above was in [12, 50, 53, 57] developed under names as *a rigorous framework for optimization using surrogates* and *model assisted grid search* into a pattern search method enforcing only very mild conditions on the surrogate — the actual implemented algorithm used interpolating DACE surrogates (see page 21), but the framework can handle non-interpolating surrogates. Global convergence to a stationary point for a bound-constrained version of (2.1) is proven in [12].

The algorithm is based on two phases in each iteration.

The first phase is a user specified method, that is allowed to perform an exhaustive search on the surrogate at design points limited to a particular grid in the design space. If one or more promising points — including at least one point for which $f(x)$ is unknown — were found in the search, the objective f is evaluated at these points. That point, if any, which provides the most decrease, compared to the current iterate, is accepted as the next iterate. If such a point of descent is found, the surrogate is re-calibrated, if necessary, and the first phase is repeated.

If the first phase fails to provide decrease in the objective function, the algorithm enters the second phase. Here is sought a point on the grid, among neighbors to the current iterate, which provides a decrease in the objective. If one such point is found, it is accepted as the next iterate; the surrogate is re-calibrated, if necessary, and the algorithm jumps to the start of the first phase. If none of the neighboring points are better than the current iterate, the grid is refined (e.g. the distance between points is halved), and the algorithm jumps to the start of the first phase.

The algorithm, for the unconstrained problem with an interpolating surrogate, is summarized in Algorithm 3. Here Λ is a measure of the grid density, halving Λ is equivalent to halving the distance between grid points. We remark that it is the second phase of the algorithm, which ensures convergence, that is, if the grid is defined from a set of basis vectors which forms a so-called *positive basis*, see [52] for convergence proof and discussion of the algorithm.

Algorithm 3 Model Assisted Grid Search

Require: $k = 0$, $x_0 \in I\!R^n$, $\Lambda_0 > 0$, initial surrogate s_0
  while not converged **do**
    Phase 1:
    Search for points $T = \{\tilde{x}_j\}$, $j = 1, 2, \ldots$, on the grid, for which $s_k(\tilde{x}_j) < f(x_k)$. If $T \neq \emptyset$, assure that T contains at least one point where $f(x)$ is unknown
    if $T \neq \emptyset$ and $f(\hat{x}) < f(x_k)$ for some $\hat{x} \in T$ **then**
      Accept $\hat{x}$ as new iterate, set $x_{k+1} = \hat{x}$
      Re-calibrate surrogate if necessary
    else
      Phase 2:
      Find a neighboring point (on the grid) $\hat{x}$ to x_k for which $f(\hat{x}) < f(x_k)$
      if $\exists \hat{x} \ : \ f(\hat{x}) < f(x_k)$ **then**
        Accept $\hat{x}$ as new iterate, set $x_{k+1} = \hat{x}$
        Re-calibrate surrogate if necessary
      else
        Keep the current iterate, set $x_{k+1} = x_k$
        Refine the mesh, set $\Lambda_{k+1} = \Lambda_k/2$
      end if
    end if
    Set $k = k + 1$
  end while

As an extension to the algorithm Torczon and Trosset [53] introduced the use of a merit function in phase 1, that balance the goals of providing decrease in the objective function and improving the overall approximation model. Essentially the proposed merit function $\hat{s}_k(x)$ balance the local predictive capability and global accuracy of the approximation;

$$\hat{s}_k(x) = s_k(x) - \rho_k d_k(x) \ , \tag{2.26}$$

where $\rho_k \geqslant 0$ and $d_k(x) = \min_i \|x - \hat{x}_i\|$ is the distance from x to the nearest

site where f previously has been evaluated. The last term is introduced to inhibit clustering and thereby to ensure that the next point is placed where information from f will improve the current approximation. The parameter ρ_k should eventually vanish to ensure global convergence of the modified algorithm.

We should note that the proposed merit function is not differentiable, which restricts the algorithm to a much smaller class of optimization methods for optimizing the merit function in the first phase of the algorithm.

The parameter ρ_k resembles in a sense the "temperature" parameter in *simulated annealing* algorithms, see e.g. [34], as increasing the parameter tends to make the search more global and reducing the parameter tends to make the search more local. However, in simulated annealing algorithms the iterates are not found in a deterministic way.

2.4.4 Space Mapping Optimization

The original formulation of the space mapping optimization problem, see [3, 7], is to solve the n nonlinear equations

$$P(x) = z^* \ , \tag{2.27}$$

for x. Here P is the space mapping described in Section 2.3.3 and z^* is an optimizer of c, the cheap model. The nonlinear equations (2.27) may be solved using any standard method for solving nonlinear equations, e.g. the Dog-Leg method described in [23].

Under certain conditions, as shown in Chapter 4, the space mapping problem (2.27) has the alternative formulation

$$\min_x s(x) \equiv c(P(x)) \ . \tag{2.28}$$

In fact, the latter formulation is preferable in the case where the two formulations do not have the same solution.

A trust region based surrogate optimization algorithm for solving the alternative problem formulation (2.28) is presented as Algorithm 4.

In the algorithm the space mapping is iteratively approximated by a linear Taylor model. So, at a given point x_k, the surrogate is

$$s_k(x) = c(P_k(x))$$

Algorithm 4 Space mapping optimization algorithm

Require: $k = 0$, $x_0 = z^*$, $\Lambda_0 > 0$, $B_0 = I(n)$
  Find $P(x_0)$
  while not converged **do**
    Solve approximately for $h = x - x_k$:
      $\underset{h}{\text{minimize}} \quad s_k(x_k + h)$
      subject to $\quad \|h\| \leqslant \Lambda_k$
    Find $P(x_k + h)$
    Compute $\rho_k \equiv \dfrac{s(x_k + h) - s(x_k)}{s_k(x_k + h) - s_k(x_k)}$
    if $s(x_k) > s(x_k + h)$ **then**
      accept the step
    else
      reject the step
    end if
    Update Λ_k, B_k
    Set $k = k + 1$
  end while

where P_k is a linear model

$$P_k(x) = B_k(x - x_k) + P(x_k) \, ,$$

and B_k is an approximation to $P'(x_k)$ obtained by a secant method. Choosing the identity matrix as initial approximation, $B_0 = I(n)$, corresponds to the (initial) assumption that the response functions are identical, $c(x) = f(x)$, see Chapter 3 (the paper [5]).

Note that the starting point x_0 is chosen as the optimizer z^* of the cheap model, following the initial assumption that the models are identical. As for the case with the trust region algorithm, Algorithm 2, the sub-problem of minimizing $s_k(x)$ within the trust region can be solved using standard non-linear programming methods, see e.g. [17, 44]. In the case of vector response functions, described in Section 2.3.3, the sub-problem can be solved using one of the methods described in [38].

If the conditions mentioned in Section 2.3.3 are satisfied the algorithm is convergent, see also Chapter 4. However, if the conditions are not satisfied, as often is the case, the algorithm may not converge to x^*.

In [51] the so-called *combined model*

$$s_k(x) = w_k\, c(P_k(x)) + (1 - w_k)\, l_k(x)$$

was introduced. Here l_k is a first order Taylor model, with a secant approximation to the derivatives, of f. The scalar w_k, $0 \leqslant w_k \leqslant 1$, is a weighting factor, controlling the actual combination of the surrogate at iteration k in the optimization process. A trust region based algorithm, similar to Algorithm 2, using this combined model was proposed in [51]. In the algorithm, the parameter w is reduced monotonically from 1 to 0 during iterations. Hence, the surrogate gradually is transformed from a mapped model approximation into a linear approximation of f. Convergence to x^*, the solution of (2.1), has been proved for algorithms of this type, the so-called *hybrid space mapping algorithms*, see [60] and Chapter 6 (the paper [39]). Other strategies for controlling w_k have been proposed, see [4, 45, 51].

Using such a hybrid algorithm, the space mapping method serves as a preconditioner for solving (2.1). That is, by using the solution provided by the space mapping method as a starting point for a Taylor based method, we may reduce the number of evaluations of f needed to determine x^*.

An alternative approach [22] is to combine a response correction, see Section 2.3.1, and space mapped model, ensuring that the interpolation conditions are satisfied for use of the trust region algorithm (Algorithm 2). Such a strategy is investigated in Chapter 5.

In [35] a space mapping approach for constrained problems where introduced, designating an individual space mapping to each constraint function.

2.4.5 Summary

In the last sections we have reviewed four methods for optimization using surrogate models. We summarize these methods in the following.

The response surface methodology is a loosely defined framework for using regression models, in particular response surface models, for optimization. Response surface methodology employs several techniques developed in the statistical community, among these the sensible technique of screening a given problem in order to identify the most important variables.

The trust region approach is a rigorously defined framework for optimization using interpolating models. The fundamental technique is well-studied, and convergence of this method can be proved.

The pattern search approach enforces only very mild conditions on the surrogate model. The concept of the method is very simple, and convergence can be proved.

The space mapping optimization method employs the response from a cheap physical model composed of an approximation to the space mapping. The space mapping connects similar responses of the cheap and the expensive models. Usually, in the context of optimization, a local linear Taylor model is used for the space mapping. The method is not convergent in general. Modifications have been suggested in order to ensure convergence.

2.5 Conclusion

This chapter presented a literature overview of surrogate modelling and optimization. We have aimed at presenting the essential parts of the many aspects presented in the literature in a consistent simple notation.

We started the chapter by reviewing the most frequently used terminology of the literature. From that, we decided to divide the study into two parts, namely methods for generating surrogate models and methods for optimization using surrogate models. Further we divided the surrogate models into two categories: Functional models and physical models. We first summarize the functional models, then the physical models and finally we summarize the methods for optimization using surrogate models.

The functional models can be constructed without any knowledge of the underlying physical system. They are generic models, based on algebraic expressions and sampled data. We presented three types of functional models, namely regression models, models based on radial functions and single point models.

The physical models incorporate knowledge from the particular physical system in question. This is usually done by manipulating a cheaper model of the same physical system, so that it better approximates the behaviour of the expensive model of the system. We presented three types of methods for generating physical models, namely the response correction method, the multipoint method and the space mapping method.

Finally, we considered methods for optimization using surrogate models. We presented four methods, namely response surface methodology, the trust region approach, the pattern search approach and the space mapping method. Convergence can be proved for the trust region and pattern search approaches. The space mapping method can be made convergent if it is combined with classical optimization methods. Convergence for such a hybrid method is proven in Chapter 6.

The space mapping method is the topic of the remaining part of this thesis. The next chapter, Chapter 3, introduces and motivates the space mapping method.

Symbols

$\|\cdot\|$	unspecified norm
$\|\cdot\|_2$	Euclidean norm, $\|x\|_2 = (x^T x)^{\frac{1}{2}}$
b	vector with the radial functions at t, see (2.6)
c	response from cheap physical model, $c : \mathbb{R}^n \mapsto \mathbb{R}^m$
f	response from expensive physical model, $f : \mathbb{R}^n \mapsto \mathbb{R}^m$
H	convex merit function, usually a norm, $H : \mathbb{R}^m \mapsto \mathbb{R}$
k	iteration counter in optimization procedure, $k = 1, 2, \ldots$
m	number of response functions, $m = 1$ except otherwise noted
n	dimensionality of the design parameter space, $x \in \mathbb{R}^n$
p	number of data points (t_j, y_j), $j = 1, \ldots, p$
P	space mapping, relating parameters of f and c, $P : \mathbb{R}^n \mapsto \mathbb{R}^n$
q	number of cheap physical models, c_i, $i = 1, \ldots, q$
x	optimizeable model parameters of f and c
x^*	minimizer of f, see (2.1)
x_k	kth iterate in an optimization procedure
s	surrogate model, to take the place of f
s_k	surrogate in the kth iteration of an optimization procedure
t	design sites, used to generate functional models, $t = (t_1, \ldots, t_p)^T$, where $t_j \in \mathbb{R}^n$, $j = 1, \ldots, p$
u	number of basis functions in a regression model
v	vector with basis functions (for regression), $v : \mathbb{R}^n \mapsto \mathbb{R}^u$
y	expensive model evaluated at the design sites, $y = (y_1, \ldots, y_p)^T$, $y_j = f(t_j)$, $j = 1, \ldots, p$
z	residual function, see section 2.2.2
β	regression constants, $\beta \in \mathbb{R}^u$
γ	radial model constants, $\gamma \in \mathbb{R}^q$
ϕ	radial function, $\phi_j(x) = \phi(x - t_j)$, $j = 1, \ldots, p$

References

[1] N. Alexandrov, J.E. Dennis, Jr., R.M. Lewis, V. Torczon, *A Trust Region Framework for Managing the Use of Approximation Models in Optimization*, Structural Optimization, vol. 15, no. 1, pp. 16-23, 1998.

[2] C. Audet, J.E. Dennis, Jr., L.N. Vicente (eds.), *Surrogate Optimization*, forthcoming special issue of Optimization and Engineering, 2003.

[3] M.H. Bakr, J.W. Bandler, N. Georgieva, *An Aggressive Approach to Parameter Extraction*, IEEE Trans. Microwave Theory Tech., vol. 47, pp. 2428–2439, 1999.

[4] M.H. Bakr, J.W. Bandler, K. Madsen, J.E. Rayas-Sánchez, J. Søndergaard, *Space Mapping Optimization of Microwave Circuits Exploiting Surrogate Models*, IEEE Trans. Microwave Theory Tech., vol. 48, no. 12, pp. 2297–2306, 2000.

[5] M.H. Bakr, J.W. Bandler, K. Madsen, J. Søndergaard, *An Introduction to the Space Mapping Technique*, Optimization and Engineering, vol. 2, no. 4, pp. 369–384, 2001.

[6] M.H. Bakr, J.W. Bandler, K. Madsen, J. Søndergaard, *Review of the Space Mapping Approach to Engineering Optimization and Modeling*, Optimization and Engineering, vol. 1, no. 3, pp. 241–276, 2000.

[7] J.W. Bandler, R.M. Biernacki, S.H. Chen, P.A. Grobelny, R.H. Hemmers, *Space Mapping Technique for Electromagnetic Optimization*, IEEE Trans. Microwave Theory Tech., vol. 42, pp. 2536–2544, 1994.

[8] J.W. Bandler, Q. Cheng, S. Dakroury, A.S. Mohamed, M.H. Bakr, K. Madsen, J. Søndergaard, *Space Mapping: The State of the Art*, submitted, IEEE Trans. Microwave Theory Tech., 2004.

[9] J.W. Bandler, K. Madsen (eds.), *Surrogate Modelling and Space Mapping for Engineering Optimization*, (special issue) Optimization and Engineering, no. 2, 2001.

[10] J.-F.M. Barthelemy, R.T. Haftka, *Approximation Concepts for Optimum Structural Design - a Review*, Structural Optimization, vol. 5, pp. 129–144, 1993.

[11] A.J. Booker, J.E. Dennis, Jr., P.D. Frank, D.B. Serafini, V. Torczon, *Optimization Using Surrogate Objectives on a Helicopter Test Example*, Technical report 97-31, Department of Computational and Applied Mathematics, Rice University, Houston, Texas 77005-1892, 1997.

[12] A.J. Booker, J.E. Dennis, Jr., P.D. Frank, D.B. Serafini, V. Torczon, M.W. Trosset, *A Rigorous Framework for Optimization of Expensive Functions by Surrogates*, Structural Optimization, vol. 17, pp. 1–13, 1999.

[13] G.E.P. Box, N.R. Draper, *Empirical Model-Building and Response Surfaces*, Wiley, 1987.

[14] G.E.P. Box, K.B. Wilson, *On the Experimental Attaintment of Optimum Conditions*, Journal of the Royal Statistical Society, Series B, vol. 13, pp. 1–45, 1951.

[15] S. Burgee, A. Giunta, V. Balabanov, B. Grossman, W. Mason, R. Narducci, R. Haftka, L. Watson, *A Coarse-grained Parallel Variable-Complexity Multidisciplinary Optimization Problem*, The International Journal of Supercomputing Applications and High Performance Computing, vol. 10, pp. 269–299, 1996.

[16] K.J. Chang, R.T. Haftka, G.L. Giles, P.-J. Kao, *Sensitivity-Based Scaling for Approximating Structural Response*, Journal of Aircraft, vol. 30, 1993.

[17] A.R. Conn, N.I.M. Gould, P.L. Toint, *Trust-Region Methods*, SIAM, Philadelphia, 2000.

[18] C. Currin, T. Mitchell, M. Morris, D. Ylvisaker, *Bayesian Prediction of Deterministic Functions, With Applications to the Design and Analysis of Computer Experiments*, Journal of the American Statistical Association, vol. 86, no. 416, pp. 953–963, 1991.

[19] V. Czitrom, P.D. Spagon, *Statistical Case Studies for Industrial Process Improvement*, ASA-SIAM series on statistics and applied probability, 1997.

[20] I. Daubechies, *Ten Lectures on Wavelets*, Philadelphia, SIAM, 1992.

[21] J.E. Dennis, Jr., *Surrogate Modelling and Space Mapping for Engineering Optimization: A Summary of the Danish Technical University November 2000 Workshop*, Technical Report, TR00-35, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, USA, 2000.

[22] J.E. Dennis, Jr., Dept. Computational and Applied Mathematics, Rice University, Houston, Texas, USA, private discussions, 2001.

[23] J.E. Dennis, Jr., R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[24] J.E. Dennis, Jr., V. Torczon, *Managing Approximation Models in Optimization*, in Multidisciplinary Design Optimization: State-of-the-Art, N.M. Alexandrov, M.Y. Hussaini, eds., SIAM, Philadelphia, 1997.

[25] O. Dubrule, *Two Methods with Different Objectives: Splines and Kriging*, Mathematical Geology, vol. 15, no. 2, 1983.

[26] A.A. Giunta, V. Balabanov, D. Haim, B. Grossman, W.H. Mason, L.T. Watson, R.T. Haftka, *Multidisciplinary Optimization of a Supersonic Transport Using Design of Experiments Theory and Response Surface Modelling*, The Aeronautic Journal, pp. 347–356, Oct. 1997.

[27] H.-M. Gutmann, *A Radial Basis Function Method for Global Optimization*, Journal of Global Optimization, no. 19, pp. 201–227, 2001.

[28] R.T. Haftka, E.P. Scott, *Optimization and Experiments — a Survey*, in Tatsumi, T. et al (eds.), Theoretical and Applied Mechanics 1996, Proc. of XIX ICTAM, Kyoto, 1996, Elsevier, pp. 303-321, 1997.

[29] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning — Data Mining, Inference and Prediction*, Springer, 2001.

[30] C.R. Hicks, *Fundamental Concepts in the Design of Experiments*, Holt, Rinehart and Winston, Inc, 1964.

[31] Howard Demuth, Mark Beale, *Neural Network Toolbox User's Guide*, The Mathworks, Inc., 1994.

[32] K.E. Johnson, K.W. Bauer, J.T. Moore, M. Grant, *Metamodelling Techniques in Multidimensional Optimality Analysis for Linear Programming*, Mathl. Comput. Modelling, vol. 23, no. 5, pp. 45–60, 1996.

[33] D.R. Jones, M. Schonlau, W.J. Welch, *Efficient Global Optimization of Expensive Black-Box Functions*, Journal of Global Optimization, vol. 13, no. 4, pp. 455-492, 1998.

[34] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, *Optimization by Simulated Annealing*, Science, vol. 220, pp. 671–680, 1993.

[35] S.J. Leary, A. Bhaskar, A.J. Keane, *A Constraint Mapping Approach to the Structural Optimization of an Expensive Model using Surrogates*, Optimization and Engineering, vol. 2, no. 4, pp. 385–398, 2001.

[36] S.N. Lophaven, H.B. Nielsen, J. Søndergaard, *Aspects of the Matlab Toolbox DACE*, technical report IMM-REP-2002-13, Informatics and Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 2002.

[37] S.N. Lophaven, H.B. Nielsen, J. Søndergaard, *DACE – A Matlab Kriging Toolbox, Version 2.0,* technical report IMM-REP-2002-12, Informatics and Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 2002.

[38] K. Madsen, *Minimization of Non-Linear Approximation Functions*, Dr. Techn. Thesis, Institute for Numerical Analysis, Technical University of Denmark, 1986.

[39] K. Madsen, J. Søndergaard, *Convergence of Hybrid Space Mapping Algorithms*, submitted, Optimization and Engineering, 2003.

[40] G. Matheron, *Principles of Geostatistics*, Economic Geology, vol. 58, pp. 1246–1266, 1963.

[41] M.D. McKay, W.J. Conover, R.J. Beckman, *A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code*, Technometrics, vol. 21, no. 2, 1979.

[42] R.H. Myers, D.C. Montgomery, *Response Surface Methodology*, second edition, John Wiley & Sons, Inc., 2002.

[43] H.B. Nielsen (ed.), *Surrogate Modelling and Space Mapping for Engineering Optimization — Papers*, from the "First International Workshop on Surrogate Modelling and Space Mapping for Engineering Optimization", Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, November 16–18, 2000.

[44] J. Nocedal, S.J. Wright, *Numerical Optimization*, Springer Series in Operations Research, Springer, 1999.

[45] F.Ø. Pedersen, *Advances on the Space Mapping Optimization Method*, Master Thesis, IMM-THESIS-2001-35, Informatics and Mathematical Modelling, Technical University of Denmark, 2001.

[46] M.J.D. Powell, *The Theory of Radial Basis Funcion Approximation in 1990*, in Advances in Numerical Analysis, vol. 2, W. Light (ed.), Oxford University Press, 1992.

[47] G.F. Roach, *Green's Functions*, second edition, Cambridge University Press, 1982.

[48] J.J. Royer, P.C. Vieira, *Dual Formalism of Kriging*, in Geostatistics for Natural Resources Characterization, Part 2, ed. G. Verly, M. David, A.G. Journel, A. Marechal, NATO ASI Series C: Mathematical and Physical Sciences, Vol. 122, pp. 691–702, D. Reidel Pub. Co., Dordrecht, 1984.

[49] J. Sacks, W.J. Welch, T.J. Mitchell, H.P. Wynn, *Design and Analysis of Computer Experiments*, Statistical Science, vol. 4, no. 4, pp. 409–435, 1989.

[50] D.B. Serafini, *A Framework for Managing Models in Nonlinear Optimization of Computationally Expensive Functions*, Ph.D. Thesis, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, USA, 1998.

[51] J. Søndergaard, *Non-Linear Optimization Using Space Mapping*, Master Thesis, IMM-EKS-1999-23, Department of Mathematical Modelling, Technical University of Denmark, 1999.
Available at `http://www.imm.dtu.dk/~km/jsmaster.ps.gz`

[52] V. Torczon, *On the convergence of pattern search algorithms*, SIAM Journal on Optimization, vol. 7, no. 1, pp. 1–25, 1997.

[53] V. Torczon, M.W. Trosset, *Using Approximations to Accelerate Engineering Design Optimization*, Tech. Report 98-33, Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, Virginia 23681-2199, 1998.

[54] V.V. Toropov, *Multipoint Approximation Method in Optimization Problems with Expensive Function Values*, in Computational Systems Analysis, A. Sydow (ed.), Elsevier, 1992.

[55] V.V. Toropov, *Simulation Approach to Structural Optimization*, Structural Optimization, vol. 1, pp. 37–46, 1989.

[56] V.V. Toropov, A.A. Filatov, A.A. Polynkin, *Multiparameter Structural Optimization Using FEM and Multipoint Explicit Approximations*, Structural Optimization, vol. 6, pp. 7–14, 1993.

[57] M.W. Trosset, V. Torczon, *Numerical Optimization Using Computer Experiments*, Tech. Report 97-38, Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, Virginia 23681-2199, 1997.

[58] L.N. Vicente (organizer), *School on Optimization using Surrogates for Engineering Design*, Center for Mathematics, University of Coimbra, Portugal, May 8–11, 2002.

[59] L.N. Vicente (ed.), *Second Workshop on Nonlinear Optimization: "Theoretical Aspects of Surrogate Optimization"*, abstracts, workshop held by Center for Mathematics, University of Coimbra, Portugal, May 16–17, 2002.

[60] L.N. Vicente, *Space Mapping: Models, Sensitivities and Trust-region Methods*, to appear in Optimization and Engineering, 2003.

[61] G.G. Walter, *Wavelets and Other Orthogonal Systems With Applications*, CRC Press, Inc., 1994.

[62] G.S. Watson, *Smoothing and Interpolation by Kriging and with Splines*, Mathematical Geology, vol. 16, no. 6, 1984.

[63] S. Yesilyurt, A.T. Patera, *Surrogates for Numerical Simulations; Optimization of Eddy-Promoter Heat Exchangers*, Computer Methods in Applied Mechanics and Engineering, vol. 121, pp. 231–257, 1995.

Chapter 3

An Introduction to the Space Mapping Technique

Included paper:

An Introduction to the Space Mapping Technique,
Mohamed H. Bakr, John W. Bandler,
Kaj Madsen and Jacob Søndergaard,
Optimization and Engineering, no. 2, pp. 369–384, 2001.

Optimization and Engineering, 2, 369–384, 2001

An Introduction to the Space Mapping Technique

MOHAMED H. BAKR, JOHN W. BANDLER
Simulation Optimization Systems Research Laboratory and the Department of Electrical and Computer Engineering, McMaster University, Hamilton, Ontario, Canada, L8S 4K1

KAJ MADSEN, JACOB SØNDERGAARD
Informatics and Mathematical Modelling, Technical University of Denmark, DK 2800 Lyngby, Denmark

Received March 30, 2001; Revised January 16, 2002

Abstract. The space mapping technique is intended for optimization of engineering models which involve very expensive function evaluations. It is assumed that two different models of the same physical system are available: Besides the expensive model of primary interest (denoted the fine model), access to a cheaper (coarse) model is assumed which may be less accurate.

The main idea of the space mapping technique is to use the coarse model to gain information about the fine model, and to apply this in the search for an optimal solution of the latter. Thus the technique iteratively establishes a mapping between the parameters of the two models which relate similar model responses. Having this mapping, most of the model evaluations can be directed to the fast coarse model.

In many cases this technique quickly provides an approximate optimal solution to the fine model that is sufficiently accurate for engineering purposes. Thus the space mapping technique may be considered a preprocessing technique that perhaps must be succeeded by use of classical optimization techniques. We present an automatic scheme which integrates the space mapping and classical techniques.

Keywords: non-linear optimization, space mapping, surrogate modelling

1. Introduction

When engineers encounter a mathematical problem which they cannot solve, it is common practice to consider another formulation which is solvable and intends to contribute to the original problem solution.

The space mapping technique, which was introduced by Bandler et al. (1994), is based on this principle. It is an optimization technique for engineering design in the following situation: Assume the performance of some physical object depends on a number of parameters. We search for an optimal parameter setting and during the search procedure we need to find model responses corresponding to some intermediate sets of parameters. This may for instance be based on function evaluations requested by a mathematical optimization algorithm. These evaluations are assumed to be so expensive that traditional optimization becomes unrealistic in practice. Even cases where function evaluations involve physical experiments may occur. Therefore, the aim is to make a shortcut using a cheaper, but presumably less accurate, model of the same physical system, in order to gain information about the optimal parameter setting of the original model.

Thus we assume two different models are available:

1. An accurate but expensive model, represented by a residual function $f : \Omega^{(f)} \to \mathbb{R}^m$, which must be minimized as indicated below. Here $\Omega^{(f)} \subseteq \mathbb{R}^n$, and $m \geq n$. This model is denoted the *fine model*. Gradients of f are assumed not to be available.
2. A cheap (i.e., fast) model, represented by a residual function $c : \Omega^{(c)} \to \mathbb{R}^m$, which must be minimizable in the same sense as f. Here $\Omega^{(c)} \subseteq \mathbb{R}^n$, and $m \geq n$. This model is denoted the *coarse model*. Gradients of c are assumed to be available.

In this context a residual function is the difference between a response function, originating from a model, and some predefined specifications. A response function may for instance be model responses at a specific set of sample points $t^{(j)}$, $j = 1, \ldots, m$, hence $f(x)$, $c(z)$ are vector functions with elements $f^{(j)}(x) = \varphi(t^{(j)}; x)$, $c^{(j)}(z) = \sigma(t^{(j)}; z)$ being the difference between the model response and the specification at a given sample point $t^{(j)}$. We wish to find an optimal set of parameters $x^* \in \Omega^{(f)}$ which makes the fine model response meet the specifications as well as possible, hence minimizing the fine model residual function f

$$x^* \in \arg\min_{x \in \Omega^{(f)}} H(f(x)) \tag{1}$$

with respect to some merit function H, e.g., a norm in $\mathbb{R}^m$. Since the fine model is considered too expensive for direct optimization, we want to use the coarse model to gain information about the fine model.

The general idea of how this is achieved can be illustrated by the following simple example:

Consider an archery contest, and assume for simplicity that the archer has a steady hand: he always shoots in exactly that direction he has planned. The goal of course is to hit the bull's-eye y^*, hence y^* represents the given set of specifications. The shooting situation is simulated with a coarse model which hits the spot the archer is pointing at, not taking forces like wind and gravity into account.

We represent the points y in the target plane as vectors in $\mathbb{R}^2$. The coarse objective function is a vector function $c : \Omega^{(c)} \to \mathbb{R}^2$, where $\Omega^{(c)} \subseteq \mathbb{R}^2$ is the set of possible directions from the archer to the target. Let $z \in \Omega^{(c)}$ be a direction pointing to the spot $y^{(c)}$ at the target. Then the objective $c(z)$ is the difference between $y^{(c)}$ and the target, i.e., $c(z) = y^{(c)} - y^*$. The fine model is a representation of the actual shot towards the target, i.e., in this case the fine model represents physical experiments. The fine objective function is a vector function $f : \Omega^{(f)} \to \mathbb{R}^2$, $\Omega^{(f)} \subseteq \mathbb{R}^2$. For a direction $x \in \Omega^{(f)}$ the objective $f(x)$ is the difference between the spot $y^{(f)}$ at the target which is hit and the target, i.e., $f(x) = y^{(f)} - y^*$. We wish to find a direction $x^* \in \Omega^{(f)}$ such that $\| f(x^*) \| = 0$.

At first the archer aims at y^*, i.e., he optimizes the coarse model by finding the direction $z^* \in \Omega^{(c)}$ which points at y^*. This can be formulated as follows,

$$z^* = \arg\min_{z \in \Omega^{(c)}} H(c(z)) \tag{2}$$

for some norm H. In this case $\|c(z^*)\| = 0$.

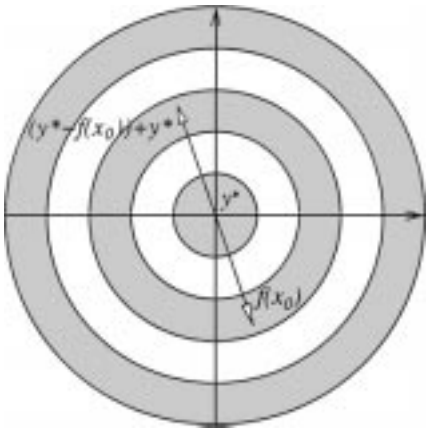


Figure 1. "Calculation" of the first shot $f(x_0)$, $x_0 = z^*$. In the next shot the archer will aim at $(y^* - f(x_0)) + y^*$.

After taking aim the archer fires the shot in the direction chosen, hence we "calculate" $f(x_0)$ for $x_0 = z^*$, as illustrated in figure 1.

Since the coarse model does not take the influence of wind and gravity into account, the arrow may fail to hit y^*, which in mathematical terms means that $x^* \neq z^*$.

After failing a shot any good archer would adjust the sight in order to obtain a better result with the next shot. The natural adjustment would be to "mirror the error". If, for instance, the first shot has hit too low on the right side of y^*, then the next aim should be directly opposite: upwards on the left side of y^*. In our notation the second shot would aim at $(y^* - f(x_0)) + y^*$. Thus, if we let z_0 be the direction which points at $f(x_0)$ (i.e., $c(z_0) = f(x_0)$), then $c(z^*) = y^*$ implies that the direction of the second shot becomes $x_1 = (z^* - z_0) + z^*$. Since $x_0 = z^*$ this is the same as the tentative iterate $\tilde{x}_1$ suggested by the first space mapping iteration (see (7) *and* (8) *below where $B_0 = I$).*

Essentially this way of a coarse model interacting with a fine model (or as here: the physical reality) has been used in engineering practice for decades.

The idea of the space mapping technique is to establish a connection between the coarse and the fine models, through a parameter mapping, and to utilize this mapping for finding an optimal set of parameters for the fine model. In other words we are interested in establishing a parameter mapping $p : \Omega^{(f)} \to \Omega^{(c)}$ which yields an approximation of the form

$$f(x) \simeq c(p(x)), \tag{3}$$

where the mapping function p relates similar responses in the following sense: For $x \in \Omega^{(f)}$ we obtain $z = p(x) \in \Omega^{(c)}$ as a solution to the subproblem

$$z \in \arg \min_{\hat{z} \in \Omega^{(c)}} \| f(x) - c(\hat{z}) \|, \tag{4}$$

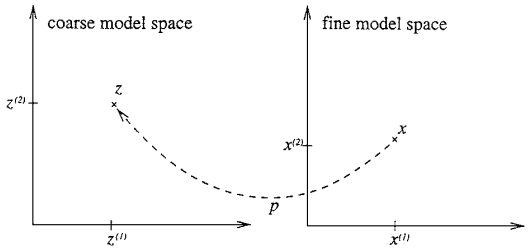


Figure 2. The mapping function relating the fine and the coarse model spaces, shown here for the two-dimensional case, $[z^{(1)}\ z^{(2)}]^T = p([x^{(1)}\ x^{(2)}]^T)$.

for some specific norm. In the present paper we assume that this optimal solution is unique. For the problem of multiple solutions we refer to Bakr et al. (2000b). The concept of the mapping function is illustrated in figure 2 for the two-dimensional case.

If the approximation (3) is close then the composite function $c \circ p$ is applicable as a surrogate for f. Hence the optimal solution of $c \circ p$ can be expected to be close to the optimal solution of f. In other words we might optimize $c \circ p$ rather than f which is expected to be easier under the condition that c and f have similar structures: Then we expect p to be a well behaved function, and since c is cheap to calculate, the composite function $c \circ p$ may be easier to optimize than f. This way of replacing f by $c \circ p$ is the basis of the space mapping technique.

Note for the subproblem (4) that for a given x, a calculation of $p(x)$ involves one evaluation of f succeeded by an optimization in the coarse model space $\Omega^{(c)}$. Hence an evaluation of the mapping function is at least as expensive as an evaluation of the fine model.

The space mapping technique assumes the two models are related in such a way that (3) is a close approximation. Hence $c \circ p$ is optimized in the effort of finding a solution to (1) and for this we apply classical optimization techniques. The problem formulation is

$$\bar{x} \in \arg \min_{x \in \Omega^{(f)}} H(c(p(x))), \tag{5}$$

where $\bar{x}$ may be close to x^* if $c \circ p$ is close to f. Observe that if the optimal solution z^* of $H \circ c$ is unique then the solution of (5) is equivalent of solving the system of n non-linear equations

$$p(x) = z^* \tag{6}$$

for x. In other words $\bar{x} = p^{-1}(z^*)$.

In the first space mapping paper Bandler et al. (1994) estimate the mapping p on the basis of some predefined weighted fundamental functions and evaluations of f at a selected set of base points in $\Omega^{(f)}$. Bandler et al. (1995) formulated the problem as solving (6) for x using Broyden's method for non-linear equations (Broyden, 1965). Bakr et al. (1998) introduced a trust-region methodology to enhance the global convergence properties. The details of

these different approaches are described in the review paper by Bakr et al. (2000b). Recent results of combining space mapping and direct optimization in the field of microwave circuit design are described in Bakr et al. (2000a)

2. Space mapping details

In our formulation the space mapping intends to solve (5) by iteration. At the kth iteration the mapping function p as defined in (4) is replaced by a local estimate p_k, and then the optimal solution of $H \circ c \circ p_k$ is the next iterate. The question is how to find a good approximation p_k. In this presentation we choose to iteratively approximate p by a first order approximation, with the Jacobian matrix approximated using Broyden's rank one update formula.

Let the kth iterate be x_k and assume $z_k = p(x_k)$ has been found by (4). Letting the kth Jacobian approximation be B_k, the corresponding linearization is

$$p_k(x) = B_k(x - x_k) + z_k. \tag{7}$$

The $(k+1)$th tentative iterate is:

$$\tilde{x}_{k+1} \in \arg \min_{x \in \Omega^{(f)}} H(c(p_k(x))). \tag{8}$$

In case of multiple optimal solutions we choose the one having the shortest distance to the previous iterate x_k. If $H(f(\tilde{x}_{k+1})) < H(f(x_k))$ then the next iterate x_{k+1} is chosen as $\tilde{x}_{k+1}$, otherwise $x_{k+1} = x_k$.

Now $\tilde{z}_{k+1} = p(\tilde{x}_{k+1})$ is found by (4) and finally the Jacobian approximation is updated by Broyden's formula:

$$B_{k+1} = B_k + \frac{\tilde{z}_{k+1} - z_k - B_k h_k}{h_k^T h_k} h_k^T, \tag{9}$$

where $h_k = \tilde{x}_{k+1} - x_k$. Notice that the update is always performed, independently of the acceptance of the tentative point $\tilde{x}_{k+1}$.

Initially the optimal solution z^* of $H \circ c$ is found and used as the first iterate: $x_0 = z^*$. This can be interpreted as an assumption that p is close to the identity mapping:

$$f(x) \simeq c(p(x)) \simeq c(Ix) \tag{10}$$

where $I = I(n)$. It corresponds to the initial aim at the bull's eye in the archery example of the previous section.

The motivation for the initial choice of the Jacobian approximation is another intuition used in the archery example: To mirror the error. This intuition is based on the assumption that the difference between the two model functions is close to a parameter translation:

$$f(x) \simeq c(p(x)) \simeq c(Ix + C_0) \tag{11}$$

where C_0 is a constant, i.e., $p_1(x) = Ix + C_0$. Since $p(x_0) = z_0$ we obtain $C_0 = z_0 - Ix_0$, and thus (11) suggests that $p_0(x)$ is given by (7) with $k = 0$ and $B_0 = I(n)$. Hence the traditional choice of B_0 in Broyden's method is motivated by the archer's simplification (11).

The validity of the mapping approximation p_k is confined to a trust region of size δ_k, hence the feasible set at iteration k is

$$x \in \Omega^{(p_k)} \equiv \{\tilde{x} \mid \|\tilde{x} - x_k\| \leq \delta_k\} \cap \Omega^{(f)}, \tag{12}$$

for some specific norm, thus (8) is replaced by

$$\tilde{x}_{k+1} \in \arg \min_{x \in \Omega^{(p_k)}} H(c(p_k(x)). \tag{13}$$

The update of the trust region size δ_k follows the classical scheme: Significant improvement in the objective compared to the predicted improvement by the approximation is rewarded by enlarging the trust region, whereas insufficient improvement leads to decreasing the trust region size, see Moré (1982) for a thorough treatment of this subject.

For many engineering purposes this formulation yields sufficiently accurate results. However, the convergence of the approach depends on the similarity between the two models. Now, assume the sequence $\{x_k\}$ generated using (13) converges to the solution $\bar{x}$ of (6), then $z^* = p(\bar{x})$. If $\bar{x} = x^*$ then $z^* = p(x^*)$; if, however, the response of the coarse model is less accurate than that of the fine model then we cannot expect z^* and x^* to correspond. Hence in general we must expect $\bar{x} \neq x^*$.

In case of convergence the typical performance we have noticed is a decrease of $\|x_k - x^*\|$ as long as this distance is of a larger order of magnitude than $\|\bar{x} - x^*\|$. Finally, as x_k approaches $\bar{x}$, $\|x_k - x^*\|$ starts to increase.

This observation indicates that the space mapping technique may be considered a good preprocessing process, but not a method for obtaining an accurate solution. If the latter is required then another (i.e., locally convergent) method of optimization will be necessary in the final stages. A switch of method should ideally take place when the distance $\|x_k - x^*\|$ has reached the same order of magnitude as $\|\bar{x} - x^*\|$. The combined strategy of the following section represents some early attempts to reach this ideal goal.

3. Combining with classical methods

This section demonstrates how the space mapping technique can be combined with classical methods of optimization, based on local Taylor type approximations.

Assume the space mapping technique has been used for a number of iterations. Hence a number of fine model evaluations $f(x_k)$ have been calculated. On the basis of these we build an approximation of the Jacobian of f using, for instance, Broyden's formula:

$$D_{k+1} = D_k + \frac{f(x_{k+1}) - f(x_k) - D_k h_k}{h_k^T h_k} h_k^T, \tag{14}$$

where $h_k = x_{k+1} - x_k$. The initial Jacobian approximation is related to the Jacobian of the mapped coarse model at x_0:

$$\begin{aligned} D_0 &= \nabla_{x=x_0}[c(p(x))] = \nabla_{z=z_0}[c(z)] \cdot \nabla_{x=x_0} p(x) \\ &\approx \nabla_{z=z_0}[c(z)] \cdot \nabla_{x=x_0} p_1(x) = \nabla_{z=z_0}[c(z)] \end{aligned} \tag{15}$$

where the "$\approx$" is probably not very precise but in accordance with the intuition (11) used when we start the space mapping. This yields a local linearization of the fine model

$$l_k(x) = D_k(x - x_k) + f(x_k). \tag{16}$$

Traditionally we would minimize $H \circ f$ iteratively using (16) as a basis for finding the $(k+1)$th tentative iterate:

$$\tilde{x}_{k+1} \in \arg \min_{x \in \Omega^{(l_k)}} H(l_k(x)), \tag{17}$$

where $\Omega^{(l_k)}$ is some trust region to be updated during the iteration. The next iterate is $x_{k+1} = \tilde{x}_{k+1}$ if the objective $H \circ f$ is improved, otherwise $x_{k+1} = x_k$. Under mild conditions this iteration yields convergence to a stationary point x^* (see (1)) of f, see e.g., Madsen (1986).

In the present context we use a combination of (16) and the space mapping model $c \circ p_k$ of f: At the kth iteration the combined surrogate for f is

$$s_k(x) = \omega_k \cdot c(p_k(x)) + (1 - \omega_k) \cdot l_k(x), \tag{18}$$

where $\omega_k \in [0; 1]$. Thus the $(k+1)$th tentative iterate is:

$$\tilde{x}_{k+1} \in \arg \min_{x \in \Omega^{(s_k)}} H(s_k(x)). \tag{19}$$

where $\Omega^{(s_k)}$ is a trust region to be updated during the iteration. In case of multiple solutions we choose the one closest to x_k. The next iterate is $x_{k+1} = \tilde{x}_{k+1}$ if the objective $H \circ f$ is improved, otherwise $x_{k+1} = x_k$.

The intention is to use the space mapping surrogate initially (i.e., $\omega_k = 1$) and the local approximation (i.e., $\omega_k = 0$) in the final stages of the iteration. Hence the weighting factor ω_k can be used in a transition from the space mapping surrogate $c \circ p_k$ to a local linearization l_k.

We expect the usefulness of the linear model to increase as the iteration approaches the optimal solution of f. On the other hand, we expect $c \circ p_k$ to be insubstantial in describing f accurately in the vicinity of the optimal solution. Hence we would like to use the information given in the coarse model at the initial stages of the iterations, and as we approach the optimal solution we would like to do a direct optimization, by having the linear model l_k dominate s_k.

In general, we do not wish to change the value of ω if the steps produced by the space mapping algorithm yield a sufficient reduction in the objective function $H \circ f$.

A very simple method of updating ω_k which fulfills these conditions is to define $\omega_{k+1} = \omega_k$ if the objective has been improved, and $\omega_{k+1} = \omega_k/2$ otherwise. More sophisticated

updating strategies are currently being investigated. Some suggestions are found in Bakr (2000), Bakr et al. (2000a), and Søndergaard (1999). The challenge is to find a good combination of the trust region radius update and the ω_k update.

4. Examples

Example 1a. To illustrate the space mapping method we consider the design of a two-section capacitively-loaded 10 : 1 impedance transformer. The coarse and the fine models are shown in figure 3. Assume that the fine model is very expensive and is not recommended for direct optimization. The values of the fine model capacitances are given in Table 1. The characteristic impedances are kept fixed at the optimal values given in Table 1. The physical lengths L_1 and L_2 of the two transmission lines are selected as designable parameters. Eleven frequency points are simulated per sweep. We consider the input reflection coefficient response $f^{(j)}(x) = |S_{11}(t^{(j)}; x)|$ (notice that $S_{11}(t^{(j)}; x) > 0$ for all x) of both models which is a function of the real frequency t and the designable parameters $x = [L_1\ L_2]^T$.

The design specifications are $|S_{11}(t^{(j)}; x)| \leq 0.50$ for the frequency interval $t \in [0.5; 1.5]$ GHz. Hence we wish to find a design $x = x^*$ of the fine model yielding

$$H(f(x)) \equiv \max_j \{ f^{(j)}(x) \} \leq 0.50. \tag{20}$$

In the following we review some results of applying the combined method (18) on this problem.

Table 1. The fine model capacitances, and the characteristic impedances for the two-section capacitively-loaded impedance transformer.

Capacitance	Value (pF)	Impedance	Value (ohm)
C_1	10	Z_1	4.47214
C_2	10	Z_2	2.23607
C_3	10		

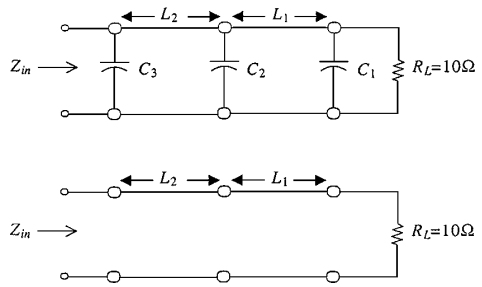


Figure 3. Fine and coarse model, two-section capacitively-loaded impedance transformer.

Table 2. The optimal coarse and fine model parameters z^* and x^* (physical lengths of the transmission lines) for the two-section capacitively-loaded impedance transformer.

z^* (m)	x^* (m)
0.01724138	0.06186103
0.01724138	0.06605482

Given the optimal coarse model parameters z^* (in Table 2), initially we let $x_0 = z^*$, figure 4 shows the fine model response $f(x_0)$. The figure illustrates how the initial fine model design at x_0 violates the specifications (20). Solving the subproblem (4) we find $z_0 = p(x_0)$, such that $c(z_0)$ (also shown in figure 4) is close to $f(x_0)$.

After the first iteration x_1 is found using (19) and from figure 5 we note how the fine model response $f(x_1)$ meets the specifications. For the engineering purpose of finding a design satisfying the specifications (20) a result like this is sufficient. Until this stage the algorithm has used two fine model evaluations.

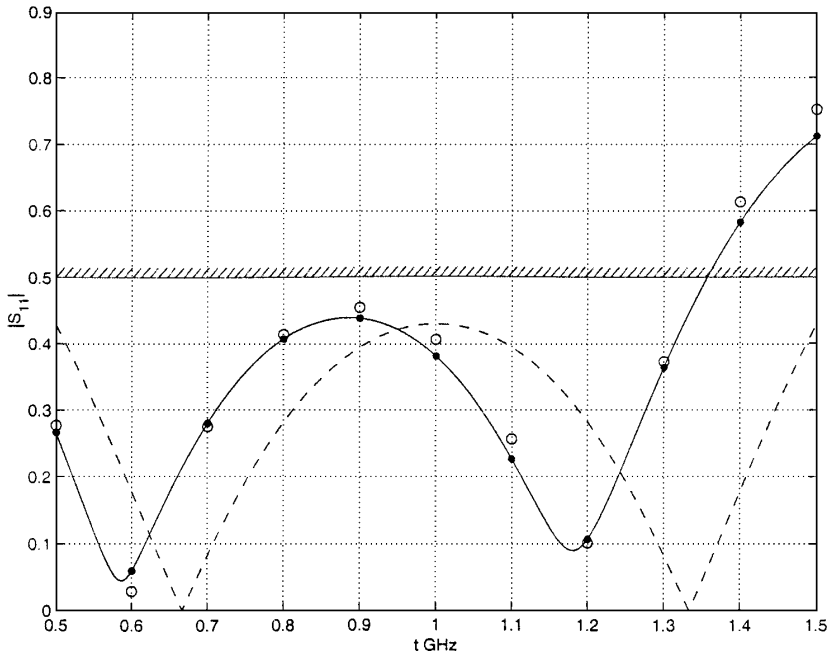


Figure 4. Two-section capacitively-loaded impedance transformer: The fine model response $f(x_0)$ ($\circ$) at the coarse model optimal solution $x_0 = z^*$ and the coarse model response $c(z_0)$ (—•—), $z_0 = p(x_0)$. The dashed curve is the optimal coarse model response $c(z^*)$ which the mapped coarse model $c \circ p$ is aiming for, see (5).

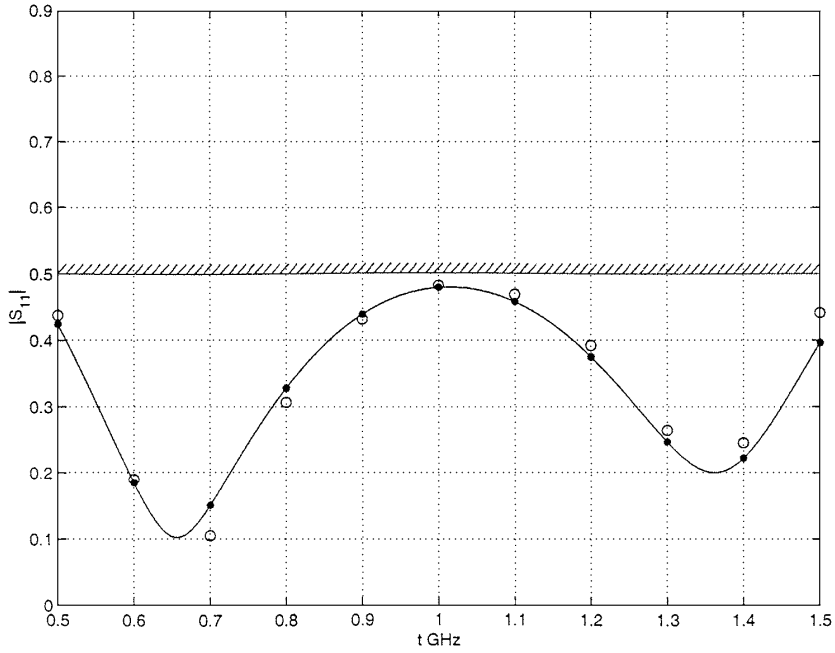


Figure 5. Two-section capacitively-loaded impedance transformer: The fine model response $f(x_1)$ ($\circ$) and the coarse model response $c(z_1)$ (—•—), $z_1 = p(x_1)$.

The visual difference from the fine model design at x_1 to the optimal design x^* (given in Table 2) is rather small: figures 5 and 6 show that from the first iteration to the solution the objective is decreased only from $H(f(x_1)) = 0.481$ to $H(f(x^*)) = 0.455$. It turns out that the distance between x_1 and the solution x^* is so small that the coarse model is unable to provide sufficient improvements after x_1 (in accordance with the argument at the end of Section 2). Hence the algorithm switches rapidly to the local linear model which—in the near neighbourhood of an iterate x_k—is more accurate than the mapped coarse model. The fact that the local linear model is preferable when only small steps are needed is illustrated in Example 1b.

Example 1b. Using the same problem, we here give a graphical illustration of how the mapped coarse model approximation $c \circ p_k$ is a valid approximation to f in a larger region than a linearization l_k of f. The following point is to be made: When large steps are needed then the mapped coarse model approximation is the better, and when small steps are needed (e.g., when we are close to x^*) then the linearization l_k is the better. In order to make the argument more clear we insert accurate Jacobian approximations, B_k to $p'(x_k)$ in (7), and D_k to $f'(x_k)$ in (16) (these approximations being found using finite differences).

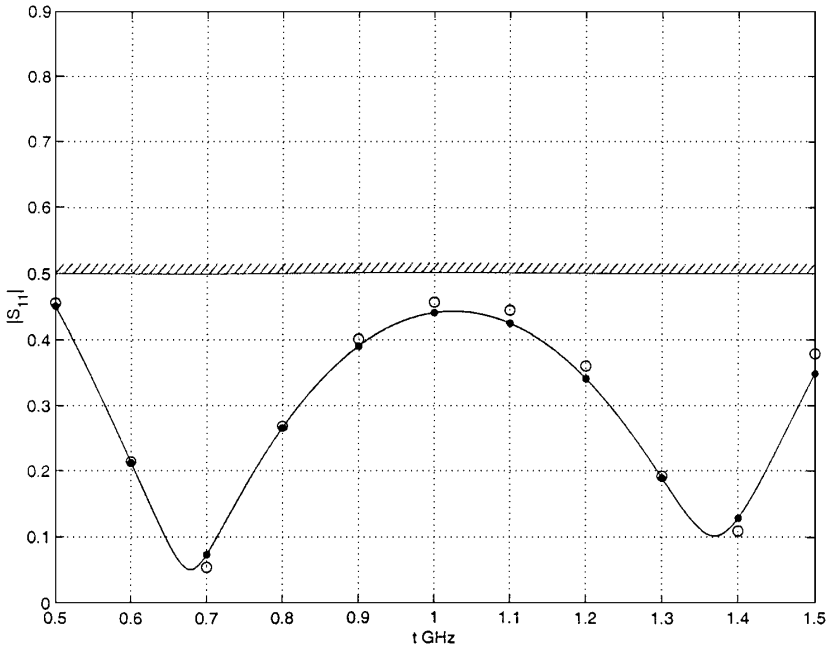


Figure 6. Two-section capacitively-loaded impedance transformer: The fine model response $f(x^*)$ (○) and the coarse model response $c(p(x^*))$ (—●—).

In figure 7 the mapped coarse model approximation error $\|c(p_k(x)) - f(x)\|_2$ is plotted for points on a mesh in a square region centered at x_k. The linearized fine model approximation error $\|l_k(x) - f(x)\|_2$ is plotted for points at the same mesh. The figure illustrates, as expected, how the approximation error of the linear approximation l_k (which is zero at x_k) grows with the square of the distance from x_k. The approximation error of the mapped coarse model $c \circ p_k$, however, does not grow systematically with the distance from x_k, in fact it is almost constant in the region considered. Furthermore we note that $c \circ p_k$ does not interpolate f, i.e., $\|c(p_k(x)) - f(x)\|_2$ is non-zero at $x = x_k$.

From these observations we conclude that close to x_k the better approximation to f is l_k, whereas $c \circ p_k$ is the better away from x_k. In fact $c \circ p_k$ is a valid approximation to f in the whole region shown in figure 7.

Example 2. In this example we consider the design of a seven-section capacitively-loaded impedance transformer. The load impedance is 100 Ω and the line impedance is 50 Ω. The coarse and the fine models are shown in figure 8. The values of the fine model capacitances are given in Table 3. The characteristic impedances are synthesized using an equi-ripple approximate design procedure (Pozar, 1998) and are kept fixed at these values given in Table 3. The physical lengths L_i, $i = 1, \ldots, 7$, of the seven transmission lines are selected

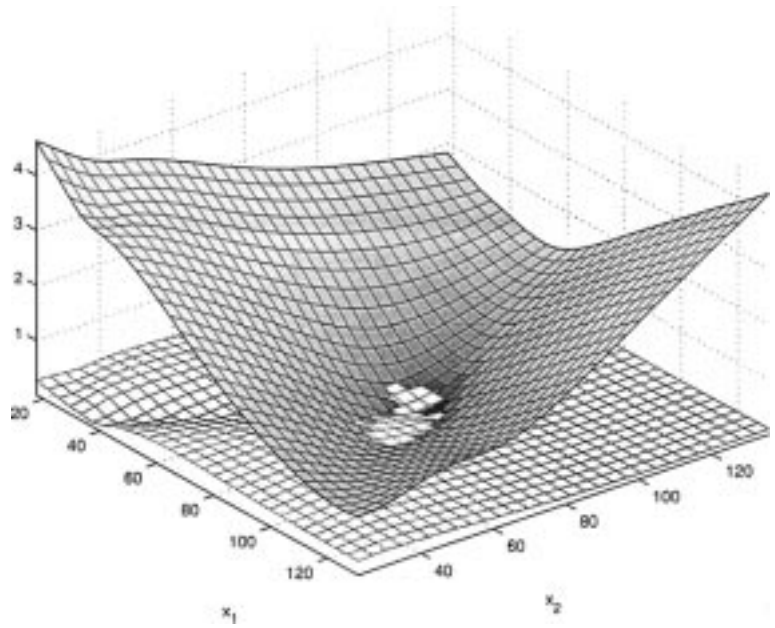


Figure 7. Two-section capacitively-loaded impedance transformer: Mapped coarse model approximation error $\|c(B_k(x-x_k)+p(x_k))-f(x)\|_2$ (white mesh), linearized fine model approximation error $\|D_k(x-x_k)+f(x_k)-f(x)\|_2$ (gray scale mesh). For both meshes: x_k the point of linearization is in the center of the plot.

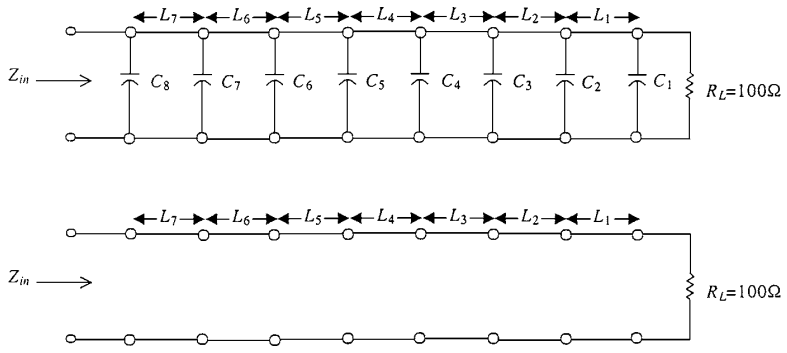


Figure 8. Fine and coarse model, seven-section capacitively-loaded impedance transformer.

Table 3. The fine model capacitances, and the characteristic impedances for the seven-section capacitively-loaded impedance transformer.

Capacitance	Value (pF)	Impedance	Value (ohm)
C_1	0.025	Z_1	91.9445
C_2	0.025	Z_2	85.5239
C_3	0.025	Z_3	78.1526
C_4	0.025	Z_4	70.7107
C_5	0.025	Z_5	63.9774
C_6	0.025	Z_6	58.4632
C_7	0.025	Z_7	54.3806
C_8	0.025		

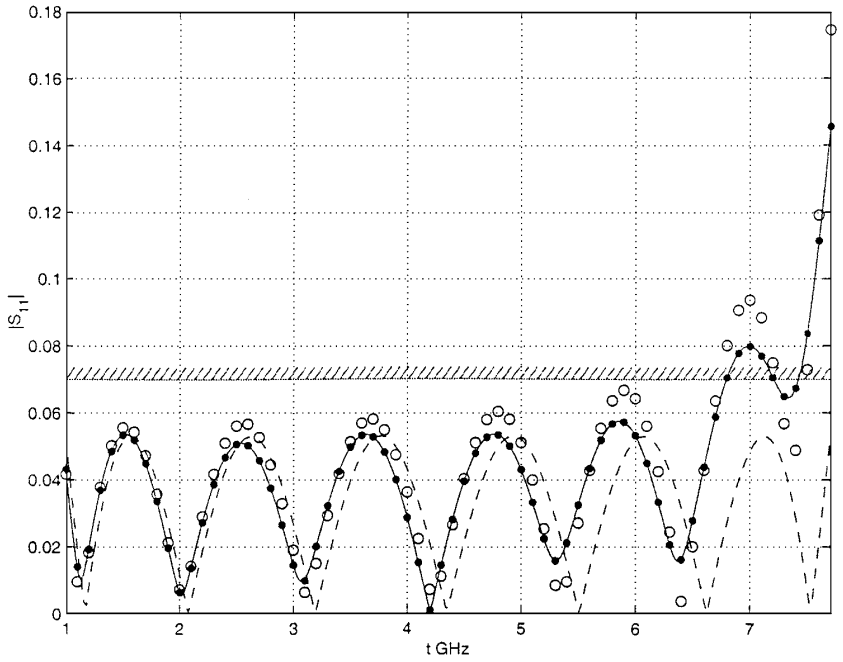


Figure 9. Seven-section capacitively-loaded impedance transformer: The fine model response $f(x_0)$ ($\circ$) at the coarse model optimal solution $x_0 = z^*$ and the coarse model response $c(z_0)$ (—•—), $z_0 = p(x_0)$. The dashed curve is the optimal coarse model response $c(z^*)$ which the mapped coarse model $c \circ p$ is aiming for, see (5).

as designable parameters. We consider the input reflection coefficient response $f^{(j)}(x) = |S_{11}(t^{(j)}; x)|$, and the design specifications are $|S_{11}(t^{(j)}; x)| \leq 0.07$ for the frequency interval $t \in [1; 7.7]$ GHz.

In figure 9 the fine model response is plotted at the optimal design of the coarse model, $x_0 = z^*$. The coarse model response at the design z_0 (being the design at which the coarse model response is closest to the fine model response $f(x_0)$) is also plotted in figure 9. It is seen that this coarse model response is not very accurate in describing the fine model response indicating that the correspondence between the two models is less obvious in this case.

In figure 10 the optimal fine model response is plotted together with the closest coarse model response. We see how the coarse model poorly describes the fine model at this design, in this case the space mapping algorithm is depending heavily on the classical method to be able to converge to the optimal solution (not another local minimum). The optimal coarse and fine model parameters are given in Table 4.

A feasible solution is found after 18 fine model evaluations. At this stage the combination parameter ω_k of (18) has been downdated to 0.016, so the space mapping is almost abandoned

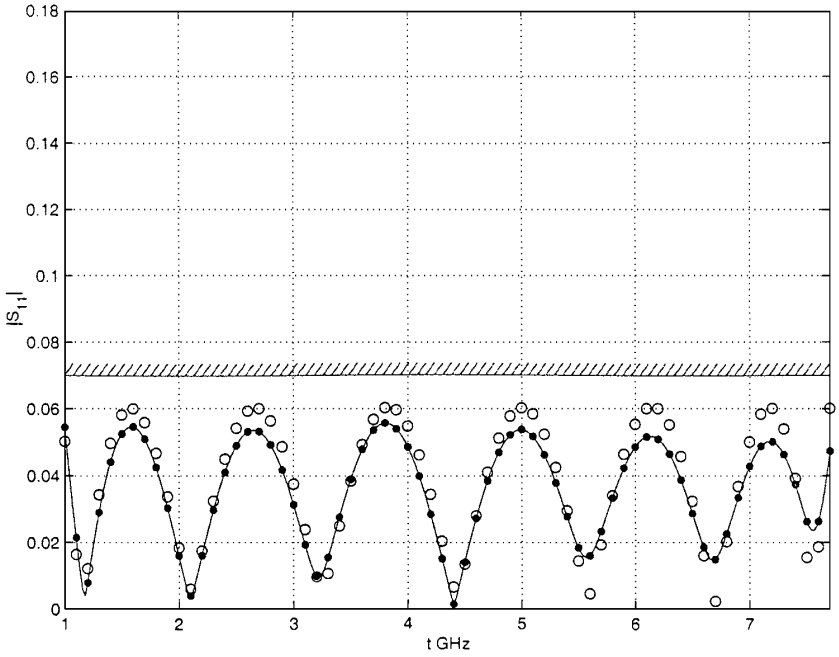


Figure 10. Seven-section capacitively-loaded impedance transformer: The fine model response $f(x^*)$ (○) and the coarse model response $c(p(x^*))$ (—●—).

Table 4. The optimal coarse and fine model parameters z^* and x^* (physical lengths of the transmission lines) for the seven-section capacitively-loaded impedance transformer.

z^* (m)	x^* (m)
0.01724138	0.01564205
0.01724138	0.01638347
0.01724138	0.01677145
0.01724138	0.01697807
0.01724138	0.01709879
0.01724138	0.01723238
0.01724138	0.01625988

from this stage, i.e., the rest of the iterations are practically speaking based on the local linear model l_k.

For comparison we have solved this problem directly using an implementation of the minimax optimization method of Hald and Madsen (1981) with finite differences to approximate the fine model Jacobians. As initial iterate we use the coarse model optimal solution, i.e., $x_0 = z^*$. This way we find a feasible solution after 25 fine model evaluations.

5. Conclusions

The basic principles of the space mapping technique have been presented. It is shown how the space mapping technique can be combined with classical optimization strategies. The combined method is illustrated by a simple two-dimesional example and a more complicated seven-dimensional example. The space mapping surrogate is shown by example to be a valid approximation to the fine model in a larger region than a corresponding fine model linearization using the same number of fine model evaluations.

The space mapping has proved to be an efficient preprocessing technique in many difficult engineering optimization problems. The solution accuracy is often sufficient for practical purposes. Otherwise the technique can be combined with other methods of optimization.

Acknowledgments

The authors would like to thank José Rayes-Sanchez and A. S. Mohamed of McMaster University. The work in Canada was supported by NSERC under Grants OGP0007239 and STR234854-00, and by Bandler Corporation.

References

M. H. Bakr, "Advances in space mapping optimization of microwave circuits," Ph.D. Thesis, McMaster University, 2000. Available at `http://www.sos.mcmaster.ca/theses.htm`

M. H. Bakr, J. W. Bandler, R. M. Biernacki, S. H. Chen, and K. Madsen, "A trust region aggressive space mapping algorithm for EM optimization," *IEEE Trans. Microwave Theory Tech*. vol. 46, no. 12, pp. 2412–2425, 1998.

M. H. Bakr, J. W. Bandler, K. Madsen, J. E. Rayas-Sánchez, and J. Søndergaard, "Space mapping optimization of microwave circuits exploiting surrogate models," *IEEE Trans. Microwave Theory Tech*. vol. 48, pp. 2297–2306, 2000a.

M. H. Bakr, J. W. Bandler, K. Madsen, and J. Søndergaard, "Review of the space mapping approach to engineering optimization and modeling," *Optimization and Engineering* vol. 1, no. 3, pp. 241–276, 2000b.

J. W. Bandler, R. M. Biernacki, S. H. Chen, P. A. Grobelny, and R. H. Hemmers, "Space mapping technique for electromagnetic optimization," *IEEE Trans. Microwave Theory Tech*. vol 42, pp. 2536–2544, 1994.

J. W. Bandler, R. M. Biernacki, S. H. Chen, R. H. Hemmers, and K. Madsen, "Electromagnetic optimization exploiting aggressive space mapping," *IEEE Trans. Microwave Theory Tech*. vol. 43, no. 12, pp. 2874–2882, 1995.

C. G. Broyden, "A class of methods for solving non-linear simultaneous equations," *Math. Comp*. vol. 19, pp. 577–593, 1965.

J. Hald and K. Madsen, "Combined LP and quasi-Newton methods for minimax optimization," *Mathematical Programming* vol. 20, pp. 49–62, 1981.

K. Madsen, "Minimax solution of non-linear equations without calculating derivatives," *Mathematical Programming Study* vol. 3. pp. 110–126, 1975.

K. Madsen, "Minimization of approximation functions," dr. techn. thesis, Technical University of Denmark, 1986.

J. J. Moré, "Recent developments in algorithms and software for trust region methods," *Mathematical Programming: The State of The Art*, pp. 258–287, 1982.

D. M. Pozar, *Microwave Engineering*, 2nd edn., John Wiley: New York, 1998.

J. Søndergaard, "Non-linear optimization using space mapping," Master Thesis IMM-EKS-1999-23 at IMM, DTU, 1999. Available at `http://www.imm.dtu.dk/~km/jsmaster.ps.gz`

Chapter 4

Space Mapping Theory and Practice

The space mapping technique was introduced by Bandler et al. in 1994 [6]. The technique relies on a parameter mapping, the so-called space mapping, between the parameter spaces of two independent models, denoted the fine model and the coarse model. This space mapping aligns the parameter spaces of the fine and the coarse model, such that a combination of the coarse model and the space mapping can serve as a surrogate for the fine model.

One should discern between space mapping for modelling and space mapping for optimization. In the first case the purpose is to obtain a surrogate which is close, i.e. a small residual measured in some norm, to the fine model over a large part of the parameter space. With space mapping for optimization the purpose is to use the surrogate to obtain the optimizer of the fine model, only scarcely evaluating the latter. The focus of this presentation is space mapping for optimization.

In order to make a successful surrogate the space mapping must meet certain conditions. We propose a set of conditions for which the minimizer of the fine model can be found using the space mapping surrogate.

The actual definition of the space mapping is not uniquely determined by theoretical conditions. In fact, there is a great deal of freedom in choosing how to define the space mapping. However, meeting the theoretical conditions

in a practical definition is not trivial. There has yet to be proposed a definition of the space mapping which is robust in most practical situations.

There has been established consensus about a certain way of defining of the space mapping, mapping similar model responses, which we denote the usual space mapping definition. We illustrate some situations where this usual definition of the space mapping fails, and show how other definitions of the space mapping may have more tractable properties.

We start this chapter by Section 4.1 which presents some theory about space mapping. First in the section, the mathematical notation is defined and a motivating example is introduced. Thereafter, some theoretical results are derived about the space mapping under certain ideal conditions. As a special case the usual definition of the space mapping is considered. In Section 4.2 follows a discussion about four alternative space mapping definitions, related to the theory and observations from numerical test problems. In Section 4.3 the approximation error of the coarse model composed with the space mapping is treated both theoretically and for a specific numerical test problem. We end this chapter by summarizing the conclusions in Section 4.4.

4.1 Space Mapping Theory

4.1.1 Theoretical Introduction

Throughout the chapter we apply the following general assumptions: Two models are available, namely the fine model and the coarse model. The fine model is represented by the response function $f : \mathbb{R}^n \mapsto \mathbb{R}^m$, with $f = (f_1, \ldots, f_m)^T$. The coarse model is represented by the response function $c : \mathbb{R}^n \mapsto \mathbb{R}^m$ with $c = (c_1, \ldots, c_m)^T$. The functions f and c are assumed to be continuously differentiable. The response functions are measured using the merit function $H : \mathbb{R}^m \mapsto \mathbb{R}$ which is a convex function, e.g. a norm.

We assume the existence of a function $p : \mathbb{R}^n \mapsto \mathbb{R}^n$, the space mapping, which by its usual definition aims to relate similar responses of f and c. We will consider several different definitions of p throughout this chapter.

We now introduce four important sets of minimizers, X^*, Z^*, X_p^* and X_{cop}^*.

The main problem we intend to solve using space mapping optimization is finding a fine model minimizer. The set of all *fine model minimizers* is defined

by

$$X^* \equiv \arg\min_{x \in \mathbb{R}^n} H(f(x)) \, .$$

For the similar problem with the coarse model, we define the set of all *coarse model minimizers*

$$Z^* \equiv \arg\min_{z \in \mathbb{R}^n} H(c(z)) \, .$$

The first application of space mapping uses what we call the *original space mapping technique*, see [6, 7], which is the problem of solving the n nonlinear equations

$$p(x) = z^* \tag{4.1}$$

for $x \in \mathbb{R}^n$, with $z^* \in Z^*$. Often (4.1) is solved using the least-squares formulation,

$$\min_{x \in \mathbb{R}^n} \|p(x) - z^*\|_2 \, . \tag{4.2}$$

The problem (4.2) can be stated in a more general setting, as finding those points, denoted X_p^*, in the space mapping image $\{p(\mathbb{R}^n)\}$ which are closest to the set of all coarse model minimizers,

$$X_p^* \equiv \arg\min_{x \in \mathbb{R}^n} d(p(x), Z^*) \, , \tag{4.3}$$

where $d(u, V)$ is the Euclidean distance from the point u to the set V,

$$d(u, V) = \inf_{v \in V} \|u - v\|_2 \, .$$

The set X_p^* is denoted as the set of all *space mapping solutions*.

Recently, in [11], a new formulation of the problem in (4.1) was proposed, namely to minimize the so-called *space mapped coarse model* $c(p(x))$. We define the *space mapped coarse model minimizers*

$$X_{c \circ p}^* \equiv \arg\min_{x \in \mathbb{R}^n} H(c(p(x))) \, , \tag{4.4}$$

We show later that the space mapping solutions (4.3) and the space mapped coarse model minimizers (4.4) are the same if the Conditions **C1** and **C3**, defined below, are satisfied.

We now define four conditions for the space mapping. The conditions can not always be expected to be satisfied in practice, however they are essential for the theoretical understanding of the space mapping technique. We are not at this point assuming that these conditions are satisfied.

C1 $Z^* \subseteq p(\mathbb{R}^n)$

C2 $Z^* \subseteq p(X^*)$

C3 p is one-to-one

C4 X^* and Z^* are singletons

The following is our interpretation of these conditions.

The Condition **C1** states that the set of all coarse model optimizers Z^* is the image of the space mapping. Hence all coarse model minimizers can be reached through the space mapping.

Condition **C1** is implied by the more strict Condition **C2**, which requires Z^* to be in the space mapping image $p(X^*)$. Condition **C2** is a generalization of the *perfect mapping* assumption, introduced in [11], namely that

$$p(x^*) = z^* \,, \tag{4.5}$$

when **C4** holds, which states that the minimizers of $H(f(x))$ and $H(c(z))$ are unique, i.e. $X^* = \{x^*\}$ and $Z^* = \{z^*\}$.

A two-dimensional conceptual illustration of the above conditions is provided by Figure 4.1. The figure shows the case where **C1** and **C4** hold, so the unique minimizer z^* is in the image of the space mapping of the fine model parameter space. If further **C2** and **C3** hold, then x^* and z^* are directly related through the space mapping as in (4.5).

To supplement this abstract interpretation we now consider an example which illustrate the above conditions on practical test problems. The example will also motivate the theory, which is presented right after the example.

4.1.2 Example: Space Mapping Images

We now introduce an example to illustrate the Conditions **C1**–**C4** introduced above and to motivate the theory we develop next. The example consists of

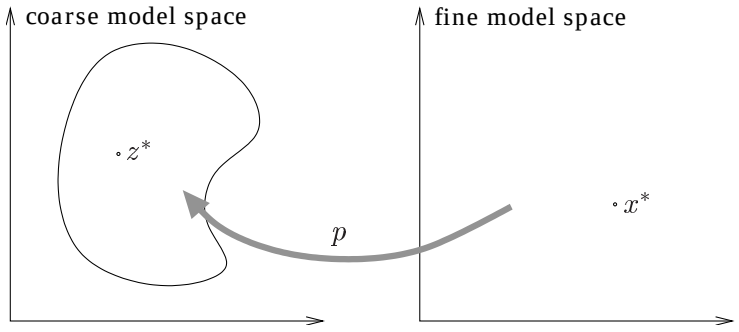


Figure 4.1: Conceptual illustration of the space mapping image of a two-dimensional fine model space into a coarse model space. Here the unique coarse model optimizer z^* is in the mapping image, i.e. $z^* \in p(x)$ for $x \in \mathbb{R}^n$, corresponding to a case where **C1** *and* **C4** *hold.*

plots showing images of the space mapping p, i.e. plots of the set $\{p(x) : x \in \mathbb{R}^2\}$, for three two-dimensional test problems. For practical reasons though, we only show the space mapping images for a subset of $\mathbb{R}^2$.

The three two-dimensional test problems are TLT2, PISTON and ROSEN, which are described in Appendix A. The space mapping images of these problems are in the Figures 4.2, 4.4 and 4.5. Table 4.1 contains the description of the markers shown in the figures.

Marker	Fine space	Coarse space
▲	$x_p^* \in X_p^*$	$p(x_p^*)$
▼	$x_{cop}^* \in X_{cop}^*$	$p(x_{cop}^*)$
$*$	$x^* \in X^*$	$z^* \in Z^*$
•	-	$p(x^*)$

Table 4.1: Description of markers in Figures 4.2, 4.4 and 4.5.

We note here that Condition **C4** is satisfied for all three test problems, hence the fine and coarse model minimizers are unique in the considered subset of $\mathbb{R}^2$. Whereas Condition **C3**, requiring a one-to-one mapping, is not completely satisfied. This, even though we are using a space mapping definition that attempts to establish uniqueness by regularization. The actual definition used is the gradient regularization definition of the space mapping, see (4.13) below.

We discuss this and other definitions of the space mapping later in this chapter. For reference, the regularization parameter used in the example is $\lambda = 10^{-4}$.

The TLT2 Space Mapping Image In Figure 4.2 the space mapping image for the TLT2 test problem is shown. We see how the fine model space is sampled densely in a rectangular area and that the image of the space mapping in the coarse model space does not have the same rectangular form. In fact, the space mapping is nonlinear, particularly around the solutions.

Further we see how the fine model minimizer x^* does not map into the coarse model minimizer z^*, hence the mapping is not perfect, as defined in (4.5), and therefore Condition **C2** is not satisfied.

Figure 4.2 also shows that the solutions x_p^* and $x_{c\circ p}^*$ do not coincide. So solving (4.1) and (4.4) do not provide the same solution. We now examine the responses at these different solutions.

In Figure 4.3 is plotted the response functions of the TLT2 problem for the fine and the coarse model evaluated at the points marked in Figure 4.2.

The objective is a minimax problem. i.e. $H(\cdot) = \max\{\cdot\}$, and the objective function values for the plotted response functions are listed in Table 4.2. The design specifications are $\max_j\{f_j\} \leqslant 0.5$, $j = 1, \ldots, 11$.

Marker	Point	$H(f(\cdot))$	$H(c(p(\cdot)))$
▲	x_p^*	0.5538	0.5217
▼	x_{cop}^*	0.4673	0.4399
$*$, •	x^*	0.4553	0.4507

Table 4.2: Description of markers in Figure 4.3. The merit function is minimax, $H(\cdot) = \max\{\cdot\}$. *Numbers are rounded to four decimals.*

We see from the figure and the table that the fine model response at the point x_p^* does not satisfy the specifications. Whereas the model responses for the two other points, x^* and $x_{c\circ p}^*$, satisfies the design specifications. In addition we see that the fine model minimizer, x^*, is better than $x_{c\circ p}^*$ only by a small margin in the objective function value.

These observations suggest that the formulation in (4.4) should be preferred over the original space mapping formulation (4.1). We will motivate this further in the discussion of the example below.

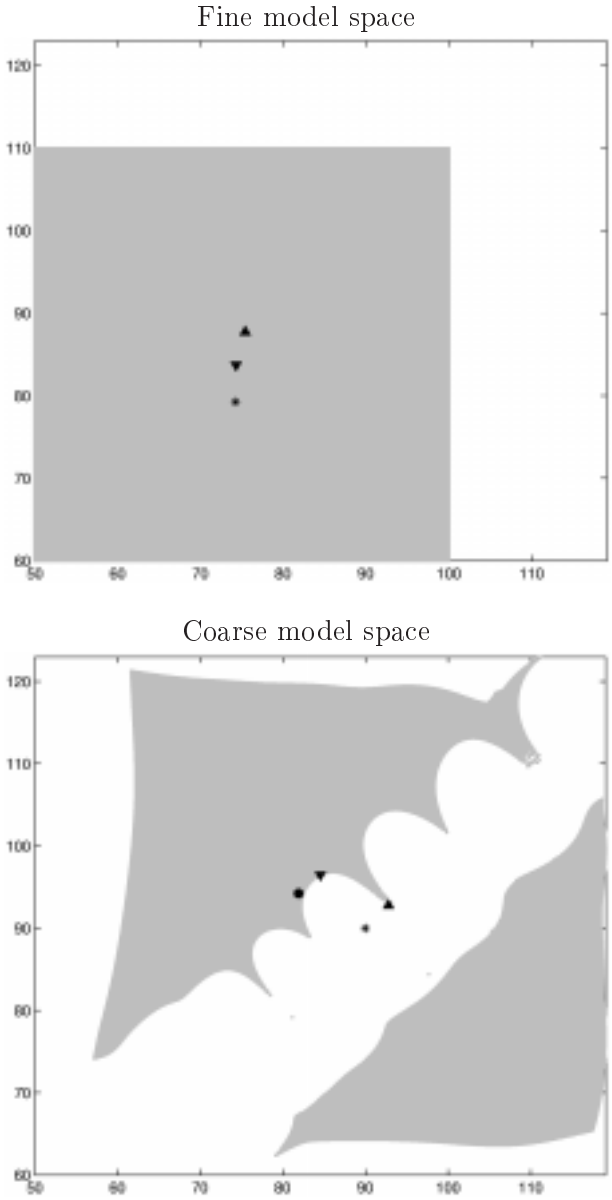


Figure 4.2: Fine model space (upper plot) and space mapping image in the coarse model space (lower plot) for the TLT2 problem. Refer to Table 4.1 for a description of the markers.

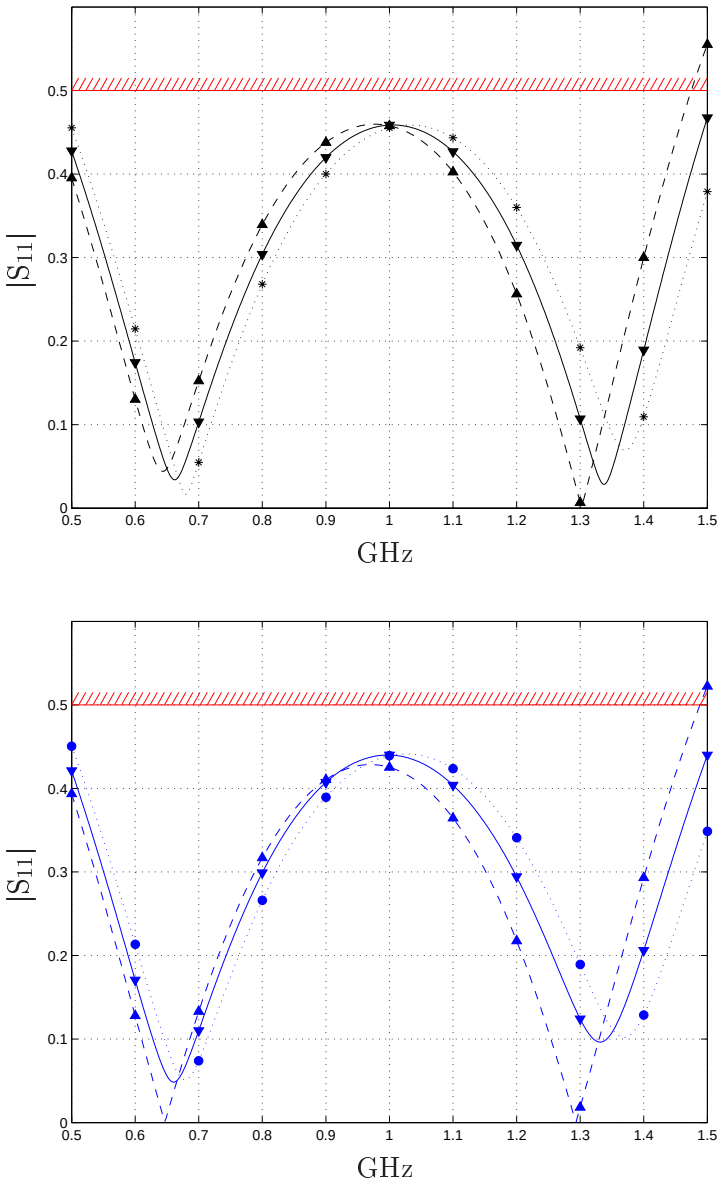


Figure 4.3: Fine model (upper) and coarse model (lower) responses at the points marked in Figure 4.2. The objective function values for the response curves are listed in Table 4.2. The hatched line indicate the response specifications.

The PISTON Space Mapping Image In Figure 4.4 the image of the space mapping is shown for the PISTON test problem. From the figure we see how z^* is in the space mapping image, $z^* \in p(x)$, corresponding to Condition **C1** being satisfied. Further, the points x_p^* and $x_{c\circ p}^*$ coincide and their image is the coarse model minimizer z^*. But, we also see from the figure that x^* does not map into z^*, hence the mapping is not perfect, $p(x^*) \neq z^*$, and therefore Condition **C2** is not satisfied for this problem.

The ROSEN Space Mapping Image In Figure 4.5 the image of the space mapping is shown for the ROSEN test problem. From the figure it is seen that for this problem the mapping is perfect, i.e. $p(x^*) = z^*$, hence Condition **C2** is satisfied. Because of this property the three points x^*, x_p^* and $x_{c\circ p}^*$ coincide. Therefore we can use the $c \circ p$ as a surrogate for f and obtain the minimizer of the latter by solving (4.4).

Discussion of the Example

As stated in the beginning of this example, all the sets of minimizers defined in Section 4.1.1 are singletons for these test problems.

The observation for the PISTON and ROSEN test problems that x_p^* and $x_{c\circ p}^*$ coincide in the fine model point mapping to the coarse model minimizer z^* is in fact a general property when Condition **C1** (or **C2**) is satisfied. This can be verified theoretically, which is done by Lemma 4.2 in the next section.

In the case where x_p^* and $x_{c\circ p}^*$ do not coincide, the TLT2 test problem indicated that the solution $x_{c\circ p}^*$ was preferable. We now argue that this may generally be the case.

The solution x_p^* maps into $p(x_p^*)$ the closest point to z^* in the space mapping image, hence

$$p(x_p^*) = \arg\min_{z \in p(\mathbb{R}^n)} \|z - z^*\|_2 \,.$$

Whereas the solution $x_{c\circ p}^*$ maps into that point in the space mapping image with lowest objective function value of the mapped coarse model, hence

$$p(x_{c\circ p}^*) = \arg\min_{z \in p(\mathbb{R}^n)} H(c(z)) \,. \tag{4.6}$$

Assume for the moment that the merit function H is a norm, $H = \|\cdot\|$. Assume also that the deviation between $c \circ p$ and f at the points $x \in \{x_p^*, x_{c\circ p}^*\}$ are

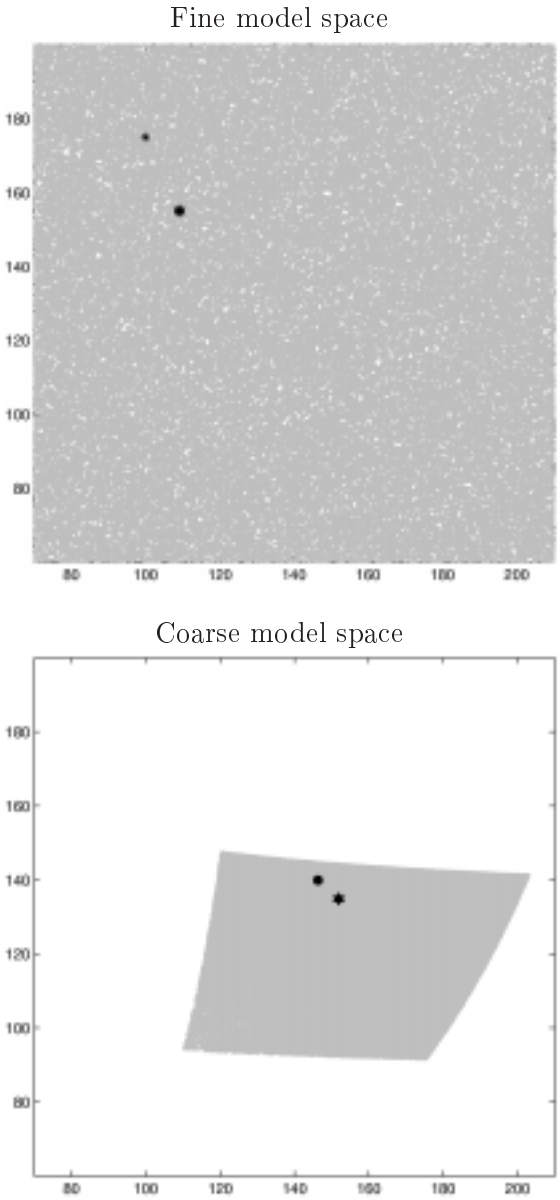


Figure 4.4: Fine model space (upper plot) and space mapping image (lower plot) for the PISTON problem. Refer to Table 4.1 for a description of the markers. Here $x_p^* = x_{c \circ p}^*$ *and their image is* z^**. The mapping is not perfect,* $p(x^*) \neq z^*$*.*

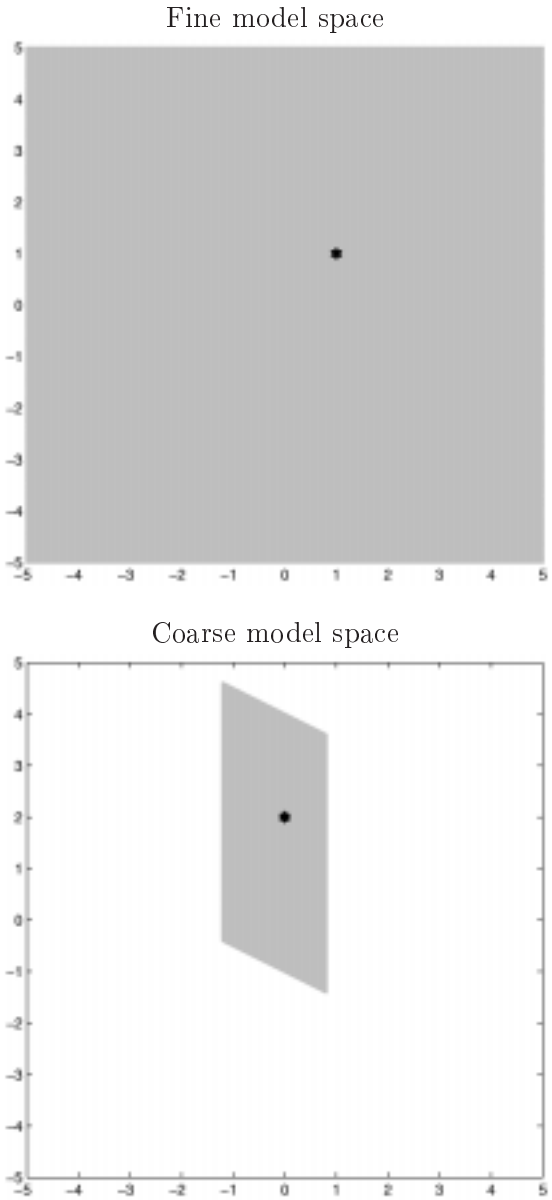


Figure 4.5: Fine model space (upper plot) and space mapping image (lower plot) for the ROSEN problem. Refer to Table 4.1 for a description of the markers. Here $x_p^* = x_{cop}^*$ *and their image is* z^**. Also, the mapping is perfect,* $p(x^*) = z^*$*.*

bounded by a constant $\varepsilon \geqslant 0$, i.e.

$$\|c(p(x)) - f(x)\| \leqslant \varepsilon\,,$$

where $\|\cdot\|$ is the same norm as H. From (4.6) we can define a $\delta \geqslant 0$ such that

$$\|c(p(x^*_{c\circ p}))\| = \|c(p(x^*_p))\| - \delta\,.$$

Then it follows that

$$\begin{aligned} \|f(x^*_{c\circ p})\| &\leqslant \|c(p(x^*_{c\circ p}))\| + \varepsilon \\ &= \|c(p(x^*_p))\| - \delta + \varepsilon \\ &\leqslant \|f(x^*_p)\| + \varepsilon - \delta + \varepsilon \\ &\leqslant \|f(x^*_p)\|\,, \quad \text{if } \; 2\varepsilon \leqslant \delta\,. \end{aligned}$$

Hence $x^*_{c\circ p}$ is a better approximation to x^* than x^*_p is. In other words, because of the arguments the problem formulation in (4.4) should be preferred over the original space mapping formulation (4.1).

The numbers for the TLT2 test problem using the L_∞-norm are:

$$\begin{aligned} \delta &= 0.0818 \qquad \text{(from Table 4.2)} \\ \varepsilon &= 0.04 \quad \Leftarrow \quad \begin{cases} \|c(p(x^*_p)) - f(x^*_p)\|_\infty = 0.0395 \\ \|c(p(x^*_{c\circ p})) - f(x^*_{c\circ p})\|_\infty = 0.0274 \end{cases} \end{aligned}$$

Hence the condition $2\varepsilon < \delta$ is met for this problem, and from Table 4.2 we verify that $x^*_{c\circ p}$ is a better than x^*_p, since $\|f(x^*_p)\|_\infty - \|f(x^*_{c\circ p})\|_\infty = 0.0864$.

With this example in mind, it would be beneficial to have a more general understanding of what conditions determine whether the space mapping technique can solve a given problem or not. The theoretical results we derive in the next sections provide a clearer understanding of this issue.

4.1.3 Theoretical Results

In this section we derive some theoretical results that characterize the space mapping technique under certain conditions.

Assuming that the Conditions **C1** and **C3** hold, we prove two lemmas about the relation between the sets of optimizers defined in Section 4.1.1.

The first lemma states that the space mapping image of $X^*_{c \circ p}$, the minimizers of the space mapped coarse model $H(c(p(x))$, is contained in the set of all coarse model minimizers Z^*.

Lemma 4.1 *If* **C1** *holds then*

$$p(X^*_{c \circ p}) = Z^* \, .$$

Proof. If $z \in p(X^*_{c \circ p})$ then z minimizes $H \circ c$ since **C1** holds, i.e. $z \in Z^*$. Reversely, if $z \in Z^*$, and since **C1** implies that $\exists x : p(x) = z$, then x minimizes $H \circ c \circ p$, i.e. $x \in X^*_{c \circ p}$, hence $p(x) = z \in p(X^*_{c \circ p})$. □

The second lemma states that the sets $X^*_{c \circ p}$ and X^*_p are identical.

Lemma 4.2 *Assume that* **C1** *holds then*

$$X^*_{c \circ p} = X^*_p \, .$$

Proof. From **C1** and the definition of X^*_p it follows that

$$x \in X^*_p \quad \Longleftrightarrow \quad p(x) \in Z^* \, . \tag{4.7}$$

If $x \in X^*_{c \circ p}$ then from Lemma 4.1 we have $p(x) \in Z^*$, and then from (4.7) it follows that $x \in X^*_p$. Reversely, if $x \in X^*_p$ then from (4.7) we have $p(x) \in Z^*$, and then from the definition of $X^*_{c \circ p}$ it follows that $x \in X^*_{c \circ p}$. □

The properties proved in the lemmas were observed in the PISTON and ROSEN test problems presented in the last section. But, in the case where **C1** does not hold we cannot be certain that the minimizers of the space mapped coarse model and the solutions to the nonlinear equations (4.1) are the same. This was the observation for the TLT2 test problem.

Next we prove a theorem which states that when **C2** and **C3** hold, minimizers of the space mapped coarse model are also minimizers of the fine model. In other words, if $x \in X^*_{c \circ p}$ then x is a minimizer of $H(f(x))$, hence $x \in X^*$.

Theorem 4.1 *If* **C2** *and* **C3** *hold then*

$$x \in X^*_{c \circ p} \quad \Rightarrow \quad x \in X^*$$

PROOF. Let $x \in X^*_{c\circ p}$ then, since **C2** implies **C1**, from lemma 4.1 we have that $p(x) \in Z^*$ and from **C2** it follows that $Z^* \subseteq p(X^*)$, hence from **C3** $x \in X^*$. Formally,

$$\begin{aligned} & x \in X^*_{c\circ p} \\ \Downarrow & \\ & p(x) \in Z^* \subseteq p(X^*) \\ \Downarrow & \\ & x \in X^* \ , \end{aligned}$$

which proves the theorem. □

A special case of the theorem is when the minimizers of $H(f(x))$ and $H(c(z))$ are unique (Condition **C4**), then the minimizer of $H(c(p(x)))$ is the minimizer of $H(f(x))$, as it is stated in the following corollary.

COROLLARY 4.1 *If* **C2**, **C3** *and* **C4** *hold then*

$$X^*_{c\circ p} = \{x^*\}$$

PROOF. The result is a consequence of theorem 4.1. □

The result of Corollary 4.1 was observed in the example in the last section for the ROSEN test problem. The ROSEN problem seems more ideal for the space mapping approach, than the two other problems that were presented. But we should keep in mind that the observations in the example were based on a particular way of defining the space mapping. So other choices may lead to different results for these test problems.

Neither theorem 4.1 nor corollary 4.1 depend on the actual form of the space mapping p. The results only define a set of conditions, for the space mapping, that are sufficient to state that the minimizers of $H \circ c \circ p$ are also minimizers of $H \circ f$. So there is actually a great deal of freedom in choosing a definition that align the models, such that the assumptions of Theorem 4.1 are met. However, we consider it to be uncommon that the assumptions of the theorem hold in practice, but the theorem is helpful as a guide to what we should aim for when defining the space mapping. The next section introduce the usual definition of the space mapping, and show some theory concerning the scalar function case of this definition.

4.1.4 The Usual Space Mapping Definition

There has been consensus in the literature, see the review papers [4, 9], to define the space mapping as a mapping relating similar responses, namely

$$p(x) \in \arg\min_{z \in \mathbb{R}^n} \|c(z) - f(x)\|_2^2 \,. \tag{4.8}$$

Sometimes the Huber-norm or the L_1-norm are used instead of the L_2-norm. Also a weighted least-squares definition has been used. It depends on the application. We denote (4.8) the *usual space mapping definition*.

In more general terms, a problem of calculating the space mapping p, as e.g. the usual space mapping definition above, is often referred to as a *parameter extraction problem*. We define the set $\mathcal{P}(x)$ to be a point-to-set mapping containing all solutions of a given parameter extraction problem. Hence, in the case of the usual space mapping, $\mathcal{P}(x)$ is for given x the point-to-set mapping containing all solutions of (4.8),

$$\mathcal{P}(x) = \arg\min_{z \in \mathbb{R}^n} \|c(z) - f(x)\|_2^2 \,. \tag{4.9}$$

Next we present two theoretical results that characterize the usual space mapping under certain conditions in the case where f and c are scalar functions. One of the results is that the usual space mapping may be nonunique, a property also observed in the vector function case, as we will see later. Hence, the parameter extraction problem may have many local solutions, i.e. **C3** does not hold and $\mathcal{P}(x)$ may not be a singleton. Then for a given point x, $p(x)$ must be chosen among the points in $\mathcal{P}(x)$. Several strategies for doing this have been proposed in the literature, we review these in Section 4.2. Now we consider some theory for the scalar case, and illustrate some properties of the usual space mapping in a numerical example.

Theory in the Scalar Case

We now treat the special case where f and c are scalar functions, i.e. $m = 1$. Two propositions are presented concerning the usual definition of the space mapping. So for now, for any given x let the space mapping $p(x)$ be an arbitrary point in $\mathcal{P}(x)$, the set of all solutions to the parameter extraction problem for the usual space mapping definition, as defined in (4.9). The propositions below are unaffected by which solution from $\mathcal{P}(x)$ that is chosen.

The first proposition describes the situation where f in a region around x^* is below all possible values of c.

PROPOSITION 4.1 *Assume that* **C4** *holds and* $f(x^*) < c(z^*)$ *then there exists an open neighbourhood* $\mathcal{N}(x^*)$ *around* x^* *such that for* $x \in \mathcal{N}(x^*)$ *the following holds*

1. $f(x) < c(z)$, $\forall z \in I\!R^n$,
2. $p(x) = z^*$.

PROOF. The first part follows directly from the smoothness assumption on f: For all sufficiently small $\varepsilon \in I\!R^n$ we have

$$f(x^* + \varepsilon) < c(z^*) \leqslant c(z) \ ,$$

for all $x \in I\!R^n$. From this, it follows that for $x \in \mathcal{N}(x^*)$

$$\arg \min_{z \in I\!R^n} (c(z) - f(x))^2 = \arg \min_{z \in I\!R^n} c(z)$$

and since $z = z^*$ is a unique minimizer of $c(z)$ it follows from (4.9) that $\mathcal{P}(x) = \{z^*\} \ \Rightarrow \ p(x) = z^*$, which concludes the proof. □

The proposition states that the space mapping is constant in a neighbourhood around x^*. Due to this, an attempt to minimize the mapped coarse model $c(p(x))$ will fail or stop when the iterates enter the neighbourhood. We note that the mapping is perfect, as defined by (4.5), hence **C2** is satisfied. So, the reason that corollary 4.1 does not apply here is that the mapping in (4.9) is not one-to-one as assumed by **C3**. In fact, the proposition shows that there exists a set of points, in the neighbourhood of x^*, which all map to the point z^*.

The next proposition describes the situation where c in a region around z^* is below all possible values of f.

PROPOSITION 4.2 *Assume that* **C4** *holds,* $f(x^*) > c(z^*)$ *and that* $\exists \bar{z} \in I\!R^n$ *for which* $f(x^*) = c(\bar{z})$*, then there exists an open neighbourhood* $\mathcal{N}(z^*)$ *around* z^* *such that for* $z \in \mathcal{N}(z^*)$ *the following holds*

1. $f(x) > c(z)$, $\forall x \in I\!R^n$,

2. $x^* \in \arg\min_{x \in \mathbb{R}^n} c(p(x))$,

3. $\mathcal{P}(x) \cap \mathcal{N}(z^*) = \emptyset$, $\forall x \in \mathbb{R}^n$.

Proof. The first part follows directly from the smoothness assumption on c: For all sufficiently small $\varepsilon \in \mathbb{R}^n$ we have

$$c(z^* + \varepsilon) < f(x^*) \leqslant f(x) \,,$$

for all $x \in \mathbb{R}^n$.

From the assumptions about $\bar{z}$ and (4.9) it follows that $\bar{z} \in \mathcal{P}(x^*)$. Then second part follows from the fact that $c(p(x^*)) = c(\bar{z})$ is the lowest possible value of $c(p(x))$, which we prove now.
To prove that $c(p(x)) \geqslant c(p(x^*))$ for all x, we assume for a moment that the reverse is true: There exists an x, $x \neq x^*$, for which $c(p(x)) < c(p(x^*))$. Then $f(x) \geqslant f(x^*) = c(p(x^*)) > c(p(x))$, hence

$$\begin{aligned} |c(p(x)) - f(x)| &= f(x) - c(p(x)) \\ &> f(x) - c(p(x^*)) \\ &= f(x) - c(\bar{z}) \\ &\geqslant 0 \end{aligned}$$

which contradicts that $z = p(x)$ minimizes $|c(z) - f(x)|$. Hence the assumption is wrong and it follows that $c(p(x^*))$ is the minimum value of $c(p(x))$.

Regarding the third part: From part two we have $c(p(x)) \geqslant c(\bar{z})$. Then, since $c(z) < c(\bar{z})$ for all $z \in \mathcal{N}(z^*)$, it follows that $p(x) \notin \mathcal{N}(z^*)$, hence $\mathcal{P}(x) \cap \mathcal{N}(z^*) = \emptyset$, which concludes the proof. □

The proposition states that it is not possible to choose any x for which $p(x)$ is in the neighbourhood around z^*. Hence p is always outside of this neighbourhood, therefore the mapping cannot be perfect, i.e. $p(x^*) \neq z^*$. Further, the proposition states that x^* minimizes $c(p(x))$.

In the cases described by the propositions above the usual mapping definition (independent of how p is chosen from $\mathcal{P}$) fails to satisfy the assumptions of corollary 4.1. The above results are illustrated in the numerical example next.

4.1.5 Example With Scalar Functions

In order to visualize the propositions presented above, we now introduce a simple one-dimensional example with quadratic functions. The example is considered again later, when alternative definitions of the space mapping are introduced.

The problem is defined by a fine model and two different coarse models. The fine model response is the quadratic

$$f(x) = \frac{1}{2}x^2 - x + 2 \,, \qquad x \in [-2, 5] \,,$$

with the unique minimizer $x^* = 1$.

The first coarse model is the fine model scaled by a factor two and with a simple shift of the parameters, the response function is

$$\begin{aligned} c_1(z) &= 2 \cdot f(z-1) \\ &= (z-1)^2 - 2(z-1) + 4 \,, \qquad z \in [-2, 5] \,, \end{aligned}$$

with the unique minimizer $z^* = 2$. Note that this coarse model is above f for all parameter values, hence $\forall x, z = x \;:\; c_1(z) > f(x)$. So the assumptions of proposition 4.1 are satisfied for the pair (f, c_1).

The second coarse model is like the first one, only shifted downwards,

$$\begin{aligned} c_2(z) &= 2 \cdot f(z-1) - 3 \\ &= (z-1)^2 - 2(z-1) - 1 \,, \qquad z \in [-2, 5] \,. \end{aligned}$$

The minimizer is the same, $z^* = 2$. Note that c_2 has a region around z^* where it is below all possible values of f. So the assumptions of proposition 4.2 are satisfied for the pair (f, c_2).

The functions are shown in the plots in the top of Figure 4.6. The usual mapping objective function $\|c(z) - f(x)\|_2$ is shown for $x = 3$ in the plots in the bottom of the figure. We see that the parameter extraction problem (4.8) has two solutions for all x where $f(x) > c(z^*)$. Further, the reader can imagine that for all x where $f(x) < c(z^*)$ (this only applies to the plots on the left) the parameter extraction problem has a unique solution, namely z^*.

From the figure, it should be clear that for the considered point $x = 4$, the right one of the two solutions is the wanted solution to the parameter extraction

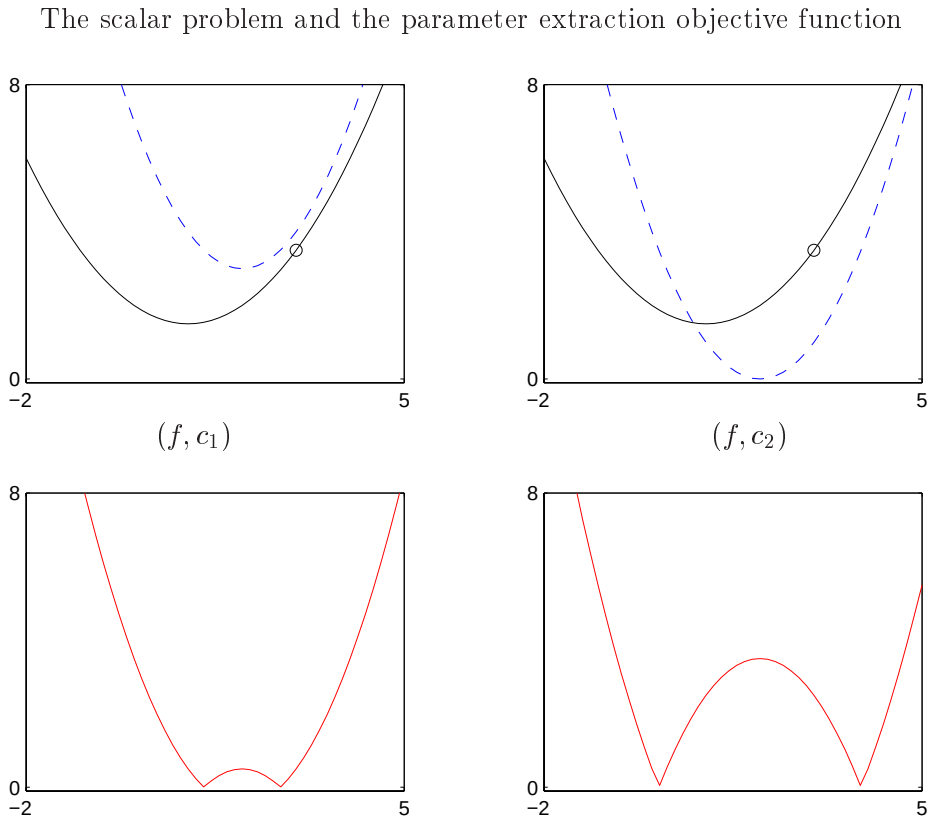


Figure 4.6: The top plots show f (——) and c (– –), the point $(3, f(3))$ is marked (∘). The lower plots show the parameter extraction objective function (4.8) for $x = 3$, $\|c(z) - f(3)\|_2$, (——). In the left plots $c = c_1$, in the right plots $c = c_2$.

problem. However, in the case where f is expensive, we would not be able to plot the function to assist in choosing the correct solution of the parameter extraction problem. The same goes in general for problems in more than two dimension. So in these cases we cannot in general tell which one of the multiple solutions to the parameter extraction problem that is preferable.

The problem of having nonunique or local solutions in the parameter extraction problem was first described in [7]. A very similar observation, to that of our simple example, was for a two dimensional vector function problem presented in [5]. In this reference it was suggested to enhance uniqueness by including more than one point in the parameter extraction problem, the so-

called multipoint parameter extraction technique. We will return to this and other approaches to enhance the uniqueness of the parameter extraction later in this chapter.

For the purpose of illustrating the mapped coarse model $c(p(x))$ for this simple problem, choosing an arbitrary solution of the parameter extraction is adequate. So, the usual space mapping definition (4.8) provides the space mapped coarse models $c_1(p(x))$ and $c_2(p(x))$ shown in Figure 4.7. For the fine and

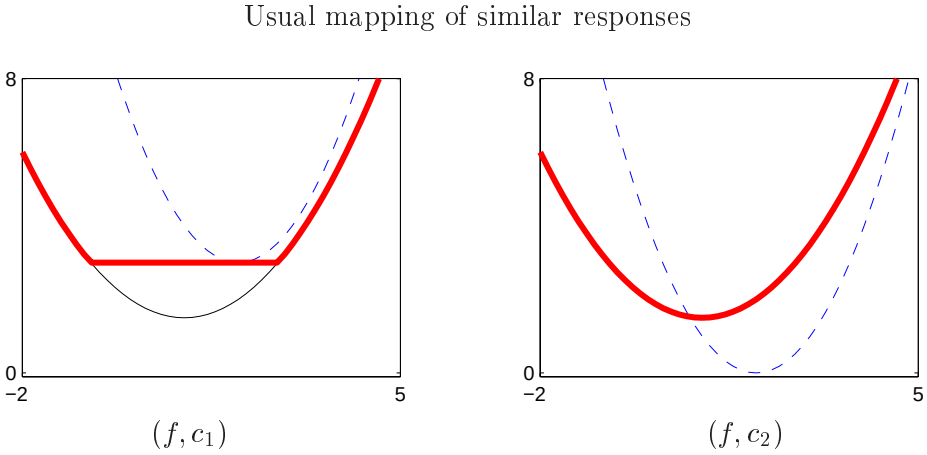


Figure 4.7: The plots show f (——), c (– –) and $c(p(x))$ (——). In the left plot $c = c_1$, in the right plot $c = c_2$. In both plots, p maps similar responses.

coarse model pair (f, c_1), we see how $c_1(p(x))$ is constant valued, at the value $c_1(z^*)$, in that range of x for which $f(x) < c_1(z^*)$. So minimizing $c_1(p(x))$ has a range of solutions where $p(x) = z^*$, including the point x^*. Hence the mapping is perfect, $p(x^*) = z^*$, as we would expect from Proposition 4.1. But as there are an infinity of solutions we cannot determine the right one, namely x^*, based on this technique.

For the fine and coarse model pair (f, c_2), we see how $c_2(p(x))$ is equal to f for all parameter values, so x^* minimizes $c_2(p(x))$, as we would expect from Proposition 4.2. We also see that the mapping is not perfect, as there is a region around z^* that cannot be reached by $p(x)$, $x \in \mathbb{R}^n$. So the original space mapping technique, solving the nonlinear equations (4.1), does not have a solution. But the least-squares formulation (4.2) of the same problem has x^* as solution.

Although we will not generalize from this simple one-dimensional scalar test example, clearly the usual space mapping definition is unsatisfactory for the problem defined by (f, c_1), since there is a set of minimizers of $c \circ p$, not just the desired point x^*.

Some of the observations in the one-dimensional scalar case, in fact, also appear in the multidimensional vector case, as we will show in the following. We now return to the case of general vector functions, to introduce another example.

4.1.6 Example With Vector Functions

We now consider the parameter extraction problem for the two-dimensional TLT2 problem, described in appendix A. The problem was also used in the example in Section 4.1.2 above. The problem is a minimax problem with 11 response functions. We choose to map the point $\tilde{x} = (90, 90)^T$, which is the coarse model minimizer z^*. This point is used for illustration here, as it is the starting point of any space mapping optimization algorithm, as this corresponds to the initial assumption that f and c are identical [4].

The minimax contours of the fine and the coarse model are shown in Figure 4.8. We see the location of the coarse model minimizer z^* and the fine model minimizer x^*. In the figure, also three additional points are marked, these are introduced next.

A contour plot of the parameter extraction problem for the usual space mapping definition (4.8) is as shown in Figure 4.9. We see that, similar to the scalar function example above, this problem has multiple solutions to the parameter extraction problem, namely the two solutions marked in the figure. We denote these local solutions by z_1 and z_2. We denote the saddlepoint between the local solutions by $\tilde{z}$.

In Figure 4.10 the actual values of the response functions are shown for the fine model at $\tilde{x}$, the coarse model at $\tilde{z}$ and the mapped coarse model at z_1 and z_2. From the figure we see that the response $c(\tilde{z})$ is below or equal to $f(\tilde{x})$ for almost every response function, except for one. Whereas the identical responses $c(z_1)$ and $c(z_2)$, at the local solutions of the parameter extraction problem, behaves like we would expect from a L_2-norm data fitting solution. That is the $c(z_1)$ is not completely above or below $f(\tilde{x})$, the residuals $c(z_1) - f(\tilde{x})$ have alternating signs.

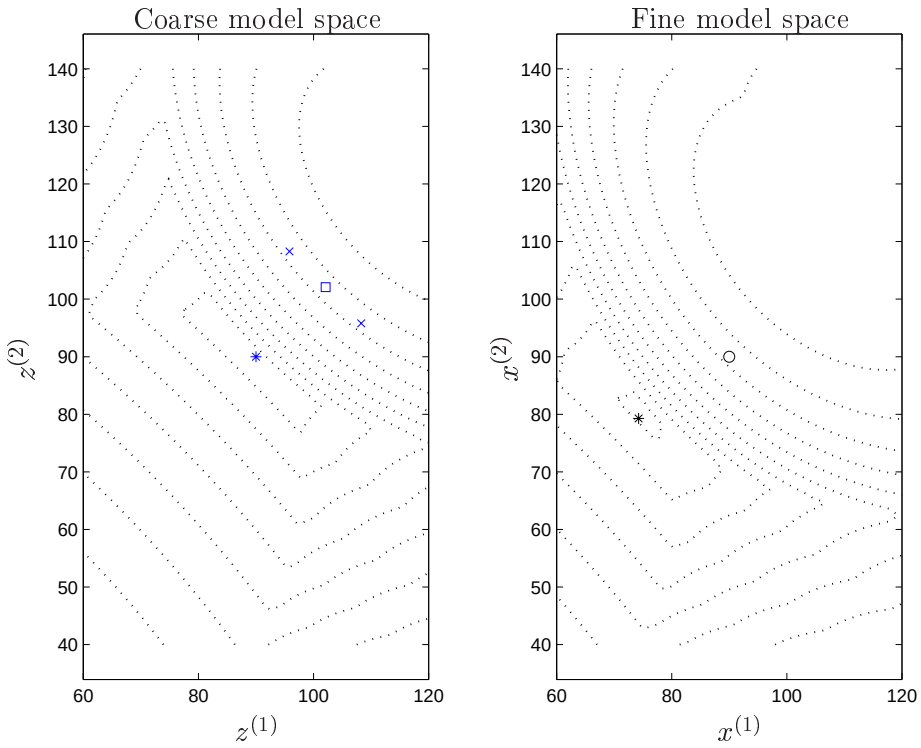


Figure 4.8: Minimax contours of coarse model (left) and fine model (right). Left plot: The coarse model minimizer z^* ($*$), two local solutions z_1 and z_2 ($\times$) of the parameter extraction problem and the saddlepoint $\tilde{z}$ between them ($\square$) are shown. Right plot: The fine model minimizer x^* ($*$) and the point to map $\tilde{x}$ ($\circ$) are shown.

The local solutions provide a better fit, in terms of lower L_2-norm of the residual, than the saddlepoint. However, since we cannot determine (from the information available here) which of the solutions z_1 and z_2 that is preferable, the saddlepoint $\tilde{z}$ between them may be the best choice. As by choosing the saddlepoint as the solution, we minimize the error of choosing the wrong solution. Further it is seen that $\tilde{z}$ is the closest point to z^* of the three points considered.

We note that $\tilde{z}$ also seems preferable in the context of aligning the parameter spaces, since it has the best visual conformance with the relative distance to the minimizer and contours in Figure 4.8.

In Figure 4.2, showing the mapping image for this problem, it was visualized

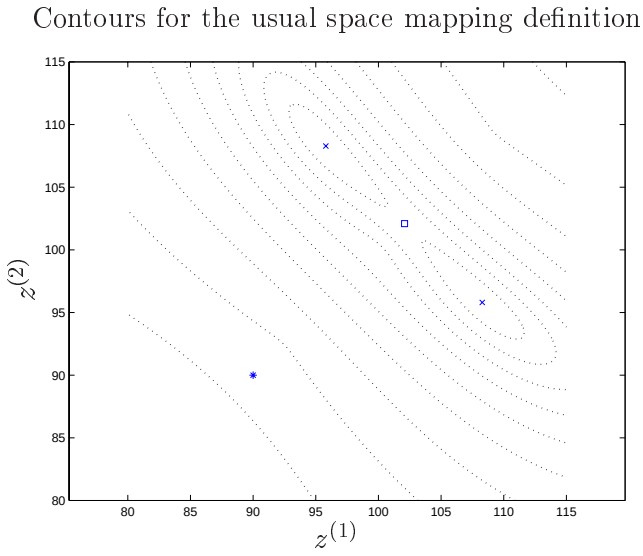


Figure 4.9: L_2-norm contours of the parameter extraction problem for the usual space mapping definition (4.8)*. The coarse model minimizer z^* ($*$), two local solutions z_1 and z_2 ($\times$) and the saddlepoint $\tilde{z}$ ($\square$) are shown.*

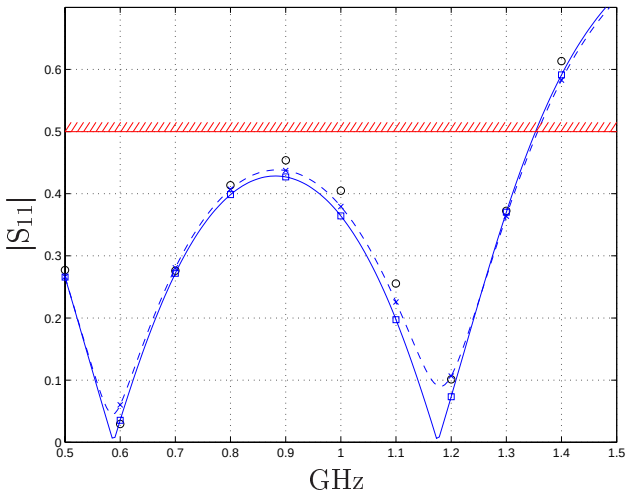


Figure 4.10: The figure shows $f(\tilde{x})$ ($\circ$), $c(\tilde{z})$ (——) and c at the local solutions z_1 and z_2 (– –) of the parameter extraction problem for the TLT2 (minimax) problem. The hatched line indicate the response specifications of the problem.

that the usual space mapping definition does not satisfy condition **C1** for this problem. Hence the coarse model minimizer is not in the image of the usual space mapping. Further, the minimizer of $H \circ c \circ p$ is not x^*.

So to summarize, the Conditions **C1** and **C3** are not met by the usual space mapping definition on neither the scalar nor the vector example. It is displeasing that there exist several solutions to the parameter extraction problem, and that the minimizer of $H \circ c \circ p$ is not the unique point x^*, in the scalar example. Now we turn the attention to alternative space mapping definitions, that have been proposed to improve on the imperfections mentioned here.

4.2 Alternative Space Mapping Definitions

The most focus, in improving the usual mapping definition (4.8), has been on avoiding non-uniqueness, i.e. to assure that **C3** holds. Some attention have also been given to the problem of assuring that the space mapping problem (4.1) actually has a solution, i.e. that **C1** holds.

In both the mentioned cases the usual approach is to constrain the parameter extraction problem to a smaller subset of points, in practice by regularizing the problem, and thereby introduce a bias toward a certain subset of points.

Other techniques have been considered. In [7] it was proposed to perform a change in the (physical) state variables (in their case frequency) of the coarse model. By doing this, local minima could be avoided in the parameter extraction problem. A similar strategy has been investigated in [3].

In the following we consider four definitions of the regularized parameter extraction problem, intended for the case where **C3** does not hold, i.e. $\mathcal{P}(x)$ may be multi-valued for a given x. The definitions are general for vector valued response functions, we illustrate some of their properties on the two examples introduced in Section 4.1.5 and Section 4.1.6 above.

4.2.1 Regularization Using z^*

In the case where there are several local solutions to the parameter extraction problem, probably the simplest strategy to help convergence of the classical space mapping problem (4.2) is to let $p(x)$ be the solution, among all solutions $\mathcal{P}(x)$ of the usual space mapping definition (4.9), closest to a coarse model

minimizer,

$$p(x) \in \arg\min_{z \in \mathcal{P}(x)} d(z, Z^*)\,. \tag{4.10}$$

In cases where Condition **C1** is not met, there is no solution to the non-linear equations of the classical space mapping problem (4.1) — a situation encountered in practice, as demonstrated by the test examples above. A way to circumvent this difficulty is to drive $\mathcal{P}$ closer to $z^* \in Z^*$ by regularizing the usual space mapping with distance to z^*. Bandler et al. [8] proposed a space mapping definition with such a property,

$$p_\lambda(x) = \arg\min_{z \in \mathbb{R}^n} \left\{ \; (1-\lambda)\, \|c(z) - f(x)\|_2^2 + \lambda\, \|z - z^*\|_2^2 \; \right\} \;, \tag{4.11}$$

for some value of $0 \leqslant \lambda < 1$.

This definition performs rather poorly on the one-dimensional scalar example from Section 4.1.5 as we show now. In Figure 4.11 the mapped coarse model $c(p_\lambda(x))$ is shown for the space mapping definition in (4.11), for two different values of λ. We see from the figure how increasing the value of λ, as one would expect, draws more points toward z^* and thereby the function value $c(z^*)$. But the definition does not help solving the problem of an infinite solutions when minimizing $H \circ c \circ p$. Actually this space mapping definition behaves worse than the original space mapping definition, as for the case with (f, c_2) the function $H \circ c \circ p$ attains its minimum value at an infinite number of points for large values of λ.

So for the scalar function example, the ill-posed problem (f, c_1), regularization only increase the set of points which map to z^*, and thereby makes the problem even more ill-posed. For the well-posed problem (f, c_2) regularization risks making the problem ill-posed by introducing a set of points which map to z^*.

For the vector function example, introduced in Section 4.1.6, the contours of the regularized parameter extraction problem (4.10) are shown in Figure 4.12 for $\lambda = 1.4 \cdot 10^{-4}$. Compared to Figure 4.9 we see how the two local solutions to the parameter extraction problem are drawn to a unique solution near $\tilde{z}$, the desired parameter extraction solution. While for larger values of λ the parameter extraction solution is drawn close to z^* as expected.

In Figure 4.13 the regularized parameter extraction solution is shown for a number of λ values.

So even though the definition (4.10) seems impractical in the scalar function example, it is definitely useful in the vector function example, provided that suitable value of λ can be estimated.

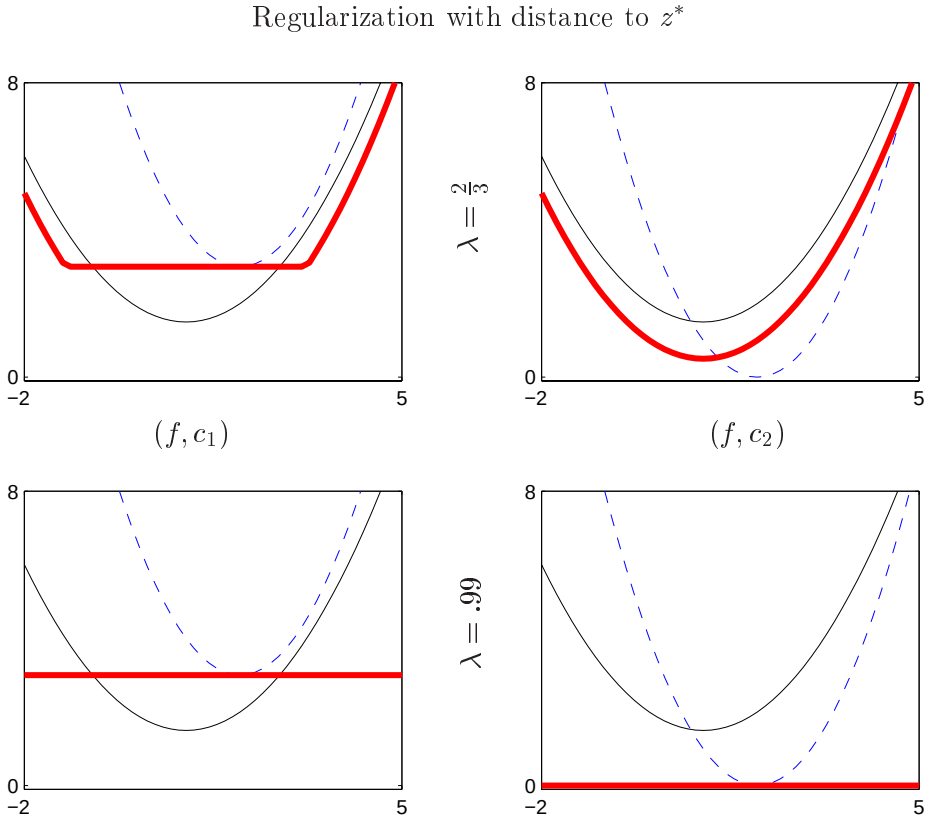


Figure 4.11: The plots show f (——), c (– –) and $c(p_\lambda(x))$ (▬). In the left plots $c = c_1$, in the right plots $c = c_2$. In all plots p_λ, as defined in (4.11), is a mapping of similar responses using regularization with distance from z^*, the coarse model optimizer. In the top plots $\lambda = \frac{2}{3}$, in the bottom plots $\lambda = .99$.

4.2.2 Regularization Using x

Another strategy is to penalize large distances to x, the current point to space map, i.e. define the space mapping as

$$p_\lambda(x) = \arg\min_{z \in \mathbb{R}^n} \left\{ (1-\lambda) \left\| c(z) - f(x) \right\|_2^2 + \lambda \left\| z - x \right\|_2^2 \right\} , \tag{4.12}$$

for some value of $0 \leqslant \lambda < 1$. Vicente [12] introduced a similar approach, but instead of a penalization strategy, he proposed the following space mapping

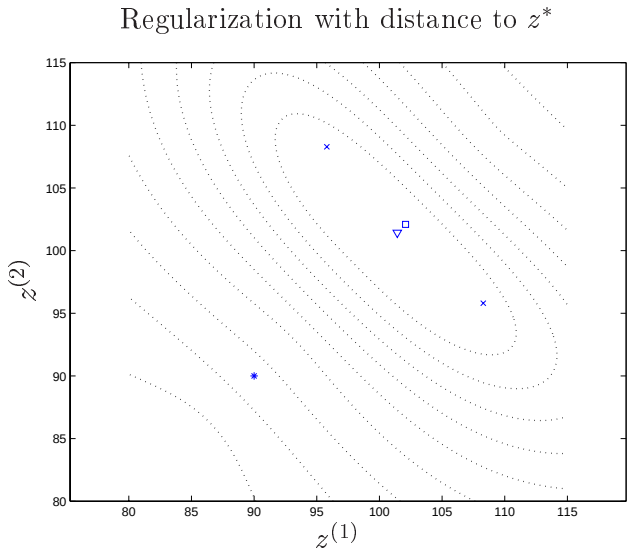


Figure 4.12: Contour plot of the regularized parameter extraction and the solution (∇) is shown for $\lambda = 1.4 \cdot 10^{-4}$. The regularization term is $\|z - z^*\|_2^2$, see (4.11).

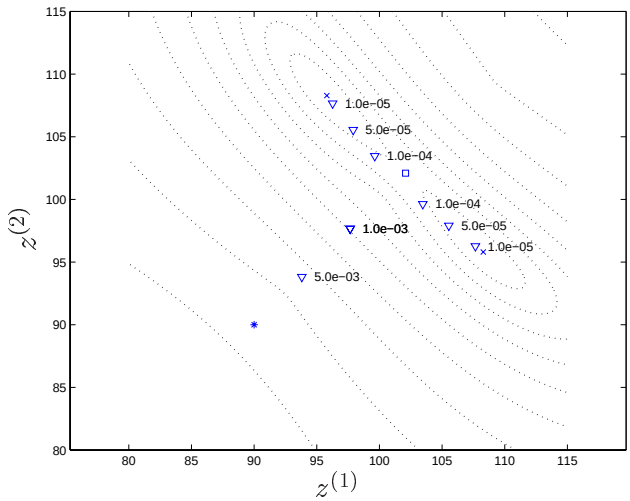


Figure 4.13: Regularized parameter extraction solutions (∇) are shown for different values of λ. The regularization term is $\|z - z^*\|_2^2$, see (4.11). The contours are from the parameter extraction problem with the usual space mapping definition (4.8).

definition

$$p(x) = \arg\min_{z\in\mathbb{R}^n} \|z-x\|_2^2 \quad \text{s.t.} \quad c(z) = f(x)$$

Vicente [12] showed regularity of his space mapping definition, in the case where f and c are scalar functions, and thus existence of directional derivatives. Considering $(1-\lambda)/\lambda$ as a Lagrange multiplier for the constraint, the two definitions are equivalent for $\lambda > 0$.

Intuitively the mapped coarse model $c(p(x))$ will approach the coarse model $c(x)$ as the regularization parameter is approaching one. So if the coarse model is a good approximation to the fine model, using this space mapping definition is a way of favoring the behaviour of the coarse model in the case of nonuniqueness in the parameter extraction.

In Figure 4.14 $c(p_\lambda(x))$ is shown for the space mapping definition in (4.12), for two different values of λ. From the figure it is seen that for sufficiently large values of λ, this definition provides a mapped coarse model with a unique minimizer. This unique minimizer is z^*, so for finding x^* the usefulness of this definition is limited though.

Solutions of the regularized parameter extraction problem of the vector function example, see Section 4.1.6, are the same as shown in Figure 4.13, as in this example the point mapped $\tilde{x}$ is chosen as z^*. So as discussed above, for values of $\lambda \gtrsim 10^{-4}$ the regularized parameter extraction problem has a unique solution.

From the two examples, we see that regularizing with the distance to x provides a unique minimizer of the mapped coarse model, provided λ is chosen large enough. In that way, this definition is preferable to the definition regularizing with distance to z^*, which had a set of solutions in the scalar case.

4.2.3 Regularization Using Gradients

A third strategy, suggested in [10], is to penalize deviation between the gradients, i.e. define the mapping as

$$p_\lambda(x) = \arg\min_{z\in\mathbb{R}^n} \left\{ (1-\lambda)\,\|c(z)-f(x)\|_2^2 + \lambda\,\|c'(z)-f'(x)\|_F^2 \right\}, \quad (4.13)$$

for some value of $0 \leqslant \lambda < 1$ and $\|\cdot\|_F$ being the Frobenius norm. If f' is not explicitly available an approximation can be used, e.g. a secant or finite difference approximation.

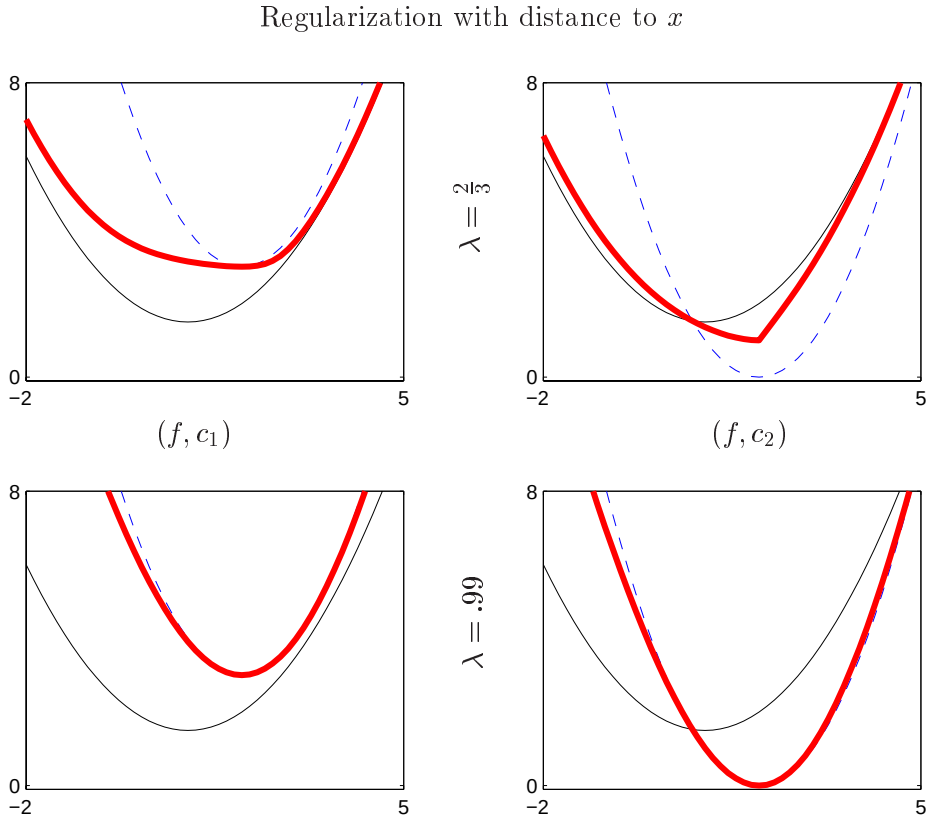


Figure 4.14: The plots show f (——), c (– –) and $c(p_\lambda(x))$ (——). In the left plots $c = c_1$, in the right plots $c = c_2$. In all plots p_λ, as defined in (4.12), is a mapping of similar responses using regularization with distance from x. In the top plots $\lambda = \frac{2}{3}$, in the bottom plots $\lambda = .99$.

The idea of matching gradient information is intuitively appealing in the context of optimization. This is because we are looking for stationary points, preferably minimizers though, in an optimization problem, and these points are characterized by the gradients of the objective function vanishing. So if we can make the gradients of $c \circ p$ match those of f the hope is that $c \circ p$ can serve as a surrogate for f in the search for x^*.

For the scalar function example, the mapped coarse model $c(p_\lambda(x))$ is shown in Figure 4.15 for the space mapping definition in (4.13), for two different values of λ. For sufficiently large values of λ (as the ones shown in the figure)

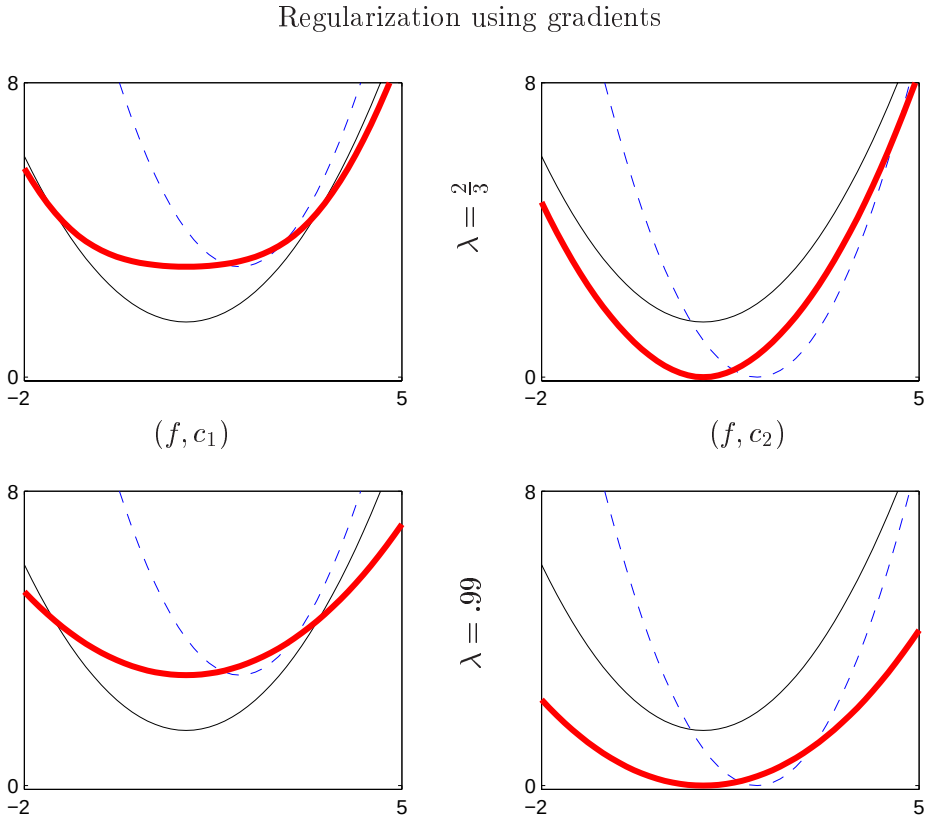


Figure 4.15: The plots show f (——), c (– –) and $c(p_\lambda(x))$ (——). In the left plots $c = c_1$, in the right plots $c = c_2$. In all plots p_λ, as defined in (4.13), is a mapping of similar responses using regularization with gradient values. In the top plots $\lambda = \frac{2}{3}$, in the bottom plots $\lambda = .99$.

this mapped coarse model has a unique minimizer at x^*. Using only gradient information to define the space mapping, i.e. for large values of λ, this desired property is retained.

For the vector function example the parameter extraction solutions are shown in Figure 4.16 for different values of λ. We see that for value of $\lambda \lesssim .97$ two local solutions exists. For larger values of λ there is a unique solution near the lower of the local solutions. When λ approach one, the solution moves to the best least-squares fit of the gradients. For this simple example we have available a finite difference approximation to both the coarse and the fine

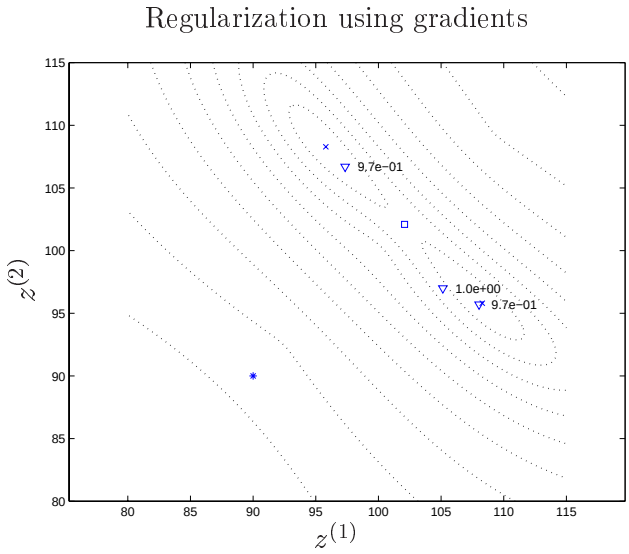


Figure 4.16: Regularized parameter extraction solutions (∇) are shown for different values of λ. The regularization term is $\|c'(z) - f'(x)\|_F$, see (4.13)*. The contours are from the parameter extraction problem with the usual space mapping definition* (4.8)*.*

model gradients. In Table 4.3 we see the angles between the gradients at the solution of the parameter extraction problem with $\lambda = 1$. Even though the table illustrates the angles from the best fit, the gradients are apparently not very well aligned. The advantage of using this space mapping definition is not obvious from this example.

Using gradients for regularization seems advantageous in the scalar case. In fact, it is much better than the other definitions presented so far, as the mapped coarse model is smooth with a unique minimizer at x^*. Though the vector function example does not encourage this definition in the same way. Another aspect that should be considered is of course the extra expense of this definition. The gradients of f rarely are explicitly available, and f usually is so expensive that a difference approximation is infeasible, only inexact approximations, like secant approximations, can be assumed available in practice. From the scalar example above we cannot claim that the same good results are preserved if using approximate gradient information.

Response function no.	Angle in degrees
1	0.83
2	28.02
3	3.24
4	3.94
5	11.36
6	1.77
7	1.15
8	10.53
9	4.03
10	3.08
11	3.11

Table 4.3: Angle between the gradients of the fine and coarse model response functions at the gradient parameter extraction solution for $\lambda = 1$. Numbers are rounded to two decimals.

4.2.4 Multiple Points

The idea of using fine model information from more than a single point in the parameter extraction process to enhance uniqueness was introduced in [5]. This reference did not specify where to position the auxiliary points, at which the fine model should be evaluated, for the multi point parameter extraction. In [1] an automated technique was presented, it was suggested to use the points previously visited by the algorithm, where f is known, in the multipoint parameter extraction.

In [2] a refined technique was suggested, aiming at minimizing the number of auxiliary points to be evaluated. The strategy was referred to as an *aggressive approach to parameter extraction*. In the aggressive approach the auxiliary points are positioned by the algorithm in certain distances $h_i \in \mathbb{R}^n$, $i = 1, \ldots, k$ from x. The positions $\{x + h_1, \ldots, x + h_k\}$ are derived such that the rank of the coarse model Jacobian is maximized at x. In practice the distances are found by solving a number of eigenvalue problems, related to the coarse model response functions derivatives and Hessians.

In our examples we consider a simpler strategy, where the auxiliary points are positioned in fixed distances $h_1, \ldots, h_k$ from x. Let the steps from x to the auxiliary points be denoted $h_1, \ldots, h_k$, then the multipoint space mapping can

be defined as

$$p_\lambda(x) = \arg\min_{z\in\mathbb{R}^n} \Big\{ (1-\lambda)\|c(z) - f(x)\|_2^2 + \lambda \|c(z+h_1) - f(x+h_1)\|_2^2 + \ldots + \lambda \|c(z+h_k) - f(x+h_k)\|_2^2 \Big\} , \tag{4.14}$$

for some value of $0 \leqslant \lambda < 1$. Separate weights, $\lambda_1, \ldots, \lambda_k$, for each regularization term could also be used. We only consider the simple case with one λ.

For small steps the multipoint definition in some sense resembles the gradient definition, particularly in the case where the gradient definition relies on an inexact gradient approximation. For large steps, provided that c and f have great similarity in their global behaviour, the idea of using multiple points is like averaging the error between the two functions over a large interval. In that way the local deviations are smeared out, but the global trend is retained.

The scalar example: When placing the auxiliary points in the small distances, e.g. $h_1 = 0.1$ and $h_2 = -0.1$, the definition has no effect on the shape of the mapped coarse model. So the functions look as for the usual space mapping definition in Figure 4.6. But, compared to the definition regularizing with distance to z^*, this definition does not destroy the good property of having a unique minimizer at x^* for the pair (f, c_2).

If the auxiliary points instead are placed in larger distances from x, e.g. $h_1 = 1$ and $h_2 = -1$, the resulting mapped coarse model is quite different. This is seen in Figure 4.17, where the mapped coarse model attains its minimum at a unique point, namely at x^*. Hence, for this simple problem, enlarging the distance to the auxiliary points makes the mapped coarse model attain the same tractable properties as when using gradient information (4.13) in the scalar case. On the other hand the number of points used is also the same as would be required to establish a central difference approximation to f'. We remark that, contrary to what would be expected, the usefulness of the definition improved when increasing the distance to the auxiliary points. A similar property is observed in the vector function example presented next.

Figure 4.18 shows a contour plot for the vector function example. In the figure the steps are relatively long, because if small steps, $\|h\| \lesssim 5$, are used the contours are nearly identical with those presented in Figure 4.9, for the parameter extraction problem of the usual space mapping. Apparently the direction or the number of auxiliary steps have no influence on the shape of the contours for small steps.

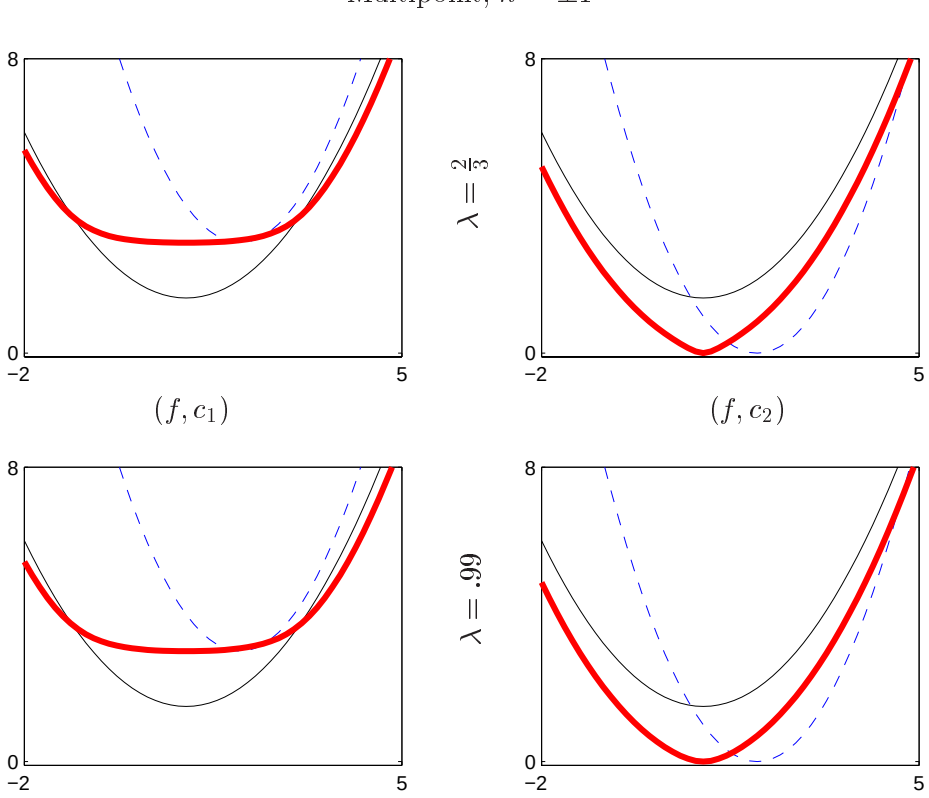


Figure 4.17: The plots show f (——), c (– –) and $c(p_\lambda(x))$ (——). In the left plots $c = c_1$, in the right plots $c = c_2$. In all plots p_λ, as defined in (4.14), is a mapping of similar responses using two auxiliary points, placed in the distances $h_1 = 1$ and $h_2 = -1$. In the top plots $\lambda = \frac{2}{3}$, in the bottom plots $\lambda = .99$.

Enlarging the distance between x and the auxiliary points improves on the uniqueness problem, though the results vary much with the actual position of the auxiliary points. In Figure 4.18 a contour plot of the multipoint parameter extraction problem is shown for $h_1 = (-10, 0)^T$ and $h_2 = (0, -10)^T$. The solution to the parameter extraction problem is close to one of the two local solutions of the usual space mapping definition (i.e. similar to what we found by parameter extraction using gradient information), and relative far from $\tilde{z}$, the saddlepoint. However as the solution is quite sensitive to changes in h_1 and h_2, we cannot from this vector example recommend this definition of the

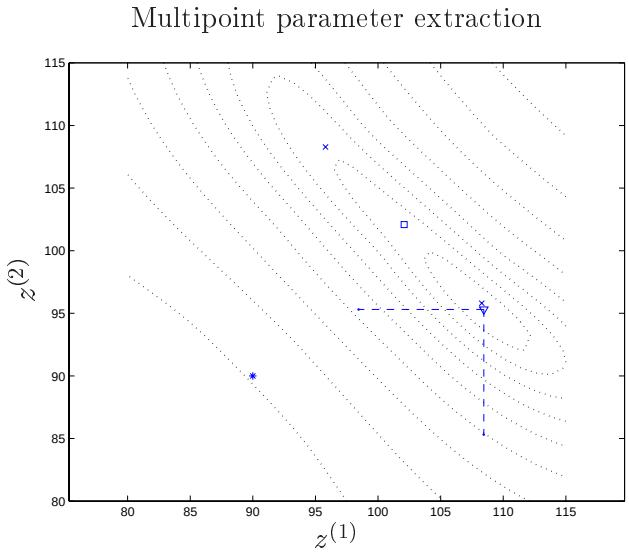


Figure 4.18: Contour plot of the multipoint parameter extraction (4.14) *for* $\lambda = .5$. *The unique minimizer* (∇) *is close to a local solution* ($\times$) *of the usual space mapping definition. Two auxiliary points* ($\cdot$) *are placed in* $h_1 = (-10, 0)^T$ *and* $h_2 = (0, -10)^T$.

space mapping.

For both the scalar and the vector case examples there is a significant effect of regularization of the parameter extraction problem if the steps are chosen long enough. Here the value of λ does not play as significant role as for the other problems, provided that it is not chosen very small. There is a need for further exploration of this definition, maybe a strategy involving second order information as in [2] is the best way, though we cannot tell from this example. Compared to the definition using gradient information, this multipoint definition seems more appealing, as it is easier to control how much and what information should be sampled from f.

4.2.5 Summary

In the preceding sections we have presented four alternatives to the usual space mapping definition (4.8). We have illustrated some characteristics of the four alternative space mapping definitions by two recurring test examples. None of the alternatives turned out exceptionally better than the other, so we cannot conclude that a particular definition is superior.

The gradient and multipoint space mapping definitions have both very nice properties in the one-dimensional scalar function case, provided that the regularization parameter is chosen properly. But, in the vector example the picture was not as clear. Here the definitions regularizing with distance to z^* and x were preferable, as they have unique solutions near $\tilde{z}$, the desired solution, for a properly chosen value of the regularization parameter λ.

The idea of including gradient information is intuitively appealing, as we in an optimization problem are looking for points where the derivatives vanish (stationary points). So being able to replicate the behaviour of the gradient of f using the mapped coarse model $c(p(x))$ would probably be sufficient to show that the conditions of theorem 4.1 are met. However, the definition used in this presentation could not present the desired properties in the vector example.

The definitive space mapping definition, which has the properties required by Theorem 4.1, has not yet been proposed. Further studies in this area are needed.

4.3 Approximation Error

We now turn the attention to the approximation ability of the space mapped coarse model $c(p(x))$ to the fine model. This subject is important both in context of modelling and optimization. In the former because one wants to accurately explore the behaviour of a system without the expense of many fine model evaluations. In the latter because in the search for an optimizer one needs to know in how large a region the mapped coarse model is a valid approximation of the fine model, e.g. for determining a good trust region size.

In practical space mapping optimization algorithms, as discussed in Chapter 5, the space mapping is approximated e.g. by a linear model. In the following we compare the theoretical approximation error of the mapped coarse model using a linear space mapping approximation with that of a classical Taylor based linear approximation of the fine model.

Consider the Taylor models

$$\begin{aligned} f(x+h) &= f(x) + f'(x)^T h + r_f(h) \\ c(p(x+h)) &= c(p(x) + p'(x)^T h + r_p(h)) \end{aligned}$$

where r_f and r_p are residual functions. We note that the cost of approximation $p'(x)$, e.g. by finite difference approximation, is around the same cost

as approximating $f'(x)$, assuming that the coarse model is cheap to evaluate compared to the fine model. So in the sense of approximation cost the two Taylor models are comparable.

Following Taylors theorem the residual functions are bounded in the following way

$$\begin{aligned} \|r_f(h)\| &\leqslant K_f \|h\|^2 \\ \|r_p(h)\| &\leqslant K_p \|h\|^2 \,, \end{aligned}$$

where $K_f, K_p \in I\!R^n$ are problem specific constants. So we can write

$$\|f(x+h) - (f(x) + f'(x)^T h)\| = \|r_f(h)\| \leqslant K_f \|h\|^2 \tag{4.15}$$

and

$$\begin{aligned} \|c(p(x+h)) - c(p(x) + p'(x)^T h)\| &\simeq \|c'(p(x))\| \, \|r_p(h)\| \\ &\leqslant K_p \, \|c'(p(x))\| \, \|h\|^2 \,. \end{aligned}$$

Now assume that the deviation between any given fine model response and the corresponding response of the mapped coarse model approximation is bounded,

$$\forall x \in I\!R^n : \|c(p(x)) - f(x)\| \leqslant \varepsilon \,,$$

with ε being a constant independent of x, then it follows that

$$\|f(x+h) - c(p(x) + p'(x)^T h)\| \lesssim \varepsilon + K_p \, \|c'(p(x))\| \, \|h\|^2 \,. \tag{4.16}$$

Comparing (4.15) and (4.16), we see that the space mapped model (with a linear Taylor model of the mapping) may provide a better approximation than the linear Taylor model of the fine model response if $K_p < K_f$ and if h is so large that $\varepsilon < K_f \|h\|^2$.

We now illustrate this conclusion by a numerical example. Consider again the TLT2 test problem described in Appendix A. We approximate the gradient of the space mapping by a finite difference approximation. To avoid wrong solutions in the parameter extraction problem, as discussed in Section 4.1.6 for this test problem, we employ the space mapping definition using gradient information in (4.13). For reference, the regularization parameter used in the example is $\lambda = 10^{-4}$.

In Figure 4.19 the approximation error of the Taylor model of the fine model response function (gray mesh), $\|f(x+h)-(f(x)+f'(x)^T h)\|$ from (4.15), and the approximation error of the mapped coarse model approximation (white mesh), $\|f(x+h)-c(p(x)+p'(x)^T h)\|$ from (4.16), are plotted for $x=z^*$, $z^*=(90,90)^T$ (the coarse model minimizer). From the figure it is apparent

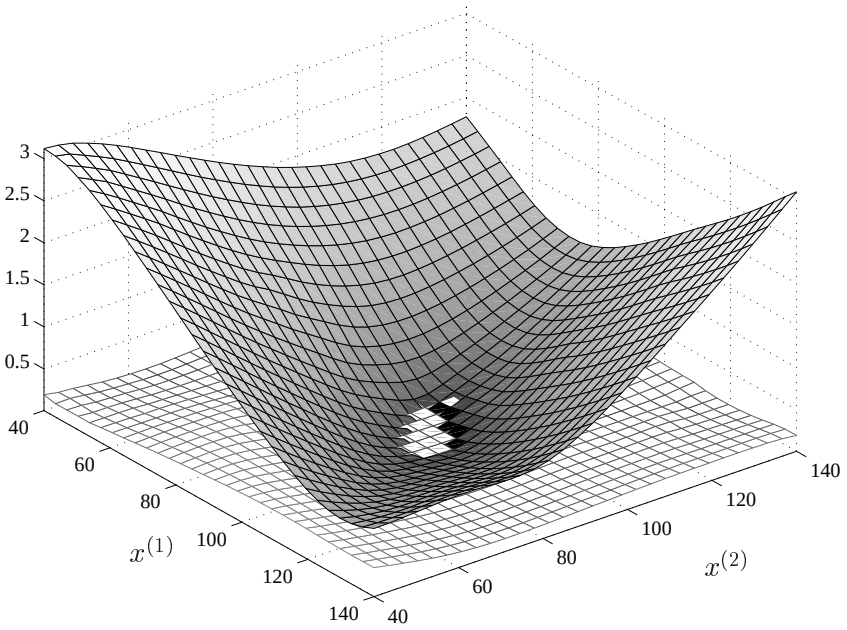


Figure 4.19: Approximation error plots for the TLT2 test problem at the coarse model optimizer $x=z^*$, $z^*=(90,90)^T$. Centered at $h=0$, the light grid shows $\|c(p(x)+p'(x)^T h)-f(x+h)\|_2$. This represents the deviation of the space mapped coarse model (using the linear Taylor approximation to the mapping) from the fine model. The dark grid shows $\|(f(x)+f'(x)^T h)-f(x+h)\|_2$. This is the deviation of the fine model from its classical linear Taylor approximation. It is seen that the Taylor approximation is most accurate close to z^* whereas the mapped coarse model approximation is best over a large region.

that the approximation error of the linear Taylor model of the fine model response grows quadratically with the length of the step $\|h\|$ from the model origin. The approximation of the mapped coarse model approximation does not exhibit the same systematic growth with distance from the model origin.

This is partly due to the response values of c and f being bounded in the interval from zero to one, and partly due to the small area of parameter values considered in Figure 4.19.

In Figure 4.20 we again consider the approximation error of the mapped coarse model approximation, $\|f(x+h)-c(p(x)+p'(x)^T h)\|$, but here on a much larger region of the parameter space. From this plot it is clear that the approximation error does in fact grow with distance from the origin of the Taylor model, as it is stated in (4.16).

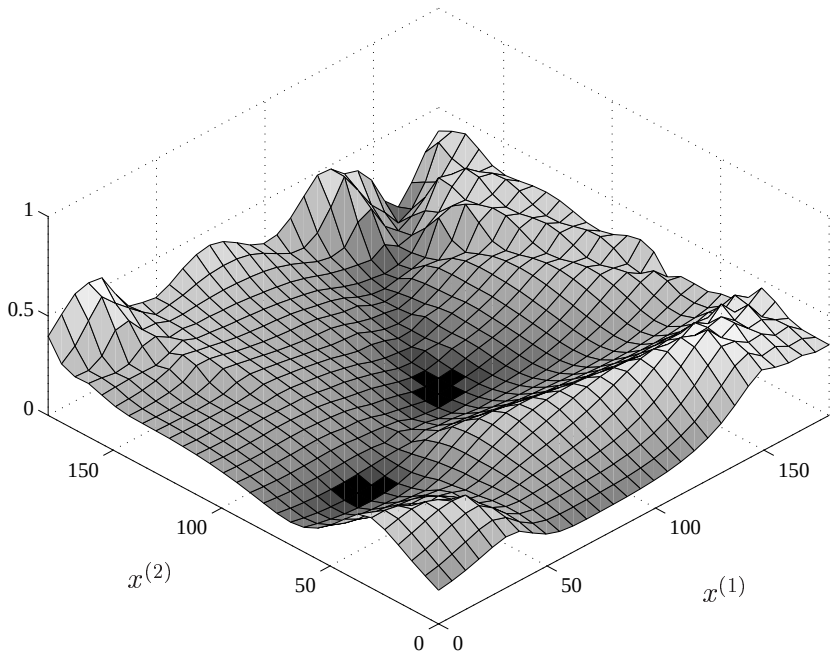


Figure 4.20: Approximation error plot for the TLT2 test problem at the coarse model optimizer $x = z^*$, $z^* = (90, 90)^T$. Centered at $h = 0$, the grid shows $\|c(p(x) + p'(x)^T h) - f(x + h)\|_2$. This represents the deviation of the space mapped coarse model (using the linear Taylor approximation to the mapping) from the fine model. It is seen that the error grows with the distance $\|h\|_2$ from the Taylor model origin.

It is evident from these figures that the mapped coarse model approximation is preferable for large steps, whereas the Taylor model of the fine model response

is preferable for small steps, due to the interpolation property of this model.

To further justify the mapped coarse model approximation, we compare the results with the coarse model without the space mapping. In Figure 4.21 we show the approximation error of the mapped coarse model approximation, $\|f(x+h)-c(p(x)+p'(x)^T h)\|$, and the approximation error of the coarse model (without the space mapping) $\|f(x+h)-c(x+h)\|$. The figure verifies two

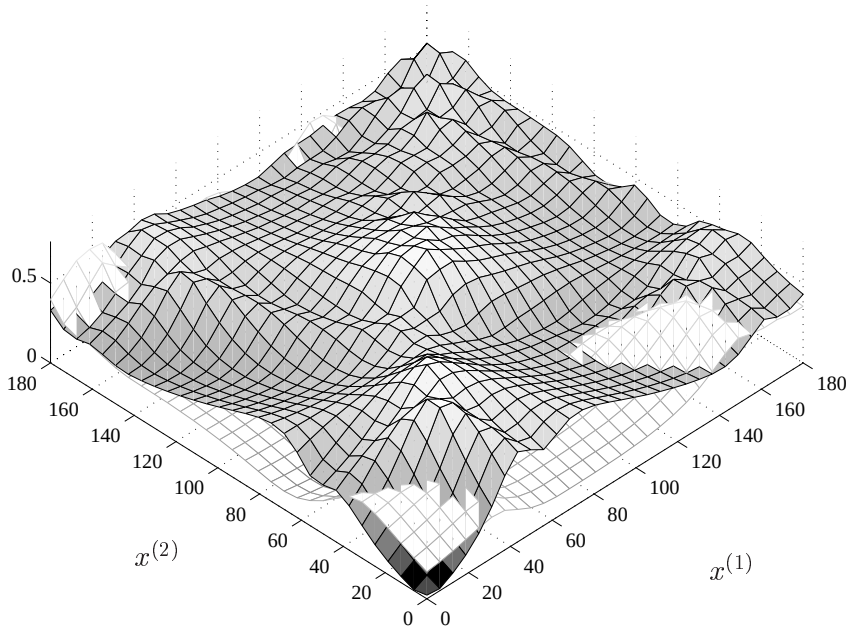


Figure 4.21: Approximation error plots for the TLT2 test problem at the coarse model optimizer $x = z^*$, $z^* = (90, 90)^T$. Centered at $h = 0$, the light grid shows $\|c(p(x) + p'(x)^T h) - f(x+h)\|_2$. This represents the deviation of the space mapped coarse model (using the linear Taylor approximation to the mapping) from the fine model. The dark grid shows $\|c(x+h) - f(x+h)\|_2$. This is the deviation of the fine model from the coarse model. It is seen that the mapped coarse model approximation is most accurate close to the origin of the Taylor model, whereas the coarse model is best very far from the origin of the Taylor model.

things. First, the figure shows that the mapped coarse model approximation indeed is better than the coarse model around the origin of the Taylor model

of the mapping. We expected this, as the space mapping at the origin of the model minimizes the deviation between the fine and the coarse model responses. Second, the figure shows that the linear approximation to the space mapping is valid for a large region, so the good agreement around the origin of the model is retained also at large distances. Hence the mapped coarse model approximation is not significantly worse for approximation at large distances than the (unmapped) coarse model. Further, these observations show that the constant K_p, from (4.16), is smaller than the constant K_f, from (4.15), so in that sense the space mapping is closer to linear than the fine model response, for the considered problem. This particular condition, that K_p is less than K_f and thereby p is closer to linear than f, very likely determines if we may benefit from the space mapping method on a particular problem. However, for practical, expensive functions we are not able to resolve if this condition is fulfilled or not.

At last we consider the method of response correction, presented in Chapter 2 (Section 2.3.1), applied to the mapped coarse model approximation. By using this method we may develop a model which has the interpolation property of the Taylor model of the fine model, but at the same time retains the nice properties of the mapped coarse model approximation for large steps.

The mapped coarse model approximation with corrected responses is

$$g .* \left[c(p(x+h)) - c(p(x))\right] + f(x)\,,$$

where $g \in \mathbb{R}^m$ are the correction factors and $*$ is element-wise multiplication. The correction factors are found by the secant update

$$g_j = \frac{f_j(x+\tilde{h}) - f_j(x)}{c_j(p(x+\tilde{h})) - c_j(p(x))}\,, \quad j = 1, \ldots, m\,,$$

where $\tilde{h}$ is a step to an auxiliary point, which has to be chosen somehow. We define that the update should only be applied to those responses where a significant change occurs from x to $x+\tilde{h}$. For those responses where there are not a significant change the correction factors are set to one.

As an example, consider the TLT2 test problem with $x^* = (74.2332, 79.2658)^T$ (rounded) as the auxiliary point, i.e. $\tilde{h} = x^* - x$, then we obtain the correction factors given in Table 4.4. From the table we see that the correction factors only deviate little from one. So the response correction should not substantially change the properties of the mapped coarse model approximation, as we verify below.

Response no.	Correction factor
1	0.9674
2	1.2075
3	1.0700
4	1.0317
5	1.1224
6	0.8455
7	0.9493
8	1.1041
9	1.0326
10	1.1095
11	1.0254

Table 4.4: Response correction factors using responses from the points $(90, 90)^T$ *and* $(74.2332, 79.2658)^T$*. Numbers are rounded to four decimals.*

Figure 4.22 shows the two-norm of the error residual of the mapped coarse model approximation to the fine model (white mesh) and the corrected mapped coarse model approximation to the fine model (gray mesh). As the corrected model response interpolates the fine model response at $h = 0$, the approximation error of the corrected model is smaller than the uncorrected mapped model in a region around $h = 0$. Further, we see that there is not introduced a significantly higher level of error further away from the interpolation point.

4.3.1 Summary

In this section we have presented a comparison of the theoretical approximation error between a linear Taylor models of the fine model and a mapped coarse model with a linear Taylor model of the space mapping. The theoretical results were illustrated on a numerical test problem.

The theory and the example showed how the mapped coarse model approximation in general does not interpolate the fine model. But on the other hand, the approximation error of the mapped coarse model approximation is considerably smaller than the corresponding Taylor model of the fine model for large steps from the model origin.

Further we illustrated the effect of applying a response correction method on the mapped coarse model approximation. For the numerical text example

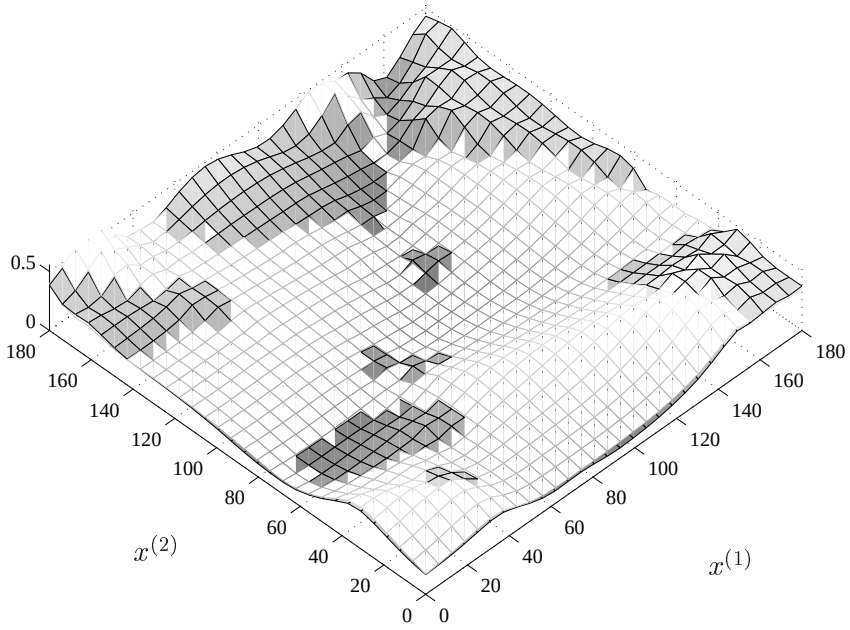


Figure 4.22: Approximation error plots for the TLT2 test problem at the coarse model optimizer $x = z^*$, $z^* = (90, 90)^T$. Centered at $h = 0$, the light grid shows $\|c(p(x) + p'(x)^T h) - f(x + h)\|_2$. This represents the deviation of the space mapped coarse model (using the linear Taylor approximation to the mapping) from the fine model. The dark grid shows $\|g .* \big(c(p(x) + p'(x)^T h) - c(p(x))\big) + f(x) - f(x + h)\|_2$. This is the deviation of the corrected mapped model from the fine model. It is seen that the corrected mapped coarse model approximation is most accurate close to z^*. Whereas the (uncorrected) mapped coarse model approximation is best further away from z^*.

it was found that the corrected model interpolates at the model origin and at the same time retains the good properties of the mapped coarse model approximation for long steps.

4.4 Conclusion

This chapter presented theoretical results which characterize the space mapping under some ideal conditions. We have shown that if these conditions are met, the solutions provided by the original space mapping technique are minimizers of the fine model response. The theoretical results were motivated and illustrated by three numerical examples displaying the image of the space mapping.

Deficiencies of the usual space mapping definition were discussed and four alternative definitions were reviewed. Some of the characteristics of the space mapping alternative definitions were illustrated by two numerical examples. From this presentation, the two space mapping definitions relying on respectively gradient information and multiple points were the most promising. But further theoretical investigations are needed in order to arrive at a more firm conclusion.

As the last part of the chapter we discussed approximation abilities of the coarse model composed with the space mapping. The theoretical approximation error was illustrated by a numerical example. The example confirmed the theoretical results, that the mapped coarse model, with a Taylor approximation to the space mapping, has a lower approximation error for long steps, compared to a Taylor model of the fine model. For short steps however, the Taylor model of the fine model is best, due to exact interpolation at the model origin. It was also shown how a response correction may enhance the mapped coarse model approximation, without compromising the small approximation error on long steps. With the response correction, the mapped coarse model approximation has the same interpolation property as the Taylor model of the fine model.

Symbols

$\|\cdot\|$	unspecified norm
$\|\cdot\|_2$	Euclidean norm, $\|x\|_2 = (x^T x)^{\frac{1}{2}}$
$\|\cdot\|_F$	Frobenious norm
c	response from the coarse model, $c : \mathbb{R}^n \mapsto \mathbb{R}^m$
d	Euclidean distance from point to set, $d(u, V) = \inf_{v \in V} \|u - v\|_2$
f	response from the fine model, $f : \mathbb{R}^n \mapsto \mathbb{R}^m$
g	response correction factors
H	convex function, used as merit function
m	number of response functions
n	dimensionality of the design parameter space
p	space mapping
p_λ	regularized space mapping
x	optimizeable model parameters of f and c
x^*	minimizer of $H(f(x))$
X^*	all minimizers of $H(f(x))$
x^*_{cop}	minimizer of $H(c(p(x)))$
X^*_{cop}	all minimizers of $H(c(p(x)))$
x^*_p	minimizer of $d(p(x), Z^*)$
X^*_p	all minimizers of $d(p(x), Z^*)$
z^*	minimizer of $H(c(z))$
Z^*	all minimizers of $H(c(z))$
λ	regularization parameter in space mapping definitions

References

[1] M.H. Bakr, J.W. Bandler, R. Biernacki, S. Chen, K. Madsen, *A Trust Region Aggressive Space Mapping Algorithm for EM Optimization*, IEEE Trans. Microwave Theory Tech., vol. 46, no. 12, pp. 2412–2425, 1998.

[2] M.H. Bakr, J.W. Bandler, N. Georgieva, *An Aggressive Approach to Parameter Extraction*, IEEE Trans. Microwave Theory Tech., vol. 47, pp. 2428–2439, 1999.

[3] M.H. Bakr, J.W. Bandler, K. Madsen, J.E. Rayas-Sánchez, J. Søndergaard, *Space Mapping Optimization of Microwave Circuits Exploiting Surrogate Models*, IEEE Trans. Microwave Theory Tech., vol. 48, no. 12, pp. 2297–2306, 2000.

[4] M.H. Bakr, J.W. Bandler, K. Madsen, J. Søndergaard, *Review of the Space Mapping Approach to Engineering Optimization and Modelling*, Optimization and Engineering, vol. 1, no. 3, pp. 241–276, 2000.

[5] J.W. Bandler, R.M. Biernacki, S.H. Chen, *Fully Automated Space Mapping Optimization of 3D Structures*, IEEE MTT-S IMS Digest, San Francisco, CA, pp. 753–756, 1996.

[6] J.W. Bandler, R.M. Biernacki, S.H. Chen, P.A. Grobelny, R.H. Hemmers, *Space Mapping Technique for Electromagnetic Optimization*, IEEE Trans. Microwave Theory Tech., vol. 42, pp. 2536–2544, 1994.

[7] J.W. Bandler, R.M. Biernacki, S.H. Chen, R.H. Hemmers, K. Madsen, *Electromagnetic Optimization Exploiting Aggressive Space Mapping*, IEEE Trans. Microwave Theory Tech., vol. 43, pp. 2874–2882, 1995.

[8] J.W. Bandler, R.M. Biernacki, S.H. Chen, Y.F. Huang, *Design Optimization of Interdigital Filters using Aggressive Space Mapping and Decomposition*, IEEE Trans. Microwave Theory Tech., vol. 45, pp. 761–769, 1997.

[9] J.W. Bandler, Q. Cheng, S. Dakroury, A.S. Mohamed, M.H. Bakr, K. Madsen, J. Søndergaard, *Space Mapping: The State of the Art*, submitted, IEEE Trans. Microwave Theory Tech., 2004.

[10] J.W. Bandler, A.S. Mohamed, M.H. Bakr, K. Madsen, J. Søndergaard, *EM-Based Optimization Exploiting Partial Space Mapping and Exact Sensitivities*, IEEE Trans. Microwave Theory Tech., vol. 50, no. 12, pp. 2741–2750, 2002.

[11] J. Søndergaard, *Non-Linear Optimization Using Space Mapping*, Master Thesis, IMM-EKS-1999-23, Informatics and Mathematical Modelling, Technical University of Denmark, 1999.
Available at `http://www.imm.dtu.dk/~km/jsmaster.ps.gz`

[12] L.N. Vicente, *Space Mapping: Models, Sensitivities and Trust-region Methods*, to appear in Optimization and Engineering, 2003.

Chapter 5

Space Mapping Optimization Algorithms

This chapter concerns space mapping optimization algorithms. Some of the algorithms proposed here are implemented in a Matlab toolbox described in Appendix A. The toolbox also contains a number of test problems, which are used in this chapter for reporting on the numerical performance of the algorithms.

The chapter is divided into four sections. First the notation is introduced. Thereafter, Section 5.2 is concerned with formulations of space mapping optimization algorithms. Section 5.3 describes numerical tests of the algorithms implemented in the toolbox on the test problems, also in the toolbox. The last section contains a summary and conclusions of the chapter.

5.1 Introduction

The main problem addressed by the optimization algorithms presented in this chapter is finding a solution to

$$\min_{x \in \mathbb{R}^n} H(f(x)) \, . \tag{5.1}$$

Where $H : \mathbb{R}^m \mapsto \mathbb{R}$ is a convex function, usually a norm, and $f : \mathbb{R}^n \mapsto \mathbb{R}^m$ is a vector function, being a response function representing some model. We assume that f is bounded below and uniform continuously differentiable. A minimizer of (5.1) is denoted x^*.

The *classical methods* for solving problems of the type (5.1) are based on first or second order Taylor models. General theory and algorithms for this type of problem are presented in [12], with special emphasis on the minimax and L_1 cases. The special case where H is the ℓ_2-norm has been studied extensively and is e.g. treated in [10].

The focus of this work is the case where f is so expensive to evaluate that direct optimization with a classical method is not feasible. To indicate the expensive nature we denote the model from which f originates the *fine model*. Hence x^* is a *fine model minimizer*.

In this setting alternative methods for solving (5.1) must be considered. This work deals with space mapping optimization techniques which solve a sequence of subproblems in which f is replaced by a cheaper surrogate function s. The surrogate is re-calibrated during iterations by scarce evaluation of f. The space mapping technique was introduced in [4]; the technique is reviewed in [3] and [6].

The space mapping technique assumes the existence of a so-called *coarse model* related to the fine model in some way. The coarse model is represented by the response function $c : \mathbb{R}^n \mapsto \mathbb{R}^m$. The function c is assumed bounded below and uniform continuously differentiable. It is assumed that c is cheaper to evaluate than f, and therefore it is most likely less accurate than f. We will denote a solution to the problem

$$\min_{z \in \mathbb{R}^n} H(c(z)) \ , \tag{5.2}$$

by z^* and use the term a *coarse model minimizer*. We assume that c is so cheap to evaluate that the gradient is readily available, e.g. by finite difference approximation.

The space mapping technique employs the coarse model in the search for a minimizer of the fine model. This is done through a parameter mapping, the so-called *space mapping* $p : \mathbb{R}^n \mapsto \mathbb{R}^n$, which defines a mathematical link between the fine model and the coarse model parameter spaces.

The usual definition of the space mapping is,

$$p(x) \in \arg\min_{z \in \mathbb{R}^n} \|c(z) - f(x)\|_2 \ , \tag{5.3}$$

hence connecting similar model responses. The problem defining the space mapping is often referred to as the *parameter extraction* problem.

Other definitions of the space mapping have been proposed, mainly to avoid problems with nonunique solutions in (5.3). Without further motivation, we state here the space mapping definitions which are implemented in the toolbox, refer to Chapter 4 for a more in-depth treatment of this subject.

Regularization with regard to the distance to z^*,

$$p_\lambda(x) = \arg\min_{z\in\mathbb{R}^n} \left\{ (1-\lambda)\,\|c(z)-f(x)\|_2^2 + \lambda\,\|z-z^*\|_2^2 \right\}, \tag{5.4}$$

for some value of $0 \leqslant \lambda < 1$.

Regularization with regard to the distance to x,

$$p_\lambda(x) = \arg\min_{z\in\mathbb{R}^n} \left\{ (1-\lambda)\,\|c(z)-f(x)\|_2^2 + \lambda\,\|z-x\|_2^2 \right\}, \tag{5.5}$$

for some value of $0 \leqslant \lambda < 1$.

Regularization using gradient information,

$$p_\lambda(x) = \arg\min_{z\in\mathbb{R}^n} \left\{ (1-\lambda)\,\|c(z)-f(x)\|_2^2 + \lambda\,\|c'(z)-f'(x)\|_F^2 \right\}, \tag{5.6}$$

for some value of $0 \leqslant \lambda < 1$. In the optimization algorithms the Jacobian matrix $f'(x)^T$ is approximated by a secant approximation $D \in \mathbb{R}^{m\times n}$ during iterations, so D^T is used instead of the gradient in (5.6).

In the implementation the above regularized problems are solved as normal nonlinear least-squares problems, exemplified here by (5.4),

$$p_\lambda(x) = \arg\min_{z\in\mathbb{R}^n} \left\| \begin{array}{c} \sqrt{(1-\lambda)}\,(c(z)-f(x)) \\ \sqrt{\lambda}\,(z-z^*) \end{array} \right\|_2^2, \tag{5.7}$$

for some value of $0 \leqslant \lambda < 1$. As the Jacobian of c is assumed available, the gradient for this least-squares objective function is available, at least for (5.4) and (5.5). In the case of (5.6) though, the gradient of the least-squares objective function depends on the second derivatives of c. So, as second order information usually is not explicitly available, the gradient of the least-squares objective function may be found by finite difference approximation.

With this introduction we are now prepared to discuss details of the space mapping optimization algorithms, which we present in the following.

5.2 Space Mapping Optimization Algorithms

We now introduce five optimization algorithms which are based on the space mapping technique. We assume that the reader is familiar with trust region terminology, at least on the level of the presentation in Chapter 2. A thorough introduction to trust region methodology is found in [8].

All algorithms we present are trust region algorithms employing a linear Taylor model of the space mapping obtained from a secant approximation. Some of the algorithms, the hybrid algorithms, also rely on a linear Taylor model of the fine model obtained from a secant approximation.

We start with two algorithms that are related to the original space mapping technique. Thereafter we present three hybrid algorithms which combine the original space mapping technique with classical optimization methods.

5.2.1 Original Space Mapping Algorithms

In the following we present two algorithms related to the original formulation of the space mapping.

The space mapping method was introduced in [4, 5] as the problem of solving the nonlinear equations

$$p(x) = z^* \tag{5.8}$$

for $x \in \mathbb{R}^n$. Define the residual function

$$r(x) = p(x) - z^* \,,$$

then solutions of (5.8) are contained in the solutions of the least-squares formulation of the problem

$$\min_{x \in \mathbb{R}^n} \|r(x)\|_2 \,. \tag{5.9}$$

Since the gradient p' of the space mapping is not explicitly available, we solve the problem using a secant method. See e.g. [10] for a thorough treatment of secant methods.

We now present an algorithm based on a secant method for solving (5.9).

The secant method involves approximating the space mapping p by a linear Taylor model. Assume we are at the kth iterate $x_k \in \mathbb{R}^n$ in the iteration process, then for a given step $h \in \mathbb{R}^n$, the Taylor model is

$$p(x_k + h) \simeq p_k(h) \equiv B_k h + p(x_k) \,, \tag{5.10}$$

where the matrix $B_k \in \mathbb{R}^{n\times n}$ is an approximation to $p'(x_k)^T$, obtained from a secant formula. In practice we use Broyden's rank one update

$$B_{k+1} = B_k + \frac{p(x_k + h) - p(x_k) - B_k h}{h^T h} h^T \,.$$

We define the initial approximation B_0 to be the identity matrix, $B_0 = I(n)$, corresponding to the (initial) assumption that the coarse model is identical to the fine model, see Chapter 3.

From (5.10) we obtain the Taylor model of r,

$$r(x_k + h) \simeq r_k(h) \equiv B_k h + r(x_k) \,, \tag{5.11}$$

where B_k is as above and $r(x_k) = p(x_k) - z^*$.

The next tentative step $h_k \in \mathbb{R}^n$ of the algorithm solves

$$h_k \in \arg\min_h \|r_k(h)\|_2 \,, \tag{5.12}$$

subject to a trust region, $\|h_k\| \leqslant \Lambda_k$, where $\Lambda_k > 0$ is current size of the trust region and $\|\cdot\|$ is a suitable norm in $\mathbb{R}^n$.

The acceptance criteria and control of the trust region size is determined in the conventional way for trust region methods, see [14]:

The *predicted decrease* by the norm of the Taylor model is

$$\text{predicted decrease} = \|r(x_k)\|_2 - \|r_k(h_k)\|_2 \,.$$

The *actual decrease* measured in the norm of the residual is

$$\text{actual decrease} = \|r(x_k)\|_2 - \|r(x_k + h_k)\|_2 \,.$$

The acceptance criteria of the tentative step h_k are

$$\text{predicted decrease} \;>\; 0 \,, \tag{5.13}$$

$$\frac{\text{actual decrease}}{\text{predicted decrease}} \;\geqslant\; \delta_1 \,, \tag{5.14}$$

for $0 < \delta_1 < 1$. The criterion (5.13) is always satisfied when h_k is a solution to solving (5.12) except if $h_k = 0$, in which case the step is rejected. The criterion (5.14) accepts the tentative step if there is a sufficient actual decrease relative to the predicted decrease. Usually δ_1 is chosen quite small e.g. $\delta_1 = 10^{-4}$.

The trust region size Λ is adjusted as follows: If the ratio of the actual decrease to the predicted decrease is less than or equal to a constant δ_2, $\delta_1 < \delta_2 < 1$,

$$\frac{\text{actual decrease}}{\text{predicted decrease}} \leqslant \delta_2 \,, \tag{5.15}$$

the trust region size is reduced by a factor K_1, $0 < K_1 < 1$,

$$\Lambda_{k+1} = K_1 \Lambda_k \,.$$

If the ratio is greater than or equal to a constant δ_3, $\delta_2 < \delta_3 < 1$,

$$\frac{\text{actual decrease}}{\text{predicted decrease}} \geqslant \delta_3 \,, \tag{5.16}$$

the trust region size is enlarged by a factor K_2, $K_2 > 1$,

$$\Lambda_{k+1} = K_2 \Lambda_k \,.$$

If neither (5.15) nor (5.16) are satisfied, the trust region size stays unchanged, $\Lambda_{k+1} = \Lambda_k$.

The complete algorithm is summarized in Algorithm 5. This algorithm is im-

Algorithm 5 Original Space Mapping Optimization

Given Λ_0, $B_0 = I$, $k = 0$
0. Find x_0 as solution to $\min_{x \in \mathbb{R}^n} H(c(x))$
1. Evaluate $f(x_0)$
2. Find $p(x_0)$
3. Find h_k by solving (5.12)
4. Evaluate $f(x_k + h_k)$
5. Find $p(x_k + h_k)$ by solving (5.3)
6. Let $x_{k+1} = x_k + h_k$ if (5.13) and (5.14) both are true; otherwise $x_{k+1} = x_k$
7. Update Λ, B
8. Let $k = k + 1$
9. If not converged, goto 3

plemented by the name `smo` in the toolbox, refer to Appendix A. If the trust region is measured using the L_∞-norm, then the subproblem (5.12) is a quadratic programming problem, which can be solved using a standard QP-solver. This is the strategy used in the toolbox. When measuring the trust region in the the L_2-norm, an approach due to Moré and Sorensen can be used, see e.g. [16].

Notice that we use the coarse model minimizer z^* as the starting point x_0, $x_0 = z^*$. This is in accordance with the initial assumption that the coarse and the fine model are identical.

The choice of starting point is characteristic for all space mapping algorithms, and it often shows in practice that having a good starting point is the key advantage of the space mapping methods. We will return to this issue in Section 5.3 discussing the numerical tests.

The convergence criteria referred to in step 9 of the algorithm are implemented in the toolbox as follows:

The algorithm should stop if the length of the tentative step found in step three is less than a threshold,

$$\|h_k\| \leqslant \varepsilon(1 + \|x_k\|) \,, \tag{5.17}$$

where $\varepsilon > 0$ is user defined. We use a threshold value scaled with the norm of the current iterate in order to avoid problems with bad scaling of the parameters.

The algorithm should stop if the number of fine model function evaluations exceeds a threshold,

$$k \geqslant k_{max} \,, \tag{5.18}$$

where $k_{max} > 0$ is user defined.

There are other possible stopping criteria that could have been implemented, for example: Stopping if the objective function gets below a user defined threshold value. Stopping if the norm of the gradients of the objective function value gets below a user defined threshold value. We do not use any of these criteria for two reasons. First, a user cannot in general be expected to define suitable threshold values, which will be very problem dependent. Second, the gradient information obtained by the algorithms originates from a secant method, thus the gradient information may be very inaccurate, and it is therefore not reliable for use in a stopping criterion.

New Space Mapping Formulation

In [20] and also in the paper included as Chapter 3 a new formulation of the original space mapping formulation (5.8) was suggested, namely to minimize the so-called *mapped coarse model* $c \circ p$. In Chapter 4 it is shown that under certain conditions, solving

$$\min_{x \in \mathbb{R}^n} H(c(p(x))) \tag{5.19}$$

is equivalent to solving (5.8). Further, in Chapter 4 it was argued that (5.19) was to be preferred over (5.8) in situations where the solutions of the two formulations do not coincide.

We now describe a trust region algorithm for solving (5.19).

The algorithm relies on the linear Taylor model of the space mapping (5.10). So the algorithm finds the next tentative step $h_k \in \mathbb{R}^n$ as a solution to

$$h_k \in \arg\min_h H(c(p_k(h))) \ , \tag{5.20}$$

subject to a trust region, $\|h_k\| \leqslant \Lambda_k$.

Since c is assumed to be cheap, the nonlinear trust region subproblem (5.20) can be solved by a classical Taylor based trust region algorithm as one of those described in [12].

The *predicted decrease* by the coarse model with the space mapping approximation is

$$\text{predicted decrease} = H(c(p(x_k))) - H(c(p_k(h_k))) \, .$$

The *actual decrease* measured in the mapped coarse model response is

$$\text{actual decrease} = H(c(p(x_k))) - H(c(p(x_k + h_k))) \, .$$

The acceptance criteria of a tentative step h_k and the control of the trust region size follows the same general strategy as presented for the trust region algorithm above. The same goes for the stopping criteria.

The algorithm is identical with Algorithm 5 presented above, except for the following: the tentative step is found by solving the trust region subproblem (5.20), and the definitions of the predicted and actual decrease are the ones above. This algorithm is implemented by the name `smon` in the toolbox, refer to Appendix A.

5.2.2 Hybrid Space Mapping Algorithms

We now turn the attention to a class of hybrid space mapping algorithms which rely on the space mapping technique as means of obtaining a good starting point for a classical Taylor based optimization technique. Hence, the algorithms use the space mapping technique as a pre-conditioner for the optimization problem. Before we state the hybrid algorithms, we briefly go over the classical technique for solving (5.1).

In the classical Taylor based optimization a Taylor model is developed at the current iterate x_k. We consider the linear Taylor model

$$f_k(h) = D_k h + f(x_k) \,, \tag{5.21}$$

where D_k is the Jacobian matrix $f'(x_k)^T$ or an approximation to it. The algorithm finds $h_k \in \mathbb{R}^n$ as a solution to

$$h_k \in \arg\min_h H(f_k(h)) \,,$$

subject to a trust region, $\|h_k\| \leqslant \Lambda_k$.

Basically the algorithm of the classic method is as Algorithm 5 above, though measuring decrease in $H \circ f_k$ and $H \circ f$ rather than $H \circ c \circ p_k$ and $H \circ c$.

Now returning to the hybrid space mapping algorithms. The first ideas of actual hybrid algorithms, combining elements of space mapping and Taylor based classical techniques, were presented in [1, 20].

In [1] the main focus is on finding an x which makes the fine model response $f(x)$ match the optimal coarse model response $c(z^*)$, the target response. Thus the algorithm does not in general provide a solution of (5.1), except if certain special conditions are satisfied, see Chapter 4.

In [20] an algorithm is presented that explicitly addresses the problem in (5.1). The same algorithm is motivated and discussed in Chapter 3 (the included paper [2]). This algorithm relies on a so-called *combined model* s_k, which is a convex combination of $c(p_k(h))$ and $f_k(h)$,

$$s_k(h) = w_k \, c(p_k(h)) + (1 - w_k) \, f_k(h) \tag{5.22}$$

where $0 \leqslant w_k \leqslant 1$ is a transition parameter. The idea is to gradually switch during iterations from the mapped coarse model to the Taylor model of the fine model. In Chapter 3 this is done by letting w_k vanish during iterations,

$w_k = o(1)$, where $o(1) \to 0$ for $k \to \infty$. The actual updating formula suggested in Chapter 3 is to halve the value of w in every tentative step that is not accepted.

The formulation (5.22) is a special case of the more general formulation of the combined model,

$$s_k(h) = w_k\, e(x) + (1 - w_k)\, f_k(h)\,, \tag{5.23}$$

where $e : I\!R^n \mapsto I\!R^m$ is a vector function that is intended to have a preconditioning effect on the problem (5.1) and at the same time is cheaper to evaluate than f. In our context e of course is related to the space mapping in some way.

In Chapter 6 convergence is proved for a class of algorithms based on the model (5.23) where D_k of f_k, see (5.21), is equal to the exact Jacobian matrix $f'(x_k)^T$ or a finite difference approximation to it. The main assumption of the proof is on how the transition parameter is controlled, the assumption is

$$w_k = \min\{\Lambda_k, 1\} \cdot o(1)\,. \tag{5.24}$$

Hence the transition parameter should vanish at least as fast as the trust region size. This is a strengthened requirement compared to that suggested in Chapter 3.

We now present a hybrid space mapping algorithm for solving (5.1) using the combined model (5.22).

The next tentative step $h_k \in I\!R^n$ is found as solution to

$$h_k \in \arg\, \min_h H(s_k(h)) \tag{5.25}$$

subject to a trust region, $\|h_k\| \leqslant \Lambda_k$. The nonlinear trust region subproblem (5.25) can be solved by a classical Taylor based trust region algorithm as one of those described in [12].

The *predicted decrease* by the combined model with the space mapping approximation is

$$\text{predicted decrease} = H(f(x_k)) - H(s_k(h_k))\,.$$

Notice here that we measure the predicted decrease according to the fine model response at the current iterate. This is because we want convergence to the

fine model optimizer x^*, refer to Chapter 6. The model $s_k(x_k)$ does not in general interpolate $f(x_k)$, except if $w_k = 0$.

The *actual decrease* measured in the fine model response is

$$\text{actual decrease} = H(f((x_k)) - H(f(x_k + h_k))\,.$$

We assume that w_k, $k = 0, 1, \ldots$, is determined such that the condition (5.24) is met. Further, the acceptance criteria of a tentative step h_k and the control of the trust region size follow the same general strategy as presented for the trust region algorithms above. The same is true for the stopping criteria.

The complete hybrid space mapping algorithm is summarized in Algorithm 6. Notice how we can obtain an initial approximation D to $f'(x_0)^T$ using the

Algorithm 6 Hybrid Space Mapping Algorithm

Given Λ_0, $B_0 = I$, $w_0 = 0$, $k = 0$
0. Find x_0 as solution to $\min_{x \in \mathbb{R}^n} H(c(x))$
1. Evaluate $f(x_0)$
2. Find $p(x_0)$
3. Set $D = c'(p(x_0))$
4. Find h_k by solving (5.25)
5. Evaluate $f(x_k + h_k)$
6. Find $p(x_k + h_k)$ by solving (5.3)
7. Let $x_{k+1} = x_k + h_k$ if (5.13) and (5.14) both are true; otherwise $x_{k+1} = x_k$
8. Update Λ, B, D, w
9. Let $k = k + 1$
10. If not converged, goto 4

coarse model, by letting $D = c'(p(x_k))^T B_k$ the first time D is to be used, i.e. the first iteration where $w_k < 1$. Thus we avoid the need for a finite difference approximation to the gradient of f. In the toolbox, the default is to initialize D as quickly as possible, hence as in step 3 of Algorithm 6. In the subsequent iterations D is updated using Broyden's rank one update

$$D_{k+1} = D_k + \frac{f(x_k + h_k) - f(x_k) - D_k h_k}{h_k^T h_k} h_k^T\,.$$

A necessary remark about the algorithm: We use the inexact gradient approximation D, even though convergence of this algorithm has only been proven for the case where D is the exact gradient f' (refer to Chapter 6 for the convergence proof). Vicente [21] proves convergence for a similar algorithm using

inexact gradient information in the case where H is a quadratic function, e.g. the L_2 norm.

In the following we present the three versions of this algorithm which are implemented in the toolbox. Thereafter we briefly discuss other related strategies for combining the space mapping method and classical optimization methods.

Linear Transition

The algorithm presented in Chapter 3, based on the combined model in (5.22), can easily be modified to fit the hybrid algorithm framework suggested above.

This is done by changing the update of the transition parameter w_k to the update formula suggested in Chapter 6,

$$w_{k+1} = K_3 w_k \min\{\Lambda_{k+1}, 1\}$$

where $0 \leqslant K_3 < 1$. This update formula is to be used each time a tentative step has been rejected, or if there have been n accepted steps (where the update formula has not been used). If the updated w_{k+1} is less than a threshold K_4, where $0 < K_4 < 1$, then w_{k+1} is set to zero.

So the modified algorithm of Chapter 3 is Algorithm 6 with the update strategy described above, e.g. with $K_3 = 0.5$. This algorithm is implemented by the name `smh` in the toolbox, refer to Appendix A.

Orthogonal Updating

A different algorithm with the hybrid structure was suggested by Pedersen in [17]. His algorithm uses the new formulation of the original space mapping method (see page 126) in at least the n first iterations, corresponding to $w_k = 1, k = 1, \ldots, n$. In these first n iterations: If the original space mapping method fails, i.e. an uphill step (in terms of $H \circ f$) is suggested, a random step of fixed length is taken, orthogonal to the previous step directions, in an effort to enhance the secant approximation of the gradient of the space mapping. After each orthogonal step a new step is taken using the original space mapping method. After n iterations the algorithm of [17] switches to a classical method right after the first step in which the original space mapping method fails. This is done by setting $w_k = 0$ at the next uphill step.

The algorithm we suggest is modified from the one in [17] on the following points:

- Orthogonal steps are constructed from coordinate directions, not random directions.
- Trust region is not updated when taking the orthogonal steps.
- For a given orthogonal direction d and fixed step length α, the step taken is $h = \pm\alpha d$, whichever has the best predicted objective function value by $H \circ c \circ p_k$.
- A maximum of $2 \cdot n$ steps with original space mapping method is allowed before setting $w_k = 0$.

With the special control of w_k and the possible use of orthogonal steps within the first n iterations, the algorithm is as Algorithm 6 above. The algorithm is implemented by the name `smho` in the toolbox, refer to Appendix A.

An alternative method for choosing the orthogonal steps has been suggested by Pedersen [18]: Let the orthogonal step $h \in \mathbb{R}^n$ be the solution of the constrained problem

$$\begin{aligned} h \quad = \quad & \arg\min_d H(c(p_k(d))) \\ & \text{s.t.} \quad \|d\| \leqslant \tilde{\Lambda} \\ & \qquad Q^T d = 0 \end{aligned}$$

where Q is an orthonormal basis for the previous steps and the maximum step length $\tilde{\Lambda} > 0$ may be chosen independently of Λ_k. In practice a lower bound on the step length may be needed as well, $\tilde{\Lambda}_2 \leqslant \|d\|$, $\tilde{\Lambda}_2 > 0$, to assure that $h = 0$ is not suggested.

The implementation in the toolbox provides the user with the choice of using the fixed length orthogonal steps (the default) or finding the orthogonal steps solving the constrained problem above.

Response Correction

The first order response correction technique described in Chapter 2 (Section 2.3.1) may be applied to the mapped coarse model $c \circ p$. By using this

method we may develop a model which has the interpolation property of the Taylor model f_k of the fine model response, but at the same time retains the nice properties of the mapped coarse model for large steps. This effect is illustrated by a numerical test problem in Chapter 4, Section 4.3.

The mapped coarse model with corrected responses is

$$g \,.*\, [c(p(x+h)) - c(p(x))] + f(x)\,, \tag{5.26}$$

where $g \in I\!R^m$ are the correction factors and $*$ is element-wise multiplication. For a given tentative step h_k, the correction factors are found by the secant update

$$[g_k]_j = \frac{f_j(x_k+h_k) - f_j(x_k)}{c_j(p(x_k+h_k)) - c_j(p(x_k))}\,, \quad j = 1,\ldots,m\,,$$

where only the correction factors for those responses with a significant change in value from x_k to $x_k + h_k$ are updated. For those responses where there is not a significant change the correction factors can either be retained at their previous value (if such exist) or set to one.

In the context of the hybrid space mapping algorithm, the combined model with the corrected mapped coarse model is

$$s_k(h) = w_k\ (g \,.*\, (c(p_k(h)) - c(p(x_k)) + f(x_k)) + (1-w_k)\, f_k(h)\ . \tag{5.27}$$

This combined model distinguishes itself from (5.22) by interpolating f at x_k, $s_k(0) = f(x_k)$. In that sense the model is closer to the original concept of trust region methodology.

The transition parameter w_k is updated using linear transition strategy presented above. So except for the modified combined model (5.27) and the need for updating g, the hybrid algorithm is otherwise as Algorithm 6 above. The algorithm is implemented by the name `smhc` in the toolbox, refer to Appendix A.

Another approach to the response correction technique [9] is to extend (5.26) to the form

$$A(c(p(x+h)) - c(p(x))) + f(x)\,,$$

where $A \in I\!R^{m\times m}$ is a matrix containing the correction coefficients. The matrix of correction coefficients may be found by a secant approximation, e.g. using Broyden's rank one update

$$A_{k+1} = A_k + \frac{f(x_k+h_k) - f(x_k) - A_k d_k}{d_k^T d_k} d_k^T\,,$$

where $d_k = c(p(x_{k+1})) - c(p(x_k))$. For the problems we consider m tends to be a rather large number, so it may not be feasible to store the full matrix A. However the limited memory techniques suggested in [7] may be applied, and thus reduce the need for storage to only a few vectors of length m.

It is not obvious how to justify the need for the off-diagonal elements of A, as these represent interaction between response functions. For this reason we have chosen not to include this extended response correction technique in the implementation in the toolbox.

Related Strategies

We briefly discuss two strategies related to the idea of combining space mapping methods and classical optimization methods.

Transition With Distance In the hybrid algorithms discussed so far, the transition parameter w_k vanishes as k increases. Another strategy is to let w_k be a function of distance $\|h\|$ from the current iterate x_k. For example by using a smooth transition function as sketched in Figure 5.1. In the example of the

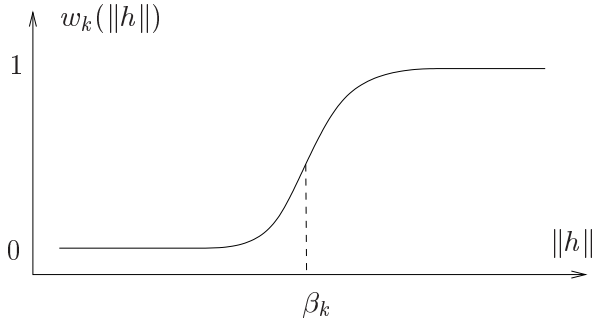


Figure 5.1: Sketch of a smooth transition function w_k depending on the length of the step $\|h\|$. The parameter β_k determines at which distance to the transition function takes the value 0.5. Other parameters could be included to describe the shape of the curve.

figure, β_k should be adjusted in order to meet the convergence condition (5.24).

Using such a strategy we have $w_k = 0$ for $\|h\| = 0$, hence the combined model interpolates f at x_k, $s_k(0) = f(x_k)$. For large enough steps we have $w_k = 1$, hence the combined model is the approximation of the new space mapping formulation, $s_k(h) = c(p_k(h))$. In this way we exploit the local Taylor model of the fine model response for small steps, and the space mapped coarse model for large steps.

Linear Correction Instead of the combined model, we may add a correction term to the mapped coarse model [9]. The correction term should model the residual $f(x) - c(p(x))$, i.e. the deviation between the fine model response and the mapped coarse model response. Let l denote the Taylor model of the correction term,

$$f(x+h) - c(p(x+h)) \simeq l(h) \equiv \tilde{D}h + f(x) - c(p(x)) \ ,$$

where $\tilde{D}$ is a secant approximation to $f'(x)^T - c'(p(x))^T p'(x)^T$. In fact, if we use the approximations from the combined model, we have $\tilde{D} = D - c'(p(x))^T B$.

So, the model to minimize in each trust region subproblem is

$$s_k(h) = c(p_k(h)) + l_k(h) \ .$$

The advantage of this formulation is that the model is interpolating at x_k. The main two disadvantages are: First, the mapped coarse model may not be continuously differentiable as the space mapping p in general is not continuous. Thus we risk that the correction term fluctuate even for small steps, and the algorithm may never converge. Second, the correction term may spoil the good properties of the mapped coarse model for the long steps. A modification to overcome the last disadvantage is to make the effect of the correction term local [19], e.g. by multiplying l_k with a weighting parameter that vanish with the length of the step, i.e. the opposite weighting function as that sketched in Figure 5.1.

5.3 Numerical Tests

This section reports numerical tests of the implementation of the algorithms described above. The implemented algorithms and the test problems are provided in the Matlab toolbox described in Appendix A.

We will only show results for the 7 test problems of the toolbox numbered 1–3, 5–8.

The test problems 4, 9 and 10 in the toolbox are not used, because they are so simple that it is not possible to compare the performance of the algorithms from these problems. These three problems were originally conceived in order to illustrate the deficiencies of the original space mapping method in a simple way.

The test problems numbered 11–13 in the toolbox, the electromagnetic simulator problems, are not used, because it has not been possible to demonstrate definite results for these problems. The two main difficulties with these problems are that the parameters are defined on a discrete grid, and that the response functions are highly sensitive for even the smallest possible change in parameter values.

Next we report on the numerical performance of the space mapping algorithms and compare to that of a classical optimization algorithm. Thereafter we illustrate the effects of using alternative space mapping definitions on one of the test problems. Finally, we examine the trajectory of the original space mapping methods and compare to that of a hybrid method.

5.3.1 Test Runs

We now consider the numerical performance of the space mapping algorithms and a classical optimization algorithm. All algorithms are implemented in the Matlab toolbox described in Appendix A.

Prerequisites

We will refer to the algorithms by the names and numbers listed in Table 5.1, see also the description of the algorithms in Section 5.2 above and in the toolbox documentation in Appendix A. For the convenience of the reader, we briefly restate some information about the algorithms here:

`smo` : original space mapping method, described at page 124.

`smon` : new formulation of the original space mapping method, described at page 126.

`smh` : hybrid space mapping algorithm with linear transition, described at page 130.

`smho` : hybrid space mapping algorithm with orthogonal updating, described at page 130.

`smhc` : hybrid space mapping algorithm with response correction, described at page 131.

`direct` : direct, classical Taylor based optimization. The algorithm is equivalent to the hybrid space mapping algorithms having $w_k = 0$ for all k, except that the initial approximation to the Jacobian of the fine model is found by finite difference approximation.

Algorithm no.	Name in toolbox	Marker in Figure 5.2
1	`smo`	∘
2	`smon`	×
3	`smh`	—
4	`smho`	-·-
5	`smhc`	– –
6	`direct`	△

Table 5.1: Numbers and markers assigned to the algorithms.

All algorithms are provided with the same starting point $x_0 = z^*$, the coarse model minimizer. We note that this provides an unfair advantage for the direct, classical method compared to usual application of such a method, which is to start the direct method in a quasi-random starting point.

All algorithms are run using the default options of the toolbox. Hence parameters like the initial trust region size and other parameters common to the algorithms are the same for all runs. The usual space mapping definition (5.3) is used for all space mapping algorithms. In Section 5.3.2 we report on the effects of using other space mapping definitions.

When reporting on the performance of the algorithms we compare the level of accuracy after the kth iteration,

$$\text{Level of accuracy} \quad = \quad \frac{H(f(x_k)) - H(f(x^*))}{H(f(x_0))}. \tag{5.28}$$

We also lists the condition that causes a given algorithm to stop iterating on a given problem. The following abbreviations are used:

SL : The algorithm stopped because the length of the last tentative step was too small, i.e. the condition (5.17) evaluated true. In these test runs we use $\varepsilon \simeq 2.22 \cdot 10^{-14}$, more precisely: a factor 100 above the calculating accuracy level. If an algorithm stops because of this condition, it is either because the tentative step from the trust region subproblem is shorter than the threshold, or because the trust region has been reduced to a size less than the threshold.

FE : The algorithm stopped because the number of fine model evaluations reached the threshold k_{max}, i.e. the condition (5.18) evaluated true. In these test runs we use $k_{max} = 200$.

ND : The algorithm stopped because the last tentative step provided no decrease in the trust region model of that iteration, i.e. the condition (5.13) evaluated false. For the hybrid algorithms, this condition can only stop the algorithms in the last stage where the transition parameter is zero (because the predicted decrease is measured relative to the fine model objective at the previous iterate, and the surrogate model is not in general interpolating the fine model, only when $w = 0$). If an algorithm stops because of this condition it is often due to rounding errors dominating the calculations.

The subproblems of the algorithms are solved using the following optimization methods:

nonlinear minimax : An implementation of the method of Hald and Madsen [11], is used for solving the subproblems (5.20) and (5.25), and the problem (5.2).

linear minimax : The linear programming method 'linprog' of the Matlab Optimization Toolbox [13], is used for solving the subproblem (5.25) when $w = 0$. The option 'Largescale' of the method is set to 'off'.

quadratic programming : The quadratic programming method 'quadprog' of the Matlab Optimization Toolbox [13], is used for solving the subproblem (5.12). The option 'Largescale' of the method is set to 'off'.

nonlinear least-squares : An implementation by Hans Bruun Nielsen, IMM, Technical University of Denmark, of the Levenberg-Marquardt method for nonlinear least-squares [15], is used for solving the parameter extraction problem (5.3), or the regularized formulation of the problem (5.7).

Results

The Tables 5.2, 5.3, 5.4 and 5.5 list the results from the test runs. The tables show the number of fine model evaluations a given algorithm needs to reach a given level accuracy. For the space mapping algorithms it is indicated to which level of accuracy the space mapping method is active, i.e. for a hybrid method it is indicated if the transition parameter is greater than zero after the iteration where a given accuracy has been reached.

Original Space Mapping Algorithms Consider first the space mapping methods concerned with the original space mapping formulation (algorithm *1* and *2*). We see from the tables that neither of the algorithms converge to the minimizer of the fine model. This is because the mapping is not perfect for any of the test problems, i.e. $p(x^*) \neq z^*$. As discussed in Chapter 4, the condition of a perfect mapping is crucial for the convergence properties of original space mapping algorithms, here algorithm *1* and *2*.

The results from the test problems no. 2, 5, 6, 7 and 8 show that algorithm *2* should be preferred over algorithm *1*, since it ultimately achieves a better accuracy for these test problems. However, for test problem 1, Table 5.2 indicates that algorithm *1* reaches the final level of accuracy more rapidly. But, we should recall the results from the examination of this test problem in Chapter 4, Section 4.1.2 (see also Section 5.3.3 below): namely that algorithm *2* finds a solution that is feasible with regard to the design specifications of the problem, which algorithm *1* does not. So even though algorithm *1* seems to converge faster to the same level of accuracy, algorithm *2* is preferable also for this test problem.

Comparing only the initial convergence of algorithm *1* and *2* with that of the other space mapping algorithms. We see that when algorithms *1* and *2* perform their best they are only as good as the performance of the other space mapping algorithms. So from these results there are no reasons to prefer algorithm *1* and *2* over any of the other space mapping algorithms.

Comparing algorithm *1* and *2* to the direct, classic algorithm, we see that algorithm *2* performs better than the direct algorithm in the initial stage, but for most test problems the direct algorithm obtains an ultimately higher final accuracy. Algorithm *1* only performs better than the direct algorithm in the first few steps. When the direct algorithm have attained a finite difference approximation of the Jacobian of the fine model, the direct algorithm faster obtains a higher level of accuracy than algorithm *1*.

Test problem no. 1

Level of	Algorithm no.					
accuracy	*1*	*2*	*3*	*4*	*5*	*6*
10^0	$1_\diamond$	$1_\diamond$	$1_\diamond$	$1_\diamond$	$1_\diamond$	3
10^{-1}	$2_\diamond$	$2_\diamond$	$2_\diamond$	$2_\diamond$	$2_\diamond$	4
10^{-2}	$4_\diamond$	$9_\diamond$	$4_\diamond$	5	$9_\diamond$	10
10^{-3}	-	-	13	19	22	15
10^{-4}	-	-	17	22	27	19
10^{-5}	-	-	20	27	32	22
10^{-6}	-	-	25	33	34	22
10^{-8}	-	-	34	43	41	31
10^{-10}	-	-	35	53	52	35
10^{-12}	-	-	45	59	56	43
10^{-14}	-	-	54	67	65	50
Stop:	SL	SL	ND	ND	ND	SL

Problem no. 2

Level of	Algorithm no.					
accuracy	*1*	*2*	*3*	*4*	*5*	*6*
10^0	$1_\diamond$	$1_\diamond$	$1_\diamond$	$1_\diamond$	$1_\diamond$	8
10^{-1}	-	$8_\diamond$	$8_\diamond$	$7_\diamond$	$7_\diamond$	19
10^{-2}	-	-	$14_\diamond$	85	21	-
10^{-3}	-	-	23	116	25	-
10^{-4}	-	-	53	123	33	-
10^{-5}	-	-	65	151	36	-
10^{-6}	-	-	70	167	42	-
10^{-8}	-	-	82	186	48	-
10^{-10}	-	-	104	-	52	-
10^{-12}	-	-	122	-	53	-
10^{-14}	-	-	-	-	54	-
Stop:	ND	SL	SL	FE	SL	FE

Table 5.2: Convergence results for test problems no. 1 (two variables) and no. 2 (seven variables). Each column of the table bodies lists the number of fine model evaluations used by the algorithm of that column to obtain the level of accuracy listed in the leftmost column, see (5.28)*. The marker* ($\diamond$) *indicates that the space mapping method was active at that level of accuracy. The bottom row lists the reasons for stopping the algorithms.*

Test problem no. 3

Level of	Algorithm no.					
accuracy	*1*	*2*	*3*	*4*	*5*	*6*
10^{0}	$1_{\diamond}$	$1_{\diamond}$	$1_{\diamond}$	$1_{\diamond}$	$1_{\diamond}$	3
10^{-1}	-	-	$5_{\diamond}$	5	$4_{\diamond}$	6
10^{-2}	-	-	$7_{\diamond}$	7	$7_{\diamond}$	9
10^{-3}	-	-	9	8	$8_{\diamond}$	10
10^{-4}	-	-	11	9	10	11
10^{-5}	-	-	11	10	11	11
10^{-6}	-	-	11	10	11	12
10^{-8}	-	-	12	-	-	13
10^{-10}	-	-	13	-	-	14
10^{-12}	-	-	14	-	-	14
10^{-14}	-	-	14	-	-	-
Stop:	SL	SL	SL	SL	SL	SL

Table 5.3: Convergence results for test problem no. 3, with two variables. Each column of the table body lists the number of fine model evaluations used by the algorithm of that column to obtain the level of accuracy listed in the leftmost column, see (5.28)*. The marker (*$\diamond$*) indicates that the space mapping method was active at that level of accuracy. The bottom row lists the reasons for stopping the algorithms.*

The only situations where algorithms *1* and *2* may be preferable are: If the mapping is perfect, $p(x^*) = z^*$, such that both algorithms converge to the fine model minimizer (refer to Chapter 4), or if the user is satisfied with the rather limited accuracy. The latter is in fact often the case in engineering design, and for such an application we recommend algorithm *2* over algorithm *1*, because of the results discussed above and because of the more theoretical considerations of Chapter 4.

Hybrid Space Mapping Algorithms We now compare the hybrid space mapping algorithms with the direct, classical algorithm. From the tables we see that the algorithms *3*, *4* and *5* converge to the correct solutions for most of the test problems. However, for some of the test problems the algorithms stops prematurely. First we analyze the cases where convergence were not achieved, then we consider the performance of the algorithms.

Algorithms *3* and *5* on test problem 7: The algorithms stopped because the step lengths were too short. By closer examination we have found that the

Test problem no. 5

Level of	Algorithm no.					
accuracy	*1*	*2*	*3*	*4*	*5*	*6*
10^{0}	$1_{\diamond}$	$1_{\diamond}$	$1_{\diamond}$	$1_{\diamond}$	$1_{\diamond}$	4
10^{-1}	-	$3_{\diamond}$	$3_{\diamond}$	$3_{\diamond}$	$3_{\diamond}$	18
10^{-2}	-	-	77	58	70	-
10^{-3}	-	-	82	62	73	-
10^{-4}	-	-	99	65	80	-
10^{-5}	-	-	112	71	117	-
10^{-6}	-	-	132	76	121	-
10^{-8}	-	-	153	84	132	-
10^{-10}	-	-	163	92	144	-
10^{-12}	-	-	172	100	157	-
10^{-14}	-	-	180	104	166	-
Stop:	SL	FE	SL	SL	SL	SL

Test problem no. 6

Level of	Algorithm no.					
accuracy	*1*	*2*	*3*	*4*	*5*	*6*
10^{0}	$1_{\diamond}$	$1_{\diamond}$	$1_{\diamond}$	$1_{\diamond}$	$1_{\diamond}$	4
10^{-1}	$65_{\diamond}$	$2_{\diamond}$	$2_{\diamond}$	$2_{\diamond}$	$2_{\diamond}$	7
10^{-2}	-	$13_{\diamond}$	24	23	24	36
10^{-3}	-	-	59	67	58	59
10^{-4}	-	-	83	-	84	86
10^{-5}	-	-	93	-	112	91
10^{-6}	-	-	123	-	124	101
10^{-8}	-	-	139	-	140	125
10^{-10}	-	-	148	-	146	149
10^{-12}	-	-	152	-	152	163
10^{-14}	-	-	154	-	152	163
Stop:	SL	SL	SL	FE	SL	FE

Table 5.4: Convergence results for the test problems no. 5 and 6, both with three variables. Each column of the table body lists the number of fine model evaluations used by the algorithm of that column to obtain the level of accuracy listed in the leftmost column, see (5.28)*. The marker* ($\diamond$) *indicates that the space mapping method was active at that level of accuracy. The bottom row lists the reasons for stopping the algorithms.*

Test problem no. 7

Level of	Algorithm no.					
accuracy	*1*	*2*	*3*	*4*	*5*	*6*
10^{0}	$1_{\diamond}$	$1_{\diamond}$	$1_{\diamond}$	$1_{\diamond}$	$1_{\diamond}$	4
10^{-1}	-	$13_{\diamond}$	11	6	11	7
10^{-2}	-	-	43	47	43	42
10^{-3}	-	-	78	66	78	49
10^{-4}	-	-	81	70	81	143
10^{-5}	-	-	87	90	87	163
10^{-6}	-	-	93	94	93	166
10^{-8}	-	-	-	100	-	-
10^{-10}	-	-	-	109	-	-
10^{-12}	-	-	-	121	-	-
10^{-14}	-	-	-	126	-	-
Stop:	SL	FE	SL	SL	SL	FE

Test problem no. 8

Level of	Algorithm no.					
accuracy	*1*	*2*	*3*	*4*	*5*	*6*
10^{0}	$1_{\diamond}$	$1_{\diamond}$	$1_{\diamond}$	$1_{\diamond}$	$1_{\diamond}$	4
10^{-1}	$4_{\diamond}$	$2_{\diamond}$	$2_{\diamond}$	$2_{\diamond}$	$2_{\diamond}$	5
10^{-2}	-	$7_{\diamond}$	$5_{\diamond}$	45	$4_{\diamond}$	8
10^{-3}	-	-	11	-	14	18
10^{-4}	-	-	13	-	20	24
10^{-5}	-	-	18	-	26	24
10^{-6}	-	-	22	-	45	42
10^{-8}	-	-	31	-	72	63
10^{-10}	-	-	38	-	77	72
10^{-12}	-	-	44	-	81	80
10^{-14}	-	-	46	-	84	82
Stop:	SL	SL	SL	SL	SL	SL

Table 5.5: Convergence results for the test problem no. 7 and 8, both with three variables. Each column of the table body lists the number of fine model evaluations used by the algorithm of that column to obtain the level of accuracy listed in the leftmost column, see (5.28)*. The marker* ($\diamond$) *indicates that the space mapping method was active at that level of accuracy. The bottom row lists the reasons for stopping the algorithms.*

algorithms converged to a local minimizer of the fine model.

Algorithm *4* and *5* on test problem 3: The algorithms stopped because the step lengths were too short. By closer examination we have found that the approximate gradient information was so inaccurate that the algorithms converged to a point that judging from the inexact gradient information is a stationary point of the fine model.

Algorithm *4* on the test problems 2, 6 and 8, the algorithm stopped prematurely. For the test problems 2 and 6 the algorithm stops because too many fine model evaluations were used, this indicates that the gradient approximation in the linear Taylor model is not good enough, but the algorithm may ultimately converge provided that enough fine model evaluations are allowed. By closer examination we have found that the algorithm converges to full accuracy after 250 fine model evaluations on problem 2, and after 283 fine model evaluations on problem 6. For problem 8, the algorithm stops because the step length was to small. By closer examination we have found that the algorithm converged to a local minimizer of the fine model.

The hybrid methods perform nearly identical in the initial stage where the space mapping method is active, i.e. $w > 0$. Due to the above mentioned problems with final convergence, we are not able to perform a fair comparison of the hybrid methods in the last stage of the iteration process where the direct, classical method is used, i.e. $w = 0$. Instead we compare the hybrid methods with the direct, classical method.

In general we see that the hybrid space mapping algorithms converge faster than the direct algorithm both in the initial phase, where the space mapping method is used, and in the final phase of the iteration process. An exception to this is test problem 1, where the direct method converges faster in the last phase of the iteration process. A clear advantage of all the hybrid space mapping algorithms is that they do not need to perform an initial finite difference approximation to obtain the fine model Jacobian approximation. The space mapping algorithms obtain this approximation using the mapped coarse model, see page 129. Hence the hybrid space mapping algorithms are in that sense pre-conditioned by the mapped coarse model, before switching to direct, classical optimization.

So in total, the results verify that a combination of the space mapping method and a direct, classical method is preferable over using either of the methods separately. With the hybrid algorithms we achieve the features of both methods: Fast initial convergence from the space mapping, and convergence to a

stationary point of the fine model from the direct, classical method.

Direct, Classical Algorithm Algorithm *6* also has some difficulties in converging for the test problems 2, 5 and 7: For problem 2 the algorithm stops because of too many fine model evaluations. If allowed to continue, the algorithm converges after 434 fine model evaluations. For test problem 5 the algorithm stops because the step length was too short. By closer examination we have found that the approximate gradient information was so inaccurate that the algorithm converged to a point that judging from the inexact gradient information is a stationary point of the fine model. For test problem 7 the algorithm stops because of too many fine model evaluations. If allowed to continue, the algorithm converges to a point that judging from the inexact gradient information is a stationary point of the fine model.

Summary of the Results

The original space mapping algorithms are in general not preferable over any of the other algorithms we have tested, except in the special (but very important) case where only very few fine model evaluations are available and where limited accuracy can be accepted. It is most likely a better choice to initialize a direct, classical method with the Jacobian approximation obtained from the mapped coarse model, than it is to use one of the original space mapping algorithms.

The hybrid space mapping algorithms showed good initial convergence, similar to that observed for the original space mapping algorithms. But also the final convergence is good, and in fact better or comparable to that of the direct, classical method started in the coarse model minimizer.

5.3.2 Space Mapping Definitions

We now examine the effects of using alternative space mapping definitions. Figure 5.2 shows results from test runs using four different space mapping definitions on test problem 2. The four definitions of the space mapping are (5.3), (5.4), (5.5) and (5.6). We use the regularization parameter $\lambda = 10^{-3}$, as the best results were obtained with this choice on this test problem.

Both the definitions (5.4) and (5.5), regularizing with regard to the distance to z^* respectively x^*, have a positive effect on the initial convergence of two of the hybrid space mapping algorithms, algorithms *3* and *5*. Further, the

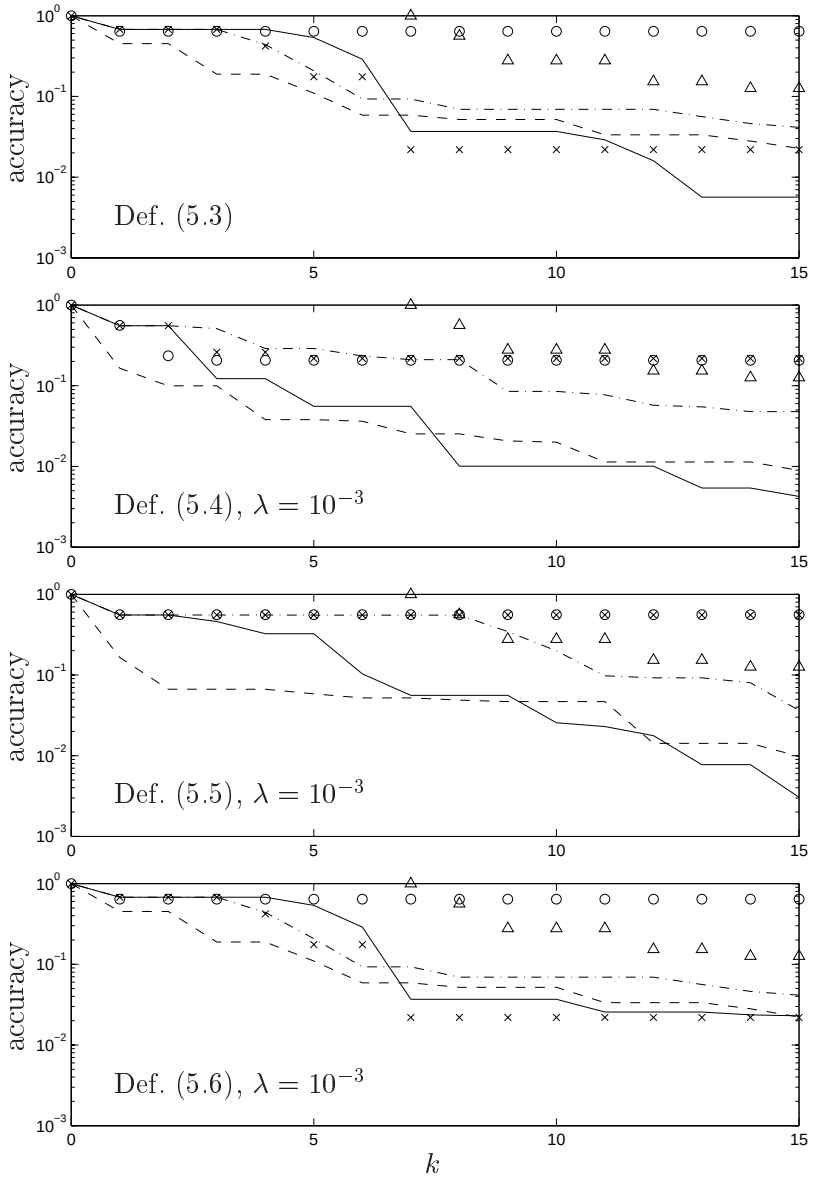


Figure 5.2: Results from the first 15 steps on problem no. 2, with seven variables, using four different definitions of the space mapping. See Table 5.1 for a description of the markers. Note that the number of fine model evaluations after iteration k is $k + 1$.

definition (5.4) has a positive effect on the original space mapping formulation, algorithm *1*. However, the new formulation of the original space mapping, algorithm *2*, does not reach the same level of accuracy as with the usual space mapping definition.

The definition (5.6), regularizing with regard to gradient information, does not have a significant effect on the iteration process compared to the usual space mapping definition (5.3).

So judging from this experiment, the definition (5.4), regularizing with regard to the distance to z^*, provides the best initial convergence for original space mapping algorithms and two of the hybrid algorithms. The final level of accuracy was better for the one of the original space mapping algorithms, but it was worse for the other. Hence it is not clear exactly when to use an alternative space mapping formulation with regularization. Further research is needed in order to draw any firm conclusions.

5.3.3 Optimization Trajectories

We consider the convergence of the two original space mapping methods `smo` and `smon`, and the hybrid space mapping method `smh` with linear transition, for the TLT2 test problem (no. 1 in the toolbox).

In the Figures 5.3, 5.4 and 5.5 are shown iteration trajectories in the coarse model space from optimization using the three methods. The trajectories are plotted on top of the space mapping image $\{p(x) : x \in \mathbb{R}^n\}$, which was introduced in Chapter 4, Section 4.1.2. The solutions referred to in the figure captions are:

x^* : a solution to (5.1),

z^* : a solution to (5.2),

x_p^* : a solution to (5.9),

x_{cop}^* : a solution to (5.19).

The TLT2 problem is interesting not only because the mapping is not perfect, $p(x^*) \neq z^*$, but also because the coarse model minimizer z^* is not in the image of the space mapping, $z^* \notin p(\mathbb{R}^n)$. The latter causes that the solutions x_p^* and x_{cop}^* are not equal, refer to Chapter 4, Section 4.1.3.

From Figure 5.3 we see how the algorithm of the original space mapping method converges to the point in the space mapping image $p(x_p^*)$ with the least distance to the coarse model minimizer z^*. Hence

$$p(x_p^*) = \arg\min_{z \in p(\mathbb{R}^n)} \|z - z^*\|_2 \,. \tag{5.29}$$

In the iterations the algorithm comes quite close to the minimizer of the fine model, however it diverges again toward the solution of the least-squares problem (5.29).

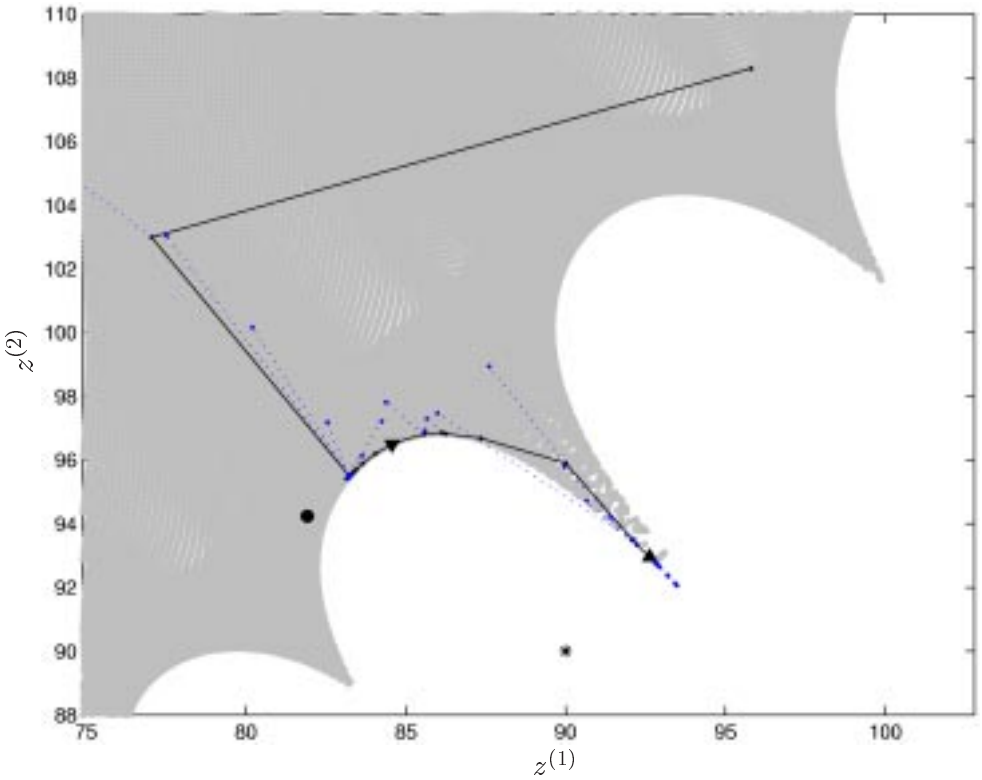


Figure 5.3: Optimization process trajectory in coarse model space using the original space mapping formulation, solving $p(x) = z^*$ with a least-squares merit function, implemented in `smo`*. The full line indicates the accepted steps, the broken lines indicate rejected tentative steps. The markers indicate the coarse model minimizer z^* ($*$), the space mapping image of: the space mapping solution $p(x_p^*)$ ($\blacktriangle$), the minimizer of the mapped coarse model $p(x_{c\circ p}^*)$ ($\blacktriangledown$) and the fine model minimizer $p(x^*)$ ($\bullet$).*

From Figure 5.4 we see how the new formulation of the original space mapping method converges to the point in the mapping image $p(x^*_{c\circ p})$ where the coarse model response has its minimum. Hence

$$p(x^*_{c\circ p}) = \arg\min_{z\in p(\mathbb{R}^n)} H(c(z))\,.$$

Similar to the other original space mapping method, this method comes quite close to the fine model minimizer before diverting again toward the minimizer of the coarse model in the mapping image.

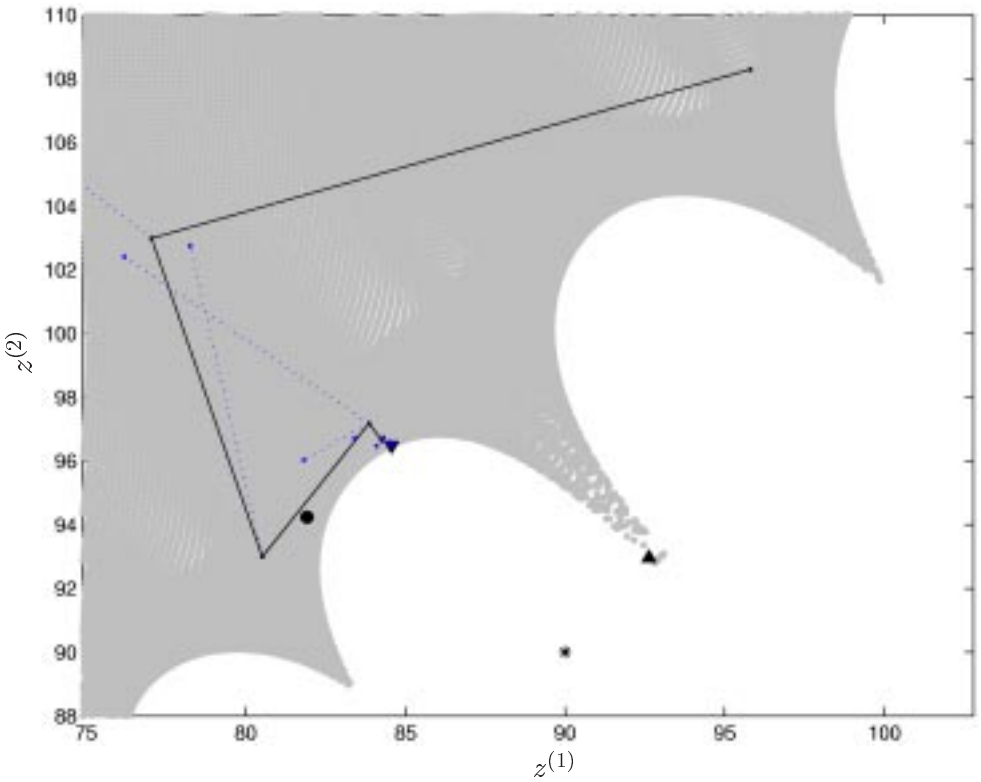


Figure 5.4: Optimization process trajectory in coarse model space using the new space mapping formulation, minimizing $H(c(p(x)))$, implemented in `smon`*. The full line indicates the accepted steps, the broken lines indicate rejected tentative steps. The markers indicate the coarse model minimizer z^* (∗), the space mapping image of: the space mapping solution $p(x^*_p)$ (▲), the minimizer of the mapped coarse model $p(x^*_{c\circ p})$ (▼) and the fine model minimizer $p(x^*)$ (•).*

From Figure 5.5 we see how the hybrid space mapping method converges to the image of the fine model minimizer $p(x^*)$. Comparing this figure with the two previous figures, it is clear to see how the space mapping method serves as a pre-conditioner. This is seen by the switch from the space mapping method to the direct, classical method that occurs around the point where the space mapping method otherwise would start to diverge from the fine model minimizer. The switch is taking place because the space mapping method fails to produce downhill steps for the fine model.

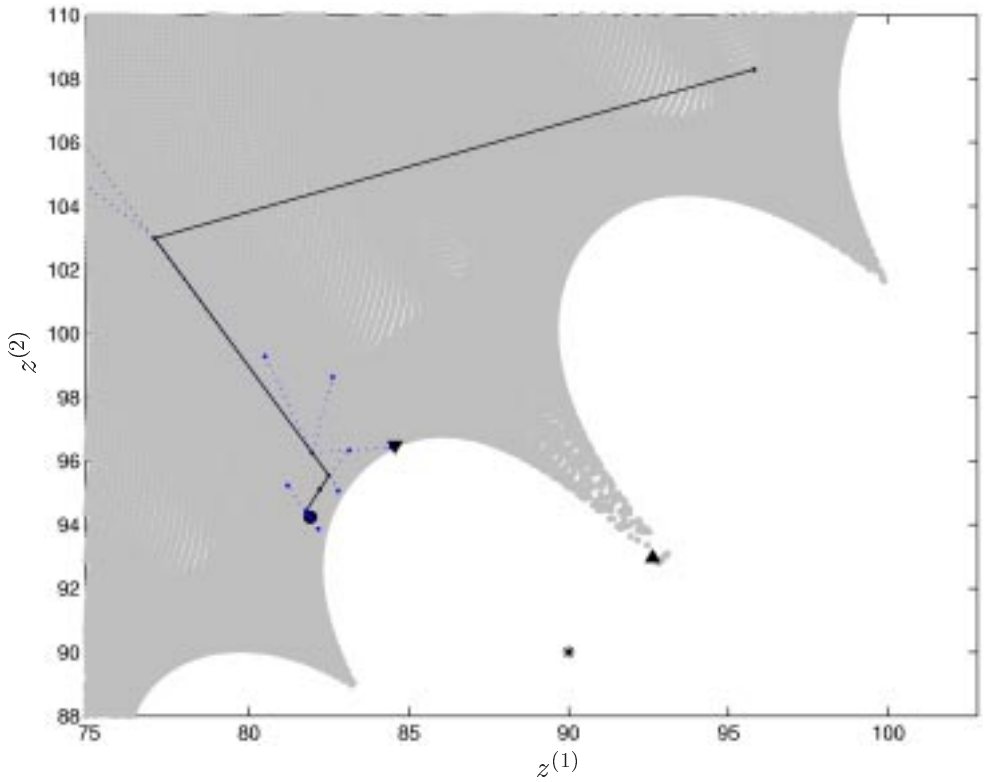


Figure 5.5: Optimization process trajectory in coarse model space using the hybrid space mapping algorithm with linear transition, implemented in `smh`*. The full line indicates the accepted steps, the broken lines indicate rejected tentative steps. The markers indicate the coarse model minimizer z^* ($*$), the space mapping image of: the space mapping solution $p(x_p^*)$ (▲), the minimizer of the mapped coarse model $p(x_{cop}^*)$ (▼) and the fine model minimizer $p(x^*)$ (•).*

This example illustrates the importance of combining the original space mapping method with classical optimization methods in order to ensure that the solution found is indeed a solution of the original problem. However, in cases where the accuracy is less important, the original space mapping methods may provide a reasonable approximation to the solution, if they are stopped before diverging toward x_p^* or x_{cop}^*.

5.4 Conclusion

This chapter has presented five optimization algorithms based on the space mapping method. Two of the algorithms are concerned with the original space mapping method. The other three algorithms are hybrid algorithms combining the space mapping method with a classical Taylor based optimization method.

The key advantage of the hybrid algorithms over the original space mapping methods is that convergence to a stationary point of the fine model can be proved. On the other hand, the original space mapping methods are conceptually simpler and also easier to implement.

With the hybrid space mapping framework we have developed a framework for pre-conditioning optimization problems using a broad class of surrogate models, not limited to those generated by the space mapping method.

The performance of the Matlab implementation of the algorithms has been reported for 7 test problems. The effects of using alternative space mapping definitions have been demonstrated on one test problem.

The algorithms based on the original space mapping formulation showed good initial performance but for all the test problems they only reached a moderate final level of accuracy. We only recommend using the original space mapping algorithms in the special case where only few fine model evaluations are available and where limited accuracy can be accepted. Otherwise we suggest to use either, one of the hybrid space mapping algorithms, or a direct, classical algorithm initialized with derivative information obtained from the space mapped coarse model in the starting point.

The hybrid space mapping algorithms showed a fast initial convergence, similar to that of the original space mapping algorithms, and they also showed comparable convergence results in the last stage of the iteration process as to those of a direct, classic method. This was expected in advance, and was confirmed by the observations from the runs on the 7 test problems. We prove the convergence of the hybrid methods in the next chapter, Chapter 6.

Symbols

$\|\cdot\|$	unspecified norm
$\|\cdot\|_2$	Euclidean norm, $\|x\|_2 = (x^T x)^{\frac{1}{2}}$
$\|\cdot\|_F$	Frobenious norm
c	response from the coarse model, $c : \mathbb{R}^n \mapsto \mathbb{R}^m$
f	response from the fine model, $f : \mathbb{R}^n \mapsto \mathbb{R}^m$
g	response correction factors
H	convex function, used as merit function
m	number of response functions
n	dimensionality of the design parameter space
p	space mapping
p_λ	regularized space mapping
x	optimizeable model parameters of f and c
x^*	minimizer of $H(f(x))$
x^*_{cop}	minimizer of $H(c(p(x)))$
x^*_p	minimizer of $\|p(x) - z^*\|_2$
z^*	minimizer of $H(c(z))$
λ	regularization parameter in space mapping definitions

References

[1] M.H. Bakr, J.W. Bandler, N.K. Georgieva, K. Madsen, *A Hybrid Aggressive Space Mapping Algorithm for EM Optimization*, IEEE Trans. Microwave Theory Tech., vol. 47, pp. 2440–2449, 1999.

[2] M.H. Bakr, J.W. Bandler, K. Madsen, J. Søndergaard, *An Introduction to the Space Mapping Technique*, Optimization and Engineering, vol. 2, no. 4, pp. 369–384, 2001.

[3] M.H. Bakr, J.W. Bandler, K. Madsen, J. Søndergaard, *Review of the Space Mapping Approach to Engineering Optimization and Modeling*, Optimization and Engineering, vol. 1, no. 3, pp. 241–276, 2000.

[4] J.W. Bandler, R.M. Biernacki, S.H. Chen, P.A. Grobelny, R.H. Hemmers, *Space Mapping Technique for Electromagnetic Optimization*, IEEE Trans. Microwave Theory Tech., vol. 42, pp. 2536–2544, 1994.

[5] J.W. Bandler, R.M. Biernacki, S.H. Chen, R.H. Hemmers, K. Madsen, *Electromagnetic Optimization Exploiting Aggressive Space Mapping*, IEEE Trans. Microwave Theory Tech., vol. 43, pp. 2874–2882, 1995.

[6] J.W. Bandler, Q. Cheng, S. Dakroury, A.S. Mohamed, M.H. Bakr, K. Madsen, J. Søndergaard, *Space Mapping: The State of the Art*, submitted, IEEE Trans. Microwave Theory Tech., 2004.

[7] R.H. Byrd, J. Nocedal, R.B. Schnabel, *Representations of Quasi-Newton Matrices and Their Use in Limited Memory Methods*, Mathematical Programming, vol. 63, pp. 129–156, 1994.

[8] A.R. Conn, N.I.M. Gould, P.L. Toint, *Trust-Region Methods*, SIAM, Philadelphia, 2000.

[9] J.E. Dennis, Jr., Dept. Computational and Applied Mathematics, Rice University, Houston, Texas, USA, private discussions, 2001.

[10] J.E. Dennis, Jr., R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[11] J. Hald, K. Madsen, *Combined LP and Quasi-Newton Methods for Minimax Optimization*, Mathematical Programming, vol. 20, pp. 49–62, 1981.

[12] K. Madsen, *Minimization of Non-Linear Approximation Functions*, Dr. Techn. Thesis, Institute for Numerical Analysis, Technical University of Denmark, 1986.

[13] Matlab^{TM} version 6.5 and Matlab^{TM} Optimization Toolbox version 2.2., The MathWorks, Inc., 3 Apple Hill Drive, Natick MA 01760-2098, 2003.

[14] J.J. Moré, *Recent Developments in Algorithms and Software for Trust Region Methods*, Mathematical Programming: The State of The Art, pp. 258–287, 1982.

[15] H.B. Nielsen, Matlab implementation of the Levenberg-Marquardt method for nonlinear least-squares, IMM, DTU, 2001.
Available at `http://www.imm.dtu.dk/~hbn/Software/`

[16] J. Nocedal, S.J. Wright, *Numerical Optimization*, Springer Series in Operations Research, Springer, 1999.

[17] F.Ø. Pedersen, *Advances on the Space Mapping Optimization Method*, Master Thesis, IMM-THESIS-2001-35, Informatics and Mathematical Modelling, Technical University of Denmark, 2001.

[18] F.Ø. Pedersen, Technical University of Denmark, Lyngby, Denmark, private discussions, 2001.

[19] F.Ø. Pedersen, Technical University of Denmark, Lyngby, Denmark, private discussions, 2003.

[20] J. Søndergaard, *Non-Linear Optimization Using Space Mapping*, Master Thesis, IMM-EKS-1999-23, Informatics and Mathematical Modelling, Technical University of Denmark, 1999.
Available at `http://www.imm.dtu.dk/~km/jsmaster.ps.gz`

[21] L.N. Vicente, *Space Mapping: Models, Sensitivities and Trust-region Methods*, to appear in Optimization and Engineering, 2003.

Chapter 6

Convergence of Hybrid Space Mapping Algorithms

Included paper:

Convergence of Hybrid Space Mapping Algorithms,
Kaj Madsen and Jacob Søndergaard,
submitted for publication, Optimization and Engineering.

Convergence of Hybrid Space Mapping Algorithms

Kaj Madsen (km@imm.dtu.dk) and Jacob Søndergaard (js@imm.dtu.dk)
Informatics and Mathematical Modelling, Technical University of Denmark

Abstract. The space mapping technique is intended for optimization of engineering models which involve very expensive function evaluations. It may be considered a preprocessing method which often provides a very efficient initial phase of an optimization procedure. However, the ultimate rate of convergence may be poor, or the method may even fail to converge to a stationary point.
We consider a convex combination of the space mapping technique with a classical optimization technique. The function to be optimized has the form $H \circ f$ where $H : \mathbb{R}^m \mapsto \mathbb{R}$ is convex and $f : \mathbb{R}^n \mapsto \mathbb{R}^m$ is smooth. Experience indicates that the combined method maintains the initial efficiency of the space mapping technique. We prove that the global convergence property of the classical technique is also maintained: The combined method provides convergence to the set of stationary points of $H \circ f$.

Keywords: space mapping, global convergence

1. Introduction

The subject of this paper is to prove global convergence of an optimization method which is a convex combination of two strategies: One which is efficient initially in an iteration and another which has guaranteed global convergence. The first algorithm is the so-called space mapping technique, described and motivated in (Bakr et al., 2001), the other one is a classical 1. order Taylor based trust region algorithm.

The problem to be solved is the following:

$$\min_{x \in \mathbb{R}^n} H(f(x)) \tag{1}$$

where $f : \mathbb{R}^n \mapsto \mathbb{R}^m$ is a smooth function, often $m \gg n$. $H : \mathbb{R}^m \mapsto \mathbb{R}$ is a convex function. It may be a norm in $\mathbb{R}^m$, typically the L_1, L_2 or the L_∞ norm. The following minimax function,

$$H(y) \equiv \max_{1 \leqslant i \leqslant m} \{y_i\}$$

where $y = (y_1, y_2, ..., y_m)^T$, is also often used, e.g., in electromagnetic design, which has been a major application area for the space mapping technique. Thus it is important to cover the case where H is non-differentiable.

In the smooth least squares case, $H = \|.\|_2^2$, the 1. order Taylor based method is the Gauss-Newton method. In this case convergence of the combined algorithm is proved in (Vicente, 2002). For the non-differentiable choices of H mentioned above special versions of the Taylor based method have been published in (Madsen, 1975) and (Hald & Madsen, 1985). For general convex H convergence of the Taylor based trust region algorithm was proved in (Madsen, 1985). Related results may be found in (Fletcher, 1982) and (Womersley & Fletcher, 1986).

The space mapping technique which was introduced in (Bandler et al., 1994) is intended for problems where f is computationally expensive. It is an optimization technique for engineering design in the following situation: f is representing an accurate model of some physical system. Besides this model of primary interest (denoted the fine model), access to a cheaper (coarse) model of the same physical system is assumed. The latter may be less accurate. The main idea of the space mapping technique is to use the coarse model to gain information about the fine model, and to apply this in the search for an optimal solution of the latter. Thus the technique iteratively establishes a mapping between the parameters of the two models which relate similar model responses. Having this mapping, most of the model evaluations can be directed to the fast coarse model.
A review of the Space Mapping approach is given in (Bakr et al., 2000).

We give a description of the combined method in Section 2. The convergence is proved in Section 3.

2. Description of the Algorithms

The two algorithms which are combined are both iterative. In the descriptions below the current iterate is $x_k \in \mathbb{R}^n$. $H \circ f$ is denoted by F.

The Space Mapping Algorithm (SMA) assumes two functions available: The function f to be minimized, and a function $c : \mathbb{R}^n \mapsto \mathbb{R}^m$ which represents the coarse model that is related to the same physical model as f. The space mapping $p : \mathbb{R}^n \mapsto \mathbb{R}^n$ is intended to connect similar values of f and c. In the present description it satisfies the following for $x \in \mathbb{R}^n$:

$$p(x) \in \arg\min_{\tilde{x} \in \mathbb{R}^n} \|c(\tilde{x}) - f(x)\| \tag{2}$$

where $\|\cdot\|$ is a norm in $\mathbb{R}^n$, usually the L_2 norm. The tentative step h_k from x_k is based on the following approximation to $p(x_k + h)$:

$$p_k(h) = B_k h + p(x_k) \ , \tag{3}$$

where the matrix B_k is intended to approximate the gradient $p'(x_k)^T$ of p at x_k. This approximation may be found using the gradients of f and c if they are available, otherwise a Broyden update (Broyden, 1965) or a difference approximation to $p'(x_k)^T$ has been used. It is not important in the present paper how B_k is found.
The SMA finds h_k as a solution to

$$h_k \in \arg\min_h H(c(p_k(h))) \ ,$$

where the minimization is usually subject to a trust region.

In the Taylor-based method for minimizing f the tentative step from x_k is based on the following 1. order approximation to $f(x_k + h)$,

$$f_k(h) = D_k h + f(x_k) \ .$$

In the present paper $D_k = f'(x_k)^T$. Otherwise a Broyden update or a difference approximation to $f'(x_k)^T$ have been used.
The tentative step h_k is found as a solution to

$$h_k \in \arg\min_h H(f_k(h)) \ , \tag{4}$$

where the minimization is usually subject to a trust region.

In the combined algorithm (SMTA) for minimizing f the tentative step h_k from x_k is based on a convex combination of $c \circ p_k$ and f_k:

$$s_k^w(h) = w\, c(p_k(h)) + (1 - w)\, f_k(h)$$

where $0 \leqslant w \leqslant 1$ is a transition parameter.
The method finds the tentative step h_k as a solution to

$$h_k \in \arg\min_h H(s_k^{w_k}(h)) \ ,$$

where the minimization is usually subject to a trust region.

In the algorithm $w_0 = 1$ and w_k is non-increasing function of k. The principle being that if $c \circ p_k$ is doing badly in approximating f then w_k is decreased, and thereby the algorithm gradually switches to using the Taylor model f_k of f.

2.1. Details of the SMTA

The trust region at x_k is the set

$$T_k = \{\ x \in I\!R^n \mid \|x - x_k\| \leqslant \Lambda_k\ \} \tag{5}$$

where $\|\cdot\|$ is a suitable norm in $I\!R^n$ and $\Lambda_k > 0$.

At the current iterate x_k, the tentative step $h_k \in I\!R^n$ is a solution to

$$\begin{aligned} h_k \in \arg\min_h \ & H(s_k^{w_k}(h)) \\ \text{s.t.} \ & x_k + h \in T_k \end{aligned} \tag{6}$$

The quality of a given tentative step h_k is measured by

$$\Delta S_k(h_k) \equiv H(s_k^{w_k}(h_k)) - H(f(x_k)) \ .$$

$-\Delta S_k(h_k)$ may be interpreted as a measure of the ability of the model s_k^w to predict a decrease in F. Notice that $\Delta S_k(0)$ is not necessarily 0, however, i.e., the model does not necessarily interpolate at x_k.
If h_k is acceptable then we use $x_k + h_k$ as the next iterate, otherwise we maintain x_k as the best iterate found. The acceptance of the step is based on the following criteria:
If the predicted decrease is non-positive,

$$-\Delta S_k(h_k) \leqslant 0 \ , \tag{7}$$

then the step is rejected. Otherwise the step is accepted if F decreases sufficiently:

$$F(x_k) - F(x_k + h_k) \geqslant \tilde{\delta}_1[-\Delta S_k(h_k)] \tag{8}$$

where $0 < \tilde{\delta}_1 < 1$.

In each iteration the local bound Λ_k and the transition parameter w_k are adjusted as follows:
If (7) is true then $\Lambda_{k+1} = \Lambda_k$, otherwise Λ_{k+1} depends on the ratio between actual and the predicted decrease. If

$$F(x_k) - F(x_k + h_k) \leqslant \tilde{\delta}_2[-\Delta S_k(h_k)] \ , \tag{9}$$

$\tilde{\delta}_1 < \tilde{\delta}_2 < 1$, is satisfied then $\Lambda_{k+1} = K_1\Lambda_k$ with $0 < K_1 < 1$.
If

$$F(x_k) - F(x_k + h_k) \geqslant \tilde{\delta}_3[-\Delta S_k(h_k)] \ , \tag{10}$$

$\tilde{\delta}_2 < \tilde{\delta}_3 < 1$, is satisfied then $\Lambda_{k+1} = K_2\Lambda_k$ with $K_2 > 1$.
If none of the conditions (9) or (10) are satisfied then we let $\Lambda_{k+1} = \Lambda_k$.
The transition parameter w_k is chosen as follows: Initially $w_k = 1$. If $x_k + h_k$ is not accepted then we wish to move weight towards the Taylor model, and therefore we let

$$w_{k+1} = K_3 \, w_k \min\{\Lambda_{k+1}, 1\} \ , \tag{11}$$

where $0 \leqslant K_3 < 1$. Otherwise $w_{k+1} = w_k$.
In order to ensure convergence, however, we need $w_k \to 0$ for $k \to \infty$. Therefore we also apply (11) if it has not been used for the previous n iterations.

2.2. Summary of the SMTA

Given Λ_0, $B_0 = I$, $w_0 = 1$, $k = 0$
0. Find x_0 as a solution to $\min_{\tilde{x}} H(c(\tilde{x}))$
1. Evaluate $f(x_0)$
2. Find $p(x_0)$ by solving (2)
3. Find h_k by solving (6)
4. Evaluate $f(x_k + h_k)$
5. Find $p(x_k + h_k)$ by solving (2)
6. If (7) is false and (8) is true then let $x_{k+1} = x_k + h_k$ otherwise $x_{k+1} = x_k$
7. Update Λ, w, B and D (only if $w < 1$)
8. Let $k = k + 1$
9. If not converged then goto 3

The steps 0, 1 and 2 are an initial phase where a (local) optimizer of $H(c(x))$ is found and the initial translation $p(x_0)$ in the linear approximation to the space mapping (3) is found.

Note that if (7) is true after step 3, then the algorithm can skip to step 8, letting x_{k+1}, Λ_{k+1}, B_{k+1} and D_{k+1} take the values from iteration k and updating w_k using (11). Hence we avoid the evaluation of $f(x_k + h_k)$ and $f'(x_k + h_k)$ in this case.

3. Proof of Convergence

We show that the SMTA satisfies the usual convergence condition for trust region methods. In the proof we do not need the actual updating scheme of the weights $\{w_k\}$, (11), we only need property (12) below. Similarly, we do not need any properties of the SMA, we only need that c is bounded (Assumption **A2** below). Thus, the proof covers a class of algorithms, including those presented in (Bakr et al., 2001) and (Pedersen, 2001). Probably also other pre-processing techniques will suit into this framework.

Throughout this section we use the notations $\{x_k\}$, $\{h_k\}$, $\{\Lambda_k\}$ and $\{w_k\}$ as they are defined in Subsection 2.1. It follows from the definition of the weights that they satisfy

$$w_k = \min\{\Lambda_{k+1}, 1\}\, o(1) \tag{12}$$

where $o(1) \to 0$ for $k \to \infty$.

We make the following assumptions

A1: For each point $z \in I\!R^n$ there exist a neighbourhood $\mathcal{N}(z) \subseteq I\!R^n$ such that

$$f(x+h) = f(x) + f'(x)h + o(\|h\|)\,, \quad x \in \mathcal{N}(z)\,,\ h \in I\!R^n\,,$$

where o is uniform for $x \in \mathcal{N}(z)$.
f' is continuous on $I\!R^n$.

A2: c is bounded in the region of interest.

A3: H is a convex function on $I\!R^m$.

A4: $\{x_k\}$ stays in a bounded region in $I\!R^n$.

3.1. Prerequisites

The convexity of H implies that it satisfies a local Lipschitz condition. H is not necessarily differentiable. However, we can define a generalized gradient which is a set of points (rather than always a single point). Since a function which is Lipschitz on a finite dimensional space is differentiable almost everywhere, the generalized gradient can be defined as follows, see (Clarke, 1975) or (Clarke, 1983), Theorem 2.5.1:

DEFINITION 1. *The* generalized gradient *of H at x, denoted $\partial H(x)$, is the convex hull of the set of all limits of the form* $\lim H'(x+e_i)$, *where H is differentiable at $x+e_i$ and $e_i \to 0$ as $i \to \infty$,*

$$\partial H(x) \equiv \mathrm{conv}\{\ \lim_{e_i \to 0} H'(x+e_i)\ \}\ .$$

It is easily seen, (Clarke, 1983), Proposition 2.1.1 and (Madsen, 1985), Proposition 2.1, that $\partial H(x)$ is non-empty and compact.
Since H is convex the generalized gradient coincides with what is called the *subdifferential* in convex analysis. The convexity of H also implies the existence of a directional derivative $H'_e(y)$ for any $y, e \in I\!R^m, e \neq 0$,

$$H'_e(y) \equiv \lim_{t \downarrow 0} \frac{H(y+te) - H(x)}{t}\ .$$

Now consider the composite function $F = H \circ f$. Since f is smooth ∂F is well defined. A *stationary point* of F is defined as follows,

DEFINITION 2. *x is a* stationary point *of F if*

$$0 \in \partial F(x).$$

Using the convexity of H and the smoothness of f we can obtain the following characterization of a stationary point (Madsen, 1985), Proposition 2.15.

PROPOSITION 1. *Let $x \in \mathbb{R}^n$. Then*

$$\begin{array}{l} 0 \in \partial F(x) \\ \Updownarrow \\ F'_h(x) \geqslant 0 \quad \text{for every direction } h \in \mathbb{R}^n, h \neq 0 \ . \end{array}$$

Below we shall use the following 1. order approximation ΔF to a change in F:

DEFINITION 3. *Let $x, h \in \mathbb{R}^n$. Define*

$$\Delta F(x;h) \equiv H(f(x) + f'(x)h) - H(f(x)) \ .$$

We shall use the following two properties of ΔF:

PROPOSITION 2. *For $x, h \in \mathbb{R}^n$ we have*

$$\Delta F(x;th) = tF'_h(x) + o(t) \ , \qquad \text{for } t > 0 \ .$$

Proof. (Madsen, 1985), Proposition 2.9.

PROPOSITION 3. *For $x, h \in \mathbb{R}^n$ and $0 \leqslant t \leqslant 1$ we have*

$$\Delta F(x;th) \leqslant t\Delta F(x;h) \ .$$

Proof. The result is a simple consequence of assumptions **A1** and **A3**.

3.2. Proof of convergence

The technicalities of the convergence proof for SMTA are contained in the following three lemmas.

LEMMA 1. *Let $x \in \mathbb{R}^n$ be a non-stationary point. Then there exist positive numbers δ_1, δ_2 and ε_1 such that for $x_k \in \mathbb{R}^n$*

$$\begin{array}{l} \|x_k - x\| \leqslant \delta_1 \quad \text{and} \quad \Lambda_k \leqslant \delta_2 \\ \Downarrow \\ \Delta S_k(h_k) \leqslant -\varepsilon_1 \Lambda_k \qquad \text{if} \quad k \geqslant \tilde{k} \end{array}$$

Proof. Since x is a non-stationary point there exist, by Proposition 1, a direction h such that $F'_h(x) < 0$. Then it follows from Proposition 2 that there exist a point $x + d$, $d = th$, $t > 0$, such that $\Delta F(x; d) < 0$.

Define d_k by

$$x_k + d_k = x + d \ .$$

If $x_k \to x$ then $d_k \to d$. Therefore it follows from the uniform continuity of f, f' and H that there exists a neighbourhood $\mathcal{N}(x)$ of x and a number δ such that

$$\Delta F(x_k; d_k) \leqslant -\delta < 0 \tag{13}$$

when $x_k \in \mathcal{N}(x)$. Define $\delta_1 > 0$ and $\delta_2 > 0$ such that if $\|x_k - x\| \leqslant \delta_1$ then $x_k \in \mathcal{N}(x)$ and $\|d_k\| \geqslant \delta_2$.

Let $h_k^t \in I\!R^n$ be a solution to (4) subject to the trust region:

$$\begin{aligned} h_k^t \in \arg\min_h \ & H(f_k(h)) \\ \text{s.t.} \ & x_k + h \in T_k \end{aligned} \tag{14}$$

Suppose $\Lambda_k \leqslant \delta_2$. Let $t_k = \Lambda_k / \|d_k\|$ and $q_k = t_k d_k$. Then $x_k + q_k \in T_k$ and since $H(f_k(h)) = \Delta F(x_k; h) + F(x_k)$ we can use the definition of h_k^t and Proposition 3 to obtain

$$\Delta F(x_k; h_k^t) \leqslant \Delta F(x_k; q_k) \leqslant t_k \Delta F(x_k; d_k) \ . \tag{15}$$

Since $\{d_k\}$ is bounded away from 0 this implies the existence of $\varepsilon > 0$, independent of k, such that

$$\Delta F(x_k; h_k^t) \leqslant -\varepsilon \Lambda_k \tag{16}$$

(using (13)). Since H is locally Lipschitz, there exists a constant K such that

$$|\Delta F(x_k; h_k^t)| \leqslant K \|h_k^t\|$$

and thus (16) implies

$$\|h_k^t\| \geqslant \frac{\varepsilon}{K} \Lambda_k \ . \tag{17}$$

Now let $u_k = \|h_k^t\| / \|d_k\|$. The arguments showing (15) imply

$$\Delta F(x_k; h_k^t) \leqslant u_k \Delta F(x_k; d_k) \ .$$

Using (13) and the definition of ΔF, this implies

$$H(f_k(h_k^t)) \leqslant -u_k \delta + H(f(x_k)) \ . \tag{18}$$

Because of (17), property (12), **A2**, and the boundedness away from 0 of $\{\|d_k\|\}$, we have the following inequalities for all sufficiently large values of k, say $k \geqslant \tilde{k}$,

$$\begin{aligned} |w_k H(f(x_k))| &\leqslant \frac{\delta}{4} u_k \ , \\ w_k H(c(p_k(h_k^t))) &\leqslant \frac{\delta}{4} u_k \ , \\ w_k &\leqslant \frac{1}{4} \ . \end{aligned}$$

Therefore, using the convexity of H and (18),

$$\begin{aligned} H(s_k^{w_k}(h_k^t)) &\leqslant w_k H(c(p_k(h_k^t))) + (1 - w_k) H(f_k(h_k^t)) \\ &\leqslant \frac{\delta}{4} u_k + (1 - w_k)(-u_k \delta + H(f(x_k))) \\ &= \frac{\delta}{4} u_k - u_k \delta + H(f(x_k)) + w_k u_k \delta - w_k H(f(x_k)) \\ &\leqslant -\frac{\delta}{4} u_k + H(f(x_k)) \ . \end{aligned}$$

Since h_k minimizes $H \circ s_k^{w_k}$ subject to the trust region, it follows that

$$\begin{aligned} \Delta S_k(h_k) &= H(s_k^{w_k}(h_k)) - H(f(x_k)) \\ &\leqslant H(s_k^{w_k}(h_k^t)) - H(f(x_k)) \\ &\leqslant -\frac{\delta}{4} \|h_k^t\| / \|d_k\| \\ &\leqslant -\frac{\delta}{4} \Lambda_k / \|d_k\| \ , \end{aligned}$$

which proves Lemma 1. □

LEMMA 2. *Let $x \in \mathbb{R}^n$ be a non-stationary point. Let δ_1 be defined as in Lemma 1. For every $\delta_3 > 0$ there exists $\varepsilon_2 > 0$ such that*

$$\|x_k - x\| \leqslant \delta_1 \quad \text{and} \quad \Lambda_k \geqslant \delta_3$$

$$\Downarrow$$

$$\Delta S_k(h_k) \leqslant -\varepsilon_2 \qquad \text{if} \quad k \geqslant \tilde{k}$$

Proof. Let $\delta_4 = \min\{\delta_2, \delta_3\}$, δ_2 being defined as in Lemma 1. Suppose $\tilde{h}_k$ is generated by (6) from x_k with the trust region bound $\tilde{\Lambda}_k = \delta_4$. Then it follows from Lemma 1 that, for $k \geqslant \tilde{k}$,

$$\Delta S_k(\tilde{h}_k) \leqslant -\varepsilon_1 \delta_4$$

Suppose h_k is generated from x_k by (6) with the local bound $\Lambda_k \geqslant \delta_3$. Then

$$\begin{aligned} \Delta S_k(h_k) &\leqslant \Delta S_k(\tilde{h}_k) \\ &\leqslant -\varepsilon_1\delta_4 \ , \end{aligned}$$

which proves Lemma 2. □

LEMMA 3. *If $\{x_k\}$ is convergent then the limit point is stationary.*

Proof. Suppose $x_k \to x$ for $k \to \infty$ and that x is non-stationary. From Assumptions **A1-A3** and (12) we obtain

$$\begin{aligned} F(x_k + h_k) &= H(f_k(h_k) + o(\|h_k\|)) \\ &= H(s_k^{w_k}(h_k) - w_k\,(c(p_k(h_k)) - f_k(h_k)) + o(\|h_k\|)) \\ &= H(s_k^{w_k}(h_k)) + \Lambda_k o(1) + o(\|h_k\|) \\ &= H(s_k^{w_k}(h_k)) + \Lambda_k o(1) \end{aligned}$$

Therefore

$$F(x_k) - F(x_k + h_k) = -\Delta S_k(h_k) + \Lambda_k o(1) \tag{19}$$

Let δ_2 be defined as in Lemma 1. Using Lemma 1 if $\Lambda_k \leqslant \delta_2$ and Lemma 2 if $\Lambda_k > \delta_2$ we find from (19)

$$F(x_k) - F(x_k + h_k) = -\Delta S_k(h_k)(1 + o(1))$$

Therefore Lemma 1 and the rules (7), (8) and (10) for accepting a new point and for adjusting Λ_k imply the existence of $\delta > 0$, with $\delta \leqslant \delta_2$ such that for sufficiently large k,

$$\Lambda_k \leqslant \delta \quad \Rightarrow \quad x_{k+1} = x_k + h_k \ \text{ and } \ \Lambda_{k+1} = K_2\|h_k\| \tag{20}$$

where $K_2 > 1$.
(20) implies that if $\Lambda_k \leqslant \delta$, then $\Lambda_{k+1} > \Lambda_k$. Hence the sequence of local bounds must be bounded away from zero,

$$\Lambda_k \geqslant \delta_3 > 0 \quad \text{for} \ \ k = 0, 1, 2, 3, \ldots \tag{21}$$

Therefore an infinite number of proposals $(x_k + h_k)$ are accepted by the algorithm, because otherwise we would have $\Lambda_k \to 0$ for $k \to \infty$ (using the fact that the bound is decreased linearly when a point is not accepted (see (9))). Furthermore, when a proposal $(x_k + h_k)$ is accepted and k is sufficiently large, then (8), Lemma 2 and (21) imply

$$F(x_k) - F(x_k + h_k) \geqslant \tilde{\delta}_1[-\Delta S_k(h_k)] \geqslant \tilde{\delta}_1\varepsilon_2 > 0 \ .$$

Since the sequence of function values $F(x_k)$ is non-increasing we obtain $F(x_k) \to -\infty$ for $k \to \infty$. This is a contradiction since $x_k \to x$ and

F is continuous at x. Thus this assumption is wrong: x is a stationary point. □

The following theorem extends the result of Lemma 3.

THEOREM 1. *Let S be the set of stationary points of (1), $S = \{ x \in \mathbb{R}^n \mid 0 \in \partial F(x) \}$. Let $d(x_k, S)$ be the distance between x_k and S. Then*

$$d(x_k, S) \to 0 \quad \text{for} \quad k \to \infty$$

Proof. Suppose $d(x_k, S) \not\to 0$. Then infinitely many points x_k must be bounded away from S, and hence Assumption **A4** implies that the sequence $\{x_k\}$ must have a cluster point x which is not stationary. According to Lemma 3 $\{x_k\}$ cannot converge to x. Thus, for all small $\varepsilon > 0$ infinitely many iterates x_k must have a distance less than ε from x and infinitely many must have a distance larger than 2ε from x. Let $\varepsilon > 0$ be chosen smaller than $\delta_1/2$, δ_1 being the bound used in Lemma 1 and Lemma 2. Then we shall prove that if $\|x_k - x\| < \varepsilon$ and $\|x_{k+p} - x\| > 2\varepsilon$ we have

$$F(x_k) - F(x_{k+p}) \geqslant \delta > 0 \,, \tag{22}$$

δ being independent of k and p. Since (22) holds for infinitely many values of k, and since the sequence $\{F(x_k)\}$ is non-increasing, we obtain that $F(x_k) \to -\infty$ for $k \to \infty$ which contradicts that F is continuous at x and $\{x_k\}$ converges to x. Therefore the result follows as a consequence of (22).

Equation (22) is proved by the following argument: Consider

$$F(x_k) - F(x_{k+p}) = \sum_{j=k}^{k+p-1} [F(x_j) - F(x_{j+1})] \,, \tag{23}$$

for $k \geqslant \tilde{k}$, with $\tilde{k}$ as in Lemma 1. The terms of the sum are non-negative. Suppose that $x_{j+1} \neq x_j$, i.e. the increment h_j is accepted. Suppose further that if $\|x_j - x\| \leqslant 2\varepsilon$ then we can use the Lemmas 1 and 2. We obtain from (8) and these lemmas,

$$\begin{aligned} F(x_j) - F(x_{j+1}) \;&\geqslant\; \tilde{\delta}_1[-\Delta S_j(h_j)] \\ &\geqslant \begin{cases} \tilde{\delta}_1 \varepsilon_1 \Lambda_j & \text{if } \Lambda_j \leqslant \delta_2 \\ \tilde{\delta}_1 \varepsilon_2 & \text{otherwise} \end{cases} \end{aligned} \tag{24}$$

Equation (22) now follows from (23) using (24): Let $\mathcal{A}_k$ be the index set corresponding to the terms in (23) with $x_j \neq x_{j+1}$. If, for all of these terms, we have $\Lambda_j \leqslant \delta_2$ then

$$\sum_{j \in \mathcal{A}_k} [F(x_j) - F(x_{j+1})] \geqslant \tilde{\delta}_1 \varepsilon_1 \sum_{j \in \mathcal{A}_k} \Lambda_j \geqslant \tilde{\delta}_1 \varepsilon_1 \sum_{j \in \mathcal{A}_k} \|h_j\| \tag{25}$$

The last sum exceeds ε since $h_j = x_{j+1} - x_j$ for $j \in \mathcal{A}_k$, $x_j = x_{j+1}$ for $j \notin \mathcal{A}_k$, and since $\|x_{k+p} - x_k\| > \varepsilon$. Thus the sum in (23) exceeds ε when $\Lambda_j \leqslant \delta_2$ for all $j \in \mathcal{A}_k$. If the last condition is not fulfilled then there exists $j \in \mathcal{A}_k$ with

$$F(x_j) - F(x_{j+1}) \geqslant \tilde{\delta}_1 \varepsilon_2$$

so in that case the sum exceeds a positive number which is independent of k and t. Thus we have proved (22) with

$$\delta = \min\{\tilde{\delta}_1 \varepsilon_1 \varepsilon, \tilde{\delta}_1 \varepsilon_2\}$$

This completes the proof of Theorem 1. □

4. Conclusion

We have considered the problem of minimizing functions of the type $H \circ f$, where $f : \mathbb{R}^n \mapsto \mathbb{R}^m$ is smooth and $H : \mathbb{R}^m \mapsto \mathbb{R}$ is convex. It is proved that the hybrid space mapping algorithm described in (Bakr et al., 2001) has guaranteed global convergence to the set of stationary points of $H \circ f$. The proof covers a class of hybrid algorithms, including those presented in (Bakr et al., 2001) and (Pedersen, 2001).

References

M.H. Bakr, J.W. Bandler, K. Madsen, J. Søndergaard, *Review of the Space Mapping Approach to Engineering Optimization and Modelling*, Optimization and Engineering, vol. 1, pp. 241–276, 2000.

M.H. Bakr, J.W. Bandler, K. Madsen and J. Søndergaard, *An introduction to the space mapping technique*, Optimization and Engineering, vol. 2, pp. 369–384, 2001.

J.W. Bandler, R.M. Biernacki, S.H. Chen, P.A. Grobelny and R.H. Hemmers, *Space mapping technique for electromagnetic optimization*, IEEE Trans. Microwave Theory Tech., vol. 42, pp. 2536–2544, 1994.

C.G. Broyden, *A Class of Methods for Solving Non-Linear Simultaneous Equations*, Math. Comp., vol. 19, pp. 577–593, 1965.

F.H. Clarke, *Generalized Gradients and Applications*, Trans. Am. Maths. Society, vol. 205, pp. 247–262, 1975.

F.H. Clarke, *Optimization and Nonsmooth Analysis*, Canadian Mathematical Society Series of Monographs and Advanced Texts, John Wiley & Sons pp. 1–308, 1983.

R. Fletcher, *A Model Algorithm for Composite Nondifferentiable Optimization Problems*, in Nondifferentiable and Variational Techniques in Optimization, D.C. Sorensen, R.J.-B. Wets, eds., Mathematical Programming Study 17, North-Holland, Amsterdam, 1982.

J. Hald, K. Madsen, *Combined LP and Quasi-Newton Methods for Non-Linear L1 Optimization*, SIAM J.Num.Anal. 22, pp. 369–384 1985.

K. Madsen, *An Algorithm for Minimax Solution of Overdetermined Systems of Non-linear Equations*, J. Inst. Math. Appl. 16 pp. 321–328, 1975.

K. Madsen, *Minimization of Non-Linear Approximation Functions*, Dr. Techn. Thesis, Institute for Numerical Analysis, Technical University of Denmark, pp. 1–141, 1985.

F.Ø. Pedersen, *Advances on the Space Mapping Optimization Method*, Master Thesis, IMM-THESIS-2001–35, Informatics and Mathematical Modelling, Technical University of Denmark, 2001.

L.N. Vicente, *Space mapping: Models, Sensitivities, and Trust-regions Methods*, to appear in Optimization and Engineering, 2003.

R.S. Womersley, R. Fletcher, *An Algorithm for Composite Nonsmooth Optimization Problems*, J. Opt. Theo. Applns., 48, 493–523, 1986.

Chapter 7

Conclusion

The aim of this study has been to provide an overview of the field of surrogate optimization and to examine theoretical properties of the space mapping technique. This chapter summarizes the conclusions of the study.

Chapter 2 concerns a literature overview of the field of surrogate modelling and optimization. The presentation divides the field into two parts, the methods for generating surrogate models and methods for conducting optimization using surrogate models. The surrogate models are again divided into two categories, the functional models and the physical models. Where the functional models are generic mathematical models which can be constructed without any particular knowledge of the underlying physical system. The physical models are system specific models. Surrogates based on physical models are usually constructed by manipulating a cheaper model of the same physical system as the expensive model in question, so that the manipulated cheap model approximates the behaviour of the expensive model. The chapter also presents four algorithms for optimization using surrogate models. Two of these can be proved convergent.

The space mapping technique is one such method for constructing and optimizing a surrogate model based on a cheap physical model. Here we use the name coarse model to denote the cheap model, and the name fine model to denote the expensive model, which is to be optimized. The space mapping sur-

rogate is the coarse model composed with a parameter mapping, the so-called space mapping, connecting similar responses of the coarse and the fine model. The space mapping technique is the focal point of the succeeding chapters of the thesis.

Chapter 3 provides an introduction and motivation for the space mapping technique. The basic principles of the space mapping technique are presented. It is shown how the space mapping technique can be combined with classical optimization strategies in a hybrid method. The hybrid method is illustrated by two test problems, and the space mapping surrogate is shown empirically to be a valid approximation in a larger area than a corresponding linear Taylor model. The space mapping technique is by example shown to be an efficient pre-processing technique for difficult engineering optimization problems. If the solution accuracy is not sufficient, the technique can be combined with other methods of optimization.

Chapter 4 concerns theoretical aspects of the space mapping technique. The chapter presents theoretical results which characterize the space mapping under some ideal conditions. It is shown that if these conditions are met, the solutions provided by the original space mapping technique are minimizers of the fine model. However, in practice we cannot expect that the ideal conditions are met, so the space mapping technique should be combined with classical optimization methods in order to be convergent. The theoretical results are motivated and illustrated by numerical examples. Deficiencies of the usual space mapping definition are discussed and four alternative definitions are reviewed. The two space mapping definitions relying on respectively gradient information and multiple points are identified to be the most promising. But further theoretical investigations are needed in order to arrive at a more firm conclusion.

Chapter 4 also discuses the approximation abilities of the coarse model composed with the space mapping. A numerical example confirms the theoretical results, that the mapped coarse model, with a Taylor approximation to the space mapping, has a lower approximation error for long steps, compared to a Taylor model of the fine model. For short steps, however, the Taylor model of the fine model is best, due to exact interpolation at the model origin. It is also shown how a response correction may enhance the mapped coarse model approximation, without compromising the small approximation error on long steps. With the response correction, the mapped coarse model approximation has the same interpolation property as the Taylor model of the fine model.

Chapter 5 concerns formulation of optimization algorithms based on the space

mapping technique. Five algorithms are presented. Two of the algorithms are concerned with the original space mapping method. The other three algorithms are hybrid algorithms combining the space mapping method with a classical Taylor based optimization method. The key advantage of the hybrid algorithms over the original space mapping methods is that convergence to a stationary point of the fine model can be proved. On the other hand, the original space mapping methods are conceptually simpler and also easier to implement.

The performance of a Matlab implementation of the algorithms are reported for 7 test problems. The effects of using alternative space mapping definitions is demonstrated on a test problem. The conclusions from these tests are that we only recommend using the original space mapping algorithms in the special case where only few fine model evaluations are available and where limited accuracy can be accepted. Otherwise we suggest to use, either one of the hybrid space mapping algorithms, or a direct, classical algorithm initialized with derivative information obtained from the space mapped coarse model in the starting point.

Chapter 6 concerns convergence of hybrid space mapping algorithms. A framework for hybrid space mapping algorithms is presented. The framework covers algorithms with pre-conditioning of optimization problems using a broad class of surrogate models, not limited to those generated by the space mapping method. A proof is presented, guaranteeing global convergence to the set of stationary points of the fine model of the hybrid space mapping algorithms.

New Contributions

We list here the main new contributions provided by this study.

- A literature overview of surrogate modelling and optimization.
- Theoretical conditions under which the original space mapping formulation and the new space mapping formulation are the same.
- Ideal conditions under which it is proved that the space mapping method works.
- A framework for hybrid space mapping algorithms.

- Convergence is proved for a class of algorithms using pre-conditioning surrogate models, not limited to those provided by the space mapping method.
- Comparison of 5 space mapping algorithms and a direct, classical optimization algorithm on 7 test problems.
- A Matlab toolbox with 13 space mapping test problems, and 6 optimization algorithms.

Unresolved Issues

We list here some of the issues that the present study does not resolve.

- An understanding of how to define, if at all possible, a space mapping that meets the ideal theoretical conditions derived in this study.
- How to establish a space mapping for problems where the coarse and the fine model parameter spaces are of unequal dimensions.
- A theoretical study of frequency space mapping and implicit space mapping, where the connection between the coarse and the fine model is found by manipulating state variables (e.g. frequency, time or preassigned variables) of the coarse model.

Complete List of References

- N. Alexandrov, J.E. Dennis, Jr., R.M. Lewis, V. Torczon, *A Trust Region Framework for Managing the Use of Approximation Models in Optimization*, Structural Optimization, vol. 15, no. 1, pp. 16-23, 1998.

- C. Audet, J.E. Dennis, Jr., L.N. Vicente (eds.), *Surrogate Optimization*, forthcoming special issue of Optimization and Engineering, 2003.

- M.H. Bakr, *Advances in Space Mapping Optimization of Microwave Circuits*, Ph.D. Thesis, McMaster University, 2000.
  Available at `http://www.sos.mcmaster.ca/theses.htm`

- M.H. Bakr, J.W. Bandler, R.M. Biernacki, S.H. Chen, K. Madsen, *A Trust Region Aggressive Space Mapping Algorithm for EM Optimization*, IEEE Trans. Microwave Theory Tech. vol. 46, no. 12, pp. 2412–2425, 1998.

- M.H. Bakr, J.W. Bandler, N. Georgieva, *An Aggressive Approach to Parameter Extraction*, IEEE Trans. Microwave Theory Tech., vol. 47, pp. 2428–2439, 1999.

- M.H. Bakr, J.W. Bandler, N.K. Georgieva, K. Madsen, *A Hybrid Aggressive Space Mapping Algorithm for EM Optimization*, IEEE Trans. Microwave Theory Tech., vol. 47, pp. 2440–2449, 1999.

- M.H. Bakr, J.W. Bandler, K. Madsen, J.E. Rayas-Sánchez, J. Søndergaard, *Space Mapping Optimization of Microwave Circuits Exploiting*

Surrogate Models, IEEE Trans. Microwave Theory Tech., vol. 48, no. 12, pp. 2297–2306, 2000.

◇ M.H. Bakr, J.W. Bandler, K. Madsen, J. Søndergaard, *An Introduction to the Space Mapping Technique*, Optimization and Engineering, vol. 2, no. 4, pp. 369–384, 2001.

◇ M.H. Bakr, J.W. Bandler, K. Madsen, J. Søndergaard, *Review of the Space Mapping Approach to Engineering Optimization and Modeling*, Optimization and Engineering, vol. 1, no. 3, pp. 241–276, 2000.

◇ J.W. Bandler, R.M. Biernacki, S.H. Chen, *Fully Automated Space Mapping Optimization of 3D Structures*, IEEE MTT-S IMS Digest, San Francisco, CA, pp. 753–756, 1996.

◇ J.W. Bandler, R.M. Biernacki, S.H. Chen, P.A. Grobelny, R.H. Hemmers, *Space Mapping Technique for Electromagnetic Optimization*, IEEE Trans. Microwave Theory Tech., vol. 42, pp. 2536–2544, 1994.

◇ J.W. Bandler, R.M. Biernacki, S.H. Chen, R.H. Hemmers, K. Madsen, *Electromagnetic Optimization Exploiting Aggressive Space Mapping*, IEEE Trans. Microwave Theory Tech., vol. 43, pp. 2874–2882, 1995.

◇ J.W. Bandler, R.M. Biernacki, S.H. Chen, Y.F. Huang, *Design Optimization of Interdigital Filters using Aggressive Space Mapping and Decomposition*, IEEE Trans. Microwave Theory Tech., vol. 45, pp. 761–769, 1997.

◇ J.W. Bandler, Q. Cheng, S. Dakroury, A.S. Mohamed, M.H. Bakr, K. Madsen, J. Søndergaard, *Space Mapping: The State of the Art*, submitted, IEEE Trans. Microwave Theory Tech., 2004.

◇ J.W. Bandler, K. Madsen (eds.), *Surrogate Modelling and Space Mapping for Engineering Optimization*, (special issue) Optimization and Engineering, no. 2, 2001.

◇ J.W. Bandler, A.S. Mohamed, M.H. Bakr, K. Madsen, J. Søndergaard, *EM-Based Optimization Exploiting Partial Space Mapping and Exact Sensitivities*, IEEE Trans. Microwave Theory Tech., vol. 50, no. 12, pp. 2741–2750, 2002.

◇ J.-F.M. Barthelemy, R.T. Haftka, *Approximation Concepts for Optimum Structural Design - a Review*, Structural Optimization, vol. 5, pp. 129–144, 1993.

◇ A.J. Booker, J.E. Dennis, Jr., P.D. Frank, D.B. Serafini, V. Torczon, *Optimization Using Surrogate Objectives on a Helicopter Test Example*, Technical report 97-31, Department of Computational and Applied Mathematics, Rice University, Houston, Texas 77005-1892, 1997.

◇ A.J. Booker, J.E. Dennis, Jr., P.D. Frank, D.B. Serafini, V. Torczon, M.W. Trosset, *A Rigorous Framework for Optimization of Expensive Functions by Surrogates*, Structural Optimization, vol. 17, pp. 1–13, 1999.

◇ G.E.P. Box, N.R. Draper, *Empirical Model-Building and Response Surfaces*, Wiley, 1987.

◇ G.E.P. Box, K.B. Wilson, *On the Experimental Attaintment of Optimum Conditions*, Journal of the Royal Statistical Society, Series B, vol. 13, pp. 1–45, 1951.

◇ C.G. Broyden, *A Class of Methods for Solving Non-Linear Simultaneous Equations*, Math. Comp., vol. 19, pp. 577–593, 1965.

◇ S. Burgee, A. Giunta, V. Balabanov, B. Grossman, W. Mason, R. Narducci, R. Haftka, L. Watson, *A Coarse-grained Parallel Variable-Complexity Multidisciplinary Optimization Problem*, The International Journal of Supercomputing Applications and High Performance Computing, vol. 10, pp. 269–299, 1996.

◇ R.H. Byrd, J. Nocedal, R.B. Schnabel, *Representations of Quasi-Newton Matrices and Their Use in Limited Memory Methods*, Mathematical Programming, vol. 63, pp. 129–156, 1994.

◇ K.J. Chang, R.T. Haftka, G.L. Giles, P.-J. Kao, *Sensitivity-Based Scaling for Approximating Structural Response*, Journal of Aircraft, vol. 30, 1993.

◇ F.H. Clarke, *Generalized Gradients and Applications*, Trans. Am. Maths. Society, vol. 205, pp. 247–262, 1975.

◇ F.H. Clarke, *Optimization and Nonsmooth Analysis*, Canadian Mathematical Society Series of Monographs and Advanced Texts, John Wiley & Sons, pp. 1–308, 1983.

◇ A.R. Conn, N.I.M. Gould, P.L. Toint, *Trust-Region Methods*, SIAM, Philadelphia, 2000.

◇ C. Currin, T. Mitchell, M. Morris, D. Ylvisaker, *Bayesian Prediction of Deterministic Functions, With Applications to the Design and Analysis of Computer Experiments*, Journal of the American Statistical Association, vol. 86, no. 416, pp. 953–963, 1991.

◇ V. Czitrom, P.D. Spagon, *Statistical Case Studies for Industrial Process Improvement*, ASA-SIAM series on statistics and applied probability, 1997.

◇ I. Daubechies, *Ten Lectures on Wavelets*, Philadelphia, SIAM, 1992.

◇ J.E. Dennis, Jr., *Surrogate Modelling and Space Mapping for Engineering Optimization: A Summary of the Danish Technical University November 2000 Workshop*, Technical Report, TR00-35, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, USA, 2000.

◇ J.E. Dennis, Jr., Dept. Computational and Applied Mathematics, Rice University, Houston, Texas, USA, private discussions, 2001.

◇ J.E. Dennis, Jr., R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

◇ J.E. Dennis, Jr., V. Torczon, *Managing Approximation Models in Optimization*, in Multidisciplinary Design Optimization: State-of-the-Art, N.M. Alexandrov, M.Y. Hussaini, eds., SIAM, Philadelphia, 1997.

◇ O. Dubrule, *Two Methods with Different Objectives: Splines and Kriging*, Mathematical Geology, vol. 15, no. 2, 1983.

◇ R. Fletcher, *A Model Algorithm for Composite Nondifferentiable Optimization Problems*, in Nondifferentiable and Variational Techniques in Optimization, D.C. Sorensen, R.J.-B. Wets, eds., Mathematical Programming Study 17, North-Holland, Amsterdam, 1982.

◇ A.A. Giunta, V. Balabanov, D. Haim, B. Grossman, W.H. Mason, L.T. Watson, R.T. Haftka, *Multidisciplinary Optimization of a Supersonic Transport Using Design of Experiments Theory and Response Surface Modelling*, The Aeronautic Journal, pp. 347–356, Oct. 1997.

◇ H.-M. Gutmann, *A Radial Basis Function Method for Global Optimization*, Journal of Global Optimization, no. 19, pp. 201–227, 2001.

◇ R.T. Haftka, E.P. Scott, *Optimization and Experiments — a Survey*, in Tatsumi, T. et al (eds.), Theoretical and Applied Mechanics 1996, Proc. of XIX ICTAM, Kyoto, 1996, Elsevier, pp. 303-321, 1997.

◇ J. Hald, K. Madsen, *Combined LP and Quasi-Newton Methods for Minimax Optimization*, Mathematical Programming, vol. 20, pp. 49–62, 1981.

◇ J. Hald, K. Madsen, *Combined LP and Quasi-Newton Methods for Non-Linear L_1 Optimization*, SIAM J.Num.Anal. 22, pp. 369–384, 1985.

◇ T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning — Data Mining, Inference and Prediction*, Springer, 2001.

◇ C.R. Hicks, *Fundamental Concepts in the Design of Experiments*, Holt, Rinehart and Winston, Inc, 1964.

◇ Howard Demuth, Mark Beale, *Neural Network Toolbox User's Guide*, The Mathworks, Inc., 1994.

◇ K.E. Johnson, K.W. Bauer, J.T. Moore, M. Grant, *Metamodelling Techniques in Multidimensional Optimality Analysis for Linear Programming*, Mathl. Comput. Modelling, vol. 23, no. 5, pp. 45–60, 1996.

◇ D.R. Jones, M. Schonlau, W.J. Welch, *Efficient Global Optimization of Expensive Black-Box Functions*, Journal of Global Optimization, vol. 13, no. 4, pp. 455-492, 1998.

◇ S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, *Optimization by Simulated Annealing*, Science, vol. 220, pp. 671–680, 1993.

◇ S.J. Leary, A. Bhaskar, A.J. Keane, *A Constraint Mapping Approach to the Structural Optimization of an Expensive Model using Surrogates*, Optimization and Engineering, vol. 2, no. 4, pp. 385–398, 2001.

◇ S.N. Lophaven, H.B. Nielsen, J. Søndergaard, *Aspects of the Matlab Toolbox DACE*, technical report IMM-REP-2002-13, Informatics and Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 2002.

◇ S.N. Lophaven, H.B. Nielsen, J. Søndergaard, *DACE – A Matlab Kriging Toolbox, Version 2.0,* technical report IMM-REP-2002-12, Informatics and Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 2002.

◇ K. Madsen, *An Algorithm for Minimax Solution of Overdetermined Systems of Non-linear Equations*, J. Inst. Math. Appl., vol. 16 pp. 321–328, 1975.

◇ K. Madsen, *Minimax Solution of Non-Linear Equations Without Calculating Derivatives*, Mathematical Programming, vol. 3, pp. 110–126, 1975.

◇ K. Madsen, *Minimization of Non-Linear Approximation Functions*, dr. techn. thesis, Institute for Numerical Analysis, Technical University of Denmark, 1986.

◇ K. Madsen, J. Søndergaard, *Convergence of Hybrid Space Mapping Algorithms*, submitted, Optimization and Engineering, 2003.

◇ G. Matheron, *Principles of Geostatistics*, Economic Geology, vol. 58, pp. 1246–1266, 1963.

◇ Matlab^{TM} version 6.5 and Matlab^{TM} Optimization Toolbox version 2.2., The MathWorks, Inc., 3 Apple Hill Drive, Natick MA 01760-2098, 2003.

◇ M.D. McKay, W.J. Conover, R.J. Beckman, *A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code*, Technometrics, vol. 21, no. 2, 1979.

◇ J.J. Moré, *Recent Developments in Algorithms and Software for Trust Region Methods*, Mathematical Programming: The State of The Art, pp. 258–287, 1982.

◇ R.H. Myers, D.C. Montgomery, *Response Surface Methodology*, second edition, John Wiley & Sons, Inc., 2002.

◇ H.B. Nielsen, Matlab implementation of the Levenberg-Marquardt method for nonlinear least-squares, IMM, DTU, 2001.
Available at `http://www.imm.dtu.dk/~hbn/Software/`

◇ H.B. Nielsen (ed.), *Surrogate Modelling and Space Mapping for Engineering Optimization — Papers*, from the "First International Workshop on Surrogate Modelling and Space Mapping for Engineering Optimization", Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, November 16–18, 2000.

◇ J. Nocedal, S.J. Wright, *Numerical Optimization*, Springer Series in Operations Research, Springer, 1999.

◇ F.Ø. Pedersen, *Advances on the Space Mapping Optimization Method*, Master Thesis, IMM-THESIS-2001-35, Informatics and Mathematical Modelling, Technical University of Denmark, 2001.

◇ F.Ø. Pedersen, Technical University of Denmark, Lyngby, Denmark, private discussions, 2001.

◇ M.J.D. Powell, *The Theory of Radial Basis Funcion Approximation in 1990*, in Advances in Numerical Analysis, vol. 2, W. Light (ed.), Oxford University Press, 1992.

◇ D.M. Pozar, *Microwave Engineering*, second edition, John Wiley, New York, 1998.

◇ G.F. Roach, *Green's Functions*, second edition, Cambridge University Press, 1982.

◇ J.J. Royer, P.C. Vieira, *Dual Formalism of Kriging*, in Geostatistics for Natural Resources Characterization, Part 2, ed. G. Verly, M. David, A.G. Journel, A. Marechal, NATO ASI Series C: Mathematical and Physical Sciences, Vol. 122, pp. 691–702, D. Reidel Pub. Co., Dordrecht, 1984.

◇ J. Sacks, W.J. Welch, T.J. Mitchell, H.P. Wynn, *Design and Analysis of Computer Experiments*, Statistical Science, vol. 4, no. 4, pp. 409–435, 1989.

◇ D.B. Serafini, *A Framework for Managing Models in Nonlinear Optimization of Computationally Expensive Functions*, Ph.D. Thesis, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, USA, 1998.

◇ J. Søndergaard, *Non-Linear Optimization Using Space Mapping*, Master Thesis, IMM-EKS-1999-23, Department of Mathematical Modelling, Technical University of Denmark, 1999.
Available at `http://www.imm.dtu.dk/~km/jsmaster.ps.gz`

◇ V. Torczon, *On the convergence of pattern search algorithms*, SIAM Journal on Optimization, vol. 7, no. 1, pp. 1–25, 1997.

◇ V. Torczon, M.W. Trosset, *Using Approximations to Accelerate Engineering Design Optimization*, Tech. Report 98-33, Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, Virginia 23681-2199, 1998.

◇ V.V. Toropov, *Multipoint Approximation Method in Optimization Problems with Expensive Function Values*, in Computational Systems Analysis, A. Sydow (ed.), Elsevier, 1992.

◇ V.V. Toropov, *Simulation Approach to Structural Optimization*, Structural Optimization, vol. 1, pp. 37–46, 1989.

◇ V.V. Toropov, A.A. Filatov, A.A. Polynkin, *Multiparameter Structural Optimization Using FEM and Multipoint Explicit Approximations*, Structural Optimization, vol. 6, pp. 7–14, 1993.

◇ M.W. Trosset, V. Torczon, *Numerical Optimization Using Computer Experiments*, Tech. Report 97-38, Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, Virginia 23681-2199, 1997.

◇ L.N. Vicente (organizer), *School on Optimization using Surrogates for Engineering Design*, Center for Mathematics, University of Coimbra, Portugal, May 8–11, 2002.

◇ L.N. Vicente (ed.), *Second Workshop on Nonlinear Optimization: "Theoretical Aspects of Surrogate Optimization"*, abstracts, workshop held by Center for Mathematics, University of Coimbra, Portugal, May 16–17, 2002.

◇ L.N. Vicente, *Space Mapping: Models, Sensitivities and Trust-region Methods*, to appear in Optimization and Engineering, 2003.

◇ G.G. Walter, *Wavelets and Other Orthogonal Systems With Applications*, CRC Press, Inc., 1994.

◇ G.S. Watson, *Smoothing and Interpolation by Kriging and with Splines*, Mathematical Geology, vol. 16, no. 6, 1984.

◇ R.S. Womersley, R. Fletcher, *An Algorithm for Composite Nonsmooth Optimization Problems*, J. Opt. Theo. Applns., vol. 48, 493–523, 1986.

◇ S. Yesilyurt, A.T. Patera, *Surrogates for Numerical Simulations; Optimization of Eddy-Promoter Heat Exchangers*, Computer Methods in Applied Mechanics and Engineering, vol. 121, pp. 231–257, 1995.

Appendix A

Space Mapping Toolbox

This manual describes a Matlab toolbox with space mapping optimization algorithms and test problems. Version 1.0 of the toolbox is covered by this manual.

The problems to be solved by the optimization algorithms in this toolbox have two models available: One model denoted the *fine model*, being the model of primary interest, and the other denoted the *coarse model*. The fine model is often expensive to evaluate, though this is not always the case with the simple test problems in this toolbox. It is expected that the coarse model somehow resembles the behaviour of the fine model. Further, it is expected that the coarse model is cheaper to evaluate than the fine model, and therefore it is most likely less accurate than the fine model.

The optimization algorithms employ the coarse model in the search for the fine model minimizer. This in done through a parameter mapping, the so-called space mapping, which in effect makes the coarse model behave as the fine model. We call this combination of the space mapping and the coarse model, the mapped coarse model. Hence, in the space mapping technique, this mapped coarse model is to take the place of the fine model in search for a minimizer of the latter. For a more thorough introduction to the space mapping technique see [1].

This manual is divided into three sections. The first section introduces the

test problems, and the second section introduces the algorithms. Both sections provide a brief description of the Matlab interface. The last section consists of two small examples of running the software.

We should note here that the optimization algorithms in this toolbox rely on the Matlab optimization toolbox [5] in order to run. The test problems do not require the Matlab optimization toolbox. The toolbox has been developed with Matlab version 6.5 (R13), though it should work with other recent versions of Matlab.

A.1 Space Mapping Optimization Test Problems

We first describe the common interface to the models, and thereafter we briefly introduce the individual test problems.

A.1.1 Interface

The definitions of the test problems are stored in the function `smprob`.

To obtain a structure for a given problem the call is

```
[prob, opts] = smprob(num, opts)
```

where the inputs are

`num`	the number of the wanted problem (see below),
`opts`	options for space mapping optimization algorithms (see `smopts`),

the outputs are

`prob`	structure with the problem definition,
`opts`	modified options.

The second input argument is optional, and is meant for the case where the user want to provide alternative default options, instead of those provided by `smopts`. The `opts` structure returned from `smprob` contains the problem specific parameters like initial trust region size etc.

The test problems are:

Tranmission Line Transformer problems
1 - two-section impedance transformer (TLT2)
2 - seven-section impedance transformer (TLT7)

Piston problem
3 - piston simulator (PISTON)

Rosenbrock problem
4 - rosenbrock function, with linear transformation (ROSEN)

Parallel Resonator problems
5 - exact linear mapping (RLCA)
6 - exact non-linear mapping (RLCB)
7 - inexact non-linear mapping (different topology) (RLCC)
8 - inexact non-linear mapping (RLCD)

Quadratic functions
9 - quada (coarse responses shifted up) (QUADA)
10 - quadb (coarse responses shifted down) (QUADB)

EM-Simulator
11 - inductive obstacle example (INDOBS)
12 - single resonator filter (SRESFIL)
13 - H-plane filter (HPLANEF)

The test problems are placed in separate directories in the toolbox. In order to access the problems, the path variable of Matlab is *automatically* modified when first interfacing the test problems through `smprob`. This modified path variable is temporary for the session. If the changes should be permanent use Matlabs `pathtool` to perform the changes.

A.1.2 The Test Problems

We now briefly introduce the individual test problems. But first some general comments:

Even though the optimization methods in the toolbox are general for all norms, the test problems presented here are posed as minimax problems.

Unless explicitly noted in the description the space mapping (using the usual formulation) is not perfect, hence $p(x^*) \neq z^*$. As described in [7, Chapter 4], this condition is critical for the success of the original space mapping algorithms, see also Section A.2 below.

All test examples except INDOBS, SRESFIL and HPLANEF are continuously differentiable in their parameters. The parameters of three mentioned test problems are defined on a discrete grid. Conformance with the grid is important when evaluating the models in the small electromagnetic simulator (in the directory `emsim`). So the parameters are automatically snapped to a nearby grid point before running the simulator. The simulator is provided by Mohamed Bakr from the Department of Electrical and Computer Engineering at McMaster University, Ontario, Canada. We will not discuss the details of the simulator in this manual.

TLT2

The problem TLT2 concerns the design of a twosection capacitively-loaded 10 : 1 impedance transformer. The exact physical origin of the problem is described in [1].

The designable parameters are the physical lengths of the two transmission lines. Eleven frequency points are simulated per sweep. The objective is to minimize the maximum input reflection coefficient over all simulated frequencies. The design specifications are that all input reflection coefficient responses should be below 50%.

Formally, the fine model response function is $f : \mathbb{R}^2 \mapsto \mathbb{R}^{11}$ and the specifications are

$$H(f(x)) = \max_{j=1}^{11}\{f_j(x)\} \leqslant 0.50\,.$$

The coarse model is as the fine model, except that coupling effects (modelled by capacitors) are not modelled.

TLT7

The TLT7 problem concerns the design of a seven-section capacitively-loaded impedance transformer. The exact physical origin of the problem is described in [1].

The designable parameters are the physical lengths of the seven transmission lines. 68 frequency points are simulated per sweep. The objective is to minimize the maximum input reflection coefficient over all simulated frequencies. The design specifications are that all input reflection coefficient responses should be below 7%.

Formally, the fine model response function is $f : \mathbb{R}^7 \mapsto \mathbb{R}^{68}$ and the specifications are

$$H(f(x)) = \max_{j=1}^{68}\{f_j(x)\} \leqslant 0.07\,.$$

The coarse model is as the fine model, except that coupling effects (modelled by capacitors) are not modelled.

PISTON

The PISTON problem is a data fitting problem, where a piston simulator should be fitted to a given target response. Here the piston simulator is a model which calculates the pressure over time at an oil producing one-dimensional well, relative to a fixed injection pressure. The target response is the fine model evaluated for a certain set of parameters, so the match of the model to the target response is exact at the optimal parameters. Because of this, we have chosen to formulate the problem as solving the nonlinear equations $f(x) = 0$ using the L_∞ merit function.

The fine model is a piston model with six sections of different reservoir permeabilities along the shaft of the well. Two of the six reservoir permeabilities are chosen as designable parameters. The coarse model is a piston model with two sections of different reservoir permeabilities along the shaft of the well. Both permeabilities in the coarse model are considered designable parameters. For both models, 20 simulation times are simulated per model evaluation.

Let the model response be $\tilde{f} : \mathbb{R}^2 \mapsto \mathbb{R}^{20}$ and let the target response be $y \in \mathbb{R}^{20}$. In the implementation we use the minimax merit for the nonlinear

equations, instead of the L_∞ merit, by introducing the residuals $f = (\tilde{f} - y, y - \tilde{f})^T$.

Formally, the deviations of the fine model response and the specifications are $f : \mathbb{R}^2 \mapsto \mathbb{R}^{40}$, and the optimization problem is

$$\min_x \max_{j=1}^{40} \{f_j(x)\} .$$

The deviation of the coarse model to the specifications is defined equivalently.

The PISTON problem is provided by Poul Erik Frandsen from Ticra Engineering Consultants, Copenhagen, Denmark.

ROSEN

The ROSEN problem involve solving the Rosenbrock equations, $f(x) = 0$, where $\tilde{f} : \mathbb{R}^2 \mapsto \mathbb{R}^2$,

$$\begin{aligned} f_1(x) &= 10 * (x_2 - x_1^2) \\ f_2(x) &= 1 - x_1 \end{aligned}$$

We formulate the problem as a minimax problem, by defining the fine model response function as $f = (f_1, f_2, -f_1, -f_2)^T$. Hence the problem is $\min_x \max_{j=1}^4 \{f_j(x)\}$.

We define the coarse model response as a linear transformation of the fine model response. Hence, $c(z) = f(Az + b)$, where

$$A = \begin{bmatrix} 1 & 2 \\ 5 & 0 \end{bmatrix} , \qquad b = \begin{bmatrix} -3 \\ 1 \end{bmatrix} .$$

We note that the space mapping between the coarse and the fine model is exact linear:

$$p(x) = A^{-1}(x - b) ,$$

since A is invertable.

It is easy to see that the mapping is perfect, i.e. $p(x^*) = z^*$, for this problem as the responses of both models vanish in their optimum.

RLC

The RLC problem concerns design of parallel RLC lumped resonators.

The coarse model is a parallel RLC lumped resonator with three designable parameters. 15 frequency points are simulated per sweep. The objective is to minimize the maximum deviation between the input reflection coefficient and some design specifications over all simulated frequencies. The specifications consists in a passband at the center frequencies and a stopband at all other frequencies.

The problem has four fine models that also model a parallel RLC lumped resonator, but the fine models also have some parasitic elements. The fine models are related to the same design problem (i.e. the same specifications) as the coarse model.

Here are the characteristics of the differences between the models:

RLCA : The fine model has an exact linear mapping to the coarse model.

RLCB : The fine model has an exact nonlinear mapping to the coarse model.

RLCC : The fine model has an inexact non-linear mapping (different topology) to the coarse model.

RLCD : The fine model has an inexact non-linear mapping to the coarse model.

The deviation of the fine model response to the specifications is $f : I\!R^3 \mapsto I\!R^{15}$, the problem is $\min \max_{j=1}^{15} \{f_j(x)\}$. The deviation of the coarse model to the specifications is defined equivalently.

QUAD

The QUAD problems, QUADA and QUADB, involve three quadratic functions. The fine model response is $f : I\!R^2 \mapsto I\!R^3$, $f = (f_1, f_2, f_3)^T$, where

$$\begin{aligned} f_1(x) &= 0.5\,x_1^2 + .1\,x_2^2 - 2\,x_2 - 2 \\ f_2(x) &= 0.2\,x_1^2 + 0.1\,x_2^2 + 2\,x_2 - 2 \\ f_3(x) &= 0.1\,x_1^2 - 3\,x_1 + 0.2\,x_2^2 - 2\,. \end{aligned}$$

The fine model is the same for both QUADA and QUADB.

The coarse model for QUADA is $c(z) = f(z + 0.1) + 0.1$ and the coarse model for QUADB is $c(z) = f(z - 0.1) + 0.1$.

The simple shift in the response functions causes that for neither problem the space mapping is perfect, $p(x^*) \neq z^*$.

INDOBS

The INDOBS problem concerns the design of an inductive obstacle in a parallel plate waveguide, the problem is described in [2]. There are two designable parameters. 11 frequencies are simulated per sweep. The problem consists in matching a given target response, which originates from the fine model.

The coarse model is the same as the fine model, except that the coarse model is simulated using a coarser discretization of the problem in the simulator.

The parameters of the problem are defined on a discrete grid. In the `opts` structure returned by `smprob`, the field `epsilon` contains the suggested minimum step length.

SRESFIL

The SRESFIL problem concerns design of a single resonator filter described in [3]. There are two designable parameters. 21 frequencies are simulated per sweep. The problem consists in matching a given target response, which originates from the fine model.

The coarse model is the same as the fine model, except that the coarse model is simulated using a coarser discretization of the problem in the simulator.

The parameters of the problem are defined on a discrete grid. In the `opts` structure returned by `smprob`, the field `epsilon` contains the suggested minimum step length.

HPLANEF

The HPLANEF problem concerns the design of a H-plane waveguide filter described in [6]. There are seven designable parameters. 11 frequencies are simulated per sweep. The problem consists in matching a given target response, which originates from the fine model.

The coarse model is the same as the fine model, except that the coarse model is simulated using a coarser discretization of the problem in the simulator.

The parameters of the problem are defined on a discrete grid. In the `opts` structure returned by `smprob`, the field `epsilon` contains the suggested minimum step length.

A.2 Space Mapping Optimization Algorithms

The toolbox contains five algorithms based on space mapping technique. Two algorithms, namely `smo` and `smon`, are related to the original space mapping formulation. Three algorithms, namely `smh`, `smho` and `smhc`, are so-called hybrid space mapping algorithms, combining space mapping technique with classical Taylor based optimization.

A.2.1 Interface

The algorithms have a common interface:

```
[xk, fk, Hfk, trace] =

      smx(H, fine, coarse, x0, A, b, eq, opts, P1, P2, ...)
```

where `smx` is one of the following

`smo`	original space mapping,
`smon`	new space mapping formulation,
`smh`	hybrid space mapping,
`smhc`	hybrid space mapping with response correction,
`smho`	hybrid space mapping with orthogonal steps.

The mandatory arguments of the algorithms are the merit function `H`, the file handles to the `fine` and the `coarse` model and a starting point `x0`. Any parameters that should be passed directly to the fine and the coarse model can be specified in the place of `P1, P2, ...`.

The algorithms return the best iterate `xk`, the fine model response `fk` at `xk`, and the merit `Hfk` of the response. A fourth output option is a `trace` structure which contains a trace of important values gathered in the iteration process.

The user may supply the algorithm with linear constraints $\mathtt{A} \cdot \mathtt{x} \leqslant \mathtt{b}$, where the first `eq` rows are equality constraints. If the problem is unconstrained empty

matrices may be passed. The toolbox provides no check for consistency of the constraints.

The constraints only apply to fine model parameters, e.g. in the trust region subproblems, hence the coarse model may be evaluated at any $z \in \mathbb{R}^n$. The only exception is in the initial phase of the algorithms, where the coarse model parameters are constrained in the search for the coarse model minimizer, z^*. This is needed because the first iterate is the coarse model minimizer, $x_0 = z^*$, i.e. the first point where the fine model is evaluated.

Specific options determining the behaviour of the optimization algorithms are passed in the structure `opts`. The default values for the structure `opts` are obtained from the function `smopts`. The structure contains all options used to determine the behaviour of the specific algorithms. If an empty matrix is passed instead of a structure, the default values are obtained from `smopts`.

The function `smopts` is called as follows

```
opts = smopts(key1, value1, key2, value2, ...)
opts = smopts(opts, key1, value1, key2, value2, ...)
```

See the source file for a more complete description of the options, than is presented in this manual. An existing structure with options can be passed as input to override default values. Individual default options can be overwritten by specifying new key-value pairs as input arguments.

We mentioned some of the more important options here:

Most of the optimization algorithms in this toolbox rely on trust region methodology to enforce convergence. Control of the trust region is determined by a number of options. The most important is `dx`, the initial trust region size, which is problem dependent. Refer to `smopts` for the other options related to the trust region handling.

The accuracy of the optimization result is determined by the option `epsilon`. The algorithms stop if the relative step length or trust region size becomes smaller than `epsilon`. Another option controlling when the algorithms stops is `kmax` which determines the maximum allowed number of fine model evaluations. So the algorithms are stopped if one of the following conditions are satisfied

$$\begin{aligned} k &\geqslant \texttt{kmax} \\ \|h\| &\leqslant \texttt{epsilon}(1 + \|x_k\|) \\ \mathrm{dx} &\leqslant \texttt{epsilon}(1 + \|x_k\|) \end{aligned}$$

where k counts fine model function evaluations, $\|h_k\|$ is the step length, dx is the size of the trust region and $\|x_k\|$ is the norm of the current iterate. The last criterion is there to avoid an unnecessary extra iteration, solving a trust region problem with a trust region size that is less than the minimum allowed step length.

A quick way to test-run the algorithms is through the `smrun` function, which is called by

```
[p, trace, opts] = smrun(num, algo, opts)
```

where `num` is the number of the test problem (see p. 185), and `algo` is the number of the algorithm to test:

1	SMO orginal space mapping formulation
2	SMON new space mapping formulation with mapped coarse model
3	SMH space mapping hybrid algorithm
4	SMHO space mapping hybrid algorithm with orthogonal steps
5	SMHC space mapping hybrid algorithm with response correction
6	direct optimization of the fine model, 1st order method
7	direct optimization of the fine model, 2nd order method
8	direct optimization of the coarse model, 2nd order method

The variables `p` and `opts` are the structures obtained from `smprob`, and `trace` is the trace of the optimization process obtained by calling one of the algorithms. See Section A.3 for examples showing the content of `trace` and `p`. For the choices 7 and 8 of `algo` there cannot be produced a trace variable.

Before we describe the algorithms we first give a brief theoretical overview of space mapping theory.

A.2.2 Theoretical Overview

The main problem consists in finding the minimizer x^* (assumed unique) of the fine model,

$$x^* = \arg\min_x H(f(x)) \,, \tag{A.1}$$

where $f : \mathbb{R}^n \mapsto \mathbb{R}^m$ is the vector response function representing the fine model, and $H : \mathbb{R}^m \mapsto \mathbb{R}$ is a convex merit function, usually a norm. We

denote x^* the *fine model minimizer*. We note that f is assumed so expensive that using a classical Taylor based optimization method is infeasible, so finding x^*, or an approximation to it, is nontrivial.

A related problem is finding the minimizer z^* (assumed unique) of the coarse model,

$$z^* = \arg\min_{z \in \mathbb{R}^n} H(c(z)) ,$$

with $c : \mathbb{R}^n \mapsto \mathbb{R}^m$ being the vector response function representing the coarse model. We denote z^* the *coarse model minimizer*. Since c is assumed cheap to evaluate the gradient is available (e.g. by finite difference approximation), hence finding z^* is a trivial problem for a classical Taylor based optimization method.

The space mapping $p : \mathbb{R}^n \mapsto \mathbb{R}^n$ linking the parameter space of the fine and the coarse model is usually defined as solving the so-called *parameter extraction problem*,

$$p(x) = \arg\min_{z \in \mathbb{R}^n} \|c(z) - f(x)\|_2 .$$

This definition of the space mapping may lead to nonuniqueness in the parameter extraction problem, so several alternative definitions are available in the toolbox:

Regularization with regard to the distance to z^*,

$$p_\lambda(x) = \arg\min_z \left\{ (1-\lambda) \|c(z) - f(x)\|_2^2 + \lambda \|z - z^*\|_2^2 \right\} , \tag{A.2}$$

for some value of $0 \leqslant \lambda < 1$.

Regularization with regard to the distance to x,

$$p_\lambda(x) = \arg\min_z \left\{ (1-\lambda) \|c(z) - f(x)\|_2^2 + \lambda \|z - x\|_2^2 \right\} , \tag{A.3}$$

for some value of $0 \leqslant \lambda < 1$.

Regularization using gradient information,

$$p_\lambda(x) = \arg\min_z \left\{ (1-\lambda) \|c(z) - f(x)\|_2^2 + \lambda \|c'(z) - f'(x)\|_F^2 \right\} , \tag{A.4}$$

for some value of $0 \leqslant \lambda < 1$. In the optimization algorithms $f'(x)^T$ is approximated by a secant approximation $D \in \mathbb{R}^{m \times n}$ during iterations, so this D is used instead of the true Jacobian matrix in (A.4).

In the implementation the above regularized problems are solved as normal nonlinear least-squares problems, exemplified here by (A.2),

$$p_\lambda(x) = \arg\min_z \left\| \begin{array}{c} \sqrt{(1-\lambda)}\,(c(z) - f(x)) \\ \sqrt{\lambda}\,(z - z^*) \end{array} \right\|_2^2 ,$$

for some value of $0 \leqslant \lambda < 1$. As the Jacobian of c is assumed available, the gradient for this least-squares objective function is available, at least for (A.2) and (A.3). In the case of (A.4) though, the gradient of the least-squares objective function depends on the second derivatives of c. So, as second order information is not available, the gradient of the least-squares objective function is found by finite difference approximation.

With this theoretical introduction we are now in a position to introduce the algorithms.

The Original Space Mapping Formulation

The original space mapping technique involves solving the nonlinear equations

$$p(x) = z^* ,$$

for $x \in \mathbb{R}^n$. The algorithm implemented in the toolbox function `smo` addresses this problem, by solving the least-squares formulation of the problem,

$$\min_{x \in \mathbb{R}^n} \|p(x) - z^*\|_2 . \tag{A.5}$$

Another space mapping technique, equivalent with the original formulation in some ways, is to minimize the mapped coarse model,

$$\min_{x \in \mathbb{R}^n} H(c(p(x))) . \tag{A.6}$$

The algorithm implemented in the toolbox function `smon` solves this problem.

The solutions of (A.5) and (A.6) are not necessarily the solution x^* of the main problem (A.1). In fact we can only be certain that the solution is x^* if the space mapping is perfect, $p(x^*) = z^*$.

In the description of the test problems above it is stated which of the test problems that have a perfect mapping. Due to this drawback, the results of the functions `smo` and `smon` are not directly comparable with the functions implementing the hybrid space mapping framework, described next.

Hybrid Space Mapping Algorithms

The toolbox contains three functions, namely `smh`, `smho` and `smhc`, implementing the hybrid space mapping framework described in [4]. Basically the algorithms rely on a model of the form

$$s(x) = w\, c(p(x)) + (1 - w)\, l(x)$$

where $0 \leqslant w \leqslant 1$ is a transition parameter, $c \circ p$ is a mapped coarse model and l is linear Taylor model of the fine model. An exception is the algorithm in `smhc` which uses a form of the mapped coarse model where the responses are corrected to match the fine model using a secant method.

All the algorithms start with $w = 1$ and end with $w = 0$, provided enough iterations. Thereby a switch from the mapped coarse model to the linear Taylor model takes place.

With k being the iteration counter, it is proven in [4] that the main condition for convergence of this class of algorithms is that

$$w_k = \mathrm{dx}_k \cdot o(1)$$

where dx_k is the size of the trust region and $o(1) \to 0$ for $k \to \infty$.

Two of the algorithms, namely `SMH` and `SMHC`, use a gradual switching strategy, whereas the third algorithm `SMHO` switches abruptly from $w = 1$ to $w = 0$ at a certain point in the iteration process.

The algorithms use linear Taylor model with secant approximations to the derivatives for both the space mapping p and l. So the last stage of the three algorithms involves sequential linear programming, where the linear model has inexact derivatives. To help speed up the convergence, the options `dofinitediff` and `maxuphill` (refer to the source of `smopts`) can force the algorithms to correct the linear model by a finite difference approximation. Further we should note that the option `initd` controls the way that the initial approximation to the derivatives of the fine model is obtained.

A.2.3 The Optimization Algorithms

SMO

The function `smo` implements the original space mapping technique solving the problem in (A.5) using a trust region secant method. The secant method

involves a linear Taylor model of the space mapping with a secant approximation to the Jacobian matrix.

SMON

The function `smon` implements the alternative space mapping technique solving the problem (A.6), using a trust region method with sequential linear approximations to p by a secant method.

SMH

The function `smh` implements the hybrid space mapping algorithm, with a gradual switching between the mapped coarse model and the linear Taylor model of the fine model.

The control of w is determined by the options `w_min`, `w_reduce` and `max_w_not_reduced`. If either a proposed step is not accepted or if the number of iterations where w has not been changed reaches $\max\{n, \texttt{max\_not\_reduced}\}$ then w is updated. The updating formula is

$$w_{k+1} = w_k \cdot \texttt{w\_reduce} \cdot \min\{\mathrm{dx}_{k+1}, 1\},$$

where dx is the size of the trust region. If w by updating gets below `w_min` then w is set to zero.

SMHO

The function `smho` implements a hybrid space mapping algorithm with orthogonal updating steps of the space mapping approximation.

If the space mapping fails within the first n iterations the algorithm evaluates the fine model at a step in a direction orthogonal to previous steps, this is in order to improve the quality of the space mapping secant approximation. Which of the orthogonal directions that is chosen and the length of the step in that direction can be controlled by the options `ortho_met`, `ortho_scale_type` and `ortho_scale`. If a single orthogonal step is not sufficient, further steps are taken, until the fine model has been evaluated at most n times. Thereafter the algorithm switches to a linear Taylor model of the fine model.

If the space mapping steps are successful the algorithm keeps taking space mapping steps until at most `max_w_not_reduced`$+n$ steps have been taken. Thereafter the algorithm is forced to switch to the linear Taylor model of the fine model.

SMHC

The function `smhc` implements a hybrid space mapping algorithm with response correction of the mapped coarse model.

The combined model for this algorithm is

$$s_k(x) = w_k(g_k .* [c(p_k(x)) - c(p(x_k))] + f(x_k)) + (1 - w_k)l_k(x)$$

where $g \in \mathbb{R}^m$ are the correction factors and $*$ is element-wise multiplication.

The correction factors are found by the secant update

$$g_{k+1}^{(j)} = \frac{f^{(j)}(x_{k+1}) - f^{(j)}(x_k)}{c^{(j)}(p(x_{k+1})) - c^{(j)}(p(x_k))}\,, \quad j = 1, \ldots, m\,,$$

where the superscript (j) indicates the jth element of the vector.

The transition parameter w is controlled as in the `SMH` algorithm described above.

A.2.4 Auxiliary Functions

Both the main directory and the directory `private` contains a number of auxiliary functions. We briefly introduce the most important ones.

`parameter_extraction` For a given x the function solves the parameter extraction problem, determining $p(x)$. The user can choose between four different space mapping definitions through the option `petype`. The starting point for the parameter extraction problem is specified by the option `pestart`, four possibilities exist. Further, for the regularization formulations, the value of λ can be specified by the option `lambda`.

`combined_model` For a given x the function calculates the response of the combined model $s_k(x) = wc(p_k(x)) + (1 - w)l_k(x)$, where $c(p_k(x))$ is the mapped coarse model (see `mapped_model` below) and $l_k(x) = D(x - x_k) + f(x_k)$ is a linear Taylor model of the fine model.

`combined_corrected_model` For a given x the function calculates the response of the combined model $s_k(x) = w(g .* (c(p_k(x)) - c(p(x_k))) + f(x_k)) + (1-w)l_k(x)$, where the first part is the response corrected model, with $c(p_k(x))$ being the mapped coarse model (see `mapped_model` below), and $l_k(x) = D(x - x_k) + f(x_k)$ is a linear Taylor model of the fine model.

`mapped_model` For a given x the function calculates the response of the mapped coarse model $c(p_k(x))$, where $p_k(x) = B(x - x_k) + p(x_k)$ is a linear Taylor model of the space mapping.

`smtrlinear` For a given x the function finds a minimizer of a given linear model subject to linear trust region constraints (infinity norm trust region) and, if any, user provided linear constraints. Formally the problem solved is

$$\begin{array}{ll} \min\limits_x H(l_k(x)) & \\ \text{s.t.} \quad \|x - x_k\|_\infty & \leqslant \ \mathrm{dx}_k \\ \qquad\quad Ax & \leqslant \ b \end{array}$$

where the first `eq` rows of the user constraints are equality constraints.

`smdirect` Direct, classical Taylor based optimization. Solves the problems of the general type

$$\begin{array}{ll} \min\limits_x H(s(x)) & \\ \text{s.t.} \quad Ax \leqslant b & \end{array} \tag{A.7}$$

where s is a nonlinear vector response function. The first `eq` rows of the linear constraints are equality constraints. Exact gradient information is assumed available.

`direct` Direct, classical Taylor based optimization with inexact gradient information. Solves (A.7) using a trust region algorithm with secant gradient approximations.

A.3 Examples

We now give two small examples to show how the toolbox can be used.

A.3.1 Quick Run

Assume that we want to run the TLT2 problem with the hybrid space mapping algorithm `smh`. The easiest way to do this is by calling `smrun`,

```
[p, trace] = smrun(1, 3);
```

(the semicolon suppresses the display of the output variables). After the algorithm has finished the iteration process we now have a trace variable with the results (the contents of the variable `p` is shown in the next example). The `trace` variable is a Matlab structure. We list the contents from this run:

```
trace =
      k: [1x66 double]
      x: [2x66 double]
      z: [2x66 double]
      f: [11x66 double]
      h: [1x66 double]
     dx: [1x66 double]
      w: [1x66 double]
    rho: [1x65 double]
    obj: [1x66 double]
```

We see that there are nine fields containing the trace of variables in the 66 steps taken by the algorithm. For example the field `trace.f` contains all fine model responses evaluated by the algorithm. In the field `trace.x` are the corresponding fine model parameters. The field `trace.z` contains the space mapped parameters, $z = p(x)$.

Now we could for example check how close the best fine model response found by the algorithm is to the optimal response of the fine model. First the best objective function value found by the algorithm:

```
>> min(max(trace.f))
ans =
         0.455324591088871
```

Then the objective function value of the optimal response:

```
>> max(p.fast)
ans =
          0.45532459108887
```

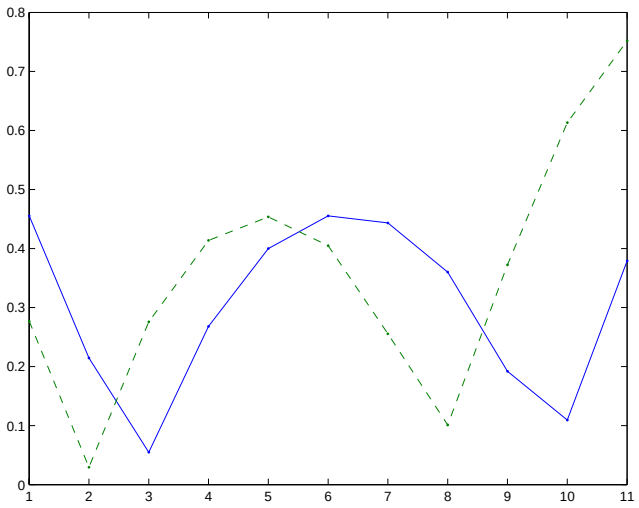
We see that the solutions only differ in the 15th decimal. In fact the difference is around full calculating accuracy.

If we want to plot the initial response (i.e. the response in the starting point $x_0 = z^*$) and the best response found by the algorithm, we first obtain the index of the best response:

```
>> [fmin idx] = min(max(trace.f))
fmin =
         0.455324591088871
idx =
    63
```

Then we plot the responses:

```
>> plot(1:p.m, trace.f(:,idx), '.-', ...
        1:p.m, trace.f(:, 1), '.--')
```



We see that the first response violated the specifications, since the maximum value of the response is above 0.5 (refer to the problem description above).

A.3.2 Examining a Problem

Now let us examine the structure with the problem returned from `smprob`.

```
>> p = smprob(1)
p =
      xast: [2x1 double]
      zast: [2x1 double]
        x0: [2x1 double]
      fine: @tlt2f
    coarse: @tlt2c
         n: 2
     fopts: [1x1 struct]
         m: 11
      fast: [11x1 double]
         A: [4x2 double]
         b: [4x1 double]
        eq: 0
         H: 'minimax'
```

We see that the problem is a minimax problem (`H = 'minimax'`). Further we see that it is a two-dimensional problem ($n = 2$ and `xast` $= x^* \in \mathbb{R}^2$). There are 11 response functions ($m = 11$ and `fast` $= f(x^*) \in \mathbb{R}^{11}$). The fine model handle refers to the function `tlt2f`.

References

[1] M.H. Bakr, J.W. Bandler, K. Madsen, J. Søndergaard, *An Introduction to the Space Mapping Technique*, Optimization and Engineering, vol. 2, no. 4, pp. 369–384, 2001.

[2] M.H. Bakr, W.J.R. Hoefer, *Electromagnetic Synthesis: A Novel Concept in Microwave Design*, 31st European Microwave Conference Dig., London, England, 2001.

[3] M.H. Bakr, W.J.R. Hoefer, *Electromagnetic Synthesis: A Novel Concept in Microwave Design*, IEEE Trans. Microwave Theory Tech., 2003.

[4] K. Madsen, J. Søndergaard, *Convergence of Hybrid Space Mapping Algorithms*, submitted, Optimization and Engineering, 2003.

[5] Matlab^{TM} version 6.5 and Matlab^{TM} Optimization Toolbox version 2.2., The MathWorks, Inc., 3 Apple Hill Drive, Natick MA 01760-2098, 2003.

[6] G. Matthaei, L. Young, E.M.T. Jones, *Microwave Filters, Impedance-Matching Networks, and Coupling Structures*, Norwood, MA, Artech House, p. 545, 1980.

[7] J. Søndergaard, *Optimization Using Surrogate Models — by the Space Mapping Technique*, Ph.D. Thesis, IMM, DTU, Lyngby, 2003.