NON-LINEAR OPTIMIZATION USING SPACE MAPPING

Jacob Søndergaard

LYNGBY 1999
IMM-EKS-1999-23

IMM

Preface

This project is a master thesis accounting for 40 points of the 300 points required for obtaining the engineering master degree at the Technical University of Denmark (DTU). The work has been carried out at Department of Mathematical Modelling (IMM), DTU, in the period 8th of February to 8th of October 1999.

During this project I have visited McMaster University in Ontario, Canada. I would like to thank all the people at S.O.S. lab, especially Mohamed Heaba Bakr and John W. Bandler, for fruitfull discussions and cooperation.

I would also like to thank my supervisors, Poul Erik Frandsen (Ticra), Hans Bruun Nielsen (IMM) and Kaj Madsen (IMM), the main supervisor, for using their competence in engaged supervision of the project. Also Henrik Bredmose is thanked for proof reading the report, and for many usefull discussions.

Lyngby, the 8th of October 1999.

Jacob Søndergaard

Abstract

In this project the optimization method named Space Mapping is presented, implemented and tested. The project presents novel theoretical additions to the conventional Space Mapping theory, that links the Space Mapping framework to surrogate and direct optimization theory. The Space Mapping theory defines a framework for optimization of an accurate, expensive non-linear model by utilizing a less accurate, cheaper model, through a parameter mapping. The implementation is based on a traditional gradient based trust region method, used in sequential optimization of surrogates aproximating the expensive model.

Three numerical tests show how the novel theoretical additions greatly enhance the Space Mapping method when considering more general problems.

Keywords: non-linear optimization, Space Mapping, surrogate modelling, trust region methodology.

Contents

Chapter 1

Introduction

This project presents the Space Mapping method for optimization of expensive non-linear models. The Space Mapping method has been treated in previous works, for instance [2, 3, 4, 5, 6, 7, 8, 9]. The theory of Space Mapping belongs to the general surrogate optimization theory. Other work on surrogate optimization are e.g. [1, 10, 11, 18].

The Space Mapping theory provides a mathematical framework which forms a natural way of incorporating traditional engineering practices, as for instance rules of thumb and other best practices, in the work with far more advanced and expensive models.

The Space Mapping method requires two models provided by the user:

- The model on which optimization is to be performed, typically an accurate, expensive and non-linear model, denoted the fine model.
- A model approximating the same system as the expensive model, typically a simpler, cheaper and less accurate model, and presumably non-linear as well, denoted the coarse model.

In previously described surrogate optimization methods, as for instance [10, 18], the coarse model is used as a surrogate for the fine model. Hence the optimization is performed on the coarse model, and only a few well targeted evaluations of the fine model are needed. Unlike the previous surrogate optimization methods, the Space Mapping method constructs a parameter mapping between the models, and thereby enhancing the usability of the

coarse model as an approximation to the fine model. The parameter mapping roughly consists in fitting the coarse model to the fine model evaluated in a given set of fine model parameters. In this project we are using local linear approximations to this mapping, and thereby generating a sequence of approximations to the fine model, using the coarse model through the parameter mapping.

This project reveals that the conventional formulation of the Space Mapping method, see for instance [2, 3], is insufficient if the coarse model lacks certain properties in describing the fine model. Due to this, we present a novel combined model, which links the Space Mapping method with direct, trust region, optimization methods, as they for instance are described in [17]. With this combined model, we use any new fine model evaluation performed, to update a local linear approximation to the fine model.

The novel formulation of the Space Mapping method directs most of the optimization computational effort towards the combined model, while maintaining the accuracy of the fine model. Hence the overall computational effort needed is smaller than that needed for direct optimization.

The structure of this report is as follows: Chapter 2 presents the conventional Space Mapping theory, and the thoughts behind the new combined model. Chapter 3 discusses various aspects of implementing the conventional and the novel Space Mapping formulation in an algorithmic form. Chapter 4 presents test results. Finally, chapter 5 presents the conclusion and ideas for future work.

Chapter 2

Space Mapping theory

In this chapter the conventional Space Mapping theory is described in detail, and the novel approach is presented.

2.1 Introducing notation

When modelling a physical system we discern between two situations:

1. Design of a new system by modelling the expected behaviour, where the final design has to meet some predefined specifications, and
2. modelling an existing system, where the model has to fit some measurements of the physical system.

With this in mind, let a finite set of either specifications or measurements of a physical system be represented by the set of points $\{t^{(i)}, y^{(i)}\}$, $i = 1, \ldots, m$, in vector notation: $\{\mathbf{t}, \mathbf{y}\}$.

2.1.1 Models

Consider a mathematical model $\mathbf{g}(\mathbf{t}; \mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^m$, $m > n$, which should approximate a given set of points $\{\mathbf{t}, \mathbf{y}\}$, representing some specifications or measurements of the system considered, in an optimal way. We define the residual vector as $\mathbf{f} = \left[f^{(1)}, \ldots, f^{(m)}\right]^T$, consisting of the residuals

$$f^{(i)}(\mathbf{x}) = g(t^{(i)}; \mathbf{x}) - y^{(i)} \quad \text{for } i = 1, \ldots, m\,, \tag{2.1}$$

hence $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$.

We wish to find a set of parameters $\mathbf{x}^* \in \mathbb{R}^n$ which makes the model $\mathbf{g}$ fit the points $\{\mathbf{t}, \mathbf{y}\}$, such that a given merit function H, usually a norm in $\mathbb{R}^m$, of the residual vector, $H(\mathbf{f}(\mathbf{x}))$, is minimized.

The choice of merit function $H : \mathbb{R}^m \to \mathbb{R}$ is arbitrary, as long as it is continuous and satisfies the requirements:

$$H(\mathbf{0}) < H(\mathbf{q}) \; \forall \, \mathbf{q} \in \mathbb{R}^m \backslash \{\mathbf{0}\}\,, \tag{2.2}$$

$$\lim_{\mathbf{q} \to \mathbf{0}} H(\mathbf{q}) = H(\mathbf{0})\,, \tag{2.3}$$

i.e. having the lowest value for the vector of zeros, and approaching the lowest value as the parameters approach the vector of zeros.

The parameters $\mathbf{x}$ might be constrained to some confined region of $\mathbb{R}^n$, due to either limitations in the model or the system considered. However, to keep the formulation simple, we will only consider the unconstrained case, i.e. $\mathbf{x} \in \mathbb{R}^n$.

Defining the objective function $F(\mathbf{x}) = H(\mathbf{f}(\mathbf{x}))$, we can write the problem of minimizing the residuals as,

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \; F(\mathbf{x})\,. \tag{2.4}$$

This problem, of making a mathematical model approximate some specifications or measurements of a physical system, is the main problem addressed in this project.

Of the general problem presented in (2.4), we will consider only the special case where the model $\mathbf{g}$, which we denote the *fine model*, is very expensive to evaluate. For instance $\mathbf{g}$ is so computationally intensive that evaluation times rather than the optimization algorithms is the bottleneck, limiting the practical use of the model. We consider the fine model as capable of accurately emulating the behaviour of the physical system, and it may even include computational utilization of actual hardware measurements.

Though this project will only consider numerical models. Note that the resulting expensive objective function F makes the problem (2.4) difficult and impractical to solve.

Coarse model

Due to the difficulties associated with the expensiveness of the fine model, we introduce a faster and cheaper, but less accurate, model $\mathbf{d}(\mathbf{t};\mathbf{z})$ of the same system, which we denote the *coarse model.* Such a model may be a good empirical model. However, the parameter values might be outside their recommended ranges. The coarse model $\mathbf{d}$ is used to gain information about the fine model $\mathbf{g}$, by obtaining a correspondence between the parameters $\mathbf{x}$ and $\mathbf{z}$. Before dealing with this, we have a few more prefatory remarks about the models.

As the fine model, the coarse model should approximate a given set of points $\{\mathbf{t},\mathbf{y}\}$. Hence we wish to find the set of parameters $\mathbf{z}^* \in \mathbb{R}^n$ which minimize the residuals

$$\mathbf{c}(\mathbf{z}) = \left[d(t^{(1)};\mathbf{z}) - y^{(1)}, \dots, d(t^{(m)};\mathbf{z}) - y^{(m)}\right]^T , \tag{2.5}$$

for some merit function H of the residuals, $H(\mathbf{c}(\mathbf{z}))$. The merit function should be the same as the one used in conjunction with the fine model residuals. Defining the objective function $C(\mathbf{z}) = H(\mathbf{c}(\mathbf{z}))$ we can write the problem of minimizing the residual as

$$\mathbf{z}^* = \underset{\mathbf{z}}{\operatorname{argmin}}\, C(\mathbf{z}) . \tag{2.6}$$

The process of finding $\mathbf{z}^*$ is considered unambiguous, which implies that $\mathbf{z}^*$ is a unique, global minimum of C.

Due to the cheap objective function C, the problem (2.6) is much easier solved than the problem (2.4) related to the fine model. Notice that the solution $\mathbf{z}^*$ of (2.6) might be very different from the solution $\mathbf{x}^*$ to (2.4), due to the differences between the fine and the coarse models.

From this place we will interchangeably use the short notations $\mathbf{g}(\mathbf{x})$ and $\mathbf{d}(\mathbf{z})$, with the longer notations $\mathbf{g}(\mathbf{t};\mathbf{x})$ and $\mathbf{d}(\mathbf{t};\mathbf{z})$.

Smoothness

We require that the vector functions $\mathbf{f}$ and $\mathbf{c}$ are continuously differentiable, i.e. the functions being differentiable and their derivatives being continuous. This implies that both functions are so *smooth* that we can express them by a first-order Taylor expansion. Hence, illustrated by $\mathbf{f}$,

$$\mathbf{f}(\mathbf{x}+\mathbf{h}) = \mathbf{f}(\mathbf{x}_k) + \mathbf{f}'(\mathbf{x}_k)\mathbf{h} + \mathbf{o}(\|\mathbf{h}\|) , \tag{2.7}$$

where $\|\cdot\|$ indicates some norm in $\mathbb{R}^n$ and where $\mathbf{f}'(\mathbf{x}_k)$ is the Jacobian in $\mathbf{x}_k$,

$$\mathbf{f}'(\mathbf{x}_k) \equiv \left[\frac{\partial f^{(i)}}{\partial x^{(j)}}\right] , \quad i = 1,\dots,m, \quad j = 1,\dots,n .$$

Due to the smoothness assumption we can define a *linearization* $\mathbf{f}_k$ of the vector function $\mathbf{f}$ around a set of parameters $\mathbf{x}_k$,

$$\mathbf{f}_k(\mathbf{x}) \equiv \mathbf{f}(\mathbf{x}_k) + \mathbf{f}'(\mathbf{x}_k)(\mathbf{x}-\mathbf{x}_k) . \tag{2.8}$$

Notice that the evaluation of the Jacobian is not considered implemented as a part of the fine model, and therefore it is not available for use during optimization. The Jacobian of the coarse model is assumed available as a part of the model evaluation, at least as a finite difference approximation or automatic differentiation.

Response

For the convenience of complying with earlier works on Space Mapping theory, we name the vector resulting from an evaluation of a vector function a *response.* Hence an evaluation of the fine model $\mathbf{g}$ for an arbitrary set of parameters $\mathbf{x}$ results in a fine model response. This concept is illustrated in figure 2.1 with an example of a fine model response.

2.1.2 The mapping function

As the coarse and the fine model approximate the same physical system, we may imply some similarity between the models. This naturally leads to the idea of using the coarse model to gain information about the fine model. We

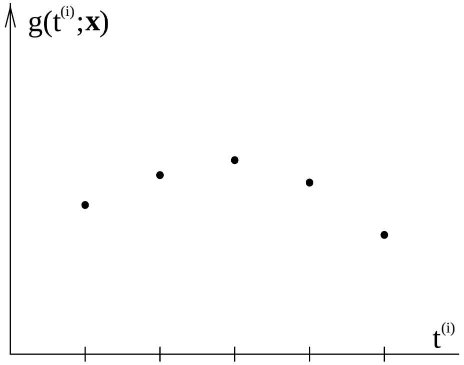

Figure 2.1: *An example fine model response of size $m = 5$. A response is the vector resulting from a vector function evaluation for a given set of parameters.*

do this by establishing a functional mapping $\mathbf{p}$ between the two parameter spaces. The mapping function $\mathbf{p}$ relates a set $\mathbf{x}$ of fine parameters with the set $\mathbf{z}$ of coarse parameters which yield the best similarity between $\mathbf{g}(\mathbf{x})$, the fine model response, and $\mathbf{d}(\mathbf{z})$, the coarse model response. Hence the functional relation between the parameter spaces is $\mathbf{z} = \mathbf{p}(\mathbf{x})$. We measure the similarity between two responses by the vector of residual errors

$$\begin{aligned}\mathbf{r}(\mathbf{x},\mathbf{z}) &= \mathbf{g}(\mathbf{x}) - \mathbf{d}(\mathbf{z}) \\ &= \left[g^{(1)}(\mathbf{x}) - d^{(1)}(\mathbf{z}), \ldots, g^{(m)}(\mathbf{x}) - d^{(m)}(\mathbf{z})\right]^T ,\end{aligned} \tag{2.9}$$

for given sets of coarse and fine model parameters. The concept of the residual error is illustrated in figure 2.2, where the components of the residual vector corresponds to the vertical distances between the fine and the coarse model responses.

For a fixed set of fine parameters $\mathbf{x}$, the set of coarse parameters $\mathbf{z}$ which yield the smallest value of some merit function H of the residual errors, $H(\mathbf{r}(\mathbf{x},\mathbf{z}))$, will be termed the mapped set of coarse parameters. We denote the process of minimizing $H(\mathbf{r}(\mathbf{x},\mathbf{z}))$ evaluation of the *mapping function*

$$\mathbf{p}(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{z}} H(\mathbf{r}(\mathbf{x},\mathbf{z})) . \tag{2.10}$$

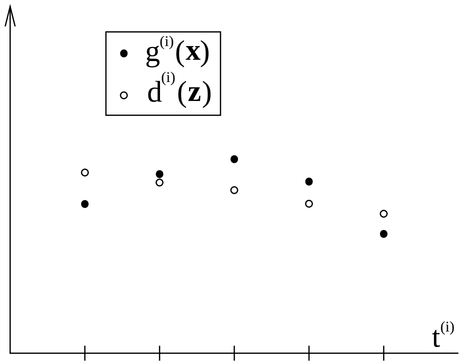

Figure 2.2: *Plot of example fine and coarse model responses ($m = 5$). The components of the residual vector, $\mathbf{r}(\mathbf{x},\mathbf{z}) = \mathbf{g}(\mathbf{x}) - \mathbf{d}(\mathbf{z})$, are the vertical distances between each corresponding set of components in the model responses.*

Hence the mapping function relates the fine and the coarse model space, in such way that a set of fine parameters $\mathbf{x}$ is mapped to the set of coarse parameters $\mathbf{z}$, which yields the best possible fit of a coarse model response to the actual fine model response.

The mapping is assumed *unique*, by which we comprehend the mapping function being injective, at least within some confined regions of interest within the coarse and the fine model space. Additionally we assume that the mapping is smooth, at least locally.

The merit function can be chosen arbitrarily, i.e. it does not have to be the same as the one used with the residuals between respectively the fine or coarse model and the data, as long as it is continuous and the requirements in (2.2) and (2.3) are satisfied. Observe that choosing a non-differentiable merit function H, e.g. the infinity norm, might make the mapping function non-differentiable at some points in the parameter spaces. In chapter 4 some of the effects of having non-differentiable points in the mapping function will be examined.

The principle of the mapping idea is sketched in figure 2.3. It is shown how a given set of parameters $\mathbf{x}$, in the fine model space, is mapped one-to-one with the corresponding set of parameters $\mathbf{z}$, in the coarse model space. Notice, that each time the mapping function is evaluated, one fine model evaluation has to be performed. Hence the mapping function is at least as expensive to evaluate as the fine model.

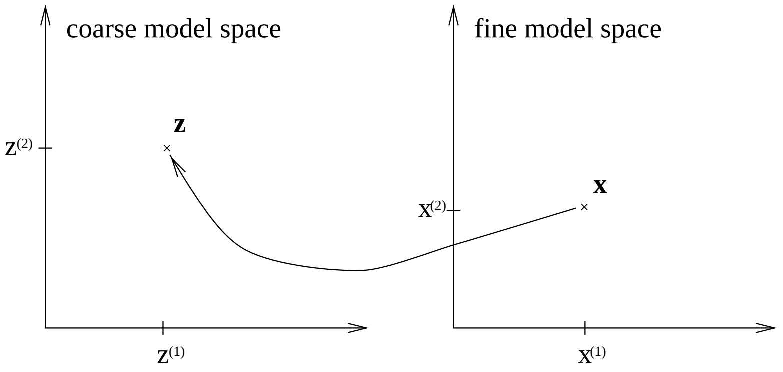

Figure 2.3: *The figure is showing an example mapping in two dimensional space ($n = 2$) between a set of fine parameters $\mathbf{x} = [x^{(1)}, x^{(2)}]^T$ and a set of coarse parameters $\mathbf{z} = [z^{(1)}, z^{(2)}]^T$. The parameters are related through the mapping function, $\mathbf{z} = \mathbf{p}(\mathbf{x})$, defined in* (2.10).

From the definition of the mapping function it follows that

$$\mathbf{g}(\mathbf{x}) \simeq \mathbf{d}(\mathbf{p}(\mathbf{x}))\,, \tag{2.11}$$

i.e. the coarse model with the mapped parameters approximates the fine model. We define the *perfect mapping* as case where the optimal set of coarse parameters $\mathbf{z} = \mathbf{z}^*$ minimizes $H(\mathbf{d}(\mathbf{z}) - \mathbf{g}(\mathbf{x}^*))$, which, according to (2.10), implies that $\mathbf{z}^* = \mathbf{p}(\mathbf{x}^*)$. According to (2.2) a special case of the perfect mapping is when $\mathbf{g}(\mathbf{x}^*) = \mathbf{d}(\mathbf{z}^*)$, hence when the models match perfectly at their optima. We use the term *unit mapping* if $\mathbf{p}(\mathbf{x}) = \mathbf{x}\ \forall\,\mathbf{x}$, i.e. the mapped coarse parameters are equal to the fine model parameters. We denote $\mathbf{d}(\mathbf{p}(\mathbf{x}))$ the *mapped coarse model.*

Parameter extraction

In earlier works the evaluation of the mapping function (2.10) is referred to as the process of parameter extraction. This name refers to the property that the process of minimizing the residual error is consistent with

extracting parameters from a data fitting problem. Hence solving (2.10) corresponds to fit the coarse model to a given response of the fine model, thereby extracting the set of coarse model parameters, which makes the coarse model response fit the given fine model response.

Ensuring a unique mapping

As the mapping function is formulated in (2.10) one fine function evaluation is used each time a parameter set is mapped between the fine and the coarse model space. If there exist several solutions to (2.10), such that the mapping is not unique, as presupposed, then further information has to be obtained about the fine model. We will leave this problem of ensuring the uniqueness of the mapping function to a later study. However we will give a very brief presentation of three approaches below.

Multi point method. One method for ensuring the uniqueness of the mapping function, is to sample one or more points in the vicinity of the current set of fine parameters and using the information obtained to make a multi point evaluation of the parameter extraction. The problem is how to select the points to include. A method using a variable number of points is described in [4]. The method aims at choosing just enough points, in the right directions from the current point, to ensure a unique mapping.

Known relation. If the relation between one or more parameters is know in advance, these parameters do not have to be mapped. If previous experiments have shown a simple relation, e.g. linear, between parts of the parameter spaces, then this knowledge could be utilized to avoid non-uniqueness in the mapping function for these parts of the parameter spaces. For instance if $z^{(1)} \simeq \alpha x^{(1)} + \beta$ for $x^{(1)}$ in some subset of $\mathbb{R}$, and for known values of α and β, then the problem (2.10) is reduced by one dimension. Hence resulting in a problem of finding the $n-1$ unknown, which minimizes the merit function of m residuals, when considering parameters in the subset of $\mathbb{R}$.

Double sided parameter extraction. Another approach to ensuring the uniqueness of the mapping function, is to have the independent parameters $\mathbf{t}$ mapped along with the dependent parameters $\mathbf{x}$. This approach is for instance described in [8].

2.2 Conventional Space Mapping

In this project we term the Space Mapping methods described in earlier works, see e.g. [2, 3, 6, 7], the conventional Space Mapping. In this section we present the conventional Space Mapping approach using a novel, concise formulation.

2.2.1 Novel formulation

Considering the relation presented in (2.11), we observe that $\mathbf{d}(\mathbf{p}(\mathbf{x}))$ may act as a surrogate for $\mathbf{g}(\mathbf{x})$ the fine model. This leads to the formulation of the conventional Space Mapping problem

$$\mathbf{x}^*_{SM} = \underset{\mathbf{x}}{\operatorname{argmin}}\; C(\mathbf{p}(\mathbf{x})), \tag{2.12}$$

where $\mathbf{x}^*_{SM}$ is termed the Space Mapping solution.

Since $\mathbf{z} = \mathbf{z}^*$ minimizes $C(\mathbf{z})$, the above problem corresponds to finding the set of parameters $\mathbf{x}^*_{SM}$, which is mapped to the optimal set of coarse model parameters, $\mathbf{p}(\mathbf{x}^*_{SM}) = \mathbf{z}^*$. Hence, the problem (2.12) consists of solving the set of non-linear equations $\mathbf{p}(\mathbf{x}) = \mathbf{z}^*$ for $\mathbf{x}$. Especially if the mapping is perfect, i.e. $\mathbf{p}(\mathbf{x}^*) = \mathbf{z}^*$, we have $\mathbf{x}^*_{SM} = \mathbf{x}^*$, the Space Mapping solution is a solution to the main problem defined in (2.4).

If there exists a set of coarse parameters $\hat{\mathbf{z}} \in \mathbb{R}^n$ which fulfills

$$H(\mathbf{r}(\mathbf{x}^*, \hat{\mathbf{z}})) < H(\mathbf{r}(\mathbf{x}^*, \mathbf{z}^*)), \tag{2.13}$$

it follows that $\mathbf{z}^* \neq \mathbf{p}(\mathbf{x}^*)$, i.e. the mapping is imperfect. For the imperfect mapping the Space Mapping solution will not be the same as the optimal solution $\mathbf{x}^*$ to the main problem (2.4), $\mathbf{x}^*_{SM} \neq \mathbf{x}^*$. Hence in general, the conventional Space Mapping method (2.12) yields no guarantee of convergence towards $\mathbf{x}^*$. The case of an imperfect mapping is addressed in section 2.3, where the conventional Space Mapping problem (2.12) is extended to work with both perfect and imperfect mapping, until then we only consider the case of a perfect mapping.

2.2.2 Previous formulations

The first presentations, see e.g. [7], of the Space Mapping method were formulated with a mapping function made up of weighted predefined fundamental functions. The Space Mapping problem was formulated as sequential solving of an over-determined system of linear equations, for the least squares solution, for each evaluation of the mapping. Later the idea of fundamental functions was left behind for a new approach given the name Aggressive Space Mapping. In [2, 3, 6] the Space Mapping problem was presented as the Aggressive Space Mapping problem

$$\mathbf{x}^*_{SM} = \underset{\mathbf{x}}{\operatorname{argmin}}\; \|\mathbf{p}(\mathbf{x}) - \mathbf{z}^*\|\,, \tag{2.14}$$

i.e. finding the solution $\mathbf{x} = \mathbf{x}^*_{SM}$ to the system of n non-linear equations $\mathbf{p}(\mathbf{x}) = \mathbf{z}^*$ for some norm in $\mathbb{R}^n$.

Observe that the problem (2.14) is identical to (2.12), since both problems consist in solving the system of non-linear equations $\mathbf{p}(\mathbf{x}) = \mathbf{z}^*$ for $\mathbf{x}$.

We will now look at various methods of solving the conventional Space Mapping problem as it is formulated in (2.12).

2.2.3 Solving the conventional Space Mapping problem

Since in general the conventional Space Mapping problem (2.12) is a nonlinear problem, we have to rely on iterative methods to find a solution. In this section we will exploit different approaches to solve the problem, still considering the case of perfect mapping.

The Jacobian of the mapping function (2.10) is not given as a part of the function evaluation. But the assumption of a smooth mapping function makes it obvious to somehow approximate the Jacobian, such that a gradient based method can be used.

Of course it is possible to solve the Space Mapping problem (2.12) without approximating the Jacobian of the mapping function. We term such an approach derivative free optimization. The derivative free optimization methods do, in most implementations, rely on extensive evaluation of the objective function. This statement applies to most of the direct search methods. Although methods aiming at reducing the number of required function evaluations by using local or global model surrogates and specialized pattern search algorithms, see e.g. [10], have been formulated, the general idea of avoiding excessive evaluation of the fine model is not favoured by such methods.

Maintaining the idea of in some way approximate the Jacobian of the mapping function, we could employ a gradient based method exploiting finite difference approximations in each iteration step. This would require $n+1$ evaluations of the mapping function, and thereby the fine model, per iteration step. When situated far away from the solution, the use of accurate derivative information would be unnecessary when taking the costly evaluations of the fine model into account. On the other hand, knowledge of derivative information close to the solution might result in a higher rate of convergence compared to the derivative free optimization methods. But the finite difference approximation is very impractical compared to the advantages of local models approximating the mapping function as we will discuss below.

Local models

The presupposed smoothness of the mapping has been utilized in earlier works [2, 3, 6] to make a linear approximation of the Jacobian in each iteration. We will follow the same approach, making sequential linear approximations $\mathbf{p}_k$ of the mapping function around the current set of parameters $\mathbf{x}_k$. Hence

$$\mathbf{p}_k(\mathbf{x}) = \mathbf{z}_k + \mathbf{B}_k(\mathbf{x} - \mathbf{x}_k)\,, \tag{2.15}$$

where $\mathbf{B}_k$ is a Broyden rank one (see e.g. [12]) approximation to the Jacobian of the mapping function. For $\mathbf{B}_k$ we use the updating formula

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{\mathbf{z}_{k+1} - \mathbf{z}_k - \mathbf{B}_k\mathbf{h}_k}{\mathbf{h}_k^T\mathbf{h}_k}\mathbf{h}_k^T\,, \tag{2.16}$$

where $\mathbf{h}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$. The mapped parameters $\mathbf{z}_k$ is the result of evaluating the mapping function (2.10) at $\mathbf{x}_k$, $\mathbf{z}_k = \mathbf{p}(\mathbf{x}_k)$.

The unsophisticated guess, with no prior knowledge, for a mapping function would be a unity mapping. For this reason, the Broyden approximation should be initialized with the identity matrix, i.e. $\mathbf{B}_0 = \mathbf{I}(n,n)$ yielding a unity mapping at the first iteration, when setting $\mathbf{z}_0 = \mathbf{x}_0$. The updating formula (2.16) ensures that the approximation is corrected at each iteration step, if the actual mapping should differ from the unity mapping used as first approximation.

Compared to the finite difference approach, use of the linearized mapping function $\mathbf{p}_k$ has the advantages of using only one function evaluation per iteration.

The approximation error between the real mapping $\mathbf{p}$ and the linearized mapping function $\mathbf{p}_k$, for some norm in $\mathbb{R}^n$,

$$\|\mathbf{p}(\mathbf{x}) - \mathbf{p}_k(\mathbf{x})\|\,, \tag{2.17}$$

depends on two factors: the distance $\|\mathbf{x} - \mathbf{x}_k\|$ from the current set of parameters $\mathbf{x}_k$, and the quality of the Broyden approximation $\mathbf{B}_k$ to the Jacobian $\mathbf{p}'(\mathbf{x}_k)$. We define a trust region limiting the use of the linear model to a region, wherein the error (2.17) is acceptable: The linearization $\mathbf{p}_k$ is only accepted for the set $\{\mathbf{x}\,|\,\|\mathbf{x} - \mathbf{x}_k\| \leqslant \delta_k\}$, which form a confined region around $\mathbf{x}_k$. The distance measure δ_k is the size of the trust region at the kth iteration.

The size of the trust region should reflect the ability of linear model $\mathbf{p}_k$ to predict descent directions towards $\mathbf{z}^*$ for the mapping function $\mathbf{p}$. A poor alignment between the mapping approximation $\mathbf{p}_k$ and the actual mapping $\mathbf{p}$ should result in a small trust region. Contrary a good alignment should result in a large trust region. The updating formula used in our actual Space Mapping implementation is presented in chapter 3.

In figure (2.4) a one dimensional example mapping function p is sketched together with a linear approximation $p_k(x)$, which is bounded by a trust region of size δ_k.

A second order model could be used instead of the linear model, but the improvement of including second order terms would not compare favourably with the heavily increased effort needed in updating the second order model.

2.2.4 Iterative formulation

Using the method of local linear models of the mapping function we can formulate the Space Mapping problem (2.12) as optimization of $C(\mathbf{p}_k(\mathbf{x}))$

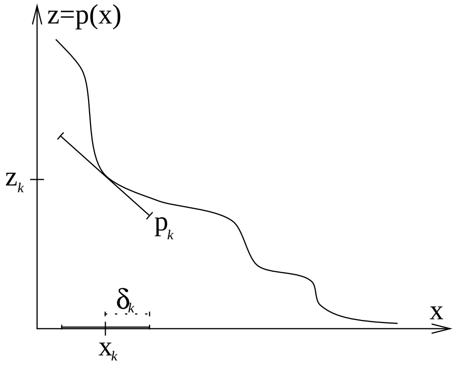

Figure 2.4: *The figure shows an example mapping function $p(x)$ and a linear mapping approximation $p_k(x)$ for a problem in one dimension.*

for sequential linear approximations $\mathbf{p}_k$ of the mapping function

$$\mathbf{x}_{k+1} = \underset{\mathbf{x}}{\operatorname{argmin}}\, C(\mathbf{p}_k(\mathbf{x})) \quad \text{for } \|\mathbf{x} - \mathbf{x}_k\| \leqslant \delta_k \,. \tag{2.18}$$

Hence $C(\mathbf{p}_k(\mathbf{x}))$, $k = 0, 1, \dots$, acts as surrogates, sequentially approximating $F(\mathbf{x})$, the main objective function, in the vicinity of the actual set of parameters $\mathbf{x}_k$. This iterative formulation is well suited for trust region, gradient based methods, like e.g. sequential linear programming for minimax [14] and various Newton like methods (for instance Levenberg-Marquardt for non-linear least squares) [15], depending on the choice of merit function.

Before we discuss previous iterative formulations, we will first have a closer look at how the Broyden formula actually works in relation to the iterative formulation above.

Broyden approximation

The Broyden formula (2.16) works by adding corrections to the approximation of the Jacobian at each iteration step. We now derive the correction to the Broyden approximation resulting from the first iteration step, using the conventional Space Mapping formulation in (2.18):

We assume that the first mapping approximation $\mathbf{p}_0$ is a unity mapping, i.e. $\mathbf{B}_0 = \mathbf{I}(n, n)$. For a given starting point $\mathbf{x}_0$ we let $\mathbf{z}_0 = \mathbf{x}_0$. It follows from (2.18) and (2.6) that,

$$\begin{aligned} \mathbf{x}_1 &= \underset{\mathbf{x}}{\operatorname{argmin}}\, C(\mathbf{p}_0(\mathbf{x})) \\ &= \underset{\mathbf{x}}{\operatorname{argmin}}\, C(\mathbf{z}_0 + \mathbf{I}(\mathbf{x} - \mathbf{x}_0)) \\ &= \underset{\mathbf{x}}{\operatorname{argmin}}\, C(\mathbf{x}) \\ &= \mathbf{z}^* \,, \end{aligned} \tag{2.19}$$

when assuming that $\|\mathbf{x}_1 - \mathbf{x}_0\| \leqslant \delta_0$. Hence if $\mathbf{z}^*$ is encompassed by the trust region, i.e. $\|\mathbf{z}^* - \mathbf{x}_0\| \leqslant \delta_0$, the second set of fine parameters $\mathbf{x}_1$ is equal to the optimal set of coarse parameters $\mathbf{z}^*$.

The mapped set of parameters $\mathbf{z}_1$ is obtained by evaluating the mapping function (2.10) in $\mathbf{x}_1$, hence $\mathbf{z}_1 = \mathbf{p}(\mathbf{x}_1)$.

The step $\mathbf{h}_0 = \mathbf{x}_1 - \mathbf{x}_0$ and the mapped set of parameters $\mathbf{z}_1$ and $\mathbf{z}_0$ are used with (2.16) to get the first correction to the Jacobian approximation of the mapping function,

$$\begin{aligned} \mathbf{B}_1 &= \mathbf{B}_0 + \frac{\mathbf{z}_1 - \mathbf{z}_0 - \mathbf{B}_0\mathbf{h}_0}{\mathbf{h}_0^T\mathbf{h}_0}\mathbf{h}_0^T \\ &= \mathbf{I} + \frac{(\mathbf{z}_1 - \mathbf{z}^*)(\mathbf{x}_1 - \mathbf{x}_0)^T}{\|\mathbf{x}_1 - \mathbf{x}_0\|_2^2} \,. \end{aligned} \tag{2.20}$$

If the actual mapping function (2.10) is a unity mapping, then $\mathbf{z}_1 = \mathbf{p}(\mathbf{x}_1) = \mathbf{z}^*$, and the correction to the Jacobian approximation is zero. If, as expected, the actual mapping function is not a unity mapping, the Jacobian approximation is corrected by $\mathbf{z}_1 - \mathbf{z}^*$ in the direction $\mathbf{h}_0$. It is easy to realize that the second mapping approximation $\mathbf{p}_1$ fulfill the conditions $\mathbf{p}_1(\mathbf{x}_0) = \mathbf{z}_0$ and $\mathbf{p}_1(\mathbf{x}_1) = \mathbf{z}_1$, and that for all directions perpendicular to $\mathbf{h}_0$, the mapping approximation is still a unity mapping.

Note the special case for the one dimensional problem: The Broyden formula (2.16) is a generalized secant method, hence for the one dimensional case it corresponds exactly to the well known secant method. Hence the updating formula is reduced to

$$\begin{aligned} B_{k+1} &= B_k + \frac{z_{k+1} - z_k - B_k(x_{k+1} - x_k)}{(x_{k+1} - x_k)^2}(x_{k+1} - x_k) \\ &= B_k + \frac{z_{k+1} - z_k}{x_{k+1} - x_k} - B_k \\ &= \frac{z_{k+1} - z_k}{x_{k+1} - x_k} . \end{aligned} \tag{2.21}$$

Previous iterative formulations

The linearization approach with Broyden update to the Jacobian, used for solving the conventional Space Mapping problem, is also used in conjunction with a quasi-Newton method in [6], and with a Levenberg-Marquardt type damped Newton in [2, 3], to iteratively solve the Aggressive Space Mapping problem (2.14)

$$\mathbf{x}_{k+1} = \underset{\mathbf{x}}{\operatorname{argmin}} \ ||\mathbf{p}_k(\mathbf{x}) - \mathbf{z}^*|| \quad \text{for } ||\mathbf{x} - \mathbf{x}_k|| \leqslant \delta_k \ , \tag{2.22}$$

where $\mathbf{p}_k$, $k = 0, 1, \ldots$, are sequential linear approximations of the mapping function (2.10). Observe that the formulation requires knowledge of the exact coarse model optimum. When implementing the formulation in (2.22), this information is utilized by setting $\mathbf{x}_0 = \mathbf{z}^*$. Recall from (2.19) how the formulation (2.18) yields $\mathbf{x}_1 = \mathbf{z}^*$ for a trust region large enough. Hence the two formulations are apparently very similar. In the next subsection we will analyse in depth the relation between the two iterative formulations of the conventional Space Mapping problem.

Relating the formulations

Both the formulations in (2.18) and (2.22) lead to sequential solving of a system of linear equations

$$\mathbf{p}_k(\mathbf{x}) = \hat{\mathbf{z}} \tag{2.23}$$

for $\mathbf{x}$, where the value of $\hat{\mathbf{z}}$ depends, as we derive below, on the size of the trust region. The solution $\mathbf{x} = \hat{\mathbf{x}}$ to (2.23) is trivial

$$\begin{aligned} & \mathbf{z}_k + \mathbf{B}_k(\hat{\mathbf{x}} - \mathbf{x}_k) = \hat{\mathbf{z}} \\ \Rightarrow \ & \hat{\mathbf{x}} = \mathbf{x}_k + \mathbf{B}_k^{-1}(\hat{\mathbf{z}} - \mathbf{z}_k) \, . \end{aligned} \tag{2.24}$$

Regarding the value of $\hat{\mathbf{z}}$, we discern between two situations:

1. The coarse model optimum $\mathbf{z}^*$ is encompassed by the current trust region, i.e. $||\mathbf{B}_k^{-1}(\mathbf{z}^* - \mathbf{z}_k)|| \leqslant \delta_k$, as sketched in figure 2.5. Then $\hat{\mathbf{z}} = \mathbf{z}^*$ for both the formulations (2.18) and (2.22).

2. The coarse model optimum $\mathbf{z}^*$ is outside the trust region, as sketched in figure 2.6:
    - The formulation (2.22) yields $\hat{\mathbf{z}} = \hat{\mathbf{z}}_1$, where $\hat{\mathbf{z}}_1$ lies on the straight line between $\mathbf{z}_k$ and $\mathbf{z}^*$ in the distance $||\hat{\mathbf{z}}_1 - \mathbf{z}_k|| = \delta_k$ from $\mathbf{z}_k$. Hence $\hat{\mathbf{z}} = \mathbf{z}_k + \alpha(\mathbf{z}^* - \mathbf{z}_k)$ for the value of $\alpha \in [0; 1]$ which makes $\hat{\mathbf{z}}$ fulfill the demand $||\mathbf{B}_k^{-1}(\hat{\mathbf{z}} - \mathbf{z}_k)|| = \delta_k$. Simple calculations lead to,
    $$\alpha = \frac{\delta_k}{||\mathbf{B}_k^{-1}(\mathbf{z}^* - \mathbf{z}_k)||} \, . \tag{2.25}$$
    - The formulation (2.18) yields $\hat{\mathbf{z}} = \hat{\mathbf{z}}_2$, where $\hat{\mathbf{z}}_2$ minimizes $C(\mathbf{z})$ for $||\mathbf{z} - \mathbf{z}_k|| \leqslant \delta_k$. This might not necessarily be on the straight line between $\mathbf{z}_k$ and $\mathbf{z}^*$, due to the non-linearity of C.

    Hence the value of $\hat{\mathbf{z}}$ might not be the same for the two formulations (2.18) and (2.22). Note that for both formulations we have that $||\hat{\mathbf{z}} - \mathbf{z}_k|| = \delta_k$, hence $\hat{\mathbf{z}}$ will lie on the edge of the trust region, due to $\mathbf{z}^*$ not being inside the trust region.

In both figure 2.5 and figure 2.6 it is illustrated how the solution $\hat{\mathbf{x}}$ to (2.23) might not lie on the straight line between $\mathbf{x}_k$ and $\mathbf{x}^*$, due to the approximation error (2.17).

Illustrating the method

The iterative formulation (2.22) of the Aggressive Space Mapping problem is easy to illustrate by geometric reflections for a one dimensional problem:

We reuse the example mapping function plotted in figure 2.4. Below we follow the iteration process of the first four iterations, $k = 0, 1, 2, 3$, using the iterative formulation in (2.22). We keep the trust region at a constant size δ, since we have not yet discussed exactly how to update it.

$k = 0$ In the first iteration step we set $B_0 = 1$, $x_0 = z^*$ and $z_0 = p(x_0)$. Hence $p_0(x) = x + z_0 - z^*$. Figure 2.7 pictures the situation, the trust

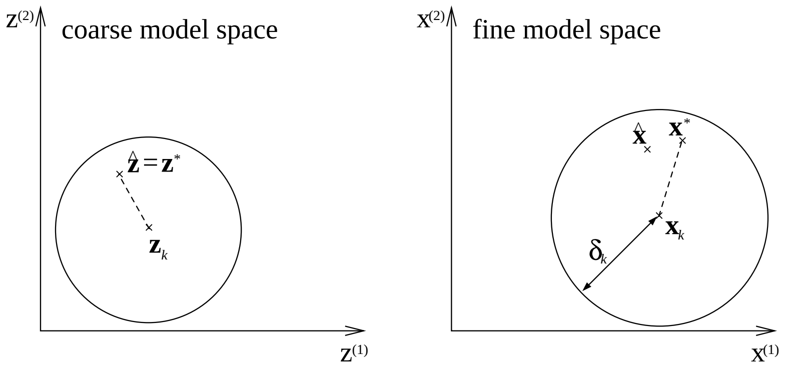

Figure 2.5: *The figure shows that the formulations* (2.18) *and* (2.22) *yield the same solution when solving* (2.23), *if* $\mathbf{z}^*$ *is encompassed by the trust region, here for a problem in two dimensional space.*

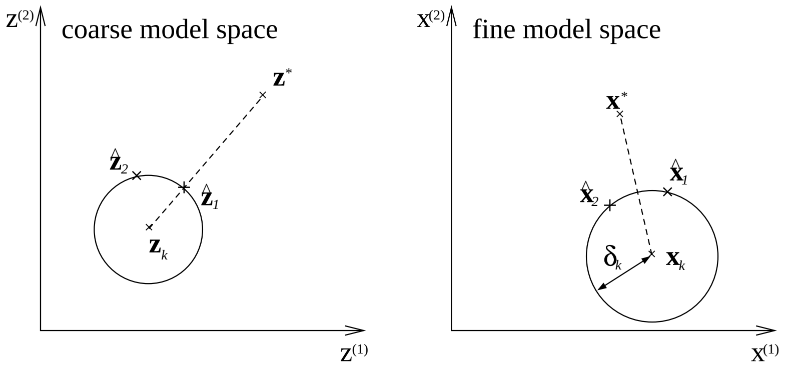

Figure 2.6: *The figure shows that the formulations* (2.18) *and* (2.22) *yield different solutions when solving* (2.23), *if* $\mathbf{z}^*$ *is not encompassed by the trust region, here for a problem in two dimensional space.*

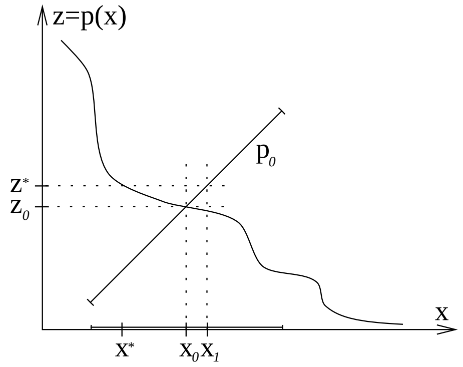

Figure 2.7: *First iteration step,* $k = 0$, *using* (2.22).

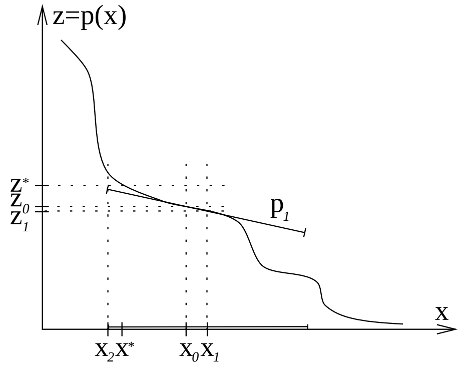

Figure 2.8: *Second iteration step,* $k = 1$, *using* (2.22).

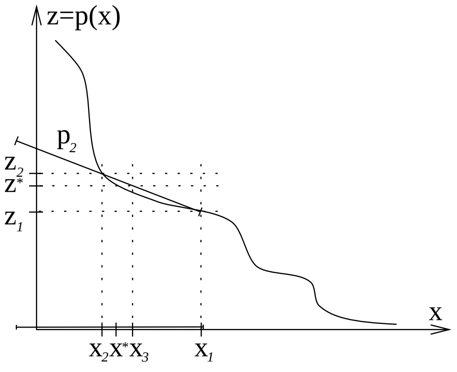

Figure 2.9: *Third iteration step, $k = 2$, using* (2.22).

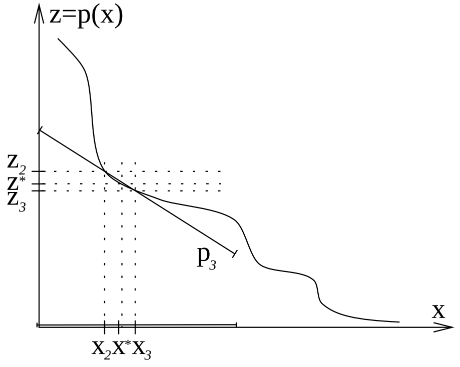

Figure 2.10: *Fourth iteration step, $k = 3$, using* (2.22).

region is marked by the line just above the first axis. Following (2.22), we wish to minimize $|p_0(x) - z^*|$ by varying x for $|x - x_0| \leqslant \delta$. The solution is $x_1 = 2z^* - z_0$, as marked on the figure, hence $x_1 = x_0 + z^* - z_0$. Graphically the solution is found, where the linearization p_0 crosses the dotted, horizontal line in z^*.

$k = 1$ In the second iteration step we evaluate the mapping function (2.10) to get $z_1 = p(x_1)$. It follows from (2.21) that $B_1 = \frac{z_1 - z_0}{x_1 - x_0}$. Relating to figure 2.8, we wish to minimize $|p_1(x) - z^*|$ by varying x for $|x - x_1| \leqslant \delta$. As seen on the figure, the solution x_2 is located pretty close to x^*, the solution we wish to find. Graphically the solution is found, where the linearization p_1 is closest to the dotted, horizontal line in z^*. The two lines do not cross each other, due to p_1 being limited by the trust region.

$k = 2$ From the above iterations, the third iteration step is straight forward: We have $z_2 = p(x_2)$, and following (2.21) we have $B_2 = \frac{z_2 - z_1}{x_2 - x_1}$. Relating to figure (2.9), we wish to minimize $|p_2(x) - z^*|$ by varying x for $|x - x_2| \leqslant \delta$. The solution x_3 is located near x^*.

$k = 3$ Fourth iteration step: Relating to figure 2.10 we have $z_3 = p(x_3)$ and $B_3 = \frac{z_3 - z_2}{x_3 - x_2}$, and we wish to minimize $|p_3 - z^*|$ for x for $|x - x_3| \leqslant \delta$. The solution x_4 (not marked on the figure) is located very close to x^* the optimal solution. It is not possible with this plot to make further iteration steps graphically.

When looking at the iteration process, it is clear that the method is converging towards the optimum. But notice what could have happened if the size of the trust region had been larger in second iteration step (refer to figure 2.8): Due to the large linearization error, where the linearization p_1 crosses the vertical dotted line in z^*, the solution to $p_1(x) = z^*$ would yield a poor step, and the convergence of the method would be disturbed. This indicates, that we need to adjust the size of the trust region during every iteration step, according to the quality of the actual linearization. In chapter 3 we will set up an updating scheme for the size of the trust region.

Other applications

In earlier works the mapping idea from the conventional Space Mapping have been used for modelling purposes. See for instance [9] for an artificial neuromodel replacement of the mapping function, or [8] for a piecewise linear model replacement of the mapping function. These two approaches both aim at establishing a complete approximation of the mapping function for some predefined regions of the parameter spaces, yielding a mapped coarse model which is a good approximation to the fine model, in the predefined regions of interest. Hence the mapped coarse model is capable of substituting the fine model in various evaluation intensive explorations of the fine model properties, e.g. Monte Carlo analysis, and thereby speeding the exploration process up by several orders of magnitude.

2.3 The novel Space Mapping approach

In this section we deal with the problems arising when the mapping is imperfect. We develop a novel approach, which utilizes a local fine model approximation and combines this with the mapped coarse model in a weighted so-called combined model. The aim is a combined model performing well in all cases whether the mapping is perfect or imperfect.

2.3.1 A local fine model approximation

We define a linear approximation to (2.8) the linearization of the fine model using a Broyden approximation to the Jacobian of the fine model[1]

$$\mathbf{l}_k(\mathbf{t};\mathbf{x}) = \mathbf{g}(\mathbf{t};\mathbf{x}_k) + \hat{\mathbf{B}}_k(\mathbf{x}-\mathbf{x}_k)\,, \tag{2.26}$$

valid for $||\mathbf{x}-\mathbf{x}_k|| \leqslant \delta_k$, where $\hat{\mathbf{B}}_k$ is a Broyden rank one approximation to the Jacobian of the fine model. We update $\hat{\mathbf{B}}$ using the formula

$$\hat{\mathbf{B}}_{k+1} = \hat{\mathbf{B}}_k + \frac{\mathbf{g}(\mathbf{x}_{k+1}) - \mathbf{g}(\mathbf{x}_k) - \hat{\mathbf{B}}_k\mathbf{h}_k}{\mathbf{h}_k^T\mathbf{h}_k}\mathbf{h}_k^T\,, \tag{2.27}$$

where $\mathbf{h}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$.

[1] We will interchangeably use the notations $\mathbf{l}_k(\mathbf{t};\mathbf{x})$ and $\mathbf{l}_k(\mathbf{x})$.

Considering the second iteration using (2.18) it follows from (2.11) that

$$\mathbf{g}(\mathbf{x}_1) \simeq \mathbf{d}(\mathbf{p}_1(\mathbf{x}_1))\,.$$

In general we have

$$\begin{aligned}\frac{\partial g^{(i)}}{\partial x^{(j)}} &\simeq \frac{\partial d^{(i)}}{\partial p^{(1)}}\frac{\partial p^{(1)}}{\partial x^{(j)}} + \frac{\partial d^{(i)}}{\partial p^{(2)}}\frac{\partial p^{(2)}}{\partial x^{(j)}} + \ldots + \frac{\partial d^{(i)}}{\partial p^{(n)}}\frac{\partial p^{(n)}}{\partial x^{(j)}} \\ &= \sum_{q=1}^{n}\frac{\partial d^{(i)}}{\partial p^{(q)}}\frac{\partial p^{(q)}}{\partial x^{(j)}}\end{aligned} \tag{2.28}$$

where

$$\frac{\partial d^{(i)}}{\partial p^{(q)}} = \frac{\partial d^{(i)}}{\partial z^{(q)}}$$

is the ith row in the Jacobian of the coarse model $\mathbf{d}'(\mathbf{z})$, and

$$\frac{\partial p^{(q)}}{\partial x^{(j)}} = \frac{\partial z^{(q)}}{\partial x^{(j)}}$$

is the jth column in the Jacobian of the mapping function $\mathbf{p}'(\mathbf{x})$. Hence for the second iteration, in matrix notation,

$$\begin{aligned}\mathbf{g}'(\mathbf{x}_1) &\simeq \mathbf{d}'(\mathbf{p}_1(\mathbf{x}_1))\mathbf{p}_1' \\ &= \mathbf{d}'(\mathbf{z}_1)\mathbf{B}_1\,.\end{aligned} \tag{2.29}$$

If we assume $||\mathbf{z}_0 - \mathbf{z}^*|| \leqslant \delta$, it follows from (2.20) and (2.29) that the approximation is

$$\mathbf{g}'(\mathbf{x}_1) \simeq \mathbf{d}'(\mathbf{z}_1)\left(\mathbf{I} + \frac{(\mathbf{z}_1 - \mathbf{z}^*)(\mathbf{x}_1 - \mathbf{x}_0)^T}{||\mathbf{x}_1 - \mathbf{x}_0||_2^2}\right)\,. \tag{2.30}$$

The observation in (2.29) gives us an approximation of the Jacobian of the fine model in $\mathbf{x}_1$. Thus we can use this approximation to initialize the Jacobian $\hat{\mathbf{B}}_1$ of the linear approximation $\mathbf{l}_1$ to the fine model.

Note that a sequential optimization of linearizations of the fine model is similar to a direct optimization of the fine model. Hence, with the objective function $L_k(\mathbf{x}) = H(\mathbf{l}_k(\mathbf{t};\mathbf{x}) - \mathbf{y})$, and the iterative formulation

$$\mathbf{x}_{k+1} = \underset{\mathbf{x}}{\operatorname{argmin}}\ L_k(\mathbf{x}) \ \text{ for } ||\mathbf{x} - \mathbf{x}_k|| \leqslant \delta_k\,, \tag{2.31}$$

we have $\mathbf{x}_k \to \mathbf{x}^*$ for $k \to \infty$. See [14] for a proof using the infinity norm. The infinity norm is not demanded by the proof, hence the proof is valid for other norms as well. This statement is supported by [17] at page 261.

2.3.2 Combined model

We introduce the *combined model*, $\mathbf{v}_k(\mathbf{t};\mathbf{x})$, being a weighted function of the mapped coarse model approximation $\mathbf{d}(\mathbf{p}_k(\mathbf{x}))$ and the linear fine model approximation $\mathbf{l}_k$. Hence

$$\mathbf{v}_k(\mathbf{t};\mathbf{x}) = \omega_k \cdot \mathbf{d}(\mathbf{t};\mathbf{p}_k(\mathbf{x})) + (1-\omega_k)\cdot \mathbf{l}_k(\mathbf{t};\mathbf{x}) \, , \tag{2.32}$$

where $\omega_k \in [0;1]$ is the weighting factor.

With the residuals $\mathbf{s}_k(\mathbf{t};\mathbf{x}) = \mathbf{v}_k(\mathbf{t};\mathbf{x}) - \mathbf{y}$ of the combined model to the given points, and the objective function $S_k(\mathbf{x}) = H(\mathbf{s}_k(\mathbf{x}))$ we can formulate an improved version of the conventional Space Mapping problem (2.12)

$$\mathbf{x}_{k+1} = \underset{\mathbf{x}}{\operatorname{argmin}} \; S_k(\mathbf{x}) \;\; \text{for} \; ||\mathbf{x}-\mathbf{x}_k|| \leqslant \delta_k \, . \tag{2.33}$$

Hence $S_k(\mathbf{x})$, $k = 0, 1, \ldots$, acts as surrogates, sequentially approximating $F(\mathbf{x})$, the main objective function, in the vicinity of the actual set of parameters $\mathbf{x}_k$.

It is easy to see that the properties of the combined model are as follows:

- For $\omega_k = 1$ (2.33) corresponds exactly to (2.18), and
- for $\omega_k = 0$ the successive solving of (2.33) corresponds exactly to a direct optimization of the fine model as in (2.31).

The parameter ω_k defines the weight of the mapped coarse model approximation and the linear fine model approximation, when calculating the combined model. We expect that the usefulness of the linear model increases as the iteration steps approach the optimum of the fine model. On the other hand, we expect the coarse model to be insubstantial in describing the fine model accurately in the vicinity of the optimum. Hence we should set $\omega_0 = 1$ to exploit the coarse model in the first iteration steps, and we should reduce ω_k during iterations as approaching the optimum, causing the linear model to dominate over the coarse model.

Determining the right size of the trust region of the combined model is not straight forward. It is not advisable to update the trust region size on the basis of the ability of S_k to predict descent directions for F. This is because S_k might promote uphill steps for F, even for a small trust region, if the mapping is imperfect, as the following reasoning shows: Consider the situation sketched in figure 2.11 of the kth iteration, using (2.33), in a two dimensional problem. We have that the actual set of parameters $\mathbf{x}_k$ is close to the main problem optimum $\mathbf{x}^*$. But, the mapping is imperfect, i.e. $\mathbf{p}(\mathbf{x}^*) \neq \mathbf{z}^*$, for which reason the Space Mapping solution $\mathbf{x}^*_{SM}$ is not equal to $\mathbf{x}^*$. From the figure we have that $\mathbf{x}^*_{SM}$ is in an uphill direction from $\mathbf{x}_k$, i.e. $F(\mathbf{x}_k) < F(\mathbf{x}^*_{SM})$. Further we have that the $\mathbf{l}_k$ part of the combined model promotes that the next step should be downhill towards $\mathbf{x}^*$, and we have that the $\mathbf{d}(\mathbf{p}_k(\mathbf{x}))$ part of the combined model promotes that the next step should be uphill towards $\mathbf{x}^*_{SM}$. The result is that the next step $\mathbf{h}_k$ resulting from (2.33) might be an uphill direction for F for a large enough value of ω_k.

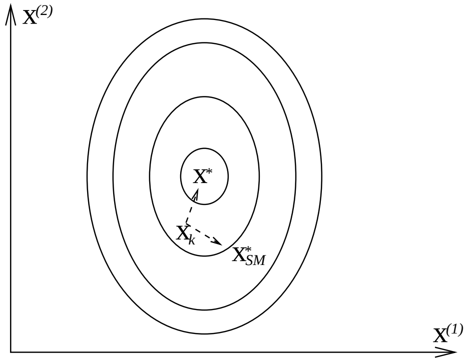

Figure 2.11: *The figure shows a contour plot, with $\mathbf{x}^*$ as minimum, of an example fine model F. The situation depicted is the kth iteration using (2.33). From the current parameters $\mathbf{x}_k$, the Space Mapping solution $\mathbf{x}^*_{SM}$ is in an uphill direction for F, and the main problem solution $\mathbf{x}^*$ is in a downhill direction for F.*

Thus we should not base the trust region size of the combined model, on information arising from a model which might not be able to describe the fine model in the vicinity of the optimum. If we somehow detect that the mapping is imperfect, we should decrease ω, such that the linear model is weighted higher. We will set the size of the trust region δ_k for the combined model according to the ability of L_k to predict descent directions for F. The updating formula used in our actual implementation is presented in chapter 3.

The advantage of the combined model is the independence of the mapping being perfect or imperfect. No matter how the coarse model is able to describe the fine model optimum, the linear model part of the combined model will ensure a good fit to the fine model, as long as the weight ω_k is reduced towards zero as approaching the optimum. Hence $\mathbf{s}_k(\mathbf{x}_k) \to \mathbf{f}(\mathbf{x}^*)$ for $\omega_k \to 0$ and $k \to \infty$, and therefore we have $\mathbf{x}_k \to \mathbf{x}^*$ for $k \to \infty$.

Notice that the objective function S_k is only slightly more expensive to evaluate than the conventional Space Mapping objective function $C(\mathbf{p}_k(\mathbf{x}))$. Since the fine model response $\mathbf{g}$, used for updating the Jacobian approximation in (2.27), has to be evaluated anyhow, for being able to evaluate the mapping function (2.10) used to update the mapping approximation $\mathbf{p}_k$.

In practical problems we believe that it is rare that the mapping is perfect between two models describing the same system. It might be more common that the mapping is near perfect, hence the difference

$$H(\mathbf{r}(\mathbf{x}^*, \hat{\mathbf{z}})) - H(\mathbf{r}(\mathbf{x}^*, \mathbf{z}^*)),$$

for $\hat{\mathbf{z}} = \mathbf{p}(\mathbf{x}^*)$, is rather small. These statements are supported by the test results obtained in [5], when applying direct optimization on the fine model using $\mathbf{x}^*_{SM}$ as starting point.

In the next chapter, chapter 3, we will look at implementation aspects and how to control the size of the trust region δ_k and the new parameter ω_k, introduced with the combined model.

Chapter 3

Aspects of implementation

The Space Mapping method, as described in chapter 2, rely on iterative numerical methods for finding (an approximation to) the solution $\mathbf{x}^*$ to the main problem defined in equation (2.4). The algorithms describing the conventional Space Mapping method and the new approach can be presented in a simple outline form similar to the one below.

```
Given $\mathbf{x}_0$, $\delta_0$
$k = 0$
loop
    $\mathbf{x}_{k+1} = \underset{\mathbf{x}}{\text{argmin}}\ \Psi_k(\mathbf{x})$
            for $||\mathbf{x}_{k+1} - \mathbf{x}_k|| \leqslant \delta_k$
    Determine $\Psi_{k+1}$ and $\delta_{k+1}$
    $k = k + 1$
until stop
```

where $\Psi_k : \mathbb{R}^n \to \mathbb{R}$, $k = 0, 1, \ldots$, are surrogate functions, valid for confined regions $||\mathbf{x} - \mathbf{x}_k|| \leqslant \delta_k$ of the parameter space $\mathbf{x} \in \mathbb{R}^n$. The surrogate functions are approximating the main objective function F in their regions of validness.

The algorithm outlined states that for a given starting point $\mathbf{x}_0$, and size of the trust region δ_0, sets of parameters $\{\mathbf{x}_k\}$, $k = 0, 1, \ldots$, are iteratively calculated in the effort of approximating the fine model optimum $\mathbf{x}^*$. In each iteration a subproblem of optimizing a surrogate Ψ_k to F is solved, and the next surrogate is formed. The iteration loop is stopped according to some predefined stopping criteria, controlling the desired precision.

The remainder of this chapter is divided into two sections: Section 3.1 deals with aspects of implementing the conventional Space Mapping theory, hence using surrogates of the form $\Psi_k(\mathbf{x}) = C(\mathbf{p}_k(\mathbf{x}))$. Section 3.2 presents a novel Space Mapping algorithm using the combined model described in section 2.3.2, hence using surrogates of the form $\Psi_k(\mathbf{x}) = S_k(\mathbf{x})$.

3.1 Conventional Space Mapping

We will now focus on the aspects of implementing the conventional Space Mapping method, as it is presented in section 2.2. Recall that the conventional Space Mapping solution $\mathbf{x}^*_{SM}$ is only a solution to the main problem (2.4) if the mapping is perfect, i.e. if $\mathbf{p}(\mathbf{x}^*) = \mathbf{z}^*$. Hence the implementation described in this section will only converge to a main problem solution in cases where the models implicated yield a perfect mapping. Thus we should keep in mind that the poorer the mapped coarse model approximates the fine model in $\mathbf{x}^*$ the optimum, the poorer the conventional Space Mapping solution will approximate a main problem solution.

At first, in this section, we present an outline algorithm of the conventional Space Mapping method, secondly we deal with aspects of controlling the size of the trust region, and finally we discuss which stopping criteria to use.

3.1.1 Simple algorithm

A simple algorithm using the iterative formulation in (2.18) of the conventional Space Mapping is presented below.

```
Given $\mathbf{x}_0$, $\delta_0$
$\mathbf{B}_0 = \mathbf{I}(n,n)$, $\mathbf{z}_0 = \mathbf{x}_0$, $k = 0$
loop
    $\mathbf{x}_{k+1} = \underset{\mathbf{x}}{\text{argmin}}\ C(\mathbf{p}_k(\mathbf{x}))$
        for $\|\mathbf{x} - \mathbf{x}_k\| \leqslant \delta_k$
    Evaluate $\mathbf{g}(\mathbf{x}_{k+1})$
    $\mathbf{z}_{k+1} = \underset{\mathbf{z}}{\text{argmin}}\ H(\mathbf{g}(\mathbf{x}_{k+1}) - \mathbf{d}(\mathbf{z}))$
    Update $\delta$, $\mathbf{B}$
    $k = k + 1$
until stop
```

Here $\mathbf{I}(n,n)$ is the identity matrix of dimension n. Observe that one evaluation of the fine model is used per iteration.

The approximation $\mathbf{B}_k$ to the Jacobian of the mapping function is updated by the Broyden rank one update, presented in (2.16). We recall that if the approximation is initialized with the identity matrix, the first mapping approximation corresponds to a unity mapping, i.e. $\mathbf{p}_0(\mathbf{x}) = \mathbf{x}\ \ \forall\,\mathbf{x}$. In (2.19) we showed how the unity mapping lead to $\mathbf{x}_1 = \mathbf{z}^*$, when assuming that $\|\mathbf{z}^* - \mathbf{x}_0\| \leqslant \delta_0$.

3.1.2 The trust region

We have mentioned earlier how the size of the trust region should be based on the performance of the mapping approximation $\mathbf{p}_k(\mathbf{x})$. If the approximating model shows a good similarity with the observed behaviour of actual mapping $\mathbf{p}(\mathbf{x})$, the trust region should be adjusted in size to reflect this. When $\mathbf{p}_k$ is good at predicting descent steps leading $\mathbf{p}$ towards $\mathbf{z}^*$, the size of the trust region should be increased, and contrary when $\mathbf{p}_k$ is not good at predicting descent steps, the size of the trust region should be decreased.

Observe that for the perfect mapping, we have from figure 2.6 at page 25 that for a descent step $\hat{\mathbf{h}}_1 = \hat{\mathbf{x}}_1 - \mathbf{x}_k$ in the conventional Space Mapping problem

$$\|\mathbf{p}_k(\mathbf{x}_k + \hat{\mathbf{h}}_1) - \mathbf{z}^*\|\,,$$

there is a corresponding descent step $\hat{\mathbf{h}}_2 = \hat{\mathbf{x}}_2 - \mathbf{x}_k$ in the novel formulation of the conventional Space Mapping problem

$$C(\mathbf{p}_k(\mathbf{x}_k + \hat{\mathbf{h}}_2))\,.$$

Hence when using the novel Space Mapping formulation we can base the adjustment of the trust region size on the ability of the mapped coarse model approximation to predict descent steps for the fine model.

To set up a measure for the ability of $C(\mathbf{p}_k(\mathbf{x}))$ to predict the improvement of F, we define ΔF and ΔC,

$$\Delta F = F(\mathbf{x}_k) - F(\mathbf{x}_{k+1})\,, \tag{3.1a}$$

$$\Delta C = C(\mathbf{p}_k(\mathbf{x}_k)) - C(\mathbf{p}_k(\mathbf{x}_{k+1}))\,. \tag{3.1b}$$

The value of ΔF is the actual decrease in the fine models deviation to the given points, when taking the step $\mathbf{h}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$. Similarly the value of ΔC is the decrease in the mapped coarse model approximations deviation to the given points, when taking the step $\mathbf{h}_k$. We observe that from relation (2.11) it follows that the decrease in the surrogate, ΔC, should predict the decrease in the main objective function ΔF, if the mapped coarse model is well aligned with the fine model. Hence the ratio $\Delta F/\Delta C$ is a local measure for the alignment or similarity of the models in the direction $\mathbf{h}_k$ from the current set of parameters $\mathbf{x}_k$.

Following [12], we define an updating scheme for the size of the trust region based on the above observation

$$\delta_{k+1} = \begin{cases} 2\delta_k & \text{if } \Delta F > 0.75\Delta C \\ \delta_k/3 & \text{if } \Delta F < 0.25\Delta C \\ \delta_k & \text{otherwise}\,. \end{cases} \tag{3.2}$$

The updating scheme works as follows: If the actual decrease, ΔF, is greater than 75% of the predicted decrease, ΔC, the size of the trust region is increased. If the actual decrease is less than 25% of the predicted decrease, implying that the models are not well aligned, the size of the trust region is decreased. In neither cases, the trust region size is left unchanged. Notice that we do not reduce the size of the trust region if $C(\mathbf{p}_k(\mathbf{x}))$ under-predicts the improvement of $F(\mathbf{x})$, since we in the context of unconstrained optimization are focused on predicting descent rather than the general quality of the approximation. Though the quality of the approximation is important in constrained optimization, if one wishes to insure feasibility of the iteration steps.

The constants (0.75, 2, 0.25, 1/3) have been chosen from practical experience, and [12] states that the resulting method is not very sensitive to minor changes in these values; as long as the scaling factors (2 and 1/3) are chosen such that the values of δ_k cannot oscillate within the same set of values.

The initial size of the trust region δ_0 and starting point $\mathbf{x}_0$ is problem specific. Hence the choice should be made by a qualified user. A poor choice of δ_0, for instance a too small initial trust region, might result in use of unnecessarily many iteration steps and thereby disturb the convergence. Notice that we do not wish to bound the first step $\mathbf{h}_0$ made by the Space Mapping algorithm. If we do bound the step, in such way that $\mathbf{z}^*$ is not encompassed by the trust region, we will have $\mathbf{x}_1 \neq \mathbf{z}^*$, and thereby we will not fully exploit the advantage of having the coarse model available.

3.1.3 Stopping criteria

Setting the right stopping criteria is important to avoid unnecessarily many calculations and to assure the desired accuracy of the solution obtained. In general, we would like to use stopping criteria reflecting the convergence $\mathbf{x}_k \rightarrow \mathbf{x}^*$, hence stopping the algorithm when getting sufficiently close to the optimum, like the following criteria:

$$\|\mathbf{x}^* - \mathbf{x}_k\| < \hat{\varepsilon}_1 \tag{3.3}$$

$$F(\mathbf{x}_k) - F(\mathbf{x}^*) < \hat{\varepsilon}_2 \,. \tag{3.4}$$

Since we do not have knowledge of $\mathbf{x}^*$ during the iterations, we use the approximation $\mathbf{x}_{k+1}$ instead.

First criterion

The step length $\|\mathbf{h}_k\| = \|\mathbf{x}_{k+1} - \mathbf{x}_k\|$ may be used to approximate the left hand side of (3.3). Thus our first stop criterion proposed is to stop the algorithm, when the step length is decreased below some small threshold relative to the value of $\|\mathbf{x}_k\|$, to avoid rounding errors for problems where $\|\mathbf{x}_k\| \gg 1$,

$$\|\mathbf{h}_k\| < \varepsilon_1(1 + \|\mathbf{x}_k\|) \,, \tag{3.5}$$

where ε_1 is a small positive number. Notice that the right hand side of (3.5) is positive even if $\mathbf{x}_k = \mathbf{0}$.

The step size can get small due to one of the following three reasons, or a combination of them,

- the actual parameters is in the vicinity of a stationary point,
- the trust region size is getting small (and thereby forcing the step size down), or
- there is a strong curvature in the step path.

To prevent the algorithm from being stopped unintentionally by any of the two last-mentioned reasons, the value of ε_1 should be set very small, near machine accuracy.

If the algorithm detects that the size of the trust region decreases to a value near machine accuracy, which might cause numerical errors, it should be stopped and an error flag should be set.

If an error occurs during solving the subproblem (2.18), we might end up having $\mathbf{x}_{k+1} = \mathbf{x}_k$, i.e. $\|\mathbf{h}_k\| = 0$, without being in the vicinity of $\mathbf{x}^*$. In this case the algorithm should be stopped and an error flag should be set.

Second criterion

We can approximate the left hand side of (3.4) by ΔC, from (3.1b). Hence our second criterion proposed is to stop the algorithm if the decrease in function value decreases below some predefined threshold relative to the value $|C(\mathbf{p}_k(\mathbf{x}_k))|$, to avoid rounding errors for problems where $|C(\mathbf{p}(\mathbf{x}_k))| \gg 1$,

$$\Delta C < \varepsilon_2(1 + |C(\mathbf{p}_k(\mathbf{x}_k))|) \,, \tag{3.6}$$

where ε_2 is a small positive number. Note that if the value of ΔC is approaching zero, then it is due to one of the following three reasons, or a combination of them,

- the algorithm is approaching a stationary point,
- the curvature of $C(\mathbf{p}_k(\mathbf{x}))$ is vanishing, or
- the step size is becoming small.

The intention is to have the last reason mentioned controlled by the criterion in (3.5), and thus setting the value of ε_2 relative to the curvature of $C(\mathbf{p}_k(\mathbf{x}))$: If $C(\mathbf{p}_k(\mathbf{x}))$ is nearly flat, then ε_2 should be set to a lower value than if the mapped coarse model mostly has large curvatures, and thereby stop when the algorithm is close to a stationary point.

Third criterion

To prevent the algorithm from not stopping, if neither (3.5), or (3.6) are fulfilled within a reasonable number of iterations, we propose a third stopping criterion

$$k > k_{max}\ , \tag{3.7}$$

hence stopping the algorithm if a predefined maximum number of iterations k_{max} is exceeded. The value of k_{max} should reflect both the accuracy desired and the resources available for solving the given problem.

3.2 The novel Space Mapping approach

We will now focus on the implementation of the new additions to the Space Mapping theory. At first, an outline algorithm is presented. Here the combined model $\mathbf{v}_k(\mathbf{x})$ described in section 2.3.2 is used in sequential approximations of the fine model $\mathbf{g}(\mathbf{x})$. Secondly, the transition between the two parts of the combined model is discussed. Finally, it is discussed how the new approach influences the control of the trust region size, and the choice of stopping criteria.

3.2.1 Simple algorithm

A novel Space Mapping algorithm, based on the conventional Space Mapping algorithm from section 3.1.1, is presented below.

```
Given $\mathbf{x}_0$, $\delta_0$
$\mathbf{B}_0 = \mathbf{I}(n,n)$, $\mathbf{z}_0 = \mathbf{x}_0$, $k = 0$, $\omega_0 = 1$
loop
    $\mathbf{x}_{k+1} = \underset{\mathbf{x}}{\text{argmin}}\ S_k(\mathbf{x})$
        for $\|\mathbf{x} - \mathbf{x}_k\| \leqslant \delta_k$
    Calculate $\mathbf{g}(\mathbf{x}_{k+1})$
    $\mathbf{z}_{k+1} = \underset{\mathbf{z}}{\text{argmin}}\ H(\mathbf{g}(\mathbf{x}_{k+1}) - \mathbf{d}(\mathbf{z}))$
    Update $\delta$, $\omega$, $\mathbf{B}$, $\hat{\mathbf{B}}$
    $k = k + 1$
until stop
```

The outline algorithm reveals that the implementation of the combined model is straight forward, and merely is an extension of the conventional Space Mapping algorithm. Hence most of the remarks about the conventional Space Mapping algorithm are the same for the novel algorithm. Below we review the few differences between the algorithms.

Observe that we have $\omega_0 = 1$, hence the combined model equals the coarse model in the first iteration. At least the first iteration is identical with the first iteration of the conventional Space Mapping algorithm, except for the updating of $\hat{\mathbf{B}}$ and ω. Notice that by not using the linear model in the first iteration, we avoid evaluating the fine model in $\mathbf{x}_0$, as it would be necessary to do if we used a direct method.

The approximation to the Jacobian of the fine model $\hat{\mathbf{B}}$ is updated in each iteration by formula (2.27). In the first iteration $\hat{\mathbf{B}}$ is initialized by $\hat{\mathbf{B}}_1 = \mathbf{d}'(\mathbf{z}_1)\mathbf{B}_1$ as derived in (2.29). In the next subsection we deal with the combined model weighting parameter ω.

3.2.2 Transition between models

Choosing the right updating scheme for the weight ω, in the combined model $\mathbf{v}$, is essential for exploiting the full potential of the combined model. As described in section 2.3.2, it is expected that the coarse model response $\mathbf{d}$ may be most useful when situated relatively far away from $\mathbf{x}^*$, and that its usefulness will decrease as approaching $\mathbf{x}^*$ the solution. The linear fine model approximation $\mathbf{l}_k$ is expected to be most useful in the vicinity of the optimum, wherein the coarse model might be inadequate in describing the fine model.

From these observations, the obvious choice of updating scheme controlling the transition would be to utilize the coarse model, when situated far away from the optimal solution, and to (gradually) weight the linear model higher as approaching the optimum.

In the next subsections we will discuss the two following approaches of performing the transition,

- soft switching, and
- hard (or direct) switching.

Soft switching

We limit the discussion of soft transition to the case of an updating scheme for ω which yields a monotonously decreasing function for an increasing number of iterations. Hence ω is not at any time increased during iterations.

Since there is no guarantee that an arbitrary problem's coarse and fine model yield a perfect mapping, we are bound to rely on the linear model (2.26) when approaching the solution. Hence we should at least demand that $\omega_k \to 0$ for $k \to \infty$, i.e. the combined model should equal to or close to the linear model as the number of iterations approaches infinity.

To avoid hard switching we desire a bound on the reduction rate per iteration, for instance reducing ω by no more than a factor two per iteration $\omega_{k+1} \geqslant \frac{\omega_k}{2}$.

In general, we do not wish to change the weighting in the combined model, if the steps produced by the algorithm yield a sufficient reduction in F. Hence the larger gain in function value relative to the actual function value $\Delta F/F(\mathbf{x}_k)$, the less change we want in ω. Though, at a given iteration step we wish to utilize the model giving the best prediction of the decrease in F. Since the linear model $\mathbf{l}_k$ is corrected during iterations, and since we expect that the ratio $\Delta F/\Delta C$ will decrease as approaching $\mathbf{x}^*$, we consider the coarse model as the limiting factor for improving the quality of the combined model. Hence we will focus on the ability of C to predict decrease in F, to decide when we should weight the linear model higher in the combined model. We will consider C to be acceptable in predicting decrease in F, if $\Delta F > 0.25\Delta C$ the actual decrease in function value ΔF is at least 25% of the decrease predicted by ΔC. If this is not the case, we consider C to be inadequate in describing the decrease in F and the linear model should be weighted higher.

From the above discussion we propose the following updating formula

$$\omega_{k+1} = \frac{\omega_k}{1+\varphi_k} \tag{3.8}$$

where $\varphi_k \in [0;1]$ is

$$\varphi_k = \begin{cases} \frac{F(\mathbf{x}_{k+1})}{F(\mathbf{x}_k)} & \text{for } \Delta C > 0 \text{ and } \Delta F > 0.25\Delta C \\ 1 & \text{otherwise}\,, \end{cases} \tag{3.9}$$

when the merit function H has zero or a positive number as its lowest value, hence including the norms. In the rare cases where the merit function has a negative lowest value, we could use

$$\varphi_k = \begin{cases} \dfrac{1}{1+\frac{\Delta F}{1+|F(\mathbf{x}_k)|}} & \text{for } \Delta C > 0 \text{ and } \Delta F > 0.25\Delta C \\ 1 & \text{otherwise}\,, \end{cases} \tag{3.10}$$

which has about the same properties as (3.8).

The proposed formula (3.8) decreases ω by a factor two if there is negative or no decrease in F or if $C(\mathbf{p}_k(\mathbf{x}))$ is unable to predict decrease in F, and it decreases ω very little if there is a large decrease in F. The criterion of $\Delta C > 0$ is necessary because there is no guarantee for a positive gain in C for the imperfect mapping, as described earlier. Note that this updating scheme may not be the best possible formulation, but it fulfills the necessary basic properties discussed above.

We summarize the properties of the updating scheme:

- Only decrease in ω in such way that $\omega_k \to 0$ for $k \to \infty$.
- The maximum reduction of ω is a factor two per iteration.
- If ΔC is poor in predicting ΔF then ω is halved,
- otherwise ω is reduced according to ΔF, as explained above.

Note that the criterion of poor predictive ability is weighted higher than the criterion of positive gain in the function value. Hence, when the mapped coarse model is poor in predicting the decrease in F, ω is reduced maximally, even though there might be a large reduction in the function value.

The choice of maximum reduction rate, a factor two, is not fixed, it is chosen from a practical point of view. In chapter 4 we will see that test results suggest that decreasing ω_k at a higher rate, or perform a hard switch, gives better results for the simple problems considered. For problems of higher complexity other results might be obtained. We will leave further investigations to a later study, where models of higher complexity are available for numerical tests.

It might be interesting, during a later study, to allow the updating formula to increase ω during iterations, while still satisfying the condition of $\omega_k \to 0$ for $k \to \infty$. This could for instance be done by taking the ability of the linear approximation to predict the behaviour of the fine model into consideration. For that purpose the gain in the linear model, $\Delta L = L_k(\mathbf{x}_k) - L_k(\mathbf{x}_{k+1})$, and the ratio $\Delta F/\Delta L$ might be useful.

Hard switching

Hard switching could for instance be when ω_k is abrupt reduced from full weight on the mapped coarse model, $\omega_k = 1$, to full weight on the linear model, $\omega_k = 0$. But it could as well be a combination of this abrupt switch and a soft switching, for instance making a hard switch, when the algorithm detects that the coarse model is not accurate enough to be of any use. We name such an approach semi-hard switching.

Determining the stage at which the hard switch should occur is not that easy. Several measures are relevant. We could for instance look at the

- ratio $\Delta F/\Delta C$, as for the soft switching, the
- ratio $\Delta F/\Delta L$, the
- relative decrease in function value of S, or the
- value of ω, performing a hard switch when ω gets below some fixed threshold.

It has not been within the limits of this study to choose an exact way of performing the hard switch. Though, some simple conclusions are drawn in chapter 4 on the basis of results from some experiments with hard switching.

Whatever measure chosen, we should at least demand that $\omega_k = 0$ for $k \geqslant \hat{k}$, where $\hat{k} \leqslant k_{max}$, hence ensuring that the combined model equals the linear model before the algorithm is terminated by the criterion in (3.7).

3.2.3 The trust region and stopping criteria

As explained in section 2.3.2 we will avoid using

$$\Delta S = S_k(\mathbf{x}_k) - S_k(\mathbf{x}_{k+1}) \,, \tag{3.11}$$

the gain in the combined model as basis of an updating formula for the size of the trust region for the combined model. Instead we base the updating of the trust region size on the ability of the linear model to predict descent directions for the fine model. Hence we compare the measure

$$\Delta L = L_k(\mathbf{x}_k) - L_k(\mathbf{x}_{k+1}) \tag{3.12}$$

with ΔF, and we use the following updating formula of the trust region size

$$\delta_{k+1} = \begin{cases} 2\delta_k & \text{if } \Delta F > 0.75\Delta L \\ \delta_k/3 & \text{if } \Delta F < 0.25\Delta L \\ \delta_k & \text{otherwise} \,. \end{cases} \tag{3.13}$$

The formula works similar to (3.2): If the actual decrease in F exceeds 75% of the predicted decrease by L_k, the trust region is doubled in size. If the actual decrease in F is below 25% of the predicted decrease, the trust region is shrunken to a third of the actual size.

The stopping criteria proposed for the conventional Space Mapping algorithm are as well directly applicable for the new algorithm. Similarly using the gain in the combined model, we summarize the stopping criteria

- $k > k_{max}$,
- $\|\mathbf{h}_k\| < \varepsilon_1(1 + \|\mathbf{x}_k\|)$,
- $\Delta S < \varepsilon_2(1 + |S_k(\mathbf{x}_k)|)$.

As presented in this section 3.2, the implementation of the novel approach to the Space Mapping method is not much more difficult than the implementation of the conventional approach. We believe that the little extra effort required when making an actual implementation is fully compensated by the generalization of the method to be independent of the mapping being perfect or not. In the next chapter, we present results from tests using an implementation of the novel algorithm.

Chapter 4

Testing

In this chapter we present results from tests of the novel Space Mapping method. For the testing a Matlab implementation of the algorithm presented in section 3.2 is used. The chapter is divided in three sections, one for each example. In each section results from tests using the Space Mapping method on the actual example are presented and discussed, together with results from a direct optimization of the fine model.

Due to the relative small time scope of this project, it has not been possible to apply the Space Mapping method to expensive models. Instead we have applied the method to simple models with only a few free parameters. Thus we are leaving thorough testing of the novel Space Mapping method to a later study. Due to these limitations we are not able to draw conclusions on the performance of the Space Mapping method when applied to complicated models.

The three examples treated are,

- a transmission line transformer example with two free parameters (section 4.1),
- a piston simulator with two free parameters (section 4.2), and
- a one dimensional data fitting problem (section 4.3).

For all the examples we choose the infinity norm as merit function, hence $F(\mathbf{x}) = ||\mathbf{f}(\mathbf{x})||_\infty$, and we measure distances in the parameter spaces using the infinity norm as well. We use the minimization routine `MINCIN` described in [16], for solving the subproblems of minimizing the current surrogate and for evaluating the mapping function. The routine performs linearly constrained minimax optimization of a vector function, the actual method is described in [13].

The direct optimization, mentioned above, consists in sequential optimization of linear approximations to the fine model, as presented in (2.31). Hence the direct optimization algorithm iteratively calculates sets of parameters $\{\mathbf{x}_k^d\}$, $k = 0, 1, \dots$, in the effort of approximating the fine model optimum $\mathbf{x}^*$. The first linear approximation $\mathbf{l}_0$ is initialized with a finite difference approximation. The following approximations $\mathbf{l}_k$, $k > 0$, are formed from $\mathbf{l}_0$ by making corrections using the Broyden rank one update formula (2.27), as it is explained in section 2.3.1. The size of the trust region is updated by the procedure presented in section 3.2.3, hence updated according to the ability of L_k to predict decrease in F. However we have replaced the constants 2 and 3 in (3.13) by the constants 1.2 and 1.8, on the basis of observed improvements of performance.

4.1 Transmission Line Transformer

This example, provided by M.H. Bakr[1], concern the design of a 10:1 transmission line transformer (TLT). The physical background concerning the problem is not a topic of this thesis. We will consider the example as a given coarse model $\mathbf{d}$ and a given fine model $\mathbf{g}$, each with two designable parameters, $n = 2$, and each with a response of size $m = 11$. There are no design specifications, i.e. $\mathbf{y} = \mathbf{0}$, hence we have $\mathbf{c} = \mathbf{d}$ and similarly $\mathbf{f} = \mathbf{g}$. The main problem is to find the set of fine model parameters $\mathbf{x} = \mathbf{x}^*$ which minimizes $F(\mathbf{x})$ with regard to the functional constraint $F(\mathbf{x}) \leqslant 0.50$. We notice that this constraint makes the problem a global optimization problem. Furthermore we are informed that the physical nature of the TLT makes both models repetitive in their response for changes in parameter values of a certain magnitude. To avoid worries about the constraint and the repetitiveness, we ignore the global constraint, and instead constrain the parameters to a small confined area of the parameter space encompassing the solution.

The models provided are quite similar in their responses, as the contour

[1] M.H. Bakr is with McMaster University, Canada.

plots in figure 4.1 reveal. However, the optima are not the same, as seen in the figure, i.e. $\mathbf{z}^* \neq \mathbf{x}^*$.

Before test results are examined, we introduce the signatures that are used subsequently in the remainder of this chapter. In figure 4.2 the signatures for the iteration process plots are presented. The signatures should be interpreted as follows: From a given set of parameters $\mathbf{x}_k$ in the k'th iteration, represented by the leftmost asterisk $(*)$, the next set of parameters resulting from the k'th step, would be

- $\mathbf{x}^c_{k+1}$ represented by the triangle $(\triangle)$, if the mapped coarse model approximation is used (i.e. $\omega_k = 1$),
- $\mathbf{x}^l_{k+1}$ represented by the circle $(\circ)$, if the linear model is used (i.e. $\omega_k = 0$), or
- $\mathbf{x}_{k+1}$ represented by the rightmost asterisk $(*)$, if the combined model is used.

Since we want to examine the iteration process, we exclusively use the stopping criterion: $k < 35$, hence allowing 35 iteration steps. For this problem, the initial size of the trust region is set to $\delta_0 = 0.01 \cdot \|\mathbf{x}_0\|_\infty$. With this choice of δ_0 the coarse model optimum $\mathbf{z}^*$ is not encompassed by the trust region $\|\mathbf{x} - \mathbf{x}_0\| \leqslant \delta_0$, but recall that we are not bounding the first step of the Space Mapping algorithm by a trust region. Hence the specified trust region is in effect from the second iteration.

4.1.1 Soft switching

The first results presented are from a test of the Space Mapping algorithm using the smooth transition approach for updating ω. Figure 4.3 and figure 4.4 show the iteration process for both the Space Mapping algorithm and for a direct optimization using same starting point and initial size of the trust region. Since the first iteration is not bounded by a trust region, we have $\mathbf{x}_1 = \mathbf{z}^*$ for the Space Mapping method.

Figure 4.3 shows that both the Space Mapping and the direct optimization converges towards $\mathbf{x}^*$, the optimal solution, as both the distances $\|\mathbf{x}_k - \mathbf{x}^*\|_\infty$ and $\|\mathbf{x}^d_k - \mathbf{x}^*\|_\infty$ are approaching zero for $k \to \infty$.

Notice in figure 4.4 that the second and third step produced by the Space Mapping algorithm are in uphill directions for F. Furthermore we notice that the mapped coarse model approximation, if used exclusively, in the

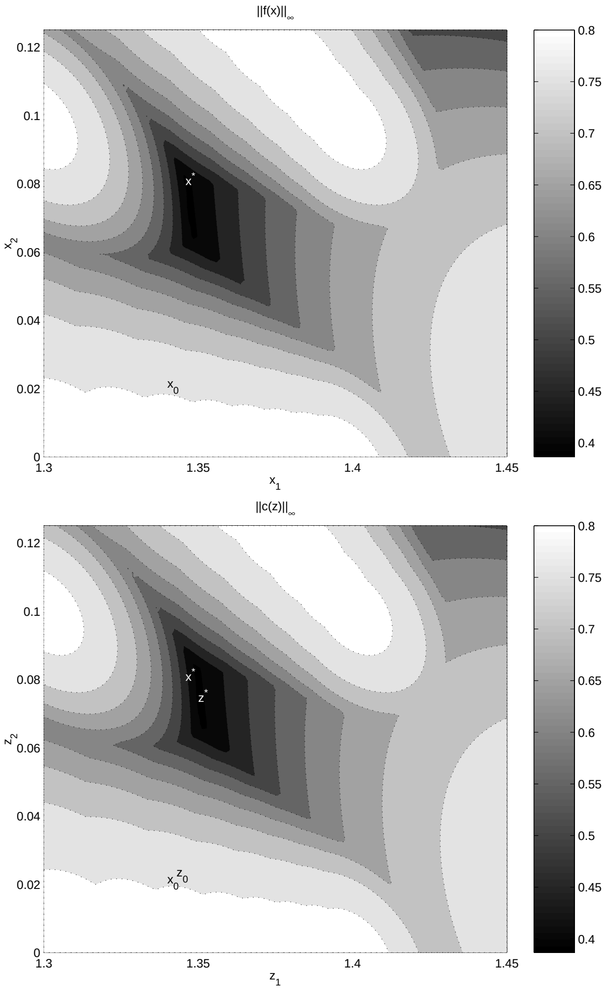

Figure 4.1: *Upper picture: Contour plot of the fine model, with marking of the initial set of parameters* $\mathbf{x}_0$*, and the optimal set of parameters* $\mathbf{x}^*$*. Lower picture: Contour plot of the coarse model, with marking of the initial fine model parameters* $\mathbf{x}_0$ *and the corresponding mapped set of parameters* $\mathbf{z}_0 = \mathbf{p}(\mathbf{x}_0)$*, as well as the two sets of optimal parameters* $\mathbf{x}^*$ *and* $\mathbf{z}^*$*. Both pictures: The visible parts of the parameter spaces reflects the chosen constraints upon the parameter values.*

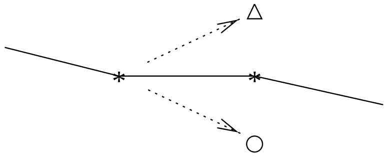

Figure 4.2: *The drawing presents the signatures for the iteration process plots, illustrated by a single iteration step. See text for details.*

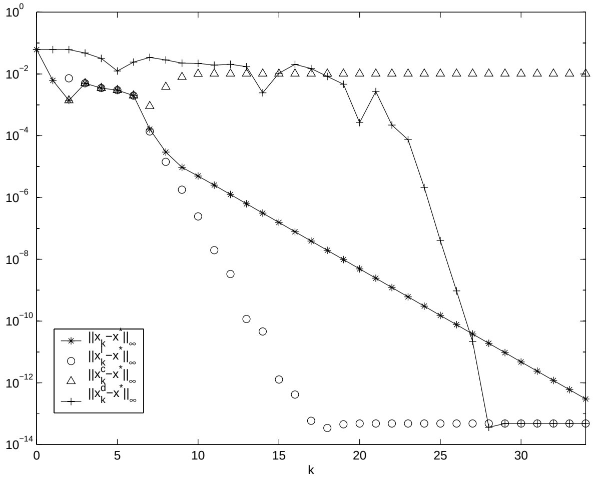

Figure 4.3: *Plot showing distance, in infinity norm, to* $\mathbf{x}^*$ *at each iteration step. The full line with plusses (+) represents* $\{\mathbf{x}_k^d\}$*, the sets of parameters resulting from a direct optimization. See text and figure 4.2 for the other signatures.*

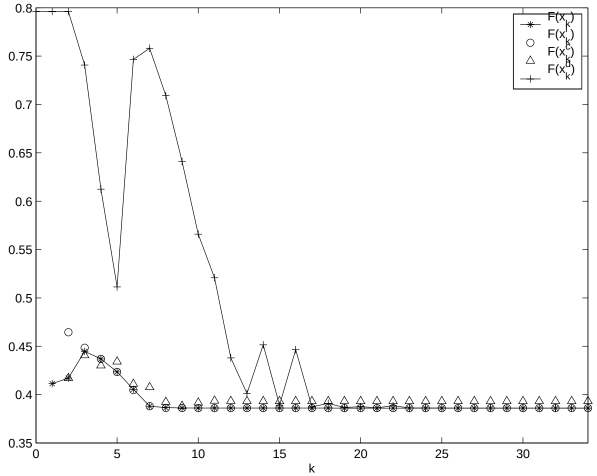

Figure 4.4: *Plot showing function value* F *at each iteration step.*

iterations $k \geqslant 5$ suggests steps $\mathbf{h}_k^c = \mathbf{x}_{k+1}^c - \mathbf{x}_k$ leading further away from $\mathbf{x}^*$ than the steps $\mathbf{h}_k^l = \mathbf{x}_{k+1}^l - \mathbf{x}_k$ suggested by the linear model: After the fourth iteration, the linear model promotes descent steps for F. In the iterations $k < 5$ the situation is opposite: The mapped coarse model approximation suggests better steps than the linear model.

We interpret the information derived from figures 4.3 and 4.4 in the following way: After initialization with gradient information from the mapped coarse model in the second iteration, the linear model requires two iterations to be corrected, before it is aligned enough with the fine model to predict descent steps for F. Further we deduce that the mapping is imperfect for this problem, since the mapped coarse model keeps on promoting steps leading away from the optimum, even when situated very close to the optimum. The uphill steps in iteration two and three suggest that we experience the situation sketched in figure 2.11 at page 32, where the algorithm produces uphill steps due to the mapping being imperfect, and due to the mapped coarse model approximation being weighted too much in the combined model.

The first two steps of the direct optimization results from a finite difference approximation. We see how the Space Mapping algorithm benefits from the knowledge of the coarse model optimum, by taking the first step directly to $\mathbf{z}^*$, in the near vicinity of $\mathbf{x}^*$. The direct method is bound to take smaller steps, for which reason it requires about 15 steps to get as close to $\mathbf{x}^*$ as the Space Mapping method gets by its first step.

In figure 4.5 the trust region size of the combined model is plotted together with the actual step size at each iteration step. From the figure it is seen, that the trust region is bounding the step size of the steps $\mathbf{h}_k$, $k = 2, \ldots, 5$. From the sixth iteration, the trust region keeps increasing in size, while the actual step size decreases.

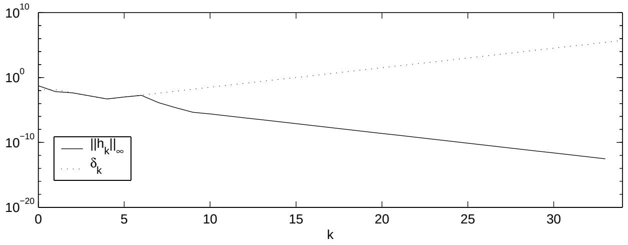

Figure 4.5: *Plot showing the step size and the actual size of the trust region at each iteration step for the combined model.*

In figure 4.6 the values of φ_k and ω_k are plotted for each iteration step, where

$$\varphi_k = \begin{cases} \frac{F(\mathbf{x}_{k+1})}{F(\mathbf{x}_k)} & \text{for } \Delta C > 0 \text{ and } \Delta F > 0.25\Delta C \\ 1 & \text{otherwise}\,, \end{cases}$$

and

$$\omega_{k+1} = \frac{\omega_k}{1 + \varphi_k}\,.$$

Notice that the value of φ_k is either very close or equal to one during all iterations, which implies that ω_k is approximately halved in each iteration, corresponding to the functional dependence

$$\omega_k \simeq \omega_0 \left(\frac{1}{2}\right)^{k-1} \quad \text{for } k > 0\,. \tag{4.1}$$

Comparing figure 4.6 with figure 4.5 and 4.3 it is evident how the exponential reduction of ω_k slows down the convergence of the Space Mapping method, as the systematic error introduced by the mapped coarse model is the dominating reason that the actual steps differ from steps directly to the optimum. Consequently, the distance from the actual parameters to the optimum is reduced at an exponential rate. A faster convergence might have been achieved by exclusively use of the linear model, as the linear model, beginning from iteration seven, promotes steps which leads to a higher rate of convergence, than the steps actually taken by the algorithm. We will examine this in detail in some of the next subsections, when dealing with the semi-hard and the hard switching approaches.

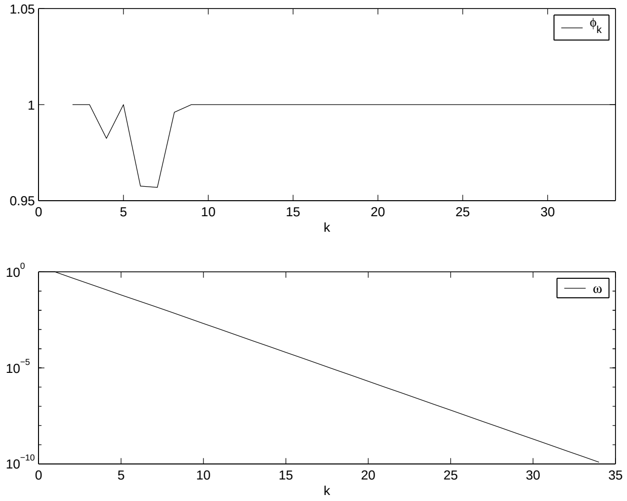

Figure 4.6: *Plots showing the values of φ_k and ω_k in each iteration step.*

The above observation is supported by figure 4.7, which shows the abilities in each iteration of the actual approximating models to predict the next fine model response, when taking the step $\mathbf{h}_k$. We measure this ability for respectively $\mathbf{d}(\mathbf{p}_k(\mathbf{x}))$ the mapped coarse model approximation, $\mathbf{l}_k$ the linear model and $\mathbf{v}_k$ the combined model by the following error measures,

$$H(\mathbf{g}(\mathbf{x}_{k+1}) - \mathbf{d}(\mathbf{p}_k(\mathbf{x}_{k+1})))\,, \tag{4.2a}$$

$$H(\mathbf{g}(\mathbf{x}_{k+1}) - \mathbf{l}_k(\mathbf{x}_{k+1}))\,, \tag{4.2b}$$

$$H(\mathbf{g}(\mathbf{x}_{k+1}) - \mathbf{v}_k(\mathbf{x}_{k+1}))\,, \tag{4.2c}$$

which we term the predicting error of the models. Notice that the lower the value of the predicting error, the better the ability of predicting the next fine model response.

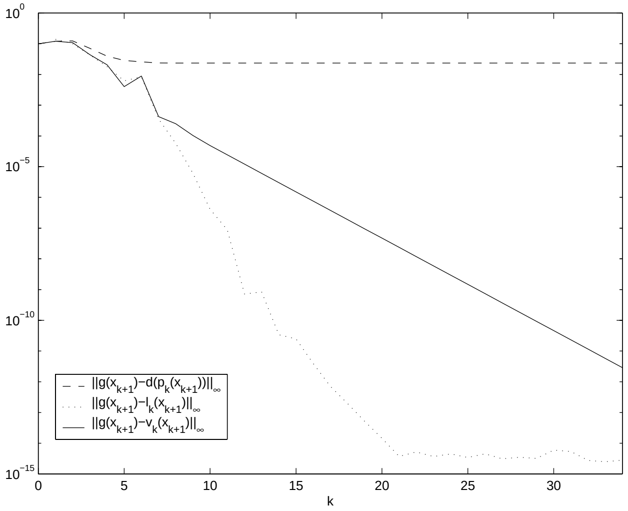

Figure 4.7: *Plot showing the predicting errors of respectively* $\mathbf{d}(\mathbf{p}_k(\mathbf{x}))$ *(−−),* $\mathbf{l}_k(\mathbf{x})$ *(···) and* $\mathbf{v}_k(\mathbf{x})$ *(—). The measures reflects the abilities in each iteration of the actual approximating models to predict the next fine model response.*

In the figure, it is seen how both $\mathbf{d}(\mathbf{p}_k(\mathbf{x}))$ and $\mathbf{l}_k$, after a couple of iterations, improve their ability of predicting the next fine model response, though $\mathbf{d}(\mathbf{p}_k(\mathbf{x}))$ stops its improvement at a certain error level. After about the fifth iteration, $\mathbf{d}(\mathbf{p}_k(\mathbf{x}))$ shows a nearly constant systematic error in predicting $\mathbf{g}(\mathbf{x}_{k+1})$. On the other hand, the linear model improves generally during all iterations. We notice how the prediction error of the combined model reflects the gradual transition from the mapped coarse model to the linear model. Observe that the increase in error of $\mathbf{l}_k$ and $\mathbf{v}_k$ from iteration five to iteration six, coincide with the reduced reduction rate of ω for these iterations.

Figure 4.8 shows the gain per iteration in $F(\mathbf{x})$, $C(\mathbf{p}_k(\mathbf{x}))$ and $S_k(\mathbf{x})$. We see how ΔF is negative for the first iterations due to the uphill steps, and how ΔC is positive in the same iteration. This indicates that there is a severe misalignment between the models. In iteration four ΔC is negative, implying that the mapped coarse model approximation predicts an increase in F for the step $\mathbf{h}_4$, even though there actually is a decrease in F. In the first iterations, the linear model is just as badly aligned with the fine model as the mapped coarse model is. From iteration three ΔL improves in alignment with ΔF.

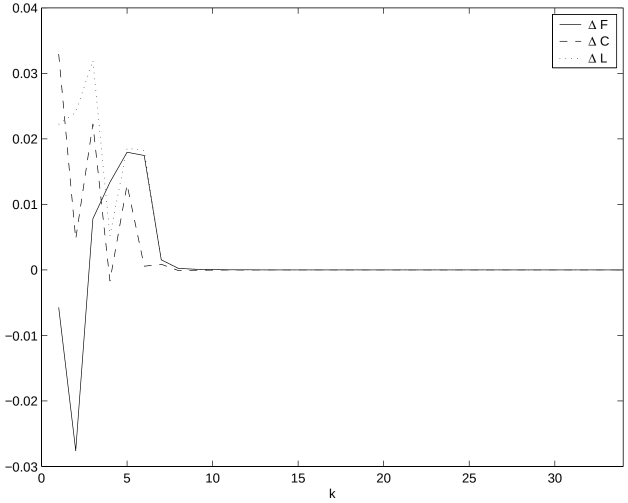

Figure 4.8: *Plot showing the gains* ΔF*,* ΔC*,* ΔL *in each iteration for respectively* $F(\mathbf{x})$*,* $C(\mathbf{p}_k(\mathbf{x}))$ *and* $L_k(\mathbf{x})$*, by taking the step* $\mathbf{h}_k$ *from* $\mathbf{x}_k$*.*

Figure 4.9 shows the ratios $\Delta F/\Delta S$ and $\Delta F/\Delta L$ for each iteration. The ratios reflect the ability of the models S_k and L_k to predict the change in F. As explained in section 3.2.3, the updating of the trust region size is based on the ratio $\Delta F/\Delta L$. Relating figure 4.8 with figure 4.5 we see how the trust region size is decreased in the first iterations where the ratio $\Delta F/\Delta L$ is negative (i.e. less than 0.25), and how the trust region size is increased when the ratio exceeds 0.75.

4.1.2 Convergence rate

As stated in the previous subsection, it is evident that the reduction of ω, even though it is exponential most of the time, is not fast enough to ensure that the approximation error of the mapped coarse model approximation

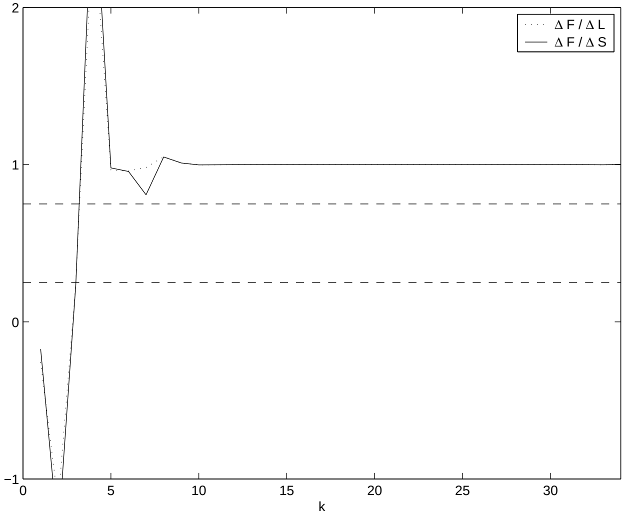

Figure 4.9: *Plot showing $\frac{\Delta F}{\Delta S}$ and $\frac{\Delta F}{\Delta L}$ for each iteration, the ratios reflect the models abilities to predict a change in F. The threshold values 0.25 and 0.75 are marked with dashed horizontal lines.*

is not influencing the final convergence rate of the Space Mapping method. We analyse the final convergence rate in the following way:

We define the deviation $\mathbf{e}_k$ of the actual parameters $\mathbf{x}_k$ to $\mathbf{x}^*$ as

$$\mathbf{e}_k = \mathbf{x}^* - \mathbf{x}_k \,. \tag{4.3}$$

Following [12] we define *linear convergence* as

$$\frac{||\mathbf{e}_{k+1}||}{||\mathbf{e}_k||} \leqslant \gamma \,, \tag{4.4}$$

where $\gamma < 1$ is a constant and $\mathbf{x}_k$ close to $\mathbf{x}^*$, and we define *super-linear convergence* as

$$\frac{||\mathbf{e}_{k+1}||}{||\mathbf{e}_k||} \rightarrow 0 \text{ for } k \rightarrow \infty \,. \tag{4.5}$$

In table 4.1 the actual parameter values of the iteration process described above are listed. The values of the error ratio in the rightmost column are plotted in figure 4.10. From the table, we see that the algorithm is converging towards $\mathbf{x}^*$ with a linear convergence rate of $\gamma \simeq 0.5$. Observe how the convergence rate corresponds to the exponential reduction rate in (4.1). We deduce that the higher value of the prediction error (4.2c) of the combined model compared to the prediction error of the linear model (4.2b) is imposed by the nearly constant systematic prediction error (4.2a) of the mapped coarse model.

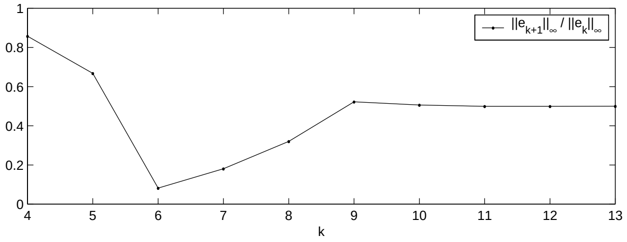

Figure 4.10: *Plot showing $\frac{||\mathbf{e}_{k+1}||}{||\mathbf{e}_k||}$ the ratio of the parameter approximation error. Values origin from table 4.1.*

We believe, that the only way to get the combined prediction error (4.2c) as low as the prediction error of the linear model, is to make a hard switch, i.e.

k	$x_k^{(1)}$	$x_k^{(2)}$	$\frac{\|\mathbf{e}_{k+1}\|_\infty}{\|\mathbf{e}_k\|_\infty}$
4	1.34601780914888	0.08465190727944	0.85760501635907
5	1.34552151285258	0.08415561098314	0.66792408877117
6	1.34476468713360	0.08316301839055	0.08178716742582
7	1.34591573154247	0.08132984256509	0.18074011302569
8	1.34588150284224	0.08119606982526	0.32050595464602
9	1.34589489041187	0.08117314760791	0.52246126727375
10	1.34589940736086	0.08117025141431	0.50637023522784
11	1.34590184681142	0.08116847426687	0.49961630419999
12	1.34590309897770	0.08116751376455	0.49968450889113
13	1.34590372449511	0.08116703482540	0.50000274299820
14	1.34590403685766	0.08116679626213	
...	...	...	
34	1.34590434922334	0.08116655770063	

Table 4.1: *Selected values from the iteration process of Space Mapping method using the soft switching approach.*

setting $\omega = 0$, after a number of iterations, since increasing the maximum reduction rate is not sufficient, if we intend to make the method generally applicable.

4.1.3 Semi-hard switching

The above discussion about final rate of convergence suggests a semi-hard switching approach, i.e. setting $\omega = 0$ somewhere in the final stage of the iteration process. In figure 4.11 the parameter distances from the optimum, resulting from experiments with semi-hard switching at $k_s = 1, \ldots, 8$, are plotted. However, the plots for $k_s = 4, 5$ are not present, because the plots for $k_s = 3, 4, 5, 6$ are visually identical. Since $\omega_k = 1$ for $k = 1, 2$ we have that the semi-hard switching is identical to hard switching in the first two iterations.

By visually comparing the plots in figure 4.11, we see that the highest rates of final convergence are for semi-hard switching at $k_s > 2$. For semi-hard switching at $k_s = 2$ the algorithm gets close to the optimum at an earlier stage in the iteration process, but the convergence rate is not as high. Further we notice that for $k_s > 6$ the combined model is disturbed, by the poor steps promoted by the mapped coarse model, before making the hard switch. Hence, judging from figure 4.11, the switch should be performed at $k_s = 3, 4, 5, 6$, for the present example, hence for ω in the interval $[0.03; 0.25]$. In a later study it might be interesting to experiment with a criterion causing a hard switch when ω gets below a certain threshold, for instance switching when $\omega < 0.1$.

We now focus on the case of semi-hard switching at $k_s = 3$. The actual parameter values for iteration step $k = 6, \ldots, 12$ are listed in table 4.2 and the measure of convergence rate, being the rightmost column of the table, is plotted in figure 4.12. In both the table and the figure we see that the rate of convergence is super-linear in the final stages of the iteration process. Hence, the semi-hard switching yields an significant improvement of the final rate of convergence for this problem. We would not expect a higher rate of convergence than superlinear, for a method only using approximations to the first order information of the problem.

k	$x_k^{(1)}$	$x_k^{(2)}$	$\frac{\|\mathbf{e}_{k+1}\|_\infty}{\|\mathbf{e}_k\|_\infty}$
6	1.34480217728959	0.08316301839055	0.06784063007082
7	1.34596220933005	0.08130199885149	0.11799000438378
8	1.34590031983539	0.08118253840237	0.02058810609003
9	1.34590461445012	0.08116622868797	0.01659136363124
10	1.34590434577409	0.08116656315912	0.00395930128719
11	1.34590434921049	0.08116655772197	0.00279958622629
12	1.34590434922363	0.08116655770042	
...	...	...	
34	1.34590434922364	0.08116655770041	

Table 4.2: *Selected values from the iteration process for the novel Space Mapping method, using semi-hard switching at $k_s = 3$.*

4.1.4 Hard switching

In this subsection we consider the hard switching approach, i.e. having $\omega_k = 1$ for $k < k_s$ and $\omega_k = 0$ for $k \geqslant k_s$, k_s being the iteration step at which the hard switch occur. In figure 4.11 we saw what happened if the hard switch where performed at $k = 1, 2$. In figure 4.13 the iteration

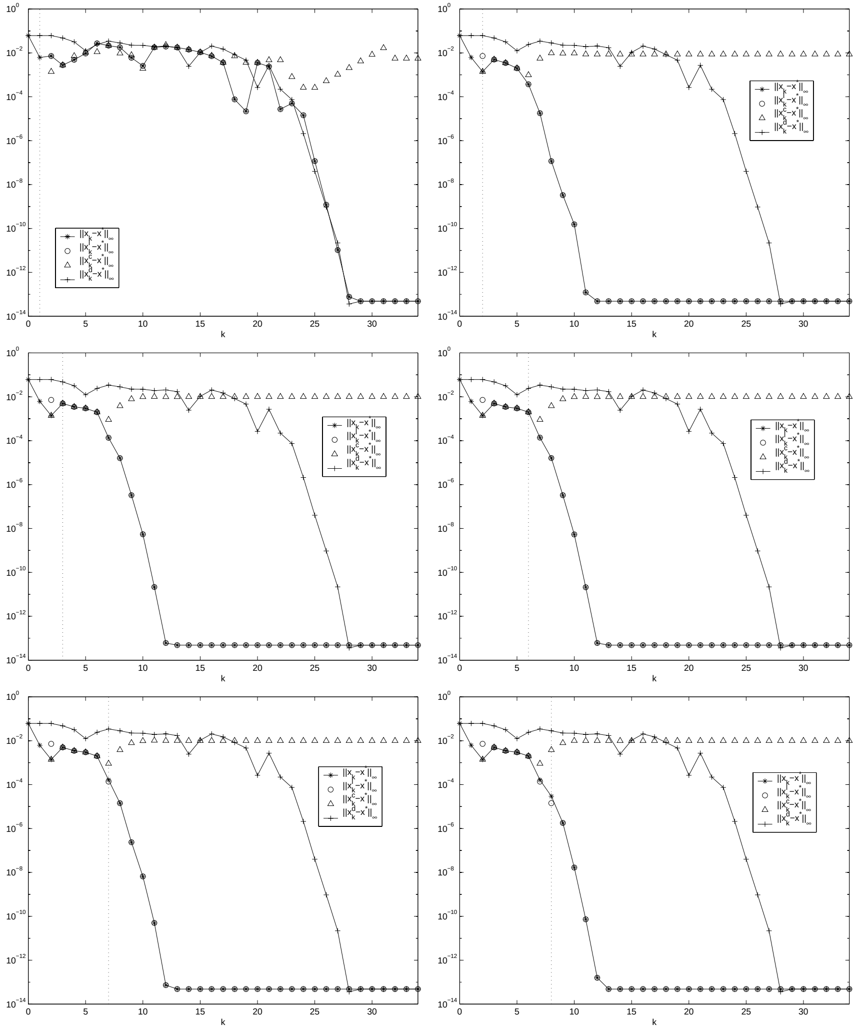

Figure 4.11: *The plots show the iteration processes for the Space Mapping method using semi-hard switching at respectively $k_s = 1, 2, 3, 6, 7, 8$. The vertical dotted line in each plot, marks the iteration at which the hard switch occur.*

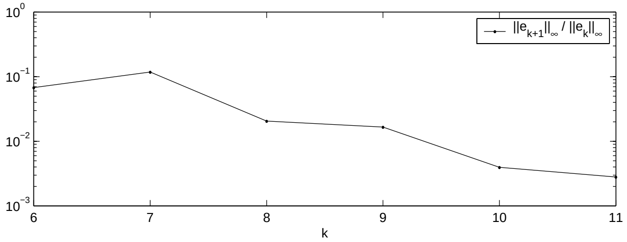

Figure 4.12: *Plot showing $\frac{\|\mathbf{e}_{k+1}\|}{\|\mathbf{e}_k\|}$ the ratio of the parameter approximation error for selected iteration steps, when using semi-hard switching at $k_s = 3$.*

processes resulting from experiments with hard switching at $k_s = 1, \ldots, 8$ are plotted.

From the plots we see how the solution is located for all displayed choices of k_s. However, for $k_s = 3, 6$ the algorithm escapes the minimizer again, most likely due to a combination of rounding errors influencing on the Broyden update and a too large trust region. Judging from the figures there are no advantages of the hard switching approach over the semi-hard switching approach. In fact, the convergence in the final stages seems, by visual estimate, to be more disturbed than what we saw in figure 4.11 for the semi-hard switching. Thus, from these experiments, we would recommend using the semi-hard switching approach rather than the hard switching or the soft switching approaches.

4.1.5 Conventional Space Mapping Solution

As stated in section 4.1.1 the mapping is imperfect for the present problem. In figure 4.14 the approximate location of the conventional Space Mapping solution $\mathbf{x}^*_{SM}$ is marked.

We have located the aproximative conventional Space Mapping solution by solving the problem

$$\mathbf{x}^*_{SM} = \underset{\mathbf{x}}{\operatorname{argmin}} \, \|\mathbf{d}(\mathbf{z}^*) - \mathbf{g}(\mathbf{x})\| \, . \tag{4.6}$$

We denote the numerical solution to (4.6) $\mathbf{x}_k$. In figure 4.14 we observe how $\mathbf{x}_k$ does not map the exact coarse model optimum, $\mathbf{p}(\mathbf{x}_k) \neq \mathbf{z}^*$. We have

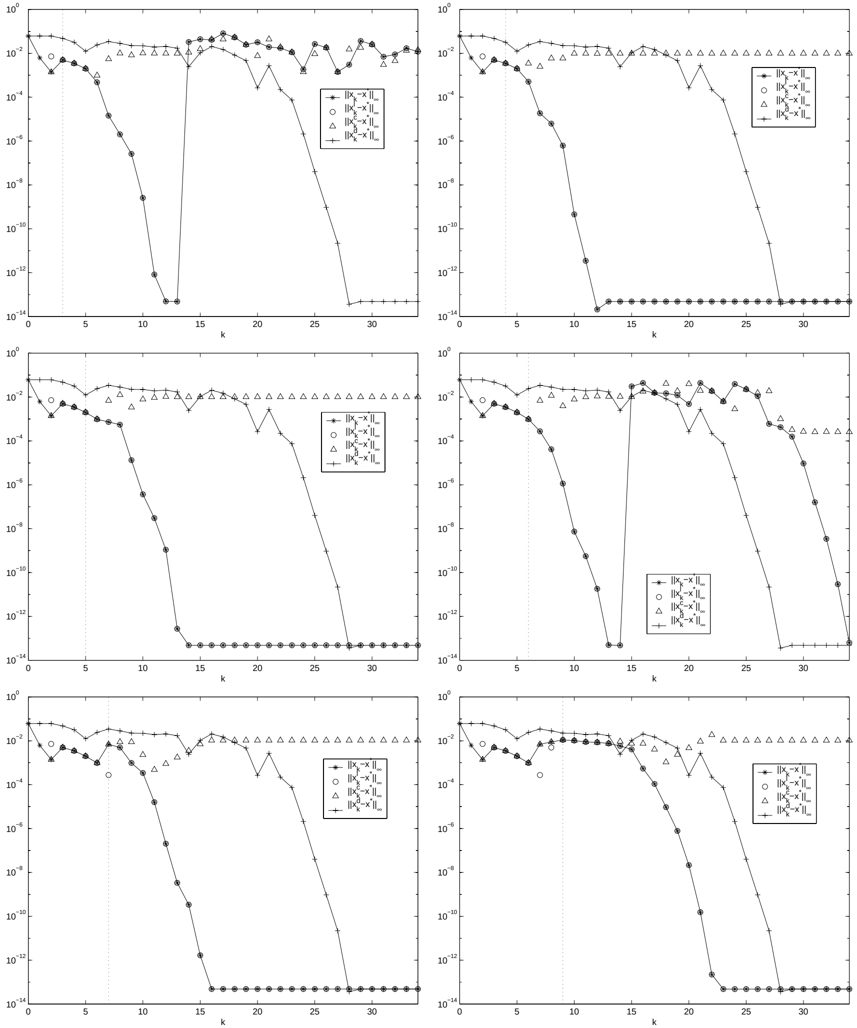

Figure 4.13: *The plots show the iteration processes for the Space Mapping method using hard switching at respectively $k_s = 3, 4, 5, 6, 7, 9$. The vertical dotted line in each plot, marks the iteration at which the hard switch occur.*

examined the problem (4.6) by contour plots and found that it actually is a well behaved problem with a regular solution. The phenomenon observed might be due to the mapping function having non-differentiable points in the vicinity of $\mathbf{x}^*_{SM}$. It has not been within the limits of this project to examine this phenomenon in depth. We refer to the discussion about non-differentiable points in the mapping function in section 4.3.

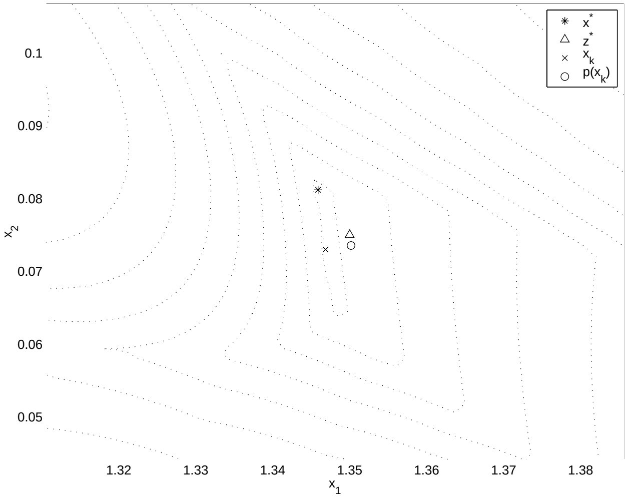

Figure 4.14: *Contour plot of the fine model, at which the location of the $\mathbf{x}_k \simeq \mathbf{x}^*_{SM}$, $\mathbf{z}_k = \mathbf{p}(\mathbf{x}_k)$, $\mathbf{x}^*$ and $\mathbf{z}^*$ are marked. Notice that $\mathbf{z}_k \neq \mathbf{z}^*$ implying that $\mathbf{x}_k$ is not the exact conventional Space Mapping solution $\mathbf{x}^*_{SM}$.*

As the final exploration of the TLT example, figure 4.15 shows results from the iteration process of a test where the direct method is started in $\mathbf{z}^*$, and where the Space Mapping uses a semi-hard switch at $k_s = 2$. In the figure we see how the direct method has the advantage of not being influenced by the mapped coarse model, as the Space Mapping method is. This indicates to us how much the choice of starting point influences on the convergence results.

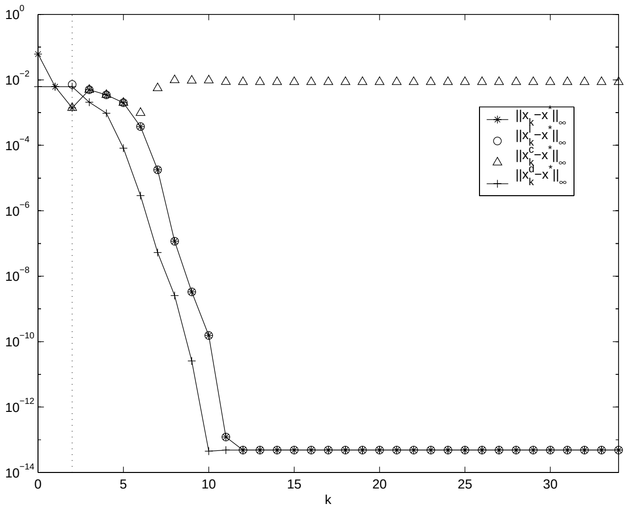

Figure 4.15: *Plot showing the iteration process when starting the direct optimization in the coarse optimum, and setting $\omega = 0$ at $k = 2$.*

4.1.6 Summarizing the TLT example

We now summarize the main conclusions to be drawn from the TLT example.

During the investigations, we have experienced how the choice of transition approach in the combined model influences on the convergence properties of the novel Space Mapping method. Judging from these simple tests, the semi-hard switching approach seems to outperform both the soft switching and the hard switching approaches, when considering the final rate of convergence and the overall performance of the method.

The example showed how an imperfect mapping can lead to situations, where the mapped coarse model promotes steps in uphill directions. Consequently a systematic error is introduced to the combined model for $\omega \neq 0$, and we showed how this error influences the final rate of convergence, being linear for the case of soft switching.

4.2 Piston

The second example, the piston model, is provided by P.E. Frandsen[2]. The model calculates the pressure over time at an oil producing well relative to a fixed injection pressure. The physical background concerning the model is not a topic of this thesis. The piston model is providing us with a fine and a coarse model each with two free parameters. We wish to fit, in the minimax sense, the fine model to a given history of oil pressures from a producing well. The history is represented by the points $\{t^{(i)}, y^{(i)}\}$, $i = 1, \ldots, 20$, originating from an evaluation of the fine model. Hence at the optimal solution $\mathbf{x}^*$, the fine model residuals to the data points are zeros, $\mathbf{f}(\mathbf{x}^*) = \mathbf{0}$. Thus the mapping is perfect for this problem, since $\hat{\mathbf{z}} = \mathbf{z}^*$ minimizes

$$\begin{aligned}\|\mathbf{r}(\mathbf{x}^*, \hat{\mathbf{z}})\| &= \|\mathbf{g}(\mathbf{x}^*) - \mathbf{d}(\hat{\mathbf{z}})\| \\ &= \|(\mathbf{g}(\mathbf{x}) - \mathbf{y}) - (\mathbf{d}(\hat{\mathbf{z}}) - \mathbf{y})\| \\ &= \|\mathbf{f}(\mathbf{x}) - \mathbf{c}(\hat{\mathbf{z}})\| \\ &= \|\mathbf{c}(\hat{\mathbf{z}})\| .\end{aligned}$$

In figure 4.16 results from the iteration process, using the novel Space Mapping method with smooth switching, are plotted together with results from a direct optimization. From the figure it is seen how both methods converge towards the optimum. We observe the Space Mapping algorithm shows a fluttering behaviour, jumping in and out of the vicinity of the solution, and that the methods do not locate the exactly same solution. The fluttering behaviour might be caused by a combination of rounding errors influencing the Broyden update and too large a trust region. However, this cannot be the complete explanation, since these conditions are present for all the iteration processes presented. We will not go further into this in the present study.

In figure 4.17 the distances from the current parameters $\mathbf{x}_k$ respectively the mapped parameters $\mathbf{z}_k = \mathbf{p}(\mathbf{x}_k)$ to the coarse model optimum $\mathbf{z}^*$ are plotted. From the figure it is seen that $\mathbf{x}_1 = \mathbf{z}^*$, and that the mapped parameters $\mathbf{z}_k$ converges towards the coarse model optimum, as we would expect in the case of a perfect mapping. But we observe that the mapped parameters show the same fluttering behaviour, as we observed with the fine model parameters in figure 4.16.

[2] P.E. Frandsen is co-supervising this project.

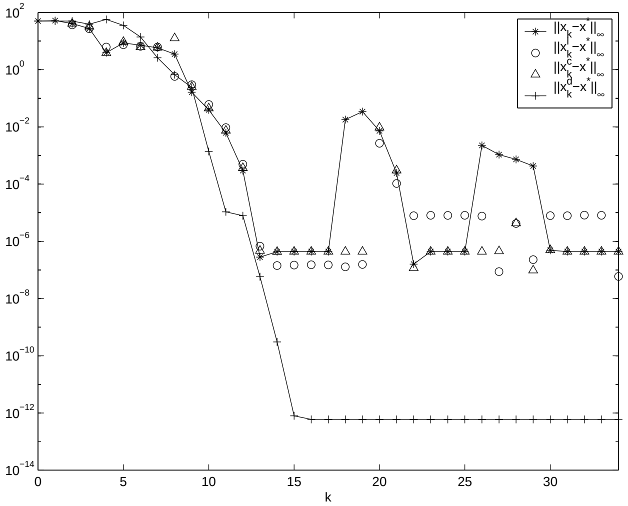

Figure 4.16: *Plot showing distance from the current parameters to the optimum* $\mathbf{x}^*$*. The signatures corresponds to the signatures presented by figure 4.2.*

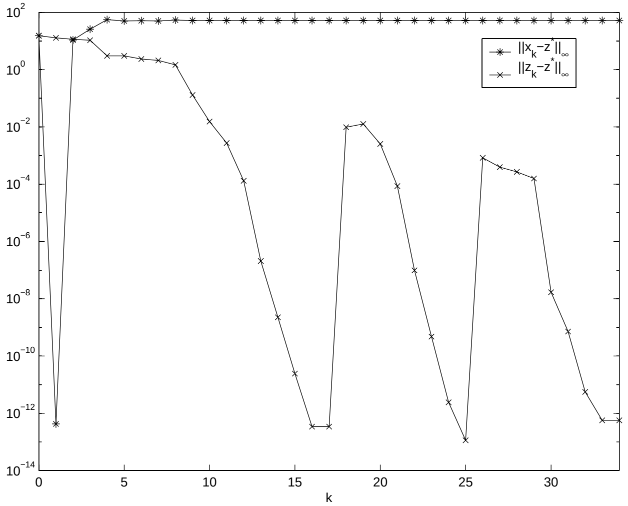

Figure 4.17: *Plot showing the distances from respectively* $\mathbf{x}_k$ *and* $\mathbf{z}_k$ *to the coarse model optimum* $\mathbf{z}^*$*.*

In figure 4.18 the gain in function value of respectively $F(\mathbf{x})$, $C(\mathbf{p}_k(\mathbf{x}))$ and $S_k(\mathbf{x})$ are plotted for the first ten iteration steps. The figure shows that the models generally are well aligned. We observe that F increases in value at iteration $k = 7$. Relating this observation to figure 4.16, it is evident that the uphill step is caused by the influence of the mapped coarse model in the combined model. The mapped coarse model promotes a step leading away from $\mathbf{x}^*$, while the linear model promotes a step leading towards $\mathbf{x}^*$. The actual step is a combination of the two steps promoted by the models.

In figure 4.19 the values of φ and ω are shown for each iteration step. From the figure it is seen how φ is almost zero at some stages in the iteration process, and that the values are changing more during the process than for the TLT example. Thus the exponential decay of ω is less than $1/2$ as observed in the TLT example.

In figure 4.20 we see the predicting errors of the approximating models. As for the TLT example, the mapped coarse model shows a nearly constant systematic error in predicting the fine model response, even though it is producing descent steps for F. This observation shows, that for the general Space Mapping problem the coarse model is allowed to show systematic

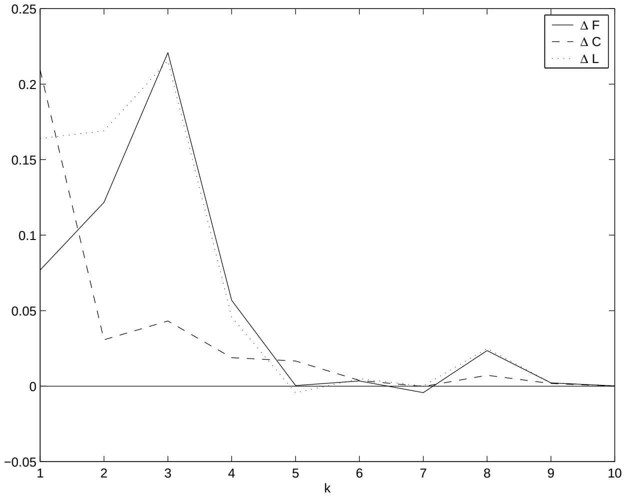

Figure 4.18: *Plot showing the gains ΔF, ΔC, ΔL for a selected interval of iteration steps.*

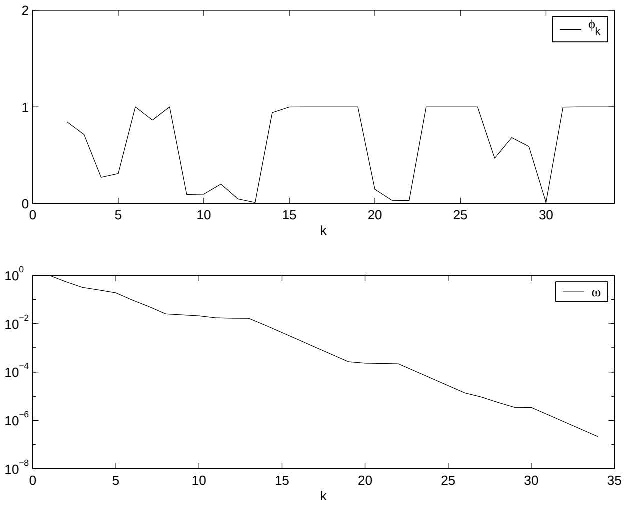

Figure 4.19: *Plots showing the values of φ_k and ω_k in each iteration step.*

errors in predicting the actual response, as long as it is performing well in predicting descent steps for the objective function, as we would expect. Notice that this would not be possible without the parameter mapping. The parameter mapping is the strength of the Space Mapping method comparing to other surrogate optimization methods, as e.g. [10].

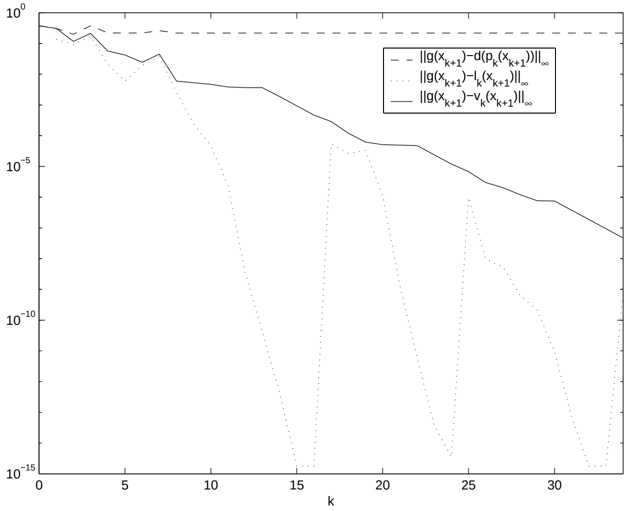

Figure 4.20: *Plot showing the predicting errors of respectively* $\mathbf{d}(\mathbf{p}_k(\mathbf{x}))$, $\mathbf{l}_k(\mathbf{x})$ *and* $\mathbf{v}_k(\mathbf{x})$. *The measures reflects the abilities in each iteration of the actual approximating models to predict the next fine model response.*

The piston example has been tested with the different approaches of performing the transition in the combined model. Here, we present results from a single test. In figure 4.21 results from a iteration process using the Space Mapping method with semi-hard switching at $k_s = 2$ are plotted. From the figure we see that there is no change in the convergence towards the solution. However the Space Mapping algorithm is not fluttering around the vicinity of the solution, as for the soft switching approach.

In this example the coarse model optimum $\mathbf{z}^*$ is placed farther away from $\mathbf{x}^*$ than the set of starting parameters $\mathbf{x}_0$ is. By tests it shows out that the direct method has no advantage of starting in the coarse model optimum, as it had in the TLT example — in fact more steps are required to reach $\mathbf{x}^*$. We cannot draw any conclusions from this observation, based on only

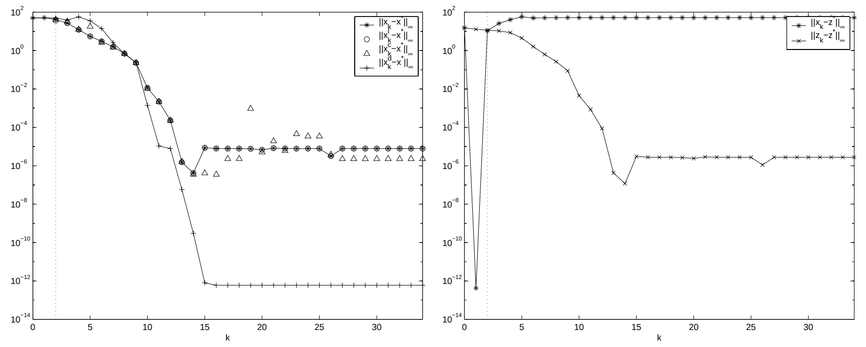

Figure 4.21: *Left figure: Plot showing distances from the actual parameters to the optimum, the novel Space Mapping method uses semi-hard switching at $k_s = 2$. The vertical dotted line, marks the iteration at which the hard switch occur. Right figure: Plot showing the distances of respectively $\mathbf{x}_k$ and $\mathbf{z}_k$ to the coarse model optimum $\mathbf{z}^*$.*

two examples. However we believe that for many problems we will have that $\mathbf{z}^*$ is not in the vicinity of $\mathbf{x}^*$, and therefore it would be natural if there in general were no benefit of starting a direct optimization in $\mathbf{z}^*$.

4.3 Data fitting

The last example presented in this report is a one dimensional data fitting problem. The example is used to plot complete mapping functions for different choices of merit functions.

We consider a set of data points $\{t^{(i)}, y^{(i)}\}$, $i = 1, \dots, 200$. The data points originate from an experiment where the light intensity from a laser source were measured in the small interval of time, after switching off the laser source, until the light dies out. We wish to approximate the data with the fine model:

$$g(t^{(i)}; x) = q_2 \exp(-x q_1 t^{(i)}) + q_3, \ q_j \in \mathbb{R}, \ j = 1, 2, 3 \,,$$

where q_j are constants. The objective function is $\mathbf{f}(\mathbf{t}; x) = [f(t^{(i)}; x)]$, where $f(t^{(i)}; x) = g(t^{(i)}; x) - y^{(i)}$. We have a coarse model which approximates

the same set of data:

$$d(t^{(i)}; z) = z w_1 (t^{(i)})^2 + w_2 t^{(i)} + w_3, \ w_j \in \mathbb{R}, \ j = 1, 2, 3 \,,$$

where w_j are constants. The objective function is here $\mathbf{c}(\mathbf{t}; z) = [c(t^{(i)}; z)]$, where $c(t^{(i)}; z) = d_i(t^{(i)}; z) - y^{(i)}$. In figure 4.22 the optimal minimax designs of both the coarse and the fine model are shown together with the set of data points.

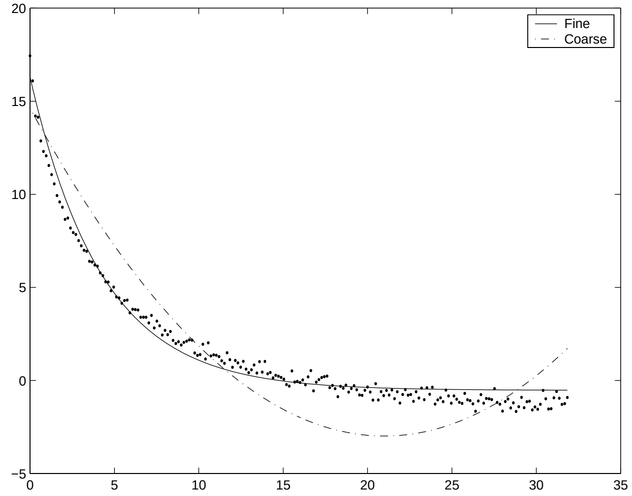

Figure 4.22: *Data points and optimal designs of the two models, optimized in the minimax sense.*

In figure 4.23 we have plotted the complete mapping function of the problem, using the infinity norm as merit function for the mapping function. In this figure, we see how the mapping function for the most part shows an exponential relation between the parameters. In figure 4.24, where we have a closer view at the vicinity around the point of the optimal solutions, we see that the point of the optimal solutions is not placed on the mapping function.

The horizontal distance between the point of optimal solutions and the mapping function is

$$\|\mathbf{x}^* - \mathbf{x}^*_{SM}\|_\infty \,.$$

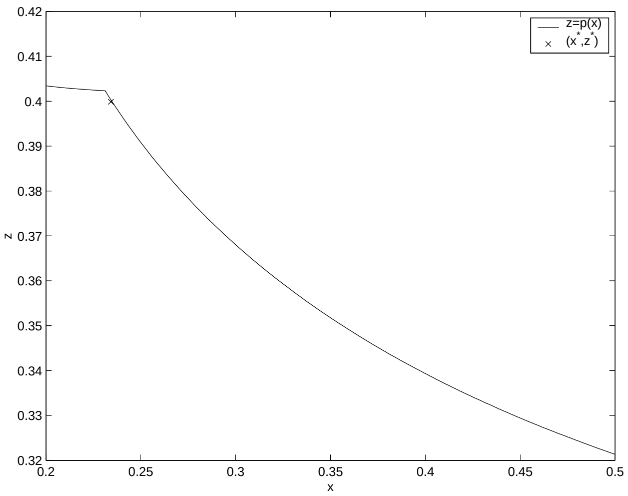

Figure 4.23: *Plot showing the complete mapping function, using the infinity norm as merit function, for the one dimensional data fitting problem. The point of the optimal solutions (x^*, z^*) is marked.*

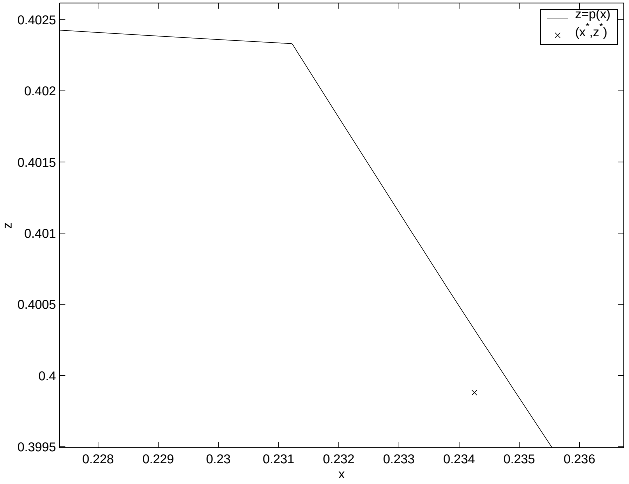

Figure 4.24: *Plot showing close up of the complete mapping function, using the infinity norm as merit function.*

The vertical distance between the point of optimal solutions and the mapping function is

$$\|\mathbf{z}^* - \hat{\mathbf{z}}\|_\infty \ ,$$

where $\hat{z}$ is the set of coarse parameters which yields the best fit of the coarse model response to the fine model response evaluated in $\mathbf{x}^*$.

Observe the bend in the mapping function, implying that the mapping function is non-differentiable at this point. It is evident that if we linearize the mapping function in the vicinity of the bend, the linearization error becomes enormous, when the linearization is used past the bend. The bend occurs due to the use of the non-differentiable infinity norm as merit function. In figure 4.25 the mapping function for the same problem, using the two norm as merit function, is plotted. From the figure it is seen that the mapping function is differentiable in the interval shown, the vicinity of the optimum.

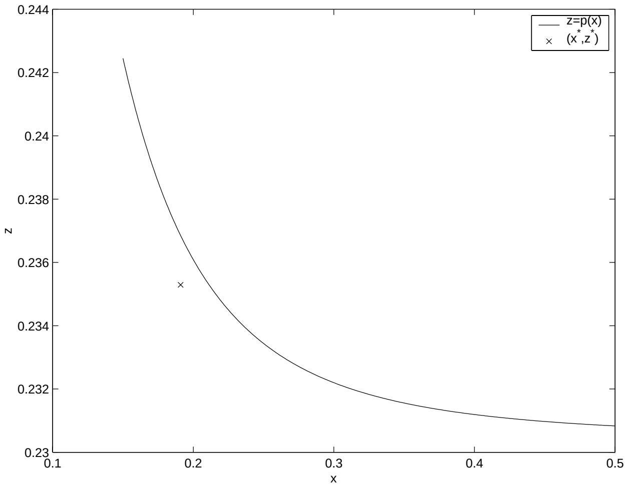

Figure 4.25: *Plot showing a close up of the complete mapping function, using the two norm as merit function.*

The above observations with a non-differentiable point in the mapping function, suggests that caution should be taken when interpreting the results from an iteration process. Non-differentiable points in the mapping function might cause fluttering in the parameter values, due to the large linearization errors introduced in the vicinity of the non-differentiable points. This is a natural consequence of violating the assumption of a differentiability, when employing a gradient based method using sequential linearizations to approximate the considered function.

Chapter 5

Conclusion

In the following the results and conclusions of the project are summarized. The material presented mostly consists in theoretical work, a smaller part consists in numerical verification and investigation of the Space Mapping method.

The theoretical work can be divided into three subjects: the conventional Space Mapping method, the novel Space Mapping method and aspects about implementation of the two methods.

We now draw up the conclusions concerning the theoretical work and the numerical testing.

Conventional Space Mapping

The concept of a mapping function, establishing the parameter mapping between the coarse and the fine model, has been described in detail. From this, a novel, concise formulation of the conventional Space Mapping problem has been derived, showing how the coarse model, by utilizing parameter mapping, can act as a surrogate to the main problem — provided the parameter mapping is injective for the areas of the parameter spaces considered.

Previous formulations of the conventional Space Mapping problem have been presented. Iterative formulations, exploiting trust region methodology and approximative linearizations of the mapping function, for both

the novel and a previous formulations of the conventional Space Mapping problem, have been derived and presented.

The exact relations between the novel and the previous iterative formulations have been described, and the previous iterative formulation has been illustrated by geometrical considerations.

The conventional Space Mapping has been proven inadequate for solving general problems, where the mapping function, relating the coarse and the fine model, is so-called imperfect.

Novel Space Mapping approach

A novel approach to the Space Mapping problem has been presented, linking the conventional Space Mapping method with a direct method in a combined model, capable of acting as a surrogate for the main problem. The combined model consists of weigthed contributions from the coarse model using parameter mapping, and a local linear approximation to the fine model. The use of the linear approximation has been prooven to make the Space Mapping generally applicable, no matter if the mapping function is perfect or imperfect — provided that the weight in the combined model is pushed towards the linear approximation, as the iterations progresses, such that the combined model equals the local approximation in the final stages of the iteration process.

Implementation aspects

Different aspects of implementing the iterative formulations, of respectively the conventional and the novel Space Mapping formulation in iterative algorithms, have been discussed. Among these aspects three approaches to performing the transition between approximating models in the combined model have been discussed. These are soft, semi-hard and hard switching.

The updating of the trust region has been discused for both formulations, and it has been proposed to use different updating approaches for the conventional and the novel Space Mapping method.

Testing

Three examples have been presented, and tested using the the novel Space Mapping method. The investigations have shown that the semi-hard approach to transition is superior to the soft transistion approach, when considering final rate of convergence.

The effect of the problem having an imperfect mapping has been investigated as well. The coarse model, using parameter mapping, have been found to promote poor steps for the algorithm, for the case of an imperfect mapping.

Further investigations have shown how the parameter mapping enables the use of coarse models showing severe approximation errors to the fine model, while still contributing positively to the convergence of the algorithm.

For further work with the Space Mapping method the following items are suggested:

- Testing on larger problems.
- Further work on transition strategies.
- Working with models not having same dimensionality.
- Working with ensuring uniqueness of the mapping function, avoiding more than one solution to the subproblem of mapping the parameter spaces.
- Working with constrained problems.
- Further exploration of non-differential points in the mapping function.

Bibliography

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

[2] M. Bakr, J. Bandler, R. Biernacki, S. Chen, K. Madsen
A Trust Region Aggressive Space Mapping Algorithm for EM Optimization
IEEE Transactions on Microwave Theory and Techniques, vol. 46, no. 12, December 1998.

[3] M. Bakr, J. Bandler, R. Biernacki, S. Chen, K. Madsen
The Trust Region Aggressive Space Mapping Technique: Theory, Implementation and Examples
Simulation Optimization Systems Research Laboratory
and Department of Electrical and Computer Engineering
McMaster University, Hamilton, Canada L8S 4K1, April 1998.

[4] M. Bakr, J. Bandler, N. Georgieva
An Aggressive Approach to Parameter Extraction
Simulation Optimization Systems Research Laboratory
and Department of Electrical and Computer Engineering
McMaster University, Hamilton, Canada L8S 4K1.
Presented at IEEE Int. Microwave Symposium, Anaheim, CA, June 1999.

[5] M. Bakr, J. Bandler, N. Georgieva, K. Madsen
A Hybrid Aggressive Space Mapping Algorithm for EM Optimization
Simulation Optimization Systems Research Laboratory
and Department of Electrical and Computer Engineering
McMaster University, Hamilton, Canada L8S 4K1.
Presented at IEEE Int. Microwave Symposium, Anaheim, CA, June 1999.

[6] J. Bandler, R. Biernacki, S. Chen, R. Hemmers, K. Madsen
Electromagnetic Optimization Exploiting Aggressive Space Mapping
IEEE Transactions on Microwave Theory and Techniques, vol. 43, no. 12, December 1995.

[7] J. Bandler, R. Biernacki, S. Chen, R. Hemmers, K. Madsen
Space Mapping Optimization For Engineering Design
Simulation Optimization Systems Research Laboratory
and Department of Electrical and Computer Engineering
McMaster University, Hamilton, Canada L8S 4K1, 1995.

[8] J. Bandler, M. Ismail, J. Rayas-Sánchez and Q. Zhang
A Generalized Mapping Tableau Approach to Device Modeling
Simulation Optimization Systems Research Laboratory
and Department of Electrical and Computer Engineering
McMaster University, Hamilton, Canada L8S 4K1
To be presented at the 29th European Microwave Conference, Munich, Germany, October 1999.

[9] J. Bandler, M. Ismail, J. Rayas-Sánchez and Q. Zhang
Neuromodeling of Microwave Circuits Exploiting Space Mapping Technology
Simulation Optimization Systems Research Laboratory
and Department of Electrical and Computer Engineering
McMaster University, Hamilton, Canada L8S 4K1
Presented at IEEE Int. Microwave Symposium, Anaheim, CA, June 1999.

[10] A. Booker, J. Dennis, P. Frank, D. Serafini, V. Torczon, M. Trosset
A Rigorous Framework for Optimization of Expensive Functions by Surrogates
Structural Optimization, 17 (1), 1999.

[11] C. Cassandras, W. Gong
Metamodelling Techniques in Multidimensional Optimality Analysis for Linear Programming
Mathematical and Computer Modelling
Vol. 23 (5) p. 45-60, 1996.

[12] P. Frandsen, K. Jonasson, H. Nielsen, O. Tingleff
Unconstrained Optimization
J. No. H69, IMM, DTU, 1999.

[13] J. Hald, K. Madsen
Combined LP and Quasi-Newton methods for Minimax Optimization
Mathematical Programming, vol. 20, pp. 49-62, 1981.

[14] K. Madsen
An Algorithm for Minimax Solution of Overdetermined Systems of Non-Linear Equations
J. Inst. Math. Appl. 16, p. 321-328, 1975.

[15] K. Madsen, H. Nielsen, O. Tingleff
Methods for Non-Linear Least Squares Problems
J. No. H38, IMM, DTU, 1999.

[16] K. Madsen, O. Tingleff, P.C. Hansen, W. Owczarz
Robust Subroutines for Non-Linear Optimization
Report No. NI-90-06, Institute for Numerical Analysis, Technical University of Denmark, 1990

[17] J. Moré
Recent Developments in Algorithms and Software for Trust Region Methods
In "Mathematical Programming - The State of the Art, Bonn 1982", ed. by A. Bachem, M. Grötschel, B. Korte, p. 258-87, 1982.

[18] V. Torczon, M. Trosset
Using approximations to accelerate engineering design optimization
Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization
AIAA Paper 98-4800, 1998.