Multi-Level Optimization: Space Mapping and Manifold Mapping

Multi-Level Optimization: Space Mapping and Manifold Mapping

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. J.W. Zwemmer
ten overstaan van een door
het college voor promoties ingestelde commissie,
in het openbaar te verdedigen in de Aula der Universiteit
op donderdag 29 maart 2007, te 12:00 uur

door

David Echeverría Ciaurri

geboren te Estella/Lizarra (Spanje)

Promotiecommissie:

Promotor:	prof. dr. P. W. Hemker
Overige leden:	prof. dr. A. Doelman
	dr. W. Hoffmann
	prof. dr. W.H.A. Schilders
	prof. dr. Ph.L. Toint
	prof. dr. ir. A.J.A. Vandenput
	prof. dr. J.G. Verwer

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

THOMAS STIELTJES INSTITUTE
FOR MATHEMATICS

Este trabajo
está dedicado
a mi familia

Acknowledgment

This research would not have been possible without the guidance of Piet Hemker. It has been a real privilege to work with him. I also want to express my gratitude to Barry Koren for his always kind leading way, too. And to my most stable office mate and colleague, Domenico Lahaye for his unbeatable tiramisu, for all the dinners together at Sa Seada and most importantly, for his friendship.

My appreciation to Walter Hoffmann for his wise Thursday's meeting feedback, to my Eindhoven workfellow Laurentiu Encica (ne vedem!) for the electromagnetism and actuator discussions, to the TU/e-EPE connection, Elena Lomonova and André Vandenput, to the Dutch Ministry of Economic Affairs for the project IOP-EMVT 02201 and to IOP for the help with the Space-Mapping day at CWI.

My recognition to the work of John Bandler and many Space-Mapping contributors (Kaj Madsen, Slawek Koziel and others) and addicts (Jan Snel). A special mention to Jean-Louis Coulomb, John Dennis, Jr. and Philippe Toint for sharing their vast optimization knowledge with me.

Thanks again to Jan ter Maten, Achie Lin and others at Philips interested in Space Mapping, to Charles Tong for giving me the chance to be at LLNL and meet my friend Guangri Xue there, to Kees Oosterlee and the Pacos (Gaspar y Lisbona) at Zaragoza for the pre-CWI initial conditions and to Michael Grant, Silviu Livescu and others at Stanford for easing my transition there.

Y muchas gracias a Anton, the Walking Club (you know who you are), Lorenzo, the Squashing People (Stefano, Nina y Cia., and Johannes Krottje), Marc, Jeroen, Yunus, Jorick, João Ramos (Section 6.3), Jana Werner (for your valuable proofreading, cover design and figures retouch) y a Nada, Susanne and other hard workers at CWI. And last but not least, to CWI itself.

I am thankful to those who helped me when I most needed it.

Amsterdam, February 2007.

Contents

Chapter 1

Introduction

Optimization problems in practice often need function evaluations which are very expensive to compute and gradient information, that can be used for efficient optimization, is in most of these situations *unavailable*. This is the case, e.g., for optimal design problems based on complex finite element simulations. As a consequence, many optimization processes may require very long computing times.

A natural option for dealing with this type of designs is to consider simpler variants of the same problem and combine *multiple levels* of accuracy of the description in order to solve the original optimization problem. We thus replace some expensive functions by others that are simpler to compute (surrogates). Surrogates have been used since long in analysis and design in engineering [18, 26, 78] and recently they have been successfully applied to very time-consuming optimization problems [1, 24, 34, 44]. In surrogate optimization the quality of the initial approximation is iteratively improved. This first surrogate yields the first iterant and with the use of it, the surrogate is improved. This iterative procedure is repeated until some stopping criteria is met. In the whole process, most of the computational effort is spent on the successive evaluation of the surrogate. The few times the expensive original function is evaluated serve for the iterative improvement of the surrogate.

The nature of the surrogates is problem dependent. We clearly can distinguish two different types of surrogates. If no information is a priori available, approximations can be obtained from scratch, in most cases combining experiment design strategies (e.g., Latin hypercube sampling [83] or orthogonal arrays [66]) with interpolation/approximation techniques (e.g., low degree polynomials [25], kriging [53] or radial basis functions [28]). The second type of approximations is found in situations where, because of e.g., experience or simple rules of thumb, the derivation of the surrogate is simplified. With these surrogates we can often include the quite valuable engineering insight already available in multiple design

fields. An example of this second type of surrogates is the use of lumped parameter models (e.g., magnetic [35], electric [7] or thermal [68] equivalent circuits).

Space mapping [7, 8] is a well-known multi-level surrogate-based optimization technique. Though it can be used in combination with any type of surrogate, it is traditionally applied with approximations of the second type described above. The space-mapping technique has been reported as an efficient minimization procedure in a large number of cases [7]. The original space-mapping approach does not require *exact* gradient information. However, it cannot be considered a direct search method [54] because some derivatives related to the expensive function are approximated in these algorithms.

There are many variants of the original space-mapping approach: aggressive space mapping (ASM) [9], trust-region aggressive space mapping (TRASM) [3], neural space mapping (NSM) [5] and implicit space mapping (ISM) [14] are the most significant examples. Although all these schemes do not always converge to the right optimum [35], the solution obtained is generally acceptable for practical purposes. Recently, in [15], the original space-mapping approach is modified according to the framework proposed in [1]. At the expense of incorporating *exact* gradient information, the new scheme yields convergence to the accurate optimum.

Defect correction theory [21] helps in understanding the original space-mapping concept. Defect correction is the underlying basis of a great number of mathematical techniques that essentially solve a complex problem by the iterative use of simpler ones. Many space-mapping procedures can be seen as special cases of the defect correction iteration [35] and this fact can be used in analysis for explaining when the space-mapping approach may fail to find the right solution. This defect correction analysis of space-mapping leads to *manifold mapping* [35], an alternative multi-level surrogate-based optimization technique.

In manifold mapping, the effects of the original expensive function and the surrogates are interpreted in terms of *manifolds*. This perspective, motivated by defect correction theory [35], naturally suggests a new class of algorithms that, without computing *exact* gradient information, have provable convergence to the right solution. In case derivative information concerning the expensive function is available, manifold mapping can take advantage of it for the solution search.

Both space-mapping and manifold-mapping theories are stated in terms of *local* optimality conditions. Though the use of multiple surrogates with several intermediate levels of accuracy might yield, at the expense of increasing the computational load, local optimizers closer to the global one(s), unfortunately for most practical situations there is no guarantee of obtaining a global solution.

Unlike space mapping, the surrogate improvement in manifold mapping is carried out in a very straightforward way. This fact endows manifold-mapping algorithms with a very attractive implementational simplicity.

The introduction of the manifold-mapping technique and its application to optimization problems of practical relevance, is one of the subjects of this thesis.

1.1 Basic definitions and terminology

In this section we describe the general optimization problem to be solved and we introduce the basic terminology used within the multi-level approach. Most of the theory is conceptualized around a two-level (fine and coarse) paradigm. The extension of this notion to a full multi-level approach can be done, specially from a purely algorithmic point of view, in a straightforward manner. In Section 3.4.1 we deal with the issue of a multi-level strategy using more than two levels.

The concept of manifold is crucial in this work. We interpret in terms of manifolds two basic entities in the optimization problem, the models and the constraints. For this reason, a few manifold-related definitions relevant for our discussion are included in this section.

The optimization problem. In this work we aim at minimizing the discrepancy between some given specifications (e.g., measured data) and a mathematical model. Not every optimization problem is of this type, but most practical design situations can be finally formulated in this model-specifications framework.

Let the *specification* of the aim (the data) in an optimization problem be denoted by $\mathbf{y} \in Y \subset \mathbb{R}^m$. Since the true mechanism for approximating this aim can be extremely complex, or even impossible to describe in all its details, we study it by mathematical models. Often, such models appear in several degrees of sophistication. It is the purpose of a multi-level approach to exploit the simpler models by combining their efficiency in computation with the accuracy of the more complex ones. Thus, we distinguish two types of model: fine and coarse.

The fine model. The fine model response function is denoted by $\mathbf{f} : X \subset \mathbb{R}^n \to \mathbb{R}^m$, and $\mathbf{x} \in X$ is the design/control variable. The set $\mathbf{f}(X) \subset \mathbb{R}^m$ represents the fine model reachable aims. The design variables considered in this work are *continuous*. The set X of possible control variables is usually a closed and bounded subset of $\mathbb{R}^n$.

The fine model is assumed to be *accurate* but *expensive* to evaluate. We assume that $\mathbf{f}(\mathbf{x})$ is differentiable but its Jacobian matrix $J_{\mathbf{f}}(\mathbf{x}) = \mathbf{df}/\mathbf{dx}$ is generally supposed to be unavailable. The fine models we analyze in the theory are smooth[1] (see Assumption 1 later in this section), nonlinear and nonconvex. A number of examples of fine models can be found along this work.

For the optimization problem a *fine model cost function*, $F(\mathbf{x}) = \| \mathbf{f}(\mathbf{x}) - \mathbf{y} \|$ is defined, which is a measure for the discrepancy between the aim and a particular response of the mathematical model. This cost function should be minimized[2].

[1] The notion *smooth* is relative to the accuracy with which the optimization problem is solved. Models involving adaptive finite elements are normally nonsmooth (see for example [54, Section 1.2.1.]). We deal with this type of designs in Chapters 5 and 6, but with the tolerances adopted, the underlying models behave smoothly.

[2] In some cases we will use the terms *model minimization* or *minimization of a model* instead

So we look for

$$\mathbf{x}^* = \text{argmin}_{\mathbf{x}\in X} \| \mathbf{f}(\mathbf{x}) - \mathbf{y} \| \ . \tag{1.1}$$

We will refer to $\mathbf{x}^*$ as the *fine model optimum* or just as the *fine optimum*. For simplicity, in most of this work we take for $\| \cdot \|$ the Euclidean norm on $\mathbb{R}^m$. All the theoretical results and algorithms presented in Chapters 2, 3 and 4 can be simply extended to differentiable cost functions.

A design is called *reachable* if there exists an $\mathbf{x}^* \in X$ such that $\mathbf{f}(\mathbf{x}^*) = \mathbf{y}$. This situation can often be expected when $n \geq m$ since in that situation the number of degrees of freedom in the design is larger than or equal to the number of specifications. Reachable designs can be formulated as equations and, hence, they can generally be solved as nonlinear systems. Then the original defect correction theory [21] can directly be applied.

The coarse model. The coarse model is denoted by $\mathbf{c} : Z \subset \mathbb{R}^n \to \mathbb{R}^m$ with $\mathbf{z} \in Z$ the coarse model continuous control variable. The set $\mathbf{c}(Z) \subset \mathbb{R}^m$ is the set of coarse model reachable aims. For the coarse model we have the *coarse model cost function*, $C(\mathbf{z}) = \| \mathbf{c}(\mathbf{z}) - \mathbf{y} \|$. We denote its minimizer (*coarse model optimum*) by $\mathbf{z}^* \in Z$,

$$\mathbf{z}^* = \text{argmin}_{\mathbf{z}\in Z} \| \mathbf{c}(\mathbf{z}) - \mathbf{y} \| \ . \tag{1.2}$$

Again, for this cost function we consider the Euclidean norm. The Jacobian matrix $J_{\mathbf{c}}(\mathbf{z}) = \mathbf{dc}/\mathbf{dz}$ might be available with no significant computational cost.

In this work, for simplicity, we consider $X = Z$, but the general case can be dealt with by the introduction of an additional mapping $\overline{\mathbf{p}} : X \to Z$, as used in [35]. (See Section 3.4.2 for a more detailed analysis of the case with X and Z of different dimensions). When we take for analysis $X = Z$, we prefer to write $\mathbf{c}(\mathbf{x})$ and $\mathbf{x}^*_{\mathbf{c}}$ instead of $\mathbf{c}(\mathbf{z})$ and $\mathbf{z}^*$, respectively. In these circumstances, the fine model optimum will be denoted as $\mathbf{x}^*_{\mathbf{f}}$.

In contrast to the fine model, the coarse model is assumed to be *cheap* to evaluate but *less accurate*. Coarse models can be obtained by simplifications of fine models, e.g., by using a coarser grid in a finite element discretization or by disregarding some nonlinearities in the equations describing the phenomenon under study. Examples for this can be found in Chapters 5 and 6. In some other situations, a strategy for the coarse model generation may already be at our disposal. This is the case with equivalent circuits that we have mentioned above.

Manifolds. If the functions $\mathbf{f}(\mathbf{x})$ and $\mathbf{c}(\mathbf{z})$ are sufficiently smooth, the sets $\mathbf{f}(X) \subset \mathbb{R}^m$ and $\mathbf{c}(Z) \subset \mathbb{R}^m$ can be considered as differential manifolds. Because the concept of manifold is central in our work, here we summarize a few basic notions, from [90].

---

of *minimization associated to a model*.

Definition 1. *An n-dimensional differentiable manifold of class C^k with $1 \leq k \leq \infty$ is a pair $(M, \mathcal{F})$ consisting of an n-dimensional, second countable, locally Euclidean space M together with a differentiable structure of class C^k. Usually, the differentiable manifold $(M, \mathcal{F})$ is denoted as M.*

Definition 2. *A locally Euclidean space M of dimension n is a Hausdorff topological space M for which each point has a neighborhood homeomorphic to an open subset of the Euclidean space $\mathbb{R}^n$.*

Definition 3. *A differentiable structure $\mathcal{F}$ of class C^k $(1 \leq k \leq \infty)$ on a locally Euclidean space M is a collection of coordinate systems (also called an atlas of charts) $\{(U_\alpha, \varphi_\alpha) : \alpha \in A\}$ satisfying:*

(a) $\bigcup_{\alpha \in A} U_\alpha = M$.

(b) $\varphi_\alpha \circ \varphi_\beta^{-1}$ *is* C^k *for every* $\alpha, \beta \in A$.

(c) The collection $\mathcal{F}$ is maximal with respect to (b); that is, if (U, φ) is a coordinate system such that $\varphi \circ \varphi_\alpha^{-1}$ and $\varphi_\alpha \circ \varphi^{-1}$ are differentiable for all $\alpha \in A$, then $(U, \varphi) \in \mathcal{F}$.

For simplicity and without loss of generality, we just consider one particular chart of the manifold that covers a sufficiently large neighborhood of the solution region.

As mentioned above, in this work we address smooth models. We formalize this by the following assumption.

Assumption 1. *The sets $\mathbf{f}(X)$ and $\mathbf{c}(Z)$ are differentiable manifolds of class C^2.*

The concept of the tangent plane for a manifold at a point is needed later in Chapter 4.

Definition 4. *Let M be a differentiable manifold and $\mathbf{v} \in M$ a point in that manifold. The tangent plane for M at $\mathbf{v} \in M$ is defined as the affine space spanned by the Jacobian of φ at $\mathbf{x}$, being φ any chart in the atlas of the manifold such that $\varphi(\mathbf{x}) = \mathbf{v}$.*

Since we only consider one chart we will denote the tangent plane for $\mathbf{f}(X)$ at $\mathbf{f}(\mathbf{x})$ simply by the Jacobian of $\mathbf{f}$ at $\mathbf{x}$, i.e., $J_{\mathbf{f}}(\mathbf{x})$. Thus, the tangent plane is well defined. Similarly, $J_{\mathbf{c}}(\mathbf{z})$ will denote the tangent plane for $\mathbf{c}(Z)$ at $\mathbf{c}(\mathbf{z})$.

Constraints. In designs where the associated constraints are given by functions that are expensive to evaluate (Section 2.4, Section 3.3.2 and Chapter 6) we will proceed in a special manner. We can state a general *constrained* optimization problem as follows

$$\begin{aligned} \mathbf{x}^* &= \operatorname{argmin}_{\mathbf{x} \in X} \|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|, \\ X &= \left\{ \mathbf{x} \in \mathbb{R}^n;\ \mathbf{k}_{\mathbf{f}}(\mathbf{x}) = 0, \quad \overline{\mathbf{k}}_{\mathbf{f}}(\mathbf{x}) \geq 0 \right\}, \end{aligned} \tag{1.3}$$

where $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$, $\mathbf{k_f} : \mathbb{R}^n \to \mathbb{R}^{n_\mathbf{k}}$ and $\overline{\mathbf{k}}_\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^{\overline{n}_\mathbf{k}}$ are assumed to be differentiable, and $n_\mathbf{k}$ and $\overline{n}_\mathbf{k}$ are the number of equality and inequality constraints respectively. The set X is known as the *feasible* set and if $\mathbf{x} \in X$ we say that $\mathbf{x}$ is a feasible point. The process of finding a feasible point could be as complicated as solving the optimization problem itself [64]. In order to have degrees of freedom left for optimization, the number of equality constraints $n_\mathbf{k}$ should be smaller than the number of design variables n, i.e., $n_\mathbf{k} < n$.

We also assume that the set X is a differentiable manifold in $\mathbb{R}^n$. We formalize this in the following assumption.

Assumption 2. *The set X is either a subset of $\mathbb{R}^n$ or a differentiable manifold in $\mathbb{R}^n$ of dimension $n - n_\mathbf{k} > 0$.*

Now we can write the constrained optimization problem (1.3) as (1.1), with X a differentiable manifold.

In general, the model and constraints cannot be expected to be defined over the entire $\mathbb{R}^n$. For example, it makes no sense in many cases to consider negative lengths. Therefore, it is useful to introduce the so called *box constraints*, i.e., lower and upper bounds, $\mathbf{x_l} \in \mathbb{R}^n$ and $\mathbf{x_u} \in \mathbb{R}^n$, respectively, for the design variable $\mathbf{x}$. (It should be noted that box-constraints can be obtained with a proper choice for $\overline{\mathbf{k}}_\mathbf{f}(\mathbf{x})$). The set of points satisfying the box constraints will be denoted by $\widehat{X} \subset \mathbb{R}^n$,

$$\widehat{X} = \{\mathbf{x} \in \mathbb{R}^n;\ \mathbf{x_l} \leq \mathbf{x} \leq \mathbf{x_u}\}, \tag{1.4}$$

where the inequalities should be read componentwise. We also assume that $\mathbf{f}$, $\mathbf{k_f}$ and $\overline{\mathbf{k}}_\mathbf{f}$ are defined correctly in $\widehat{X}$. The set $\widehat{X}$ has been expressly introduced for optimization problems with constraints given by functions that are very expensive to compute. For that reason, $\widehat{X}$ will not be used within other types of designs (for example, those in Chapter 5).

The set $\overline{X}$, defined as those points in $\mathbb{R}^n$ satisfying the equality constraints,

$$\overline{X} = \{\mathbf{x} \in \mathbb{R}^n;\ \mathbf{k_f}(\mathbf{x}) = 0\}, \tag{1.5}$$

will be also used later in Section 3.2.

In the two-level approach for constrained optimization we introduce the coarse model optimum as

$$\begin{aligned} \mathbf{z}^* &= \operatorname{argmin}_{\mathbf{z} \in Z} \|\mathbf{c}(\mathbf{z}) - \mathbf{y}\|, \\ Z &= \left\{\mathbf{z} \in \mathbb{R}^n;\ \mathbf{k_c}(\mathbf{z}) = 0, \quad \overline{\mathbf{k}}_\mathbf{c}(\mathbf{z}) \geq 0\right\}, \end{aligned} \tag{1.6}$$

and $\mathbf{c} : \mathbb{R}^n \to \mathbb{R}^m$, $\mathbf{k_c} : \mathbb{R}^n \to \mathbb{R}^{n_\mathbf{k}}$ and $\overline{\mathbf{k}}_\mathbf{c} : \mathbb{R}^n \to \mathbb{R}^{\overline{n}_\mathbf{k}}$ are also assumed to be differentiable and much easier to compute than their counterparts $\mathbf{f}$, $\mathbf{k_f}$ and $\overline{\mathbf{k}}_\mathbf{f}$, respectively. As for the models, gradients might be only available for the coarse constraints $\mathbf{k_c}$ and $\overline{\mathbf{k}}_\mathbf{c}$. The same examples given above for fine and coarse models can be taken for fine and coarse constraints. We also consider the

set $\widehat{Z} \subset \mathbb{R}^n$ defined as in (1.4) by means of additional upper and lower bounds, $\mathbf{z_l} \in \mathbb{R}^n$ and $\mathbf{z_u} \in \mathbb{R}^n$, for the coarse variable $\mathbf{z}$. The set $\widehat{Z}$ might differ from $\widehat{X}$ though box constraints are inexpensive to evaluate. This could be the case for two simulators that cannot be applied over the same set of designs. In this work, with the exception of Chapter 2, we consider $\widehat{X} = \widehat{Z}$. The general case can be dealt with by the introduction of a mapping $\widehat{\mathbf{p}} : \widehat{X} \to \widehat{Z}$ between these two spaces.

In the algorithms presented in Chapter 2 and Chapter 3 we do not make an essential distinction between equality and inequality constraints. Thus, for a simpler notation we prefer to write (1.3) and (1.6) as, respectively,

$$\mathbf{x}^* = \text{argmin}_{\mathbf{x}\in\widehat{X}} \|\mathbf{f}(\mathbf{x}) - \mathbf{y}\| \quad \text{subject to} \quad \mathbf{k_f}(\mathbf{x}) \geq 0\,, \tag{1.7}$$

$$\mathbf{z}^* = \text{argmin}_{\mathbf{z}\in\widehat{Z}} \|\mathbf{c}(\mathbf{z}) - \mathbf{y}\| \quad \text{subject to} \quad \mathbf{k_c}(\mathbf{z}) \geq 0\,, \tag{1.8}$$

i.e., we include all the constraints in $\mathbf{k_f}$ and $\mathbf{k_c}$. When this notation is used, we refer to the total number of constraints by $m_{\mathbf{k}}$.

First order necessary conditions for constrained optimization. Many local optimization algorithms are designed so that the solution given satisfies some necessary conditions for optimality that involve first order derivatives of the cost and constraints functions. In cases like the ones considered here, where even gradients are difficult to compute, these conditions represent a rigorous way of certifying some validity of the schemes proposed. If $\mathbf{x}^*$ is a local optimum of the constrained problem (1.3), then it satisfies the *Karush-Kuhn-Tucker* (KKT) conditions [65], i.e., there are two Lagrange multiplier vectors $\lambda^* \in \mathbb{R}^{n_\mathbf{k}}$ and $\overline{\lambda}^* \in \mathbb{R}^{\overline{n}_\mathbf{k}}$ such that

$$\mathbf{d}/\mathbf{dx}\,(F - \lambda^{*T}\,\mathbf{k_f} - \overline{\lambda}^{*T}\,\overline{\mathbf{k}}_\mathbf{f})\,(\mathbf{x}^*) = \mathbf{0}\,, \tag{1.9}$$

$$\mathbf{k_f}(\mathbf{x}^*) = \mathbf{0}\,, \tag{1.10}$$

$$\overline{\mathbf{k}}_\mathbf{f}(\mathbf{x}^*) \geq \mathbf{0}\,, \tag{1.11}$$

$$\lambda^*,\, \overline{\lambda}^* \geq \mathbf{0}\,, \tag{1.12}$$

$$\text{diag}\left(\overline{\lambda}^* \cdot \overline{\mathbf{k}}_\mathbf{f}(\mathbf{x}^*)\right) = \mathbf{0}\,, \tag{1.13}$$

where $\cdot$ denotes here the vector direct product. The equations (1.9)-(1.13) should be interpreted componentwise. We notice that in the KKT conditions above we have expressed both equality and inequality constraints (including those of box type) .

Remark 1. *In the case of an unconstrained optimization based on the Euclidean norm, the KKT conditions represent the orthogonality between the tangent plane for* $\mathbf{f}(X)$ *at* $\mathbf{x}^*$ *and the optimal model-specifications discrepancy* $\mathbf{f}(\mathbf{x}^*) - \mathbf{y}$*, i.e.,*

$$J_\mathbf{f}^T(\mathbf{x}^*)\,(\mathbf{f}(\mathbf{x}^*) - \mathbf{y}) = 0\,. \tag{1.14}$$

The following concept will be very useful in Chapter 4 when formalizing the similarity between models.

Definition 5. *We say that an optimization problem is locally convex at* $\mathbf{x} \in \mathbb{R}^n$ *if and only if there is a neighborhood* $U_{\mathbf{x}}$ *of* $\mathbf{x}$ *such that every point in* $U_{\mathbf{x}}$ *satisfying the KKT conditions is a local minimum.*

The assumption for a well defined optimization problem. In order to be sure that the problems we want to solve make sense, we have to assume that solutions for these designs do exist.

Assumption 3. *The fine and coarse optimization problems, characterized by* $\mathbf{y}$, $\mathbf{f}$ *and* X, *and* $\mathbf{c}$ *and* Z, *respectively, are uniquely solvable, i.e.,*

$$\forall \mathbf{y} \in Y \ \exists! \ \mathbf{x}^* \in X \qquad \mathbf{x}^*(\mathbf{y}) = \operatorname{argmin}_{\mathbf{x}\in X} \|\mathbf{f}(\mathbf{x}) - \mathbf{y}\| \,, \tag{1.15}$$

and

$$\forall \mathbf{y} \in Y \ \exists! \ \mathbf{z}^* \in Z \qquad \mathbf{z}^*(\mathbf{y}) = \operatorname{argmin}_{\mathbf{z}\in Z} \|\mathbf{c}(\mathbf{z}) - \mathbf{y}\| \,. \tag{1.16}$$

In most practical cases this assumption is reasonable. If X or Z are closed and bounded non-empty sets and $\mathbf{f}$ and $\mathbf{c}$ are continuous functions, the existence of the solutions is guaranteed. Generally, uniqueness can be achieved by properly reducing the sets X or Z.

The assumption of well-posedness is particularly important for the coarse model and will be discussed in Chapter 4 (Assumption 12).

We also consider in this work practical cases where the fine model is ill-posed, i.e., multiple optima exist (see Section 5.2.2, Section 6.2.1, Section 6.3.2 and Section 6.3.3). Within the required accuracy in the solution, only the design problem in Section 6.2.1 needs a regularization strategy for the two-level approaches applied there.

1.2 Thesis outline

Above we motivated and briefly presented the research carried out in this work. The key concepts underlying *multi-level* techniques, aiming at the efficient solution of *time-expensive optimal designs*, have been exposed. We specified the type of problems to solve and we introduced the notation and terminology adopted. Below, we will sketch the contents of this work by a brief description of each chapter.

In Chapter 2 we first make a review of the space-mapping approach (main theory and related algorithms). The concept of flexibility, a key for understanding when space mapping yields accurate solutions, is then introduced. Then, space mapping is interpreted from the mathematically well-understood defect correction theory. By means of that new analysis perspective, manifold mapping

is presented as an improvement on space mapping, in the sense that the accurate solution is always efficiently found.

In Chapter 3 we describe the manifold-mapping approach, the principal algorithm is derived and also variants for an optimization problem with constraints described by functions that are inexpensive to evaluate. Then, these methods are extended with a trust-region methodology and adapted to designs with constraints expressed by functions with an elevated computational cost. Eventually, we contemplate the use of more than two models in a multi-level approach and we discuss the possibility of fine and coarse design spaces with a different number of dimensions.

In Chapter 4 we present a convergence analysis of the manifold-mapping algorithms in the case of an optimal design, possibly with constraints given by functions that are inexpensive to compute. Consistency is studied first, i.e., every stationary point for the manifold-mapping algorithms coincides with a fine model optimum. Then, we formulate for these schemes precise convergence conditions. We illustrate the theoretical results with some simple optimization examples.

In the Chapters 5 and 6 we apply space-mapping and manifold-mapping algorithms to design problems of practical relevance and we compare them with several other single- and two-level optimization schemes. These optimization problems have been selected from the fields of magnetics, electronics and photonics and they have constraints associated with functions that are either inexpensive (Chapter 5) or expensive (Chapter 6) to evaluate.

Chapter 2

Space-Mapping Review

2.1 Introduction

The space-mapping idea was conceived by Bandler [8] in the field of microwave filter design. It aims at reducing the cost of accurate optimization processes by iteratively correcting a sequence of rougher approximations. In technological applications this allows us to couple, for example, simple rules that represent expert knowledge accumulated over the years with the accuracy of expensive simulation techniques based on the numerical solution of partial differential equations. This combination may yield an efficient method with good accuracy of the final solution.

The space-mapping technique has been mainly applied in electromagnetics [7, 13] but, since the underlying principles are quite general, it could also be used in other areas such as structural [56], vehicle crashworthiness [74] or thermal [68] design. Space mapping has developed enormously during the last decade. As can be seen in the rather complete survey [7], the original idea has gone through a large number of changes and improvements. Further, it can be applied in model reduction for the construction of cheap surrogates [16, 80]. But this modeling purpose is in our work of secondary importance.

In this chapter we analyze space mapping from the perspective of defect correction. Defect correction is a well-understood principle in computational mathematics [22] to solve accurate but complex operator equations in an iterative manner with the help of approximate but easier operators. The idea of solving a problem with the help of a sequence of simpler versions is quite old and well established. Famous examples are, e.g., Newton iteration and multigrid methods [49].

Also the space-mapping technique can be considered as defect correction iteration. This viewpoint allows some mathematical analysis and the construction

of a family of more general iteration procedures, generalizing space mapping.

In this chapter we introduce the concept of model *flexibility*, that gives additional insight and understanding of space-mapping performance. The most significant conclusion derived from that concept is that the best space-mapping techniques are those in which both models involved have equal flexibility, while the less accurate model can be solved much faster.

Traditional space-mapping techniques cannot change the model flexibility. We present manifold mapping, a new space-mapping approach that, by additional left preconditioning [88] with an affine operator, aims at locally endowing the two given models with equal flexibility. Our analysis makes clear that this properly adapted left-preconditioning is essentially more effective than the traditional space mapping, which corresponds with right-preconditioning.

The recently published output space mapping (OSM) [12] and space-mapping based interpolating surrogate (SMIS) scheme [15] are approaches that, following the framework proposed in [1], yield convergence to the accurate solution at the expense of incorporating in the algorithms *exact* gradient information. The manifold-mapping technique has provable convergence to the true solution and can be used without computing that usually expensive derivative information.

The present chapter is structured as follows. Section 2.2 briefly presents the most fundamental concepts and definitions of space mapping. Then it introduces the two basic types of space-mapping solutions and presents typical algorithms found in literature for their computation. The notions of perfect mapping and flexibility of a model are presented in Section 2.3 and there also, some relations between these are explained. In Section 2.4 we indicate how the space-mapping technique can be efficiently applied in designs where the constraint evaluation is time-expensive. In Section 2.5 the defect correction principle is introduced and its relation with space mapping is shown. Some defect correction iteration schemes can be recognized as space-mapping methods but also generalizations can be found. These processes are analyzed and a procedure is constructed so that locally the model flexibility for the different models coincides. This yields our improved space-mapping method. We conclude by means of a few simple examples in Section 2.6 where we illustrate some essential concepts and lemmas from the space-mapping theory presented along the chapter.

2.2 Space-Mapping optimization

Although space mapping is mostly employed as a means for optimization, it is useful to know that it can be applied with two different objectives in mind — optimization and modeling— and that, with the different use, different emphasis is put on particular aspects of space mapping.

We distinguish between: (1) The space-mapping technique aiming at efficient optimization of complex technical problems. The idea is to combine the

advantages of simple (coarse, less accurate, easy to evaluate) models and more complex (fine, more accurate, expensive) models of a phenomenon, to accelerate the numerical computation of a fine model optimum. Typical examples of such an optimization problem are, e.g. fitting a model to a set of measured data or determining the shape of a device in order to achieve a physical response with certain properties. More generally, (2) space mapping deals with constructing a simpler or surrogate model instead of a more expensive and accurate one, and with calibrating this surrogate.

The main difference between the two purposes of space mapping (optimization or modeling) is the required validity region in which the surrogate approximation should correspond with the fine one. In the first case it is sufficient if the two models correspond in the neighborhood of the optimum where the optimization procedure dwells. In the latter case the surrogate model should be sufficiently accurate over a larger range of its parameters.

Though most of the concepts can also be applied in modeling, our work is primarily concerned with the use of space mapping in optimization.

2.2.1 The space-mapping function

It is supposed to be much easier to compute $\mathbf{z}^*$ than $\mathbf{x}^*$. With this knowledge and with the information regarding the similarity between the two models $\mathbf{f}$ and $\mathbf{c}$, we aim at an *efficient* algorithm to compute an accurate approximation of $\mathbf{x}^*$. Obviously, the similarity or discrepancy between the responses of two models used for the same phenomenon is here an important property. It is expressed by the *misalignment function* (see Figure 2.1),

$$r(\mathbf{z}, \mathbf{x}) = \| \mathbf{c}(\mathbf{z}) - \mathbf{f}(\mathbf{x}) \| \ .$$

In particular, for a given $\mathbf{x} \in X$ it is useful to know which $\mathbf{z} \in Z$ yields the smallest discrepancy. This information would allow us to improve the coarse model. Therefore, the *space-mapping function* is introduced.

Definition 6. *The space-mapping function* $\mathbf{p} : X \subset \mathbb{R}^n \to Z \subset \mathbb{R}^n$ *is defined by*

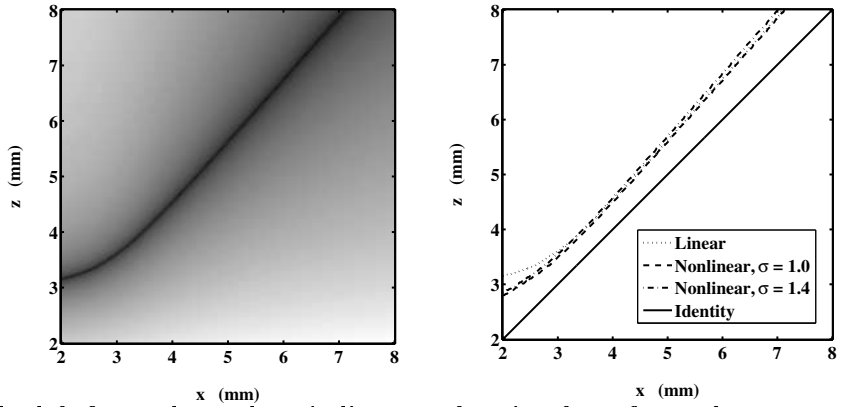
$$\mathbf{p}(\mathbf{x}) = \underset{\mathbf{z} \in Z}{\operatorname{argmin}} \ r(\mathbf{z}, \mathbf{x}) = \underset{\mathbf{z} \in Z}{\operatorname{argmin}} \ \| \mathbf{c}(\mathbf{z}) - \mathbf{f}(\mathbf{x}) \| \ . \tag{2.1}$$

If (2.1) does not allow for a unique minimum, no space mapping $\mathbf{p} : X \to Z$ *exists for that* $\mathbf{x} \in X$. □

The definition implies that for a given control $\mathbf{x}$ the function $\mathbf{p}$ delivers the best coarse model control $\mathbf{z}$ that yields a similar (or the same) response as $\mathbf{x}$ for the fine model. The existence of a unique minimizer is implied by Assumption 3 if $\mathbf{f}(X) \subset Y$, which can be expected in practice.

The process of finding $\mathbf{p}(\mathbf{x})$ for a given $\mathbf{x}$ is called *parameter extraction* or single point extraction because it finds the best coarse model parameter that

Figure 2.1: Misalignment and space-mapping function.



The left figure shows the misalignment function for a fine and a coarse model for the problem in Section 5.2.1. Darker shading shows a smaller misalignment. The right figure shows the identity function and a few space-mapping functions for different coarse models used in Section 5.2.1. (Only the variable x_3 is shown; $x_1 = 5$ mm, $x_2 = 7.5$ mm are kept fixed.)

corresponds with a given fine model control $\mathbf{x}$. It should be noted that this evaluation of the space-mapping function $\mathbf{p}(\mathbf{x})$ requires both an evaluation of $\mathbf{f}(\mathbf{x})$ and a minimization process with respect to $\mathbf{z}$ in $\| \mathbf{c}(\mathbf{z}) - \mathbf{f}(\mathbf{x}) \|$. Hence, in algorithms we should make economic use of space-mapping function evaluations.

The success of the space-mapping technique relies on the existence of desirable properties for the space-mapping function. One is its injectivity. If the model $\mathbf{f}$ has more control parameters than $\mathbf{c}$, the space-mapping function cannot be injective. For that reason, together with the fact that both fine and coarse control variables mostly have the same interpretation, we restrict ourselves to spaces X and Z having the same dimension.

For ill-conditioned designs we expect the space-mapping function to be not injective (different design configurations may yield identical responses). In these situations, it makes sense to consider a coarse model based on a smaller number of design variables (selected for example, with a sensitivity analysis). This is the approach followed in partial space-mapping (PSM) [17]. Ill-conditioned problems properly reformulated (for example, with the introduction of additional regularizing constraints) can be solved within the two-level framework of design spaces with the same number of dimensions.

In implicit space-mapping (ISM) [14] some extra coarse model parameters are introduced for a better model alignment. Though this approach seems to be based on a coarse model with a larger number of design variables than the fine one, the space-mapping function considered in ISM is the original one, with

spaces of equal dimension.

In circumstances where the coarse and the fine model reachable aims are one a subset of the other, i.e. if $\mathbf{c}(Z) \subset \mathbf{f}(X)$ or $\mathbf{f}(X) \subset \mathbf{c}(Z)$, or if they coincide, the notion of flexibility of a model can be introduced (Section 2.3) to derive properties of the corresponding space-mapping function. A priori there is no reason to assume that these sets have such a relation. Even a non-empty intersection might occur. However, given that both models describe the same phenomenon, in some situations such inclusion relation may exist.

Perfect mapping. For reasons that will become immediately clear in the next section, the following definition is introduced [82, Chapter 4].

Definition 7. *A space-mapping function* $\mathbf{p}$ *is called a perfect mapping iff*

$$\mathbf{z}^* = \mathbf{p}(\mathbf{x}^*) \ . \tag{2.2}$$

□

Using the definition of the space-mapping function we see that (2.2) can be written as

$$\underset{\mathbf{z}\in Z}{\operatorname{argmin}} \left\| \mathbf{c}(\mathbf{z}) - \mathbf{y} \right\| = \underset{\mathbf{z}\in Z}{\operatorname{argmin}} \left\| \mathbf{c}(\mathbf{z}) - \mathbf{f}(\mathbf{x}^*) \right\| , \tag{2.3}$$

i.e., a perfect space-mapping function maps $\mathbf{x}^*$, the solution of the fine model optimization, exactly onto $\mathbf{z}^*$, the minimizer of the coarse model design. However, we notice that *perfection* is not a property of the space-mapping function alone but it also depends on the data $\mathbf{y}$ considered. A space-mapping function can be perfect for one set of data but imperfect for a different data set.

The space-mapping function is a crucial element within the space-mapping technique. Quite often in practice multiple solutions to (2.1) are found [7]. Several regularization criteria have been suggested in order to alleviate that difficulty. Examples of these are Multipoint Parameter Extraction (MPE) [11], Statistical Parameter Extraction (SPE) [11], Penalized Parameter Extraction (PPE) [10] or Gradient Parameter Extraction (GPE) [17]. As we will see in the next chapter, manifold mapping avoids the parameter extraction process by taking instead of $\mathbf{p}$ a known mapping $\overline{\mathbf{p}} : X \to Z$ with an easy-to-compute inverse. The significant problem of non-uniqueness associated to (2.1) is thus avoided in manifold mapping.

2.2.2 Space-Mapping approaches

Many space-mapping based algorithms can be found in literature [6, 7], but they all have the same basis. In this section we first describe the original space-mapping idea and two principal approaches (primal and dual). In Section 2.2.3 we show the most important space-mapping algorithms used in practice, and

Figure 2.2: The original space-mapping algorithm.

$\mathbf{x}_0 = \mathbf{z}^* = \operatorname{argmin}_{\mathbf{z}\in Z} \| \mathbf{c}(\mathbf{z}) - \mathbf{y} \|$;
$\mathbf{p}_0 = I$;
for $k = 0, 1, \cdots$, **while** $\| \mathbf{p}(\mathbf{x}_k) - \mathbf{z}^* \| >$ tolerance
do $\mathbf{z}_k = \mathbf{p}(\mathbf{x}_k) = \operatorname{argmin}_{\mathbf{z}\in Z} \| \mathbf{c}(\mathbf{z}) - \mathbf{f}(\mathbf{x}_k) \|$;
from $\{(\mathbf{x}_j, \mathbf{z}_j)\}_{j=0}^{k}$ determine an updated approximation[a] $\mathbf{p}_{k+1}$ for $\mathbf{p}$;
$\mathbf{x}_{k+1} = \mathbf{p}_{k+1}^{-1}(\mathbf{z}^*)$;
enddo

[a] the approximation is assumed to be invertible

Figure 2.3: The primal space-mapping algorithm.

$\mathbf{x}_0 = \mathbf{z}^* = \operatorname{argmin}_{\mathbf{z}\in Z} \| \mathbf{c}(\mathbf{z}) - \mathbf{y} \|$;
for $k = 0, 1, \ldots$, **while** ...
do $\mathbf{z}_k = \mathbf{p}(\mathbf{x}_k)$;
compute an updated approximation $\mathbf{p}_{k+1}$ by means of $\{(\mathbf{x}_j, \mathbf{z}_j)\}_{j=0}^{k}$;
$\mathbf{x}_{k+1} = \operatorname{argmin}_{\mathbf{x}\in X} \|\mathbf{p}_{k+1}(\mathbf{x}) - \mathbf{z}^*\|$;
enddo

later, in Section 2.3, we analyze what approximations to a solution are obtained under certain conditions.

The basic idea behind space-mapping optimization is the following: if either the fine model allows for an almost reachable design (i.e., $\mathbf{f}(\mathbf{x}^*) \approx \mathbf{y}$) or if both models are similar near their respective optima (i.e., $\mathbf{f}(\mathbf{x}^*) \approx \mathbf{c}(\mathbf{z}^*)$) we expect

$$\mathbf{p}(\mathbf{x}^*) = \operatorname*{argmin}_{\mathbf{z}\in Z} \| \mathbf{c}(\mathbf{z}) - \mathbf{f}(\mathbf{x}^*) \| \approx \operatorname*{argmin}_{\mathbf{z}\in Z} \| \mathbf{c}(\mathbf{z}) - \mathbf{y} \| = \mathbf{z}^* .$$

Based on this relation, the *original* space-mapping approach [8] assumes $\mathbf{p}(\mathbf{x}^*) \approx \mathbf{z}^*$. It first determines $\mathbf{z}^*$ and then tries to solve

$$\mathbf{p}(\mathbf{x}_{sm}^*) = \mathbf{z}^* . \tag{2.4}$$

The algorithm for its computation is shown in Figure 2.2. However, in general $\mathbf{p}(\mathbf{x}^*) \neq \mathbf{z}^*$ and even $\mathbf{z}^* \in \mathbf{p}(X)$ is not guaranteed, so that the existence of $\mathbf{x}_{sm}^*$ cannot be assumed. Therefore, the *primal* space-mapping approach seeks for a solution of the minimization problem

$$\mathbf{x}_p^* = \operatorname*{argmin}_{\mathbf{x}\in X} \|\mathbf{p}(\mathbf{x}) - \mathbf{z}^*\| . \tag{2.5}$$

This leads to the primal space-mapping algorithm as shown in Figure 2.3.

An alternative approach can be taken. The idea behind space-mapping optimization is the replacement of the expensive fine model optimization by a surrogate model one. For the surrogate model we can take the coarse model $\mathbf{c}(\mathbf{z})$,

Figure 2.4: The dual space-mapping algorithm.

$\mathbf{x}_0 = \mathbf{z}^* = \operatorname{argmin}_{\mathbf{z}\in Z} \| \mathbf{c}(\mathbf{z}) - \mathbf{y} \|$;
for $k = 0, 1, \ldots,$ **while** ...
do $\mathbf{z}_k = \mathbf{p}(\mathbf{x}_k)$;
compute an updated approximation $\mathbf{p}_{k+1}$ by means of $\{(\mathbf{x}_j, \mathbf{z}_j)\}_{j=0}^{k}$;
$\mathbf{x}_{k+1} = \operatorname{argmin}_{\mathbf{x}\in X} \| \mathbf{c}(\mathbf{p}_{k+1}(\mathbf{x})) - \mathbf{y} \|$;
enddo

and improve its accuracy by the space-mapping function $\mathbf{p} : X \to Z$. Now the improved or *mapped coarse model* $\mathbf{c}(\mathbf{p}(\mathbf{x}))$ may serve as the better surrogate model. Because of (2.1) we expect that $\mathbf{c}(\mathbf{p}(\mathbf{x})) \approx \mathbf{f}(\mathbf{x})$ and hence

$$\| \mathbf{f}(\mathbf{x}) - \mathbf{y} \| \approx \| \mathbf{c}(\mathbf{p}(\mathbf{x})) - \mathbf{y} \| .$$

Then the minimization of $\| \mathbf{c}(\mathbf{p}(\mathbf{x})) - \mathbf{y} \|$ will usually give us a value, $\mathbf{x}_d^*$, close to the desired optimum $\mathbf{x}^*$:

$$\mathbf{x}_d^* = \operatorname*{argmin}_{\mathbf{x}\in X} \| \mathbf{c}(\mathbf{p}(\mathbf{x})) - \mathbf{y} \| . \tag{2.6}$$

This leads to the *dual* space-mapping algorithm as shown in Figure 2.4.

We will see in Section 2.3 that all three approaches coincide when $\mathbf{z}^* \in \mathbf{p}(X)$ and $\mathbf{p}$ is injective. In Section 2.6 the case where the space-mapping solutions are different will be illustrated. A notable difficulty of all the approaches is the expensive evaluation of $\mathbf{p}(\mathbf{x})$. One way to deal with this problem is via convenient choices for the approximations to $\mathbf{p}(\mathbf{x})$ indicated in each of the generic schemes given in Figures 2.2–2.4. The selection of a suitable approximation is made in the specific algorithms below.

2.2.3 Space-Mapping algorithms

The three algorithms given in the Figures 2.2–2.4 are vague in the sense that no choice has been made yet for the approximation of the space-mapping function $\mathbf{p}$. Using a linear approximation gives rise to the more popular space-mapping optimization algorithms. The two most representative examples are shown below. An extensive survey of available algorithms can be found in [7].

The ASM algorithm [9]. The space-mapping function is approximated by linearization to obtain

$$\mathbf{p}_k(\mathbf{x}) = \mathbf{p}(\mathbf{x}_k) + \mathbf{B}_k (\mathbf{x} - \mathbf{x}_k) .$$

In each space-mapping iteration step the matrix $\mathbf{B}_k$ is adapted by a rank-one update. For that purpose a Broyden-type approximation [27] for the Jacobian of

Figure 2.5: The ASM algorithm.

$\mathbf{x}_0 = \mathbf{z}^* = \text{argmin}_{\mathbf{z}\in Z} \| \mathbf{c}(\mathbf{z}) - \mathbf{y} \|$;
$\mathbf{B}_0 = I$;
for $k = 0, 1, \ldots$
while $\| \mathbf{p}(\mathbf{x}_k) - \mathbf{z}^* \| >$ tolerance
do[a] $\mathbf{h}_k = -\mathbf{B}_k^{\dagger}(\mathbf{p}(\mathbf{x}_k) - \mathbf{z}^*)$;
$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{h}_k$;
$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{(\mathbf{p}(\mathbf{x}_{k+1}) - \mathbf{p}(\mathbf{x}_k) - \mathbf{B}_k\mathbf{h})\mathbf{h}^T}{\mathbf{h}^T\mathbf{h}}$;
enddo

[a] $\mathbf{B}_k^{\dagger}$ denotes the pseudo-inverse of $\mathbf{B}_k$.

the space-mapping function $\mathbf{p}(\mathbf{x})$ is used,

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

where $\mathbf{h} = \mathbf{x}_{k+1} - \mathbf{x}_k$. This is combined with original space mapping, so that $\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{B}_k^{-1}(\mathbf{p}(\mathbf{x}_k) - \mathbf{z}^*)$ and the *aggressive space mapping* (ASM) is obtained (Figure 2.5). Note that Broyden's method cannot assure that every $\mathbf{B}_k$ is nonsingular, so, the inverse in the original space-mapping scheme should be read as a pseudo-inverse. ASM just solves equation (2.5) by a quasi-Newton iteration with an approximate Jacobian. We notice that only one evaluation of the fine model is needed per iteration.

The TRASM algorithm [3]. Like ASM, the *trust-region aggressive space mapping* (TRASM) algorithm solves (2.5), now taking into account a trust region. I.e., to stabilize the algorithm, specially in its initial stage, every iterant $\mathbf{x}_{k+1}$ is found in a region not too far away from the earlier one. Further, making use of the Levenberg-Marquardt algorithm [60], TRASM restricts itself to minimizing the residue in the Euclidean norm. By Levenberg-Marquardt the inner loop of TRASM uses the update formula

$$(\mathbf{B}_k^T\mathbf{B}_k + \lambda_k\mathbf{I})\,\mathbf{h}_k = -\mathbf{B}_k^T(\mathbf{p}(\mathbf{x}_k) - \mathbf{z}^{\star}) ,$$
$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{h}_k ,$$

where $\mathbf{B}_k$ is the k-th approximation of the Jacobian of $\mathbf{p}(\mathbf{x})$ at $\mathbf{x}_k$. Notice that the adaptation of the trust region is controlled by adapting λ_k. For $\lambda_k = 0$ the iteration step reduces to a quasi-Newton one, whereas for larger λ_k the step-length reduces and the method tends to steepest descent for minimizing $\| \mathbf{p}(\mathbf{x}) - \mathbf{x}^* \|$. Generally $\mathbf{B}_k$ is constructed by rank-one updates as in ASM. The TRASM algorithm is shown in Figure 2.6.

Figure 2.6: The TRASM algorithm.

$\mathbf{x}_0 = \mathbf{z}^* = \text{argmin}_{\mathbf{z}\in Z} \| \mathbf{c}(\mathbf{z}) - \mathbf{y} \|$;
$\mathbf{B}_0 = \mathbf{I}$, $\lambda_0 = 0$;
for $k = 0, 1, \ldots$
while $\| \mathbf{p}(\mathbf{x}_k) - \mathbf{z}^* \| >$ tolerance
do $\mathbf{h}_k = -(\mathbf{B}_k^T \mathbf{B}_k + \lambda_k \mathbf{I})^{-1} \mathbf{B}_k^T (\mathbf{p}(\mathbf{x}_k) - \mathbf{z}^*)$;
$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{h}_k$;
$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{(\mathbf{p}(\mathbf{x}_{k+1}) - \mathbf{p}(\mathbf{x}_k) - \mathbf{B}_k \mathbf{h}) \mathbf{h}^T}{\mathbf{h}^T \mathbf{h}}$;
update λ_k depending on the local linearity assumption and the trust region required ;
enddo

The HASM algorithm [4]. In general the space-mapping function $\mathbf{p}$ will not be perfect, and hence, a space-mapping based algorithm will *not* yield the solution of the fine model optimization. Therefore, space mapping is sometimes combined with a classical optimization method. This can be done in several ways. One obvious way is using a space-mapping solution as an initial guess for an arbitrary other (classical) optimization algorithm. Another approach is by constructing a combination of a space-mapping and a classical optimization method. Such an algorithm is called *hybrid aggressive space mapping* (HASM). It exploits space mapping when effective, otherwise it defaults to the linearization of the fine model response $\mathbf{f}(\mathbf{x})$

$$\mathbf{l}_k(\mathbf{x}) = \mathbf{f}(\mathbf{x}_k) + \widehat{\mathbf{B}}_k (\mathbf{x} - \mathbf{x}_k) ,$$

where $\widehat{\mathbf{B}}_k$ is a Broyden rank one update approximation to the Jacobian of $\mathbf{f}(\mathbf{x})$ in $\mathbf{x}_k$. It is proved in [58] that (under mild conditions) the iterative process

$$\mathbf{x}_{k+1} = \underset{\mathbf{x}\in X}{\text{argmin}} \| \mathbf{l}_k(\mathbf{x}) - \mathbf{y} \|$$

converges to $\mathbf{x}^*$.

The combined model $\mathbf{v}_k(\mathbf{x})$ is introduced as a convex combination of the mapped coarse and the linearized model:

$$\mathbf{v}_k(\mathbf{x}) = \omega_k \, \mathbf{c}(\mathbf{p}_k(\mathbf{x})) + (1 - \omega_k) \, \mathbf{l}_k(\mathbf{x}) , \tag{2.7}$$

with $\omega_k \in [0, 1]$. Often ω_k is used as a simple switch: either $\omega_k = 1$ or $\omega_k = 0$. Now $\| \mathbf{v}_k(\mathbf{x}) - \mathbf{y} \|$ is the functional to be minimized in the HASM scheme (Figure 2.7). Conditions for convergence are given in [59]. It should be noted that the combined model $\mathbf{v}_k(\mathbf{x})$ is only slightly more expensive than the mapped coarse one $\mathbf{c}(\mathbf{p}_k(\mathbf{x}))$ since for both, the same number of fine model evaluations is used.

Figure 2.7: A HASM algorithm.

$\mathbf{x}_0 = \mathbf{z}^* = \operatorname{argmin}_{\mathbf{z}\in Z} \| \mathbf{c}(\mathbf{z}) - \mathbf{y} \|$;
$\mathbf{p}_0 = I$;
$\widehat{\mathbf{B}}_0 = \mathbf{dc}/\mathbf{dz}\ (\mathbf{z}^*)$;
for $k = 0, 1, \ldots$
do select ω_k;
$\mathbf{x}_{k+1} = \operatorname{argmin}_{\mathbf{x}\in X} \| \omega_k\ \mathbf{c}(\mathbf{p}_k(\mathbf{x})) + (1-\omega_k)(\mathbf{f}(\mathbf{x}_k) + \widehat{\mathbf{B}}_k(\mathbf{x}-\mathbf{x}_k)) - \mathbf{y} \|$;
$\mathbf{z}_{k+1} = \mathbf{p}(\mathbf{x}_{k+1})$;
compute updated $\mathbf{p}_{k+1}$ by means of $\{(\mathbf{x}_j, \mathbf{z}_j)\}_{j=0}^{k}$;
compute updated $\widehat{\mathbf{B}}_{k+1}$ by means of $\{(\mathbf{x}_j, \mathbf{f}(\mathbf{x}_j))\}_{j=0}^{k}$;

enddo

2.3 Perfect mapping, flexibility and reachability

By its definition, perfect mapping relates the similarity of the models and the specifications. If the fine model allows for a *reachable design* (i.e., $\mathbf{f}(\mathbf{x}^*) = \mathbf{y}$), then it is immediate that, independent of the coarse model used, the mapping is always perfect. Also if the coarse and the fine model optimal responses are identical, the space-mapping function is perfect. These two facts are summarized in the following lemma.

Lemma 1. *If either a design is reachable for the fine model, or if the fine and the coarse model give the same optimal response, then the corresponding space-mapping function is perfect. In formula: (i) If* $\mathbf{f}(\mathbf{x}^*) = \mathbf{y}$ *then* $\mathbf{p}(\mathbf{x}^*) = \mathbf{z}^*$.
(ii) If $\mathbf{f}(\mathbf{x}^*) = \mathbf{c}(\mathbf{z}^*)$ *then* $\mathbf{p}(\mathbf{x}^*) = \mathbf{z}^*$. □

The fine and the coarse model in space mapping should, by their nature, show important similarities. Normally the coarse model needs only a small correction in order to be aligned with the fine one. Then the space-mapping function is just a small perturbation of the identity. Therefore, we can expect that, with the tolerances handled in practice, there exists a point $\widetilde{\mathbf{x}} \in X$ such that $\mathbf{f}(\widetilde{\mathbf{x}}) = \mathbf{c}(\mathbf{z}^*)$ and thus, that $\mathbf{p}(\widetilde{\mathbf{x}}) = \mathbf{z}^*$. This is a most interesting situation in which many of the different cases discussed above coincide and space mapping may lead us to the desired optimum. This is shown in the following lemma.

Lemma 2. *(i) If* $\mathbf{z}^* \in \mathbf{p}(X)$, *then* $\mathbf{p}(\mathbf{x}^*_{sm}) = \mathbf{p}(\mathbf{x}^*_p) = \mathbf{p}(\mathbf{x}^*_d) = \mathbf{z}^*$;
(ii) If, in addition, $\mathbf{p}$ *is an injective perfect mapping then* $\mathbf{x}^* = \mathbf{x}^*_{sm} = \mathbf{x}^*_p = \mathbf{x}^*_d$.

Proof. In case $\mathbf{z}^* \in \mathbf{p}(X)$ by definition (2.4) $\mathbf{x}^*_{sm}$ exists and $\mathbf{p}(\mathbf{x}^*_{sm}) = \mathbf{z}^*$. Now by definition (2.5) we see $\mathbf{p}(\mathbf{x}^*_p) = \mathbf{p}(\mathbf{x}^*_{sm})$. Further, $\mathbf{c}(\mathbf{p}(\mathbf{x}^*_{sm})) - \mathbf{y} = \mathbf{c}(\mathbf{z}^*) - \mathbf{y}$, so that by definition (2.6) we have $\mathbf{p}(\mathbf{x}^*_d) = \mathbf{p}(\mathbf{x}^*_{sm})$. Summarizing,

$\mathbf{p}(\mathbf{x}_{sm}^*) = \mathbf{p}(\mathbf{x}_p^*) = \mathbf{p}(\mathbf{x}_d^*) = \mathbf{z}^*$. If, further, $\mathbf{p}$ is injective, the original, primal and dual approach yield the same result. If, in addition, $\mathbf{p}$ is a perfect mapping then $\mathbf{x}_{sm}^* = \mathbf{x}_p^* = \mathbf{x}_d^* = \mathbf{x}^*$. □

In some cases we can expect that the sets of fine and coarse reachable aims $\mathbf{f}(X)$ and $\mathbf{c}(Z)$, respectively, may overlap in a region of $\mathbb{R}^m$ close to their respective optima. The concept of *model flexibility* is now introduced and from that notion some results concerning properties of the space-mapping function can be derived.

Definition 8. *A model is called more flexible than another if the set of its reachable aims contains the set of reachable aims of the other. Two models are equally flexible if their respective sets of reachable aims coincide.* □

Thus, a coarse model $\mathbf{c}$ is more flexible than the fine one $\mathbf{f}$ if $\mathbf{c}(Z) \supset \mathbf{f}(X)$, i.e., if the coarse model response can reproduce all the fine model reachable aims. E.g., if $n \geq m$, linearization makes a flexible coarse model (as long as the Jacobian is regular). Similarly the fine model is more flexible if $\mathbf{f}(X) \supset \mathbf{c}(Z)$. Model flexibility is closely related to properties of the space-mapping function. This is shown in the following lemmas, where $\mathbf{p}$ denotes the space-mapping function.

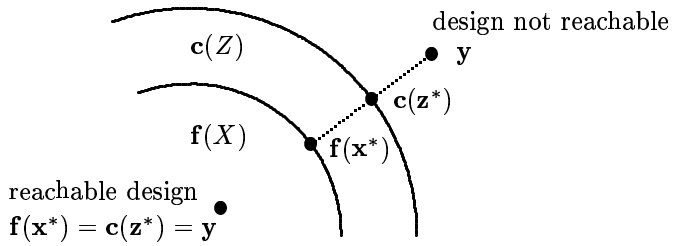
Lemma 3. *If* $\mathbf{c}$ *is more flexible than* $\mathbf{f}$ *then*
(i) $\mathbf{c}(\mathbf{p}(\mathbf{x})) = \mathbf{f}(\mathbf{x}) \quad \forall \mathbf{x} \in X$;
(ii) $\mathbf{p} : X \to Z$ *is a perfect mapping if and only if* $\mathbf{c}(\mathbf{z}^*) = \mathbf{f}(\mathbf{x}^*)$;
(iii) *if* $\mathbf{f} : X \to Y$ *is injective then* $\mathbf{p} : X \to Z$ *is injective;*
(iv) *if* $\mathbf{c}(Z) \setminus \mathbf{f}(X) \neq \emptyset$, *then* $\mathbf{p} : X \to Z$ *cannot be surjective.*

Proof. (i) $\forall \mathbf{f}(\mathbf{x}) \, \exists \mathbf{c}(\mathbf{z}) : \; \mathbf{c}(\mathbf{z}) = \mathbf{f}(\mathbf{x}) \Rightarrow \mathbf{c}(\mathbf{p}(\mathbf{x})) = \mathbf{f}(\mathbf{x}) \quad \forall \, \mathbf{x} \in X$.
(ii⇒) $\mathbf{f}(\mathbf{x}^*) = \mathbf{c}(\mathbf{p}(\mathbf{x}^*)) = \mathbf{c}(\mathbf{z}^*)$ because of (i) and perfect mapping.
(ii⇐) $\mathbf{p}(\mathbf{x}^*) = \operatorname{argmin}_{\mathbf{z} \in Z} \| \mathbf{c}(\mathbf{z}) - \mathbf{f}(\mathbf{x}^*) \| = \operatorname{argmin}_{\mathbf{z} \in Z} \| \mathbf{c}(\mathbf{z}) - \mathbf{c}(\mathbf{z}^*) \| = \mathbf{z}^*$ because the optimum is unique.
(iii) Let $\mathbf{p}(\mathbf{x}_1) = \mathbf{p}(\mathbf{x}_2) \Rightarrow \mathbf{c}(\mathbf{p}(\mathbf{x}_1)) = \mathbf{c}(\mathbf{p}(\mathbf{x}_2)) \Rightarrow \mathbf{f}(\mathbf{x}_1) = \mathbf{f}(\mathbf{x}_2)$ because of (i) $\Rightarrow \mathbf{x}_1 = \mathbf{x}_2$ because $\mathbf{f}$ is injective.
(iv) Let be $\widetilde{\mathbf{z}} : \; \mathbf{c}(\widetilde{\mathbf{z}}) \in \mathbf{c}(Z) \setminus \mathbf{f}(X)$. But then, because of (i), we see that it is not possible to find $\widetilde{\mathbf{x}} \in X : \; \mathbf{p}(\widetilde{\mathbf{x}}) = \widetilde{\mathbf{z}}$. □

Remark. Because of (ii) generally we cannot expect space-mapping functions to be perfect for flexible coarse models unless the two models are equally flexible near the optimum. This fact is illustrated in Figure 2.8. However, we notice that if the design is reachable, the perfect mapping property holds, even if $\mathbf{c}(Z) \setminus \mathbf{f}(X) \neq \emptyset$.

Lemma 4. *If* $\mathbf{f}$ *is more flexible than* $\mathbf{c}$ *then*
(i) $\mathbf{p} : X \to Z$ *is surjective;*
(ii) *if* $\mathbf{f}(X) \setminus \mathbf{c}(Z) \neq \emptyset$, *then* $\mathbf{p}$ *cannot be injective.*

Figure 2.8: Perfect mapping cannot be expected in general for more flexible coarse models.



Proof. (i) $\mathbf{f}(X) \supset \mathbf{c}(Z) \Rightarrow \forall \mathbf{c}(\mathbf{z}) \exists \mathbf{f}(\mathbf{x}) : \ \mathbf{c}(\mathbf{z}) = \mathbf{f}(\mathbf{x})$
$\Rightarrow \forall \mathbf{z} \exists \mathbf{x} : \ \mathbf{z} = \text{argmin}_{\zeta \in Z} \| \mathbf{c}(\zeta) - \mathbf{f}(\mathbf{x}) \| \Rightarrow \forall \mathbf{z} \exists \mathbf{x} : \ \mathbf{z} = \mathbf{p}(\mathbf{x})$.
(ii) All elements $\mathbf{f}(\mathbf{x}) \in \mathbf{f}(X)$ can be divided over equivalence classes for which $\mathbf{c}(\mathbf{z})$ is the closest element in $\mathbf{c}(Z)$, $\mathcal{F}_{\mathbf{z}} = \{\mathbf{f}(\mathbf{x}) \mid \text{argmin}_{\zeta \in Z} \| \mathbf{c}(\zeta) - \mathbf{f}(\mathbf{x}) \| = \mathbf{z}\}$ with $\mathbf{z} \in Z$. All $\mathbf{x}$ with $\mathbf{f}(\mathbf{x}) \in \mathcal{F}_{\mathbf{z}}$ now share the property that $\mathbf{p}(\mathbf{x}) = \mathbf{z}$. If $\mathbf{f}(\mathbf{x}) \in \mathbf{f}(X) \setminus \mathbf{c}(Z)$ it follows that $\| \mathbf{c}(\mathbf{p}(\mathbf{x})) - \mathbf{f}(\mathbf{x}) \| \neq 0$ and hence $\mathcal{F}_{\mathbf{p}(\mathbf{x})}$ contains at least two elements. □

We combine the previous two lemmas in the following.

Lemma 5. *If* $\mathbf{f}$ *and* $\mathbf{c}$ *are equally flexible and* $\mathbf{f}$ *is injective, then* (i) $\mathbf{p}$ *is a bijection, and* (ii) $\mathbf{p}$ *is a perfect mapping.*

Proof. (i) Combine lemmas 3 and 4.
(ii) Because $\mathbf{f}(X) \equiv \mathbf{c}(Z)$, the optimum verifies $\mathbf{f}(\mathbf{x}^*) = \mathbf{c}(\mathbf{z}^*)$. Since the coarse model is more flexible than the fine one we conclude that $\mathbf{p}$ is a perfect mapping. □

The conclusions in Lemma 2 can now be derived from assumptions about model flexibility. Therefore, we can express when the basic space-mapping approach yields the true optimum $\mathbf{x}^*$ in terms of properties concerning the models and their mutual relation (and with independence of the particular problem specifications $\mathbf{y}$).

Lemma 6. (i) *If* $\mathbf{f}$ *is more flexible than* $\mathbf{c}$*, then* $\mathbf{p}(\mathbf{x}^*_{sm}) = \mathbf{p}(\mathbf{x}^*_p) = \mathbf{p}(\mathbf{x}^*_d) = \mathbf{z}^*$.
(ii) *If* $\mathbf{f}$ *and* $\mathbf{c}$ *are equally flexible and* $\mathbf{f}$ *is injective, then* $\mathbf{x}^* = \mathbf{x}^*_{sm} = \mathbf{x}^*_p = \mathbf{x}^*_d$. □

Remark. It is not really necessary for the space-mapping function to be a bijection over the whole domain in which it is defined. In fact, perfect mapping is a property that concerns only a point, and it is enough if the function is injective in a (small) neighborhood. Thus, the assumptions for the lemmas above can be relaxed and stated just locally.

2.4 Constrained optimization with space mapping

The space-mapping theory presented in the previous section is general and it can also be applied to constrained optimization. Just rewriting (2.1) with explicit constraints, we obtain

$$\mathbf{p}(\mathbf{x}) = \text{argmin}_{\mathbf{z}\in\widehat{Z}} \| \mathbf{c}(\mathbf{z}) - \mathbf{f}(\mathbf{x}) \| \quad \text{subject to} \quad \mathbf{k_c}(\mathbf{z}) \geq 0\,. \tag{2.8}$$

The space-mapping solutions should be computed according to (2.5) and (2.6) respectively

$$\mathbf{x}_p^* = \text{argmin}_{\mathbf{x}\in\widehat{X}} \|\mathbf{p}(\mathbf{x}) - \mathbf{z}^*\| \quad \text{subject to} \quad \mathbf{k_f}(\mathbf{x}) \geq 0\,, \tag{2.9}$$

$$\mathbf{x}_d^* = \text{argmin}_{\mathbf{x}\in\widehat{X}} \| \mathbf{c}(\mathbf{p}(\mathbf{x})) - \mathbf{y} \| \quad \text{subject to} \quad \mathbf{k_f}(\mathbf{x}) \geq 0\,. \tag{2.10}$$

If the fine constraints $\mathbf{k_f}$ are expensive to evaluate, this approach is not likely to be the most efficient one.

The double space-mapping approach. In [56] an alternative approach is suggested. There, the models are aligned without taking into account the constraints, and with the aid of an *additional* space mapping. The evaluation of $\mathbf{k_f}$ is replaced by something computationally cheaper. This is an interesting point of view because in many cases the models are not closely related to their respective constraints (for example, a model can be proposed for a certain physical quantity and the constraints are associated with a completely different one). Thus, it makes no much sense to try to align them simultaneously with just one single space-mapping function. In [56] the following two space-mapping functions $\mathbf{p}_m : \widehat{X} \to \widehat{Z}$ and $\mathbf{p}_k : \widehat{X} \to \widehat{Z}$ are introduced

$$\mathbf{p}_m(\mathbf{x}) = \text{argmin}_{\mathbf{z}\in\widehat{Z}} \| \mathbf{c}(\mathbf{z}) - \mathbf{f}(\mathbf{x}) \|\,, \tag{2.11}$$

$$\mathbf{p}_k(\mathbf{x}) = \text{argmin}_{\mathbf{z}\in\widehat{Z}} \| \mathbf{k_c}(\mathbf{z}) - \mathbf{k_f}(\mathbf{x}) \|\,, \tag{2.12}$$

for the alignment of the models and the constraints, respectively. Regularization might be needed in many situations because the constraints and the model minimization have been dropped in (2.11) and in (2.12), respectively.

The constraint space-mapping solution is defined in this case as

$$\mathbf{x}_{csm}^* = \text{argmin}_{\mathbf{x}\in\widehat{X}} \| \mathbf{c}(\mathbf{p}_m(\mathbf{x})) - \mathbf{y} \| \quad \text{subject to} \quad \mathbf{k_c}(\mathbf{p}_k(\mathbf{x})) \geq 0\,. \tag{2.13}$$

Algorithms for this approach can be obtained through approximations of both space-mapping functions $\mathbf{p}_m$ and $\mathbf{p}_k$, as it was done in Section 2.2.3 for standard space mapping.

If we define the function $\mathbf{p}_{csm} : \mathbb{R}^n \to \mathbb{R}^{n+m_\mathbf{k}}$ as a concatenation of $\mathbf{p}_m$ and $\mathbf{k_c} \circ \mathbf{p}_k$, i.e., $\mathbf{p}_{csm} = [\mathbf{p}_m; \mathbf{k_c} \circ \mathbf{p}_k]$, the next result, similar to Lemma 2, can be formulated.

Lemma 7. *If*

$$\mathbf{p}_{csm}(\mathbf{x}^*) = [\mathbf{z}^*; \mathbf{k_c}(\mathbf{z}^*)] \tag{2.14}$$

and in addition, $\mathbf{p}_{csm}$ *is injective then* $\mathbf{x}^*_{csm} = \mathbf{x}^*$.

Proof. By definition of the constraint space-mapping solution $\mathbf{x}^*_{csm}$ and because of (2.14) we have that $\mathbf{p}_m(\mathbf{x}^*_{csm}) = \mathbf{z}^*$ and $\mathbf{k_c}(\mathbf{p}_k(\mathbf{x}^*_{csm})) = \mathbf{k_c}(\mathbf{z}^*)$. If $\mathbf{p}_{csm}$ is injective, we then have that $\mathbf{x}^*_{csm} = \mathbf{x}^*$. □

We recognize in (2.14) a condition similar to perfect-mapping, but now involving the constraints. It should be noticed that the assumption of $\mathbf{k_c}$ being injective cannot be made here, because in most cases the number of design variables is larger than the number of constraints (i.e., $n > m_{\mathbf{k}}$). As with the general space-mapping theory, condition (2.14) might hold only by approximation. For the following case of practical relevance (see [36] or Section 6.2), we can prove that $\mathbf{x}^*_{csm} = \mathbf{x}^*$.

A particular case: $\mathbf{f} \equiv \mathbf{c}$ and $\mathbf{n} \geq \mathbf{m_k}$. In some cases most of the computational effort during an optimization is concentrated in the evaluation of the constraints. For example, when minimizing the mass of a device with a desired functionality. In that case, the model is the mass computation and the function performed by the device can be described by a certain number of constraints. Mass computation is very often a simple task and, thus, the fine and coarse models coincide. The situation $n \geq m_{\mathbf{k}}$ is also observed in practice [36].

The model space-mapping function $\mathbf{p}_m$ can be taken as the identity since both models are already aligned ($\widehat{X} = \widehat{Z}$ and $\mathbf{f} \equiv \mathbf{c}$). Since $n \geq m_{\mathbf{k}}$, it can be assumed that both constraint models $\mathbf{k_c}$ and $\mathbf{k_f}$ are equally flexible in a neighborhood $U_{\mathbf{x}^*} \subset \widehat{X}$ of the true solution $\mathbf{x}^*$. For the same reasons as in Lemma 3 we have

$$\mathbf{k_c}(\mathbf{p}_k(\mathbf{x})) = \mathbf{k_f}(\mathbf{x}) \quad \forall \mathbf{x} \in U_{\mathbf{x}^*} . \tag{2.15}$$

Thus,

$$\begin{aligned}
\mathbf{x}^*_{csm} &= \operatorname{argmin}_{\mathbf{x}\in\widehat{X}} \| \mathbf{c}(\mathbf{p}_m(\mathbf{x})) - \mathbf{y} \| \quad \text{subject to} \quad \mathbf{k_c}(\mathbf{p}_k(\mathbf{x})) \geq 0 \\
&= \operatorname{argmin}_{\mathbf{x}\in U_{\mathbf{x}^*}} \| \mathbf{c}(\mathbf{p}_m(\mathbf{x})) - \mathbf{y} \| \quad \text{subject to} \quad \mathbf{k_c}(\mathbf{p}_k(\mathbf{x})) \geq 0 \\
&= \operatorname{argmin}_{\mathbf{x}\in U_{\mathbf{x}^*}} \| \mathbf{f}(\mathbf{x}) - \mathbf{y} \| \quad \text{subject to} \quad \mathbf{k_f}(\mathbf{x}) \geq 0 .
\end{aligned}$$

We can interpret this result as follows: if the optimization dwells in a neighborhood of the true optimum (for example, with a good initial guess), then the constraint space-mapping solution $\mathbf{x}^*_{csm}$ is the fine optimum $\mathbf{x}^*$. The situation in this particular case is essentially the same as for a reachable design, but now stated for the constraints.

When $n > m_{\mathbf{k}}$, in general, the minimization in (2.12) will not have a unique solution (multiple points in the design space $\widehat{X}$ may satisfy the same constraints).

For example, the following two regularization criteria can be used

$$\begin{aligned} \mathbf{p}_{k_1}(\mathbf{x}) &= \operatorname{argmin}_{\mathbf{z}\in\widehat{Z}} \| \mathbf{c}(\mathbf{z}) - \mathbf{y} \| \quad \text{subject to} \quad \mathbf{k}_{\mathbf{c}}(\mathbf{z}) = \mathbf{k}_{\mathbf{f}}(\mathbf{x}) \,, \\ \mathbf{p}_{k_2}(\mathbf{x}) &= \operatorname{argmin}_{\mathbf{z}\in\widehat{Z}} \| \mathbf{c}(\mathbf{z}) - \mathbf{f}(\mathbf{x}) \| \quad \text{subject to} \quad \mathbf{k}_{\mathbf{c}}(\mathbf{z}) = \mathbf{k}_{\mathbf{f}}(\mathbf{x}) \,. \end{aligned}$$

In the first case the preferred solution is the closest to the specifications $\mathbf{y}$. The second criterion is based on the proximity to the fine model response $\mathbf{f}(\mathbf{x})$. The respective constraint space-mapping solutions coincide with the fine optimum (if for example, a good enough initial guess can be taken for the associated algorithms; i.e., the assumption of being in a neighborhood of the true solution is satisfied).

2.5 Defect correction and space mapping

The main idea underlying the space-mapping technique, i.e. the efficient solution of a complex problem by the iterative use of a simpler one, is known since long in computational mathematics. Generally, it is known as defect correction iteration (of which, e.g., Newton's method, relaxation procedures [22, 49], iterative refinement [22] and multigrid methods [49] are examples). The general principles of the idea are well understood [21, 22, 84]. In this section we first briefly summarize the defect correction principle for the solution of operator equations and extend the idea to optimization problems. We compare defect correction with space mapping and show how space mapping can be derived from defect correction. Then, in the general framework of defect correction new space-mapping algorithms are constructed and analyzed.

2.5.1 The defect correction principle

Let us first consider the problem of solving a nonlinear operator equation

$$\mathcal{F}\,\mathbf{x} = \mathbf{y}, \tag{2.16}$$

where $\mathcal{F} : D \subset E \to \widehat{D} \subset \widehat{E}$ is a continuous, generally nonlinear operator and E and $\widehat{E}$ are Banach spaces. In general, neither injectivity nor surjectivity of the mapping are assumed, but in many cases these properties can be achieved by a proper choice of the subsets D and $\widehat{D}$.

The classical *defect correction* iteration for the solution of the equation (2.16) with $\mathbf{y} \in \widehat{D}$ is based on a sequence of operators $\widetilde{\mathcal{F}}_k : D \to \widehat{D}$ approximating $\mathcal{F}$. We assume that each $\widetilde{\mathcal{F}}_k$ has an easy-to-calculate inverse $\widetilde{\mathcal{G}}_k : \widehat{D} \to D$. Actually, it is the existence of the easy-to-evaluate operator $\widetilde{\mathcal{G}}_k$, rather than the existence of $\widetilde{\mathcal{F}}_k$, what is needed for defect correction, and we do not need to assume neither $\widetilde{\mathcal{F}}_k$ nor $\widetilde{\mathcal{G}}_k$ to be invertible.

Defect correction comes in two brands [21], depending on the space, E or $\widehat{E}$, in which linear combinations for extrapolation are made. The two basic iterative defect correction procedures to generate a sequence of approximations to the solution of (2.16) are

$$\left\{ \begin{array}{lcl} \mathbf{x}_0 & = & \widetilde{\mathcal{G}}_0\,\mathbf{y}\,, \\ \mathbf{x}_{k+1} & = & (I - \widetilde{\mathcal{G}}_{k+1}\,\mathcal{F})\,\mathbf{x}_k + \widetilde{\mathcal{G}}_{k+1}\,\mathbf{y}, \end{array} \right. \tag{2.17}$$

and

$$\left\{ \begin{array}{lcl} \mathbf{l}_0 & = & \mathbf{y}\,, \\ \mathbf{l}_{k+1} & = & (I - \mathcal{F}\,\widetilde{\mathcal{G}}_k)\,\mathbf{l}_k + \mathbf{y}. \end{array} \right. \tag{2.18}$$

In the latter we identify the approximate solution as $\mathbf{x}_k \equiv \widetilde{\mathcal{G}}_k \mathbf{l}_k$. We see that the two iteration processes are dual in the sense that the extrapolation in (2.17) is in the space E, whereas the additions in (2.18) are in $\widehat{E}$. If $\widetilde{\mathcal{G}}_k$ is injective, then an operator $\widetilde{\mathcal{F}}_k$ exists such that $\widetilde{\mathcal{F}}_k\widetilde{\mathcal{G}}_k = \mathcal{I}_{\widehat{D}}$, i.e., $\widetilde{\mathcal{F}}_k$ is the left-inverse of $\widetilde{\mathcal{G}}_k$. Then $\widetilde{\mathcal{F}}_k\mathbf{x}_k = \mathbf{l}_k$ and (2.18) is equivalent with the iterative procedure

$$\left\{ \begin{array}{lcl} \widetilde{\mathcal{F}}_0\mathbf{x}_0 & = & \mathbf{y}, \\ \widetilde{\mathcal{F}}_{k+1}\mathbf{x}_{k+1} & = & \widetilde{\mathcal{F}}_k\,\mathbf{x}_k - \mathcal{F}\widetilde{\mathcal{G}}_k\widetilde{\mathcal{F}}_k\,\mathbf{x}_k + \mathbf{y}\,. \end{array} \right. \tag{2.19}$$

In order to apply (2.19), the injectivity of $\widetilde{\mathcal{G}}_k$ is not really needed and it is immediately seen that neither (2.18) nor (2.19) converge if $\mathbf{y} \notin \widehat{D}$. However, (2.19) can be modified so that it can be used for $\mathbf{y} \notin \widehat{D}$. Therefore, we require injectivity for $\widetilde{\mathcal{F}}_k$ and take $\widetilde{\mathcal{G}}_k$ its left-inverse, i.e., $\widetilde{\mathcal{G}}_k\widetilde{\mathcal{F}}_k = \mathcal{I}_D$. Then (2.19) leads to

$$\left\{ \begin{array}{lcl} \mathbf{x}_0 & = & \widetilde{\mathcal{G}}_0\,\mathbf{y}\,, \\ \mathbf{x}_{k+1} & = & \widetilde{\mathcal{G}}_{k+1}\left(\widetilde{\mathcal{F}}_k\,\mathbf{x}_k - \mathcal{F}\,\mathbf{x}_k + \mathbf{y}\right)\,. \end{array} \right. \tag{2.20}$$

Because (2.20) allows for a non-injective $\widetilde{\mathcal{G}}_k$ this procedure can be used for optimization purposes. In case of an invertible $\widetilde{\mathcal{G}}_k$ both (2.19) and (2.20) are equivalent with (2.18).

Assuming that for large enough k the approximate operators $\widetilde{\mathcal{G}}_k = \widetilde{\mathcal{G}}$ and $\widetilde{\mathcal{F}}_k = \widetilde{\mathcal{F}}$ do not change anymore, upon convergence we see that with $\lim_{k\to\infty}\mathbf{x}_k = \overline{\mathbf{x}}$ and $\lim_{k\to\infty}\mathbf{l}_k = \overline{\mathbf{l}}$ we find for (2.17), (2.18), (2.19) and (2.20) respectively

$$\widetilde{\mathcal{G}}\,\mathcal{F}\,\overline{\mathbf{x}} = \widetilde{\mathcal{G}}\,\mathbf{y}\,, \tag{2.21}$$

$$\mathcal{F}\,\widetilde{\mathcal{G}}\,\overline{\mathbf{l}} = \mathcal{F}\,\overline{\mathbf{x}} = \mathbf{y}\,, \tag{2.22}$$

$$\mathcal{F}\,\widetilde{\mathcal{G}}\,\widetilde{\mathcal{F}}\,\overline{\mathbf{x}} = \mathbf{y}\,, \qquad \text{and} \tag{2.23}$$

$$\overline{\mathbf{x}} = \widetilde{\mathcal{G}}\left(\widetilde{\mathcal{F}}\,\overline{\mathbf{x}} - \mathcal{F}\,\overline{\mathbf{x}} + \mathbf{y}\right). \tag{2.24}$$

The process (2.17) is convergent if the operator $\mathcal{I} - \widetilde{\mathcal{G}}\mathcal{F} : D \to D$ is a contraction, i.e., if the Lipschitz constant $\|\!| \mathcal{I} - \widetilde{\mathcal{G}}\mathcal{F} |\!\|_{D \subset E}$ is less than one. This implies that $\mathcal{F}$ should be injective. Similarly (2.18) converges if $\mathcal{I} - \mathcal{F}\widetilde{\mathcal{G}} : \widehat{D} \to \widehat{D}$ is a contraction, which implies that $\mathcal{F}$ should be surjective. When $\mathcal{F}$ is not surjective, (2.22) shows that (2.18) and (2.19) do not always allow for a fixed point. However, the processes (2.17) and (2.20) may allow for one, provided $\widetilde{\mathcal{G}}$ is not injective. Further we see that, for constant, non-singular affine $\widetilde{\mathcal{G}} = \widetilde{\mathcal{G}}_k$, all processes (2.17)–(2.20) yield identical sequences of approximants. If a regular $\widetilde{\mathcal{G}}$ is not affine, the processes (2.17) and (2.18) give different results.

For our optimization problems, where the design may be not reachable, $\mathbf{y} \in \widehat{D}$, but $\mathbf{y} \notin \mathcal{F}(D)$, i.e., $\mathcal{F}$ is no surjection so that no solution for (2.16) exists and (2.18)-(2.19) cannot converge. Therefore, we drop the idea of finding an $\mathbf{x} \in D$ satisfying (2.16) and we replace the aim by looking for a solution $\mathbf{x}^* \in D$ so that the distance between $\mathcal{F}\mathbf{x}$ and $\mathbf{y}$ is minimal, i.e., we want to find

$$\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x} \in D} \|\!| \mathcal{F}\mathbf{x} - \mathbf{y} |\!\|_{\widehat{E}}\,.$$

For a compact non-empty D and a continuous $\mathcal{F}$, at least a solution exists. If the operators $\widetilde{\mathcal{G}}_k$ are such that (2.17) or (2.20) converges, the stationary point $\overline{\mathbf{x}}$ satisfies (2.21) or (2.24), respectively. For a linear operator $\widetilde{\mathcal{G}}$ this means that $\mathcal{F}\overline{\mathbf{x}} - \mathbf{y} \in \operatorname{Ker}(\widetilde{\mathcal{G}})$.

2.5.2 Defect correction and space mapping

Because the theory of defect correction is formulated in terms of Banach spaces, to compare it with the space-mapping paradigm we restrict ourselves to cost functions $\|\!| \cdot |\!\|$ that take the form of a norm. In fact, for our space-mapping algorithms we simply take $E = \mathbb{R}^n$, $\widehat{E} = \mathbb{R}^m$ and $D = X \subset \mathbb{R}^n$ a compact subset. It is tempting to identify the operator $\mathcal{F}$ with the fine model $\mathbf{f}$, i.e.,

$$\mathcal{F}\,\mathbf{x} = \mathbf{f}(\mathbf{x}), \qquad \text{with } \mathbf{x} \in D = X \subset \mathbb{R}^n\,, \tag{2.25}$$

but, to be more precise, we denote by $\mathcal{F}^\dagger$ the operator for the minimization problem

$$\mathcal{F}^\dagger \mathbf{y} = \operatorname*{argmin}_{\mathbf{x} \in X} \|\!| \mathbf{f}(\mathbf{x}) - \mathbf{y} |\!\|\,. \tag{2.26}$$

Now the operator $\mathcal{F} : D \to \widehat{D}$ in (2.25) is the right-inverse of $\mathcal{F}^\dagger$. We stress that $\mathcal{F}$ generally is not surjective. Defect correction for operator equations is extended to optimization problems by approximating the operator $\mathcal{F}^\dagger$.

The space-mapping idea comes back by identifying the role of the approximate pseudo-inverse. Since the mapped coarse model $\mathbf{c} \circ \mathbf{p}_k$ acts as a surrogate for the

fine model, one clear choice of the approximate operator can be

$$\widetilde{\mathcal{F}}_k\,\mathbf{x} = \mathbf{c}(\mathbf{p}_k(\mathbf{x})) \quad \forall k\,. \tag{2.27}$$

Then $\widetilde{\mathcal{G}}_k$, the approximate inverse of $\mathcal{F}$, is the mapped coarse model optimization

$$\widetilde{\mathcal{G}}_{k+1}\,\mathbf{y} = \operatorname*{argmin}_{\mathbf{x}\in X} \|\,\mathbf{c}(\mathbf{p}_k(\mathbf{x})) - \mathbf{y}\|\,. \tag{2.28}$$

At this stage we make no particular choice for the functions $\mathbf{p}_k$, but we only assume that (2.28) is easily evaluated. For example, if the spaces X and Z can be identified, we simply may take $\mathbf{p}_k = I$, the identity operator. We can try and find better choices for $\mathbf{p}_k$, but —as we assumed that both X and Z are in $\mathbb{R}^n$— only bijections for the functions $\mathbf{p}_k$ are considered. As for $\widetilde{\mathcal{G}}_k$, we also assume for $\mathbf{p}_k$ that they do not to change anymore for k large enough, i.e., $\mathbf{p}_k = \overline{\mathbf{p}}$ for $k \geq k_0$.

2.5.3 Defect correction in optimization

Construction of new iterative optimization algorithms

Using the basic defect correction processes and the relations (2.25), (2.27) and (2.28) that are associated with space mapping, we derive two new defect correction iteration schemes that can be used for optimization. Substitution of the relations into the processes (2.17) and (2.20), respectively, yields the following initial estimate and iteration processes for $k = 0, 1, 2, \cdots$

$$\begin{aligned}
\mathbf{x}_0 &= \operatorname{argmin}_{\mathbf{x}\in X} \|\,\mathbf{c}(\mathbf{p}_0(\mathbf{x})) - \mathbf{y}\|\,, && (2.29)\\
\mathbf{x}_{k+1} &= \mathbf{x}_k - \operatorname{argmin}_{\mathbf{x}\in X} \|\,\mathbf{c}(\mathbf{p}_k(\mathbf{x})) - \mathbf{f}(\mathbf{x}_k)\| \\
&\qquad + \operatorname{argmin}_{\mathbf{x}\in X} \|\,\mathbf{c}(\mathbf{p}_k(\mathbf{x})) - \mathbf{y}\|\,, && (2.30)\\
\mathbf{x}_{k+1} &= \operatorname{argmin}_{\mathbf{x}\in X} \|\,\mathbf{c}(\mathbf{p}_k(\mathbf{x})) - \mathbf{c}(\mathbf{p}_k(\mathbf{x}_k)) + \mathbf{f}(\mathbf{x}_k) - \mathbf{y}\|\,. && (2.31)
\end{aligned}$$

We denote (2.30) and (2.31) as DeC-X and DeC-Y, respectively. Again, the two processes (2.30) and (2.31) are dual in the sense that extrapolation is applied in the space X for process (2.30) and extrapolation in Y for process (2.31). In the case when the spaces X and Z can be identified, for the initial estimate (2.29) we take $\mathbf{p}_0 = I$, the identity, like in the space-mapping algorithms.

In the above iterations all iterated minimizations are over functions involving the surrogate model, $\mathbf{c} \circ \mathbf{p}_k$. However, it is the coarse model that was assumed to be cheaply optimized. Therefore, it is convenient to derive procedures, similar to the ones given, such that direct optimization over the coarse model becomes central. By taking $\mathcal{F}\,\mathbf{z} = \mathbf{f}(\mathbf{q}(\mathbf{z}))$, $\widetilde{\mathcal{F}}_k\,\mathbf{z} = \mathbf{c}(\mathbf{z})$ and $\widetilde{\mathcal{G}}_k\,\mathbf{y} = \operatorname{argmin}_{\mathbf{z}\in Z} \|\,\mathbf{c}(\mathbf{z}) - \mathbf{y}\|\,$, with $\mathbf{q}$ and $\mathbf{q}_k$ bijections from Z to X fulfilling in every iteration $\mathbf{q}\,\mathbf{z}_k = \mathbf{q}_k\,\mathbf{z}_k$

we obtain for $k = 0, 1, 2, \cdots$

$$\begin{aligned} \mathbf{z}_0 &= \mathbf{z}^* = \operatorname{argmin}_{\mathbf{z}\in Z} \| \mathbf{c}(\mathbf{z}) - \mathbf{y} \| , && (2.32) \\ \mathbf{z}_{k+1} &= \mathbf{z}_k - \operatorname{argmin}_{\mathbf{z}\in Z} \| \mathbf{c}(\mathbf{z}) - \mathbf{f}(\mathbf{q}_k(\mathbf{z}_k)) \| + \mathbf{z}^* , && (2.33) \\ \mathbf{z}_{k+1} &= \operatorname{argmin}_{\mathbf{z}\in Z} \| \mathbf{c}(\mathbf{z}) - \mathbf{c}(\mathbf{z}_k) + \mathbf{f}(\mathbf{q}_k(\mathbf{z}_k)) - \mathbf{y} \| . && (2.34) \end{aligned}$$

We denote (2.33) and (2.34) as DeC-Z and DeC-Y, respectively (the reason for the latter being identified with DeC-Y will be given in Lemma 8). Because the solution is wanted in terms of fine model control variables, the procedures are complemented with $\mathbf{x}_k = \mathbf{q}_k(\mathbf{z}_k)$. The bijections can be interpreted as $\mathbf{q}_k = \mathbf{p}_k^{-1}$. Again, for $k > k_0$, we assume the iteration process to be stationary: $\mathbf{q}_k = \overline{\mathbf{q}}$.

Lemma 8. (i) *The initial estimates (2.29) and (2.32) are equivalent, as well as the processes (2.31) and (2.34).* (ii) *If $\mathbf{p}_k$ is linear and the same for all k, $\mathbf{p}_k = \overline{\mathbf{p}}$, then also (2.30) and (2.33) are equivalent.*

Proof. (i) We substitute in (2.29) $\mathbf{p}_0(\mathbf{x}_0) = \mathbf{z}_0$ to obtain (2.32). In (2.31) we substitute $\mathbf{p}_k(\mathbf{x}_k) = \mathbf{z}_k$, $\mathbf{p}_{k+1}(\mathbf{x}_{k+1}) = \mathbf{z}_{k+1}$ and $\mathbf{q}(\mathbf{z}_k) = \mathbf{x}_k$ to obtain (2.34). (ii) The same substitutions in (2.30) make

$$\begin{aligned} \mathbf{q}_{k+1}\,\mathbf{z}_{k+1} = \mathbf{q}_k\,\mathbf{z}_k &- \mathbf{q}_k \operatorname{argmin}_{\mathbf{z}\in X} \| \mathbf{c}(\mathbf{x}) - \mathbf{f}(\mathbf{q}_k(\mathbf{z}_k)) \| \\ &+ \mathbf{q}_k \operatorname{argmin}_{\mathbf{x}\in X} \| \mathbf{c}(\mathbf{x}) - \mathbf{y} \| . \end{aligned}$$

Now linearity of $\mathbf{q}_k$ and $\mathbf{p}_k\,\mathbf{q}_{k+1}\,\mathbf{z}_k = \mathbf{z}_k$ complete the proof. □

Recovering space mapping from defect correction

With a proper choice of $\mathbf{p}_k$ we can recover the space-mapping approaches presented in Section 2.2.2. The primal space-mapping generic scheme is obtained just by applying defect correction to the problem

$$\mathcal{F}^\dagger\,\mathbf{z}^* = \operatorname{argmin}_{\mathbf{x}\in X} \|\mathbf{p}(\mathbf{x}) - \mathbf{z}^*\| ,$$

where the easy to calculate pseudo-inverses are

$$\widetilde{\mathcal{G}}_k\,\mathbf{z}^* = \operatorname{argmin}_{\mathbf{x}\in X} \|\mathbf{p}_k(\mathbf{x}) - \mathbf{z}^*\| ,$$

and $\mathbf{p}_k$ is an arbitrary approximation of $\mathbf{p}$ satisfying $\mathbf{p}_k(\mathbf{x}_k) = \mathbf{p}(\mathbf{x}_k)$.

The dual space-mapping approach can be derived as follows. Let $\mathbf{p}_k$ be an approximation of $\mathbf{p}$ satisfying $\mathbf{p}_k(\mathbf{x}_k) = \mathbf{p}(\mathbf{x}_k)$, then (2.30) reduces to

$$\mathbf{x}_{k+1} = \operatorname{argmin}_{\mathbf{x}\in X} \| \mathbf{c}(\mathbf{p}_k(\mathbf{x})) - \mathbf{y} \| , \qquad (2.35)$$

which coincides with the dual space-mapping algorithm.

A special case. If $X = Z$ and $\mathbf{p}_k = I$, then (2.29) and (2.30) reduce to

$$\left\{ \begin{array}{lcl} \mathbf{x}_0 & = & \operatorname{argmin}_{\mathbf{z}\in Z} \| \mathbf{c}(\mathbf{z}) - \mathbf{y} \| = \mathbf{z}^* \,, \\ \mathbf{x}_{k+1} & = & \mathbf{x}_k - \operatorname{argmin}_{\mathbf{z}\in Z} \| \mathbf{c}(\mathbf{z}) - \mathbf{f}(\mathbf{x}_k) \| + \mathbf{z}^* \,, \\ & = & \mathbf{x}_k - \mathbf{p}(\mathbf{x}_k) + \mathbf{z}^* \,, \qquad k = 0, 1, \cdots \,. \end{array} \right. \tag{2.36}$$

This is the iterative method to solve the original space-mapping problem (2.4) which has been published in [20, 67]. Clearly, the iteration can only converge if $\mathbf{p}(\mathbf{x}^*_{sm}) = \mathbf{z}^*$ allows for a solution and the convergence rate is determined by the Lipschitz constant $\| I - \mathbf{p}(\cdot) \|$. Hence, a necessary and sufficient condition for convergence of (2.36) is $\| I - \mathbf{p}(\cdot) \| < 1$.

Properties of the new algorithms

Convergence of DeC-X. In the case of convergence of (2.30) we obtain, with fixed point $\lim_{k\to\infty} \mathbf{x}_k = \overline{\mathbf{x}}$, the equality

$$\widehat{\mathbf{x}} := \operatorname{argmin}_{\mathbf{x}\in X} \| \mathbf{c}(\overline{\mathbf{p}}(\mathbf{x})) - \mathbf{f}(\overline{\mathbf{x}}) \| = \operatorname{argmin}_{\mathbf{x}\in X} \| \mathbf{c}(\overline{\mathbf{p}}(\mathbf{x})) - \mathbf{y} \| \,, \tag{2.37}$$

which shows that the design $\overline{\mathbf{x}} \in X$ is associated with $\widehat{\mathbf{y}} = \mathbf{c}(\overline{\mathbf{p}}(\widehat{\mathbf{x}})) \in \mathbf{c}(\overline{\mathbf{p}}(X))$ that is the best approximation of $\mathbf{y}$ reachable in $\mathbf{c}(\overline{\mathbf{p}}(X)) = \mathbf{c}(Z)$, and at the same time such that $\mathbf{f}(\overline{\mathbf{x}})$ is best approximated in $\mathbf{c}(Z)$ by the same $\widehat{\mathbf{y}}$. Hence, both $\mathbf{f}(\overline{\mathbf{x}})$ and $\mathbf{y}$ belong to the class $W_{\widehat{\mathbf{y}}} \subset \mathbb{R}^m$ of points in $\mathbb{R}^m$ for which $\widehat{\mathbf{y}}$ is the closest point in $\mathbf{c}(Z) \subset \mathbb{R}^m$. Noting that $\overline{\mathbf{p}}$ is a bijection, the equality (2.37) can be written briefly as

$$\widehat{\mathbf{x}} := \overline{\mathbf{p}}^{-1}(\mathbf{p}(\overline{\mathbf{x}})) = \overline{\mathbf{p}}^{-1}(\mathbf{z}^*) \,, \tag{2.38}$$

where $\mathbf{p}$ denotes the space-mapping function (2.1).

If we assume $\widehat{E}$ to be a Hilbert space, from (2.37) we see that both $\mathbf{c}(\overline{\mathbf{p}}(\widehat{\mathbf{x}})) - \mathbf{f}(\overline{\mathbf{x}})$ and $\mathbf{c}(\overline{\mathbf{p}}(\widehat{\mathbf{x}})) - \mathbf{y}$ are perpendicular to the tangent plane for $\mathbf{c}(\overline{\mathbf{p}}(X))$ at $\mathbf{c}(\overline{\mathbf{p}}(\widehat{\mathbf{x}}))$. We denote the linear manifold perpendicular to the tangent plane for $\mathbf{c}(\overline{\mathbf{p}}(X))$ at $\mathbf{c}(\overline{\mathbf{p}}(\widehat{\mathbf{x}}))$ by $\mathbf{c}(\overline{\mathbf{p}}(X))^{\perp}(\widehat{\mathbf{x}})$. Thus

$$\begin{array}{rcl} \mathbf{c}(\overline{\mathbf{p}}(\widehat{\mathbf{x}})) - \mathbf{f}(\overline{\mathbf{x}}) & \in & \mathbf{c}(\overline{\mathbf{p}}(X))^{\perp}(\widehat{\mathbf{x}}) \,, \\ \mathbf{c}(\overline{\mathbf{p}}(\widehat{\mathbf{x}})) - \mathbf{y} & \in & \mathbf{c}(\overline{\mathbf{p}}(X))^{\perp}(\widehat{\mathbf{x}}) \,, \end{array}$$

and hence

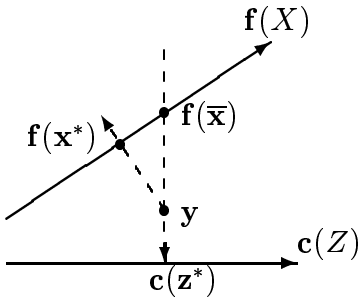
$$\mathbf{f}(\overline{\mathbf{x}}) - \mathbf{y} \quad \in \quad \mathbf{c}(\overline{\mathbf{p}}(X))^{\perp}(\widehat{\mathbf{x}}) = \mathbf{c}(\overline{\mathbf{p}}(X))^{\perp}(\overline{\mathbf{p}}^{-1}\mathbf{p}(\overline{\mathbf{x}})) = \mathbf{c}(Z)^{\perp}(\mathbf{z}^*) \,.$$

This situation is shown in Figure 2.9. We summarize this result in the following lemma.

Lemma 9. *Let $\widehat{E}$ be a Hilbert space. In case of convergence of (2.30) with a fixed point $\overline{\mathbf{x}}$ we obtain*

$$\mathbf{f}(\overline{\mathbf{x}}) - \mathbf{y} \quad \in \quad \mathbf{c}(Z)^{\perp}(\mathbf{z}^*) = \mathbf{c}(\overline{\mathbf{p}}(X))^{\perp}(\overline{\mathbf{p}}^{-1}\mathbf{p}(\overline{\mathbf{x}})) \,. \qquad \square$$

Figure 2.9: The relative location of $\mathbf{c}(\mathbf{z}^*)$, $\mathbf{f}(\mathbf{x}^*)$ and $\mathbf{f}(\overline{\mathbf{x}})$.



Because $\overline{\mathbf{p}}$ is a bijection, from (2.38) we directly see the following result.

Lemma 10. *Let* $\mathbf{p}(\mathbf{x})$ *be the space-mapping function as defined in (2.1). If (2.30) converges to* $\overline{\mathbf{x}}$ *then* $\mathbf{p}(\overline{\mathbf{x}}) = \mathbf{z}^*$. □

Corollary. It follows from the definition of a *perfect mapping* that, with $\overline{\mathbf{x}}$ the convergence result of (2.30), $\mathbf{p}(\overline{\mathbf{x}}) = \mathbf{p}(\mathbf{x}^*)$ if and only if $\mathbf{p}$ is a perfect mapping.
Remark. We note that convergence of (2.30) implies $\mathbf{z}^* \in \mathbf{p}(X)$. Hence, in the case $\mathbf{z}^* \notin \mathbf{p}(X)$ the algorithm cannot reach a stationary point.

In case we have knowledge about the flexibility of the models we can go a bit further with the analysis and state the next lemma.

Lemma 11. *If the coarse model is more flexible than the fine one, i.e.,* $\mathbf{c}(Z) \supset \mathbf{f}(X)$, *and* $\mathbf{y} \in \mathbf{c}(Z) \setminus \mathbf{f}(X)$, *then (2.30) cannot converge.*

Proof. Because $\mathbf{y} \in \mathbf{c}(\overline{\mathbf{p}}(X))$ there exists a $\widehat{\mathbf{x}}$ such that $\mathbf{y} = \mathbf{c}(\overline{\mathbf{p}}(\widehat{\mathbf{x}}))$. If (2.30) converges, then (2.37) holds and $\widehat{\mathbf{x}} = \operatorname{argmin}_{\mathbf{x}\in X} \| \mathbf{c}(\overline{\mathbf{p}}(\mathbf{x})) - \mathbf{f}(\overline{\mathbf{x}}) \|$.

Since the surrogate model is more flexible than the fine model $\mathbf{c}(\overline{\mathbf{p}}(\widehat{\mathbf{x}})) = \mathbf{f}(\overline{\mathbf{x}})$. This yields a contradiction. Hence (2.30) cannot converge. □

Remark. In the end we have what one would have expected. If $\mathbf{c}(Z) = \mathbf{f}(X)$ and the process (2.30) converges, the space mapping is perfect and the fixed point of the iteration is the solution of our optimization problem.

Convergence of DeC-Y. In Section 2.5.3 Lemma 8, it was shown that the iterations (2.31) and (2.34) are equivalent. Hence, their convergence results coincide. For (2.31) we derive

Lemma 12. *Let the space* $\widehat{E}$ *be a Hilbert space. In the case of convergence of (2.31), with fixed point* $\lim_{k\to\infty} \mathbf{x}_k = \overline{\mathbf{x}}$ *we obtain*

$$\mathbf{f}(\overline{\mathbf{x}}) - \mathbf{y} \in \mathbf{c}(\overline{\mathbf{p}}(X))^{\perp}(\overline{\mathbf{x}}) .$$

Proof. In the case of convergence of (2.31), with fixed point $\lim_{k\to\infty} \mathbf{x}_k = \overline{\mathbf{x}}$ we obtain

$$\overline{\mathbf{x}} := \underset{\mathbf{x}\in X}{\operatorname{argmin}} \| \mathbf{c}(\overline{\mathbf{p}}(\mathbf{x})) - \mathbf{c}(\overline{\mathbf{p}}(\overline{\mathbf{x}})) + \mathbf{f}(\overline{\mathbf{x}}) - \mathbf{y}\| \ . \tag{2.39}$$

So, it follows that for all $\mathbf{h} \in \mathbb{R}^n$ with $\|\mathbf{h}\| \geq 0$

$$\| \mathbf{c}(\overline{\mathbf{p}}(\overline{\mathbf{x}} + \mathbf{h})) - \mathbf{c}(\overline{\mathbf{p}}(\overline{\mathbf{x}})) + \mathbf{f}(\overline{\mathbf{x}}) - \mathbf{y}\| \ \geq \| \mathbf{f}(\overline{\mathbf{x}}) - \mathbf{y}\| \ .$$

With $\widehat{E}$ a Hilbert space, this implies that

$$\mathbf{f}(\overline{\mathbf{x}}) - \mathbf{y} \perp \mathbf{c}(\overline{\mathbf{p}}(\overline{\mathbf{x}} + \mathbf{h})) - \mathbf{c}(\overline{\mathbf{p}}(\overline{\mathbf{x}})) \qquad \forall \mathbf{h} \in \mathbb{R}^n \ ,$$

so that $\mathbf{f}(\overline{\mathbf{x}}) - \mathbf{y}$ is perpendicular to the linear manifold through $\mathbf{c}(\overline{\mathbf{p}}(\overline{\mathbf{x}}))$ that is tangential to $\mathbf{c}(\overline{\mathbf{p}}(X))$. □

Remark. Notice the slight difference with the result for DeC-X.

Lemma 13. *With a flexible coarse model and* $\mathbf{p}_k(\mathbf{x}_k) = \mathbf{p}(\mathbf{x}_k)$ *the iteration processes (2.30) and (2.31) yield identical sequences of approximations. Hence, under these conditions, Lemmas 10 and 11 for (2.30) also hold for (2.31).*

Proof. The main iteration in DeC-Y is

$$\mathbf{x}_{k+1} = \operatorname{argmin}_{\mathbf{x}\in X} \| \mathbf{c}(\mathbf{p}_k(\mathbf{x})) - \mathbf{c}(\mathbf{p}_k(\mathbf{x}_k)) + \mathbf{f}(\mathbf{x}_k) - \mathbf{y}\| \ .$$

From Lemma 3 for a more flexible coarse model

$$\mathbf{c}(\mathbf{p}(\mathbf{x}_k)) = \mathbf{f}(\mathbf{x}_k) \ .$$

Since $\mathbf{p}_k(\mathbf{x}_k) = \mathbf{p}(\mathbf{x}_k)$ we have

$$\mathbf{x}_{k+1} = \operatorname{argmin}_{\mathbf{x}\in X} \| \mathbf{c}(\mathbf{p}_k(\mathbf{x})) - \mathbf{y}\| \ .$$

which coincides with DeC-X (2.30) because we satisfy the hypothesis of recovering space mapping from defect correction. □

Convergence of DeC-Z. Here the results concerning the analysis of (2.33) are given. Because of the similarity of the lemmas and the proofs for the scheme (2.33) and those for (2.30), we omit the proofs.

Lemma 14. *Let* $\widehat{E}$ *be a Hilbert space. In case of convergence of (2.33) with a fixed point* $\overline{\mathbf{x}} = \overline{\mathbf{q}}(\overline{\mathbf{z}})$ *we obtain*

$$\mathbf{f}(\overline{\mathbf{x}}) - \mathbf{y} \quad \in \quad \mathbf{c}(Z)^{\perp}(\mathbf{z}^*) \equiv \mathbf{c}(\overline{\mathbf{p}}(X))^{\perp}(\overline{\mathbf{q}}\,\mathbf{p}(\overline{\mathbf{x}})) \ . \qquad \square$$

Lemma 15. *Let* $\mathbf{p}(\mathbf{x})$ *be the space-mapping function as defined in (2.1). If (2.33) converges to* $\overline{\mathbf{x}} = \overline{\mathbf{q}}(\overline{\mathbf{z}})$ *then* $\mathbf{p}(\overline{\mathbf{x}}) = \mathbf{z}^*$. □

Again it follows that, with $\overline{\mathbf{x}}$ the stationary point of (2.33), $\mathbf{p}(\overline{\mathbf{x}}) = \mathbf{p}(\mathbf{x}^*)$ if and only if $\mathbf{p}$ is a perfect mapping. If $\mathbf{c}(Z) = \mathbf{f}(X)$ and the process converges, the fixed point of the iteration is the solution of our optimization problem.

Lemma 16. *If the coarse model is more flexible than the fine one, i.e.,* $\mathbf{c}(Z) \supset \mathbf{f}(X)$*, and* $\mathbf{y} \in \mathbf{c}(Z) \setminus \mathbf{f}(X)$*, then (2.33) cannot converge.* □

The processes with linear coarse model and mappings

A coarse model that is linear in its control variables, in combination with linear mappings for either $\mathbf{p}_k$ and $\overline{\mathbf{p}}$ or $\mathbf{q}_k$ and $\overline{\mathbf{q}}$, can be attractive in practice and the respective analysis can give some additional insight. Under these linearity conditions, all defect correction processes above coincide. For the analysis we take the iteration (2.33) with the Euclidean norm as a starting point.

Denoting by C the $m \times n$ matrix that describes the linear coarse model, we obtain

$$\left\{ \begin{array}{lcl} \mathbf{z}_0 & = & \mathbf{z}^* = \mathrm{argmin}_{\mathbf{z}\in Z} \|C\mathbf{z} - \mathbf{y}\|_2 \,, \\ \mathbf{z}_{k+1} & = & \mathbf{z}_k - \mathrm{argmin}_{\mathbf{z}\in Z} \|C\mathbf{z} - \mathbf{f}(\mathbf{q}_k(\mathbf{z}_k))\|_2 + \mathbf{z}^* \,, \end{array} \right. \tag{2.40}$$

or

$$\left\{ \begin{array}{lcl} \mathbf{z}_0 & \in & C^\dagger \mathbf{y} + \mathrm{Span}(V_0) \,, \\ \mathbf{z}_{k+1} & \in & \mathbf{z}_k - C^\dagger(\mathbf{f}(\mathbf{q}_k(\mathbf{z}_k)) - \mathbf{y}) + \mathrm{Span}(V_0) \,, \end{array} \right. \tag{2.41}$$

where $C^\dagger$ is the pseudo-inverse of C and $\mathrm{Span}(V_0) \subset \mathbb{R}^m$ is the kernel of $C^T C$. The fixed point $\mathbf{f}(\overline{\mathbf{x}}) = \mathbf{f}(\overline{\mathbf{q}}(\overline{\mathbf{z}}))$ of iteration (2.41) is characterized by

$$C^\dagger(\mathbf{y} - \mathbf{f}(\overline{\mathbf{x}})) \in \mathrm{Span}(V_0) \,.$$

In the case $m \geq n$ and with a full rank C operator, the term $\mathrm{Span}(V_0)$ vanishes, so that $C^T(\mathbf{y} - \mathbf{f}(\overline{\mathbf{x}})) = 0$, i.e., the discrepancy between the data and the optimal response is in the kernel of C^T.

Convergence of the linearized process. From (2.41) it is clear that for a non-full rank C matrix or if $m < n$, the solution is not uniquely determined and the iteration cannot be a contraction, whereas for a full-rank C with $m \geq n$, the convergence rate is determined by the Lipschitz constant

$$\| \, I - C^\dagger \mathbf{f}(\overline{\mathbf{q}}(\cdot)) \| \,.$$

If we consider the Euclidean norm, and the operators $\mathbf{f} : X \to Y$ and $\overline{\mathbf{q}} : Z \to X$ have Jacobians $F \in \mathbb{R}^{m\times n}$ and $\overline{Q} \in \mathbb{R}^{n\times n}$ respectively, then the convergence rate is bounded by

$$\|I - C^\dagger F\overline{Q}\|_2 = \|(C^T C)^{-1} C^T \left(C - F\overline{Q}\right)\|_2 \leq \|(C^T C)^{-1}\|_2 \, \|C^T \left(C - F\overline{Q}\right)\|_2 \,.$$

This shows how the operators $\mathbf{c}$ and $\mathbf{f} \circ \overline{\mathbf{q}}$ should be related for the iteration process to converge: the coarse model $\mathbf{c}$ should be full rank and sufficiently stable

(well-conditioned) and the fine model $\mathbf{f}$, preconditioned by $\overline{\mathbf{q}}$, should sufficiently correspond with the coarse model $\mathbf{c}$ (at least with respect to the components in its range).

Left- and right-preconditioning

The efficiency of the space-mapping algorithms depends essentially on the similarity between the fine and the coarse model used and their relative computational costs for solution. The space-mapping procedure is then determined by the choice how the space-mapping function is approximated, with ASM and TRASM as obvious choices.

In the defect correction algorithms we also have to deal with the similarity between the fine and the coarse model but, instead of in the selection of the space-mapping approximation, there the freedom is in the preconditioning mapping $\overline{\mathbf{p}}$ or $\overline{\mathbf{q}} = \overline{\mathbf{p}}^{-1}$. This choice can be used to improve the convergence of the procedure. From the linear case we see that better convergence can be expected if the approximation $\mathbf{c}(\overline{\mathbf{p}}(\mathbf{x})) \approx \mathbf{f}(\mathbf{x})$ is more accurate. We can consider $\overline{\mathbf{p}}$ as a right-preconditioner [88] of $\mathbf{c}$ to better approximate $\mathbf{f}$.

On the other hand, by the orthogonality relations in the lemmas above, we see that it is advantageous if the manifolds $\mathbf{f}(X)$ and $\mathbf{c}(Z)$ are found parallel in the neighborhood of the solution. However, by space mapping or by right-preconditioning the relation between the manifolds $\mathbf{f}(X)$ and $\mathbf{c}(Z)$ remains unchanged. This relation can be improved by the effect of an additional left-preconditioner. Therefore, we have to introduce such a preconditioner $\mathbf{r}$ so that near $\mathbf{f}(\mathbf{x}^*) \in Y$ the manifold $\mathbf{c}(Z) \subset Y$ is mapped onto $\mathbf{f}(X) \subset Y$:

$$\mathbf{r}(\mathbf{c}(\overline{\mathbf{p}}(\mathbf{x}))) \approx \mathbf{f}(\mathbf{x}) .$$

Hence, in the next section we propose such an operator $\mathbf{S}$, which maps $\mathbf{c}(Z)$ onto $\mathbf{f}(X)$, at least in the neighborhood of the solution. This improves significantly the traditional approach. The recently introduced output space mapping and space-mapping based interpolating surrogate scheme [15] also take such left-preconditioning into account, but in a different manner.

2.5.4 Improving space mapping

Except for the hybrid algorithm HASM, all space-mapping methods thus far have the clear disadvantage that, in general, the fixed point of the iteration does not necessarily coincide with the solution of the fine model minimization problem. This is due to the fact that the approximate solution $\overline{\mathbf{x}}$ satisfies

$$\begin{aligned} \mathbf{f}(\overline{\mathbf{x}}) - \mathbf{y} &\in \mathbf{c}(\overline{\mathbf{p}}(X))^{\perp}(\overline{\mathbf{p}}^{-1}\mathbf{p}(\overline{\mathbf{x}})) && \text{for DeC-X and DeC-Z,} \\ \mathbf{f}(\overline{\mathbf{x}}) - \mathbf{y} &\in \mathbf{c}(\overline{\mathbf{p}}(X))^{\perp}(\overline{\mathbf{x}}) && \text{for DeC-Y,} \end{aligned}$$

whereas $\mathbf{x}^*$, a (local) minimum for $\|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|$, satisfies

$$\mathbf{f}(\mathbf{x}^*) - \mathbf{y} \in \mathbf{f}(X)^{\perp}(\mathbf{x}^*) .$$

Hence, differences between $\overline{\mathbf{x}}$ and $\mathbf{x}^*$ will be larger for larger distances between $\mathbf{y}$ and the sets $\mathbf{f}(X)$ and $\mathbf{c}(Z)$, and for larger angles between the linear manifolds tangential at $\mathbf{c}(Z)$ and $\mathbf{f}(X)$ near the optima. This disadvantage, however, can be removed by introducing a mapping $\mathbf{S} : Y \to Y$ such that, for a proper $\overline{\mathbf{z}} \in Z$, $\mathbf{S}\,\mathbf{c}(\overline{\mathbf{z}}) = \mathbf{f}(\mathbf{x}^*)$ and the tangent plane for $\mathbf{c}(Z)$ at $\mathbf{c}(\overline{\mathbf{z}})$ is mapped onto the tangent plane for $\mathbf{f}(X)$ at $\mathbf{f}(\mathbf{x}^*)$. Because, in the non-degenerate case when $m \geq n$, both $\mathbf{f}(X)$ and $\mathbf{c}(Z)$ are n-dimensional manifolds in $\mathbb{R}^m$, $\mathbf{S}$ can be described by an affine mapping

$$\mathbf{S}\,\bullet = \mathbf{f}(\mathbf{x}^*) + S\ (\bullet - \mathbf{c}(\overline{\mathbf{z}}))\ ,$$

where S is an $m \times m$-matrix. This approach was introduced in [35] and is the basis of the *manifold-mapping* technique. We will see in the next chapter that the mapping $\mathbf{S}$ depends on the solution of the problem $\mathbf{x}^*$ and, thus, that it is not a priori available. Manifold-mapping algorithms iteratively approximate $\mathbf{S}$ by $\mathbf{S}_k : Y \to Y$ given by

$$\mathbf{S}_k\ \bullet = \mathbf{f}(\mathbf{x}_k) + S_k(\bullet - \mathbf{c}(\overline{\mathbf{p}}(\mathbf{x}_k))) , \tag{2.42}$$

with $\overline{\mathbf{p}} : X \to Z$ being a (known) arbitrary mapping and S_k a proper $m \times m$ matrix. We can write the next manifold-mapping iterant as

$$\mathbf{x}_{k+1} = \underset{\mathbf{x} \in X}{\operatorname{argmin}} \|\mathbf{S}_k(\mathbf{c}(\overline{\mathbf{p}}_k(\mathbf{x}))) - \mathbf{y}\| , \tag{2.43}$$

and then, because $\mathbf{S}_k\,\mathbf{c}(\overline{\mathbf{p}}_k(\mathbf{x}_k)) = \mathbf{f}(\mathbf{x}_k)$, we can also interpret it as a defect correction iteration with either $\widetilde{\mathcal{F}}_k = \mathbf{S}_k \circ \mathbf{c} \circ \overline{\mathbf{p}}_k$ and $\mathcal{F} = \mathbf{f}$ in (2.30) or (2.31), or with $\widetilde{\mathcal{F}}_k = \mathbf{S}_k \circ \mathbf{c}$ and $\mathcal{F} = \mathbf{f} \circ \overline{\mathbf{q}}_k$ in (2.33) or (2.34).

In Chapter 4 we will see that, under mild assumptions, if a manifold-mapping based algorithm converges to $\overline{\mathbf{x}}$, this fixed point is the fine model optimum $\mathbf{x}^*$. Convergence of these manifold-mapping based schemes will be studied further in that chapter.

2.6 Simple examples and counterexamples

Coinciding space-mapping optima and deviation from perfect mapping

Each of the four points $\mathbf{z}^*$, $\mathbf{x}_p^*$, $\mathbf{x}_d^*$ and $\mathbf{x}^*$ introduced in the previous sections is the solution of a minimization problem, the first three being approximations to the last. In this section by a few very simple examples we illustrate that in some cases these approximations coincide, but in other cases they can be all different.

As a first example we consider a simple least squares data fitting problem: the data are $\mathbf{y} = [y_{-1}, y_0, y_1]$ for $\mathbf{t} = [-1, 0, 1]$, so we have $m = 3$. The fine model is the family of polynomials over $X = \mathbb{R}^2$,

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}(x_1, x_2) = \left[x_1^2 t_i^2 + x_1 t_i + x_2\right]_{i=1,2,3} . \tag{2.44}$$

The coarse model is the linear $\mathbf{c}(\mathbf{z}) = \mathbf{c}(z_1, z_2) = z_1 t_i + z_2$. The number of control parameters is $n = 2$ and the two models are similar for small x_1. The search is made over $X = Z = \mathbb{R}^2$. It is immediate that the coarse model minimum is

$$\mathbf{z}^* = \left[\frac{y_1 - y_{-1}}{2}, \frac{y_{-1} + y_0 + y_1}{3}\right] . \tag{2.45}$$

Using the definition of the space-mapping function, it is easily determined as

$$\mathbf{p}(\mathbf{x}) = \left[x_1, x_2 + \frac{2}{3}x_1^2\right] ,$$

which for small x_1 clearly resembles the identity operator. It is a bijection between X and Z and by applying Lemma 2 in Section 2.3 we see that for this problem all the space-mapping solutions exist and coincide

$$\mathbf{x}_{sm}^* = \mathbf{x}_p^* = \mathbf{x}_d^* = \left[\frac{y_1 - y_{-1}}{2}, \frac{y_{-1} + y_0 + y_1 - (y_1 - y_{-1})^2/2}{3}\right] . \tag{2.46}$$

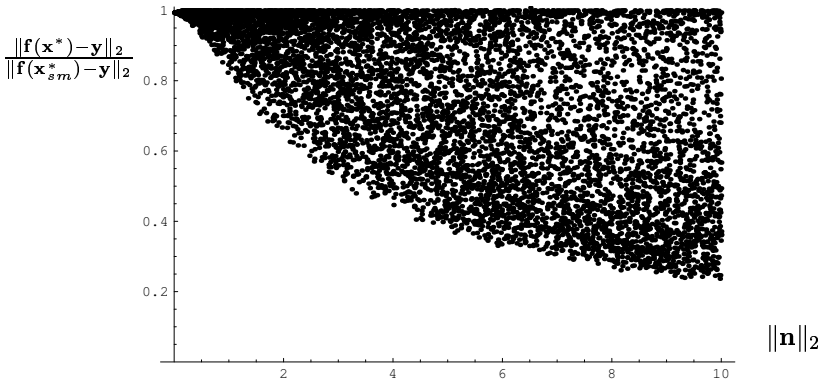
Notice that this is not the true solution for fitting with (2.44). In the case of perfect mapping the solution $\mathbf{x}_{sm}^*$ corresponds with the true optimum $\mathbf{x}^*$.

Deviation from perfect mapping. In order to check the value of the space-mapping approach and see the effect of a non-perfect mapping, we perform the following experiment. We create the data $\mathbf{y} = \mathbf{f}(\hat{\mathbf{x}}) + \mathbf{n}$, with $\hat{\mathbf{x}} = [0.1, 0.1]$ and $\mathbf{n}$ a random vector, uniformly distributed on a sphere with radius $\|\mathbf{n}\|_2 = r$. Figure 2.10 shows the quotient $\|\mathbf{f}(\mathbf{x}^*) - \mathbf{y}\|_2 / \|\mathbf{f}(\mathbf{x}_{sm}^*) - \mathbf{y}\|_2$. When $\mathbf{n}$ is not significant compared to the fine model response, then the mapping is almost perfect and the space-mapping solution yields a response with a cost function close to the true optimum. With this test we corroborate that a coarse model that can accurately reproduce the response of the fine one is a good candidate for being used within space mapping, and that fine models that meet the specifications well improve the quality of the space-mapping result.

Different optima $\mathbf{x}^*$, $\mathbf{x}_p^*$ and $\mathbf{x}_d^*$

However, in general, if $\mathbf{z}^* \notin \mathbf{p}(X)$ the solutions $\mathbf{x}_p^*$ and $\mathbf{x}_d^*$ are expected to be different. By increasing the misalignment between the models, the condition

Figure 2.10: Quality of the space-mapping optimum depending on the deviation from perfect mapping.



Results for 8000 experiments with random perturbations of size $\|\mathbf{n}\|_2 < 10$. In the example $\|\mathbf{f}(\widehat{\mathbf{x}})\|_2 = 0.2328$

$\mathbf{z}^* \in \mathbf{p}(X)$ may no longer be satisfied. This is shown by taking another simple fine model:

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}(x_1, x_2) = [x_1\,(x_2\,t_i + 1)^2]_{i=1,2,3}\,. \tag{2.47}$$

We keep the linear coarse model $\mathbf{c}(\mathbf{z}) = \mathbf{c}(z_1, z_2) = z_1\mathbf{t} + z_2$. Proceeding as above, the space-mapping function can be determined as

$$\mathbf{p}(\mathbf{x}) = \left[2\,x_1\,x_2\,, x_1\,(1 + \frac{2}{3}\,x_2^2)\right]. \tag{2.48}$$

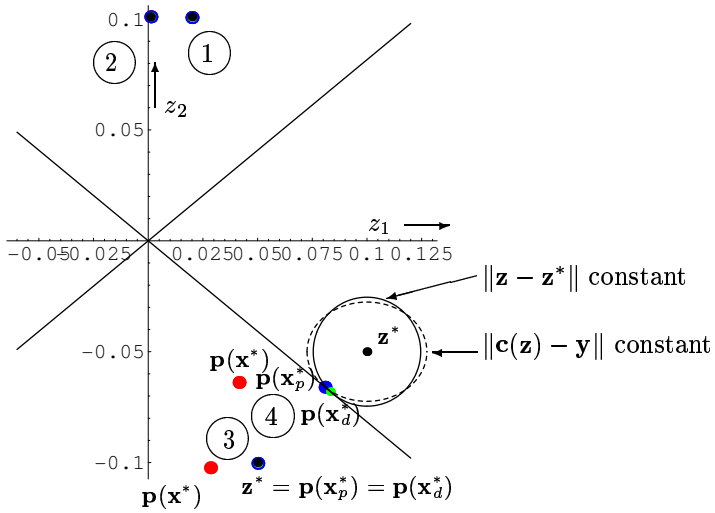
This mapping is not injective and is substantially different from the identity. Furthermore, $\mathbf{p}(X) \neq \mathbb{R}^2$ and it is easily verified that $\mathbf{p}(X) = \{\mathbf{z} \in Z;\, 3\,z_2^2 \geq 2\,z_1^2\}$.

For this example we can distinguish four different situations: (1) the design is reachable, (2) the mapping is perfect but the design is not reachable, (3) the primal and the dual solution correspond but the mapping is not perfect, and (4) the four points $\mathbf{z}^*$, $\mathbf{p}(\mathbf{x}_p^*)$, $\mathbf{p}(\mathbf{x}_d^*)$ and $\mathbf{p}(\mathbf{x}^*)$ are all different.

These four cases are shown in Figure 2.11 and in Table 2.1. Case (1) is characterized by the fact that the cost function for the fine model vanishes at $\mathbf{x}_p^* = \mathbf{x}_d^* = \mathbf{x}^*$. This occurs if the data $\mathbf{y}$ fit the model (2.47) so that a reachable design is obtained. In case (2) we also see $\mathbf{x}_p^* = \mathbf{x}_d^* = \mathbf{x}^*$, but the cost function for the fine model does not vanish. In case (3) still $\mathbf{z}^* \in \mathbf{p}(X)$, but the data are such that $\mathbf{p}$ is no perfect mapping, so that $\mathbf{x}_p^* = \mathbf{x}_d^*$ but they do not coincide with $\mathbf{x}^*$. For the case (4) the data $\mathbf{y}$ are selected such that $\mathbf{z}^* \notin \mathbf{p}(X)$. Now $\mathbf{x}_p^* \neq \mathbf{x}_d^*$ and, in fact, all values differ.

Because of the quadratic form in $\mathbf{p}(\mathbf{x})$, each of the minimization procedures for the primal and the dual approach yields two solutions. In Table 2.1 we omit

Four situations: (1) and (2): coinciding $\mathbf{z}^* = \mathbf{p}(\mathbf{x}_p^*) = \mathbf{p}(\mathbf{x}_d^*) = \mathbf{p}(\mathbf{x}^*)$;
(3): coinciding $\mathbf{z}^* = \mathbf{p}(\mathbf{x}_p^*) = \mathbf{p}(\mathbf{x}_d^*)$, separate $\mathbf{p}(\mathbf{x}^*)$;
(4): separate $\mathbf{z}^*$, $\mathbf{p}(\mathbf{x}_p^*)$, $\mathbf{p}(\mathbf{x}_d^*)$ and $\mathbf{p}(\mathbf{x}^*)$.



The top and bottom quadrant are $\mathbf{p}(X)$. In case (4) $\mathbf{z}^* \notin \mathbf{p}(X)$.

Figure 2.11: Different space-mapping solutions for problem (2.47).

the one that has the larger fine cost function.

The optimization problem that defines the primal approach finds the point in $\mathbf{p}(X)$ that is closest in the Euclidean norm to $\mathbf{z}^*$. It lies on a circle centered in $\mathbf{z}^*$, with minimum radius so that it has one point of contact with $\mathbf{p}(X)$. For the dual approach the contour lines for the cost function are ellipses in $\mathbf{p}(X)$ centered in $\mathbf{z}^*$. Therefore, $\mathbf{p}(\mathbf{x}_p^*)$ and $\mathbf{p}(\mathbf{x}_d^*)$, and consequently $\mathbf{x}_p^*$ and $\mathbf{x}_d^*$, do not coincide.

Remark. Case (4) has two local optima. The behavior of the fine cost function in the neighborhood of one of them is similar to that of the coarse cost function close to the coarse optimum. But for the other one, the global fine optimum, the difference is substantial. Space mapping finds solutions close to the local optimum with the similar behavior (see Figure 2.12). This observation emphasizes the importance of the similarity between the models used.

The importance of the use of similar models

Now follows an example showing how strong dissimilarities between the models make both space-mapping approaches fail. For the fine model we take the family

Table 2.1: Different space-mapping solutions for the least squares data fitting problem associated with (2.47).

	y_{-1}	y_0	y_1	$\mathbf{z}^*$	$C(\mathbf{z}^*)$	$F(\mathbf{z}^*)$
1	0.08100	0.10000	0.12100	[0.020,0.101]	8.16e-04	1.41e-01
2	0.10011	0.10125	0.10241	[0.001,0.101]	8.16e-06	1.73e-01
3	0.00000	-0.40000	0.10000	[0.050,-0.100]	3.67e-01	4.58e-01
4	0.00000	-0.35000	0.20000	[0.100,-0.050]	3.67e-01	4.76e-01

	$\mathbf{x}^*$	$\mathbf{x}_p^*$	$\mathbf{x}_d^*$	$F(\mathbf{x}^*)$	$F(\mathbf{x}_p^*)$	$F(\mathbf{x}_d^*)$
1	[0.100,0.100]	[0.100,0.100]	[0.100,0.100]	1.96e-17	3.10e-17	1.96e-17
2	[0.101,0.006]	[0.101,0.006]	[0.101,0.006]	5.50e-06	5.50e-06	5.50e-06
3	[-0.101,-0.141]	[-0.096,-0.261]	[-0.096,-0.261]	3.70e-01	3.73e-01	3.73e-01
4 a	[-0.059,-0.352]	[-0.034,-1.225]	[-0.033,-1.225]	3.83e-01	4.12e-01	4.10e-01
b	[0.007,4.007]			3.63e-01		

$$C(\mathbf{z}) = \|\mathbf{c}(\mathbf{z}) - \mathbf{y}\|_2 \text{ and } F(\mathbf{x}) = \|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|_2.$$

of exponentials

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}(x_1, x_2) = [x_1 \exp(x_2\, t_i)]_{i=1,2,3} \ ,$$

and the coarse one is linear as before. In both cases $X = Z = \mathbb{R}^2$. The space-mapping function is again uniquely determined

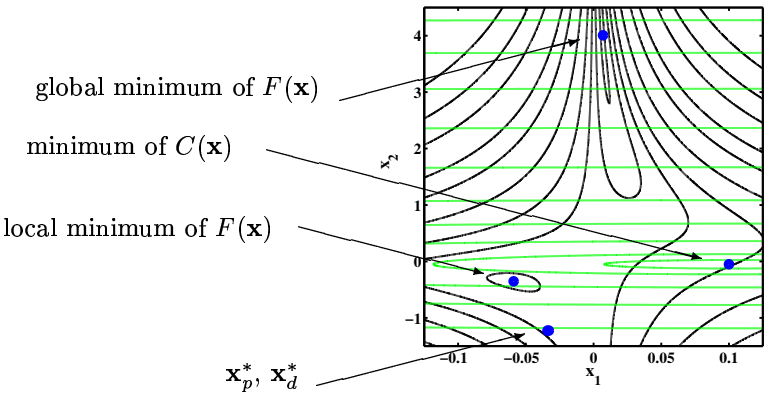
$$\mathbf{p}(\mathbf{x}) = \left[x_1 \sinh(x_2)\,, \frac{x_1}{3}\,(1 + 2\,\cosh(x_2)\right] \ .$$

Now $\mathbf{p}(X) = \{\mathbf{z} \in Z;\, 9\, z_2^2 > 4\, z_1^2\}$ is an open set and, thus, in the case $\mathbf{z}^* \notin \mathbf{p}(X)$, the primal and dual space-mapping solutions cannot be found because no minimum exists for the minimization problems that define these approximations. The fine and the coarse model have little in common, to the extent that for certain specifications the space-mapping technique cannot be applied. The conclusion is that in order to have a proper space-mapping scheme, it is crucial to use models that are sufficiently similar.

Preference for the dual approach

The dual space-mapping solution for case (4) in Table 2.1 above has a smaller cost function associated than the primal one. Additional experiments with the same problem were carried out in order to check if this better performance can generally be expected, but the results were inconclusive. In [82, section 4.1.2] the following lemma in favor of the dual approach is given and checked experimentally.

Figure 2.12: Level curves and minima of the fine and coarse cost functions for the case (4) in Table 2.1. The solutions found by space-mapping, $\mathbf{x}_p^*$ and $\mathbf{x}_d^*$, correspond with the local optimum for the fine model, which shows similarity with the coarse model near $\mathbf{z}^*$.



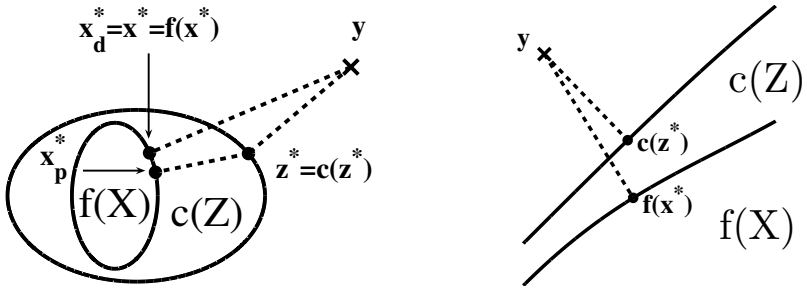
Lemma 17. *If* $\exists\, \epsilon, \delta > 0$, $\epsilon \leq \delta$ *such that*
(i) $\| \mathbf{c}(\mathbf{p}(\mathbf{x}_p^*)) - \mathbf{f}(\mathbf{x}_p^*) \| \leq \epsilon/2$, $\| \mathbf{c}(\mathbf{p}(\mathbf{x}_d^*)) - \mathbf{f}(\mathbf{x}_d^*) \| \leq \epsilon/2$, *and*
(ii) $\| \mathbf{c}(\mathbf{p}(\mathbf{x}_p^*)) - \mathbf{y} \| - \| \mathbf{c}(\mathbf{p}(\mathbf{x}_d^*)) - \mathbf{y} \| \geq \delta$
then $\| \mathbf{f}(\mathbf{x}_d^*) - \mathbf{y} \| \leq \| \mathbf{f}(\mathbf{x}_p^*) - \mathbf{y} \|$. □

We give two theoretical examples supporting a choice in favor of the dual approach, specially in the case where the flexibility concept can be used.

Example I. In this example X is a non-trivial subset of Z. On X the fine and coarse model responses are taken equal. Then the space-mapping function is the identity and the mapped coarse model matches the fine one, i.e., $\mathbf{c}(\mathbf{p}(\mathbf{x})) = \mathbf{f}(\mathbf{x})$ $\forall \mathbf{x} \in X$ (see Lemma 3 (i) Section 2.3). Under these circumstances the dual solution $\mathbf{x}_d^*$ coincides with the fine optimum $\mathbf{x}^*$. But, in general, this is not the case for the primal approach, as can be seen for the particular sets X and Z shown in Figure 2.13 (left), where $\mathbf{c}$ is the identity, the Euclidean norm is used, and with $\mathbf{y} \in Y \setminus \mathbf{c}(Z)$ we check that $\mathbf{x}_p^* \neq \mathbf{x}^*$. This argument, based on projection onto subsets, is more general and can be used in less obvious situations.

Example II. In Figure 2.13 (right) we analyze the possibility of $\mathbf{x}_p^* = \mathbf{x}^* \neq \mathbf{x}_d^*$ with $\mathbf{c}$ (locally) more flexible than $\mathbf{f}$ in the region near the optimal responses. This means that we assume the existence of a set $R \subset \mathbb{R}^m$ such that $R \supset R \cap \mathbf{c}(Z) \supset R \cap \mathbf{f}(X)$ and that $\mathbf{c}(\mathbf{p}(\mathbf{x}_d^*)), \mathbf{f}(\mathbf{x}^*), \mathbf{c}(\mathbf{z}^*) \in R$. We define the auxiliary set $Q = \{\mathbf{x} \in X | \mathbf{f}(\mathbf{x}) \in R\}$. By Lemma 3 (i) Section 2.3, applied lo-

Figure 2.13: Left : A simple counterexample for $\mathbf{x}_p^* \neq \mathbf{x}^* = \mathbf{x}_d^*$. The Euclidean norm is taken as cost function. Right: Usually the fine and coarse models are similar enough near the optima that $\mathbf{c}(\mathbf{z}^*) \notin \mathbf{f}(X)$ implies a behavior as shown.



cally, we know $\mathbf{c}(\mathbf{p}(\mathbf{x})) = \mathbf{f}(\mathbf{x})$ for all $\mathbf{x} \in Q$, consequently

$$\begin{aligned} \mathbf{x}^* &= \operatorname{argmin}_{\mathbf{x}\in X} \| \mathbf{f}(\mathbf{x}) - \mathbf{y} \| = \operatorname{argmin}_{\mathbf{x}\in Q} \| \mathbf{f}(\mathbf{x}) - \mathbf{y} \| \\ &= \operatorname{argmin}_{\mathbf{x}\in Q} \| \mathbf{c}(\mathbf{p}(\mathbf{x})) - \mathbf{y} \| = \operatorname{argmin}_{\mathbf{x}\in X} \| \mathbf{c}(\mathbf{p}(\mathbf{x})) - \mathbf{y} \| = \mathbf{x}_d^*, \end{aligned}$$

i.e., the dual space-mapping approach also yields the fine optimum. We conclude that in this case the equality $\mathbf{x}_p^* = \mathbf{x}^*$ implies $\mathbf{x}_d^* = \mathbf{x}^*$.

The result does not hold for a more flexible fine model in R. Since $\mathbf{x}_p^* \neq \mathbf{x}_d^*$, by Lemma 2 (Section 2.3) the coarse optimum $\mathbf{z}^*$ cannot be in $\mathbf{p}(X)$. From $\mathbf{z}^* \notin \mathbf{p}(X)$ it follows that $\mathbf{c}(\mathbf{z}^*) \notin \mathbf{f}(X)$, otherwise $\mathbf{c}(\mathbf{z}^*) = \mathbf{f}(\hat{\mathbf{x}})$ for some $\hat{\mathbf{x}} \in X$, would yield $\mathbf{z}^* = \mathbf{p}(\hat{\mathbf{x}})$. Because $\mathbf{c}(\mathbf{z}^*) \in R$ we have a clear contradiction with the fine model being more flexible than the coarse one in R.

2.7 Conclusions

The space-mapping technique aims at accelerating expensive optimization procedures by combining problem descriptions with different degrees of accuracy. In numerical analysis, for the solution of operator equations, the same principle is known as defect correction iteration. Taking the defect correction principle as a starting point, space mapping can be interpreted, extended and analyzed.

Introducing the concept of flexibility of a model, important properties of space mapping can be formulated in these terms, and better understood. Equally flexible models are most desirable in space mapping, but this property does not generally hold for every pair of problem formulations used in practice. As a consequence, the solution found by traditional space mapping may differ from the solution for the fine model. By introducing proper left-preconditioning we can improve the flexibility of the models. Thus, we derive manifold mapping,

a new improved space-mapping algorithm which yields the accurate solution (a precise description and analysis of manifold-mapping will be given in the next two chapters). The phenomena analyzed in this chapter are eventually illustrated by a few simple examples. Practical design problems that confirm the efficiency of the space-mapping/defect correction approach in optimization can be found in Chapters 5 and 6.

Chapter 3

The Manifold-Mapping Technique

3.1 Introduction

In Chapter 2 we used the concept of flexibility and defect correction for explaining why the standard space-mapping technique may in general fail to find the right optimum. That analysis led to manifold mapping, an improved space-mapping approach. Manifold mapping is an alternative surrogate-based optimization technique that can be used without computing *exact* gradient information and that shows, as will be discussed in Chapter 4, provable convergence to the right solution. This chapter starts a thorough analysis of the manifold-mapping approach and it is structured as follows. In Section 3.2 some additional assumptions are presented and the basic algorithms are derived. In the next chapter it will be seen that, under mild assumptions, the fixed point for all those manifold-mapping based algorithms is the accurate optimum (convergence theorems and illustrations by means of simple design problems will also be given there). The basic schemes will be extended in Section 3.3 with a trust-region strategy and the possibility of handling complex optimization constraints. Manifold mapping and space mapping can also be based on more than two models and can be applied when the dimensions of the fine and coarse design spaces, X and Z, respectively, differ. In Section 3.4 we will briefly study under what circumstances it makes sense to consider such possibilities for efficient optimization alternatives. In Chapters 5 and 6, the manifold-mapping technique will be eventually compared with other efficient methods used in practice for optimal design problems.

3.2 Manifold-Mapping algorithms

Here we introduce the basic manifold-mapping algorithms. The notation is the same as in Chapter 1. We have already commented that the case $n \geq m$ usually implies that the optimization problem can be analyzed and solved as a nonlinear system, since there are at least as many design variables as specifications. For this reason, manifold-mapping theory will be studied for the overdetermined case $n < m$. We formulate this as the following assumption.

Assumption 4. *The dimension of the space of possible aims exceeds the dimension of the control space: $n < m$.*

For simplicity we will consider the case that the fine and coarse model design spaces coincide, i.e., $X = Z$. With $X = Z$, we prefer to write $\mathbf{c}(\mathbf{x})$ and $\mathbf{x}_\mathbf{c}^*$ instead of $\mathbf{c}(\mathbf{z})$ and $\mathbf{z}^*$, respectively. The fine model optimum will be denoted as $\mathbf{x}_\mathbf{f}^*$.

The analysis in this chapter will be only for those design cases in which the function $\mathbf{k_f}(\mathbf{x})$ is easy to compute. Below in Section 3.3.2, we show how more complex constraints can be handled within the manifold-mapping framework.

If the equality constraints $\mathbf{k_f}(\mathbf{x}) \in \mathbb{R}^{n_\mathbf{k}}$ can be easily evaluated and a chart φ_X for $\overline{X} = \{\mathbf{x} \in \mathbb{R}^n;\ \mathbf{k_f}(\mathbf{x}) = 0\}$ in the region of interest can be obtained without significant computational cost, then the constrained optimization problem can be restated with $\mathbf{f} \circ \varphi_X$ as fine model and a subset of $\mathbb{R}^{n-n_\mathbf{k}}$ as the control space.

By the above argument we first assume that the equality constraints can be eliminated and, hence, that X can be considered as a subset of $\mathbb{R}^n$, understanding n as the number of design variables left after removing the equality constraints. As a consequence, both Jacobians $J_\mathbf{f}(\mathbf{x})$ and $J_\mathbf{c}(\mathbf{x})$ have both rank n.

3.2.1 Original manifold mapping

In [35] the *manifold mapping* $\mathbf{S} : \mathbf{c}(X) \to \mathbf{f}(X)$ is introduced with the aim of correcting the misalignment between the manifolds $\mathbf{f}(X)$ and $\mathbf{c}(X)$. By $\mathbf{S}$, the point $\mathbf{c}(\mathbf{x}_\mathbf{f}^*)$ is mapped to $\mathbf{f}(\mathbf{x}_\mathbf{f}^*)$ and the tangent plane for $\mathbf{c}(X)$ at $\mathbf{c}(\mathbf{x}_\mathbf{f}^*)$ to the tangent plane for $\mathbf{f}(X)$ at $\mathbf{f}(\mathbf{x}_\mathbf{f}^*)$ (see Figure 3.1). Other approaches are possible but in this work we define $\mathbf{S}$ as the affine mapping

$$\mathbf{S}\,\mathbf{c}(\mathbf{x}) = \mathbf{f}(\mathbf{x}_\mathbf{f}^*) + \overline{S}\,(\mathbf{c}(\mathbf{x}) - \mathbf{c}(\mathbf{x}_\mathbf{f}^*))\,, \tag{3.1}$$

where

$$\overline{S} = J_\mathbf{f}(\mathbf{x}_\mathbf{f}^*)\,J_\mathbf{c}^\dagger(\mathbf{x}_\mathbf{f}^*)\,. \tag{3.2}$$

Here the pseudoinverse, denoted by †, is defined by $J_\mathbf{c}^\dagger(\mathbf{x}_\mathbf{f}^*) = V_\mathbf{c}\,\Sigma_\mathbf{c}^\dagger\,U_\mathbf{c}^T$, being $U_\mathbf{c}$, $\Sigma_\mathbf{c}$ and $V_\mathbf{c}$ the factors in the singular value decomposition (SVD) of $J_\mathbf{c}(\mathbf{x}_\mathbf{f}^*) = U_\mathbf{c}\,\Sigma_\mathbf{c}\,V_\mathbf{c}^T$. The matrix $\Sigma_\mathbf{c}^\dagger$ is the result of inverting the nonzero entries in $\Sigma_\mathbf{c}$, leaving the zeroes invariant. It should be noted that, because $J_\mathbf{f}(\mathbf{x}_\mathbf{f}^*)$ and $J_\mathbf{c}(\mathbf{x}_\mathbf{f}^*)$ are both full rank, also the $m \times m$ matrix $\overline{S}$ has rank n.

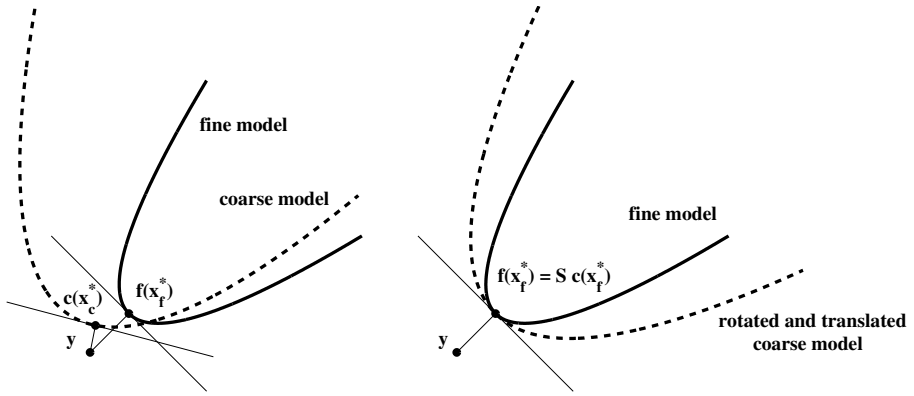


Figure 3.1: Manifold-Mapping model alignment.

$\mathbf{x}_0 = \mathbf{x}_\mathbf{c}^* = \text{argmin}_{\mathbf{x}\in X}\|\mathbf{c}(\mathbf{x}) - \mathbf{y}\|$;
$\mathbf{S}_0 \bullet = \mathbf{f}(\mathbf{x}_0) + (\bullet - \mathbf{c}(\mathbf{x}_0))$;
for $k = 0, 1, \ldots,$ **while** ...
do
$\mathbf{x}_{k+1} = \text{argmin}_{\mathbf{x}\in X}\|\mathbf{S}_k(\mathbf{c}(\mathbf{x})) - \mathbf{y}\|$;
break if ...
$\Delta F = [\mathbf{f}(\mathbf{x}_{k+1}) - \mathbf{f}(\mathbf{x}_k), \cdots, \mathbf{f}(\mathbf{x}_{k+1}) - \mathbf{f}(\mathbf{x}_{\max(k+1-n,0)})]$;
$\Delta C = [\mathbf{c}(\mathbf{x}_{k+1}) - \mathbf{c}(\mathbf{x}_k), \cdots, \mathbf{c}(\mathbf{x}_{k+1}) - \mathbf{c}(\mathbf{x}_{\max(k+1-n,0)})]$;
$(U_{\Delta C}, \Sigma_{\Delta C}, V_{\Delta C}) = \text{SVD}(\Delta C)$;
$\Delta C^\dagger = V_{\Delta C}\, \Sigma_{\Delta C}^\dagger\, U_{\Delta C}^T$;
$S_{k+1} = \Delta F\, \Delta C^\dagger$;
$\mathbf{S}_{k+1} \bullet = \mathbf{f}(\mathbf{x}_{k+1}) + S_{k+1}\,(\bullet - \mathbf{c}(\mathbf{x}_{k+1}))$;
enddo

Figure 3.2: The original manifold-mapping (OMM) algorithm.

The manifold-mapping solution $\mathbf{x}_\mathbf{mm}^*$ is defined as

$$\mathbf{x}_\mathbf{mm}^* = \text{argmin}_{\mathbf{x}\in X}\|\mathbf{S}\,\mathbf{c}(\mathbf{x}) - \mathbf{y}\|\,. \tag{3.3}$$

Now the combination $\mathbf{S} \circ \mathbf{c}$ acts as the *surrogate model* for $\mathbf{f}$ and under a few assumptions about the similarity between the fine and the coarse model that usually hold in practice, it will be shown in the next chapter that the manifold-mapping solution $\mathbf{x}_\mathbf{mm}^*$ is a local minimizer of the fine cost function.

The mapping $\mathbf{S}$ is not known a priori, because it depends on the solution of the optimization problem. We propose an algorithm (Figure 3.2) that, when it converges, yields both the mapping $\mathbf{S}$ and the desired fine model optimum $\mathbf{x}_\mathbf{f}^*$.

$\mathbf{x}_0 = \mathbf{x}_\mathbf{c}^* = \text{argmin}_{\mathbf{x}\in X}\|\mathbf{c}(\mathbf{x}) - \mathbf{y}\|$;
$T_0 = I_{m\times m}$;
for $k = 0, 1, \ldots,$ **while** ...
do
$\mathbf{y}_k = \mathbf{c}(\mathbf{x}_k) - T_k\,(\mathbf{f}(\mathbf{x}_k) - \mathbf{y})$;
$\mathbf{x}_{k+1} = \text{argmin}_{\mathbf{x}\in X}\|\mathbf{c}(\mathbf{x}) - \mathbf{y}_k\|$;
break if ...
$\Delta F = [\mathbf{f}(\mathbf{x}_{k+1}) - \mathbf{f}(\mathbf{x}_k),\ \cdots,\ \mathbf{f}(\mathbf{x}_{k+1}) - \mathbf{f}(\mathbf{x}_{\max(k+1-n,0)})]$;
$\Delta C = [\mathbf{c}(\mathbf{x}_{k+1}) - \mathbf{c}(\mathbf{x}_k),\ \cdots,\ \mathbf{c}(\mathbf{x}_{k+1}) - \mathbf{c}(\mathbf{x}_{\max(k+1-n,0)})]$;
$(U_{\Delta F},\ \Sigma_{\Delta F},\ V_{\Delta F}) = \text{SVD}(\Delta F)$;
$\Delta F^\dagger = V_{\Delta F}\,\Sigma_{\Delta F}^\dagger\,U_{\Delta F}^T$;
$T_{k+1} = \Delta C\,\Delta F^\dagger$;
enddo

Figure 3.3: The manifold-mapping (MM) algorithm. Note that $T_k = S_k^\dagger$ for all $k > 0$.

We will refer to this scheme as the *original manifold-mapping* (OMM) algorithm. The model alignment can be improved by an additional (right-preconditioning) mapping $\overline{\mathbf{p}} : X \to X$. This mapping $\overline{\mathbf{p}}$ is optional in the present case, but it is obligatory in the case when the fine and coarse design spaces X and Z are different. For simplicity, in the algorithm in Figure 3.2 we take $\overline{\mathbf{p}} = I$, the identity. The optimization procedure needed to compute $\mathbf{x}_{k+1}$ is not essentially different from the one to obtain the coarse model optimum $\mathbf{x}_\mathbf{c}^*$. Therefore, we may expect that optimization problem to be well defined in each iteration step. However, formally this has to be introduced as an assumption.

Assumption 5. *The minimization*

$$\mathbf{x}_{k+1} = \text{argmin}_{\mathbf{x}\in X}\|\mathbf{S}_k(\mathbf{c}(\mathbf{x})) - \mathbf{y}\| \tag{3.4}$$

is well defined for every k.

This assumption is the surrogate equivalent of Assumption 3.

Remark 2. *The original manifold-mapping algorithm can be extended to any such algorithm based on a matrix $\overline{S}$ satisfying $\overline{S}\,J_\mathbf{c}(\mathbf{x}_\mathbf{f}^*) = J_\mathbf{f}(\mathbf{x}_\mathbf{f}^*)$. There is complete freedom how $\overline{S}$ handles components in the complement of the range of $U_\mathbf{c}$. The general case is $\overline{S} = J_\mathbf{f}(\mathbf{x}_\mathbf{f}^*)\,J_\mathbf{c}^\dagger(\mathbf{x}_\mathbf{f}^*) + A\,(I - U_\mathbf{c}\,U_\mathbf{c}^T)$ with A any $m\times m$ matrix. This freedom can be used to stabilize the algorithm.*

3.2.2 Manifold mapping

In the OMM algorithm, $\mathbf{S}_k \circ \mathbf{c}$ is used as the surrogate model, i.e., it is updated during iteration, and the aim $\mathbf{y}$ is kept constant. From an implementational point

$\mathbf{x}_0 = \mathbf{x}_\mathbf{c}^* = \text{argmin}_{\mathbf{x}\in X}\|\mathbf{c}(\mathbf{x}) - \mathbf{y}\|$;
$T_0 = I_{m\times m}$;
for $k = 0, 1, \ldots,$ **while** ...
do
$\mathbf{y}_k = \mathbf{c}(\mathbf{x}_k) - T_k\,(\mathbf{f}(\mathbf{x}_k) - \mathbf{y})$;
$\mathbf{x}_{k+1} = \text{argmin}_{\mathbf{x}\in X}\|\mathbf{c}(\mathbf{x}) - \mathbf{y}_k\|$;
break if ...
$(\widetilde{U}_\mathbf{f},\, \widetilde{\Sigma}_\mathbf{f},\, \widetilde{V}_\mathbf{f}) = \text{SVD}(\widetilde{J}_\mathbf{f}(\mathbf{x}_{k+1}))$;
$\widetilde{J}_\mathbf{f}^\dagger(\mathbf{x}_{k+1}) = \widetilde{V}_\mathbf{f}\,\widetilde{\Sigma}_\mathbf{f}^\dagger\,\widetilde{U}_\mathbf{f}^T$;
$T_{k+1} = J_\mathbf{c}(\mathbf{x}_{k+1})\,\widetilde{J}_\mathbf{f}^\dagger(\mathbf{x}_{k+1})$;
enddo

Figure 3.4: The generalized manifold-mapping (GMM) algorithm. Note that $T_k = S_k^\dagger$ for all $k > 0$. $\widetilde{J}_\mathbf{f}(\mathbf{x}_{k+1})$ is either $J_\mathbf{f}(\mathbf{x}_{k+1})$ in case it is available or, in other cases, it is an estimate of it.

of view it is interesting to proceed the other way round: the model is kept fixed and the aim is updated in each step. Then it is particularly attractive to take for this model the available coarse model, which is easily solved by assumption. This leads to a modification of the OMM algorithm. The procedure is shown in Figure 3.3 and we denote it simply as the *manifold-mapping* (MM) *algorithm*. As in the OMM algorithm we need an assumption to assure that $\mathbf{x}_{k+1}$ is well-defined.

Assumption 6. *In the MM algorithm, the updated aims satisfy* $\mathbf{y}_k \in Y\ \forall k$.

In the next chapter we show that both schemes, OMM and MM, behave asymptotically the same and that the convergence studies for these two algorithms coincide.

Remark 3. *As in Remark 2 it can be shown that the manifold-mapping algorithm can be based on any matrix* $\overline{S}$ *satisfying* $J_\mathbf{c}^\dagger(\mathbf{x}_\mathbf{f}^*)\,\overline{S}^\dagger = J_\mathbf{f}^\dagger(\mathbf{x}_\mathbf{f}^*)$. *In fact there is complete freedom with respect to how* $\overline{S}$ *handles components in the complement of the range of* $U_\mathbf{c}$. *The general case is* $\overline{S}^\dagger = J_\mathbf{c}(\mathbf{x}_\mathbf{f}^*)\,J_\mathbf{f}^\dagger(\mathbf{x}_\mathbf{f}^*) + (I - U_\mathbf{c}\,U_\mathbf{c}^T)\,A$ *with* A *any* $m \times m$ *matrix. In [35], the particular cases* $A = (I - U_\mathbf{f}\,U_\mathbf{f}^T)$ *and* $A = I$ *are considered.*

3.2.3 Generalized manifold mapping

As will become clear in the next chapter, the two algorithms, OMM and MM, can be generalized by selecting any S_k for which the fact that $\mathbf{x}_k$ converges to $\overline{\mathbf{x}}$ implies that S_k converges to $J_\mathbf{f}(\overline{\mathbf{x}})\,J_\mathbf{c}^\dagger(\overline{\mathbf{x}})$. We call this more general algorithm generalized manifold mapping (GMM). A natural choice for S_k would be

$J_{\mathbf{f}}(\mathbf{x}_k)\, J_{\mathbf{c}}^{\dagger}(\mathbf{x}_k)$. This makes sense in particular in case the Jacobian of the fine model is available, e.g., via automatic differentiation [48]. But in most cases that information cannot be obtained efficiently and, therefore, we rather write $S_k = \widetilde{J}_{\mathbf{f}}(\mathbf{x}_k)\, J_{\mathbf{c}}^{\dagger}(\mathbf{x}_k)$, with $\widetilde{J}_{\mathbf{f}}(\mathbf{x}_k)$ an approximation of $J_{\mathbf{f}}(\mathbf{x}_k)$, e.g., computed by means of Broyden's method [27]. In Figure 3.4 the GMM scheme corresponding to the MM algorithm is shown. If an approximation for $\widetilde{J}_{\mathbf{f}}(\mathbf{x}_0)$ is available, $T_0 = J_{\mathbf{c}}(\mathbf{x}_0)\, \widetilde{J}_{\mathbf{f}}(\mathbf{x}_0)$ could be used as initial guess instead of the identity matrix. We consider Assumption 6 to be also applicable to GMM.

3.3 Algorithm extensions

The iterative procedures presented in Section 3.2 will be analyzed in detail in Chapter 4. In this section we introduce important algorithmic improvements that will be used in Chapters 5 and 6 for different design problems. The purpose of the first extension is to better handle difficult nonlinear problems. The heuristic base for this improvement is the assumption that — even for strongly nonlinear problems — linearization still gives a reasonable approximation in a sufficiently small neighborhood. The second extension addresses designs where the constraints are expressed by functions with a high computational cost. We essentially extend the manifold-mapping idea from the models to these functions that define the problem constraints. Though in the next chapter some brief remarks will be made with respect to the constrained optimization version of the manifold-mapping scheme, no strict theoretical study will be given for the mentioned algorithm extensions.

3.3.1 Trust-Region manifold mapping

The success of MM iteration hinges on the similarity between $\mathbf{c}(\mathbf{x})$ and $\mathbf{f}(\mathbf{x})$ and on the efficient minimization of the coarse model problems. One important requirement for convergence is that the matrices ΔF and ΔC are sufficiently well-conditioned. However, in practice it is hard to foresee the effect of not satisfying this requirement. In fact, in the initial phase of the iteration we cannot give any a priori guarantee concerning the condition number associated with the matrices ΔF and ΔC. The optimization problem is generally nonlinear and, because we want to reduce the number of iterations as much as possible, large steps in the design variable $\mathbf{x}$ can be expected. For these reasons, the conditioning of ΔC and ΔF can become arbitrarily bad. Moreover, ill-conditioning of ΔF may further lead to large steps, which reduce the feasibility of the assumption that ΔF and ΔC approximate the tangential planes to, respectively, the manifolds $\mathbf{f}(X)$ and $\mathbf{c}(X)$.

Therefore, we have to introduce a (trust-region) strategy [32] which controls the step length in such a way that the undesired effects do no harm and conver-

Figure 3.5: The trust-region manifold-mapping (TRMM) algorithm. Typical values for the parameters are: $\delta = 0$, $\alpha = 1 + \tau > 1$, $\tau = 10^{-10}$, $\beta = 1/10$, $\lambda_{TR} = 1$ and $R_\lambda = r_\lambda = 2$. As with the MM algorithm, the matrix T_{k+1} can, e.g., also be taken as $U_{\Delta C} D U_{\Delta F}^T + (I_{m\times m} - U_{\Delta C} U_{\Delta C}^T)$.

$\mathbf{x}_0 = \mathbf{x}_\mathbf{c}^* = \text{argmin}_{\mathbf{x}\in X} \|\mathbf{c}(\mathbf{x}) - \mathbf{y}\|$;
$T_0 = I_{m\times m}$;
$\lambda_0 = 1$;
for $k = 0, 1, \ldots,$ **while** ...
do
  $\mathbf{y}_k = \mathbf{c}(\mathbf{x}_k) - T_k\,(\mathbf{f}(\mathbf{x}_k) - \mathbf{y})/(1 + \delta\,\lambda_k)$;
  $\mathbf{x}_{k+1} = \text{argmin}_{\mathbf{x}\in X} \|\mathbf{c}(\mathbf{x}) - \mathbf{y}_k\|$;
  break if ...
  while $\|\mathbf{f}(\mathbf{x}_{k+1}) - \mathbf{y}\| > \alpha\,\|\mathbf{f}(\mathbf{x}_k) - \mathbf{y}\|$,
    $\mathbf{x}_{k+1} = \mathbf{x}_k + \tau^\beta\,(\mathbf{x}_{k+1} - \mathbf{x}_k)$;
    $\lambda_k = \max(\lambda_{TR}, R_\lambda\,\lambda_k)$;
  endwhile ;
  $\lambda_{k+1} = \max(\tau, \lambda_k / r_\lambda)$;
  $\Delta F = [\mathbf{f}(\mathbf{x}_{k+1}) - \mathbf{f}(\mathbf{x}_k),\ \cdots,\ \mathbf{f}(\mathbf{x}_{k+1}) - \mathbf{f}(\mathbf{x}_{\max(k+1-n,0)})]$;
  $\Delta C = [\mathbf{c}(\mathbf{x}_{k+1}) - \mathbf{c}(\mathbf{x}_k),\ \cdots,\ \mathbf{c}(\mathbf{x}_{k+1}) - \mathbf{c}(\mathbf{x}_{\max(k+1-n,0)})]$;
  $(U_{\Delta F}, U_{\Delta C}, \Sigma_{\Delta F}, \Sigma_{\Delta C}, V) = \text{GSVD}(\Delta F, \Delta C)$;
  $D = \text{diag}\left(\frac{\sigma_i^{\Delta C} + \lambda_k(\sigma_1^{\Delta C} + \tau)}{\sigma_i^{\Delta F} + \lambda_k(\sigma_1^{\Delta F} + \tau)}\right)$;
  $T_{k+1} = U_{\Delta C} D U_{\Delta F}^T$;
enddo

gence is accelerated. In addition to the reduction of unwanted large step sizes, we should take care of the possible ill-conditioning. With an (almost) singular matrix, the optimal solution could be sought only in a lower-dimensional submanifold of $\mathbf{c}(X)$, and this fact reinforces the rank deficiency of the matrices ΔF and ΔC. Besides, there is a high risk of extremely large steps due to a pseudo-inverse computed for an (almost) singular matrix. Both effects are counteracted by regularization of the matrices ΔF and ΔC. The *generalized singular value decomposition* (GSVD) for the pair $(\Delta F, \Delta C)$ will serve that purpose, in a more elegant way than the two separate singular value decompositions of ΔF and ΔC.

The GSVD of the matrices (A, B) is (cf. [45]) a set of five matrices U_A, U_B, Σ_A, Σ_B and V, such that

$$A = U_A \Sigma_A V^T \qquad \text{and} \qquad B = U_B \Sigma_B V^T\,,$$

with V a regular matrix, U_A and U_B unitary matrices and $\Sigma_A = \text{diag}\left(\sigma_1^A, \cdots, \sigma_n^A\right)$ and $\Sigma_B = \text{diag}\left(\sigma_1^B, \cdots, \sigma_n^B\right)$ diagonal matrices.

The matrix $\Delta C \Delta F^\dagger$ in the MM algorithm can now be written

$$\begin{aligned} \Delta C \Delta F^\dagger &= U_{\Delta C}\, \Sigma_{\Delta C} V^T\, (V^T)^{-1} \Sigma^\dagger_{\Delta F}\, U^T_{\Delta F} = U_{\Delta C}\, \mathrm{diag}\left(\sigma_i^{\Delta C}/\sigma_i^{\Delta F}\right)\, U^T_{\Delta F} \\ &\approx U_{\Delta C}\, \mathrm{diag}\left(\frac{\sigma_i^{\Delta C} + \lambda \sigma_1^{\Delta C}}{\sigma_i^{\Delta F} + \lambda \sigma_1^{\Delta F}}\right)\, U^T_{\Delta F}, \qquad \text{with} \quad \lambda \geq 0\,. \end{aligned}$$

Regularization is introduced by taking $\lambda > 0$. For vanishing λ no regularization takes place and for $\lambda \to \infty$ the matrix reduces to $U_{\Delta C}\, U^T_{\Delta F} \sigma_1^{\Delta C}/\sigma_1^{\Delta F}$, which is close to the identity if $\mathbf{c}(\mathbf{x})$ and $\mathbf{f}(\mathbf{x})$ are similar.

The choice of the parameter λ in the trust-region MM algorithm (TRMM) is based on the success of the previous iteration steps. If the residual $\|\mathbf{f}(\mathbf{x}_{k+1}) - \mathbf{y}\|$ decreases, the value of λ is divided by r_λ, otherwise it is multiplied by R_λ. In practice, λ is never reduced below a small tolerance value τ. The resulting TRMM procedure is shown in Figure 3.5. It should be noticed that the values suggested for a few constants in the algorithm are only based on heuristics. Further, the scheme contains an (optional) damping parameter $\delta \geq 0$. For strongly nonlinear problems it can stabilize the convergence process at the expense of additional function evaluations (we will see an example of it in the next chapter).

3.3.2 Constrained optimization with manifold mapping

The manifold-mapping idea can be easily extended to a constrained optimization problem. An additional (affine) mapping $\mathbf{K} : \mathbf{k_f}(\widehat{X}) \to \mathbf{k_c}(\widehat{X})$ between the constraint manifolds, $\mathbf{k_f}(\widehat{X})$ and $\mathbf{k_c}(\widehat{X})$, is introduced so that, under convergence, the solution found coincides with the right optimum $\mathbf{x_f^*}$. By $\mathbf{K}$, the point $\mathbf{k_c}(\mathbf{x_f^*})$ is mapped to $\mathbf{k_f}(\mathbf{x_f^*})$ and the tangent manifold for $\mathbf{k_c}(\widehat{X})$ at $\mathbf{k_c}(\mathbf{x_f^*})$ to the tangent manifold for $\mathbf{k_f}(\widehat{X})$ at $\mathbf{k_f}(\mathbf{x_f^*})$. Again, the mapping $\mathbf{K}$ is not known a priori because it depends on the solution of the optimization problem and thus, it has to be iteratively approximated. We indicate in Figure 3.6 how the OMM algorithm has to be modified in order to deal with constraints (the approximated constraint manifold mapping is denoted by $\mathbf{K}_k$). The extended scheme for the MM and GMM algorithms can be constructed in an analogous manner.

3.4 Further considerations

We present in this section two special strategies within manifold and space mapping: multi-level optimization and the use of fine and coarse control spaces with a different number of dimensions. In the practical problems described in Chapters 5 and 6 they do not represent significant improvements with respect to the standard approaches introduced up till now, but they can be alternatives to consider in more involved designs.

$\mathbf{x}_0 = \mathbf{x}_\mathbf{c}^* = \text{argmin}_{\mathbf{x}\in\widehat{X}} \|\mathbf{c}(\mathbf{x}) - \mathbf{y}\|$ subject to $\mathbf{k}_\mathbf{c}(\mathbf{x}) \geq 0$;
$\mathbf{S}_0 \bullet = \mathbf{f}(\mathbf{x}_0) + (\bullet - \mathbf{c}(\mathbf{x}_0))$;
$\mathbf{K}_0 \bullet = \mathbf{k}_\mathbf{f}(\mathbf{x}_0) + (\bullet - \mathbf{k}_\mathbf{c}(\mathbf{x}_0))$;
for $k = 0, 1, \ldots,$ **while** ...
do
$\mathbf{x}_{k+1} = \text{argmin}_{\mathbf{x}\in\widehat{X}} \|\mathbf{S}_k(\mathbf{c}(\mathbf{x})) - \mathbf{y}\|$ subject to $\mathbf{K}_k(\mathbf{k}_\mathbf{c}(\mathbf{x})) \geq 0$;
break if ...
$\Delta F = [\mathbf{f}(\mathbf{x}_{k+1}) - \mathbf{f}(\mathbf{x}_k), \cdots, \mathbf{f}(\mathbf{x}_{k+1}) - \mathbf{f}(\mathbf{x}_{\max(k+1-n,0)})]$;
$\Delta C = [\mathbf{c}(\mathbf{x}_{k+1}) - \mathbf{c}(\mathbf{x}_k), \cdots, \mathbf{c}(\mathbf{x}_{k+1}) - \mathbf{c}(\mathbf{x}_{\max(k+1-n,0)})]$;
$\Delta K_\mathbf{f} = [\mathbf{k}_\mathbf{f}(\mathbf{x}_{k+1}) - \mathbf{k}_\mathbf{f}(\mathbf{x}_k), \cdots, \mathbf{k}_\mathbf{f}(\mathbf{x}_{k+1}) - \mathbf{k}_\mathbf{f}(\mathbf{x}_{\max(k+1-n,0)})]$;
$\Delta K_\mathbf{c} = [\mathbf{k}_\mathbf{c}(\mathbf{x}_{k+1}) - \mathbf{k}_\mathbf{c}(\mathbf{x}_k), \cdots, \mathbf{k}_\mathbf{c}(\mathbf{x}_{k+1}) - \mathbf{k}_\mathbf{c}(\mathbf{x}_{\max(k+1-n,0)})]$;
$(U_{\Delta C}, \Sigma_{\Delta C}, V_{\Delta C}) = \text{SVD}(\Delta C)$;
$(U_{\Delta K_\mathbf{c}}, \Sigma_{\Delta K_\mathbf{c}}, V_{\Delta K_\mathbf{c}}) = \text{SVD}(\Delta K_\mathbf{c})$;
$\Delta C^\dagger = V_{\Delta C}\, \Sigma_{\Delta C}^\dagger\, U_{\Delta C}^T$;
$\Delta K_\mathbf{c}^\dagger = V_{\Delta K_\mathbf{c}}\, \Sigma_{\Delta K_\mathbf{c}}^\dagger\, U_{\Delta K_\mathbf{c}}^T$;
$\mathbf{S}_{k+1} \bullet = \mathbf{f}(\mathbf{x}_{k+1}) + \Delta F\, \Delta C^\dagger\, (\bullet - \mathbf{c}(\mathbf{x}_{k+1}))$;
$\mathbf{K}_{k+1} \bullet = \mathbf{k}_\mathbf{f}(\mathbf{x}_{k+1}) + \Delta K_\mathbf{f}\, \Delta K_\mathbf{c}^\dagger\, (\bullet - \mathbf{k}_\mathbf{c}(\mathbf{x}_{k+1}))$;
enddo

Figure 3.6: The original manifold-mapping algorithm for constrained optimization.

3.4.1 Multi-Level optimization

In some occasions, a whole hierarchy of models (ordered with respect to computational cost or accuracy) is available. For example, this is the case of optimization based on solving a discretized partial differential equation by means of fine and coarse grids. Inspired by multigrid we can apply to multiple models the two-level ideas described thus far. Multi-Level optimization is a relatively new research venue that has received some attention recently [19, 38, 46, 47, 57, 62, 63, 75].

In a two-level strategy the fine optimum $\mathbf{x}_\mathbf{f}^*$ is computed through a sequence of coarse model optimizations. Each of these minimization processes might be accelerated by means of a two-level optimization method if proper and even coarser models are available, i.e., the two-level approach is applied recursively. Because the convergence of a scheme based on two models is related to the similarity between these models, the lack of convergence may be overcome by introducing intermediate ones, at the expense of (slightly) increasing the total computational cost. These issues will be investigated in the following analysis.

The three-level approach

Model distance. During the treatment of multi-level efficiency the notion of *model distance* will be useful.

Definition 9. *Given two models defined over the same space,* $\mathbf{f} : X \subset \mathbb{R}^n \to \mathbb{R}^m$ *and* $\mathbf{c} : X \to \mathbb{R}^m$*, we define the distance between* $\mathbf{f}$ *and* $\mathbf{c}$*,* $d(\mathbf{f}, \mathbf{c})$*, as the norm of the model difference*

$$d(\mathbf{f}, \mathbf{c}) = \|\mathbf{f} - \mathbf{c}\| \,. \tag{3.5}$$

For example, we can take the following

$$d(\mathbf{f}, \mathbf{c}) = \| \|\mathbf{f}(\mathbf{x}) - \mathbf{c}(\mathbf{x})\|_{\mathbb{R}^m} \|_{C^2(\mathbb{R}^n)} \,. \tag{3.6}$$

Computational cost analysis. We will first make a simple study of the amount of work related with different types of two- and three-level optimization schemes. Three models are considered: the medium one $\mathbf{m} : X \to \mathbb{R}^m$ has an intermediate accuracy with respect to $\mathbf{f}$ and $\mathbf{c}$. More formally, given three models defined over the same space, $\mathbf{f}$, $\mathbf{m}$ and $\mathbf{c}$, we say that $\mathbf{m}$ is an intermediate model between $\mathbf{f}$ and $\mathbf{c}$ if $d(\mathbf{f}, \mathbf{m}) < d(\mathbf{f}, \mathbf{c})$.

We introduce the following notation. Given a model $\boldsymbol{\alpha}$ we denote by $W_{\boldsymbol{\alpha}}$ and by $W^*_{\boldsymbol{\alpha}}$ the average amount of work associated with one evaluation of that model and with its (single-level) optimization[1], respectively. This last quantity can be written as $W^*_{\boldsymbol{\alpha}} = n_{\boldsymbol{\alpha}} W_{\boldsymbol{\alpha}}$ where $n_{\boldsymbol{\alpha}}$ expresses the (average) number of model evaluations needed for the model $\boldsymbol{\alpha}$ minimization. Thus, the (average) cost of the single-level optimization of the fine model is

$$W^*_{\mathbf{f}} = n_{\mathbf{f}} W_{\mathbf{f}} \,. \tag{3.7}$$

In order to diminish the processor dependency we introduce the *model cost ratio* $r_{\boldsymbol{\alpha}\boldsymbol{\beta}} = W_{\boldsymbol{\beta}}/W_{\boldsymbol{\alpha}}$. If $\mathbf{m}$ and $\mathbf{c}$ are two models derived by means of a simplification of a model $\mathbf{f}$, it should be expected that

$$r_{\mathbf{fm}} > r_{\mathbf{fc}} \Rightarrow d(\mathbf{f}, \mathbf{m}) < d(\mathbf{f}, \mathbf{c}) \,, \tag{3.8}$$

i.e., the faster the model, the larger the error.

The average amount of work of a two-level optimization considering models $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ will be denoted as $W_{\boldsymbol{\alpha}\boldsymbol{\beta}}$ and the average number of iterations associated to that procedure as $n_{\boldsymbol{\alpha}\boldsymbol{\beta}}$. Given three models $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, we denote by $W_{\boldsymbol{\alpha}\boldsymbol{\beta}\boldsymbol{\gamma}}$ the amount of work of a three-level optimization that uses these three models.

The average amount of work of the two-level approach that uses the fine and the coarse model $W_{\mathbf{fc}}$ can be written as

$$W_{\mathbf{fc}} = n_{\mathbf{fc}} W_{\mathbf{f}} + n_{\mathbf{fc}} W^*_{\mathbf{c}} = n_{\mathbf{fc}} (1 + n_{\mathbf{c}} r_{\mathbf{fc}}) W_{\mathbf{f}} \,. \tag{3.9}$$

[1] We assume that the single-level optimization scheme used is the same for every model.

If we proceed analogously for the average amount of work associated with the fine-medium and medium-coarse model approach, $W_{\mathbf{fm}}$ and $W_{\mathbf{mc}}$, we have, respectively

$$W_{\mathbf{fm}} = n_{\mathbf{fm}} W_{\mathbf{f}} + n_{\mathbf{fm}} W^*_{\mathbf{m}} = n_{\mathbf{fm}} (1 + n_{\mathbf{m}} r_{\mathbf{fm}}) W_{\mathbf{f}} , \tag{3.10}$$

$$W_{\mathbf{mc}} = n_{\mathbf{mc}} W_{\mathbf{m}} + n_{\mathbf{mc}} W^*_{\mathbf{c}} = n_{\mathbf{mc}} r_{\mathbf{fm}} (1 + n_{\mathbf{c}} r_{\mathbf{mc}}) W_{\mathbf{f}} . \tag{3.11}$$

Then the amount of work corresponding with the three-level approach $W_{\mathbf{fmc}}$ can be expressed just by replacing $W^*_{\mathbf{m}}$ by $W_{\mathbf{mc}}$ in equation (3.10):

$$W_{\mathbf{fmc}} = n_{\mathbf{fm}} [1 + n_{\mathbf{mc}} r_{\mathbf{fm}} (1 + n_{\mathbf{c}} r_{\mathbf{mc}})] W_{\mathbf{f}} . \tag{3.12}$$

The expression for a larger number of levels can be derived in a straightforward fashion but its analysis, due to the many variables involved, would add no additional information compared to the case of three levels.

Just by comparing (3.7) and (3.9) we see that the two-level approach is more efficient than the single-level one if and only if

$$n_{\mathbf{f}} > n_{\mathbf{fc}} (1 + n_{\mathbf{c}} r_{\mathbf{fc}}) . \tag{3.13}$$

We always look first for coarse models with $r_{\mathbf{fc}} \approx 0$. The model similarity is translated into $n_{\mathbf{f}} > n_{\mathbf{fc}}$. Notice that in the limit $n_{\mathbf{ff}} = 1$ (at the expense of $r_{\mathbf{ff}} = 1$).

Similarly, with equations (3.9) and (3.12) we obtain a similar result concerning the two- and three-level methodologies

$$n_{\mathbf{fc}} (1 + n_{\mathbf{c}} r_{\mathbf{fc}}) > n_{\mathbf{fm}} [1 + n_{\mathbf{mc}} r_{\mathbf{fm}} (1 + n_{\mathbf{c}} r_{\mathbf{mc}})] . \tag{3.14}$$

Below, by some special cases we will gain some additional insight in these inequalities.

A simple example

This illustrative example represents a case of optimization based on the discretization of a partial differential equation. We control the quality of the underlying models by selecting the size of the associated grid.

We consider the Poisson equation in the unit square $\Omega = [0, 1]^2$ with homogeneous Dirichlet boundary conditions. There are two fixed point sources at $\mathbf{s}_1 = [1/4, 1/2]$ and $\mathbf{s}_2 = [1/2, 3/4]$ and their unknown magnitude is the two-dimensional design variable $\mathbf{x} = [x_1, x_2]$, corresponding x_1 and x_2 to the first and second source respectively. The model response is a four-dimensional vector with the solution of the discretized partial differential equation at $\mathbf{t}_1 = [3/8, 5/8]$, $\mathbf{t}_2 = [5/8, 5/8]$, $\mathbf{t}_3 = [5/8, 3/8]$ and $\mathbf{t}_4 = [3/8, 3/8]$. The specifications are equal to $\mathbf{y} = [1, 1, 1, 1]$. The fine model $\mathbf{f}$ is based on a 257×257 rectangular grid. The coarse model $\mathbf{c}$ considers only a 7×7 rectangular grid and it is around 2000

Table 3.1: Results for the Poisson-based optimization.

	#f evals.	**#c evals.**	$F(\cdot)$	**Final design**
Cobyla	90	92	28.129	[4.0762, 4.0760]
Nelder-Mead	74	125	28.129	[4.0761, 4.0761]
SM	7	1175	28.194	[4.3165, 3.7769]
MM	5	424	28.129	[4.0761, 4.0761]

Powell's Cobyla is used with space mapping and Nelder-Mead simplex with manifold mapping. The coarse model is around 2000 times faster than the fine one. $F(\cdot)$ represents the cost function. The stopping criterion is in every case $h_k = \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2 < 10^{-4}$.

Table 3.2: Iteration history for the Poisson-based optimization.

	SM		**MM**	
# iters.	**#(f, c) evals.**	$(F(\cdot), h_k)$	**#(f, c) evals.**	$(F(\cdot), h_k)$
1	(1, 187)	(39.354,)	(1, 126)	(39.354,)
2	(2, 447)	(36.388, 4.1500)	(2, 220)	(36.388, 4.1500)
3	(3, 676)	(28.196, 1.6400)	(3, 303)	(28.340, 1.4500)
4	(4, 867)	(28.194, 0.0189)	(4, 369)	(28.129, 0.6070)
5	(5, 980)	(28.194, 0.0026)	(5, 424)	(28.129, 0.0000)
6	(6, 1105)	(28.194, 0.0002)		
7	(7, 1175)	(28.194, 0.0000)		

times faster than the fine one. The set X is the square $[0.01, 100]^2$. We use the following cost function $F(\mathbf{x})$ for the fine model

$$F(\mathbf{x}) = \frac{\|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|_2}{\|\mathbf{y}\|_2} \times 100\,, \tag{3.15}$$

i.e., a relative Euclidean norm of the model discrepancy. A barrier term is added to take into account the bounds in X. The cost function for the coarse model is analogous to that for the fine one. With the specifications above, $F(\mathbf{x})$ has a unique minimum.

Table 3.1 shows the results of the optimization for the four schemes tried. Both Powell's Cobyla [71] and Nelder-Mead simplex [55] start from the coarse model optimum $\mathbf{x}_\mathbf{c}^* = [4.0992, 6.3160]$. For this reason, the field corresponding to the number of coarse model evaluations for these two methods, is not in blank. Nelder-Mead simplex yields the fine optimum $\mathbf{x}_\mathbf{f}^* = [4.0761, 4.0761]$ and Powell's Cobyla solution practically coincides, with roughly the same computational effort. Space mapping represents an important speed-up with respect to

the one-level approach, but the solution differs from the precise optimum[2] [3]. Manifold mapping conserves the low computational cost of space mapping and according to theory, gives the correct answer. Table 3.2 shows a comparison of both two-level schemes, iteration by iteration until the stopping criterion is met.

A three-level approach is considered for manifold mapping applied to this test problem. The fine model is again based on a 257×257 grid and the two companion models are constructed upon a 33×33 and a 4×4 grid. These two coarse models are around 250 and 3000 times faster than the fine one, respectively. Though even simpler models could be tried (3×4 for example), the last coarse model can be considered as the lower limit in the hierarchy. Table 3.3 shows the results concerning the three-level approach.

We denote the three-level scheme by **MM-(257/33/4)**. **MM-(257/33)** and **MM-(257/4)** refer to the two-level algorithms that compute the fine optimum aided by the 33×33 and 4×4 grid based models, respectively. **MM-(257/33)** presents a monotonous convergence in just four iterations. With the three-level approach, the convergence history is almost identical and the total computational cost is slightly reduced (from 5.13 equivalent fine model evaluations to 4.53). **MM-(257/4)** also yields the fine optimum, but the cost function does not decrease monotonously (the number of equivalent fine model evaluations is 5.12)[4]. From Table 3.3 we can estimate some of the average values introduced above: $n_{\mathbf{m}} \approx 70$, $r_{\mathbf{fm}} \approx 1/250$, $n_{\mathbf{mc}} \approx 5$, $n_{\mathbf{c}} \approx 70$ and $r_{\mathbf{mc}} \approx 250/3000$. This yields a speed-up of the three-level approach **MM-(257/33/4)** of 1.13 with respect to the two-level scheme **MM-(257/33)** (the value measured is also 1.13).

Multi-Level approach and speed-up

The acceleration obtained with the three-level manifold-mapping approach in this problem is not significant. We will next analyze in which situations the multi-level approach might be an alternative to a strategy based on two levels. Since we always look for a very fast coarse model when compared with the fine one, we will assume in this analysis that $n_{\mathbf{c}}\, r_{\mathbf{fc}} \approx 0$. If we then disregard that term in (3.12), we can approximate the amount of work in the three-level approach by

$$W_{\mathbf{fmc}} \approx n_{\mathbf{fm}} \left(1 + n_{\mathbf{mc}}\, r_{\mathbf{fm}}\right) W_{\mathbf{f}} \,. \tag{3.16}$$

[2]Indeed, perfect mapping is not held: $\mathbf{p}(\mathbf{x}_{\mathbf{f}}^*) = [3.9171,\, 6.7854] \neq \mathbf{x}_{\mathbf{c}}^*$.

[3]For no other reason than diversity in the schemes applied, we use Powell's Cobyla with space mapping and Nelder-Mead simplex with manifold mapping. The results obtained with space mapping combined with Nelder-Mead simplex are essentially the same as those with Powell's Cobyla: identical final design and cost function in 6 and 818 fine and coarse model evaluations, respectively.

[4]Two-level space mapping is not performing as its manifold-mapping counterpart when the fine model is aided with that extremely coarse model. The space-mapping solution obtained $\mathbf{x}_{sm}^* = [0.0100,\, 7.3243]$ lies very far away from the precise minimum.

Table 3.3: Three-level approach for the manifold-mapping Poisson-based optimization.

MM-(257/33)				
# iters.	**#f evals.**	**#c evals.**	$F(\cdot)$	$\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2$
1	1	(120, 0)	28.130	
2	2	(176, 0)	28.129	0.0135
3	3	(230, 0)	28.129	0.0012
4	4	(283, 0)	28.129	0.0000

MM-(257/4)				
# iters.	**#f evals.**	**#c evals.**	$F(\cdot)$	$\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2$
1	1	(0, 105)	34.881	
2	2	(0, 175)	29.165	1.4900
3	3	(0, 237)	30.454	0.9010
4	4	(0, 313)	28.129	2.1100
5	5	(0, 369)	28.129	0.0000

MM-(257/33/4)				
# iters.	**#f evals.**	**#c evals.**	$F(\cdot)$	$\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2$
1	1	(5, 365)	28.130	
2	2	(10, 693)	28.129	0.0134
3	3	(15, 1024)	28.129	0.0012
4	4	(20, 1352)	28.129	0.0000

MM-(257/33/4) denotes the three level approach based on the 257×257, 33×33 and 4×4 grid based models. **MM-(257/33)** and **MM-(257/4)** refer to the two-level algorithms that consider the two previous coarse models. These two coarse models are around 250 and 3000 times faster than the fine one, respectively. The first and second figures in the **#c evals.** field denote the number of evaluations of the 33×33 and 4×4 grid based models, respectively. Nelder-Mead simplex was used for the coarsest minimization in all cases. The stopping criterion is in every case $h_k = \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2 < 10^{-4}$.

When this quantity is compared with $W_{\mathbf{fm}} = n_{\mathbf{fm}}\,(1 + n_{\mathbf{m}}\,r_{\mathbf{fm}})\,W_{\mathbf{f}}$ we can see that a speed-up with respect to the scheme **f-m** can be expected if $n_{\mathbf{m}}\,r_{\mathbf{fm}} \gg 1$. This might be the case for **m** still being a relatively expensive model (large $r_{\mathbf{fm}}$) for which many iterations are needed in its optimization (large $n_{\mathbf{m}}$). This last fact is typical in optimization problems with a large number of design variables. The speed-up of the three-level approach **f-m-c** with respect to the strategy **f-m** can be pessimistically estimated by the quotient $n_{\mathbf{m}}/n_{\mathbf{mc}}$. This value will be greater than one with any sensible choice for the medium and coarse models.

With the assumption of $r_{\mathbf{fc}} \approx 0$, the amount of work related to the **f-c** scheme can be approximated by $W_{\mathbf{fc}} \approx n_{\mathbf{fc}} W_{\mathbf{f}}$. The analysis done in the previous paragraph seems consistent with $n_{\mathbf{mc}} r_{\mathbf{fm}} \gg 1$, i.e., a medium model more similar to the fine one. With this assumption, we approximate the speed-up of the three-level approach with respect to the the two-level strategy **f-c** by $n_{\mathbf{fc}}/(n_{\mathbf{fm}} n_{\mathbf{mc}} r_{\mathbf{fm}})$. If we want this speed-up to be greater than one, it is then needed that $n_{\mathbf{fc}}/n_{\mathbf{fm}} \gg 1$. This inequality seems again reasonable for $\mathbf{m}$ much closer to the fine model $\mathbf{f}$ than to the coarse one $\mathbf{c}$.

For a quantitative impression of the models requirements in the previous analysis we can consider the following arbitrary selection: $r_{\mathbf{fm}} = 0.1$, $n_{\mathbf{m}} = 1000$, $n_{\mathbf{fm}} = 6$, $n_{\mathbf{fc}} = 120$, $n_{\mathbf{mc}} = 100$ and a very inexpensive coarse model ($n_{\mathbf{c}} r_{\mathbf{fc}} \approx 0$). For these values we obtain accelerations of 9.18 and 1.82 for the three-level approach **f-m-c** with respect to the two-level schemes **f-m** and **f-c**, respectively. This *purely artificial scenario* might correspond to an optimization design with a large number of variables and a very expensive fine model. Still, a speed-up of only two is not a spectacular improvement.

The minimization problems we deal with in this work are such that the three-level approach cannot be expected to yield an important speed-up increase. In all these cases, the most important features in the fine model can be well represented by a very simple cost model (in this work the magnetic equivalent circuit, see Chapters 5 and 6, is archetypal) and/or the number of design variables is not large. Thus, any two-level scheme needs essentially the same (small) number of iterations, i.e., $n_{\mathbf{fc}} \approx n_{\mathbf{mc}} \approx n_{\mathbf{fm}}$. Consequently, the quantities $W_{\mathbf{fm}}$, $W_{\mathbf{fc}}$ and $W_{\mathbf{fmc}}$ will be very similar (as we have seen in the previous section for the Poisson-based optimization). A three-level strategy does not seem attractive for these optimization problems purely from a cost saving perspective. But for problems with multiple local minima, an approach based on multiple levels can be a much more robust strategy than another one that uses only two (and with not much additional computational cost). We will illustrate that in the next paragraph with another example.

Multi-Level approach and robustness

An intermediate model could be beneficial, conferring robustness to an optimization scheme. The convergence of the two-level approach depends on the distance of the fine model $\mathbf{f}$ and the *corrected* coarse one ($\mathbf{c} \circ \mathbf{p}$ and $\mathbf{S} \circ \mathbf{c}$ for space mapping and manifold mapping, respectively). Because that correction is iteratively approximated in the algorithms presented, convergence is greatly influenced by the distance between the fine and the coarse model. Thus, the introduction of an intermediate model might render methods with better converging properties. The next simple example will illustrate this. We will see that the pair $\mathbf{f} - \mathbf{c}$ yields convergence to just a *local* optimum of the fine optimization problem and that

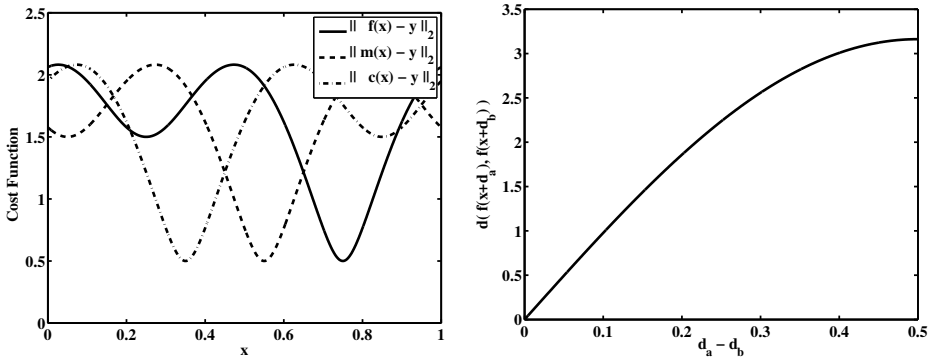


Figure 3.7: The left figure shows the fine $\|\mathbf{f}(x) - \mathbf{y}\|_2$, medium $\|\mathbf{m}(x) - \mathbf{y}\|_2$ and $\|\mathbf{c}(x) - \mathbf{y}\|_2$ cost functions. The right figures represents the model distance between $\mathbf{f}_{d_a} = \mathbf{f}(x + d_a)$ and $\mathbf{f}_{d_b} = \mathbf{f}(x + d_b)$, plotted against $d_a - d_b$.

the three-level approach $\mathbf{f} - \mathbf{m} - \mathbf{c}$ returns the right *global* minimum[5].

The fine model $\mathbf{f}$ is defined as a parametrized ellipse from $X = [0, 1]$ to $\mathbb{R}^2$

$$\mathbf{f}(x) = [2\,\cos(2\,\pi\,x),\; 0.5 + \sin(2\,\pi\,x)]\,, \tag{3.17}$$

i.e., an ellipse centered at $(0,\, 0.5)$ with semi-major and semi-minor axis equal to two and one, respectively. The specifications are $\mathbf{y} = [0,\, 0]$ and the cost function is defined as $\|\mathbf{f}(x) - \mathbf{y}\|_2$, the Euclidean norm of the discrepancy between the model and the specifications, i.e., the closest point of the trajectory to the origin. The fine cost function is shown in Figure 3.7. Two minima can be found, a local one at $x = 0.25$ and the global one at $x_{\mathbf{f}}^* = 0.75$.

The coarse model is built by introducing a delay of $d = 0.4$ units in the fine one

$$\mathbf{c}(x) = \mathbf{f}(x + 0.4)\,. \tag{3.18}$$

If the Euclidean norm is taken for the model distance in (3.6), the distance between the fine and coarse model is $d(\mathbf{f}, \mathbf{c}) = 3.01$. One can easily check that for this fine model the distance between $\mathbf{f}_{d_a} = \mathbf{f}(x + d_a)$ and $\mathbf{f}_{d_b} = \mathbf{f}(x + d_b)$, with d_a and d_b two arbitrary delays, is

$$d(\mathbf{f}_{d_a}, \mathbf{f}_{d_b}) = \sqrt{10}\,\sin(\pi\,|d_a - d_b|)\,. \tag{3.19}$$

Figure 3.7 shows this distance function plotted against $d_a - d_b$.

The manifold-mapping algorithm is applied to this problem with the stopping criterion $\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2 \leq 10^{-4}$ and the global optimizer DIRECT [42, 52] for every

[5]We should notice that the models in this example (with $r_{\mathbf{fm}} \approx r_{\mathbf{mc}} \approx 1$) were selected in order to *only* illustrate the strong relation between convergence and model distance. Therefore, in this simple problem the speed-up analysis of the multi-level approach makes no sense.

minimization within that algorithm. Convergence is achieved, but to the *local* minimizer at $x = 0.25$. Manifold-Mapping theory is not contradicted since it is stated in terms of local optima. The underlying reason of this behavior is that the distance between the fine and the coarse model is large enough to mislead the algorithm towards the wrong solution. Because all the minimization problems in the manifold-mapping scheme are solved by means of a global minimizer, we disregard the possibility of spoiling converge due to either a local intermediate optimum or an inappropriate initial guess.

We consider, in the three-level approach, the medium model obtained with a delay of $d = 0.2$ units with respect to the fine one

$$\mathbf{m}(x) = \mathbf{f}(x + 0.2)\,. \tag{3.20}$$

This model is equidistant to $\mathbf{f}$ and $\mathbf{c}$, i.e., $d(\mathbf{f}, \mathbf{m}) = d(\mathbf{m}, \mathbf{c}) = 1.8785$. The three-level manifold-mapping approach $\mathbf{f} - \mathbf{m} - \mathbf{c}$ is considered with the same stopping criterion as before and the global optimum $x_{\mathbf{f}}^* = 0.75$ is obtained. We should notice that the two-level manifold-mapping scheme $\mathbf{f} - \mathbf{m}$ yields also $x_{\mathbf{f}}^*$ and the iterants practically coincide with those of the commented three-level strategy. (But, as it was commented in the previous section, a three-level optimization procedure $\mathbf{f} - \mathbf{m} - \mathbf{c}$ usually accelerates significantly the two-level scheme $\mathbf{f} - \mathbf{m}$, with a speed-up factor approximated by $n_{\mathbf{m}}/n_{\mathbf{mc}}$).

The main conclusion that can be drawn from this simple example is as follows. The first candidate in a multi-level optimization approach should be a two-level strategy $\mathbf{f} - \mathbf{c}$ with a computationally inexpensive coarse model ($r_{\mathbf{fc}} \approx 0$). Two-model optimization schemes are much simpler to implement than the three-level approach and in many cases they perform similar. But even if convergence is obtained, there is a possibility of not having reached the right optimum. This problem can be alleviated with the aid of an intermediate model framed in a three-level strategy and with usually not much additional computing cost.

A pessimistic estimate of that extra cost, for coarse models fulfilling $r_{\mathbf{fc}} \approx 0$, can be provided. Just from (3.9) and (3.12) we see that

$$\frac{W_{\mathbf{fmc}}}{W_{\mathbf{fc}}} \approx \frac{n_{\mathbf{fm}}}{n_{\mathbf{fc}}}\,(1 + n_{\mathbf{mc}}\, r_{\mathbf{fm}})\,. \tag{3.21}$$

We can always expect that if $\mathbf{m}$ is an intermediate model between $\mathbf{f}$ and $\mathbf{c}$ then $n_{\mathbf{fm}} < n_{\mathbf{fc}}$. Thus, directly from (3.21)

$$\frac{W_{\mathbf{fmc}}}{W_{\mathbf{fc}}} \lesssim 1 + n_{\mathbf{mc}}\, r_{\mathbf{fm}}\,. \tag{3.22}$$

The product $n_{\mathbf{mc}}\, r_{\mathbf{fm}}$ is usually a small quantity (for example, in the Poisson-based optimization problem studied above, we have $n_{\mathbf{mc}}\, r_{\mathbf{fm}} \approx 1/50$).

In the traditional two-level approach, the selection of the coarse model could be in some cases a problematic issue. The three-level approach may overcome

that difficulty since we often find that the computational cost $W_{\mathbf{fmc}}$ is not much larger than the minimum of $W_{\mathbf{fm}}$ and $W_{\mathbf{fc}}$.

3.4.2 Different dimensions of the design spaces

In this section we will denote by $n_{\mathbf{f}}$ and $n_{\mathbf{c}}$ the number of design variables for the fine and coarse model, respectively. Thus far, we have considered models defined over the same set (i.e., $n_{\mathbf{f}} = n_{\mathbf{c}} = n$). Now, the case $\dim(X) \neq \dim(Z)$ will be studied. We are tempted to consider coarse models with less dimensions ($n_{\mathbf{c}} < n_{\mathbf{f}}$), since they are easier to optimize. In case the coarse model optimization is not a problematic issue, we may introduce additional coarse model design variables ($n_{\mathbf{c}} > n_{\mathbf{f}}$), so that the model misalignment is reduced and as a consequence, faster convergence is obtained. Here, we will analyze the two situations, both for space and manifold mapping.

Coarse design space with less dimensions ($n_{\mathbf{c}} < n_{\mathbf{f}}$). The principal difficulty when applying space mapping in this case comes from the fact that the space-mapping function $\mathbf{p}$ cannot be injective, because it maps a space into another one with less dimensions. Thus, even if the fine and coarse optima, $\mathbf{x}^*$ and $\mathbf{z}^*$ respectively, are unique, the space-mapping solutions $\mathbf{x}_p^*$ and $\mathbf{x}_d^*$ might not be sets with a single element. Using the flexibility concept introduced in Chapter 2 we cannot obtain models equally flexible and, thus, the accurate optimum is not expected as the space-mapping solution. We can illustrate this problem with the following simple example.

The data fitting example in Section 2.6 is slightly modified. The data $\mathbf{y}$ and the fine model are kept the same but the coarse model is now

$$\mathbf{c}(z) = [z\, t_i + z]_{i=1,2,3}\,, \tag{3.23}$$

with $Z = \mathbb{R}$. The space-mapping function $\mathbf{p} : \mathbb{R}^2 \to \mathbb{R}$ is uniquely defined

$$\mathbf{p}(\mathbf{x}) = \mathbf{p}(x_1, x_2) = \frac{2}{5}\,(x_1^2 + x_1) + \frac{3}{5}\,x_2\,. \tag{3.24}$$

The fine and coarse model optima are in general unique too. Since $\mathbf{p}$ is surjective, the dual space-mapping solution is found by solving the same system as for the primal one, i.e., $\mathbf{p}(\mathbf{x}_p^*) = \mathbf{z}^*$ (Chapter 2, Lemma 2). The solutions to that system form a manifold in $\mathbb{R}^2$ (a parabola), and because the fine model optimum $\mathbf{x}^*$ is in general unique, we can conclude that the space-mapping framework does not yield the correct answer.

However, space mapping has been applied using coarse models with less dimensions than the fine model [17, 43, 80]. These at first sight surprising results can be understood in case the fine model is ill-conditioned. Under these circumstances, a number of fine design variables could be eliminated, thus, leading to

a model defined over a space of smaller dimension than the original $n_{\mathbf{f}}$ and valid for practical purposes. This is the case of partial space mapping (PSM) [17] where a sensitivity analysis stage, previous to the space-mapping optimization, determines the subset of most significant fine design variables.

Manifold mapping finds an essential difficulty because the spaces X and Z cannot be identified and hence, the mapping $\overline{\mathbf{p}}$ cannot be taken as a bijection, as done in Sections 3.2 and 3.3. But more importantly, the whole manifold-mapping idea makes no sense, because the two tangent manifolds have a different number of dimensions. Without further analysis we might think that this problem can be corrected by establishing a mapping between manifolds of dimension $n_{\mathbf{c}}$ (a tangent manifold for $\mathbf{c}(Z)$ is mapped onto another one of the same dimension *contained* in a tangent manifold of $\mathbf{f}(X)$). However, we will see by means of the example introduced above for space mapping, that this may lead to a stationary point of the algorithm that does not coincide with the fine model optimum $\mathbf{x}^*$.

We study the same least squares approximation as above with the same fine and coarse models, (2.44) and (3.23), respectively. The manifold $\mathbf{c}(Z)$ is a straight line and it can be checked that it is contained in the tangent manifold for $\mathbf{f}(X)$ at any $\mathbf{f}(\mathbf{x})$, with $\mathbf{x} \in X$. Before applying the manifold-mapping algorithm we need to select the function $\overline{\mathbf{p}}$. It seems natural to consider $\overline{\mathbf{p}} = \mathbf{p}$, with $\mathbf{p}$ the space-mapping function in (3.24). The iterant $\mathbf{x}_k$ depends on the regularization criteria adopted for computing the inverse of $\overline{\mathbf{p}}$. For that reason, it is easier to study the manifold-mapping algorithm in terms of the z variable. The starting guess is $z_0 = z^*$ and $\overline{\mathbf{p}}(\mathbf{x}_0) = z_0$. The iterant z_1 is given by

$$z_1 = \operatorname{argmin}_{z \in Z} \|\mathbf{c}(z) - \mathbf{c}(z_0) + \mathbf{f}(\mathbf{x}_0) - \mathbf{y}\| \,. \tag{3.25}$$

Taking into account the definition of the space-mapping function and the fact that $\mathbf{c}(z)$ is linear we can write

$$z_1 = z_0 - \overline{\mathbf{p}}(\mathbf{x}_0) + z^* = z_0 \,. \tag{3.26}$$

We obtain convergence in the first iteration ($\overline{z} = z_0 = z^*$ and $\overline{z} = \overline{\mathbf{p}}(\overline{\mathbf{x}})$), independent of the regularization criteria used. Moreover, $\mathbf{S}\,\mathbf{c}(\overline{\mathbf{p}}(\overline{\mathbf{x}})) = \mathbf{f}(\overline{\mathbf{x}})$ (the matrix S is the identity) and the tangent manifold for $\mathbf{c}(Z)$ at $\mathbf{c}(\overline{z})$ (i.e., $\mathbf{c}(Z)$ itself) is mapped onto another one-dimensional manifold contained in the tangent one for $\mathbf{f}(X)$ at $\mathbf{f}(\overline{\mathbf{x}})$. But, as we have commented, $\overline{\mathbf{x}} = \overline{\mathbf{p}}^{-1}(\overline{z})$ does not in general coincide with $\mathbf{x}^*$.

The standard space-mapping and manifold-mapping approaches do not yield the accurate optimum in the case $n_{\mathbf{c}} < n_{\mathbf{f}}$. But if the fine model is ill-conditioned, a coarse model with less (properly selected) design variables is a valid regularization strategy that might yield very good approximations to the right solution.

Coarse design space with more dimensions ($n_{\mathbf{c}} > n_{\mathbf{f}}$). From an intuitive point of view we understand that a fine model optimization cannot be generally solved by means of a coarse model containing *significantly* less information

(i.e., less design variables). The approach $n_{\mathbf{c}} > n_{\mathbf{f}}$ seems doable, both for the space-mapping and the manifold-mapping frameworks.

In this case, the space-mapping function $\mathbf{p}$ might be injective but not surjective. This means that the primal and the dual approach might lead to singletons sets but $\mathbf{p}$ cannot be expected to be perfect (see the analogy with the case of a more flexible coarse model, Figure 2.8). We cannot be sure that space-mapping yields the right solution but in most practical cases a good approximation will be obtained. Implicit space mapping (ISM) [14] can be interpreted as a space-mapping realization with a coarse model with $n_{\mathbf{c}} > n_{\mathbf{f}}$.

As an illustration we return to the least squares problem but now with the following fine model

$$\mathbf{f}(x) = [x^2 t_i^2 + x\, t + x]_{i=1,2,3}\,, \tag{3.27}$$

with $X = \mathbb{R}$. As in Section 2.6, the coarse model is linear and with two design variables $\mathbf{c}(\mathbf{z}) = \mathbf{c}(z_1, z_2) = z_1\, \mathbf{t} + z_2$. The space-mapping function $\mathbf{p}(x)$ can be determined and reads

$$\mathbf{p}(x) = \left[x, x + \frac{2}{3}x^2\right]\,. \tag{3.28}$$

As can be seen, the set $\mathbf{p}(X)$ is a one-dimensional manifold (parabola) in $Z = \mathbb{R}^2$. Since we may not expect $\mathbf{z}^* \in \mathbf{p}(X)$, the results of the primal and dual approach will not coincide. But in most cases the two space-mapping solutions will be (acceptable) approximations of the fine model optimum x^*.

The manifold-mapping concept is built around the fine model manifold $\mathbf{f}(X)$. The procedures presented need manifolds with the same number of dimensions. If the coarse model manifold $\mathbf{c}(Z)$ is defined over a set in $\mathbb{R}^{n_{\mathbf{c}}}$, with $n_{\mathbf{c}} > n_{\mathbf{f}}$, in order to apply the manifold-mapping algorithms, we just have to select a manifold of dimension $n_{\mathbf{f}}$ contained in $\mathbf{c}(Z)$. This selection procedure can be done by means of the right-preconditioning $\overline{\mathbf{p}} : X \to Z$. Then, the standard manifold-mapping reasoning can be used for the manifolds $\mathbf{f}(X)$ and $\mathbf{c}(\overline{\mathbf{p}}(Z))$, both of dimension $n_{\mathbf{f}}$. In the particular example analyzed within space mapping (fine model given by (3.27) and coarse model as in Section 2.6) we can select the $\overline{\mathbf{p}}$ function

$$\overline{\mathbf{p}}(x) = [x, x]\,, \tag{3.29}$$

which leads to the one-dimensional surrogate model $\mathbf{c}(\overline{\mathbf{p}}(x)) = x\, \mathbf{t} + x$. With this right-preconditioned model and the fine model, the right optimum is obtained, as manifold-mapping theory predicts.

3.5 Conclusions

Manifold mapping was introduced in [35] as an efficient two-level optimization approach to be considered for very time-demanding design problems. In this chapter a number of possible basic variants are identified: original manifold

mapping (OMM), manifold mapping (MM) and generalized manifold mapping (GMM). OMM and MM are asymptotically equivalent and they perform identically in most practical situations. MM is preferred to OMM because it is easier to implement. GMM represents a whole family of schemes based on the manifold-mapping approach. If a good approximation for $J_{\mathbf{f}}(\mathbf{x}_k)$ is available, then GMM is recommended. In all cases, if the iteration converges, the solution of the accurate model is the fixed point of the iteration.

Two extensions of the basic algorithms have been also presented. The first one uses a trust-region strategy and yields the trust-region manifold-mapping algorithm (TRMM). The second one indicates how to deal with complex constraints within the manifold-mapping framework.

A multi-level strategy might not drastically improve the two-model approach, as far as computing time is concerned, but it is a more robust procedure with usually not much additional computational cost. The use of a coarse model with less dimensions than the fine one is not recommended unless as a means to regularize an ill-conditioned fine model. Coarse models with more dimensions than the fine one can be used within space and manifold mapping, and in some cases they may help in reducing the model misalignment. The practical performance of the given algorithms (MM and the two extensions) will be shown in Chapters 5 and 6.

Chapter 4

Manifold-Mapping Analysis

4.1 Introduction

A detailed study of the manifold-mapping concept was started in the last chapter. There, three basic algorithms and two main extensions were introduced. In this chapter we will prove that for the basic schemes, the stationary point of the iteration is a local minimizer of the fine model cost function. We will refer to this properly as the *consistency* of manifold mapping. Later, in Section 4.3, we will also give conditions for convergence. Analogous results for the extensions would require variations of the reasoning that are slightly more complex and need a substantial overload in notation. However, the ideas would be essentially the same as used in the proofs for the basic algorithms.

In general, manifold mapping yields linear convergence. The most important conditions needed for convergence are: (i) $\mathbf{f}(X)$ and $\mathbf{c}(X)$ being manifolds of class C^2, (ii) the models $\mathbf{f}(\mathbf{x})$ and $\mathbf{c}(\mathbf{x})$ showing a similar behavior in the neighborhood of the solution and (iii) the matrices ΔF and ΔC being sufficiently well-conditioned. In some cases it might be difficult to guarantee condition (iii). Trust-region manifold mapping was devised as a means to be used in those more difficult situations.

This chapter is structured as follows. In Section 4.2 consistency is proven for the basic manifold-mapping schemes. Section 4.3 is devoted to a detailed convergence study of those algorithms. Two simple problems illustrate in Section 4.4 some of the theoretical aspects dealt with along the chapter. They also show the convergence behavior of the trust-region manifold-mapping iteration. Chapters 5 and 6 represent the application of the manifold-mapping approach to design problems of practical relevance.

4.2 Consistency

In this section we will specify the conditions under which any of the basic algorithms presented in Section 3.2 is consistent, i.e., if a stationary point of the iteration is obtained, then this point coincides with the fine model optimum $\mathbf{x}_\mathbf{f}^*$.

As one may expect, not every coarse model can be successfully used within the manifold-mapping framework. We partly formalize this by two assumptions.

Assumption 7. *If* $\|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|$ *is locally convex at* $\mathbf{x}_\mathbf{f}^*$ *then* $\|\mathbf{S}\,\mathbf{c}(\mathbf{x}) - \mathbf{y}\|$ *is also locally convex at* $\mathbf{x}_\mathbf{f}^*$.

Assumption 8. *If* $\mathbf{x}_\mathbf{f}^*$ *is a local optimum of* $\|\mathbf{S}\,\mathbf{c}(\mathbf{x}) - \mathbf{y}\|$ *then* $\mathbf{x}_\mathbf{f}^*$ *is the global optimum of* $\|\mathbf{S}\,\mathbf{c}(\mathbf{x}) - \mathbf{y}\|$.

In these assumptions $\mathbf{S}$ is the affine mapping introduced in Section 3.2.1, i.e.,

$$\mathbf{S}\,\mathbf{c}(\mathbf{x}) = \mathbf{f}(\mathbf{x}_\mathbf{f}^*) + \overline{S}\,(\mathbf{c}(\mathbf{x}) - \mathbf{c}(\mathbf{x}_\mathbf{f}^*))\,, \tag{4.1}$$

where

$$\overline{S} = J_\mathbf{f}(\mathbf{x}_\mathbf{f}^*)\,J_\mathbf{c}^\dagger(\mathbf{x}_\mathbf{f}^*)\,. \tag{4.2}$$

These are mild assumptions for the models used in practice. Assumption 7 specifies only a similar *local* behavior in the region of interest, i.e., in a neighborhood of the specifications $\mathbf{y}$. Assumption 8 means that the surrogate optimization does not allow a spurious global optimum near the *true* minimizer $\mathbf{x}_\mathbf{f}^*$.

The manifold-mapping solution $\mathbf{x}_\mathbf{mm}^*$ was defined as

$$\mathbf{x}_\mathbf{mm}^* = \operatorname{argmin}_{\mathbf{x}\in X}\|\mathbf{S}\,\mathbf{c}(\mathbf{x}) - \mathbf{y}\|\,. \tag{4.3}$$

The following two lemmas indicate that this manifold-mapping solution is a local minimizer of the fine model cost function $\|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|$. This fact motivates the basic algorithms for constrained and unconstrained optimization presented in the previous chapter.

Lemma 18. *Any (local) minimizer of the fine model cost function* $\|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|$ *is a (local) minimizer of* $\|\mathbf{S}\,\mathbf{c}(\mathbf{x}) - \mathbf{y}\|$.

Proof. We denote a minimizer of the fine cost function by $\mathbf{x}_\mathbf{f}^*$. First we see that $\mathbf{x}_\mathbf{f}^*$ satisfies the KKT conditions associated with the minimization in (4.3). From (4.1) and (4.2) we have $\mathbf{S}\,\mathbf{c}(\mathbf{x}_\mathbf{f}^*) = \mathbf{f}(\mathbf{x}_\mathbf{f}^*)$ and $J_{\mathbf{S}\,\mathbf{c}}(\mathbf{x}_\mathbf{f}^*) = \overline{S}\,J_\mathbf{c}(\mathbf{x}_\mathbf{f}^*) = J_\mathbf{f}(\mathbf{x}_\mathbf{f}^*)$. Thus, the first derivatives of $F(\mathbf{x})$ and of the surrogate cost function $\|\mathbf{S}\,\mathbf{c}(\mathbf{x}) - \mathbf{y}\|$ coincide at $\mathbf{x}_\mathbf{f}^*$. Since the constraints are the same in both optimization problems[1] and $\mathbf{x}_\mathbf{f}^*$ is a local optimum of $F(\mathbf{x})$ (i.e., the fine KKT conditions hold), we conclude that $\mathbf{x}_\mathbf{f}^*$ satisfies the surrogate KKT conditions. The fine model cost function is locally convex at $\mathbf{x}_\mathbf{f}^*$. Because of Assumption 7, the fine model optimum is also a local minimum of $\|\mathbf{S}\,\mathbf{c}(\mathbf{x}) - \mathbf{y}\|$. □

[1] The constraint function $\mathbf{k}_\mathbf{f}$ in this analysis is assumed to be easy to compute (see Section 3.2).

Lemma 19. $\mathbf{x}^*_{\mathbf{mm}}$ *is a local minimizer of* $\|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|$.

Proof. Use Lemma 18 and Assumption 8. □

Remark 4. *We cannot directly conclude from Lemma 18 that* $\mathbf{x}^*_{\mathbf{mm}}$ *is a minimizer of the fine cost function because the point* $\mathbf{x}^*_{\mathbf{f}}$ *in the lemma could be just a local minimizer of* $\|\mathbf{S}\,\mathbf{c}(\mathbf{x}) - \mathbf{y}\|$ *(the manifold mapping is introduced as a local model correction).*

4.2.1 Original manifold-mapping algorithm

We consider the original manifold-mapping (OMM) algorithm introduced in Section 3.2.1. The following lemma will be very useful for proving that, if the OMM scheme converges, it yields the fine model optimum.

Lemma 20. *Let* $\widetilde{\mathbf{x}} \in X$ *be the minimizer of a surrogate model problem*

$$\widetilde{\mathbf{x}} = \operatorname{argmin}_{\mathbf{x}\in X} \|\widetilde{\mathbf{S}}\,\mathbf{c}(\mathbf{x}) - \mathbf{y}\|\,, \tag{4.4}$$

with

$$\widetilde{\mathbf{S}}\,\mathbf{c}(\mathbf{x}) = \mathbf{f}(\widetilde{\mathbf{x}}) + J_{\mathbf{f}}(\widetilde{\mathbf{x}})\, J^{\dagger}_{\mathbf{c}}(\widetilde{\mathbf{x}})\,(\mathbf{c}(\mathbf{x}) - \mathbf{c}(\widetilde{\mathbf{x}}))\,, \tag{4.5}$$

where $\|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|$ *is locally convex at* $\widetilde{\mathbf{x}}$*, then* $\widetilde{\mathbf{x}}$ *is a (local) minimizer of* $\|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|$.

Proof. Clearly, $\widetilde{\mathbf{x}}$ satisfies the KKT conditions associated with $\|\widetilde{\mathbf{S}}\,\mathbf{c}(\mathbf{x}) - \mathbf{y}\|$, and because of (4.5) we have $\widetilde{\mathbf{S}}\,\mathbf{c}(\widetilde{\mathbf{x}}) = \mathbf{f}(\widetilde{\mathbf{x}})$ and $J_{\widetilde{\mathbf{S}}\,\mathbf{c}}(\widetilde{\mathbf{x}}) = J_{\mathbf{f}}(\widetilde{\mathbf{x}})$. Proceeding as in Lemma 18, we see that the point $\widetilde{\mathbf{x}}$ satisfies also the fine KKT conditions and, because $\|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|$ is locally convex at $\widetilde{\mathbf{x}}$, this point is a (local) minimizer of the fine cost function $F(\mathbf{x})$. □

Remark 5. *Note that from Lemma 20 it also follows that* $\widetilde{\mathbf{S}} = \mathbf{S}$.

We can replace the requirement of $\|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|$ being locally convex at $\widetilde{\mathbf{x}}$ from Lemma 20 by an assumption, very similar in nature to Assumption 7, and also likely to hold in practice:

Assumption 9. *If* $\|\widetilde{\mathbf{S}}\,\mathbf{c}(\mathbf{x}) - \mathbf{y}\|$ *is locally convex at* $\widetilde{\mathbf{x}}$*, then* $\|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|$ *is locally convex at* $\widetilde{\mathbf{x}}$.

Remark 6. *The manifold-mapping theory is generally stated in terms of a local alignment between the surrogate model and the fine model. As a consequence, we can only state results concerning local optima of the fine cost function.*

Now we will show that, if the OMM algorithm converges to a fixed point $\overline{\mathbf{x}}$, this fixed point is a (local) minimizer of the fine cost function. Since we are

studying the fixed point situation, we may assume $k > n$. The iterants of the OMM algorithm are denoted by $\mathbf{x}_k$.

Further, some more mild additional assumptions are needed for proving that $\overline{\mathbf{x}}$ (locally) minimizes $\|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|$. Since the Jacobians $J_{\mathbf{f}}(\mathbf{x})$ and $J_{\mathbf{c}}(\mathbf{x})$ have both rank n, we expect that the matrices ΔF and ΔC in the OMM algorithm are also full-rank. In practice, this will generally be the case and for the exceptional situation where it is not, minor changes can be made in the algorithm to cope with this problem. One possibility, e.g., is the introduction of a stabilization parameter as in trust-region MM (see the scheme in Figure 3.5). Such changes will have no real influence in the computation of the final results. So, in order to prevent minor details in the analysis making the discussion much more complex, we introduce the following assumption.

Assumption 10. *For k large enough, the $m \times n$ matrices ΔF and ΔC have rank n and there are constants K_1, $K_2 > 0$ independent of k such that*

$$\left(\max_{i=0,\ldots,n-1} \|\mathbf{x}_{k+1-i} - \overline{\mathbf{x}}\|^2\right) \|\Delta F^{\dagger}\|_2^2 \leq K_1\,, \tag{4.6}$$

$$\left(\max_{i=0,\ldots,n-1} \|\mathbf{x}_{k+1-i} - \overline{\mathbf{x}}\|^2\right) \|\Delta C^{\dagger}\|_2^2 \leq K_2\,. \tag{4.7}$$

We will see in Lemma 21 that Assumption 10 together with Assumption 11 below guarantee, that $\Delta F\,\Delta C^{\dagger}$ converges to $J_{\mathbf{f}}(\overline{\mathbf{x}})\,J_{\mathbf{c}}^{\dagger}(\overline{\mathbf{x}})$ and thus, that Lemma 20 can be applied.

Assumption 11. *For k large enough the matrix ΔX_{k+1} defined by*

$$\Delta X_{k+1} = [\mathbf{x}_{k+1} - \mathbf{x}_k,\, \mathbf{x}_{k+1} - \mathbf{x}_{k-1},\, \ldots,\, \mathbf{x}_{k+1} - \mathbf{x}_{k-n+1}] \tag{4.8}$$

is regular and there is a constant $K_3 > 0$ independent of k such that

$$\left(\max_{i=0,\ldots,n-1} \|\mathbf{x}_{k+1-i} - \overline{\mathbf{x}}\|^2\right) \|\Delta X_{k+1}^{-1}\|_2^2 \leq K_3\,. \tag{4.9}$$

Remark 7. *Assumption 11 refers to the condition of the matrix ΔX_{k+1} and equivalently to the scaled step directions $(\mathbf{x}_{k+1-i} - \mathbf{x}_{k-i}$, with $i = 0, \ldots, n-1)$. In the exceptional situations where the condition becomes too bad, the algorithm can be easily modified by introducing a stabilization parameter in order to alleviate that, cf. the scheme in Figure 3.5. Assumption 10 is related to Assumption 11 and to the well-posedness of the inverse model operators (see Assumption 12 concerning the coarse model). Because $\mathbf{c}(X)$ is a differentiable manifold we have $\Delta C \approx J_{\mathbf{c}}(\overline{\mathbf{x}})\,\Delta X_{k+1}$ in a neighborhood of $\overline{\mathbf{x}}$, and thus, as it will become clear in the proof for Lemma 21, that $\Delta C^{\dagger} \approx \Delta X_{k+1}^{-1}\,J_{\mathbf{c}}^{\dagger}(\overline{\mathbf{x}})$. Thus, we can expect Assumption 10 to be satisfied if Assumption 11 holds and $\|J_{\mathbf{c}}^{\dagger}(\overline{\mathbf{x}})\|_2^2$ is bounded. This last fact can be expressed as the inverse coarse model operator being Lipschitz in the region of interest. The inequality (4.6) is the analogous relation with respect to the fine model.*

Lemma 21. *Let the sequence of iterants $\mathbf{x}_k$ and operators S_{k+1} be defined by the original manifold-mapping algorithm (OMM). Then, under Assumptions 10 and 11 the operators S_{k+1} converge to $J_{\mathbf{f}}(\overline{\mathbf{x}})\, J_{\mathbf{c}}^{\dagger}(\overline{\mathbf{x}})$, where $\overline{\mathbf{x}}$ is the fixed point of the iteration.*

Proof. By Assumption 10 and because $\mathbf{f}$ and $\mathbf{c}$ are differentiable, we have

$$\Delta F = J_{\mathbf{f}}(\overline{\mathbf{x}})\, \Delta X_{k+1} + M_{\mathbf{f}}\, O(\max_{i=1,\ldots,n} \|\mathbf{x}_{k+1-i} - \overline{\mathbf{x}}\|^2)\,, \tag{4.10}$$

$$\Delta C = J_{\mathbf{c}}(\overline{\mathbf{x}})\, \Delta X_{k+1} + M_{\mathbf{c}}\, O(\max_{i=1,\ldots,n} \|\mathbf{x}_{k+1-i} - \overline{\mathbf{x}}\|^2)\,, \tag{4.11}$$

where $M_{\mathbf{f}}$ and $M_{\mathbf{c}}$ are some $m \times n$ matrices that depend on the smoothness of the manifolds $\mathbf{f}(X)$ and $\mathbf{c}(X)$ but not on k. We can use a generalization of the Banach Lemma for the inverse of a perturbed matrix [45, Theorem 6.1-2] applied to ΔC and conclude that

$$\|\Delta C^{\dagger} - \Delta X_{k+1}^{-1}\, J_{\mathbf{c}}^{\dagger}(\overline{\mathbf{x}})\|_2 \leq 2\, \|M_{\mathbf{c}}\|_2 \max\left\{K_2, K_3\, \|J_{\mathbf{c}}^{\dagger}(\overline{\mathbf{x}})\|_2^2\right\}\,. \tag{4.12}$$

Because of (4.10) and the fact that the norm of $\Delta C^{\dagger} - \Delta X_{k+1}^{-1}\, J_{\mathbf{c}}^{\dagger}(\overline{\mathbf{x}})$ is bounded by a constant independent of k, we obtain that $S_{k+1} = \Delta F\, \Delta C^{\dagger}$ converges to $J_{\mathbf{f}}(\overline{\mathbf{x}})\, J_{\mathbf{c}}^{\dagger}(\overline{\mathbf{x}})$. □

Using this result, we can apply Lemma 20 and conclude that if the OMM algorithm converges, then the fixed point of the iteration is a local minimizer of the fine model cost function $\|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|$. This is summarized in the following theorem.

Theorem 22. *Let $\overline{\mathbf{x}}$ be the fixed point of the original manifold-mapping (OMM, Fig. 3.2) iteration and let the fine model cost function $F(\mathbf{x}) = \|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|$ be locally convex at $\overline{\mathbf{x}}$, then under Assumptions 1, 2, 3, 5, 10 and 11, the point $\overline{\mathbf{x}}$ is a local minimizer of $F(\mathbf{x})$.*

Remark 8. *The assumption of local convexity of $F(\mathbf{x})$ can be replaced by Assumption 9 (model similarity) with $\mathbf{S}$ as in (4.1)-(4.2).*

Remark 9. *By similar arguments as those used here we can obtain similar consistency results for those OMM general algorithms based on mappings $\overline{S}$ satisfying $\overline{S}\, J_{\mathbf{c}}(\mathbf{x}_{\mathbf{f}}^*) = J_{\mathbf{f}}(\mathbf{x}_{\mathbf{f}}^*)$ (see Remark 2, Chapter 3).*

Remark 10. *We can make a similar consistency reasoning for the constraint manifold-mapping algorithm described in Section 3.3.2. We will not analyze it in detail because in essence the proofs are similar to those formulated above. The key point is again that for the stationary point, the KKT conditions related to the*

surrogate optimization, associated with the mappings $\mathbf{S}$ *and* $\mathbf{K}$*, reproduce those for the fine model optimization. I.e., we have that*

$$\begin{aligned} \mathbf{S}\,\mathbf{c}(\mathbf{x}_\mathbf{f}^*) &= \mathbf{f}(\mathbf{x}_\mathbf{f}^*)\,, & (4.13)\\ J_{\mathbf{S}\,\mathbf{c}}(\mathbf{x}_\mathbf{f}^*) &= J_\mathbf{f}(\mathbf{x}_\mathbf{f}^*)\,, & (4.14)\\ \mathbf{K}\,\mathbf{k}_\mathbf{c}(\mathbf{x}_\mathbf{f}^*) &= \mathbf{k}_\mathbf{f}(\mathbf{x}_\mathbf{f}^*)\,, & (4.15)\\ J_{\mathbf{K}\,\mathbf{k}_\mathbf{c}}(\mathbf{x}_\mathbf{f}^*) &= J_{\mathbf{k}_\mathbf{f}}(\mathbf{x}_\mathbf{f}^*)\,. & (4.16) \end{aligned}$$

4.2.2 Manifold-Mapping algorithm

We will consider now the manifold-mapping (MM) algorithm introduced in Section 3.2.2. We can prove a theorem similar to Theorem 22 if Assumption 5 is replaced by Assumption 6.

Theorem 23. *Let* $\overline{\mathbf{x}}$ *be the fixed point of the manifold-mapping (MM, Fig. 3.3) iteration, and let the fine model cost function* $F(\mathbf{x}) = \|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|$ *be locally convex at* $\overline{\mathbf{x}}$*, then under Assumptions 1, 2, 3, 6, 10 and 11, the point* $\overline{\mathbf{x}}$ *is a local minimizer of* $F(\mathbf{x})$*.*

Proof. Proceeding as in Lemma 21 we can see that the sequence of operators $T_{k+1} = S_{k+1}^\dagger$ converges to $(J_\mathbf{f}(\overline{\mathbf{x}})\, J_\mathbf{c}^\dagger(\overline{\mathbf{x}}))^\dagger = J_\mathbf{c}(\overline{\mathbf{x}})\, J_\mathbf{f}^\dagger(\overline{\mathbf{x}})$. Also in the limit

$$\overline{\mathbf{x}} = \mathrm{argmin}_{\mathbf{x}\in X} \|\mathbf{c}(\mathbf{x}) - \mathbf{c}(\overline{\mathbf{x}}) + (J_\mathbf{f}(\overline{\mathbf{x}})\, J_\mathbf{c}^\dagger(\overline{\mathbf{x}}))^\dagger\, (\mathbf{f}(\overline{\mathbf{x}}) - \mathbf{y})\|\,. \tag{4.17}$$

Since $\mathbf{c}(X)$ is a manifold of class C^2, (4.17) is equivalent to

$$\overline{\mathbf{x}} = \mathrm{argmin}_{\mathbf{x}\in X} \|J_\mathbf{c}(\overline{\mathbf{x}})\,(\mathbf{x} - \overline{\mathbf{x}}) + (J_\mathbf{f}(\overline{\mathbf{x}})\, J_\mathbf{c}^\dagger(\overline{\mathbf{x}}))^\dagger\, (\mathbf{f}(\overline{\mathbf{x}}) - \mathbf{y})\|\,. \tag{4.18}$$

Because $J_\mathbf{c}(\overline{\mathbf{x}})$ and $J_\mathbf{f}(\overline{\mathbf{x}})$ are full rank, we can write the former equality as

$$\overline{\mathbf{x}} = \mathrm{argmin}_{\mathbf{x}\in X} \|J_\mathbf{f}(\overline{\mathbf{x}})\, J_\mathbf{c}^\dagger(\overline{\mathbf{x}})\, J_\mathbf{c}(\overline{\mathbf{x}})(\mathbf{x} - \overline{\mathbf{x}}) + \mathbf{f}(\overline{\mathbf{x}}) - \mathbf{y}\|\,, \tag{4.19}$$

and with the same reasoning as in (4.17)-(4.18), equation (4.19) is equivalent to

$$\overline{\mathbf{x}} = \mathrm{argmin}_{\mathbf{x}\in X} \|J_\mathbf{f}(\overline{\mathbf{x}})\, J_\mathbf{c}^\dagger(\overline{\mathbf{x}})\,(\mathbf{c}(\mathbf{x}) - \mathbf{c}(\overline{\mathbf{x}})) + \mathbf{f}(\overline{\mathbf{x}}) - \mathbf{y}\|\,. \tag{4.20}$$

Then Lemma 20 is applied to show that $\overline{\mathbf{x}}$ is a local minimizer of $\|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|$. □

As a consequence of this theorem and by just rewriting (4.17) with $\overline{\mathbf{x}}$ a local minimizer of the fine cost function $F(\mathbf{x})$, the following interesting property for the fine model optimum can be formulated.

Corollary 1. *The fixed point of the manifold-mapping (MM) iteration* $\mathbf{x}_\mathbf{f}^*$ *satisfies*

$$\mathbf{x}_\mathbf{f}^* = \mathrm{argmin}_{\mathbf{x}\in X} \|\mathbf{c}(\mathbf{x}) - \mathbf{c}(\mathbf{x}_\mathbf{f}^*) + \overline{S}^\dagger\, (\mathbf{f}(\mathbf{x}_\mathbf{f}^*) - \mathbf{y})\|\,. \tag{4.21}$$

Remark 11. *Proceeding analogously we can extend these results to those MM algorithms based on mappings $\overline{S}$ satisfying $J_{\mathbf{c}}^{\dagger}(\mathbf{x}_{\mathbf{f}}^{*})\,\overline{S}^{\dagger} = J_{\mathbf{f}}^{\dagger}(\mathbf{x}_{\mathbf{f}}^{*})$ (see Remark 3, Chapter 3).*

Remark 12. *If in the MM trust-region strategy presented in Section 3.3.1 we assume that $\lambda_k \to 0$, then consistency can be obtained with similar arguments as those in this section. We will not go further and state the strict conditions that guarantee $\lambda_k \to 0$. Roughly, λ_k will vanish for large k if the cost function has a regular behavior and it can locally be approximated by linearization. In the examples described in Section 4.4.3, it will be checked that the value of λ_k tends to zero in a neighborhood of the stationary point.*

4.2.3 Generalized manifold-mapping algorithm

Under convergence of the GMM scheme (Section 3.2.3), we can also see that the fixed point for the iteration is again the accurate solution of the optimization problem, $\mathbf{x}_{\mathbf{f}}^{*}$. The proof for the following theorem is completely analogous to those given for Theorems 22 and 23.

Theorem 24. *Let $\overline{\mathbf{x}}$ be the fixed point of the generalized manifold-mapping algorithms (GMM, Fig. 3.4) and let the fine model cost function $F(\mathbf{x}) = \|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|$ be locally convex at $\overline{\mathbf{x}}$, then under Assumption 1, 2, 3, 6, 10 and 11, the point $\overline{\mathbf{x}}$ is a local minimizer of $F(\mathbf{x})$.*

4.3 Convergence

In this section we will give conditions for convergence, i.e., for assuring the existence of a stationary point. The consistency theorems in Section 4.2 guarantee that this point will be the right optimum. We will first present conditions for convergence of the manifold-mapping (MM) algorithm. Then we will show that the original manifold-mapping (OMM) iteration is asymptotically equivalent to the MM iteration and thus, a convergence theorem can be stated with a proof analogous to that for MM.

Convergence depends strongly on the well-posedness of the generalized inverse of the coarse model operator. The *generalized inverse coarse model operator* $\mathbf{c}^{\dagger} : Y \subset \mathbb{R}^m \to X \subset \mathbb{R}^n$ is defined as

$$\mathbf{c}^{\dagger}\,\overline{\mathbf{y}} = \operatorname{argmin}_{\mathbf{x}\in X} \|\mathbf{c}(\mathbf{x}) - \overline{\mathbf{y}}\|\,. \tag{4.22}$$

This operator is well defined because of Assumption 3. We are interested in coarse models whose associated inverse operators are well-posed. This is formalized in the next condition that should be satisfied by any coarse model used in practice.

Assumption 12. *The generalized inverse coarse model operator, $\mathbf{c}^{\dagger}$, is Lipschitz with a Lipschitz constant bounded by $L_{\mathbf{c}^{\dagger}}$.*

4.3.1 Manifold-Mapping algorithm

We start the convergence analysis by rewriting (4.21) and the expression for iterant $\mathbf{x}_{k+1}$ in the MM algorithm in terms of the inverse coarse model operator:

$$\mathbf{x}_{\mathbf{f}}^* = \mathbf{c}^\dagger\left(\mathbf{c}(\mathbf{x}_{\mathbf{f}}^*) - \overline{S}^\dagger\left(\mathbf{f}(\mathbf{x}_{\mathbf{f}}^*) - \mathbf{y}\right)\right), \tag{4.23}$$

$$\mathbf{x}_{k+1} = \mathbf{c}^\dagger\left(\mathbf{c}(\mathbf{x}_k) - S_k^\dagger\left(\mathbf{f}(\mathbf{x}_k) - \mathbf{y}\right)\right). \tag{4.24}$$

Subtracting (4.23) from (4.24) and using Assumption 12, we obtain

$$\|\mathbf{x}_{k+1} - \mathbf{x}_{\mathbf{f}}^*\| \leq L_{\mathbf{c}^\dagger}\left\|\mathbf{c}(\mathbf{x}_k) - S_k^\dagger\left(\mathbf{f}(\mathbf{x}_k) - \mathbf{y}\right) - \mathbf{c}(\mathbf{x}_{\mathbf{f}}^*) + \overline{S}^\dagger\left(\mathbf{f}(\mathbf{x}_{\mathbf{f}}^*) - \mathbf{y}\right)\right\|. \tag{4.25}$$

We can write the expression in the norm at the right-hand side as

$$\begin{aligned}
\mathbf{c}(\mathbf{x}_k) - S_k^\dagger\left(\mathbf{f}(\mathbf{x}_k) - \mathbf{y}\right) - \mathbf{c}(\mathbf{x}_{\mathbf{f}}^*) + \overline{S}^\dagger\left(\mathbf{f}(\mathbf{x}_{\mathbf{f}}^*) - \mathbf{y}\right)) &= \\
\mathbf{c}(\mathbf{x}_k) - \mathbf{c}(\mathbf{x}_{\mathbf{f}}^*) - S_k^\dagger\left(\mathbf{f}(\mathbf{x}_k) - \mathbf{f}(\mathbf{x}_{\mathbf{f}}^*)\right) + (\overline{S}^\dagger - S_k^\dagger)\left(\mathbf{f}(\mathbf{x}_{\mathbf{f}}^*) - \mathbf{y}\right) &= \\
(J_{\mathbf{c}}(\mathbf{x}_{\mathbf{f}}^*) - S_k^\dagger J_{\mathbf{f}}(\mathbf{x}_{\mathbf{f}}^*))\left(\mathbf{x}_k - \mathbf{x}_{\mathbf{f}}^*\right) + (\overline{S}^\dagger - S_k^\dagger)\left(\mathbf{f}(\mathbf{x}_{\mathbf{f}}^*) - \mathbf{y}\right) + O(\|\mathbf{x}_k - \mathbf{x}_{\mathbf{f}}^*\|^2). &
\end{aligned} \tag{4.26}$$

We now analyze the term $(\overline{S}^\dagger - S_k^\dagger)$ in (4.26). We proceed as in the proof of Lemma 21, with the difference that we cannot assume convergence of the algorithm. Since we know the *possible* fixed point of the iteration, with assumptions analogous to those for Lemma 21 (Assumptions 10 and 11), we will be able to establish a relation between $\|\overline{S}^\dagger - S_k^\dagger\|$ and $\|\mathbf{x}_{\mathbf{f}}^* - \mathbf{x}_k\|$.

Assumption 13. *For k large enough, there are constants K_4, $K_5 > 0$ independent of k such that*

$$\left(\max_{i=1,\ldots,n}\|\mathbf{x}_{k+1-i} - \mathbf{x}_{k+1}\|^2\right)\|\Delta F^\dagger\|_2^2 \leq K_4, \tag{4.27}$$

$$\left(\max_{i=1,\ldots,n}\|\mathbf{x}_{k+1-i} - \mathbf{x}_{k+1}\|^2\right)\|\Delta C^\dagger\|_2^2 \leq K_5. \tag{4.28}$$

Assumption 14. *For k large enough, there is a constant $K_6 > 0$ independent of k such that*

$$\left(\max_{i=1,\ldots,n}\|\mathbf{x}_{k+1-i} - \mathbf{x}_{k+1}\|^2\right)\|\Delta X_{k+1}^{-1}\|_2^2 \leq K_6, \tag{4.29}$$

where ΔX_{k+1} is the square matrix defined in Assumption 11.

Remark 13. *We recognize in $\left(\max_{i=1,\ldots,n}\|\mathbf{x}_{k+1-i} - \mathbf{x}_{k+1}\|^2\right)^{1/2}$ a matrix-norm for ΔX_{k+1}. Thus, Assumption 14 can be stated in terms of $\kappa(\Delta X_{k+1})$ the condition number of ΔX_{k+1}, i.e., $\kappa(\Delta X_{k+1}) \leq K_6$, with $K_6 > 0$ a constant independent of k. In the rare situations where linear dependence in the step directions is*

obtained, the algorithm can be slightly modified to cope with that issue, with no significant influence in the final result. Assumption 13 can be related to the condition number of ΔX_{k+1} and the well-posedness of the inverse model operators in the region of interest.

Lemma 25. *Under Assumptions 1, 2, 3, 6, 9, 10, 11, 13 and 14,*

$$\|S_k^\dagger - J_{\mathbf{c}}(\mathbf{x}_k)\, J_{\mathbf{f}}^\dagger(\mathbf{x}_k)\| \leq M_1 \max_{i=0,\,\ldots,\,n-1} \|\mathbf{x}_{k-i} - \mathbf{x}_{\mathbf{f}}^*\|\,, \tag{4.30}$$

where $M_1 > 0$ is a constant that depends on the smoothness of the manifolds $\mathbf{f}(X)$ and $\mathbf{c}(X)$ but not on k.

Proof. As in Lemma 21 we can write

$$\Delta F = J_{\mathbf{f}}(\mathbf{x}_{k+1})\,\Delta X_{k+1} + M_{\mathbf{f}_k}\, O(\max_{i=1,\,\ldots,\,n} \|\mathbf{x}_{k+1-i} - \mathbf{x}_{k+1}\|^2)\,, \tag{4.31}$$

$$\Delta C = J_{\mathbf{c}}(\mathbf{x}_{k+1})\,\Delta X_{k+1} + M_{\mathbf{c}_k}\, O(\max_{i=1,\,\ldots,\,n} \|\mathbf{x}_{k+1-i} - \mathbf{x}_{k+1}\|^2)\,, \tag{4.32}$$

where $M_{\mathbf{f}_k}$ and $M_{\mathbf{c}_k}$ are some $m \times n$ matrices that depend on the smoothness of the manifolds $\mathbf{f}(X)$ and $\mathbf{c}(X)$ but not on k. Again with [45, Theorem 6.1-2] and by Assumptions 13 and 14 we can conclude that $S_k^\dagger - J_{\mathbf{c}}(\mathbf{x}_k)\, J_{\mathbf{f}}^\dagger(\mathbf{x}_k)$ is bounded by a constant multiplied by $\max_{i=1,\,\ldots,\,n} \|\mathbf{x}_{k+1-i} - \mathbf{x}_k\|$. Clearly, this is equivalent to $S_k^\dagger - J_{\mathbf{c}}(\mathbf{x}_k)\, J_{\mathbf{f}}^\dagger(\mathbf{x}_k)$ being bounded by $M_1 \max_{i=0,\,\ldots,\,n-1} \|\mathbf{x}_{k-i} - \mathbf{x}_{\mathbf{f}}^*\|$ where $M_1 > 0$ is a constant that depends on the smoothness of $\mathbf{f}(X)$ and $\mathbf{c}(X)$ but not on k. □

Remark 14. *The constant M_1 depends on the smoothness of $\mathbf{f}(X)$ and $\mathbf{c}(X)$. (If both manifolds are linear in the neighborhood of the solution then $M_1 = 0$.)*

Lemma 26. *Under Assumptions 1, 2, 3, 6, 9, 10, 11, 13 and 14,*

$$\|\overline{S}^\dagger - J_{\mathbf{c}}(\mathbf{x}_k)\, J_{\mathbf{f}}^\dagger(\mathbf{x}_k)\| \leq M_2 \|\mathbf{x}_k - \mathbf{x}_{\mathbf{f}}^*\|\,, \tag{4.33}$$

where $M_2 > 0$ is a constant that depends on the smoothness of the manifolds $\mathbf{f}(X)$ and $\mathbf{c}(X)$ but not on k.

Proof. We can write

$$\begin{aligned}
\overline{S}^\dagger - J_{\mathbf{c}}(\mathbf{x}_k)\, J_{\mathbf{f}}^\dagger(\mathbf{x}_k) &= J_{\mathbf{c}}(\mathbf{x}_{\mathbf{f}}^*)\, J_{\mathbf{f}}^\dagger(\mathbf{x}_{\mathbf{f}}^*) - J_{\mathbf{c}}(\mathbf{x}_k)\, J_{\mathbf{f}}^\dagger(\mathbf{x}_k) \\
&= J_{\mathbf{c}}(\mathbf{x}_{\mathbf{f}}^*)\, J_{\mathbf{f}}^\dagger(\mathbf{x}_{\mathbf{f}}^*) - J_{\mathbf{c}}(\mathbf{x}_k)\, J_{\mathbf{f}}^\dagger(\mathbf{x}_{\mathbf{f}}^*) + \\
&\quad + J_{\mathbf{c}}(\mathbf{x}_k)\, J_{\mathbf{f}}^\dagger(\mathbf{x}_{\mathbf{f}}^*) - J_{\mathbf{c}}(\mathbf{x}_k)\, J_{\mathbf{f}}^\dagger(\mathbf{x}_k) \\
&= (J_{\mathbf{c}}(\mathbf{x}_{\mathbf{f}}^*) - J_{\mathbf{c}}(\mathbf{x}_k))\, J_{\mathbf{f}}^\dagger(\mathbf{x}_{\mathbf{f}}^*) + \\
&\quad + J_{\mathbf{c}}(\mathbf{x}_k)\, (J_{\mathbf{f}}^\dagger(\mathbf{x}_{\mathbf{f}}^*) - J_{\mathbf{f}}^\dagger(\mathbf{x}_k))\,.
\end{aligned} \tag{4.34}$$

Since $\mathbf{f}(X)$ and $\mathbf{c}(X)$ are manifolds of class C^2, we can bound $\|\overline{S}^\dagger - J_\mathbf{c}(\mathbf{x}_k)\, J_\mathbf{f}^\dagger(\mathbf{x}_k)\|$ by $M_2 \|\mathbf{x}_k - \mathbf{x}_\mathbf{f}^*\|$, where $M_2 > 0$ is a constant that depends on the smoothness of the two manifolds but not on k. □

Corollary 2. *Under Assumptions 1, 2, 3, 6, 9, 10, 11, 13 and 14,*

$$\|\overline{S}^\dagger - S_k^\dagger\| \leq M \max_{i=0,\,\ldots,\,n-1} \|\mathbf{x}_{k-i} - \mathbf{x}_\mathbf{f}^*\|\,, \tag{4.35}$$

where $M > 0$ is a constant that depends on the smoothness of the manifolds $\mathbf{f}(X)$ and $\mathbf{c}(X)$ but not on k.

Proof. We apply Lemmas 25 and 26 and set $M = \max(M_1,\, M_2)$. □

Now, combining (4.25) and (4.26) we get

$$\begin{aligned} \|\mathbf{x}_{k+1} - \mathbf{x}_\mathbf{f}^*\| \quad \leq \quad & L_{\mathbf{c}^\dagger}\, \|J_\mathbf{c}(\mathbf{x}_\mathbf{f}^*) - S_k^\dagger\, J_\mathbf{f}(\mathbf{x}_\mathbf{f}^*)\|\, \|\mathbf{x}_k - \mathbf{x}_\mathbf{f}^*\| + \\ & + \|\overline{S}^\dagger - S_k^\dagger\|\, \|\mathbf{f}(\mathbf{x}_\mathbf{f}^*) - \mathbf{y}\| + O(\|\mathbf{x}_k - \mathbf{x}_\mathbf{f}^*\|^2)\,. \end{aligned} \tag{4.36}$$

Due to (4.35) we can finally write

$$\begin{aligned} \|\mathbf{x}_{k+1} - \mathbf{x}_\mathbf{f}^*\| \quad \leq \quad & L_{\mathbf{c}^\dagger}\, (\|J_\mathbf{c}(\mathbf{x}_\mathbf{f}^*) - S_k^\dagger\, J_\mathbf{f}(\mathbf{x}_\mathbf{f}^*)\| + \\ & + M\, \|\mathbf{f}(\mathbf{x}_\mathbf{f}^*) - \mathbf{y}\|) \max_{i=0,\,\ldots,\,n-1} \|\mathbf{x}_{k-i} - \mathbf{x}_\mathbf{f}^*\| + \\ & + O(\|\mathbf{x}_k - \mathbf{x}_\mathbf{f}^*\|^2)\,. \end{aligned} \tag{4.37}$$

We formulate this result in the following theorem.

Theorem 27. *Under Assumptions 1, 2, 3, 6, 9, 10, 11, 12, 13 and 14, and the condition*

$$L_{\mathbf{c}^\dagger}\, (\|J_\mathbf{c}(\mathbf{x}_\mathbf{f}^*) - S_k^\dagger\, J_\mathbf{f}(\mathbf{x}_\mathbf{f}^*)\| + M\, \|\mathbf{f}(\mathbf{x}_\mathbf{f}^*) - \mathbf{y}\|) < 1 \qquad \text{for} \quad k \geq k_0\,, \tag{4.38}$$

where $M > 0$ is a constant that depends on the smoothness of the manifolds $\mathbf{f}(X)$ and $\mathbf{c}(X)$ but not on k, the manifold-mapping algorithm (MM, Fig. 3.3) yields (linear) convergence to $\mathbf{x}_\mathbf{f}^*$.

Corollary 3. *If, in addition to the assumptions for Theorem 27, we have $\mathbf{f}(\mathbf{x}_\mathbf{f}^*) = \mathbf{y}$ (i.e., a reachable design), then the convergence of the MM algorithm is superlinear.*

Corollary 4. *If in addition to the assumptions for Theorem 27 we have $S_k = \overline{S} = J_\mathbf{f}(\mathbf{x}_\mathbf{f}^*)\, J_\mathbf{c}^\dagger(\mathbf{x}_\mathbf{f}^*)$ for every $k \geq k_0$, then the convergence of the MM algorithm is quadratic.*

4.3.2 Original manifold-mapping algorithm

OMM convergence results will be obtained from those derived for MM in the previous subsection. For that purpose we will just see that —under convergence— the original (OMM) and the manifold-mapping (MM) iterations are asymptotically equivalent.

Lemma 28. *If the iteration in the original manifold-mapping (OMM) algorithm converges, for a large enough k we find for its iterant* $\mathbf{x}_{k+1}$

$$\mathbf{x}_{k+1} = \operatorname{argmin}_{\mathbf{x}\in X} \|\mathbf{c}(\mathbf{x}) - \mathbf{c}(\mathbf{x}_k) + S_k^\dagger \left(\mathbf{f}(\mathbf{x}_k) - \mathbf{y}\right) + O(\|\mathbf{x} - \mathbf{x}_k\|^2)\| \,. \tag{4.39}$$

Proof. Because $\mathbf{c}(X)$ is differentiable in a neighborhood of the fixed point, for a large enough k we can write

$$\begin{aligned} \mathbf{x}_{k+1} &= \operatorname{argmin}_{\mathbf{x}\in X} \|S_k\mathbf{c}(\mathbf{x}) - S_k\mathbf{c}(\mathbf{x}_k) + \mathbf{f}(\mathbf{x}_k) - \mathbf{y}\| \\ &= \operatorname{argmin}_{\mathbf{x}\in X} \|S_k J_\mathbf{c}(\mathbf{x}_k)\left(\mathbf{x} - \mathbf{x}_k\right) + \mathbf{f}(\mathbf{x}_k) - \mathbf{y} + S_k\, O(\|\mathbf{x} - \mathbf{x}_k\|^2)\| \\ &= \operatorname{argmin}_{\mathbf{x}\in X} \|S_k J_\mathbf{c}(\mathbf{x}_k)\left(\mathbf{x} - \mathbf{x}_k\right) + \mathbf{f}(\mathbf{x}_k) - \mathbf{y} + O(\|\mathbf{x} - \mathbf{x}_k\|^2)\| \,. \end{aligned} \tag{4.40}$$

and for the last equality we remember that S_k converges to $\overline{S}$ (Lemma 21). The iterant $\mathbf{x}_{k+1}$ can be expressed as

$$\begin{aligned} \mathbf{x}_{k+1} = \operatorname{argmin}_{\mathbf{x}\in X} \|&S_k\, \Delta C\, \Delta X_k^{-1}\left(\mathbf{x} - \mathbf{x}_k\right) + \\ &+\mathbf{f}(\mathbf{x}_k) - \mathbf{y} + O(\|\mathbf{x} - \mathbf{x}_k\|^2)\| \,, \end{aligned} \tag{4.41}$$

where, for a large enough k we have $O(\|\mathbf{x} - \mathbf{x}_k\|) = O(\max_{i=1,\,\ldots,\,n} \|\mathbf{x}_{k-i} - \mathbf{x}_k\|)$ in the Taylor expansion, since there is convergence. Because $S_k = \Delta F\, \Delta C^\dagger$ and $\Delta C^\dagger\, \Delta C = I$, we have

$$\begin{aligned} \mathbf{x}_{k+1} = \operatorname{argmin}_{\mathbf{x}\in X} \|&\Delta F\, \Delta X_k^{-1}\left(\mathbf{x} - \mathbf{x}_k\right) + \\ &+\mathbf{f}(\mathbf{x}_k) - \mathbf{y} + O(\|\mathbf{x} - \mathbf{x}_k\|^2)\| \,. \end{aligned} \tag{4.42}$$

Further, since ΔF and ΔC are full-rank and ΔX_k is regular

$$\begin{aligned} \mathbf{x}_{k+1} = \operatorname{argmin}_{\mathbf{x}\in X} \|&\Delta C\, \Delta X_k^{-1}\left(\mathbf{x} - \mathbf{x}_k\right) + \\ &+S_k^\dagger \left(\mathbf{f}(\mathbf{x}_k) - \mathbf{y}\right) + O(\|\mathbf{x} - \mathbf{x}_k\|^2)\| \,. \end{aligned} \tag{4.43}$$

The lemma follows immediately from this last equation. □

Lemma 29. *If the iteration in the manifold-mapping (MM) algorithm converges, for a large enough k we find for its iterant* $\mathbf{x}_{k+1}$

$$\mathbf{x}_{k+1} = \operatorname{argmin}_{\mathbf{x}\in X} \|S_k \left(\mathbf{c}(\mathbf{x}) - \mathbf{c}(\mathbf{x}_k)\right) + \mathbf{f}(\mathbf{x}_k) - \mathbf{y} + O(\|\mathbf{x} - \mathbf{x}_k\|^2)\| \,. \tag{4.44}$$

Proof. Analogous to that for Lemma 28. □

Corollary 5. *Under convergence, the OMM algorithm and the MM algorithm are asymptotically equivalent.*

Remark 15. *Due to Corollary 5, Theorem 27 and Corollaries 3 and 4 are also valid for the original manifold-mapping algorithm (OMM, Fig. 3.2) when Assumption 6 is replaced by Assumption 5.*

Remark 16. *By similar arguments as those given for mappings based on* $\overline{S} = J_{\mathbf{f}}(\mathbf{x}_{\mathbf{f}}^*)\, J_{\mathbf{c}}^{\dagger}(\mathbf{x}_{\mathbf{f}}^*)$*, analogous convergence results can be obtained for the other versions of the algorithms introduced in Remarks 2 and 3 in Chapter 3.*

Remark 17. *In the case of the constrained version of the MM scheme we can write the expression for iterant* $\mathbf{x}_{k+1}$ *as*

$$\mathbf{x}_{k+1} \quad = \quad \mathbf{c}_k^{\dagger}\left(\mathbf{c}(\mathbf{x}_k) - S_k^{\dagger}\left(\mathbf{f}(\mathbf{x}_k) - \mathbf{y}\right)\right), \tag{4.45}$$

where $\mathbf{c}_k^{\dagger}$ *denotes the general inverse operator of the coarse model with a constraint function given by* $\mathbf{K}_k \circ \mathbf{k}_{\mathbf{c}}$*. We can assume that for* k *large enough and in a neighborhood of the problem solution*

$$\|\mathbf{c}^{\dagger}(\mathbf{y}_1) - \mathbf{c}_k^{\dagger}(\mathbf{y}_2)\| \leq L_{\mathbf{c}^{\dagger}}\, \|\mathbf{y}_1 - \mathbf{y}_2\|, \tag{4.46}$$

with $L_{\mathbf{c}^{\dagger}}$ *a constant independent of* k*. A similar expression to (4.25) can be obtained and then, the rest of the proof would be essentially the same as that for the MM algorithm. The additional condition (4.46) measures again the similarity between the fine and the coarse constraints and in many practical cases is a reasonable assumption.*

4.3.3 Generalized manifold-mapping algorithm

The proof for generalized manifold mapping (GMM) is analogous to that for MM. The two possible GMM algorithms, corresponding to OMM and MM, are also asymptotically equivalent. We formulate this in the following theorem.

Theorem 30. *Under Assumptions 1, 2, 3, 6, 9, 10, 11, 12, 13 and 14, and the condition*

$$L_{\mathbf{c}^{\dagger}}\left(\|J_{\mathbf{c}}(\mathbf{x}_{\mathbf{f}}^*) - S_k^{\dagger}\, J_{\mathbf{f}}(\mathbf{x}_{\mathbf{f}}^*)\| + \overline{M}\, \|\mathbf{f}(\mathbf{x}_{\mathbf{f}}^*) - \mathbf{y}\|\right) < 1 \qquad \text{for} \quad k \geq k_0\,, \tag{4.47}$$

where $\overline{M} > 0$ *is a constant that depends on the smoothness of the manifolds* $\mathbf{f}(X)$ *and* $\mathbf{c}(X)$ *but not on* k*, the generalized manifold-mapping algorithms (GMM, Fig. 3.4) yield (linear) convergence to* $\mathbf{x}_{\mathbf{f}}^*$*.*

Remark 18. *In the case* $S_{k+1} = J_{\mathbf{f}}(\mathbf{x}_{k+1})\, J_{\mathbf{c}}^{\dagger}(\mathbf{x}_{k+1})$ *an analog to Lemma 25 can be trivially stated. The constant* $\overline{M}$ *introduced in the theorem for GMM could be in some situations smaller than* M*, the one for MM. As a consequence, the (linear) convergence for the GMM schemes may have smaller associated constants than that for the MM iterations (see Sections 4.4.1 and 4.4.2).*

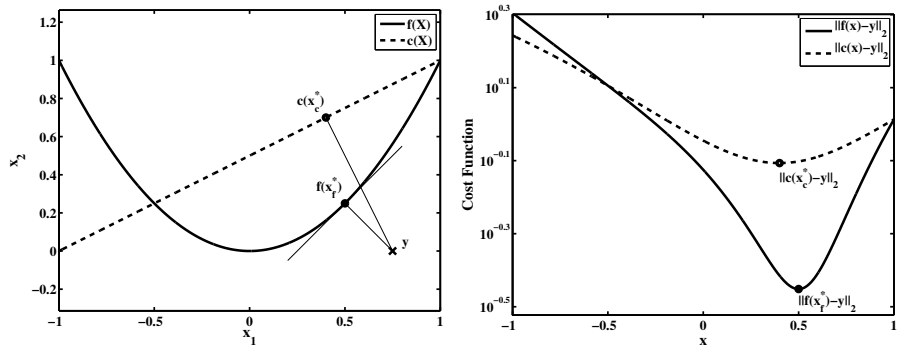


Figure 4.1: Left: The sets $\mathbf{f}(X)$ and $\mathbf{c}(X)$, specifications $\mathbf{y}$ and $\mathbf{f}(x_{\mathbf{f}}^*)$ and $\mathbf{c}(x_{\mathbf{c}}^*)$ for the example in Section 4.4.1. Right: the fine and coarse cost functions, $\|\mathbf{f}(x) - \mathbf{y}\|_2$ and $\|\mathbf{c}(x) - \mathbf{y}\|_2$, respectively, for the same example.

4.4 Examples

In Section 4.4.1 and 4.4.2 we illustrate the most significant results obtained in this chapter by means of two simple optimization problems. We also include a third section showing the convergence behavior of the TRMM scheme for different situations of the second of these two examples.

4.4.1 Linear and superlinear convergence

We illustrate the convergence Theorems 27 and 30 with this example. The fine model is defined over $X = [-1,\ 1]$ by

$$\mathbf{f}(x) = [x, x^2]\,. \tag{4.48}$$

The set $\mathbf{f}(X) \subset \mathbb{R}^2$ is part of a parabola and we want to find the point in that set closest in Euclidean norm to the specifications $\mathbf{y} = [3/4,\ 0]$. The coarse model is defined over $Z = X$ and is the linear $\mathbf{c}(x) = [x,\ (1+x)/2]$. Figure 4.1 shows a representation of the problem and the fine and coarse cost functions, $\|\mathbf{f}(x) - \mathbf{y}\|_2$ and $\|\mathbf{c}(x) - \mathbf{y}\|_2$, respectively. The corresponding optima are $x_{\mathbf{f}}^* = 0.5$ and $x_{\mathbf{c}}^* = 0.4$.

Though the two models are not specially similar around the solution region, the similarity between them is enough for obtaining convergence with the manifold-mapping approach. Since both manifolds $\mathbf{f}(X)$ and $\mathbf{c}(X)$ are smooth, we expect a reasonable small constant M in (4.38) for Theorem 27. The design is not reachable and convergence is linear for the manifold-mapping (MM) and the generalized manifold-mapping (GMM) iterations (see Figure 4.2). In this problem it is easy to check that both original manifold mapping (OMM) and MM

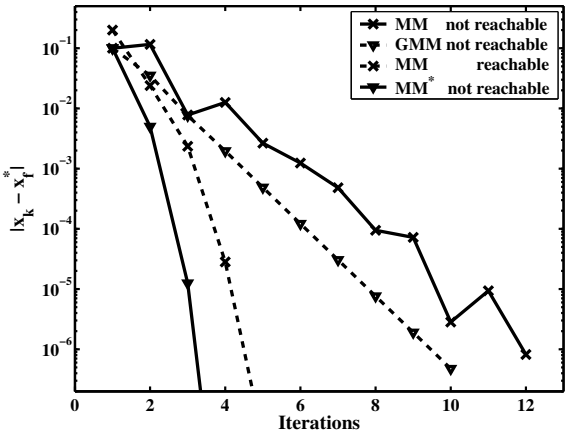


Figure 4.2: Convergence history for $|x_k - x_{\mathbf{f}}^*|$ for the manifold-mapping (MM) and the generalized manifold-mapping (GMM) iteration for a reachable and a non-reachable design in the example in Section 4.4.1. MM* denotes the MM algorithm with $S_k = \overline{S}$.

coincide, iterant by iterant. We see that MM needs twelve iterations for getting $|x_k - x_{\mathbf{f}}^*|$ smaller than 10^{-6}. The GMM scheme, using the exact Jacobian, yields the same accuracy in ten iterations. In Figure 4.2 we also check that the constant in the linear convergence rate is smaller for the latter algorithm. If the Jacobian is estimated by Broyden's method, the complete iteration history coincides with that for MM. The reason is that for a function of one variable, Broyden's method coincides with the secant algorithm for approximating a derivative, and that procedure is essentially the one followed in the computation of ΔF and ΔC in the MM algorithm (not only MM and GMM use the same Jacobian estimation, also the iterants x_0 and x_1 coincide). In the next example, when we consider a function of two variables, we see that the Broyden-based GMM algorithm differs significantly from the MM one.

In Figure 4.2 we also observe two cases of superlinear convergence for MM (cf., Corollaries 3 and 4). If we apply MM with $S_k = \overline{S}$ (denoted by MM* in Figure 4.2) we obtain a solution with the same accuracy of 10^{-6} in only four iterations. Nevertheless, that situation is unrealistic because the necessary information is not available before the optimization problem has been solved. In the case of the reachable design, given by $\mathbf{y} = [1/2,\ 1/4]$ (yielding again $x_{\mathbf{f}}^* = 0.5$), the solution is obtained with an accuracy of 10^{-6} in five iterations. The same superlinear convergence is observed for the GMM algorithm.

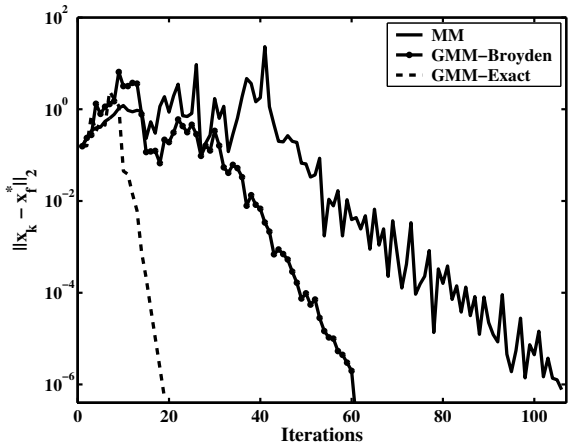


Figure 4.3: Convergence history for $\|\mathbf{x}_k - \mathbf{x}_\mathbf{f}^*\|_2$ for the manifold-mapping (MM) and two different generalized manifold-mapping (GMM) iterations in the example in Section 4.4.2 (non-reachable design and a bad correspondence between the fine and the coarse model). GMM-Exact denotes the GMM scheme with the exact Jacobian $J_\mathbf{f}(\mathbf{x}_k)$. GMM-Broyden approximates that Jacobian via Broyden's method.

4.4.2 Different types of linear convergence

With this example we show that different choices for the fine model Jacobian estimation at the k-th iteration, $\widetilde{J}_\mathbf{f}(\mathbf{x}_k)$, yield distinct convergence histories: the better $\widetilde{J}_\mathbf{f}(\mathbf{x}_k)$ approximates $J_\mathbf{f}(\mathbf{x}_k)$, the smaller the constant associated with the linear convergence. The example has been introduced in Section 2.6 (see (3) in Table 2.1) as a least squares best approximation of the data vector $\mathbf{y} = [0,\ -0.4,\ 0.1]$ by the fine model

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}(x_1, x_2) = [x_1\,(x_2 - 1)^2,\ x_1,\ x_1\,(x_2 + 1)^2] \tag{4.49}$$

defined over $X = \mathbb{R}^2$. The design is not reachable. The coarse model

$$\mathbf{c}(\mathbf{x}) = \mathbf{c}(x_1, x_2) = [-x_1 + x_2,\ x_2,\ x_1 + x_2] \tag{4.50}$$

is again linear and it is also defined over $Z = X$.

We solve the problem with the manifold-mapping (MM) and the generalized manifold-mapping (GMM) algorithms. In all these schemes the matrix A in Remark 3 (Chapter 3) is the null matrix. For GMM, two variants are compared, one with the exact Jacobian for the fine model and the other with an approximation based on Broyden's method. It should be noticed that for most time-expensive fine models, the availability of the exact Jacobian is an unrealistic assumption.

All the schemes yield the fine optimum $\mathbf{x}_\mathbf{f}^* = [-0.101, -0.141]$. The convergence history for the three methods is shown in Figure 4.3. We clearly observe that the convergence is linear in all cases and that the constant M in the convergence theorems (indicating the slope of the convergence history) is different for each algorithm. In this problem the discrepancy between the fine and coarse models in the solution region is large and this fact is recognized in an elevated number of iterations needed, when compared with the previous example. In practice, fine and coarse models are much more similar and, hence, convergence is generally achieved with much less function evaluations.

4.4.3 TRMM convergence

As an example of the convergence behavior of the TRMM algorithm we consider all the different cases for the problem discussed in the previous subsection and introduced in Chapter 2. Different specifications $\mathbf{y}$ created essentially distinct situations: (1) a reachable design, (2) a perfect mapping but a non-reachable design, (3) a non-perfect mapping, and (4) a design with multiple minima. In the four cases the fine and the coarse model are as in (4.49) and (4.50), respectively. The specifications $\mathbf{y}$ are indicated in Table 2.1.

The values for the parameters in the TRMM scheme are as they appear in Figure 3.5: $\alpha = 1 + \tau > 1$, $\tau = 10^{-10}$, $\beta = 1/10$, $\lambda_{TR} = 1$ and $R_\lambda = r_\lambda = 2$. We set the maximum number of iterations to $N_{MAX} = 100$, and the damping parameter is not used ($\delta = 0$), except for the last example. Convergence results for the TRMM scheme are compared to those for the MM algorithm (no trust-region strategy). In all cases, the matrix A introduced in Remark 3 (Chapter 3) is now the identity. As commented in Section 4.4.2, the coarse model is not specially adapted to the fine one, and particularly the last example shows what adverse effects can be expected in a case like this one.

1. Reachable design. The TRMM scheme takes 12 iterations (13 $\mathbf{f}$-evaluations) until $\|\mathbf{x}_k - \mathbf{x}_\mathbf{f}^*\|_2$ is smaller than 10^{-6}. Figure 4.4 represents the history of $\|\mathbf{x}_k - \mathbf{x}_\mathbf{f}^*\|_2$ (left) and of λ_k (right). In contrast with all other examples shown, for this simple problem the trust-region strategy has no positive effect. Without the trust-region strategy a solution with similar quality is also obtained after 12 iterations.

2. Perfect mapping. The TRMM scheme takes 13 iterations (16 $\mathbf{f}$-evaluations) until $\|\mathbf{x}_k - \mathbf{x}_\mathbf{f}^*\|_2$ is smaller than 10^{-6}. The convergence history is shown in Figure 4.5. The behavior is similar to that for the reachable design: first the parameter λ does not decrease because of the initial dissimilarity between the fine and the coarse model and then, the reduction is monotonous. The matrices ΔF and ΔC remain well-conditioned. Without the trust-region strategy a solution with similar quality is obtained after 48 iterations.

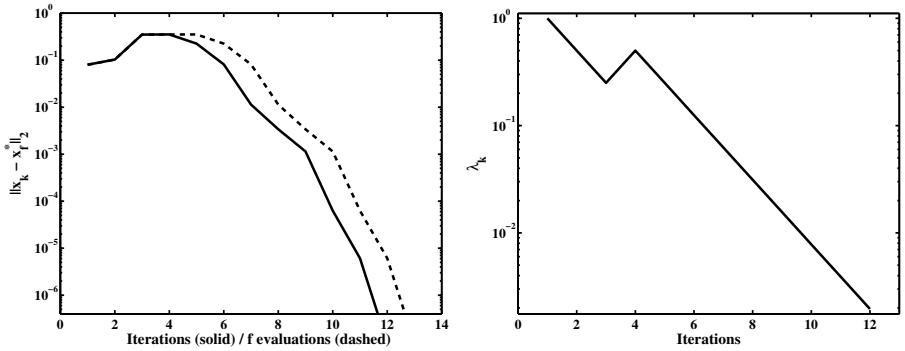


Figure 4.4: Reachable design. Left: Convergence history for $\|\mathbf{x}_k - \mathbf{x}_\mathbf{f}^*\|_2$ for the trust-region manifold-mapping (TRMM) algorithm. Right: history of λ_k for the same algorithm.

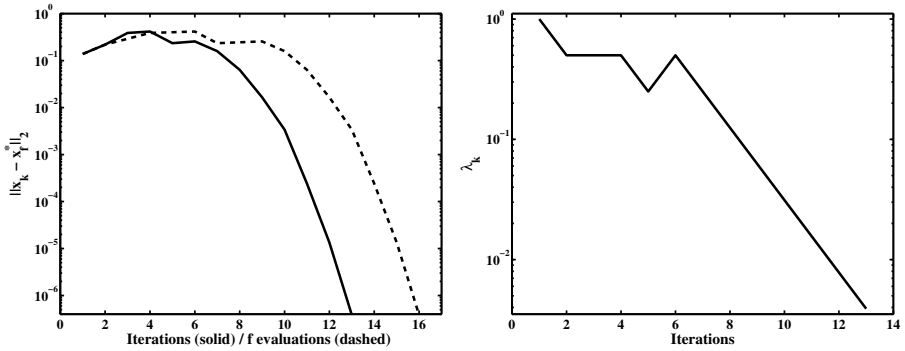


Figure 4.5: Perfect mapping and non-reachable design. Left: Convergence history for $\|\mathbf{x}_k - \mathbf{x}_\mathbf{f}^*\|_2$ for the trust-region manifold-mapping (TRMM) algorithm. Right: history of λ_k for the same algorithm.

3. Non-perfect mapping. The TRMM scheme takes 21 iterations (29 **f**-evaluations) until $\|\mathbf{x}_k - \mathbf{x}_\mathbf{f}^*\|_2$ is smaller than 10^{-6}. The convergence history is shown in Figure 4.6. In this case ΔF becomes very ill-conditioned and during the iteration process λ has to be significantly increased several times. Without the trust-region strategy a solution with similar quality is obtained after 63 iterations[2]. If we compare Figure 4.6 with Figure 4.3 we can appreciate that the TRMM algorithm yields a much more stable convergence history than the MM scheme.

[2]If the matrix A in the MM scheme is the null-matrix, the number of iterations is larger than 100 (see Figure 4.3).

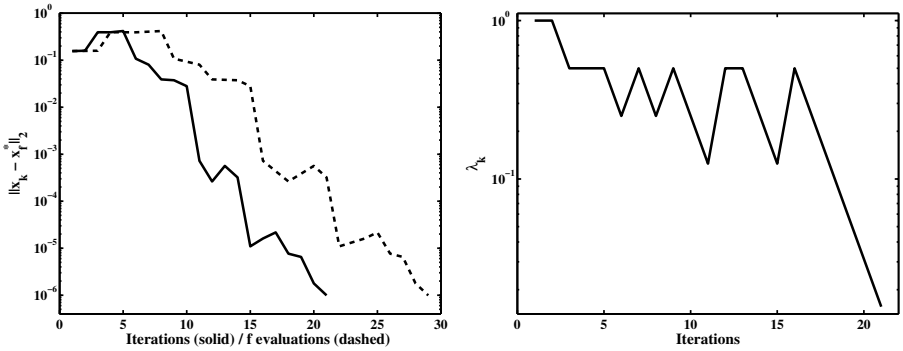


Figure 4.6: Non-perfect mapping. Left: Convergence history for $\|\mathbf{x}_k - \mathbf{x}_\mathbf{f}^*\|_2$ for the trust-region manifold-mapping (TRMM) algorithm. The TRMM algorithm yields a much more stable convergence history than the MM scheme (compare with Figure 4.3). Right: history of λ_k for the same algorithm.

4. Multiple minima. In this example the fine model optimization has two local minima. The behavior of the fine model near the global minimum is much different from that of the coarse model, which is more similar to the (other) local minimum. In this example TRMM without damping ($\delta = 0$) takes 53 iterations (66 $\mathbf{f}$-evaluations) until $\|\mathbf{x}_k - \mathbf{x}_\mathbf{f}^*\|_2$ is smaller than 10^{-6}. The stationary point obtained $\mathbf{x}_\mathbf{f}^*$ is the local minimum (*not* the global one). The convergence history is shown in Figure 4.7.

Using the damping parameter δ we can force the algorithm to find the global minimum. By taking $\delta > 0$ the method selects a path with smaller steps. The effect is stronger for larger δ. The iteration process arrives in the attraction area of the global minimum, but because of the large discrepancy between the behavior of $\|\mathbf{c}(\mathbf{x}) - \mathbf{y}\|$ and $\|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|$ in that region, the convergence is relatively slow. TRMM with damping ($\delta = 100$) takes 89 iterations (99 $\mathbf{f}$-evaluations) until $\|\mathbf{x}_k - \mathbf{x}_\mathbf{f}^*\|_2$ is smaller than 10^{-6}. Now, $\mathbf{x}_\mathbf{f}^*$ represents the *global optimum* and the convergence history is shown in Figure 4.8. Without the trust-region strategy the global optimum is approximated with a similar quality within 95 iterations (there is no solid theoretical reasoning that explains this convergence to the global minimum for this particular example).

4.5 Conclusions

With this chapter we finish the detailed study of manifold mapping. In Chapter 3 we introduced a number of possible algorithmic variants: original manifold mapping (OMM), manifold mapping (MM) and generalized manifold map-

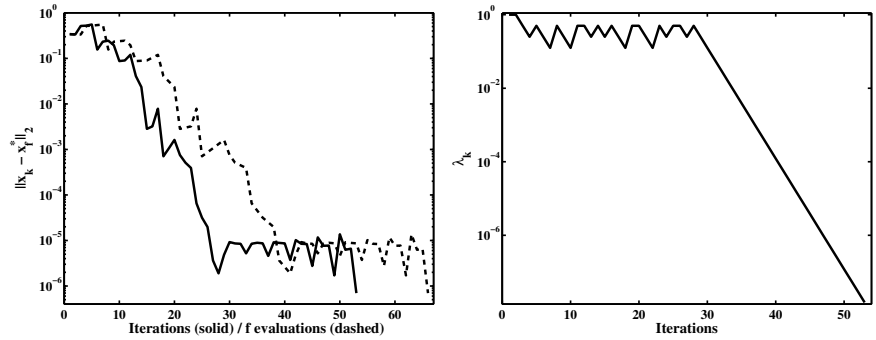


Figure 4.7: Multiple minima, damping parameter $\delta = 0$. Left: Convergence history for $\|\mathbf{x}_k - \mathbf{x}_\mathbf{f}^*\|_2$ for the trust-region manifold-mapping (TRMM) algorithm. Right: history of λ_k for the same algorithm.

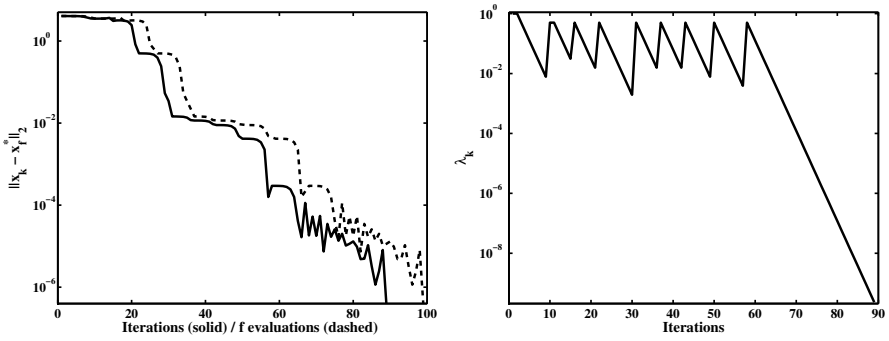


Figure 4.8: Multiple minima, damping parameter $\delta = 100$. Left: Convergence history for $\|\mathbf{x}_k - \mathbf{x}_\mathbf{f}^*\|_2$ for the trust-region manifold-mapping (TRMM) algorithm. Right: history of λ_k for the same algorithm.

ping (GMM). Moreover, two scheme extensions were considered: trust region manifold-mapping (TRMM) and constrained optimization with manifold mapping. In this chapter we have seen that the stationary points of all these schemes are local minimizers of the fine model cost function. Conditions for convergence have also been formulated for OMM, MM and GMM. By simple examples some of the theoretical aspects dealt with in the chapter have eventually been illustrated. In the following two chapters, the basic algorithms and extensions will be applied to a number of design problems of practical relevance.

Chapter 5

Practical Applications I: Optimization with Inexpensive Constraints

5.1 Introduction

In this chapter we apply some of the two-level optimization methods described and analyzed in Chapters 2, 3 and 4, to problems of practical relevance. Two different fields are considered: magnetostatics and steady state photon transport equations (Sections 5.2 and 5.3, respectively). In both cases we consider exact gradient information as unavailable. This circumstance, together with the intrinsic multi-level nature of a modeling procedure, makes this type of problems well suited for the space-mapping and the manifold-mapping approaches. The optimization constraints for the problems in this chapter are always of the box type, i.e., they are expressed in terms of constant upper and lower bounds for each component of the design variable $\mathbf{x}$. Therefore, the constraint manifolds $\mathbf{k_f}(\widehat{X})$ and $\mathbf{k_c}(\widehat{X})$ coincide and thus, there is no need to introduce a mapping between them (Section 3.3.2).

This chapter is structured as follows. In Section 5.2.1 we present a reachable design in magnetostatics and with that we verify that space mapping does perform efficiently and accurately. In the example in Section 5.2.2 the perfect mapping condition holds in such an approximate way that, within the tolerance required, the space-mapping solution coincides with the true optimum. In these two cases we expect space mapping and manifold mapping to perform equally well. The design problem in Section 5.2.3 combines the difficulties of those in the previous subsections (material nonlinearities and force computation) and it

represents a case where the space-mapping and manifold-mapping solutions differ significantly. In Section 5.3 we treat a simulation-based optimization in the field of photon transport. Besides rechecking the efficiency of the two-level approach, there we see that a proper choice of the coarse model can also be used to improve design problems that are badly scaled.

The examples in Section 5.2 have been taken from publications in the field of magnetics. They are a first step towards the designs proposed in Chapter 6 (Sections 6.2.1 and 6.2.2). These problems and other applications from electronics described in that chapter have in common that the constraints are given by functions that require a significant computational cost.

5.2 Finite-Element based applications

Optimization problems with objective functions based on finite-element simulations are a suited field for applying two-level approaches [7, 30, 50, 51, 56, 74, 86]. Accurate finite-element solutions are in many cases computationally expensive and it is not complicated to find cheaper and less accurate approximations of them (for example by increasing the size of the elements). In this section we apply space mapping and manifold mapping to three optimal design problems from magnetostatics proposed earlier in [30, 77] and we compare our results with other efficient and commonly used optimization techniques. The software package FemLab [31] is used for solving the discrete finite-element formulations.

5.2.1 A C-shaped circuit (EPE1)

This design problem (denoted by EPE1) was introduced in [30] to check the performance of the SM technique. EPE1 is a two-dimensional magnetostatics problem for a C-shaped circuit. We use it as a first comparison of SM with other minimization schemes. We show that the success of the SM technique depends on the similarity between the models considered and their relative computational cost. For this problem it is possible to improve the accuracy of the coarse model without significantly increasing its execution time. This apparently results in a more efficient SM based algorithm.

A permanent magnet with residual flux density $\mathbf{B}_r = 1.0$ T is placed between two ferromagnetic cores as shown in Figure 5.1. The sizes of the air gap, d and g, are taken as 10 and 1 mm, respectively. In the first experiment with EPE1 the core is assumed to be a linear material with magnetic relative permeability equal to 5000. The magnet is always considered linear with unit magnetic relative permeability. The design specifications are the flux densities at the air gap center and in the core[1], $\mathbf{B}_g$ and $\mathbf{B}_c$, respectively, and the magnet perme-

[1] Taking the upper left corner of the C-shaped circuit as the origin, the core flux density $\mathbf{B}_c$ is measured at $[\mathbf{x}_1 + d, -\mathbf{x}_3/2]$.

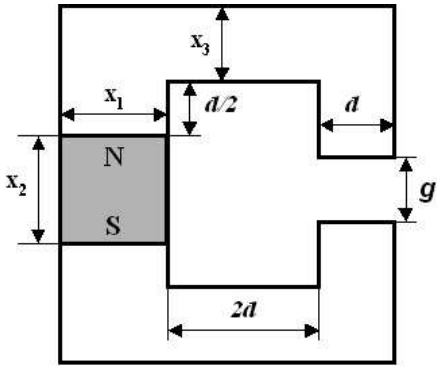


Figure 5.1: EPE1: a two-dimensional magnetostatics problem.

ance coefficient $\mathbf{P}_m$ (equivalently the magnet working point). The particular design objective is $\mathbf{y} = [\mathbf{B}_g, \mathbf{B}_c, \mathbf{P}_m] = [0.5, 1.0, 14]$. The design parameters $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3]$ are the dimensions of the magnet, $\mathbf{x}_1$ and $\mathbf{x}_2$, and the main width of the core $\mathbf{x}_3$. The control parameter spaces X and Z coincide and they are equal to the positive octant of $\mathbb{R}^3$.

The fine model $\mathbf{f}(\mathbf{x})$ is a finite-element resolution of a vector magnetic potential formulation of the magnetostatic equations [79]. The basis functions are quadratic Lagrangian interpolants and adaptive mesh refinement is applied, leading to discretizations of around 40000 degrees of freedom. This fine model takes into account phenomena like the fringing effect at the air gap and different flux leakages.

The coarse model $\mathbf{c}(\mathbf{z})$ is a magnetic equivalent circuit (MEC) [29], as used in [30]. Here, the flux density is assumed to be confined to the magnet, the core and the air gap, and to be constant in each of these three regions. The magnetic permeability in the core is taken as infinity. No fringing effect around the air gap or flux leakage is considered. Just by applying the divergence and Stokes' theorems to the integral form of the magnetostatic equations we obtain the coarse model

$$\mathbf{c}(\mathbf{z}) = \mathbf{c}([\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3]) = \left[\frac{\mathbf{B}_r\, \mathbf{z}_1\, \mathbf{z}_2}{g\, \mathbf{z}_1 + d\, \mathbf{z}_2}, \frac{d\, \mathbf{B}_r\, \mathbf{z}_1\, \mathbf{z}_2}{g\, \mathbf{z}_1\, \mathbf{z}_3 + d\, \mathbf{z}_2\, \mathbf{z}_3}, \frac{d\, \mathbf{z}_2}{g\, \mathbf{z}_1} \right], \tag{5.1}$$

where the first component approximates the flux density at the air gap, the second one the flux density in the core and the third one the magnet permeance coefficient. Computing the coarse optimum $\mathbf{z}^*$ is straightforward. It should be noted that the coarse model minimization time can be neglected with respect to that for the finite-element computation. The resulting value for $\mathbf{z}^*$ is found in Table 5.1.

	# fevals.	Final design (mm)
$\mathbf{z}^*$		[5.3571, 7.5000, 5.0000]
SM I	4	[7.9897, 7.5821, 6.5396]
SM II	6	[7.9797, 7.5808, 6.5381]
qN	6	[7.9891, 7.5821, 6.5394]
NMS	62	[8.0000, 7.5872, 6.5372]
DIRECT	186	[7.9806, 7.5823, 6.5432]

Table 5.1: Efficiency comparison between different optimization methods applied to EPE1. SM I : the ASM algorithm; SM II : the simplified ASM algorithm with $\mathbf{B}_k = I$; qN : a quasi-Newton scheme; NMS : Nelder-Mead simplex and DIRECT: a direct search method. The first column indicates the number of fine model evaluations required to solve the EPE1 problem for a tolerance $\tau = 0.001$ (approximately proportional to the total computing time). The second column represents the resulting vector of design parameters. For comparison, in the first row also the coarse optimum $\mathbf{z}^*$ is included.

Results

Two versions of the SM algorithm are compared with three standard optimization schemes. Because the number of design parameters is equal to the number of specifications, we may expect that the design is reachable (i.e., $\mathbf{f}(\mathbf{x}^*) = \mathbf{y}$) and therefore, that the condition for a perfect mapping holds. Thus, it is justified to use the stopping criterion adopted in [30]

$$\frac{\|\mathbf{p}(\mathbf{x}_k) - \mathbf{z}^*\|_2}{\|\mathbf{z}^*\|_2} \leq \tau\,. \tag{5.2}$$

The advantage of (5.2) over, for example, the residual criterion $\|\mathbf{f}(\mathbf{x}_k) - \mathbf{y}\|_2 / \|\mathbf{y}\|_2$ is that in the former the three vector components are well scaled (the optimal dimensions are comparable in size).

The optimization problem is solved by means of five different algorithms. ASM (see Figure 2.5) is implemented together with a variant in which $\mathbf{B}_k$ is taken as the identity in every iteration. (This is the special case (2.36) considered in Section 2.5.3.) A quasi-Newton method was also tried for solving $\mathbf{f}(\mathbf{x}^*) = \mathbf{y}$. The Jacobian was approximated by Broyden's method and for the initial estimates for the solution and for the Jacobian we took $\mathbf{z}^*$ and $J_\mathbf{c}(\mathbf{z}^*)$, respectively. As this method uses coarse model information as well, it cannot be considered completely different from two-level optimization. Since derivative information for the fine model is not easily obtained, two direct optimization schemes are used: the Nelder-Mead simplex method (NMS) [55][2] and the global

[2]The MatLab function *fminsearch* is used.

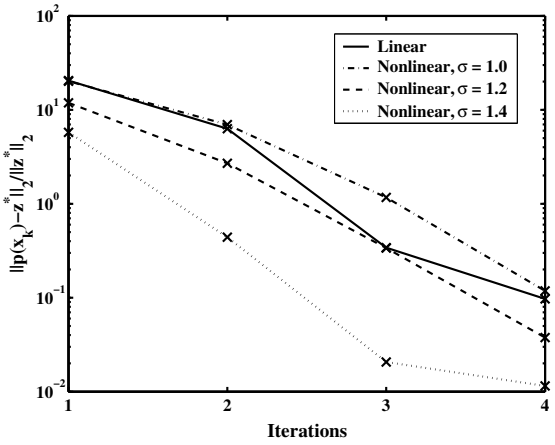


Figure 5.2: Problem EPE1 solved for a nonlinear fine model. The convergence history of the cost function is shown for different coarse models.

search algorithm DIRECT [52][3]. The former takes the coarse optimum as the initial guess. The latter does not require a starting value but the space where the solution has to be found (Z or a subset) should be specified. (We took the cube centered in $\mathbf{z}^*$ and with the point $\mathbf{z}^*/2$ at a corner.) Table 5.1 shows the number of fine model evaluations for each method until the stopping criterion with $\tau = 0.001$ is satisfied. It should be emphasized that the number of evaluations is approximately proportional to the total computing time. ASM turns out to perform best, and the quasi-Newton scheme still seems to be efficient. Nelder-Mead shows a slow convergence. DIRECT's global optimization nature justifies its poor results.

In the second experiment with EPE1, the core is assumed to be made of steel and, hence, it shows a nonlinear $\mathbf{B}-\mathbf{H}$ characteristic. The *fine* model is refined by taking this fact into account. We want to show how important the selection of the coarse model is for the SM approach. Therefore, we improve the coarse model above in two different ways. First, in the part of the core where its width is $\mathbf{x}_3$, the magnetic permeability is now finite and derived from the $\mathbf{B}-\mathbf{H}$ curve. (The treatment of the rest of the core seems to be of no relevance for the results.) Second, the fringing effect is taken into account in the coarse model by increasing the width of the air gap d to $\sigma\, d$, with $\sigma > 1$ a correction factor. The experiments show that taking into account the fringing effect is more important than adding the nonlinearity. In Figure 5.2 the convergence history is shown for ASM applied to the nonlinear fine model with four different coarse models. The first one is

[3]MatLab implementation by D.E. Finkel [42] is employed.

the same linear model as above in (5.1), the second takes the core nonlinearity into account and the other two incorporate the fringing effect at the air gap by increasing its cross section.

All these coarse models have a negligible computational cost compared with the fine model. Hence, the number of SM iterations within ASM is still a good measure for the final computing time. The proper selection of the coarse model is an important key for the SM technique since convergence can be greatly improved by an adequate choice. However, a better coarse model requires additional knowledge and insight about the system modeled.

5.2.2 A coreless actuator (EPE2)

This optimization problem (here denoted by EPE2) was proposed in [77] as a test for nonlinear optimization applied to electro-magnetic actuator design. EPE2 is a three-dimensional magnetostatic problem for an axisymmetric coreless actuator. An electro-magnetic actuator [23] is a device that converts electro-magnetic energy into mechanical force and motion. This one is called *coreless* because actuators usually have ferromagnetic cores (see Sections 5.2.3, 6.2.1 and 6.2.2).

Here we study EPE2 with a double aim. Apart form seeing the effect of SM compared with other efficient minimization methods, we also want to find a proper strategy for the construction of an appropriate coarse model. In view of our experience with the EPE1 experiment, the latter is probably a delicate task. We will apply a whole hierarchy of coarse models and this will provide us with some insight into the question how coarse a model can be selected in a problem like EPE2. We will verify that in this problem, the solution found by space mapping has a similar cost function value to those for the MM and GMM algorithms.

The geometry of the actuator is shown in Figure 5.3. It consists of a moving cylindrical magnet with length $l_m = 36$ mm and radius $r_m = 14.7$ mm and two fixed toroidal coils with inner radius $r_c = 15$ mm. The geometry is further specified by the design parameters $\mathbf{x}_1$ (internal thickness), $\mathbf{x}_2$ (half distance), $\mathbf{x}_3$ (length) and $\mathbf{x}_4$ (external thickness). We combine these four parameters in the control variable vector $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4]$. The design in [77] is bounded by

$$\begin{cases} 2.5\,\text{mm} \leq \mathbf{x}_1 \leq 20\,\text{mm} \\ 5.0\,\text{mm} \leq \mathbf{x}_2 \leq 25\,\text{mm} \\ 2.5\,\text{mm} \leq \mathbf{x}_3 \leq 50\,\text{mm} \\ 2.5\,\text{mm} \leq \mathbf{x}_4 \leq 20\,\text{mm} \end{cases} . \tag{5.3}$$

Both the sets X and Z coincide with this region.

The permanent magnet in [77] has a constant residual flux density $\mathbf{B}_r = 0.35$ T and it is assumed to be linear with a magnetic relative permeability equal to one. The current density in the coils is taken constant, with a

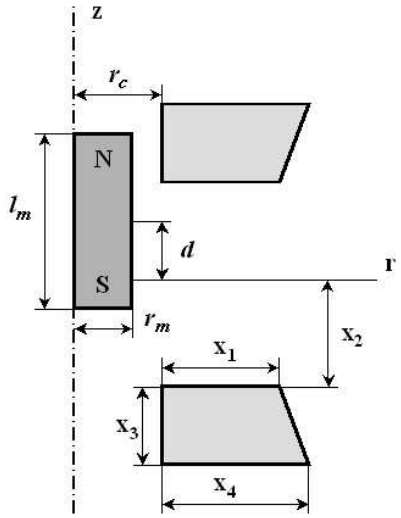


Figure 5.3: EPE2: geometry of the coreless actuator.

value $\mathbf{J}_c = 2.68$ A/mm^2. The magnet can be moved from the center, along its axis over some distance d.

The specification $\mathbf{y}$ is the force response exerted on the rod at different locations d, which should be constant and equal to 5 N. By *force response* we mean the force (statically) computed when the magnet is positioned over a distance d along the z-axis. For this purpose the force is determined at 20 regularly distributed displacements d_i between 0 and 15 mm from the center. Thus, both the specification and the computed force are vectors of length 20.

Again, finite elements are employed in the fine model. The axisymmetric vector potential formulation of the magnetostatic equations is solved using second order Lagrangian interpolants as basis functions. The mesh is adaptively refined and the number of degrees of freedom is around $7000 - 18000$. The force is computed via Lorentz' formula [29]. This is done by post-processing the finite-element computation.

A hierarchy of coarse models

Although we know that the coarse model should be an easy-to-calculate approximation of the fine one, it is difficult to establish beforehand how (in)accurate it may be, still providing an acceleration of the optimization process. In order to investigate this question, here we build a coarse model that can be tuned as (im)precise as the user desires by means of two parameters.

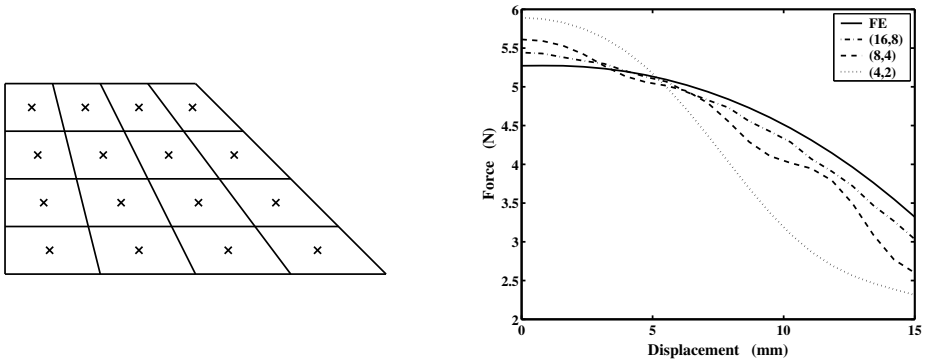


Figure 5.4: A hierarchy of models for EPE2.
The left figure shows the cells ($N_c = 5$) in the coil for which the flux density is considered constant, with its value computed at the barycenter (indicated by a cross). These points also indicate the position of the loops that approximate the coil. The right figure shows the force response for different coarse models, compared with the fine model force response. In the right figure legend, FE refers to the finite-element simulation (fine model).

The first parameter, N_m, refers to the treatment of the magnet. In EPE2 the effect of the magnetization is equivalent to a surface current density $\mathbf{J}_m = \mathbf{B}_r/\mu_0$ [29]. In our coarse model, this current density is approximated by N_m current loops. They are equally distributed over the vertical surface of the magnet, each with radius r_m and carrying a current $\mathbf{J}_m\, l_m/N_m$. The magnetic flux density is computed by the Biot-Savart law.

The second parameter, N_c, relates to the representation of the coil. Since the coil volume-current density is constant and because of the axial symmetry of the problem, the force computation is reduced to the integration of the radial component of the flux-density caused by the magnet over each coil cross-section. This cross section is divided into $(N_c-1)\times(N_c-1)$ pieces. The regular partition is made by $N_c - 1$ layers of $N_c - 1$ cells, as shown in Figure 5.4 for $N_c = 5$. The coarse model considers the flux density to be piecewise constant over this partitioning. The value for the radial flux density in every cell is computed at its barycenter. The integral over the cross-section is approximated by the sum of the calculated values, weighted by their cell area. This is equivalent to substituting the coil by $(N_c - 1)^2$ loops situated in the barycenter of each region and with a current equal to the one in that small area.

Thus, a coarse model is identified by the pair of integer parameters (N_m, N_c). As long as the values for N_m and N_c are increased, the coarse model will be more accurate but also computationally more expensive. This behavior is reflected in

Figure 5.4 where $\mathbf{x}_1 = \mathbf{x}_4 = 16$ mm, $\mathbf{x}_2 = 18.50$ mm and $\mathbf{x}_3 = 15$ mm, and the coarse models $(N_m, N_c) = (4,2)$, $(8,4)$ and $(16,8)$ are used. In that figure, FE refers to the finite-element simulation (fine model).

At first sight, the response associated with model $(4,2)$ seems very different from the one obtained with the finite-element simulation. But apparently this does not influence its use as a coarse model. As will be seen in the next subsection, this very coarse model performs more efficiently than $(8,4)$. The SM function compensates for the strong model misalignment and guarantees the final accuracy.

Results

In this section we compare the efficiencies for several optimization techniques and different SM strategies. The SM technique performs best. Surprisingly, extremely inaccurate coarse models lead to very efficient optimization schemes.

Unlike EPE1, the design is not reachable and hence, we cannot expect a perfect mapping. Therefore, the criterion (5.2) is no longer adequate for comparing SM with some other schemes. As in [77], we take $F(\mathbf{x}) = \|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|_2 / \|\mathbf{y}\|_2$ for the cost function. Except for the methods in [77], in the experiments the solution process is stopped when this measure is below 0.08.

It is interesting to notice that there is a whole region in the parameter space X where the fine model satisfies the specifications with the desired accuracy. We see in Table 5.2 how the final design parameters obtained by the different techniques are not unique. In fact it appears that there exists a whole manifold of possible solutions in X, so additional design constraints can be imposed. No regularization is needed in this problem to cope with difficulties arising from such manifold of solutions. In Section 6.2.1 we will see a situation where this is not the case.

The techniques analyzed in [77] are the penalty method (PM) and the method of moving asymptotes (MMA). PM reduces the cost function to 0.074 in 14 fine model evaluations and MMA to 0.083 in only 9. Both algorithms need an initial guess. In the paper [77] two of them are considered. The given results start from the point $\mathbf{x}_1 = \mathbf{x}_4 = 16$ mm, $\mathbf{x}_2 = 18.50$ mm and $\mathbf{x}_3 = 15$ mm and for both PM and MMA they are better (less fine model evaluations for an equal or smaller cost function value) than those corresponding to the other starting point.

The NMS is modified (by penalizing the cost function) in order to deal with the constraints. The initial guesses are again the same as in [77]. Now the best results are obtained when the starting point is $\mathbf{x}_1 = \mathbf{x}_3 = \mathbf{x}_4 = 18$ mm and $\mathbf{x}_2 = 10$ mm. The cost function decreases to 0.070 in 12 fine model evaluations. The performance of DIRECT is much better than in EPE1. It appears that the existence of multiple solutions makes the global search easier. The cost function drops to 0.068 in 11 fine model evaluations. A method based on sequential quadratic programming (SQP) [65][4] is also tested. It yields to 0.047 in 7 fine

[4]The MatLab function *fmincon* is used.

	# f evals.	Cost function	Final design (mm)
PM	14	0.074	[17.20, 24.10, 23.90, 15.87]
MMA	9	0.083	[16.77, 23.50, 24.27, 16.48]
NMS	12	0.070	[16.13, 20.53, 16.46, 16.51]
DIRECT	11	0.068	[17.08, 21.67, 26.25, 11.25]
SQP	7	0.047	[10.25, 17.50, 34.00, 10.25]
SM-(8,4)	7.9	0.050	[11.25, 21.75, 24.49, 17.08]
SM-(4,2)	2.5	0.041	[14.33, 20.02, 27.16, 10.75]
SM-(6,2)	1.5	0.039	[17.73, 21.67, 20.97, 11.25]

Table 5.2: Efficiency comparison between different optimization methods applied to EPE2. PM: penalty method; MMA: method of moving asymptotes; DIRECT: direct search method; NMS: Nelder-Mead simplex; SQP: sequential quadratic programming (MatLab's *fmincon*) and SM-(N_m, N_c): space mapping with coarse model (N_m, N_c). The first column indicates the total amount of computational work expressed in the *equivalent* number of fine model evaluations (approximately proportional to the total computing time). The second column shows the minimum cost function obtained. The design tolerance is 0.08, except for PM and MMA for which comparable results were taken from [77]. The third column represents the final design parameters.

model evaluations. The initial guess is the same as for the NMS.

Applying the SM technique entails solving two types of optimization subproblems. The first one is the computation of the coarse optimum $\mathbf{z}^*$. Because the coarse model is very cheap to evaluate we can afford a global minimization method such as DIRECT for that task. The second subproblem concerns the SM function $\mathbf{p}$. Since usually this function is close to the identity in a great part of the domain (see Figure 2.1), the computation of $\mathbf{p}(\mathbf{x})$ brings no significant problems. The NMS is used there and the coarse optimum $\mathbf{z}^*$ is always taken as the initial guess.

Testing several coarse models in the hierarchy shows that the inaccurate $(6, 2)$ performs best in optimizing EPE2, within the tolerance required: only two iterations (about 2.5 fine model evaluations) are needed to reach a cost function value of 0.026, a value clearly acceptable in practice. We notice that the computational cost associated to all coarse model evaluations is taken into account in Table 5.2 (a fine model evaluation is approximately equivalent in computational time to 20, 200 or 300 of the coarse models $(8, 4)$, $(6, 2)$ or $(4, 2)$, respectively).

The manifold-mapping algorithms MM and GMM are also applied to EPE2. In GMM we estimate the Jacobians of both the fine and the coarse model by means of Broyden's method. The results are shown in Table 5.3. Both the MM and GMM scheme yield a design solution with a small cost function (0.022) in

Table 5.3: Efficiency comparison between space mapping and manifold mapping applied to EPE2.

	# (f, c) evals.	# f evals.	Cost function
MM	(2, 132)	2.4	0.022
GMM	(2, 148)	2.5	0.022
SM	(3, 150)	3.5	0.023

MM: manifold mapping; GMM: generalized manifold mapping; SM: space mapping. The second column shows the total amount of *equivalent* fine model evaluations needed in the optimization (approximately proportional to the total computing time). The coarse model used is $(N_m, N_c) = (4, 2)$ and it is approximately 300 times faster than the fine one.

less than three equivalent fine model evaluations. The number of coarse model evaluations is larger for the second scheme because in each iteration step the coarse model Jacobian is approximated by finite differences. In order for SM to obtain a comparable value for the cost function as for the two manifold-mapping algorithms, all coarse models from Table 5.2 are considered. The best option found is the coarse model with $(N_m, N_c) = (4, 2)$ which requires more than three equivalent fine model evaluations.

The results obtained with SM are equally cheap and as accurate as those with MM. From this fact we can conclude that (for the accuracy required) the perfect mapping condition practically holds for EPE2. However, this is not the case for the actuator design in the next section.

5.2.3 An automotive actuator (EPE25)

We denote this design problem by EPE25. Automotive actuators typically generate high levels of force and they can be used in devices such as electromagnetic switches, relays, valves, etc.. Figure 5.5 is a schematic view of the cylindrical plunger electromagnet. It was originally introduced in [77] and it consists of a core and a plunger, both made of iron, and a copper coil. The sizes $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2]$ have to be optimized in order to assure a maximum magnetic thrust force versus displacement in the z-axis. The volume of the device is kept constant. The specifications $\mathbf{y}$ are a constant force response of 100 N, for six vertical plunger displacements. The design space X proposed in [77] is the rectangle $[2.5,\ 4.5] \times [12,\ 18]$, where all the bounds have been specified in mm. The set Z coincides with X. More details on the problem are found in [76, 77].

The fine model $\mathbf{f}(\mathbf{x})$ is based on a static force computation for the six plunger displacements. This force is evaluated numerically by quadratic Lagrangian finite elements with adaptive refinement. The number of degrees of freedom in every

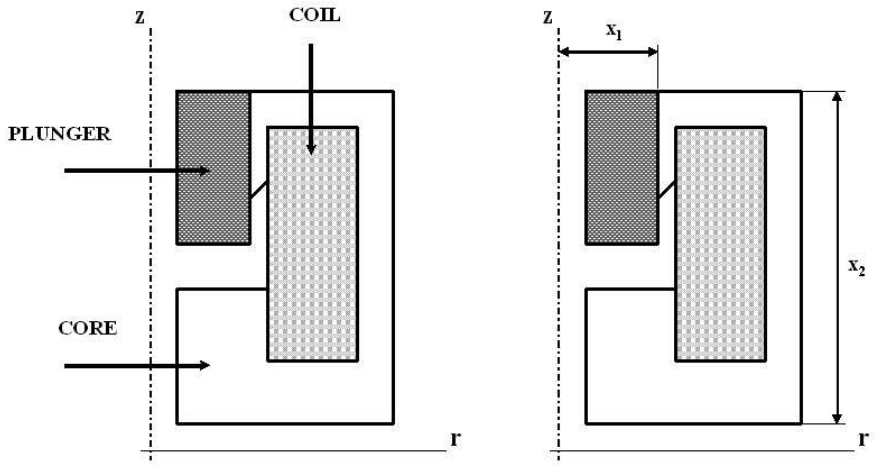


Figure 5.5: EPE25: Axisymmetrical geometry and design variables of the automotive actuator.

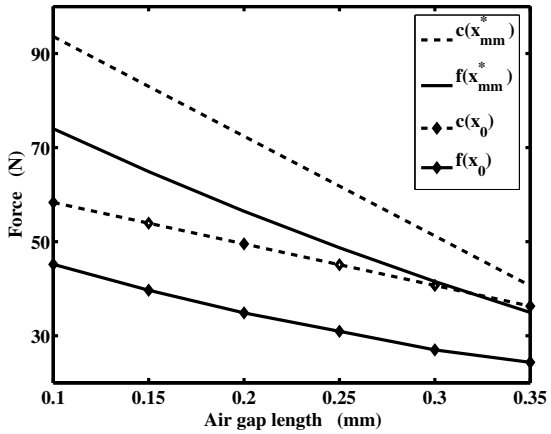


Figure 5.6: Force responses for the fine and coarse model for EPE25 in two points of X.

case is around $5000 - 7000$ yielding three digits of accuracy in the force. The objective function here is $F(\mathbf{x}) = \|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|_2 / \|\mathbf{y}\|_2 \times 100$, i.e., a relative Euclidean norm of the model discrepancy.

The coarse model $\mathbf{c}(\mathbf{z})$ is based on a magnetic equivalent circuit (MEC) [29]

Table 5.4: Force optimization of the automotive actuator EPE25 in the original design space.

	# fevals.	Cost function
MMA	7	49.0
PM	9	48.8
EM	13	48.8
RQP	7	48.8
SQP	3.0	48.5
SM	1.0	48.5
MM	1.0	48.5

MMA: method of moving asymptotes; PM: penalty method; EM: ellipsoid method; RQP: recursive quadratic programming; SQP: sequential quadratic programming; SM: space mapping; MM: manifold mapping. The first column indicates the total amount of computational work expressed in the *equivalent* number of fine model evaluations (approximately proportional to the total computing time). The second column shows the minimum cost function obtained.

derived from the actuator. The MEC is nonlinear, in the sense that a $\mathbf{B}-\mathbf{H}$ curve is used for the plunger. An infinite core magnetic permeability is taken and neither fringing nor leakage effects are included. A linear interpolation is applied for the force between the points for maximum and minimum displacement. This coarse model is around 400 times faster than the fine one. In Figure 5.6 the fine and coarse model responses are shown for two points in the design space X: the geometrical center $\mathbf{x}_0$ and the optimum obtained by MM. Each optimization involving the coarse model is solved by sequential quadratic programming (SQP).

Table 5.4 compares the performance of seven minimization methods. The figures in the first column (amount of work expressed in number of equivalent fine model evaluations) are approximately proportional to the total computing time. All seven techniques yield solutions with almost the same objective function value. The first two (MMA: method of moving asymptotes, PM: penalty method) correspond with the methods from [77] commented in Section 5.2.2, and the next two (EM: ellipsoid method, RQP: recursive quadratic programming) to other schemes studied by the same author in a different work [76]. The coarse model optimum $\mathbf{z}^* = [4.5\,\text{mm},\ 18\,\text{mm}]$ has a lower fine cost function than any of the solutions given by the methods above. SM and MM converge to $\mathbf{z}^*$ (i.e., the coarse model optimum $\mathbf{z}^*$ is a minimizer of $F(\mathbf{x})$) after just one fine model evaluation (the number of coarse model evaluations is also taken into account in the amount of work shown in Table 5.4). SQP applied to the fine model and starting from the coarse model optimum performs also very efficiently.

Table 5.5: Force optimization of the automotive actuator EPE25 in the modified design space.

	# fevals.	F(x)	Final design (mm)
SQP	12.1	45.0	[5.00, 16.45]
MM	3.2	45.0	[5.00, 16.61]
SM	6.9	45.3	[5.00, 15.00]

SQP: sequential quadratic programming; MM: manifold mapping; SM: space mapping. The first column indicates the total amount of computational work expressed in the *equivalent* number of fine model evaluations. The second and third columns show the minimum cost function obtained and the resulting vector of design parameters, respectively. MM and SM stop when the reduction in the cost function value is smaller than 0.01. SQP takes the coarse model optimum $\mathbf{z}^*$ as its initial guess and iterates, for comparison purposes, until the cost function value decreases below 45.0.

Next, the space X is slightly enlarged to $[2.5,\ 5] \times [15,\ 30]$ (again in mm) in order to obtain a design in which the coarse optimum does not solve the optimization process[5]. Thus, a better comparison can be made between the three methods that perform similarly above. Now, the coarse optimum is the point [4.68 mm, 17.72 mm] which has an associated cost function of 46.5. The optimization results are given in Table 5.5. MM and SQP yield both a solution with the same quality, but MM is almost four times faster. SM improves also SQP in efficiency but the result obtained is not the optimal design. This experiment corroborates MM as an efficient method with the additional property of convergence towards the fine optimum. GMM, with the fine model Jacobian estimated via Broyden's method, performs identically to MM.

5.3 A design problem in photon transmission

This optimization problem, introduced in [38], has practical relevance in photonics. Unlike the previous designs, its associated fine model is not based on finite elements. The phenomenon under analysis is the photon transmission and absorption in random media [70]. For example, the propagation of light in biological tissues can be studied within this framework. We will verify again that both the MM and SM approaches solve the design problem efficiently and that the former yields a more accurate solution than the latter. Additionally, we will

[5] This modification of the design space is introduced for testing purposes. As observed in [76], large values for $\mathbf{x}_1$ could lead to devices with significant mass and a poor dynamic regime performance.

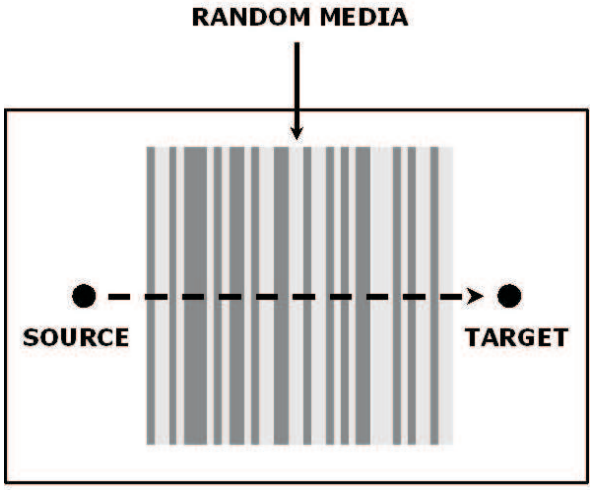


Figure 5.7: The photon transmission takes places through a cylinder which contains random media. The two-dimensional design variable $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2]$ represents the absorption coefficients of the random media considered.

see that coarse models can be used as an efficient means for improving situations where a bad model scaling complicates the whole optimization process.

The setup is a cylinder of length L which contains two different materials distributed randomly (see Figure 5.7). Photons are emitted at one end of the cylinder (source) and collected at the other end (target). During their travel, some of the photons will be scattered and some others will be absorbed. The one-dimensional steady state transport equation [70] is solved over the cylinder. Due to the stochastic nature of the media, Monte Carlo simulations are performed.

The model output is two-dimensional: the first component is the average transmission coefficient and the second one is its variance. We perform N direct numerical simulations of the transport equation in order to estimate these expected values. The two-dimensional design variable $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2]$ represents the absorption coefficients of the random media considered. The sets X and Z coincide and they are equal to $[1, 10]^2$.

The model is considered to be fine enough for practical purposes when $L = 1$ (complete cylinder) and when the number of simulations is $N = 100000$. Thus, we characterize the fine model $\mathbf{f}(\mathbf{x})$ by the pair $(L, N) = (1, 100000)$. We can obtain cheaper (and less accurate) models either by considering a smaller domain for solving the transport equation (i.e., $L < 1$) or by performing much less transport simulations (i.e., $N < 100000$). This is shown in Figure 5.8 and Figure 5.9 where the transmission coefficient and its variance are computed for different

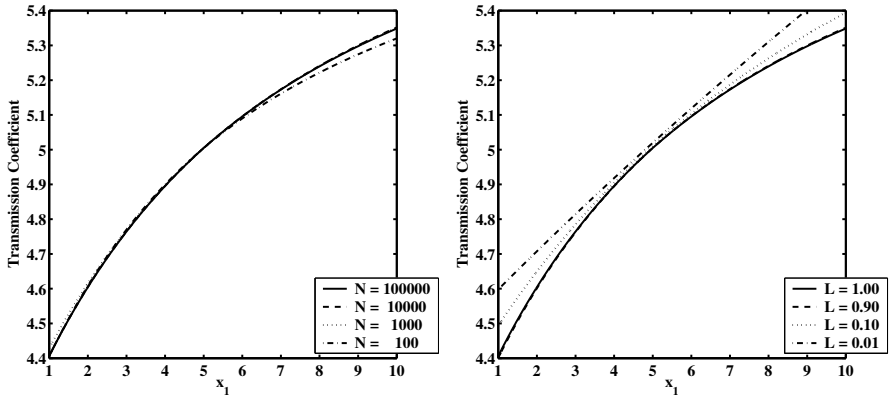


Figure 5.8: Different models for the transmission coefficient. The second component of the design variable $\mathbf{x}_2$ is kept constant and equal to five. The length L is taken as one in the left figure. The number of realizations N is 100000 for the right plot.

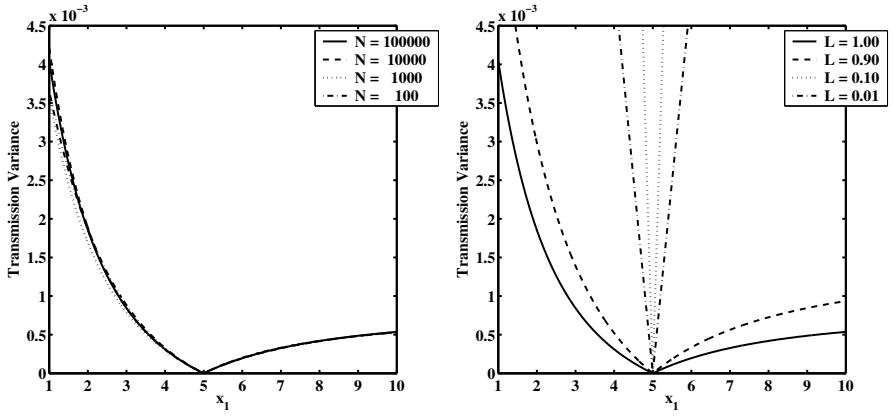


Figure 5.9: Different models for the transmission variance. The second component of the design variable $\mathbf{x}_2$ is kept constant and equal to five. The length L is taken as one in the left figure. The number of realizations N is 100000 for the right plot.

values of L and N. There, the second component of the design variable $\mathbf{x}_2$ is kept constant and equal to five. (When $\mathbf{x}_1 = \mathbf{x}_2$ the cylinder contains just one medium and then the transmission variance is zero.) We observe that for the transmission coefficient, L and N can be reduced to a very low value without

Table 5.6: Results for the reachable photon design problem.

	#f evals.	$F(\cdot)$	**Final design**
NMS	74.2	0.000	[3.4549, 7.6548]
SM	55.8	0.000	[3.4556, 7.6546]
MM	16.9	0.000	[3.4507, 7.6560]

Table 5.7: Results for the non-reachable photon design problem.

	#f evals.	$F(\cdot)$	**Final design**
NMS	66.3	0.020	[1.0002, 5.7685]
SM	15.7	0.020	[1.0015, 5.7680]
MM	13.6	0.020	[1.0034, 5.7674]

NMS: Nelder-Mead simplex; SM: space mapping; MM: manifold mapping. The first column indicates the *equivalent* number of fine model evaluations required to solve the optimization problem. $F(\cdot)$ represents the cost function. The stopping criterion is in every case $h_k = \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2 < 10^{-4}$. The coarse model employed is around forty times faster than the fine one.

(apparently) losing much accuracy in the model. The transmission variance is very sensitive to the varying of the length L, but concerning the number of realizations N, the behavior is analog to the one observed in Figure 5.8. Thus, the same pair $(L,\ N)$ can be acceptable for computing the transmission coefficient but not for its variance.

We will take a coarse model $\mathbf{c}(\mathbf{z})$ given by the pair $(L,\ N) = (1,\ 100)$. This model is around forty times faster than the fine one. The cost function considered is again $F(\mathbf{x}) = \|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|_2/\|\mathbf{y}\|_2 \times 100$, a relative measurement of the model discrepancy. Every optimization process stops when the increase of the Euclidean norm of the design parameters is smaller than 10^{-4}, i.e., $h_k = \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2 < 10^{-4}$.

In this design problem we first try a reachable design. The specifications are $\mathbf{y} = \mathbf{f}([3.4567,\ 7.6543]) = [7.0348,\ 0.0002]$. We observe from Figure 5.8 and Figure 5.9 (and also from $\mathbf{y}$) that the transmission variance takes much smaller values than the transmission coefficient. This difference in size suggests that a scaling could be necessary since the cost function considered gives more emphasis to the first component of every model than to the second one. Consequently, values far away from the fine model optimum could yield, for practical purposes, the same cost function value.

We can use inexpensive coarse model evaluations in order to approximate a scaling for the fine model. We proceed as follows. An average value of the transmission variance is first computed based on the coarse model. Then, the output of both fine and coarse models is weighted according to that value. (From

another perspective, a scaled cost function is used.)

The results of this reachable two-dimensional optimization problem are shown in Table 5.6. Space mapping (with NMS for the coarse model optimizations) leads to the fine model optimum (again perfect mapping holds). However, SM takes almost the same computational effort as NMS. The MM algorithm (with NMS for all the coarse model optimizations) yields a speed-up of around a factor four. This acceleration value is similar to those obtained in the previous section.

If no scaling is introduced, the problem is ill-conditioned and the solutions obtained differ clearly from the fine optimum ([3.2784, 7.7052] for NMS and [3.3043, 7.6975] for MM).

If manifold mapping is applied with the coarse model characterized by $(L,\ N) = (0.01, 100000)$ (which is around thirty times faster than the fine one), the solution [3.4516, 7.6557] is obtained after 85.6 equivalent fine model evaluations. The associated computational cost is even higher than that for the one-level NMS scheme. If we use the model given by $(L,\ N) = (1,\ 100)$ just for computing the coarse model optimum $\mathbf{z}^*$ and then we continue with $(L,\ N) = (0.01,\ 100000)$, the resulting procedure accelerates NMS by a factor of three ([3.4526, 7.6554] is obtained after 26.9 equivalent fine model evaluations). We can infer that intermediate coarse models can be used for computing better initial guesses and with that improve a previously observed speed-up.

We also try a non-reachable design, given by the specifications $\mathbf{y} = [5,\ 0.004]$. The second component of $\mathbf{y}$ is about twenty times larger than in the reachable case and this seems to be enough for not needing the additional weighting. The results without scaling are shown in Table 5.7. The speed-up factors for SM and MM are five and four, respectively. With the use of scaling, the convergence behavior is about the same and the limit point is [1.0000, 5.6710] for SM and [1.0007, 5.6967] for MM.

Though in this design problem a hierarchy of models is characterized by the pair $(L,\ N)$, the difference in computational cost between the fine and coarse models considered is not big enough to fully exploit the advantages of the multi-level approach. Moreover, a really simple model as $(L,\ N) = (1,\ 100)$ presents no important convergence difficulties in the two-level iteration. An intermediate model would presumably be of not much help. If higher accuracy is desired in solving the transport equation, the distance between the fine and coarse models will increase, and then, we believe that a multi-level scheme could offer some noticeable improvement over the two-level strategy.

5.4 Conclusions

In this chapter we have verified that the two-level optimization approach is an efficient strategy for solving various design problems of practical relevance. The constraints in these problems do not require special care because they are inex-

pensive to evaluate. The first three examples (EPE1, EPE2 and EPE25) have a cost function based on solving a finite-element discretization of the magnetostatics equations. EPE1 is a reachable design that can be solved by classical space mapping. It also shows how improving a coarse model can lead to an improvement in performance. EPE2 and EPE25 are examples of electro-magnetic actuator design. EPE2 represents a case where the perfect mapping condition holds approximately and thus, the solution found by space mapping coincides (within the tolerance needed) with the true optimum. In EPE2 a hierarchy of models is available and there even a very simple model can accelerate the optimization process significantly. In EPE25 there is no perfect mapping and classical space mapping yields a solution with a cost function noticeably larger than the fine model optimum. The last optimization problem is taken from the field of photon transport in random media. The fine model requires the solution of the discretized steady state transport equation. The results are consistent with those from the previous three examples, in the sense that space mapping and manifold mapping are also efficient minimization techniques. Moreover, we see with the photon transport example that coarse models can be beneficial when used with model scaling purposes. The design problems in the next chapter are comparable in complexity to those studied here. However, they carry the additional difficulty of having optimization constraints that are described by functions of considerable computational cost.

Chapter 6

Practical Applications II: Optimization with Expensive Constraints

6.1 Introduction

The space-mapping and manifold-mapping algorithms are here applied to optimization problems with constraint functions that are expensive to compute (approaches in Section 2.4 and Section 3.3.2). The chapter is structured as follows. In Section 6.2 we study two linear actuators. The one in Section 6.2.1 has axial symmetry and therefore, it is described by the magnetostatic equations in two dimensions. We consider two different design problems for this actuator. The first one addresses a force response optimization with inexpensive constraints. (In order to include a continuous actuator study and due to the briefness of this first problem we prefer to present it at this point.) In the second problem the mass of the device is minimized. This optimization problem has seven design variables and is subject to constraints that require solving the magnetostatic equations. The actuator in Section 6.2.2 has to be analyzed by the full three-dimensional magnetostatic equations. The design of this actuator is similar in nature to the mass minimization of the axisymmetric one. In Section 6.3 we treat some recently proposed optimization problems in electronics [73]. These represent more complex designs because, both the models and the associated constraints, involve time-consuming circuit simulations.

Figure 6.1: A cylindrical voice-coil actuator consisting of a ferromagnetic core, permanent magnet and coil.

6.2 Electro-Magnetic actuator designs

In this section we deal with two different configurations of a short-stroke linear actuator. The cost function is, in both cases, inexpensive to compute (mass of the device) but the constraints require finite-element simulations (two- and three-dimensional in Section 6.2.1 and 6.2.2, respectively). Both designs are efficiently solved by means of space and manifold mapping, with magnetic equivalent circuits as coarse models. The software package FemLab [31] is used for the two-dimensional partial differential equations in Section 6.2.1. The optimal solution for the design in that section has already been manufactured and tested by research partners [40]. The three-dimensional simulations in Section 6.2.2 are carried out with Maxwell 3D [2].

6.2.1 A cylindrical voice-coil actuator (EPE3)

We refer to the two designs concerning the voice-coil actuator generically as EPE3. The two optimization problems will be denoted as EPE3a and EPE3b. Linear voice-coil actuators are non-commutated electromechanical devices that provide cogging-free force outputs directly proportional to the applied current. An axisymmetric variant consisting of a ferromagnetic core, a permanent magnet and a moving coil is shown in Figure 6.1. The ferromagnetic material is nonlinear and its magnetic behavior is characterized by a given **B**–**H** curve. The permanent magnet is magnetized in the vertical direction. The coil, actuated by the magnetic force, moves along the z-axis in the gap of the core. The position of the coil relative to the top of the core is denoted by D (see Figure 6.2). The minimal and maximal positions of the coil are referred to by D = s_{min} and D = s_{max}, respectively. The range between s_{min} and s_{max} is referred to as the *stroke*. Due to the axisymmetric geometry, the force has an axial component only.

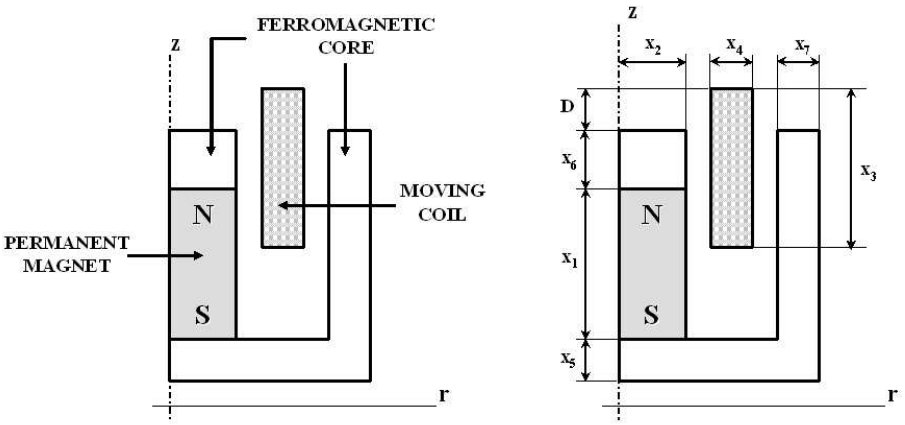


Figure 6.2: EPE3: Axisymmetric geometry and design variables of the cylindrical voice-coil actuator.

The design variables are also found in Figure 6.2: $\mathbf{x}_1$ and $\mathbf{x}_2$ denote the height and radius of the magnet, $\mathbf{x}_3$ and $\mathbf{x}_4$ the height and thickness of the coil and $\mathbf{x}_5$, $\mathbf{x}_6$ and $\mathbf{x}_7$ are the sizes of the core. The bounds for the design variables are given in [37]. Two additional linear inequality constraints are imposed. The first one guarantees sufficient magnetic coupling when the coil is in its top position by imposing a lower limit on the height of the coil

$$\mathbf{x}_3 \geq \mathbf{x}_6 + s_{max} - s_{min} \,. \tag{6.1}$$

The second inequality constraint ensures the geometry to remain feasible when the coil is in its lowest position by imposing an upper limit on the height of the coil

$$\mathbf{x}_3 \leq \mathbf{x}_1 + \mathbf{x}_6 - s_{min} \,. \tag{6.2}$$

Numerical values for the parameters s_{min} and s_{max} used in the design problems are listed in [37]. The materials properties of the actuator can be found in the same reference.

For this actuator we consider two different design problems, EPE3a and EPE3b. The first one consists in a force response optimization with *inexpensive* constraints and will be solved only for one or two design variables. In addition, it illustrates some basic aspects of the manifold-mapping theory and it shows why a trust-region strategy is required when the optimization problem is ill-conditioned. Though EPE3a does not fit into the type of problems treated in this chapter, we prefer to present it here rather than divide the actuator study in two parts. The second experiment describes a practical design of an actuator with seven control variables.

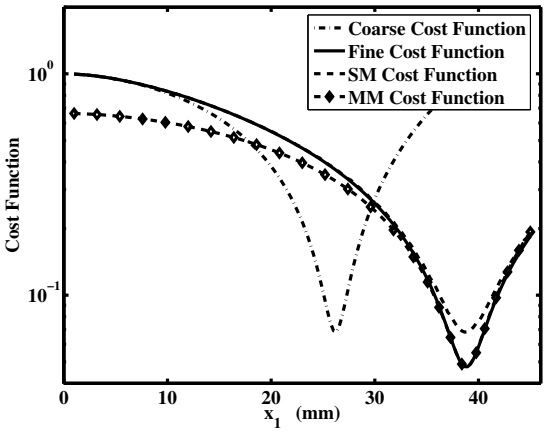


Figure 6.3: Cost functions associated with the coarse (simplified FE), fine, SM and MM surrogate models for the one-dimensional EPE3a problem.

Force response optimization (EPE3a)

In the first problem we allow the coil to move over a 4 mm stroke, i.e., $0 \leq \mathrm{D} \leq 4\,\mathrm{mm}$. The coil location is represented at nine equidistant points in this interval. Values for the design variables have to be found such that the force response is as flat and as close to $\mathbf{y} = 24\,\mathrm{N}$ as possible. For this force level the effect of the ferromagnetic core has to be taken into account in an accurate fine model description. The permanent magnet demagnetization curve is assumed to be linear with $\mu_r = 1$. We first solve this optimization problem for the design variables $\mathbf{x} = [\mathbf{x}_1]$ and then for $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2]$.

In the fine model $\mathbf{f}(\mathbf{x})$ the force is computed by Lorentz' formula at each of the nine coil positions. For this purpose we use a vector potential formulation of the magnetostatic equations discretized by quadratic Lagrangian triangular finite elements (FE). The number of degrees of freedom is between 8000 and 11000, yielding three digits of accuracy in the force. The cost function is $F(\mathbf{x}) = \|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|_2 / \|\mathbf{y}\|_2 \times 100$. The design space X is a line segment in the one-dimensional optimization problem and a rectangle in the two-dimensional one. In both cases the set Z coincides with X. Details about the material parameters and the bounds for the design variables are given in [37].

We consider two coarse models. The first one is a FE-based model in which the nonlinear material characteristic of the core is linearized. Depending on the number of Newton iterations required, this model is a factor between thirty and fifty cheaper than the fine model. The second coarse model is a magnetic equivalent circuit (MEC). This model has negligible computational cost compared to the fine model.

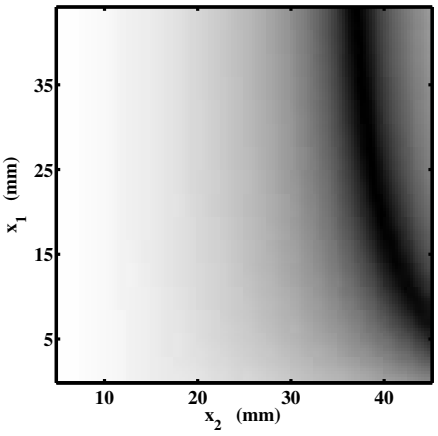


Figure 6.4: Logarithm of the fine model cost function for the two-dimensional EPE3a problem. Dark shading indicates low values for the cost function.

We start with the one-dimensional design problem, only considering changes in the first design variable $\mathbf{x}_1$, i.e., $\mathbf{x} = [\mathbf{x}_1]$. The purpose is to illustrate how the cost function for the coarse model, as used and corrected in the MM algorithm, closely approximates the fine model cost function in a region close to $\mathbf{x}^*$. In Figure 6.3 four cost-related functions in $\mathbf{x}_1$ are plotted, viz. the coarse model $\|\mathbf{c}(\mathbf{x}_1) - \mathbf{y}\|_2/\|\mathbf{y}\|_2$, the fine model $\|\mathbf{f}(\mathbf{x}_1) - \mathbf{y}\|_2/\|\mathbf{y}\|_2$, the SM $\|\mathbf{c}(\mathbf{p}(\mathbf{x}_1)) - \mathbf{y}\|_2/\|\mathbf{y}\|_2$ and the MM $\|\mathbf{S}(\mathbf{c}(\mathbf{x}_1)) - \mathbf{y}\|_2/\|\mathbf{y}\|_2$ cost functions. The figure shows that the coarse and fine model optima are clearly different, and it illustrates how, in a region close to $\mathbf{x}^*$, MM improves the coarse cost function, as compared with SM.

In this one-dimensional problem, both SM and MM, using either the linear FE or the MEC as coarse model, converge in four fine model evaluations. Compared with the sequential quadratic programming (SQP) or the Nelder-Mead simplex (NMS) method, the two-level methods are similar in performance and both deliver a speed-up factor between four and five.

The two-dimensional design problem, with the design variables $\mathbf{x}_1$ and $\mathbf{x}_2$, i.e., $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2]$, reveals that the problem is ill-conditioned. In Figure 6.4, the cost function is shown. The dark region in this figure represents a long and steep valley in the design space where the cost function takes about the same small value. This means that, within the accuracy required, there is no unique solution. The design variables found by optimization depend on factors such as the starting guess and algorithm used. Uniqueness can be restored by regularization, for example by imposing a minimal mass constraint.

Numerical results comparing MM with NMS, SQP and SM for this two-

Table 6.1: Two-dimensional optimization of the voice-coil actuator (problem EPE3a).

	# f evals.	Cost function
NMS	24	4.6
SQP	31	4.6
SM_1	9	4.6
SM_2	6	6.5
MM_1	9	4.6
MM_2	4	4.6

NMS: Nelder-Mead simplex; SM_1: space mapping with the simplified FE as coarse model; SM_2: space mapping with the MEC as coarse model; MM_1: manifold mapping with the simplified FE as coarse model; MM_2: manifold mapping with the MEC as coarse model. The first column shows the total amount of *equivalent* fine model evaluations needed in the optimization (approximately proportional to the total computing time). The coarse model based on the simplified FE is between thirty and fifty times faster than the fine one. The coarse model obtained with the MEC has a negligible computational cost with respect to the fine one.

dimensional problem are given in Table 6.1. As starting guess for the optimization procedures we used the values obtained by optimizing the MEC model. To stabilize the convergence of MM, the Levenberg-Marquardt method [65] is used. The best results in terms of computational efficiency (speed-up by a factor of six with respect to NMS) are obtained using MM with the MEC as coarse model. The NMS and SQP algorithms are clearly less efficient than the two-level procedures, and SM does not always guarantee a correct solution.

The need of a trust region strategy

Here we show that the original manifold-mapping approach might fail in absence of a trust-region strategy. We consider the same two-dimensional force optimization problem as above but with a considerably simpler coarse model than the two proposed there. It has been deliberately selected this way in order to illustrate what terrible effects the lack of a trust-region strategy may cause in an ill-conditioned problem like this one. The coarse model is a linear interpolation of the FE model with the linear core characteristic curve. Only four points are used in this interpolation: the four corners of the control space X. The computational cost associated with the construction and evaluation of this coarse model can be considered negligible compared to the evaluation of the fine one. In Figure 6.5 the fine and coarse model responses are shown for $\mathbf{x}^*$ and $\mathbf{z}^*$.

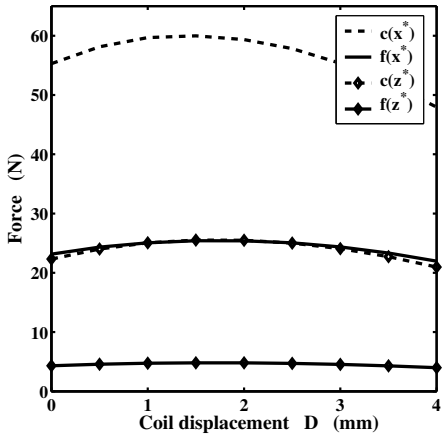


Figure 6.5: Fine and coarse model force responses at $\mathbf{x}^*$ and $\mathbf{z}^*$ for EPE3a. The model discrepancy is large: in both cases the coarse force response takes values significantly larger than the fine one.

With this very rude coarse model, trust-region manifold mapping (TRMM) yields a minimal value for the cost function (4.7) after six iterations (eight fine model evaluations). The original manifold-mapping algorithm does not converge for this coarse model, and it oscillates between two corners of the design space X. With SQP, taking as initial guess the coarse model optimum $\mathbf{z}^*$, a value for the cost function of 4.7 is obtained after four iterations (24 fine model evaluations).

Optimization with seven design variables (EPE3b)

The second problem, introduced in [39], has seven design variables, i.e., $\mathbf{x} = [\mathbf{x}_1, \ldots, \mathbf{x}_7]$, and several nonlinear equality and inequality constraints. The total mass of the actuator has to be minimized. Thus, the cost function is the total mass of the device (notice that $\mathbf{y} = 0$). The nonlinear constraints are specified by two equalities and three inequalities. The force at coil position D = 4.25 mm should be kept at 5 N and the average magnetic flux density in three regions of the core (fully specified in [37]) should not exceed 1 T. The mass of the coil is constrained at 10 g. Material parameters and bounds for the design variables are also found in [37].

Because the constraints involve expensive computations, it makes sense to proceed in a two-level fashion as far as the evaluation of these constraints is concerned. The fine and coarse constraints will be combined in the versions of the SM and MM algorithms for this type of design problems (see Section 2.4 and Section 3.3.2). In the fine model, the constraints are evaluated by the same FE model used in the previous subsection. In the coarse model, the constraints

Table 6.2: Seven-dimensional optimization of the voice-coil actuator (problem EPE3b).

	# $\mathbf{k_f}$ evals.	F(x)	Final design (mm)
SQP	56	81.86	[8.543 9.793 11.489 1.876 3.876 3.197 2.524]
SM	7	81.11	[8.500 9.786 11.450 1.883 3.838 3.200 2.497]
MM	6	81.45	[8.500 9.784 11.452 1.883 3.860 3.202 2.515]

SQP: sequential quadratic programming; SM: space mapping; MM: space mapping. The first column indicates the total amount of computational work expressed in the *equivalent* number of fine constraints evaluations. The second and third columns show the minimum cost function (mass of the actuator) obtained and the resulting vector of design parameters, respectively. SQP takes the coarse model optimum $\mathbf{z}^*$ as its initial guess. The versions for the SM and MM algorithms are those aimed at design problems with constraints expressed by functions that are expensive to evaluate. The coarse constraints are based on a magnetic equivalent circuit and the associated computational cost is negligible when compared to that for the fine constraints.

are based on a MEC model. Since the mass computation carries no significant computational cost we can take $\mathbf{f} \equiv \mathbf{c}$. The design space $\widehat{X}$ is a hypercube in $\mathbb{R}^7$. The set $\widehat{Z}$ coincides with $\widehat{X}$.

Numerical results for this problem are given in Table 6.2 where the performance of SM and MM is compared with that of SQP. The optimization process was stopped if the relative decrease of both the cost function and Euclidean norm of the design parameters was smaller than $5 \cdot 10^{-3}$, provided that the constraints were met with three digits of accuracy. Each coarse model related optimization is accomplished by using SQP. SM and MM show a similar behavior: convergence is reached in seven and six equivalent fine constraints evaluations respectively. SQP alone converged within 56 equivalent fine constraints evaluations. Because in this problem the number of constraints ($m_{\mathbf{k}} = 5$) is smaller than the number of design parameters ($n = 7$), we expect the SM solution to coincide with the fine model optimum $\mathbf{x}^*$ (the assumptions for the special case in Section 2.4 are fulfilled). We recognize this to be the case because SQP, SM and MM yield the same solution. But even in this situation, MM offers an additional advantage over SM. The computation of the SM function $\mathbf{p}$ is sometimes a very delicate issue [7]. MM replaces it by the identity, and in a realistic problem like this one, the efficiency results are comparable. The optimal solution for this actuator has been manufactured and tested by research partners [40].

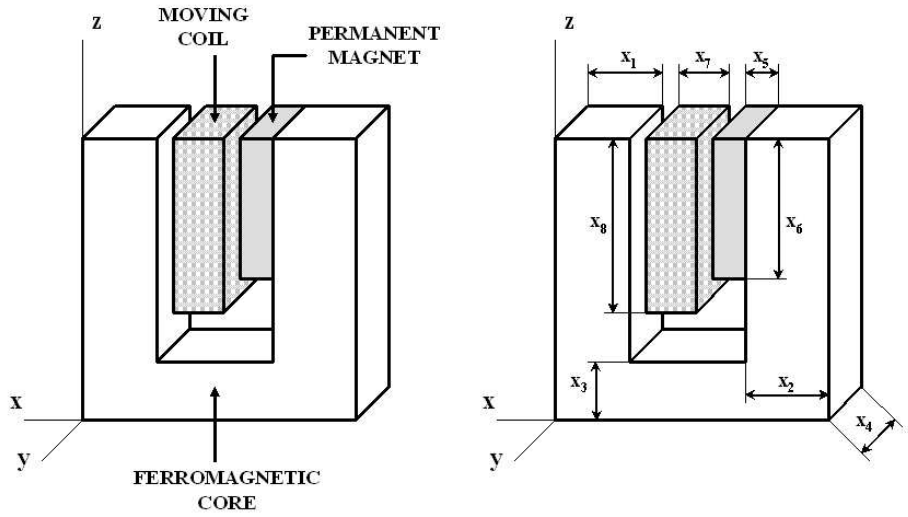


Figure 6.6: EPE4: Geometry and design variables of the rectangular voice-coil actuator. The actuator is symmetric with respect to the planes yz and xz.

6.2.2 A rectangular voice-coil actuator (EPE4)

We denote this problem as EPE4. The design is very similar in nature to EPE3b, but with the additional difficulty that the fine model is based on a time-expensive three-dimensional FE discretization (there is no symmetry that, as in EPE2, EPE25 and EPE3, permits a FE dimension reduction). EPE4 is a rectangular linear actuator and it consists of a ferromagnetic core, a moving coil and a permanent magnet (see Figure 6.6). In some practical applications the spatial region for placing the actuator is determined a priori. Therefore, a cylindrical device might not be the best option when the three spatial dimensions differ significantly. The rectangular actuator EPE4 represents a versatile solution in these cases. The core is nonlinear, the permanent magnet is magnetized in the x-direction and the coil moves vertically. The EPE4 design is introduced in [41] and it is solved efficiently with MM. The original SM approach yields a good approximation of the true optimum. Output polynomial mapping (OPM), an additional two-level optimization technique inspired in SM and MM, is also presented in the mentioned reference.

The design variable $\mathbf{x}$ is eight-dimensional, i.e., $\mathbf{x} = [\mathbf{x}_1, \ldots, \mathbf{x}_8]$, and all its components are indicated in Figure 6.6: $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$ and $\mathbf{x}_4$ refer to the core, $\mathbf{x}_5$, $\mathbf{x}_6$ and $\mathbf{x}_4$ to the magnet, and $\mathbf{x}_7$, $\mathbf{x}_8$ and $\mathbf{x}_4$ to the moving coil. The objective is to minimize the actuator mass. There are one equality and four inequality constraints. The equality constraint is stated in terms of a force output. The

Table 6.3: Eight-dimensional optimization of the rectangular voice-coil actuator (problem EPE4).

	# $\mathbf{k_f}$ evals.	$\mathbf{F(x)}$	Final design (mm)
SM	10	473	[11.38 3.85 4.07 72.46 6.63 15.39 4.75 21.16]
MM	9	473	[11.96 3.44 5.20 75.00 6.36 15.00 4.46 20.00]
OPM	10	472	[11.56 3.38 5.13 75.00 6.39 15.28 4.41 20.28]

SM: space mapping; MM: manifold mapping; OPM: output polynomial mapping. The first column indicates the total amount of computational work expressed in the *equivalent* number of fine constraints evaluations. The second and third columns show the minimum cost function (mass of the actuator) obtained and the resulting vector of design parameters, respectively. All the algorithm versions are those aimed at design problems with constraints expressed by functions that are expensive to evaluate. The coarse constraints are based on a magnetic equivalent circuit and they can be computed between 2500 and 7500 times faster than the fine constraints.

nonlinear inequality constraints specify upper limits for the coil mass and for the average flux density in three regions of the core. The inexpensive constraints involve upper and lower bounds for the design variable and a linear inequality constraint, similar in nature as (6.1) for EPE3, that helps to reduce the variation of the magnetic coupling. A more complete problem description can be found in [41].

Since the fine model $\mathbf{f}(\mathbf{x})$ involves a rather simple mass computation, we can consider $\mathbf{f} \equiv \mathbf{c}$. The same remark can be applied to the coil mass constraint. Second order Lagrangian tetrahedral adaptively refined finite elements (FE) based on a scalar potential formulation of the magnetostatic equations are used to evaluate the remaining fine model constraints. The discretization yields, for example at the optimal solution $\mathbf{x}^*$, around 10000 elements (about 12000 degrees of freedom), resulting in computations of the force and magnetic flux density with two digits of accuracy. The coarse model constraints $\mathbf{k_c}(\mathbf{z})$ are based on a magnetic equivalent circuit and they can be computed between 2500 and 7500 times faster than those for the fine model. The design space $\widehat{X}$ is a hypercube in $\mathbb{R}^8$. The set $\widehat{Z}$ coincides with $\widehat{X}$.

We show in Table 6.3 the results for the actuator optimization. Output Polynomial Mapping (OPM) is the hybrid variant of SM and MM proposed in [41]. The stopping criterion for all the algorithms is a constraint violation or a change in the fine model response smaller than one per cent. (Though this criterion could in general cause a premature stop in the optimization process, we observe that this is not the case for EPE4). Every coarse model optimization is solved with a combination of SQP and CONDOR [89], the latter being an extension of Powell's

UOBYQA algorithm [72]. The computing effort for a coarse model optimization is negligible when compared with one evaluation of the fine constraints. Hence, the figures indicated in the first column of Table 6.3 express approximately the total amount of computational work. The SM algorithm oscillates around the optimum and fails to converge with the required accuracy (the algorithm is stopped at iteration number ten). MM and OPM perform similarly and they efficiently yield the optimal actuator design.

6.3 Electronic designs

In this section we apply the two-level optimization approach to electronic design problems and we suggest coarse models that can be used in these situations. In many electronic circuits, the proper device operation is specified by a number of constraints that very often are expensive to evaluate (for example, in some occasions the voltage in a node should be larger than a certain threshold). The cost function is usually a measure of the power consumed. Therefore, in these optimization problems, unlike in any other we have studied in this work so far, both models and constraints are non-trivial. This fact makes the designs here attractive also from a theoretical point of view.

In this section we analyze three design problems and the last two, recently introduced in [73], are of practical relevance. The models and constraints used are based on commercial simulators[1] and their computation is not specially expensive (a few seconds). When these simulators are called in the optimization process, a significant amount of time is dedicated to other tasks than solving an electronic circuit (e.g., dumping results into a file). For this reason, we prefer to express the total optimization cost in terms of the number of fine and coarse simulations performed, together with an indication of their relative computational complexity (in absence of the phenomena mentioned above).

Nevertheless, the designs in this section are solved efficiently in terms of the *total* computing time. For the optimization problems in Section 6.3.2 and Section 6.3.3, we find feasible solutions at least ten times faster and with equal or smaller cost function values than in [73]. The results expressed in terms of the number of fine and coarse simulations required, serve as an indication of the performance of a two-level strategy in more complicated electronic design problems (where the computational cost is mainly associated with that of the simulator).

6.3.1 A digital inverter (ASCO1)

This rather simple optimization problem will be denoted as ASCO1. The circuit in Figure 6.7 is a digital inverter. It consists of two CMOS transistors M_p and M_n

[1]For Section 6.3.1, Section 6.3.2 and Section 6.3.3 the electronic simulators used are Pstar [69], Gnucap [33] and HSPICE [61], respectively.

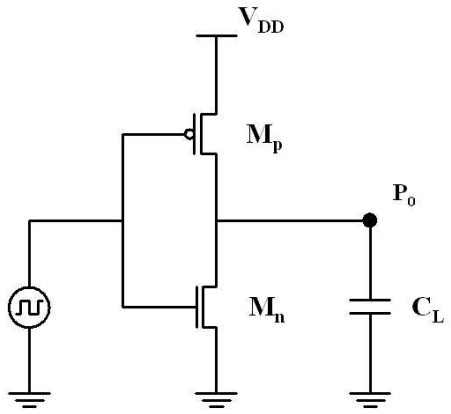


Figure 6.7: ASCO1: Digital CMOS Inverter. It consists of two CMOS transistors M_p and M_n (p- and n-type, respectively) operating as switches.

(p- and n-type, respectively) and a load capacitor C_L. The transistors act like switches: high (low) voltage values in the 2 V input square wave open (close) M_n and close (open) M_p. The input is, thus, accordingly inverted.

The design is stated in terms of a minimization of the total supplied power. The model response is a scalar and the specification $\mathbf{y}$ can be taken as zero. Because the supplied voltage V_{DD} is constant (and equal to 2 V), in order to obtain the total supplied power we just have to compute the current from V_{DD}. The constraints guarantee the practical inversion of the input signal. They are two-dimensional and they are expressed by means of some thresholds for the voltage in node P_0 at two different times. The computation of the supplied power and of the constraints require transient circuit simulations. The model used for the transistors is compatible with the MOS Level 1 available in SPICE version 2G6 [87]. The design variable $\mathbf{x}$ is one-dimensional and it represents the internal channel width of the transistor M_p (the internal width of the transistor M_n is $\mathbf{x}/3$). The set $\widehat{X}$ is the interval [1000 μm, 10000 μm] and it coincides with $\widehat{Z}$. The internal channel length for both transistors is 0.35 μm. The load capacitance value C_L is 10 pF and the operating temperature is 25° Celsius.

The transient circuit analysis for the fine model $\mathbf{f}(\mathbf{x})$ and constraints $\mathbf{k_f}(\mathbf{x})$ consists of four cycles with a time resolution of 2500 samples per cycle (the frequency of operation is 850 MHz). The fine cost function is $F(\mathbf{x}) = |\mathbf{f}(\mathbf{x}) - \mathbf{y}|$. The coarse model $\mathbf{c}(\mathbf{z})$ and constraints $\mathbf{k_c}(\mathbf{z})$ are computed via a transient simulation with also four cycles but only 10 samples per cycle (we can thus expect an amount of computational work of around 250 times smaller than in the fine simulation). In all cases an adaptive high order backward differentiation formula (BDF) is

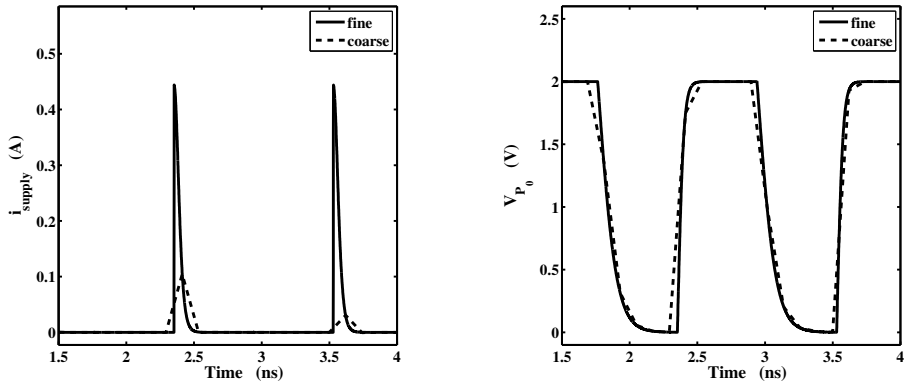


Figure 6.8: Similarity between fine and coarse models and constraints for ASCO1 when the design variable is the fine optimum $\mathbf{x}^*$. The left figure shows the supplied current. The right figure represents the voltage in node P_0.

used for integrating the circuit equations with Pstar. The power is measured in the fourth cycle. The constraints result from the voltage in node P_0 at one and three cycle quarters, also in the last cycle. (The signal at $\mathbf{t} = 3.82$ ns has to be greater than 1.95 V and at $\mathbf{t} = 4.41$ ns lower than 0.05 V.) In Figure 6.8 we give an impression of the similarity between fine and coarse models and constraints. Though the differences in the voltage in node P_0 seem small, the discrepancy in the supplied current is significant.

The manifold-mapping algorithm for constrained optimization yields the solution $\mathbf{x}^* = 3858\,\mu$m after 7 fine and 188 coarse transient simulations. The same optimal design is obtained with sequential quadratic programming (SQP) (with the coarse model optimum $\mathbf{z}^* = 3279\,\mu$m as initial guess) in 8 fine and 21 coarse transient simulations. Because the SQP algorithm performs often fast in one-dimensional problems (specially with a good initial guess) we can consider that MM has solved this design efficiently. In the following two more complicated examples, the difference between the results for SQP and MM will become more evident.

In a second experiment with the inverter we take into account design uncertainties (also known as *design corners*). These uncertainties imply additional circuit simulations and they will be considered only for the fine model and constraints. In this case we analyze corners for the supplied voltage V_{DD}. The fine model and constraints are based on three different simulations, each of them with a different value for the supplied voltage (2.0 V, 2.1 V and 2.2 V). The less favorable result is considered separately for both the model and the constraints. With that, we approximate the worse situation in a design where the supplied voltage V_{DD} may take any value in the interval [2.0, 2.2]. The coarse model

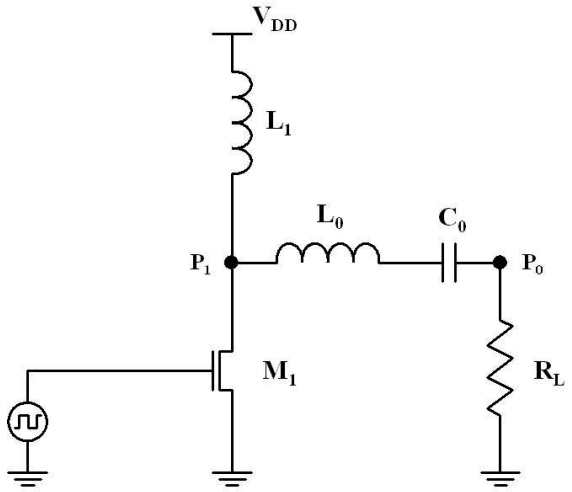


Figure 6.9: ASCO2: Class-E power amplifier. It consists of a CMOS transistor M_1, two inductors L_0 and L_1, one capacitor C_0 and a load resistor R_L.

and constraints are computed after only one transient simulation with a supplied voltage of 2.0 V (and with a computational work around 750 times smaller than in the fine simulation).

The optimization results are almost identical to those reported in the previous paragraph. The fine model optimum $\mathbf{x}^*$ takes again a value of 3858 μm and it is obtained by manifold mapping in 7 fine and 194 coarse transient simulations. SQP, having the coarse model optimum $\mathbf{z}^* = 3279$ μm as initial guess, takes 8 fine and 21 coarse transient simulations for computing the true optimum $\mathbf{x}^*$.

With the inverter in this section we have illustrated a simple case of an optimization problem with non-trivial model and constraints, and uncertainties in some of the design parameters. We have also suggested a manner of constructing coarse models in electronics which, within a framework such as manifold mapping, may lead efficiently to optimal design solutions.

6.3.2 A class-E power amplifier (ASCO2)

We will denote this design problem as ASCO2. Class-E power amplifiers [73, 81] are widely used in the radio frequency and microwave fields because of their high efficiency. In the amplifier in Figure 6.9 we can see a transistor M_1 followed by a resonant filter L_0-C_0. The transistor acts as a switch and drives the current through the inductor L_1.

The design problem consists in minimizing the supplied power (i.e., the model

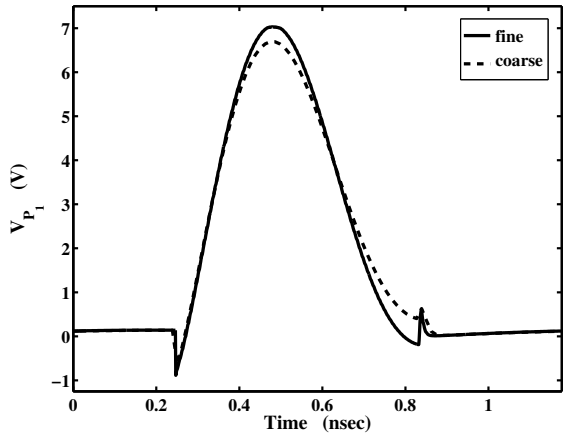


Figure 6.10: Transient voltages in node P_1 for the fine and coarse simulations for ASCO2 when the design variable is the fine model optimum $\mathbf{x}^*$. The conditions ensuring that the device is an amplifier are based on different features of this waveform.

response is a scalar and $\mathbf{y} = 0$). The constraints ensure that the circuit behaves like an amplifier and that it delivers an output power larger than a minimum value. They are of inequality type and they are defined over $\mathbb{R}^5$ (and based on three voltages and one voltage derivative, the four of them in node P_1, and on the total output power in node P_0). In order to obtain the supplied power we need to compute the current through L_1 (the supplied voltage V_{DD} is constant and equal to 2 V). The nature of the amplifier implies in all cases expensive transient simulations. The design variable $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5]$ is defined over $\mathbb{R}^5$: $\mathbf{x}_1$ is the internal drawn channel width of the transistor, $\mathbf{x}_2$ and $\mathbf{x}_3$ the inductances L_1 and L_0, respectively, $\mathbf{x}_4$ the capacitance C_0 and $\mathbf{x}_5$ the load resistance R_L. For a more complete problem description see [73].

The fine model response $\mathbf{f}(\mathbf{x})$ and constraints function $\mathbf{k_f}(\mathbf{x})$ result from a transient simulation where the input wave is a square signal of 50 cycles with a time resolution of 1000 samples per cycle (the frequency of operation is 850 MHz). The cost function is $F(\mathbf{x}) = |\mathbf{f}(\mathbf{x}) - \mathbf{y}|$. The coarse model $\mathbf{c}(\mathbf{z})$ and constraints function $\mathbf{k_c}(\mathbf{z})$ use only five cycles with 100 samples per cycle. In both cases, the backward Euler scheme is used for the integration of the corresponding differential equation with Gnucap. In the second simulation the pure computation of the model and constraints function is 100 times faster than that in the first one. In Figure 6.10 we show the transient voltages (last cycle) in node P_1 for the fine and coarse simulations with $\mathbf{x} = \mathbf{x}^*$. Though the differences may seem small, $\mathbf{x}^*$ is only feasible for the fine constraints (i.e., $\mathbf{k_f}(\mathbf{x}^*) \geq 0$ but $\mathbf{k_c}(\mathbf{x}^*) \not\geq 0$).

The manifold-mapping variant for constrained optimization yields the solution $\mathbf{x}^* = [17530\ \mu\text{m}, 3.54\ \text{nH}, 1.33\ \text{nH}, 63.1\ \text{pF}, 3.03\ \Omega]$ after 7 fine and 1891 coarse transient simulations. The associated (fine) supplied power is 0.694 W. For the coarse model computation we use SQP, having as initial guess a feasible point obtained with the global optimizer differential evolution (DE) [85]. We consider SQP alone for the rest of the minimization procedures within MM. The solution given in [73] is also feasible and shows a supplied power of 0.774 W. However, it is computed with DE and therefore, a few thousand simulations are needed. SQP, with the coarse model optimum $\mathbf{z}^*$ as initial guess, finds a feasible point (associated supplied power of 0.700 W) after 38 fine and 1759 coarse transient simulations.

Subsequent optimization runs with MM detect different design solutions with analogous cost function and constraints values. The costs to compute them are also similar to the one indicated above. The presence of multiple (non-isolated) optima is a clear sign of ill-conditioned model responses and/or constraints functions. Within the required accuracy, the MM scheme seems to be able to converge without any trust-region or regularization strategy. These safeguards will be fundamental if a much more precise solution is needed.

6.3.3 A three-stage operational amplifier (ASCO3)

This optimization problem is denoted by ASCO3. Operational amplifiers can nowadays be found inside an immense number of electronic devices. The one studied in this section aims at a high gain with a low power consumption. It was introduced in [73] to be used in portable devices with transmitting-receiving purposes. This allows to reduce the size of the batteries required. The amplifier in Figure 6.11 is composed of three stages (multiple stages are introduced in order to increase the overall gain). The capacitors C_1 and C_2 compensate for the instability usually introduced with a multi-stage strategy.

The main difficulty in ASCO3 compared with the previous problems is the high number of design variables (twenty-one). In this design finding a feasible point is a very difficult task. Though each fine simulation has a cost of only a few seconds, an acceptable solution for the optimization problem was found in [73] after about fifteen hours. By manifold mapping (and with similar computing resources) we speed-up this process by a factor larger than ten and the solution obtained has a cost function value which is twenty-five per cent smaller than the one in [73]. This result shows the potential of multi-level optimization in situations where electronic devices might be composed of several operational amplifiers and then, each simulation takes considerably more than a few seconds.

The design concerns finding the most power efficient configuration for the operational amplifier showed in Figure 6.11. The power is computed not in the time but in the frequency domain (the supplied voltage is 3 V). The amplifier frequency response and *slew rate* (maximum rate of change in the output) are

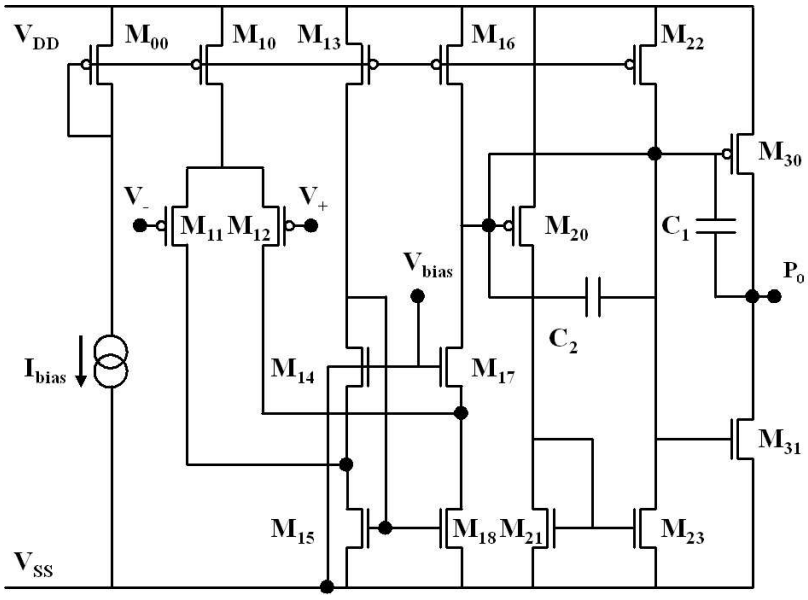


Figure 6.11: ASCO3: Three-stage operational amplifier. The first digit in every transistor name denotes the stage they belong to. The transistors M_{10}, M_{13}, M_{16}, M_{30} and M_{31} have a multiplicity of six, three, three, twenty-two and five, respectively. This means that, for example, M_{10} represents a set of six transistors. The capacitors C_1 and C_2 compensate for the instability usually introduced with a multi-stage strategy.

considered acceptable for practical use when some constraints are satisfied. These constraints are defined over $\mathbb{R}^5$ and they are of inequality type (viz., a lower bound for the gain at 0 Hz, a lower bound for the frequency at which the gain is 0 dB, upper and lower bounds for the phase at that frequency and a lower bound for the slew rate, all of them based on the voltage in node P_0). The problem has twenty-one design variables, i.e., $\mathbf{x} = [\mathbf{x}_1, \ldots, \mathbf{x}_{21}]$: seventeen internal transistor parameters (ten widths and seven lengths), one bias voltage (V_{bias}), one bias current (I_{bias}) and two compensation capacitors (C_1 and C_2). For additional problem details, see [73].

Both the fine and the coarse models and constraints functions require simulations in the frequency domain with HSPICE. In this type of analysis the circuit is first linearized and then a linear system is solved for each frequency. The amount of computational work needed is generally smaller than that for a transient simulation. The fine model response $\mathbf{f}(\mathbf{x})$ and constraints function $\mathbf{k_f}(\mathbf{x})$ require a frequency analysis from 0 to 1 GHz with 100 frequency points per decade. For

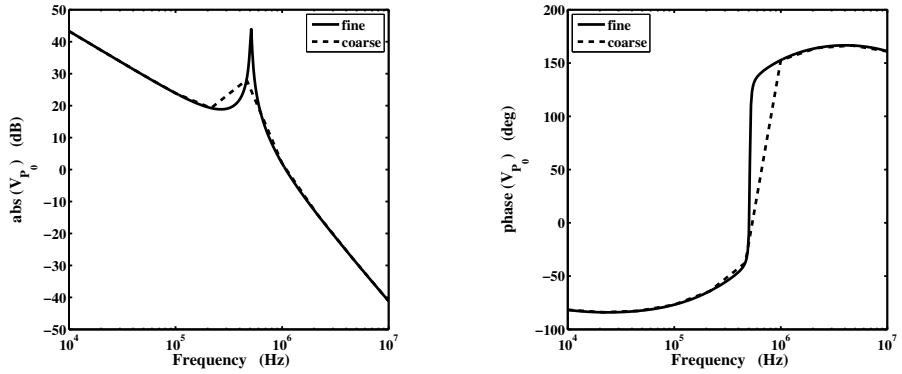


Figure 6.12: Similarity between fine and coarse model and constraints for ASCO3 with the design variable $\mathbf{x}$ being one of the vertexes of the design space $\widehat{X}$. The left figure shows the amplitude of the voltage in node P_0 for a certain range of frequencies. The right figure shows the phase of the voltage in node P_0 for the same range of frequencies.

the coarse model response $\mathbf{c}(\mathbf{z})$ and constraints function $\mathbf{k_c}(\mathbf{z})$ the same frequency range is swept but with only three points per decade. Though the discrepancy between the two simulations is small in a neighborhood of the optimum, in some other cases, as can be seen in Figure 6.12, it may be considerable. The computational complexity of the coarse simulation is a factor thirty lower than that of the fine one, but for the reasons commented above, the time difference observed is much smaller.

In [73] a feasible solution with an associated power consumption of 0.413 mW is found by means of differential evolution (DE) after around 19611 fine simulations. In the two-level approach the coarse model optimum $\mathbf{z}^*$ is obtained with a combination of DE and SQP, as in ASCO2. The manifold-mapping algorithm yields a solution with a cost function value of 0.300 mW after 2 fine and 3447 coarse simulations (SQP is used for solving every coarse model optimization). SQP requires some more computational work than MM but the cost function value is slightly reduced: 0.299 mW after 467 fine and 3239 coarse simulations. The fact that the solutions obtained by SQP and MM are considerably different and have almost identical cost function values indicates that the problem might be ill-conditioned.

6.4 Conclusions

In this chapter we have studied the two-level optimization approach applied to designs that were formulated as constrained minimization problems where the

constraints require time-consuming computations. EPE3 and EPE4 are examples from practical linear actuator designs. EPE3a has been used first to illustrate that in MM a trust region is necessary when an ill-conditioned problem is expected. EPE3b, a design involving seven design parameters, yields a minimal mass solution that satisfies the expensive nonlinear functions (two-dimensional finite-element computations) that express the constraints. This optimum is achieved by either SM or MM around eight times faster than by SQP. The EPE3b solution has been manufactured and properly tested by research partners [40]. EPE4 is an alternative to EPE3b that can be used in those circumstances where the space region for the actuator is given a priori and not suited for an axisymmetric device. As EPE3b, EPE4 is formulated by minimizing the mass of a device subject to constraints that require time-demanding three-dimensional finite-element computations. The solution for EPE4 is obtained efficiently by either SM or MM. ASCO1, ASCO2 and ASCO3 are design problems in electronics in which the models and the constraints are given by non-trivial functions. In all cases the model response is the power supplied and the constraints are evaluated by electronic circuit simulations. The strategy followed for the generation of the coarse models consists in performing simulations with significantly less sample points. ASCO1 deals with a digital inverter. ASCO1 has only one design parameter and it exemplifies the use of a two-level approach such as MM in electronics. ASCO2 and ASCO3 look for low power consuming amplifiers aimed at high frequency applications. MM delivers for both ASCO2 and ASCO3 a speed-up larger than a factor of ten with respect to the single-level optimization technique used in the work where these two problems were presented.

Summary

A number of design problems in practice aim at matching mathematical models with some given specifications. Quite often, the model does not completely meet the specifications and thus, the design consists in minimizing this discrepancy, which is precisely expressed by means of some cost function. Therefore, an optimization problem has to be solved. Though not all practical optimization problems fit in this model versus specifications structure, many of them can be eventually formulated in that way.

Nowadays it is common to find applications where these mathematical models are very expensive to compute, for example when they are based on finite elements. Optimization algorithms are typically of iterative nature and hence, the models are normally evaluated an elevated number of times. This fact implies that in these situations the optimum is found after a very long (and possibly prohibitive) computing time.

Frequently, these expensive mathematical models can be simplified, i.e., by taking coarser discretization grids or by selecting a less complicated physical representation of the phenomenon under study. If with the simplified models, we formulate a similar optimization problem as for the complex original model, the solution could be obtained with much less computational effort. However, because of the imprecision in the underlying models, this solution will not be, in general, an accurate one.

The quality of this coarse solution is improved with the use of data extracted from the accurate model. The imprecision in the simple model can be reduced with the information available after a few evaluations of the more complicated models. As a consequence, by proceeding in a two-model fashion we can obtain efficiently a precise solution. Within this multi-level approach, we will refer to the accurate and simplified models as fine and coarse, respectively.

Space mapping is a multi-level optimization technique that has been reported efficient in a large number of design cases. The coarse model is improved through a correction only in the design variables and not in the model output manifold. Though this correction has a very simple mathematical structure (it is linear), the way in which it is determined within the space-mapping framework is not straightforward at all. The solutions given by space mapping have normally an

accuracy that is acceptable for practical purposes and they can be computed with a few fine model evaluations. However, they do not generally coincide with the fine model optimum. The first part of this thesis is a complete survey of the space-mapping paradigm.

The idea of solving a complex problem by the iteration of simpler ones is old in mathematics (Newton's method for finding local minimizers is a significant example). This strategy is studied in detail by the theory of defect correction. Classical defect correction aims at accelerating the solution process of a complicated nonlinear operator equation aided by a collection of easy-to-evaluate approximate inverses of it. The classical defect correction concept is briefly explained in this work and then its applicability is extended to general optimization problems.

The similarity between space mapping and defect correction is clear. In this thesis, once both approaches are introduced, we interpret the former from the well-understood framework of the latter and with this machinery we are able to better understand why space mapping is not generally yielding the precise optimum. By defect correction we also see that, in order to obtain the accurate solution, the coarse model output manifold needs to be modified, and that the mapping for the design variables is optional. This model output correction is very simple (it is affine), unlike the one in space mapping, very easy to implement and it constitutes the cornerstone of the manifold-mapping technique.

The second part of this thesis deals with manifold mapping. The basic algorithms and also some very useful variants are presented. We prove that the stationary point for the basic schemes is the accurate solution and we also state some precise conditions for convergence.

The thesis is completed with an application of both space mapping and manifold mapping to design problems of practical relevance in the fields of magnetics and electronics. By comparison with several other efficient optimization schemes we show the potential of the multi-level approach as a means of speeding-up many designs that otherwise demand an intensive use of computational resources.

Samenvatting

Bij een aantal ontwerpproblemen uit de praktijk wordt ernaar gestreefd een wiskundig model aan een aantal gegeven specificaties te laten voldoen. Dikwijls echter laat het wiskundig model niet toe dat exact aan die specificaties voldaan wordt en dan bestaat het ontwerpproces eruit de discrepantie –die wiskundig in een kostenfunctie wordt uitgedrukt– te minimaliseren. Zo moet er dan een optimaliseringsprobleem worden opgelost. Hoewel niet alle optimaliseringsproblemen van deze model-versus-specificatie structuur zijn, kunnen er uiteindelijk toch veel op deze wijze geformuleerd worden.

Tegenwoordig zijn er veel toepassingen waarbij het doorrekenen van wiskundige modellen veel rekentijd vergt, bijvoorbeeld wanneer die berekening gebaseerd is op de eindige-elementen-methode. Optimalisatie-algorithmen zijn typisch iteratieve processen en daarom moeten daarbij de wiskundige modellen een aanzienlijk aantal keren doorgerekend worden. Dit impliceert dat in die situaties het optimum gevonden wordt na een (mogelijk onaanvaardbaar) lange rekentijd.

Dikwijls kunnen de dure wiskundige modellen vereenvoudigd worden, bijvoorbeeld door een grover rekenrooster te kiezen of door een eenvoudiger beschrijving van het onderliggende fysische proces te gebruiken. Als we met deze eenvoudiger beschrijving net zo'n optimaliseringsprobleem formuleren als we bij het ingewikkelder model deden, zou de oplossing met veel minder rekenwerk kunnen worden uitgevoerd. Door de onnauwkeurigheid van het gebruikte model zou de oplossing dan echter –in het algemeen– veel minder nauwkeurig zijn.

De kwaliteit van die 'grove' oplossing kan worden verbeterd door *ook* gebruik te maken van het nauwkeurige model. De onnauwkeurigheid van het simpele model kan worden gereduceerd door een paar keer het ingewikkelder model te gebruiken. Zo kunnen we, door de *twee* modellen te gebruiken, op een efficiënte manier een nauwkeurige oplossing bepalen. In zo'n multi-level aanpak noemen we het nauwkeurige en het vereenvoudigde model resp. het 'fijne' en het 'grove' model.

In de literatuur blijkt 'space-mapping' voor een groot aantal ontwerpproblemen een efficiënte optimalisatietechniek van het multi-level type te zijn. Het grove model wordt gecorrigeerd door een verbetering te construeren voor de ontwerpvariabelen, maar niet voor de model-output. Hoewel deze verbete-

ring een eenvoudige wiskundige structuur heeft (hij is bijna lineair), moet hij op een gecompliceerde manier bepaald worden. De oplossing die space-mapping levert, heeft in de praktijk weliswaar dikwijls een aanvaardbare nauwkeurigheid en kan met een klein aantal fijn-model evaluaties berekend worden, maar hij levert i.h.a. niet het optimum dat bij het fijne model hoort. In het eerste deel van dit proefschrift geven we een volledig overzicht van het space-mapping paradigma.

Een ingewikkeld probleem oplossen d.m.v. een iteratief proces waarin telkens een eenvoudiger probleem wordt opgelost, is een van oudsher bekend principe in de wiskunde. Newton's methode voor het oplossen van niet-lineaire vergelijkingen is een gerenommeerd voorbeeld. Deze algemene aanpak wordt bestudeerd in de theorie der defect-correctie processen. Klassieke defect-correctie streeft naar het versnellen van oplossingsprocessen voor ingewikkelde niet-lineaire operatorvergelijkingen met behulp van een serie eenvoudig-te-berekenen benaderende inversen. In dit proefschrift geven we een korte beschrijving van het defect-correctie principe en we breiden de toepasbaarheid uit tot algemene optimaliseringsproblemen.

De overeenkomst tussen space-mappng en defect-correctie is duidelijk. Nadat we in dit proefschrift beide aanpakken geïntroduceerd hebben, interpreteren we de eerste als een speciaal geval van de tweede, en omdat de laatste uitgebreid bestudeerd is, gebruiken we deze techniek om beter te begrijpen waarom space-mapping i.h.a. niet het juiste antwoord levert. D.m.v. de defect-correctie-aanpak zien we ook dat de variëteit die de output van het grove model beschrijft, aangepast moet worden en dat de aanpassing van de ontwerpvariabelen (d.m.v. de space-mapping functie) niet strikt noodzakelijk is. In tegenstelling tot de space-mapping functie is de aanpassing van de output-variëteit erg eenvoudig (affien) te construeren en zo wordt ze de hoeksteen van de manifold-mapping techniek. (manifold = variëteit.)

Het tweede deel van het proefschrift gaat over manifold-mapping. We presenteren de basis-algorithme en een aantal nuttige varianten. We geven een grondige analyse van de methode, waarin we o.a. bewijzen dat het stationaire punt van het iteratieve proces juist de exacte oplossing voor het nauwkeurige model levert. We formuleren ook preciese voorwaarden voor de convergentie van het proces.

Het proefschrift besluit met de toepassing van zowel space-mapping als manifold-mapping op een aantal ontwerpproblemen uit de praktijk. De toepassingen komen o.a. uit de magnetostatica en de electronica. Door de vergelijking te maken met verscheidene andere optimalisatietechnieken laten we zien waartoe de multilevel aanpak in staat is als een middel om ontwerpprocessen door te rekenen die anders een zeer reken-intensief zouden zijn.

Bibliography

[1] N. Alexandrov, J. E. Dennis, Jr., R. M. Lewis, and V. Torczon. A trust region framework for managing the use of approximation models in optimization. *Structural and Multidisciplinary Optimization*, 15(1):16–23, 1998.

[2] Ansoft Corp. ANSOFT Maxwell 3D User manual, 2001.

[3] M.H. Bakr, J.W. Bandler, R.M. Biernacki, S.H. Chen, and K. Madsen. A trust region aggressive space mapping algorithm for EM optimization. *IEEE Trans. on Microwave Theory and Techniques*, 46(12):2412–2425, 1998.

[4] M.H. Bakr, J.W. Bandler, N.K. Georgieva, and K. Madsen. A hybrid aggressive space mapping algorithm for EM optimization. *IEEE Trans. Microwave Theory Tech*, 47:2440–2449, 1999.

[5] M.H. Bakr, J.W. Bandler, M.A. Ismail, J.E. Rayas-Sánchez, and Q.J. Zhang. Neural space-mapping optimization for EM-based design. *IEEE Trans. on Microwave Theory and Techniques*, 48:2307–2315, 2000.

[6] M.H. Bakr, J.W. Bandler, K. Madsen, and J. Søndergaard. Review of the space mapping approach to engineering optimization and modeling. *Optimization and Engineering*, 1(3):241–276, 2000.

[7] J. W. Bandler, Q. S. Cheng, S. A. Dakroury, A. S. Mohamed, M. H. Bakr, K. Madsen, and J. Søndergaard. Space mapping: The state of the art. *IEEE Trans. on Microwave Theory and Techniques*, 52(1):337–361, 2004.

[8] J.W. Bandler, R.M. Biernacki, S.H. Chen, P.A. Grobelny, and R.H. Hemmers. Space mapping technique for electromagnetic optimization. *IEEE Trans. Microwave Theory Tech*, 42(12):2536–2544, 1994.

[9] J.W. Bandler, R.M. Biernacki, S.H. Chen, R.H. Hemmers, and K. Madsen. Electromagnetic optimization exploiting aggressive space mapping. *IEEE Trans. on Microwave Theory and Techniques*, 43:2874–2882, 1995.

[10] J.W. Bandler, R.M. Biernacki, S.H. Chen, and Y.F. Huang. Design optimization of interdigital filters using aggressive space mapping and decomposition. *IEEE Trans. Microwave Theory Tech*, 45:761–769, 1997.

[11] J.W. Bandler, R.M. Biernacki, S.H. Chen, and D. Omeragic. Space mapping optimization of waveguide filters using finite element and mode-matching electromagnetic simulators. *Int. J. RF Microwave Computer-Aided Eng.*, 9:54–70, 1999.

[12] J.W. Bandler, Q.S. Cheng, D.H. Gebre-Mariam, K. Madsen, F. Pedersen, and J. Søndergaard. EM-based surrogate modeling and design exploiting implicit, frequency and output space mappings. *IEEE MTT-S International Microwave Symposium Digest*, pages 1003–1006, 2003.

[13] J.W. Bandler, Q.S. Cheng, D.M. Hailu, and N.K. Nikolova. A space-mapping design framework. *IEEE Trans. Microwave Theory Tech.*, 52(11):2601–2610, 2004.

[14] J.W. Bandler, Q.S. Cheng, N.K. Nikolova, and M.A. Ismail. Implicit space mapping optimization exploiting preassigned parameters. *IEEE Trans. on Microwave Theory and Techniques*, 52:378–385, 2004.

[15] J.W. Bandler, D.M. Hailu, K. Madsen, and F. Pedersen. A space-mapping interpolating surrogate algorithm for highly optimized EM-based design of microwave devices. *IEEE Trans. on Microwave Theory and Techniques*, 52(11):2593–2600, 2004.

[16] J.W. Bandler, M.A. Ismail, J.E. Rayas-Sanchez, and Q.J. Zhang. Neuromodeling of microwave circuits exploiting space mapping technology. *IEEE Trans. Microwave Theory Tech.*, 47:2417 – 2427, 1999.

[17] J.W. Bandler, A.S. Mohamed, M.H. Bakr, K. Madsen, and J. Søndergaard. EM-based optimization exploiting partial space-mapping and exact sensitivities. *IEEE Trans. Microwave Theory Tech*, 50:2741–2750, 2002.

[18] J.-F.M. Barthelemy and R.T. Haftka. Approximation concepts for optimum structural design - a review. *Structural Optimization*, 5:129–144, 1993.

[19] M. Benzi, E. Haber, and L. Hanson. Multilevel algorithms for large-scale interior point methods in bound constrained optimization. Technical Report TR-2005-002-A, Dept of Mathematics and Computer Science, Emory University, Atlanta, GA, US, November 2006.

[20] S. Bila, D. Baillargeat, S. Verdeyme, and P. Guillon. Automated design of microwave devices using full EM optimization method. *IEEE MTT-S Int. Microwave Symp. Dig.*, pages 1771–1774, 1998.

[21] K. Böhmer, P.W. Hemker, and H.J. Stetter. The defect correction approach. In K. Böhmer and H.J. Stetter, editors, *Defect Correction Methods: Theory and Applications*, volume 5 of *Computing Suppl.*, pages 1–32. Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1984.

[22] K. Böhmer and H.J. Stetter. *Defect Correction Methods: Theory and Applications.* Springer, Berlin, 1984.

[23] I. Boldea. Linear electromagnetic actuators and their control: a review. *EPE Journal*, 14(1):43–50, 2004.

[24] A.J. Booker, J.E. Dennis, Jr., P.D. Frank, D.B. Serafini, V. Torczon, and M.W. Trosset. A rigorous framework for optimization of expensive functions by surrogates. *Structural and Multidisciplinary Optimization*, 17(1):1–13, 1999.

[25] C. De Boor and A. Ron. Computational aspects of polynomial interpolation in several variables. *Mathematics of Computation*, 58:705–727, 1992.

[26] V. Braibant and C. Fleury. An approximation-concepts approach to shape-optimal design. *Computer Methods in Applied Mechanics and Engineering*, 53:119–148, 1985.

[27] C.G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Math.Comp.*, 19:577–593, 1965.

[28] M.D. Buhmann and M.J. Ablowitz. *Radial Basis Functions : Theory and Implementations.* Cambridge University, 2003.

[29] D.K. Cheng. *Field and Wave Electromagnetics.* Addison-Wesley, 1989.

[30] H. Choi, D. Kim, I. Park, and S. Hahn. A new design technique of magnetic systems using space mapping algorithm. *IEEE Transactions on Magnetics*, 37(5):3627–3630, 2001.

[31] COMSOL. FEMLAB 3: User's Guide, 2004.

[32] A.R. Conn, N. Gould, and Ph.L. Toint. *Trust-Region Methods.* MPS/SIAM Series on Optimization. SIAM, Philadelphia, August 2000.

[33] A. Davis. Gnucap: The Gnu Circuit Analysis Package Users manual, 2002.

[34] J.E. Dennis, Jr., and V. Torczon. Managing approximation models in optimization. In N. M. Alexandrov and M. Y. Hussaini, editors, *Multidisciplinary Design Optimization: State of the Art*, pages 330–347. SIAM, 1997.

[35] D. Echeverría and P.W. Hemker. Space mapping and defect correction. *Comp. Methods in Appl. Math.*, 5(2):107–136, 2005.

[36] D. Echeverría, D. Lahaye, L. Encica, P.W. Hemker E.A. Lomonova, and A.J.A. Vandenput. Manifold-mapping optimization applied to linear actuator design. *IEEE Transactions on Magnetics*, 42(4):1183–1186, 2006.

[37] D. Echeverría, D. Lahaye, L. Encica, E.A. Lomonova, P.W. Hemker, and A.J.A. Vandenput. Manifold-mapping optimization applied to linear actuator design. Technical Report MAS-E0606, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, The Netherlands, January 2006.

[38] D. Echeverría and C.H. Tong. Towards multi-level optimization. Technical Report UCRL-TR-223288, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA, US, August 2006.

[39] L. Encica, D. Echeverría, E.A. Lomonova, A. J. A. Vandenput, P. W. Hemker, and D. Lahaye. Efficient Optimal Design of Electromagnetic Actuators Using Space Mapping. *Structural and Multidisciplinary Optimization*, 2006. Accepted for publication.

[40] L. Encica, J. Makarovic, E.A. Lomonova, and A.J.A. Vandenput. Space Mapping Optimization of a Cylindrical Voice Coil Actuator. In *5th IEEE International Electric Machines and Drives Conference*, pages 1–7, May 2005.

[41] L. Encica, J.J.H. Paulides, D. Echeverría, E.A. Lomonova, and A.J.A. Vandenput. A framework for efficient automated optimal design of electromagnetic actuators. In *12th Biennial IEEE Conference on Electromagnetic Field Computation (CEFC 2006)*, 2006.

[42] D.E. Finkel. Direct optimization algorithm user guide. Technical Report CRSC-TR03-11, Center for Research in Scientific Computation, North Carolina State University, 2003.

[43] A.A. Giunta, J.P. Castro, P.D. Hough, G.A. Gray, and M.S. Eldred. Multifidelity modeling approaches in simulation-based optimization. In *Eighth SIAM Conference on Optimization*, May 2005.

[44] A.A. Giunta and M.S. Eldred. Implementation of a Trust Region Model Management Strategy in the DAKOTA Optimization Toolkit. In *Proceedings of the 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA-2000-4935, September 2000.

[45] G.H. Golub and C.F. van Loan. *Matrix Computations*. North Oxford Academic, Oxford, England, 1983.

[46] S. Gratton, A. Sartenaer, and Ph.L. Toint. Recursive Trust-Region Methods for Multilevel Nonlinear Optimization (Part I): Global Convergence and Complexity. Technical Report 04/06, Dept of Mathematics, University of Namur, Namur, Belgium, 2004.

[47] S. Gratton, A. Sartenaer, and Ph.L. Toint. Numerical Experience with a Recursive Trust-Region Method for Multilevel Nonlinear Optimization. Technical Report 06/01, Dept of Mathematics, University of Namur, Namur, Belgium, 2006.

[48] A. Griewank. *Evaluating Derivatives. Principles and Techniques of Algorithmic Differentiation*, volume 19 of *SIAM Frontiers in Applied Mathematics*. SIAM, Philadelphia, 2000.

[49] W. Hackbush. *Multigrid methods and Applications*, volume 4 of *Series in Computational Mathematics*. Springer-Verlag, Berlin, 1985.

[50] C. Ignatovich and A. Diaz. Physical surrogates in design optimization for enhanced crashworthiness. In *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, number AIAA-2002-5537, September 2002.

[51] M.A. Ismail, D. Smith, A. Albert, Y. Yang, and M. Yu. EM based design of large-scale dielectric resonator multiplexers by space-mapping. In *IEEE MTT-S Int. Microwave Symp. Dig.*, pages 291–294, 2003.

[52] D.R. Jones, C.D. Perttunen, and B.E. Stuckman. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Application*, 79(1):157–181, 1993.

[53] A.G. Journel and Ch.J. Huijbregts. *Mining Geostatistics*. Academic Press, 1981.

[54] T.G. Kolda, R.M. Lewis, and V. Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Review*, 45(3):385–482, March 2003.

[55] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the Nelder-Mead simplex method in low dimensions. *SIAM Journal of Optimization*, 9(1):112–147, 1998.

[56] S. J. Leary, A. Bhaskar, and A. J. Keane. A constraint mapping approach to the structural optimization of an expensive model using surrogates. *Optimization and Engineering*, 2(4):385–398, 2001.

[57] R.M. Lewis and S. Nash. A multigrid approach to the optimization of systems governed by differential equations. In *8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, number AIAA-2000-4890, 2000.

[58] K. Madsen. *Minimization of Non-Linear Approximation Functions*. PhD thesis, Institute for Numerical Analysis, Technical University of Denmark, Lyngby, Denmark, 1986.

[59] K. Madsen and J. Søndergaard. Convergence of hybrid space mapping algorithms. *Optimization and Engineering*, 5(2):145–156, jun 2004.

[60] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11:431–441, 1963.

[61] META Software. HSPICE user's manual, 1996.

[62] J.J. More. Multilevel algorithms in optimization. In *Eighth SIAM Conference on Optimization*, May 2005.

[63] S. Nash. A multigrid approach to discretized optimization problems. *Journal of Optimization Methods and Software*, 14:99–116, 2000.

[64] A. Neumaier. Complete search in continuous global optimization and constraint satisfaction. *Acta Numerica 2004*, pages 271–369, 2004.

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

[66] A.B. Owen. Orthogonal arrays for computer experiments, integration and visualization. *Statistica Sinica*, 2:439–452, 1992.

[67] A.M. Pavio. The electromagnetic optimization of microwave circuits using companion models. In *IEEE MTT-S Int. Microwave Symp. Workshop*, 1999.

[68] F. Pedersen, P. Weitzmann, and S. Svendsen. Modeling thermally active building components using space mapping. In *7th Symposium on Building Physics in the Nordic Countries*, pages 896–903, 2005.

[69] Philips ED&T / Analogue Simulation. Pstar User Manual for Pstar 4.5, 2003.

[70] G.C. Pomraning. *Linear Kinetic Theory and Particle Transport in Stochastic Mixtures.* Series on Advances in Mathematics for Applied Sciences. World Scientific, 1992.

[71] M.J.D. Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. Technical Report DAMTP/NA5, University of Cambridge, Cambridge, England, 1992.

[72] M.J.D. Powell. UOBYQA: Unconstrained optimization by quadratic approximation. *Mathematical Programming*, 92(3):555–582, May 2002.

[73] J. Ramos. *CMOS Operational and RF Power Amplifiers for Mobile Communications.* PhD thesis, K.U. Leuven, Belgium, March 2005.

[74] M. Redhe and L. Nilsson. Using space mapping and surrogates models to optimize vehicle crashworthiness design. In *Proceedings of the 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA-2002-5536, 2002.

[75] S. Safavi-Naeini, S.K. Chaudhuri, N. Damavandi, and A. Borji. A multilevel design optimization strategy for complex RF/microwave structures. In *IEEE MTT-S Int. Microwave Symp. Workshop*, 2000.

[76] R.R. Saldanha. *Optimisation en Electromagnetisme par Application Conjointe des Methodes de Programmation Non Lineaire et de la Methode des Elements Finis.* PhD thesis, Institut National Polytechnique de Grenoble, Grenoble, France, 1992.

[77] R.R. Saldanha, S. Pelissier, K. Kadded, Y.P. Yonnet, and J.L. Coulomb. Nonlinear optimization methods applied to magnetic actuators design. *IEEE Transactions on Magnetics*, 28(2):1581–1584, March 1992.

[78] L.A. Schmit, Jr., and B. Farshi. Some approximation concepts for structural synthesis. *AIAA Journal*, 12:692–699, 1974.

[79] P.P. Silvester and R.L. Ferrari. *Finite Element for Electrical Engineers.* Cambridge University Press, New York, 1996.

[80] J. Snel. Space mapping models for RF components. *IEEE MTT-S Int. Microwave Symp. Workshop*, 2001.

[81] N.O. Sokal and A.D. Sokal. Class E: A New Class of High-Efficiency Tuned Single-Ended Switching Power Amplifiers. *Journal of Solid-State Circuits*, 10(3):168–176, June 1975.

[82] J. Søndergaard. *Optimization Using Surrogate Models - by the Space Mapping Technique.* PhD thesis, IMM, DTU, DK-2800 Lyngby, 2003.

[83] M. Stein. Large sample properties of simulating using Latin hypercube sampling. *Technometrics*, 29(2):143–151, 1987.

[84] H.J. Stetter. The defect correction principle and discretization methods. *Numer. Math*, 29:425–443, 1978.

[85] R. Storn and K. Price. Differential Evolution – A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces. Technical Report TR-95-012, International Computer Science Institute (ICSI), Berkeley, California, US, March 1995.

[86] D.G. Swanson, Jr., and R.J. Wenzel. Fast analysis and optimization of combline filters using FEM. In *IEEE MTT-S Int. Microwave Symp. Dig.*, pages 1159–1162, 2001.

[87] University of California, Berkeley, Department of Electrical Engineering and Computer Science. SPICE version 2G6 source code, 1983.

[88] H.A. van der Vorst. *Iterative Krylov Methods for Large Linear Systems*, volume 13 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, 2003.

[89] F. Vanden Berghen and H. Bersini. CONDOR, a new parallel, constrained extension of Powell's UOBYQA algorithm: experimental results and comparison with the DFO algorithm. *Journal of Computational and Applied Mathematics*, 181(1):157–175, September 2005.

[90] F.W. Warner. *Foundations of Differentiable Manifolds and Lie Groups*. Graduate Texts in Mathematics. Springer-Verlag, New York-Berlin-Heidelberg-Tokyo, 1983.

List of Acronyms

ASM	:	Aggressive Space Mapping
BDF	:	Backward Differentiation Formula
DE	:	Differential Evolution
EM	:	Ellipsoid Method
FE	:	Finite Element(s)
GMM	:	Generalized Manifold Mapping
GPE	:	Gradient Parameter Extraction
GSVD	:	Generalized Singular Value Decomposition
HASM	:	Hybrid Aggressive Space Mapping
ISM	:	Implicit Space Mapping
KKT	:	Karush-Kuhn-Tucker (conditions)
MEC	:	Magnetic Equivalent Circuit
MM	:	Manifold Mapping
MMA	:	Method of Moving Asymptotes
MPE	:	Multipoint Parameter Extraction
NMS	:	Nelder-Mead Simplex
NSM	:	Neural Space Mapping
OMM	:	Original Manifold Mapping
OPM	:	Output Polynomial Mapping
OSM	:	Output Space Mapping
PM	:	Penalty Method
PPE	:	Penalized Parameter Extraction
PSM	:	Partial Space Mapping
RQP	:	Recursive Quadratic Programming
SM	:	Space Mapping
SMIS	:	Space-Mapping Interpolating Surrogate
SPE	:	Statistical Parameter Extraction
SQP	:	Sequential Quadratic Programming
SVD	:	Singular Value Decomposition
TRASM	:	Trust-Region Aggressive Space Mapping
TRMM	:	Trust-Region Manifold Mapping

Index