View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by CWI's Institutional Repository

Space Mapping and Defect Correction

David Echeverría, Domenico Lahaye, and Piet W. Hemker

CWI, Kruislaan 413, 1054 AL Amsterdam, The Netherlands**
{D.Echeverria,P.W.Hemker,Domenico.Lahaye}@cwi.nl

Summary. In this chapter we present the principles of the space-mapping iteration techniques for the efficient solution of optimization problems. We also show how space-mapping optimization can be understood in the framework of defect correction.

We observe the difference between the solution of the optimization problem and the computed space-mapping solutions. We repair this discrepancy by exploiting the correspondence with defect correction iteration and we construct the manifold-mapping algorithm, which is as efficient as the space-mapping algorithm but converges to the true solution.

In the last section we show a simple example from practice, comparing space-mapping and manifold mapping and illustrating the efficiency of the technique.

1 Introduction

Space mapping is a technique, using simple surrogate models, to reduce the computing time in optimization procedures where time-consuming computer-models are needed to obtain sufficiently accurate results. Thus, space mapping makes use of both accurate (and time-consuming) models and less accurate (but cheaper) ones.

In fact, the original space-mapping procedure corresponds with right-preconditioning the coarse (inaccurate) model in order to accelerate the iterative procedure for the optimization of the fine (accurate) one. The iterative procedure used in space mapping for optimization can be seen as a defect correction iteration and the convergence can be analyzed accordingly. In this paper we show the structure of space mapping iteration. We also show that right-preconditioning is generally insufficient and (also) left-preconditioning is needed to obtain the solution for the accurate model. This leads to the improved space-mapping or 'manifold-mapping' procedure. This manifold mapping is shown in some detail in Section 5 and in the last section a few examples of an application are given.

The space-mapping idea was introduced by Bandler [3] in the context of microwave filter design and it has developed significantly over the last decade. In the

** This research was supported by the Dutch Ministry of Economic Affairs through the project IOP-EMVT-02201 B

rather complete survey [4] we see that the original idea has gone through a large number of changes and improvements. The reader is referred to the original literature [1, 2, 5] for a review of earlier achievements and for a classical introduction for engineers.

2 Fine and Coarse Models in Optimization

The Optimization Problem.

Let the specifications for the data of an optimization problem be denoted by $(\mathbf{t}, \mathbf{y}) \equiv (\{t_i\}, \{y_i\})_{i=1,\ldots,m}$. The independent variable $\mathbf{t} \in \mathbb{R}^m$ could be, e.g., time, frequency, space, etc. The dependent variable $\mathbf{y} \in Y \subset \mathbb{R}^m$ represents the quantities that describe the behavior of the phenomena under study or design. The set $Y \subset \mathbb{R}^m$ is called the *set of possible aims*.

The behavior of the variable $\mathbf{y}$ not only depends on the independent variable $\mathbf{t}$ but also on an additional set of control/design variables. With $\mathbf{x}$ the vector of relevant control variables, we may write the components of $\mathbf{y}$ as $y_i \approx y(t_i, \mathbf{x})$. The behavior of the phenomenon is described by the function $y(t, \mathbf{x})$ and the difference between the measured data y_i and the values $y(t_i, \mathbf{x})$ may be the result of, e.g., measurement errors or the imperfection of the mathematical description.

Models to describe reality appear in several degrees of sophistication. Space mapping exploits the combination of the simplicity of the less sophisticated methods with the accuracy of the more complex ones. Therefore, we distinguish two types of model: fine and coarse.

The Fine Model.

The *fine model* response is denoted by $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^m$, where $\mathbf{x} \in X \subset \mathbb{R}^n$ is the *fine model control variable*. The set X of possible control variables is usually a closed and bounded subset of $\mathbb{R}^n$. The set $\mathbf{f}(X) \subset \mathbb{R}^m$ of all possible fine model responses is the set of *fine model reachable aims*. The fine model is assumed to be *accurate* but *expensive* to evaluate. We also assume that $\mathbf{f}(\mathbf{x})$ is continuous.

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

$$\mathbf{x}^* = \operatorname*{argmin}_{\mathbf{x} \in X} \| \mathbf{f}(\mathbf{x}) - \mathbf{y} \| \ . \tag{1}$$

A design problem, characterized by the model $\mathbf{f}(\mathbf{x})$, the aim $\mathbf{y} \in Y$, and the space of possible controls $X \subset \mathbb{R}^n$, is called a *reachable design* if the equality $\mathbf{f}(\mathbf{x}^*) = \mathbf{y}$ can be achieved for some $\mathbf{x}^* \in X$. □

The Coarse Model.

The *coarse model* is denoted by $\mathbf{c}(\mathbf{z}) \in \mathbb{R}^m$, with $\mathbf{z} \in Z \subset \mathbb{R}^n$ the *coarse model control variable*. This model is assumed to be *cheap* to evaluate but *less accurate* than the fine model. 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*, $\| \mathbf{c}(\mathbf{z}) - \mathbf{y} \|$. We denote its minimizer by $\mathbf{z}^*$,

$$\mathbf{z}^* = \operatorname*{argmin}_{\mathbf{z} \in Z} \| \mathbf{c}(\mathbf{z}) - \mathbf{y} \| \ . \tag{2}$$

We assume that the fine and coarse optimization problems, characterized by $\mathbf{y}$, $\mathbf{f}(\mathbf{x})$ and X, respectively $\mathbf{y}$, $\mathbf{c}(\mathbf{z})$ and Z, are uniquely solvable and well defined. If X and Z are closed and bounded non-empty sets in $\mathbb{R}^n$ and $\mathbf{f}$ and $\mathbf{c}$ continuous functions, the existence of the solutions is guaranteed. Generally, uniqueness can be achieved by properly reducing the sets X or Z. If the models are non-injective (or extremely ill-conditioned) in a small neighborhood of a solution, essential difficulties may arise.

The Space-Mapping Function.

The similarity or discrepancy between the responses of two models used for the same phenomenon is an important property. It is expressed by the *misalignment function*

$$r(\mathbf{z}, \mathbf{x}) = \| \mathbf{c}(\mathbf{z}) - \mathbf{f}(\mathbf{x}) \| \ . \tag{3}$$

For a given $\mathbf{x} \in X$ it is useful to know which $\mathbf{z} \in Z$ yields the smallest discrepancy. This information can be used to improve the coarse model. Therefore, the space-mapping function is introduced. The *space-mapping function* $\mathbf{p} : X \subset \mathbb{R}^n \to Z \subset \mathbb{R}^n$ is defined[1] by

$$\mathbf{p}(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{z} \in Z} r(\mathbf{z}, \mathbf{x}) = \operatorname*{argmin}_{\mathbf{z} \in Z} \| \mathbf{c}(\mathbf{z}) - \mathbf{f}(\mathbf{x}) \| \ . \tag{4}$$

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. In Figure 1 we see an example of a misalignment function and of a few space mapping functions.

Perfect Mapping.

In order to identify the cases where the accurate solution $\mathbf{x}^*$ is related with the less accurate solution $\mathbf{z}^*$ by the space mapping function, the following definition is introduced. A space-mapping function $\mathbf{p}$ is called a *perfect mapping* iff

[1] 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 corresponds with a given fine-model control $\mathbf{x}$.

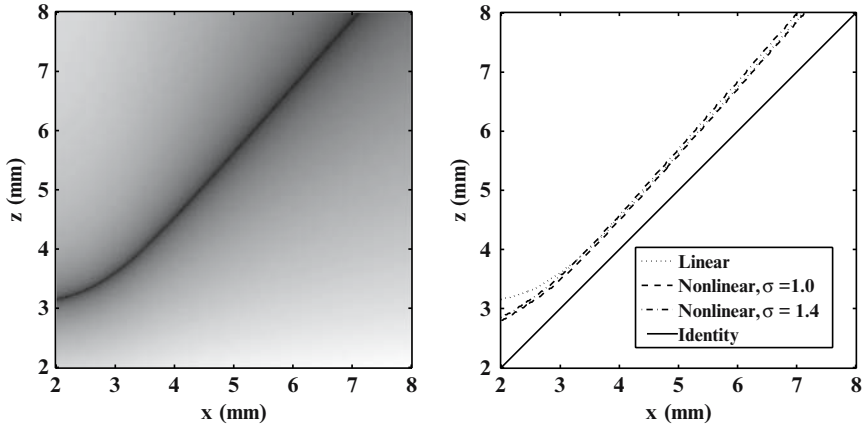


Fig. 1. Misalignment and space-mapping function

The left figure shows the misalignment function for a fine and a coarse model. Darker shading shows a smaller misalignment. The right figure shows the identity function and a few space-mapping functions for different coarse models (example taken from [9]).

$$\mathbf{z}^* = \mathbf{p}(\mathbf{x}^*) \ . \tag{5}$$

Using the definition of space mapping we see that (5) can be written as

$$\underset{\mathbf{z}\in Z}{\operatorname{argmin}} \, \| \mathbf{c}(\mathbf{z}) - \mathbf{y} \| = \underset{\mathbf{z}\in Z}{\operatorname{argmin}} \, \| \mathbf{c}(\mathbf{z}) - \mathbf{f}(\mathbf{x}^*) \| \ , \tag{6}$$

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.

Remark. We notice that *perfection* is not only a property of the space-mapping function, 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. In this sense 'perfect mapping' can be a confusing notion.

3 Space-Mapping Optimization

In literature many space mapping based algorithms can be found [1, 4], but they all have the same basis. We first describe the original space-mapping idea and the resulting two principal approaches (primal and dual).

3.1 Primal and Dual Space-Mapping Solutions

The 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

$$\mathbf{p}(\mathbf{x}^*) = \underset{\mathbf{z}\in Z}{\operatorname{argmin}} \left\| \mathbf{c}(\mathbf{z}) - \mathbf{f}(\mathbf{x}^*) \right\| \approx \underset{\mathbf{z}\in Z}{\operatorname{argmin}} \left\| \mathbf{c}(\mathbf{z}) - \mathbf{y} \right\| = \mathbf{z}^* . \tag{7}$$

Based on this relation, the space-mapping approach assumes $\mathbf{p}(\mathbf{x}^*) \approx \mathbf{z}^*$. However, in general $\mathbf{p}(\mathbf{x}^*) \neq \mathbf{z}^*$ and even $\mathbf{z}^* \in \mathbf{p}(X)$ is not guaranteed. Therefore the *primal* space-mapping approach seeks for a solution of the minimization problem

$$\mathbf{x}_p^* = \underset{\mathbf{x}\in X}{\operatorname{argmin}} \left\| \mathbf{p}(\mathbf{x}) - \mathbf{z}^* \right\| . \tag{8}$$

An alternative approach can be chosen. The idea behind space-mapping optimization is the replacement of the expensive fine model optimization by a surrogate model. For the surrogate model we can take the coarse model $\mathbf{c}(\mathbf{z})$, and improve its accuracy by the space mapping function $\mathbf{p}$. Now the improved or *mapped coarse model* $\mathbf{c}(\mathbf{p}(\mathbf{x}))$ may serve as the better *surrogate model*. Because of (4) we expect $\mathbf{c}(\mathbf{p}(\mathbf{x})) \approx \mathbf{f}(\mathbf{x})$ and hence $\left\| \mathbf{f}(\mathbf{x}) - \mathbf{y} \right\| \approx \left\| \mathbf{c}(\mathbf{p}(\mathbf{x})) - \mathbf{y} \right\|$. Then the minimization of $\left\| \mathbf{c}(\mathbf{p}(\mathbf{x})) - \mathbf{y} \right\|$ will usually give us a value, $\mathbf{x}_d^*$, close to the desired optimum $\mathbf{x}^*$:

$$\mathbf{x}_d^* = \underset{\mathbf{x}\in X}{\operatorname{argmin}} \left\| \mathbf{c}(\mathbf{p}(\mathbf{x})) - \mathbf{y} \right\| . \tag{9}$$

This is the *dual* space-mapping approach.

We will see in Section 3.3 that both approaches coincide when $\mathbf{z}^* \in \mathbf{p}(X)$ and $\mathbf{p}$ is injective, and if the mapping is perfect both $\mathbf{x}_p^*$ and $\mathbf{x}_d^*$ are equal to $\mathbf{x}^*$. However, 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. The principle of the approach is summarized in Figure 2.

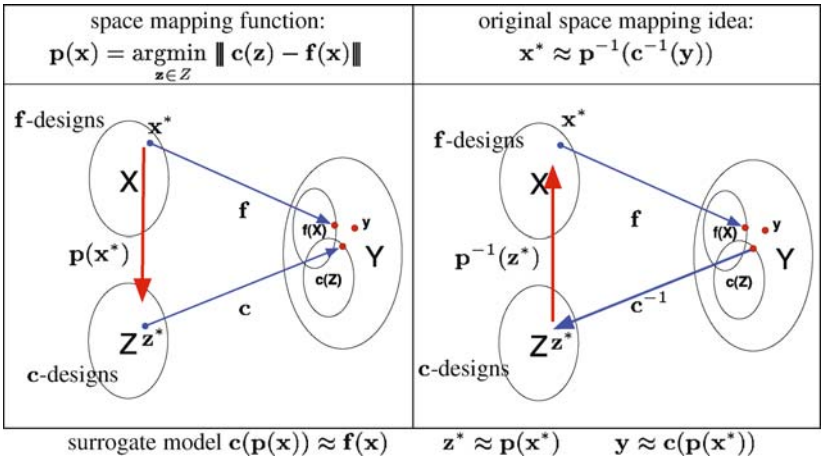


Fig. 2. Diagram showing the main idea of space mapping

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

Fig. 3. The ASM algorithm

3.2 Space-Mapping Algorithms

Because the evaluation of the space-mapping function is expensive, algorithms to compute $\mathbf{x}_p^*$ or $\mathbf{x}_d^*$ are based on iterative approximation of $\mathbf{p}(\mathbf{x})$. By the similarity of $\mathbf{f}(\mathbf{x})$ and $\mathbf{c}(\mathbf{z})$, a first approximation is the identity, $\mathbf{p}_0 = I$.

Linear approximations form the basis for the more popular space-mapping optimization algorithms. An extensive survey of available algorithms can be found in [4]. The most representative example is ASM (the 'Aggressive Space Mapping' shown in Figure 3), where the space-mapping function is approximated by linearisation to obtain

$$\mathbf{p}_k(\mathbf{x}) = \mathbf{p}(\mathbf{x}_k) + \mathbf{B}_k\left(\mathbf{x} - \mathbf{x}_k\right). \tag{10}$$

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

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{\mathbf{p}(\mathbf{x}_{k+1}) - \mathbf{p}(\mathbf{x}_k) - \mathbf{B}_k\mathbf{h}}{\mathbf{h}^T\mathbf{h}}\,\mathbf{h}^T\,, \tag{11}$$

where $\mathbf{h} = \mathbf{x}_{k+1} - \mathbf{x}_k$. This is combined with original space mapping, so that $\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{B}_k^{-1}(\mathbf{p}(\mathbf{x}_k) - \mathbf{z}^*)$.

3.3 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*, 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. *(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 following lemma [9] follows from the definitions (8) and (9).

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

In some cases we can expect that the sets of fine and coarse reachable aims overlap in a region of $\mathbb{R}^m$ close to their respective optima. The concept of model flexibility is introduced and from that some results concerning properties of the space-mapping functions can be derived.

Definition 1. *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 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. 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. Proofs are found in [9].

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* $\Leftrightarrow \mathbf{c}(\mathbf{z}^*) = \mathbf{f}(\mathbf{x}^*)$*;*
(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.*

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. However, we remind 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.*

We combine the previous two lemmas in the following.

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

The conclusions in Lemma 2 can now be derived from assumptions about model flexibility.

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

Remark. It is not really needed 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 former lemmas can be relaxed and stated just locally.

4 Defect Correction and Space Mapping

The technique underlying space-mapping, i.e. the efficient solution of a complex problem by the iterative use of a simpler one, is known since long in computational mathematics. In numerical analysis it is known as defect correction iteration and studied in a number of papers [6, 7]. Below we first briefly summarize the defect correction principle for solving operator equations and then we apply the idea to optimization problems.

4.1 Defect Correction for Operator Equations

We first consider the problem of solving a nonlinear operator equation

$$\mathcal{F}\,\mathbf{x} = \mathbf{y}, \tag{12}$$

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 is 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 equation (12) with $\mathbf{y} \in \mathcal{F}(D) \subset \widehat{D}$ 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 $\widetilde{\mathcal{F}}_k$, that is needed for defect correction and we do not need to assume $\widetilde{\mathcal{G}}_k$ to be invertible.

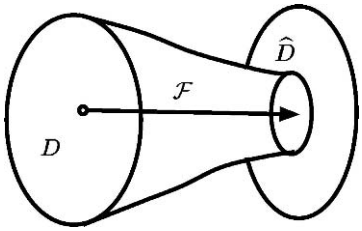
Defect correction comes in two brands [6], depending on the space, E or $\widehat{E}$, in which linear combinations for extrapolation are made. The two basic iterative defect correction procedures to generate a (hopefully convergent) sequence of approximations to the solution of (12) are

$$\begin{cases} \mathbf{x}_0 &= \widetilde{\mathcal{G}}_0\,\mathbf{y} \\ \mathbf{x}_{k+1} &= (I - \widetilde{\mathcal{G}}_{k+1}\,\mathcal{F})\,\mathbf{x}_k + \widetilde{\mathcal{G}}_{k+1}\,\mathbf{y} \end{cases} \quad \text{and} \quad \begin{cases} \mathbf{l}_0 &= \mathbf{y} \\ \mathbf{l}_{k+1} &= (I - \mathcal{F}\,\widetilde{\mathcal{G}}_k)\,\mathbf{l}_k + \mathbf{y}\,. \end{cases} \tag{13}$$

In the second, (13b), 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 in the first, (13a), the extrapolation is in the space D, whereas the additions in (13b) are in $\widehat{D}$. 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 (13b) is equivalent with the iterative procedure

$$\begin{cases} \widetilde{\mathcal{F}}_0 \mathbf{x}_0 &= \mathbf{y}\,, \\ \widetilde{\mathcal{F}}_{k+1} \mathbf{x}_{k+1} &= \widetilde{\mathcal{F}}_k\,\mathbf{x}_k - \mathcal{F}\widetilde{\mathcal{G}}_k \widetilde{\mathcal{F}}_k\,\mathbf{x}_k + \mathbf{y}\,. \end{cases} \tag{14}$$

In order to apply (14), the injectivity of $\widetilde{\mathcal{G}}_k$ is not really needed and it is immediately seen that neither (13b) nor (14) converges if $\mathbf{y} \notin \mathcal{F}(D)$. However, (14) can be modified so that it can be used for $\mathbf{y} \notin \mathcal{F}(D)$. Then we need injectivity for $\widetilde{\mathcal{F}}_k$ and we take $\widetilde{\mathcal{G}}_k$ its left-inverse, i.e., $\widetilde{\mathcal{G}}_k \widetilde{\mathcal{F}}_k = \mathcal{I}_D$. Then (14) leads to



$\mathcal{F}$ not surjective $\rightarrow$ left-inverse $\mathcal{G}$: $\mathcal{G}\mathcal{F} = I_D$

Fig. 4. The non-surjective operator in the optimization problem

$$\begin{cases} \mathbf{x}_0 &= \widetilde{\mathcal{G}}_0\, \mathbf{y}\,, \\ \mathbf{x}_{k+1} &= \widetilde{\mathcal{G}}_{k+1} \left(\widetilde{\mathcal{F}}_k\, \mathbf{x}_k - \mathcal{F}\, \mathbf{x}_k + \mathbf{y} \right)\,. \end{cases} \tag{15}$$

Because (15) allows for a non-injective $\widetilde{\mathcal{G}}_k$, this iteration can be used for optimization purposes. In case of an invertible $\widetilde{\mathcal{G}}_{k+1}$ both (14) and (15) are equivalent with (13b).

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 (12) exists and (13b)-(14) cannot converge (Figure 4). Therefore, we drop the idea of finding an $\mathbf{x} \in D$ satisfying (12) 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

$$\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x} \in D} \|\mathcal{F}\mathbf{x} - \mathbf{y}\|_{\widehat{E}}\,. \tag{16}$$

For a compact non-empty D and a continuous $\mathcal{F}$, at least a solution exists and if the operators $\widetilde{\mathcal{G}}_k$ are such that (13a) or (15) converges, the stationary point $\overline{\mathbf{x}}$ satisfies $\widetilde{\mathcal{G}}\mathcal{F}\,\overline{\mathbf{x}} = \widetilde{\mathcal{G}}\mathbf{y}$ or $\overline{\mathbf{x}} = \widetilde{\mathcal{G}}(\widetilde{\mathcal{F}}\,\overline{\mathbf{x}} - \mathcal{F}\,\overline{\mathbf{x}} + \mathbf{y})$ respectively. (We assume that $\widetilde{\mathcal{G}}_k = \widetilde{\mathcal{G}}$ and $\widetilde{\mathcal{F}}_k = \widetilde{\mathcal{F}}$ for k large enough.)

Now we can associate with each defect correction iteration a process for iterative optimization by taking $E = \mathbb{R}^n$, $\widehat{E} = \mathbb{R}^m$, $D = X$, $\widehat{D} = Y$ and $\overline{\mathbf{p}} : X \rightarrow Z$, and by substitution of the corresponding operators:

$$\begin{aligned} \mathcal{F}\mathbf{x} = \mathbf{y} \quad &\Leftrightarrow \mathbf{f}(\mathbf{x}) = \mathbf{y}\,, \\ \mathbf{x} = \mathcal{G}\mathbf{y} \quad &\Leftrightarrow \mathbf{x} = \operatorname*{argmin}_{\xi} \|\mathbf{f}(\xi) - \mathbf{y}\|\,, \\ \widetilde{\mathcal{F}}\mathbf{x} = \mathbf{y} \quad &\Leftrightarrow \mathbf{c}(\overline{\mathbf{p}}(\mathbf{x})) = \mathbf{y}\,, \\ \mathbf{x} = \widetilde{\mathcal{G}}\mathbf{y} \quad &\Leftrightarrow \mathbf{x} = \operatorname*{argmin}_{\xi} \|\mathbf{c}(\overline{\mathbf{p}}(\xi)) - \mathbf{y}\|\,. \end{aligned} \tag{17}$$

Remark. Notice that $\overline{\mathbf{p}}$ is *not* the space mapping function but an arbitrary (easy to compute) bijection, e.g., the identity.

4.2 Defect Correction for Optimization

With (17) we derive from (13a) and (15) two defect-correction iteration schemes for optimization. Substitution of (17) yields the initial estimate and two iteration processes for $k = 0, 1, 2, \cdots$, with $\mathbf{p}_{k+1}$ for $\overline{\mathbf{p}}$ in every step,

$$\mathbf{x}_0 = \operatorname*{argmin}_{\mathbf{x}\in X} \| \mathbf{c}(\mathbf{p}_0(\mathbf{x})) - \mathbf{y} \| \ , \tag{18}$$

$$\begin{aligned}\mathbf{x}_{k+1} = \mathbf{x}_k &- \operatorname*{argmin}_{\mathbf{x}\in X} \| \mathbf{c}(\mathbf{p}_{k+1}(\mathbf{x})) - \mathbf{f}(\mathbf{x}_k) \| \\ &+ \operatorname*{argmin}_{\mathbf{x}\in X} \| \mathbf{c}(\mathbf{p}_{k+1}(\mathbf{x})) - \mathbf{y} \| \ ,\end{aligned} \tag{19}$$

$$\mathbf{x}_{k+1} = \operatorname*{argmin}_{\mathbf{x}\in X} \| \mathbf{c}(\mathbf{p}_{k+1}(\mathbf{x})) - \mathbf{c}(\mathbf{p}_k(\mathbf{x}_k)) + \mathbf{f}(\mathbf{x}_k) - \mathbf{y} \| \ . \tag{20}$$

The two processes (19) and (20) are still dual in the sense that extrapolation is applied in the space X for process (19) and in Y for process (20). The operators $\mathbf{p}_k$ are right-preconditioners for the coarse model, which may be adapted during the initial steps of the iteration. We take $\mathbf{p}_k$ non-singular and for the initial estimate (18), and if $X = Z$ we usually take $\mathbf{p}_0 = I$, the identity.

In the above iterations every minimization involves 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 more convenient to write the procedures such that optimization over the coarse model becomes obvious. By taking in (13a) and (15) $\mathcal{F}\,\mathbf{z} = \mathbf{f}(\mathbf{q}(\mathbf{z}))$, $\widetilde{\mathcal{F}}_k\,\mathbf{z} = \mathbf{c}(\mathbf{z})$ 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 $k = 0, 1, 2, \cdots$,

$$\mathbf{z}_0 = \mathbf{z}^* = \operatorname*{argmin}_{\mathbf{z}\in Z} \| \mathbf{c}(\mathbf{z}) - \mathbf{y} \| \ , \tag{21}$$

$$\mathbf{z}_{k+1} = \mathbf{z}_k - \operatorname*{argmin}_{\mathbf{z}\in Z} \| \mathbf{c}(\mathbf{z}) - \mathbf{f}(\mathbf{q}_k(\mathbf{z}_k)) \| + \mathbf{z}^* \ , \tag{22}$$

$$\mathbf{z}_{k+1} = \operatorname*{argmin}_{\mathbf{z}\in Z} \| \mathbf{c}(\mathbf{z}) - \mathbf{c}(\mathbf{z}_k) + \mathbf{f}(\mathbf{q}_k(\mathbf{z}_k)) - \mathbf{y} \| \ . \tag{23}$$

As 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 $\mathbf{q}_k = \mathbf{p}_k^{-1}$. For $k > k_0$, we assume the iteration process to be stationary: $\mathbf{p}_k = \overline{\mathbf{p}}$ and $\mathbf{q}_k = \overline{\mathbf{q}}$. It is a little exercise to see by proper simplifications of (19) and (20) that space-mapping iteration can be recovered from defect correction [9, Section 4.3.2].

Orthogonality and the Need for Left-preconditioning.

For the stationary points of the above processes, we can derive the following lemma [9].

Lemma 7. *In the case of convergence of (23), with fixed point* $\lim_{k\to\infty} \mathbf{x}_k = \overline{\mathbf{x}}$ *we obtain*

$$\mathbf{f}(\overline{\mathbf{x}}) - \mathbf{y} \in \mathbf{c}(Z)^{\perp}(\overline{\mathbf{p}}(\overline{\mathbf{x}})) \, . \tag{24}$$

In case of convergence of (22) with a fixed point $\overline{\mathbf{x}}$ *we obtain*

$$\mathbf{f}(\overline{\mathbf{x}}) - \mathbf{y} \in \mathbf{c}(Z)^{\perp}(\mathbf{z}^*) \ . \tag{25}$$

Like the space-mapping methods, the above iterations have the disadvantage that, in general, the fixed point of the iteration does not coincide with the solution of the fine model minimization problem. This is due to the fact that the

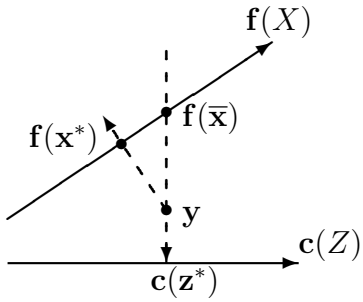


Fig. 5. The relative location of $\mathbf{c}(\mathbf{z}^*)$, $\mathbf{f}(\mathbf{x}^*)$ and $\mathbf{f}(\overline{\mathbf{x}})$

approximate solution $\overline{\mathbf{x}}$ satisfies either (24) or (25). whereas a (local) minimum $\mathbf{x}^* = \operatorname{argmin}_{x \in X} \|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|$ satisfies (see Figure 5)

$$\mathbf{f}(\mathbf{x}^*) - \mathbf{y} \in \mathbf{f}(X)^{\perp}(\mathbf{x}^*)\,. \tag{26}$$

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.

By the orthogonality relations above, we see that it is advantageous, both for the conditioning of the problem and for the minimization of the residual, 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 causes that the fixed point of traditional space mapping does generally not correspond with $\mathbf{x}^*$. This relation, however, can be improved by the introduction of an additional left-preconditioner. Therefore we introduce such a preconditioner $\mathbf{S}$ so that near $\mathbf{f}(\mathbf{x}^*) \in Y$ the manifold $\mathbf{c}(Z) \subset Y$ is mapped onto $\mathbf{f}(X) \subset Y$:

$$\mathbf{f}(\mathbf{x}) \approx \mathbf{S}(\mathbf{c}(\overline{\mathbf{p}}(\mathbf{x})))\,. \tag{27}$$

In the next section we propose a new algorithm where an affine operator maps $\mathbf{c}(Z)$ onto $\mathbf{f}(X)$ in the neighborhood of the solution. (More precisely: it approximately maps one tangential linear manifold onto the other.) This restores the orthogonality relation $\mathbf{f}(\overline{\mathbf{x}}) - \mathbf{y} \perp \mathbf{f}(X)(\mathbf{x}^*)$. Thus it improves significantly the traditional approach and makes the solution $\mathbf{x}^*$ a stationary point of the iteration. Details on the convergence of the processes can be found in [10].

5 Manifold Mapping, the Improved Space Mapping Algorithm

We introduce the affine mapping $\mathbf{S} : Y \to Y$ such that $\mathbf{S}\,\mathbf{c}(\overline{\mathbf{z}}) = \mathbf{f}(\mathbf{x}^*)$ for a proper $\overline{\mathbf{z}} \in Z$, and the linear manifold tangential to $\mathbf{c}(Z)$ in $\mathbf{c}(\overline{\mathbf{z}})$ maps onto the one tangential to $\mathbf{f}(X)$ in $\mathbf{f}(\mathbf{x}^*)$. Because, in the non-degenerate case when $m \geq n$, both $\mathbf{f}(X)$ and $\mathbf{c}(Z)$ are n-dimensional sets in $\mathbb{R}^m$, the mapping $\mathbf{S}$ can be described by

$$\mathbf{S}\,\mathbf{v} = \mathbf{f}(\mathbf{x}^*) + S\ \left(\mathbf{v} - \mathbf{c}(\overline{\mathbf{z}})\right)\ , \tag{28}$$

where S is an $m \times m$-matrix S of rank n. This mapping $\mathbf{S}$ is not a priori available, but an approximation to it can be computed iteratively during the optimization. A full rank $m \times m$-matrix S can be constructed, which has a well-determined part of rank n, while a remaining part of rank $m - n$ is free to choose. Because of the supposed similarity between the models $\mathbf{f}$ and $\mathbf{c}$ we keep the latter part close to the identity. The meaning of the mapping $\mathbf{S}$ is illustrated in the Figures 6 and 7

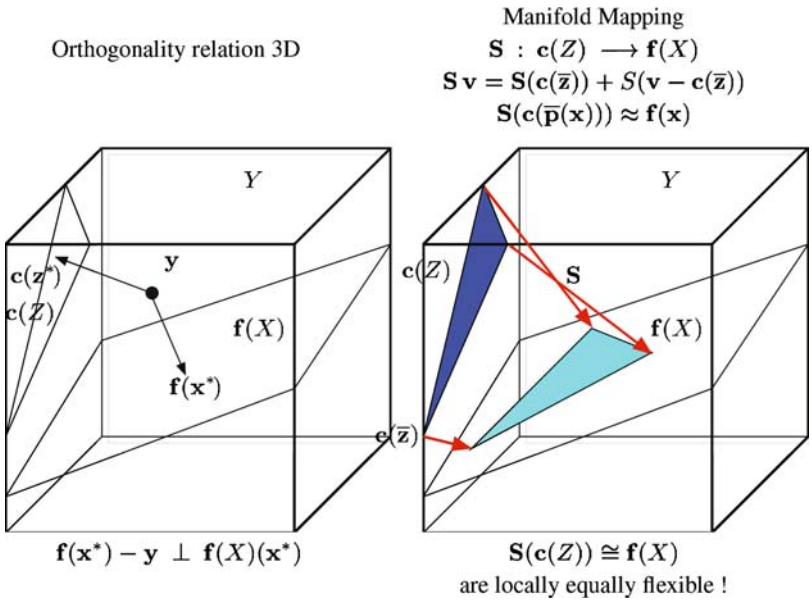
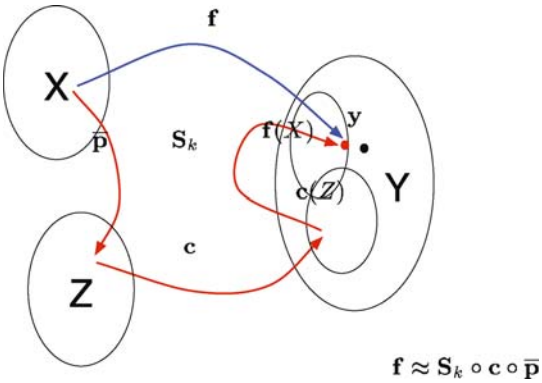


Fig. 6. Restoring the orthogonality relation by manifold mapping



Better mapping by left and right preconditioning.

Fig. 7. Manifold Mapping

So we propose the following algorithm (where the optional right-preconditioner $\overline{\mathbf{p}} : X \to Z$ is still an arbitrary non-singular operator. It can be adapted to the problem. Often we will simply take the identity.)

1. Set $k = 0$, set $S_0 = I$ the $m \times m$ identity matrix, and compute
$$\mathbf{x}_0 = \operatorname{argmin}_{\mathbf{x}\in X} \| \mathbf{c}(\overline{\mathbf{p}}(\mathbf{x})) - \mathbf{y}\| \ . \tag{29}$$
2. Compute $\mathbf{f}(\mathbf{x}_k)$ and $\mathbf{c}(\overline{\mathbf{p}}(\mathbf{x}_k))$.
3. If $k > 0$, with $\Delta\mathbf{c}_i = \mathbf{c}(\overline{\mathbf{p}}(\mathbf{x}_{k-i})) - \mathbf{c}(\overline{\mathbf{p}}(\mathbf{x}_k))$ and $\Delta\mathbf{f}_i = \mathbf{f}(\mathbf{x}_{k-i}) - \mathbf{f}(\mathbf{x}_k)$, $i = 1, \cdots, \min(n, k)$, we define ΔC and ΔF to be the rectangular $m \times \min(n, k)$-matrices with respectively $\Delta\mathbf{c}_i$ and $\Delta\mathbf{f}_i$ as columns. Their singular value decompositions are respectively $\Delta C = U_c \Sigma_c V_c^T$ and $\Delta F = U_f \Sigma_f V_f^T$.
4. The next iterand is computed as
$$\mathbf{x}_{k+1} = \operatorname*{argmin}_{\mathbf{x}\in X} \|\mathbf{c}(\overline{\mathbf{p}}(\mathbf{x})) - \mathbf{c}(\overline{\mathbf{p}}(\mathbf{x}_k)) + \left[\Delta C \Delta F^\dagger + I - U_c U_c^T\right] (\mathbf{f}(\mathbf{x}_k) - \mathbf{y})\|. \tag{30}$$
5. Set $k := k + 1$ and goto 2.

Here, † denotes the pseudo-inverse: $\Delta F^\dagger = V_f \Sigma_f^{-1} U_f^T$. It can be shown that (30) is asymtotically equivalent to
$$\mathbf{x}_{k+1} = \operatorname{argmin}_{\mathbf{x}\in X} \|\mathbf{S}_k(\mathbf{c}(\overline{\mathbf{p}}(\mathbf{x}))) - \mathbf{y}\| \ . \tag{31}$$

Above, the matrix $S_k = \Delta F \, \Delta C^\dagger + (I - U_\mathbf{f} \, U_\mathbf{f}^T) \, (I - U_\mathbf{c} \, U_\mathbf{c}^T)$ and the approximate affine mapping is
$$\mathbf{S}_k \, \mathbf{v} = \mathbf{f}(\mathbf{x}_k) + S_k(\mathbf{v} - \mathbf{c}(\overline{\mathbf{p}}(\mathbf{x}_k)) \, , \quad \forall \mathbf{v} \in Y,$$
which, for $l > 0$ and $l = k - 1, \cdots, \max(0, k - n)$, satisfies
$$S_k \, (\mathbf{c}(\overline{\mathbf{p}}(\mathbf{x}_l)) - \mathbf{c}(\overline{\mathbf{p}}(\mathbf{x}_k))) = \mathbf{f}(\mathbf{x}_l) - \mathbf{f}(\mathbf{x}_k) \ .$$
In (30), the freedom in making S_k full-rank is used, replacing $\Delta C \ \Delta F^\dagger + (I - U_c U_c^T)(I - U_f U_f^T)$ by $\Delta C \ \Delta F^\dagger + I - U_c U_c^T$, in order to stabilize the algorithm. This does not change the solution.

If the above iteration converges with fixed point $\overline{\mathbf{x}}$ and mappings $\overline{\mathbf{S}}$ and $\overline{\mathbf{p}}$, we have
$$\mathbf{f}(\overline{\mathbf{x}}) - \mathbf{y} \in \overline{\mathbf{S}}(\mathbf{c}(\overline{\mathbf{p}}(X)))^\perp(\overline{\mathbf{x}}) = \mathbf{f}(X)^\perp(\overline{\mathbf{x}}) \ . \tag{32}$$
From this relation and the fact that $\mathbf{S}_k(\mathbf{c}(\overline{\mathbf{p}}(\mathbf{x}_k))) = \mathbf{f}(\mathbf{x}_k)$, it can be concluded that, under convergence to $\overline{\mathbf{x}}$, the fixed point is a (local) optimum of the fine model minimization.

The improved space-mapping scheme
$$\mathbf{x}_{k+1} = \operatorname*{argmin}_{\mathbf{x}} \| \mathbf{S}_k(\mathbf{c}(\overline{\mathbf{p}}_k(\mathbf{x})))) - \mathbf{y}\| \tag{33}$$
can also be recognized as defect correction iteration with either $\widetilde{\mathcal{F}}_k = \mathbf{S}_k \circ \mathbf{c} \circ \overline{\mathbf{p}}$ and $\mathcal{F} = \mathbf{f}$ in (19) or (20), or with $\widetilde{\mathcal{F}}_k = \mathbf{S}_k \circ \mathbf{c}$ and $\mathcal{F} = \mathbf{f} \circ \overline{\mathbf{p}}^{-1}$ in (22) or (23).

6 Examples

We illustrate the application of space-mapping and manifold-mapping by a design problem for a linear actuator. We compare the performance of these algorithms with that of two classical optimization methods: Nelder-Mead Simplex (NMS) and Sequential Quadratic Programming (SQP).

Linear actuators are electromechanical devices that convert electromechanical power into linear motion. An axi-symmetrical variant, called a *voice-coil* actuator, consisting of a permanent magnet, a current-carrying coil and a ferromagnetic core is shown in Figure 8. The permanent magnet is magnetized in the vertical direction. The coil, steered by the magnetic force, moves along the z-axis in the gap of the core, as illustrated in Figure 9. The position of the coil relative to the top of the

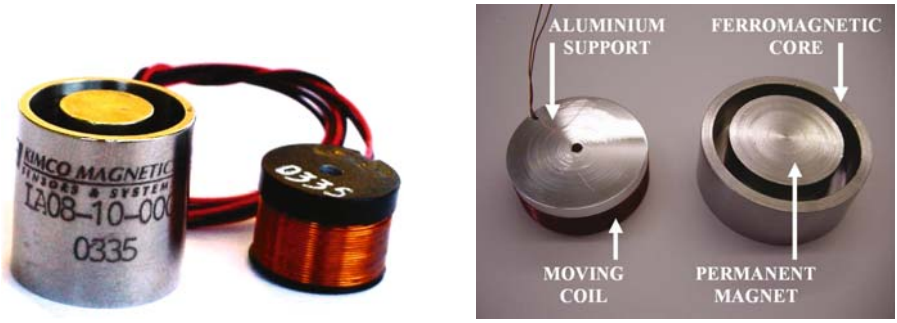
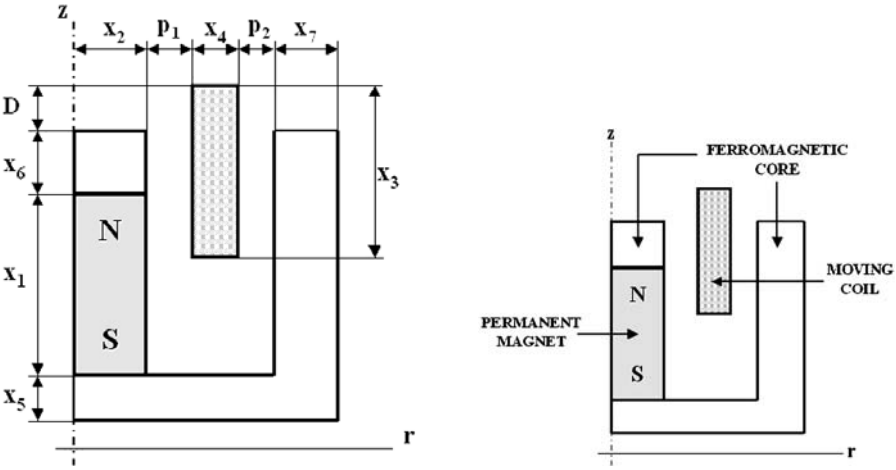


Fig. 8. A cylindrical voice-coil actuator consisting of a ferromagnetic core, permanent magnet and coil



Design Variables. Geometry.

Fig. 9. Geometry and design variables of the cylindrical voice-coil actuator

core is denoted by D. Due to the axisymmetrical geometry, the force has an axial component only. It will be denoted by $F_z(D)$.

The design variables [13] are shown in Figure 9: x_1 and x_2 denote the height and radius of the magnet, x_3 and x_4 the height and thickness of the coil and x_5, x_6 and x_7 the sizes of the core. Two additional linear inequality constraints define feasible coil positions. The air-gap sizes p_1 and p_2 to the left and right of the coil are kept fixed. Remaining details are found in [11].

We allow the coil to move over a $4\,\mathrm{mm}$ range, i.e., $0 \leq \mathrm{D} \leq 4\,\mathrm{mm}$. The force on the coil is computed at nine equidistant points D_i in this interval. Values for the design variables have to be found such that the force response is flat and as close to $\mathbf{y} = 24\,\mathrm{N}$ as possible. The cost function is

$$\left(\sum_{i=1}^{9} [F_z(D_i) - \mathbf{y}(D_i)]^2 / \sum_{i=1}^{9} \mathbf{y}(D_i)^2 \right)^{1/2} . \tag{34}$$

The fine model is a second order Lagrangian finite element (FE) model in which the non-linear BH-curve of the ferromagnetic core is taken into account. The force is computed by means of the Lorentz Force Law [8]. The number of degrees of freedom in the FE model is between 8000 and 11000, yielding three digits of accuracy in the computed force.

The first of two coarse models is a FE model in which the BH-curve of the actuator core is linearized. Depending on the number of Newton iterations required in the non-linear case, this model is a factor between 30 and 50 cheaper than the fine one. The second coarse model is a lumped parameter model. This so-called magnetic equivalent circuit (MEC) [8] model has a negligible computational cost compared to the fine one. In both the FE and the MEC coarse models, the relative magnetic permeability in the core was overestimated and set equal to 1000. This was done for illustration purposes.

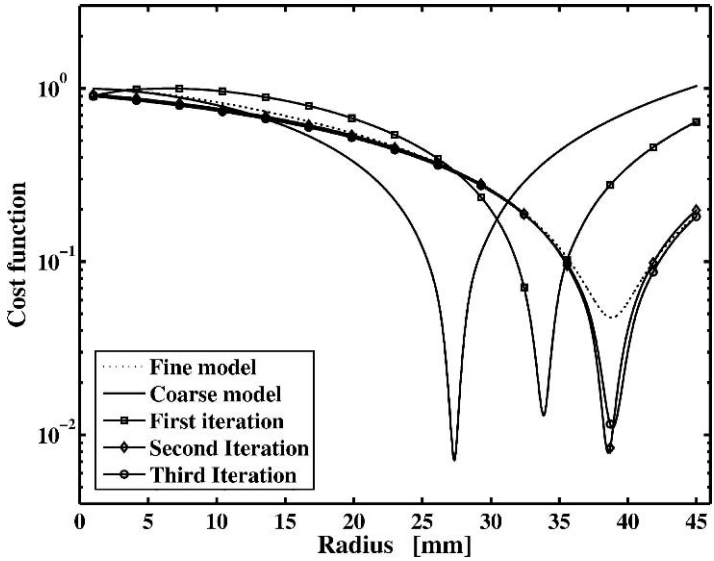
Below we will consider three variants of modelling approaches for this type of problem. The use of manifold mapping with the linearized finite element, respectively the MEC as coarse model, will be denoted by FE/MM and MEC/MM. Similar notations FE/SM and MEC/SM are used for space mapping.

6.1 A Variant with One Design Variable

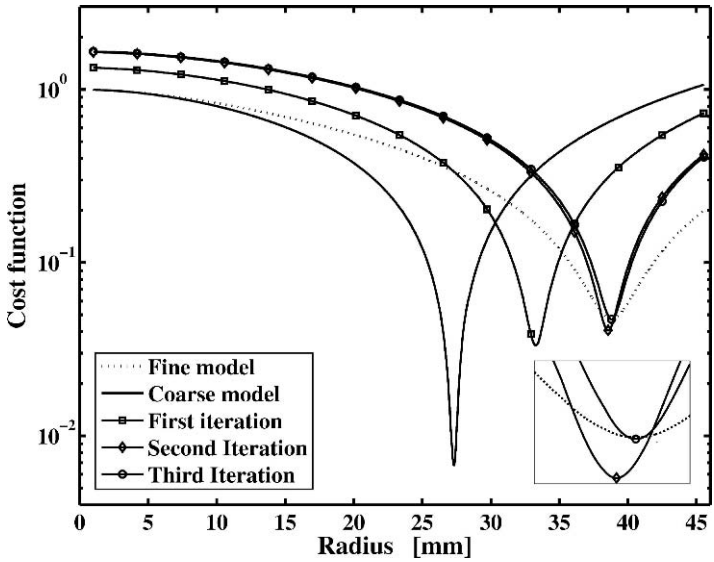
We initially consider a design problem with a single design variable, only varying the radius of the permanent magnet. We denote the design variable x_2 simply by $\mathbf{x}$. As a starting guess we use the coarse model optimum, i.e., $x_0 = z^*$, as in Section 4.2, where the choice $\mathbf{p}_0 = I$ was made. For this one-parameter problem both space mapping (SM) and manifold mapping (MM), with either the linear FE or the MEC as coarse model, converge to the unique $\mathbf{x}^*$ in four iterations and both methods deliver a speed-up with a factor between four and five compared with the NMS or the SQP algorithm [11].

The cost function associated with the surrogate model that MM builds in the final iteration step approximates the fine model cost function in a neighbourhood

$\mathbf{x}^*$ much better than its SM counterpart. We illustrate the convergence of SM and MM by looking at the cost function of the surrogate models during successive iterations. Figure 10 (top) shows the cost functions of the surrogate model during the first



Space-Mapping (SM).



Manifold-Mapping (MM).

Fig. 10. Convergence history of SM and MM using the MEC as coarse model

Table 1. Computational efficiency of SM and MM for an example with a single design variable, compared with the NMS and the SQP method

	# iters	# **f** evals.	# **c** evals
NMS	10	20	20
SQP	5	18	20
MEC/SM	4	4	80
MEC/MM	4	4	80

MEC/SM iterations, i.e., $\|\mathbf{c}(\mathbf{p}_k(\mathbf{x})) - \mathbf{y}\|_2/\|\mathbf{y}\|_2$, for $k = 1, \ldots, 3$ as function of $\mathbf{x}$. The coarse ($k = 0$) and fine model cost functions are also shown. Figure 10 (bottom) shows the same for MEC/MM with $\|\mathbf{S}_k(\mathbf{c}(\mathbf{x})) - \mathbf{y}\|_2/\|\mathbf{y}\|_2$ for successive k. The overestimation of the magnetic permeability of the core in the coarse models is such that for these models a smaller radius is required to reach the design objective, i.e, $\mathbf{z}^* < \mathbf{x}^*$. The figures also illustrate the convergence of the iterands $\mathbf{x}_k$ to $\mathbf{x}^*$. They furthermore show that the mapping of the tangent manifold in MM provides a better approximation of the fine model cost function in a neighbourhood of $\mathbf{x}_f^*$.

To show the speed-ups that SM and MM-algorithms may yield, in Table 1 we show the number of fine and coarse model evaluations of MEC/SM and MEC/MM as well as the number required by NMS and SQP. For the latter two, the coarse model was used to generate an appropriate initial guess. In the other two algorithms each iteration requires one fine and twenty coarse model evaluations. From the table the computational speed-up is obvious. Even though the coarse model was chosen to be quite inaccurate, the SM based algorithms deliver a significant speed-up.

To quantify the difference between the two coarse models, in Figure 11 we show the decrease in cost function during SM and MM iteration with both coarse models. From this figure we conclude that the linear FE coarse model does not accelerate the converge of SM or MM better than the (much cheaper) MEC model. A linear FE coarse model can however be advantageous in more complex design problems.

6.2 A Variant with Two Design Variables

We now consider a design problem with two design variables, allowing changes in height (x_1) and radius (x_2) of the permanent magnet. Numerical results comparing the performance of SM and MM with NMS and SQP for this problem are given in Table 2. The first row in this table gives the total amount of work expressed in number of equivalent fine model evaluations. These figures are approximately proportional to the total computing time. As starting guess for the optimization procedures we used the values obtained by optimizing the MEC model. This design problem is extremely ill-conditioned and has a manifold of equivalent solutions. To stabilize the convergence of MM, the Levenberg-Marquardt method is used. The best results in terms of computational efficiency (speed-up by a factor of six) are obtained using MM with the MEC as coarse model. Full details about this problem and its solution by SM or MM are found in [11].

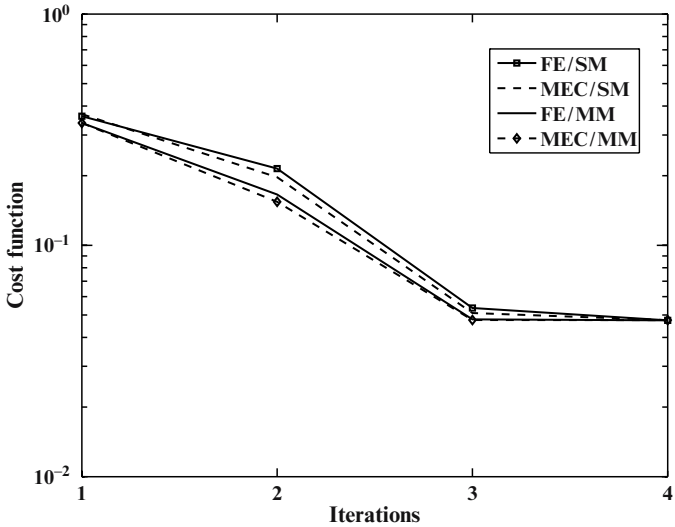


Fig. 11. Reduction in cost function value in successive iterations of SM and MM

Table 2. Computational efficiency of SM and MM for the example with two design variables. The total amount of computational work is approximately equal to the cost of the fine model function evaluations (# $\mathbf{f}$ evals.)

	NMS	SQP	FE/SM	MEC/SM	FE/MM	MEC/MM
# $\mathbf{f}$ evals.	24	31	9	6	9	4
cost function	0.046	0.046	0.046	0.065	0.046	0.046

6.3 A Variant with Seven Design Variables

In the last example we show the potential of MM and SM in the problem with all seven design variables and non-linear equality and inequality constraints. This design problem was introduced in [12] and details can be found in [11]. The total mass of the actuator has to be minimized, while the mass of the coil is constrained to $10\,\mathrm{g}$. Thus, the cost function is the total mass of the device. The force at coil position $\mathrm{D} = 4.25\,\mathrm{mm}$ should be kept at $5\,\mathrm{N}$ and the magnetic flux density in three regions of the core should not exceed $1\,\mathrm{T}$. In the fine model the constraints are evaluated by the same FE model as used in the two previous design problems. In the coarse model the constraints are based on a MEC model. Each coarse model related optimization is solved by SQP. Either MM or SM is applied for the constraints evaluation.

Numerical results for this problem are shown in Table 3. SM and MM show a similar behaviour: convergence is reached in seven or six fine constraint evaluations respectively. Having the coarse model optimum $\mathbf{z}^*$ as the initial guess, SQP converges within 56 fine constraint evaluations. MM offers an additional advantage over SM: the computation of the SM function $\mathbf{p}(\mathbf{x})$ is a very delicate issue [4], but MM replaces it simply by the identity.

Table 3. Computational efficiency of SM and MM for an example with seven design variables

	# evals.	total mass	final design (mm)
SQP	56	81.86 g	[8.543, 9.793, 11.489, 1.876, 3.876, 3.197, 2.524]
SM	7	81.11 g	[8.500, 9.786, 11.450, 1.883, 3.838, 3.200, 2.497]
MM	6	81.45 g	[8.500, 9.784, 11.452, 1.883, 3.860, 3.202, 2.515]

The initial guess for SQP is the coarse model optimum $\mathbf{z}^*$.
The total amount of work is approximately equal to the cost of the fine model constraint evaluations (# evals.).

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.

When analyzing the behaviour of space-mapping iteration, it is important to know the notions of reachability of a design and flexibility of the underlying models. One can show that if neither the design is reachable nor the models are equally flexible, space mapping iteration does generally not converge to the (accurate) solution of the optimization problem.

Using the principle of defect correction iteration, we can repair this deficiency and construct the manifold-mapping iteration, which is as efficient as space mapping, but converges to the right solution.

Our findings are illustrated by an example from electromagnetics. Here parameters for the design of a voice coil actuator are determined, using a finite element discretization for the fine model and an equivalent magnetic circuit description for the coarse one.

References

1. 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.
2. M.H. Bakr, J.W. Bandler, K. Madsen, and J. Søndergaard. An introduction to the space mapping technique. *Optimization and Engineering*, 2(4):369–384, 2001.
3. 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*, 42:2536–2544, 1994.
4. J.W. Bandler, Q.S. Cheng, A.S. Dakroury, A.S. Mohamed, M.H. Bakr, K. Madsen, and J. Søndergaard. Space mapping: The state of the art. *IEEE Transactions on Microwave Theory and Techniques*, 52:337–360, 2004.
5. 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.

6. 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*, Computing Suppl. 5, pages 1–32. Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1984.
7. K. Böhmer and H.J. Stetter. *Defect Correction Methods: Theory and Applications*. Springer, Berlin, 1984.
8. D.K. Cheng. *Field and Wave Electromagnetics*. Cambridge University Press, 1989.
9. D. Echeverría and P.W. Hemker. Space mapping and defect correction. *Comp. Methods in Appl. Math.*, 5(2):107–136, 2005.
10. D. Echeverría and P.W. Hemker. On the manifold mapping optimization technique. Technical Report MAS-E0612, CWI, 2006. Submitted for publication.
11. 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. *IEEE. Trans. on Magn.*, 42(4):1183–1186, 2006. (Also: Technical Report MAS-E0612, CWI).
12. 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. *Struct. Multidisc. Optim.*, 2007. DOI 10.1007/s00158-006-0054-6. To appear. (Also in: J. Herkovits, Mazorche S, and A. Canelas, editors, Sixth World Congress on Structural and Multidisciplinary Optimization (WCSMO6), Brazil, 2005. paper 5631).
13. L. Encica, J. Makarovic, E.A. Lomonova, and A.J.A. Vandenput. Space mapping optimization of a cylindrical voice coil actuator. *IEEE IEMDC-2005*, 2005. Conference proceedings.