Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 224 (2007) 464–475
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A trust-region strategy for manifold-mapping optimization

P.W. Hemker *, D. Echeverría

Centrum voor Wiskunde en Informatica Kruislaan 413, NL 1098 SJ Amsterdam, The Netherlands

Received 1 September 2006; received in revised form 13 February 2007; accepted 2 April 2007
Available online 11 April 2007

Abstract

Studying the space-mapping technique by Bandler et al. [J. Bandler, R. Biernacki, S. Chen, P. Grobelny, R.H. Hemmers, Space mapping technique for electromagnetic optimization, IEEE Trans. Microwave Theory Tech. 42 (1994) 2536–2544] for the solution of optimization problems, we observe the possible difference between the solution of the optimization problem and the computed space-mapping solution. 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 exact solution.

To increase the robustness of the algorithm we introduce a trust-region strategy (a regularization technique) based on the generalized singular value decomposition of the linearized fine and coarse manifold representations. The effect of this strategy is shown by the solution of a variety of small non-linear least squares problems. Finally we show the use of the technique for a more challenging engineering problem.

PACS: 02.60.Pn

Keywords: Multilevel optimization; Defect-correction; Space-mapping; Manifold-mapping; Trust-region methods

1. Introduction

In practice design problems often require expensive numerical simulations to model the phenomenon that has to be optimized. Because, generally, numerical optimization is an iterative process, many of these time-consuming simulations are needed before a satisfactory solution is found. Space mapping, introduced by Bandler et al. [1–3], is a technique for reducing the computing time in such procedures by the use of simpler surrogate models. Space mapping involves both the accurate (and time-consuming) model and a less accurate (but cheaper) one.

The classical space-mapping procedure uses right-preconditioning of the coarse (inaccurate) model in order to accelerate the iterative procedure for the optimization of the fine (accurate) one. We show in [4] that right-preconditioning is generally insufficient and (also) left-preconditioning is needed. This leads to the improved

* Corresponding author.
E-mail address: pieth@cwi.nl (P.W. Hemker).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2007.04.003

space-mapping or 'manifold-mapping' procedure[1]. In addition, to make the method more efficient for non-linear and strongly ill-conditioned problems, we introduce a trust-region strategy that prevents the method from seeking a local optimum that is too far away from the solution of the coarse problem.

2. Fine and coarse models in optimization

2.1. The optimization problem

Let the specifications of an optimization problem be denoted by $(\mathbf{t},\mathbf{y}) \equiv (\{t_i\},\{y_i\})_{i=1,\ldots,m}$. Here, the independent variable is $\mathbf{t} \in \mathbb{R}^m$, and the dependent variable $\mathbf{y} \in Y$ represents the quantities that describe the behaviour of the phenomena under study. The set $Y \subset \mathbb{R}^m$ is the *set of possible aims*.

The variable $\mathbf{y}$ does not only depend on $\mathbf{t}$ but also on control/design variables, $\mathbf{x}$. 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 that 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 the fine and the coarse model.

2.2. The fine model

The *fine model* response is denoted by $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^m$, with $\mathbf{x} \in X \subset \mathbb{R}^n$ the *fine model control variable*. The set $\mathbf{f}(X) \subset \mathbb{R}^m$ represents the fine model reachable aims. The fine model is assumed to be *accurate* but *expensive* to evaluate. For the optimization problem a *fine model cost function*, $\|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|$ should be minimized. This cost function measures the discrepancy between the aim, $\mathbf{y}$, and the response of the mathematical model, $\mathbf{f}(\mathbf{x})$. So we look for

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

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 a *reachable design* if the equality $\mathbf{f}(\mathbf{x}^*) = \mathbf{y}$ can be achieved for some $\mathbf{x}^* \in X$.

2.3. The coarse model

The *coarse model* gives a simpler but less accurate description of the same phenomenon as the fine model. It 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*, $\|\mathbf{c}(\mathbf{z}) - \mathbf{y}\|$ and we denote its minimizer by $\mathbf{z}^*$,

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

2.4. The space-mapping function

The similarity or discrepancy between the responses of two models is expressed by the *misalignment function* $r(\mathbf{z},\mathbf{x}) = \|\mathbf{c}(\mathbf{z}) - \mathbf{f}(\mathbf{x})\|$. For a given $\mathbf{x} \in X$ it is useful to know which $\mathbf{z} \in Z$ yields the smallest discrepancy. Therefore, the *space-mapping function* is introduced, $\mathbf{p} : X \subset \mathbb{R}^n \to Z \subset \mathbb{R}^n$ defined by

$$\mathbf{p}(\mathbf{x}) = \arg\min_{\mathbf{z}\in Z} r(\mathbf{z},\mathbf{x}) = \arg\min_{\mathbf{z}\in Z} \|\mathbf{c}(\mathbf{z}) - \mathbf{f}(\mathbf{x})\|. \tag{3}$$

[1] In space-mapping terminology, manifold-mapping (MM) is a special case of output space mapping (OSM).

2.5. Perfect mapping

To identify the cases where the accurate solution $\mathbf{x}^*$ is related with the less accurate solution $\mathbf{z}^*$ by the space-mapping function, a space-mapping function $\mathbf{p}$ is called a *perfect mapping* iff $\mathbf{z}^* = \mathbf{p}(\mathbf{x}^*)$.

We notice that *perfection* is not a genuine property of the space-mapping function, because it also depends on the data $\mathbf{y}$ considered. It is important to be aware that a space-mapping function can be perfect for one set of data but imperfect for a different data set, and if a design is reachable a space-mapping is always perfect irrespective of the coarse model used.

3. Primal and dual space-mapping solutions

In literature many variants of the space-mapping algorithm can be found [2,3]. The best known are aggressive space-mapping (ASM) and trust-region ASM (TRASM). However, we can distinguish two fundamentally different types: the primal and the dual approach.

The *primal* space-mapping approach seeks for a solution of the minimization problem

$$\mathbf{x}_p^* = \arg\min_{\mathbf{x}\in X} \|\mathbf{p}(\mathbf{x}) - \mathbf{z}^*\|, \tag{4}$$

whereas the *dual* determines

$$\mathbf{x}_d^* = \arg\min_{\mathbf{x}\in X} \|\mathbf{c}(\mathbf{p}(\mathbf{x})) - \mathbf{y}\|. \tag{5}$$

In the last equation we recognize $\mathbf{c}(\mathbf{p}(\mathbf{x}))$ as a *surrogate model*.

Both approaches coincide when $\mathbf{z}^* \in \mathbf{p}(X)$ and $\mathbf{p}$ is injective. If, in addition, 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 algorithm, based on either the primal or the dual approach, may *not* yield the solution of the fine model optimization. The principle of the approach is summarized in Fig. 1.

4. Defect-correction iteration

The efficient solution of a complex problem by the iterative use of a simpler one, is known since long in computational mathematics as defect-correction iteration [5]. To solve a nonlinear operator equation

$$\mathscr{F}\mathbf{x} = \mathbf{y}, \tag{6}$$

where $\mathscr{F} : D \subset E \to \widehat{D} \subset \widehat{E}$ is a continuous, generally nonlinear operator and E and $\widehat{E}$ are Banach spaces, defect-correction iteration reads

$$\begin{cases} \mathbf{x}_0 = \widetilde{\mathscr{G}}_0 \mathbf{y}, \\ \mathbf{x}_{k+1} = \widetilde{\mathscr{G}}_{k+1}(\widetilde{\mathscr{F}}_k \mathbf{x}_k - \mathscr{F}\mathbf{x}_k + \mathbf{y}), \end{cases} \tag{7}$$

where $\widetilde{\mathscr{F}}_k$ is a simpler version of $\mathscr{F}$ and $\widetilde{\mathscr{G}}_k$ is the (simple-to-evaluate) left-inverse of $\widetilde{\mathscr{F}}_k$.

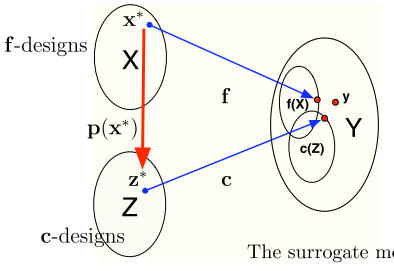


Fig. 1. The space-mapping function $\mathbf{p}(\mathbf{x}) = \arg\min_{\mathbf{z}\in Z}\|\mathbf{c}(\mathbf{z}) - \mathbf{f}(\mathbf{x})\|$.

For our optimization problems, where the design may be not reachable, $\mathbf{y} \in \widehat{D}$, but $\mathbf{y} \notin \mathscr{F}(D)$, so that no solution for (6) exists. In this case we want to find the solution

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in D} \|\mathscr{F}\mathbf{x} - \mathbf{y}\|_{\widehat{E}}. \tag{8}$$

With this problem we associate a defect-correction process for iterative optimization by taking $E = \mathbb{R}^n$, $\widehat{E} = \mathbb{R}^m$, $D = X$, $\widehat{D} = Y$, and by substitution of the operators:

$$\begin{aligned}
&\mathscr{F}\mathbf{x} = \mathbf{y} \Longleftrightarrow \mathbf{f}(\mathbf{x}) = \mathbf{y},\\
&\mathbf{x} = \mathscr{G}\mathbf{y} \Longleftrightarrow \mathbf{x} = \arg\min_{\xi\in X} \|\mathbf{f}(\xi) - \mathbf{y}\|,\\
&\widetilde{\mathscr{F}}_k\mathbf{x} = \mathbf{y} \Longleftrightarrow \mathbf{c}(\overline{\mathbf{p}}_k(\mathbf{x})) = \mathbf{y},\\
&\mathbf{x} = \widetilde{\mathscr{G}}_k\mathbf{y} \Longleftrightarrow \mathbf{x} = \arg\min_{\xi\in X} \|\mathbf{c}(\overline{\mathbf{p}}_k(\xi)) - \mathbf{y}\|.
\end{aligned} \tag{9}$$

Here $\overline{\mathbf{p}}_k$ is *not* the space-mapping function but an arbitrary (easy-to-compute) bijection, e.g., if $X = Z$ we choose the identity. Notice that, in principle, also $\mathbf{c} = \mathbf{c}_k$ might be adapted during the iteration.

With (9) we derive from (7) the defect-correction iteration scheme for optimization:

$$\begin{cases}
\mathbf{x}_0 = \arg\min_{\mathbf{x}\in X} \|\mathbf{c}(\overline{\mathbf{p}}_0(\mathbf{x})) - \mathbf{y}\|,\\
\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}\in X} \|\mathbf{c}(\overline{\mathbf{p}}_{k+1}(\mathbf{x})) - \mathbf{c}(\overline{\mathbf{p}}_k(\mathbf{x}_k)) + \mathbf{f}(\mathbf{x}_k) - \mathbf{y}\|.
\end{cases} \tag{10}$$

In this iteration every minimization involves the surrogate model, $\mathbf{c} \circ \overline{\mathbf{p}}_k$.

4.1. Orthogonality and the need for left-preconditioning

For the stationary points of the above process, $\lim_{k\to\infty}\mathbf{x}_k = \overline{\mathbf{x}}$, we can derive [4]:

$$\mathbf{f}(\overline{\mathbf{x}}) - \mathbf{y} \in \mathbf{c}(Z)^{\perp}(\mathbf{z}^*). \tag{11}$$

Like the space-mapping methods, the above iteration has the disadvantage that, in general, the fixed point, $\overline{\mathbf{x}}$, does not coincide with the solution of the fine model minimization problem, $\mathbf{x}^*$. This is due to the fact that the approximate solution $\overline{\mathbf{x}}$ satisfies (11), whereas the (local) minimum $\mathbf{x}^*$ 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 $\mathbf{f}(X)$ and $\mathbf{c}(Z)$ near the optima.

By these orthogonality relations 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 neighbourhood of the solution. However, by space-mapping or by right-preconditioning the relation between $\mathbf{f}(X)$ and $\mathbf{c}(Z)$ remains unchanged. The fact that the fixed point of traditional space-mapping does, generally, not correspond with $\mathbf{x}^*$ can be corrected 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 $\mathbf{f}(X) \subset Y$:

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

This restores the orthogonality relation $\mathbf{f}(\overline{\mathbf{x}}) - \mathbf{y} \perp \mathbf{f}(X)(\overline{\mathbf{x}})$. Thus it improves significantly the traditional space-mapping approach and makes the solution $\mathbf{x}^*$ a stationary point of the iteration. In the next section we propose our *manifold-mapping* algorithm, where $\mathbf{S}$ is an affine operator which maps $\mathbf{c}(\overline{\mathbf{p}}(\mathbf{x}_k))$ to $\mathbf{f}(\mathbf{x}_k)$ and, approximately, one tangential linear manifold onto the other.

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 $\mathbf{c}(\overline{\mathbf{z}})$ is mapped onto the one tangential to $\mathbf{f}(X)$ in $\mathbf{f}(\mathbf{x}^*)$. Because both $\mathbf{f}(X)$ and $\mathbf{c}(Z)$ are n-dimensional manifolds in $\mathbb{R}^m$, the mapping $\mathbf{S}$ can be described by

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

where S is an $m \times m$-matrix of rank n. A full rank $m \times m$-matrix S can be constructed, with a well-determined part of rank n and a remaining part of rank $m - n$ which 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 Fig. 2.

So, we propose the following manifold-mapping (MM) algorithm [4], where the optional right-preconditioner $\overline{\mathbf{p}}: X \to Z$ is still an arbitrary non-singular operator, e.g., the identity.

(1) Set $k = 0$ and compute

$$\mathbf{x}_0 = \arg\min_{\mathbf{x}\in X} \|\mathbf{c}(\overline{\mathbf{p}}(\mathbf{x})) - \mathbf{y}\|.$$

(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 $\Delta\mathbf{f}_i = \mathbf{f}(\mathbf{x}_{k-i}) - \mathbf{f}(\mathbf{x}_k)$, $i = 1, \ldots, \min(n, k)$; 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 (SVD) are respectively $\Delta C = U_c \Sigma_c V_c^{\mathrm{T}}$ and $\Delta F = U_f \Sigma_f V_f^{\mathrm{T}}$.

(4) The next iterant is computed as

$$\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}\in X} \|\mathbf{c}(\overline{\mathbf{p}}(\mathbf{x})) - \mathbf{c}(\overline{\mathbf{p}}(\mathbf{x}_k)) + [\Delta C \Delta F^{\dagger} + (I - U_c U_c^{\mathrm{T}})A](\mathbf{f}(\mathbf{x}_k) - \mathbf{y})\|. \tag{12}$$

(5) Stop if some convergence criterion is satisfied, otherwise set $k := k + 1$ and go to (2).

Here, A is an arbitrary $m \times m$-matrix for which we prefer the identity and $\Delta F^{\dagger}$ denotes the pseudo-inverse of ΔF. It can be shown [6] that (12) is asymptotically equivalent to

$$\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}\in X} \|\mathbf{S}_k(\mathbf{c}(\overline{\mathbf{p}}(\mathbf{x}))) - \mathbf{y}\|, \tag{13}$$

where the affine mapping $\mathbf{S}_k$, an approximation to $\mathbf{S}$, is given by

$$\mathbf{S}_k\mathbf{v} = \mathbf{f}(\mathbf{x}_k) + S_k(\mathbf{v} - \mathbf{c}(\overline{\mathbf{p}}(\mathbf{x}_k))), \tag{14}$$

with $S_k = \Delta F \Delta C^{\dagger} + (I - U_{\mathbf{f}} U_{\mathbf{f}}^{\mathrm{T}})(I - U_{\mathbf{c}} U_{\mathbf{c}}^{\mathrm{T}})$.

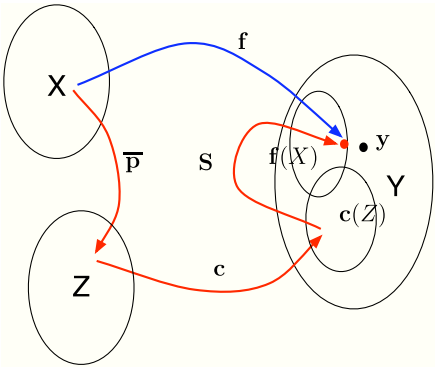


Fig. 2. Manifold mapping.

If the above iteration converges with fixed point $\overline{\mathbf{x}}$ and mapping $\overline{\mathbf{S}}$, we have

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

This and the fact that $\mathbf{S}_k(\mathbf{c}(\overline{\mathbf{p}}(\mathbf{x}_k))) = \mathbf{f}(\mathbf{x}_k)$, makes that, under convergence to $\overline{\mathbf{x}}$, the fixed point is a (local) optimum of the fine model minimization and as a consequence $\overline{\mathbf{S}} = \mathbf{S}$, cf. [6].

The improved space-mapping scheme,

$$\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}\in X} \|\mathbf{S}_k(\mathbf{c}(\overline{\mathbf{p}}(\mathbf{x}))) - \mathbf{y}\|,$$

can also be recognized as defect-correction iteration with either $\widetilde{\mathscr{F}}_k = \mathbf{S}_k \circ \mathbf{c} \circ \overline{\mathbf{p}}$ and $\mathscr{F} = \mathbf{f}$ or with $\widetilde{\mathscr{F}}_k = \mathbf{S}_k \circ \mathbf{c}$ and $\mathscr{F} = \mathbf{f} \circ \overline{\mathbf{p}}^{-1}$.

6. A trust-region strategy

Linear convergence of the MM-algorithm is proved in [6] under the conditions that: (i) $\mathbf{c}(\overline{\mathbf{p}}(X))$ and $\mathbf{f}(X)$ are C^2-manifolds, (ii) the models $\mathbf{c}(\overline{\mathbf{p}}(\mathbf{x}))$ and $\mathbf{f}(\mathbf{x})$ show a sufficiently similar behaviour in the neighbourhood of the solution, and (iii) the matrices ΔC and ΔF are sufficiently well-conditioned. A precise formulation of these conditions is found in [6].

Although common choices for $\mathbf{c}(\mathbf{z})$ and $\mathbf{f}(\mathbf{x})$ easily imply the smoothness of the manifolds $\mathbf{c}(\overline{\mathbf{p}}(X))$ and $\mathbf{f}(X)$, the success of MM-iteration hinges on the efficiency of the minimization for the coarse model problems and on the similarity between $\mathbf{c}(\mathbf{z})$ and $\mathbf{f}(\mathbf{x})$. However, in practice it is much harder to foresee the effect of requirement (iii). In fact, in the initial phase of the iteration we cannot give any a priori guarantee that the matrices ΔC and ΔF are well-conditioned. The problem is generally nonlinear and, because we want to reduce the number of iterations as much as possible, large steps $\mathbf{x}_k - \mathbf{x}_{k-i}$ 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 $\mathbf{x}_k - \mathbf{x}_{k-i}$, which reduce the feasibility of the assumption that ΔC and ΔF approximate the tangential planes to, respectively, the manifolds $\mathbf{c}(\overline{\mathbf{p}}(X))$ and $\mathbf{f}(X)$.

Therefore we have to introduce a trust-region strategy that controls the step-length in such a way that the unwanted effects do no harm and convergence is accelerated. In addition to the reduction of unwanted large step-sizes, we should take care of the possible ill-conditioning. Viz., the effect of an (almost) singular matrix ΔC would be that the solution is sought in only a lower-dimensional sub-manifold of $\mathbf{c}(\overline{\mathbf{p}}(X))$, and an almost singular matrix ΔF can lead to extremely large steps. Both effects are counteracted by regularization of the matrices. The generalized singular value decomposition (GSVD) for the pair $(\Delta C, \Delta F)$ will serve that purpose, instead of the SVD of ΔC and ΔF.

The GSVD of the matrices (A, B) is (cf. [7]) a set of five matrices U_A, U_B, Σ_A, Σ_B, and V, such that

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

with a regular matrix V, unitary matrices U_A, U_B, and diagonal matrices $\Sigma_A = \mathrm{diag}(\sigma_1^A, \ldots, \sigma_n^A)$, $\Sigma_B = \mathrm{diag}(\sigma_1^B, \ldots, \sigma_n^B)$. The matrix $\Delta C \Delta F^{\dagger}$ in (12) can now be written

$$\Delta C \Delta F^{\dagger} = U_c \Sigma_c V^{\mathrm{T}} (V^{\mathrm{T}})^{-1} \Sigma_f^{\dagger} U_f^{\mathrm{T}} = U_c \mathrm{diag}(\sigma_i^c/\sigma_i^f) U_f^{\mathrm{T}} \approx U_c \mathrm{diag}\left(\frac{\sigma_i^c + \lambda\sigma_1^c}{\sigma_i^f + \lambda\sigma_1^f}\right) U_f^{\mathrm{T}}, \quad \text{with } \lambda \geqslant 0.$$

Regularization is introduced by taking $\lambda > 0$. For vanishing λ no regularization takes place and for $\lambda \to \infty$ the matrix reduces to $U_c U_f^{\mathrm{T}} \sigma_1^c/\sigma_1^f$, which is close to the identity if $\mathbf{c}(\mathbf{z})$ and $\mathbf{f}(\mathbf{x})$ have similar behaviour.

The choice of the parameter λ in the trust-region MM-algorithm (TR-MM) is based on the success of the previous iteration steps. If an iteration step is successful (i.e., the residual decreases) the value of λ is divided by 2, otherwise λ is multiplied by 2. In practice, λ is never reduced below a tolerance value τ, the machine accuracy or the precision of the data (in the scheme introduced here: the residue squared). The resulting TR-MM algorithm is shown in Fig. 3. It should be noticed that the value of a few constants in the algorithm, like the factor two for the amplification or reduction of λ, is only based on heuristics. Further, the scheme contains an (optional) damping parameter $\delta \geqslant 0$. For strongly nonlinear problems it can stabilize the convergence process at the expense of additional function evaluations. We will see an example of its use in the next section.

$\mathbf{x}_0 = \text{argmin}_{\mathbf{x}\in X} \|\mathbf{c}(\mathbf{x}) - \mathbf{y}\|;$
$r_0 = \|\mathbf{f}(\mathbf{x}_0) - \mathbf{y}\|^2;$
$\mathbf{y}_0 = \mathbf{c}(\mathbf{x}_0) - (\mathbf{f}(\mathbf{x}_0) - \mathbf{y});$
$\lambda_0 = 1;$
for $k = 1,\ k \le M,\ k = k+1,$
  $\mathbf{x}_k = \text{argmin}_{\mathbf{x}\in X} \|\mathbf{c}(\mathbf{x}) - \mathbf{y}_k\|;$
  $s_k = \|\mathbf{x}_{k-1} - \mathbf{x}_k\|^2;$
  $r_k = \|\mathbf{f}(\mathbf{x}_k) - \mathbf{y}\|^2;$
  if $s_k < \tau$ **and** $\lambda_{k-1} < 1/8$ **then break endif**;
  while $r_k > r_{k-1}(1 + \tau(1+\lambda)),$
    $\mathbf{x}_k = \mathbf{x}_{k-1} + \tau^{1/4}\,(\mathbf{x}_k - \mathbf{x}_{k-1});$
    $s_k = \|\mathbf{x}_{k-1} - \mathbf{x}_k\|^2;$
    $r_k = \|\mathbf{f}(\mathbf{x}_k) - \mathbf{y}\|^2;$
    $\lambda_{k-1} = \max(8, 2\lambda_{k-1});$
  endwhile ;
  $\lambda_k = \max(\tau, \lambda_{k-1}/2);$
  $\Delta C = \Big((\mathbf{c}(\mathbf{x}_{k-i}) - \mathbf{c}(\mathbf{x}_k))_{i=1,\ldots,\min(k,n)}\Big);$
  $\Delta F = \Big((\mathbf{f}(\mathbf{x}_{k-i}) - \mathbf{f}(\mathbf{x}_k))_{i=1,\ldots,\min(k,n)}\Big);$
  $(U_f, U_c, \Sigma_f, \Sigma_c, V) = \text{GSVD}(\Delta F, \Delta C);$
  $D = \text{diag}\left(\frac{\sigma_i^c + \lambda_k(\sigma_1^c+\tau)}{\sigma_i^f + \lambda_k(\sigma_1^f+\tau)}\right);$
  $\mathbf{y}_k = \Big(\mathbf{c}(\mathbf{x}_k) - (U_c D U_f^T + I - U_c U_c^T)(\mathbf{f}(\mathbf{x}_k) - \mathbf{y})\Big)/(1 + \delta\lambda_k);$
endfor

Fig. 3. The trust-region manifold-mapping (TR-MM) algorithm.

7. Examples showing the effect of regularization

As an example of the use of regularization in the TR-MM algorithm, we first show its behaviour for a simple problem from [4]. In the four cases considered below, the same fine and coarse model are used, but different specifications $\mathbf{y}$ create essentially distinct situations (see [4, Section 5.1.2]): (i) reachable design, (ii) perfect mapping, (iii) non-perfect mapping, and (iv) close local minima.

The examples in this section use $\mathbf{f}(\mathbf{x}) = \mathbf{f}(x_1, x_2) = (x_1(x_2 t_i + 1)^2)_{i=1,2,3}$ and $\mathbf{c}(\mathbf{x}) = \mathbf{c}(x_1, x_2) = (x_1 t_i + x_2)_{i=1,2,3}$ with $t_i \in \{-1, 0, 1\}$. For the tolerance parameter τ we take $\tau = 10^{-12}$, we set the maximum number of iterations to $M = 100$, and the damping parameter is not used ($\delta = 0$), except for the last example. We emphasize that the MM-optimization method is not designed for such simple least squares problems, but for problems where the **f**-evaluation is quite expensive. Examples of such elaborate cases from practice can be found, e.g., in [8], and another example of practical relevance is shown in the last section of this paper. In the simple examples shown here, the coarse model is not specially adapted to the fine one, and particularly the last example shows what adverse effects can be expected in such cases. Nevertheless, these problems can be solved by manifold mapping.

(i) *Reachable design*: We take $\mathbf{y} = (0.081, 0.100, 0.121)$. This makes a reachable design with solution $\mathbf{x}^* = (0.1, 0.1)$. The problem is solved in 14 iterations (16 **f**-evaluations). In Fig. 4a we can see the iteration path. It shows level curves of $\|\mathbf{c}(\mathbf{x}) - \mathbf{y}\|$ (the horizontal ellipse) and of $\|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|$ (the more complex shapes). The dots represent the iteration steps, starting from $\mathbf{x}_0$, in the centre of the ellipse. Fig. 4b zooms in on the region near the fixed point of the iteration (the last 4 iterations and the **f**-level curves). Fig. 4c shows the convergence history (the logarithm of the residue). In Fig. 4d we see the logarithm of

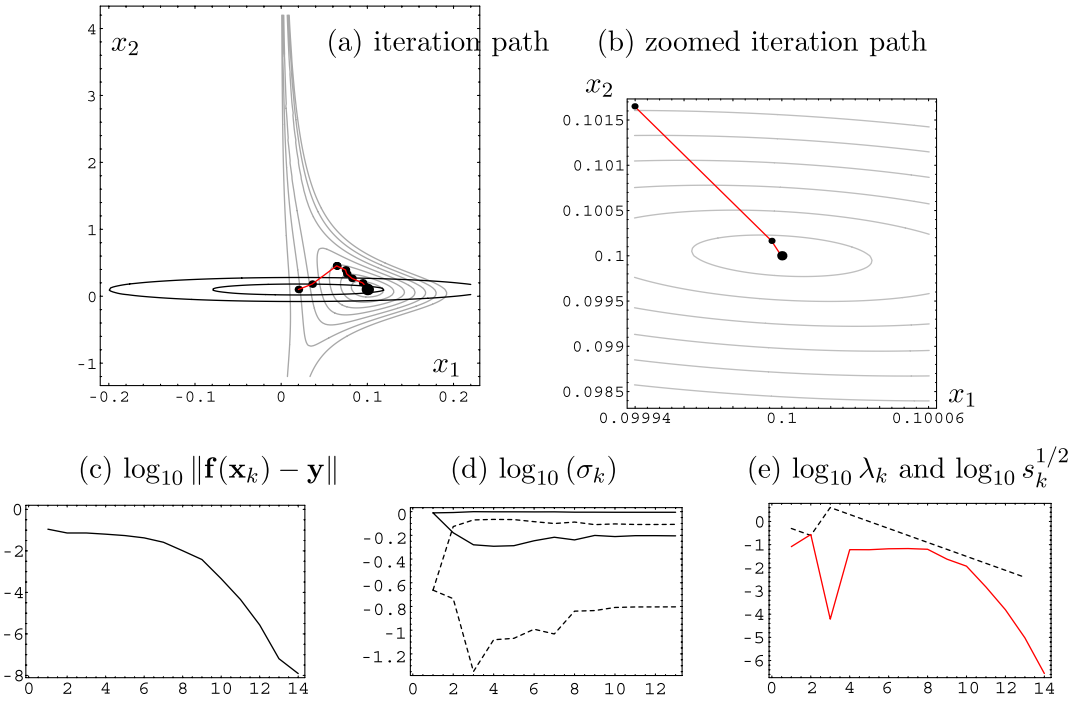


Fig. 4. Reachable design: solid line: step length $\log_{10}(s_k^{1/2})$; dashed line: $\log_{10}(\lambda_k)$.

the extreme singular values of ΔC (dashed) and ΔF (solid), and so, the condition numbers during iteration. Fig. 4e represents the history of λ_k (dashed) and of the step-length $\sqrt{s_k}$ (solid). The iteration stops because the step-length becomes small enough. In contrast with all other examples shown, for this simple problem the trust-region strategy has no positive effect. Without the trust-region strategy (i.e. $\lambda \equiv \tau$) the solution is obtained after 12 iterations.

(ii) *Perfect mapping*: Here we take $\mathbf{y} = (0.10011, 0.10125, 0.10241)$. This makes a residual norm of 5.5×10^{-6} at the solution $\mathbf{x}^* = (0.10125, 0.00568)$. The problem is solved in 13 iterations (15 **f**-evaluations). The iteration stops because the step-length becomes small enough. The convergence history is shown in Fig. 5. The behaviour is similar to that for the reachable design: first the parameter λ increases because of the initial dissimilarity between the fine and the coarse model and then we see a monotonous decrease until convergence is obtained. The matrices ΔC and ΔF remain well-conditioned. Without the trust-region strategy (i.e. $\lambda \equiv \tau$) the solution is obtained after 51 iterations.

(iii) *Non-perfect mapping*: We take $\mathbf{y} = (0.00, -0.40, 0.10)$. This makes a residual norm of 0.370 at the solution $\mathbf{x}^* = (-0.10069, -0.14121)$. The problem is solved in 32 iterations (44 **f**-evaluations). The iteration stops because the step-length becomes small enough. The convergence history is shown in Fig. 6. In this case ΔF becomes very ill-conditioned and during the iteration process λ has to be increased several times. Without the trust-region strategy (i.e. $\lambda = \tau$) the solution is obtained after 52 iterations.

(iv) *Nearby local minima*: We take $\mathbf{y} = (0.00, -0.35, 0.20)$. In this example the fine model has two local minima in the neighbourhood of the coarse-model minimum. The local behaviour of the fine model near the global minimum is much different from the behaviour of the coarse model, which is more similar to the other local minimum. We recognize this in the Figs. 7 and 8, which show the level curves for this problem. The local behaviour of $\|\mathbf{c}(\mathbf{x}) - \mathbf{y}\|$ is represented by the horizontal ellipses. The global minimum of $\|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|$ (value = 0.36320) is found at $\mathbf{x}^* = (0.00656, 4.00689)$, whereas the other local minimum (value = 0.38319) is seen at $\mathbf{x}^* = (-0.05887, -0.35221)$. In this example TR-MM without damping ($\delta = 0$, Fig. 7) converges to the second local minimum (*not* the global one). The problem is solved in

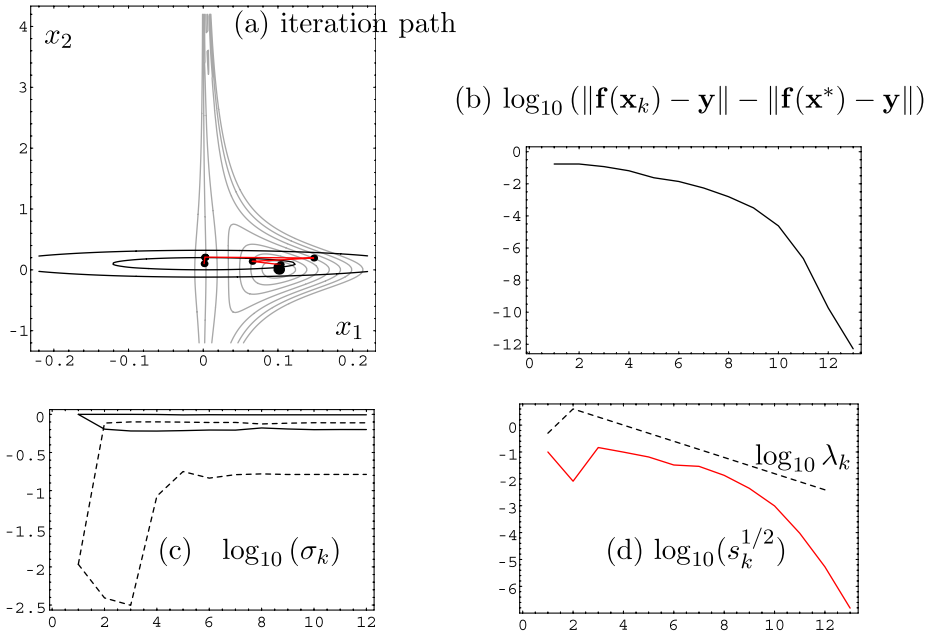


Fig. 5. Perfect mapping (see the legend of Fig. 4).

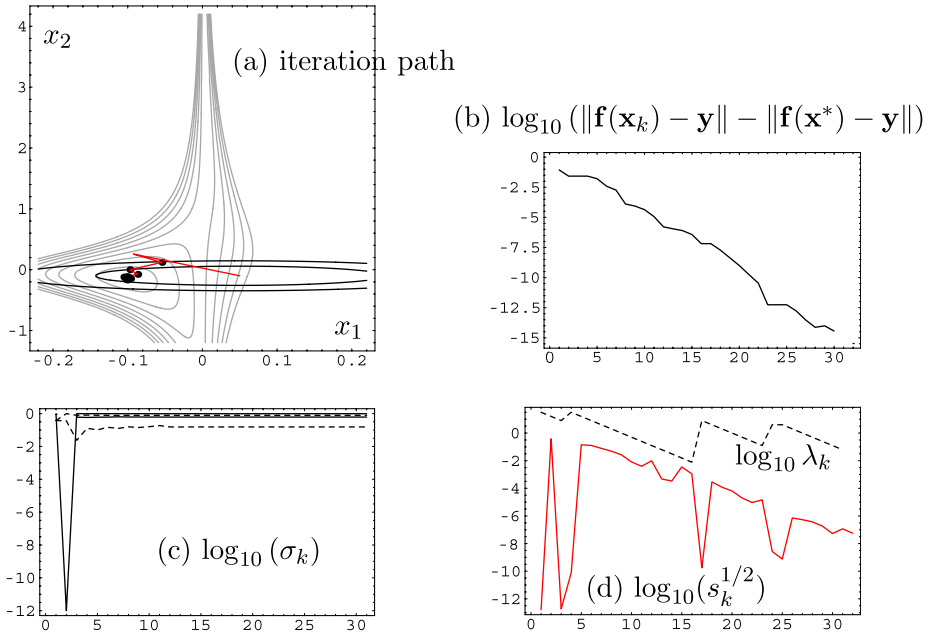


Fig. 6. Non-perfect mapping (see the legend of Fig. 4).

38 iterations (49 **f**-evaluations). The iteration stops because the step-length becomes small enough. The convergence history is shown in Fig. 7c. Without the trust-region strategy (i.e. $\lambda \equiv \tau$) the iteration process does not converge within $M = 100$ iterations.

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 λ. The iteration process arrives in the

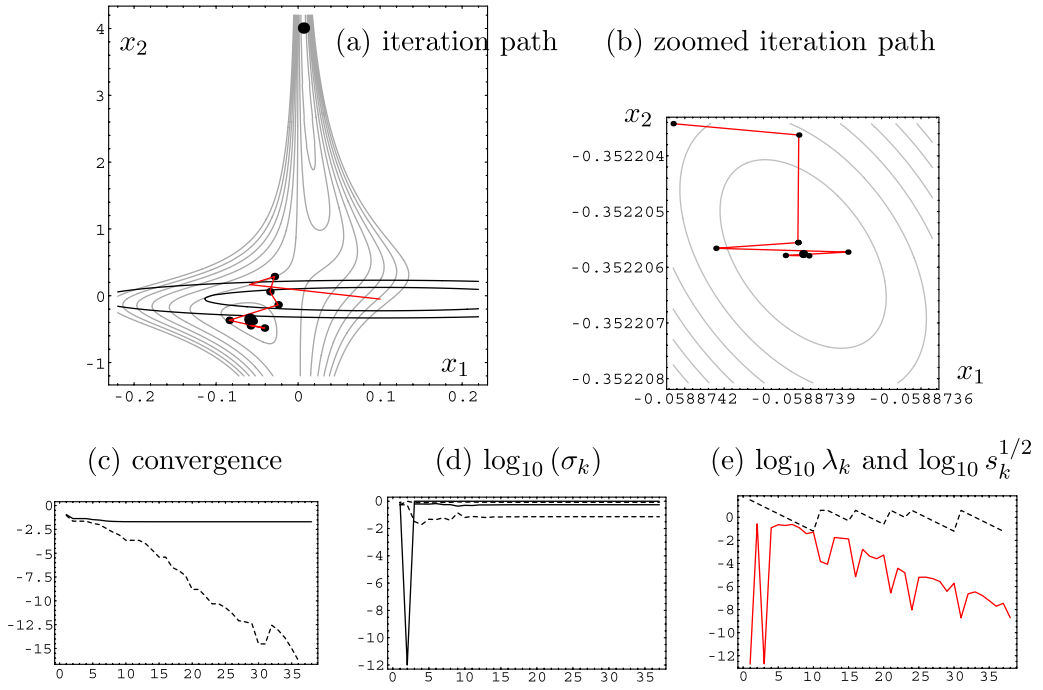


Fig. 7. Two nearby local minima, damping parameter $\delta = 0$ (see the legend of Fig. 8).

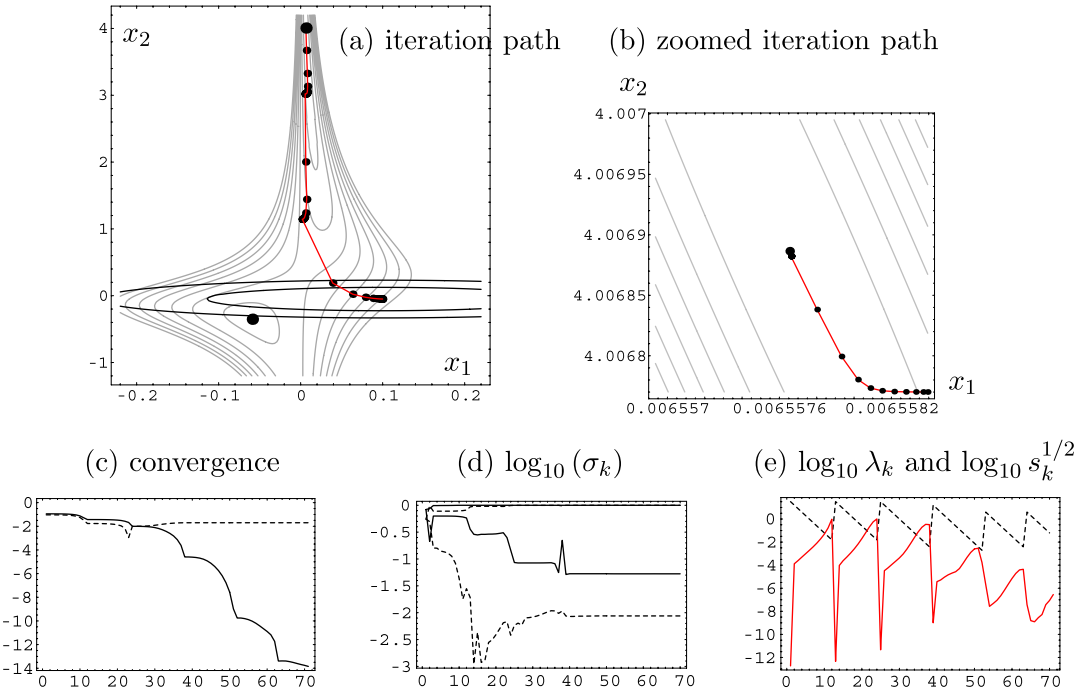
attraction area for the global minimum, but because of the large discrepancy between the behaviour of $\|\mathbf{c}(\mathbf{x}) - \mathbf{y}\|$ and $\|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|$ near the global minimum, the convergence is relatively slow. The convergence history is shown in Fig. 8.

8. An example from engineering

We study the optimization of a die press [9] used for manufacturing anisotropic permanent magnets. It is benchmark Problem 25 from the International Compumag Society, Testing Electromagnetic Analysis Methods (T.E.A.M.), www.compumag.co.uk/team.html. The molds and the pole are made of steel (see Fig. 9). The specifications $\mathbf{y}$ are certain magnetic flux values computed at ten points along the curve e–f in the cavity. The design variable $\mathbf{x} = [x_1, x_2, x_3, x_4]$ determines the geometry of the molds. The space of controls X is simply a polytope in $\mathbb{R}^4$. The full problem description can be found in [9].

The fine model $\mathbf{f}$ is based on simulation with second order triangular finite elements (around 120000 degrees of freedom). The cost function is $F(\mathbf{x}) = \|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|_2^2$. For the coarse model $\mathbf{c}$ we build a least squares quadratic approximation by means of sixteen finite element solutions (vertices of the polytope X) where the characteristics of the molds and the pole have been linearized. (These discretizations are rather coarse, yielding less than 1000 degrees of freedom). The computational cost associated with the construction and evaluation of the coarse model is negligible when compared with that for the fine model.

We show in Table 1 the results for the die press optimization. MM without regularization moves away from the optimum after the fourth iteration. The diverging tendency continues during the next iterations. The fine model optimum $\mathbf{x}^*$ is computed by TRMM in five iterations (eight equivalent fine model evaluations). We use differential evolution (DE) [10] for the coarse model optimum computation and for every minimization process within the manifold-mapping approach. The alternative method used in practice, SQP, applied with the coarse model optimum $\mathbf{z}^*$ as initial guess, performs much worse than TRMM.



(c): $\log_{10}(\|\mathbf{f}(\mathbf{x}_k)-\mathbf{y}\|-\|\mathbf{f}(\mathbf{x}^*)-\mathbf{y}\|)$, dashed: $\mathbf{x}^*$ global, solid: $\mathbf{x}^*$ local minimum,
(d): solid lines: $\sigma=\sigma^f_{\min}, \sigma^f_{\max}$; dashed lines: $\sigma=\sigma^c_{\min}, \sigma^c_{\max}$,
(e): solid line: step length $\log_{10}(s_k^{1/2})$; dashed line: $\log_{10}(\lambda_k)$.

Fig. 8. Two nearby local minima, damping parameter $\delta = 64$.

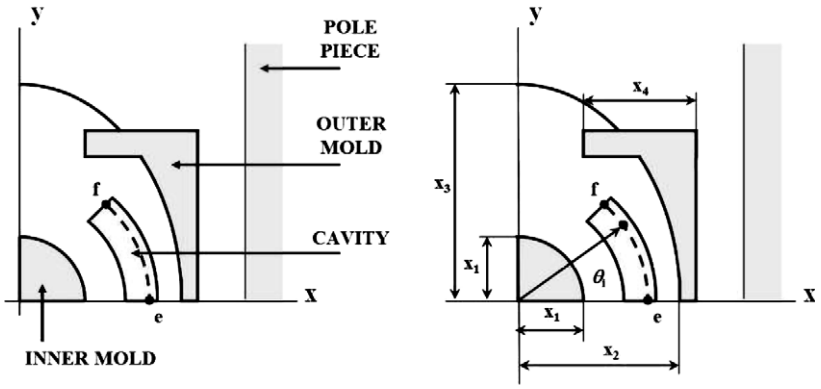


Fig. 9. Geometry and design variables of the die press.

9. Conclusion

In this paper we treat a two-level optimization method. Such methods use simple surrogate models to accelerate optimization processes in which the objective-function evaluation requires time-consuming computer simulations. We take the traditional space-mapping method as a starting point and we show how this method can be improved to obtain the manifold-mapping algorithm. Further, we present a new trust-region strategy fitted for this manifold method and we give a number of examples to show its behaviour.

Table 1
Iteration history in the die-press optimization problem

# Iteration	SQP		MM		TRMM	
	# f evals	$F(\mathbf{x})$	# f evals	$F(\mathbf{x})$	# f evals	$F(\mathbf{x})$
1	11	0.0016	1	0.0017	1	0.0018
2	17	0.0005	2	0.0029	3	0.0013
3	23	0.0005	3	0.0007	4	0.0007
4	29	0.0005	4	0.0008	7	0.0007
5	35	0.0005	5	0.0013	8	0.0004

SQP was applied with the coarse model optimum $\mathbf{z}^*$ as initial guess. The field **# f evals** denotes the cumulative number of equivalent fine model evaluations (approximately proportional to computing time). The field $F(\mathbf{x})$ indicates the cost function.

Acknowledgement

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

References

[1] J. Bandler, R. Biernacki, S. Chen, P. Grobelny, R.H. Hemmers, Space mapping technique for electromagnetic optimization, IEEE Trans. Microwave Theory Tech 42 (1994) 2536–2544.
[2] M. Bakr, J. Bandler, K. Madsen, J. Søndergaard, Review of the space-mapping approach to engineering optimization and modeling, Opt. Eng. 1 (3) (2000) 241–276.
[3] J. Bandler, Q. Cheng, A. Dakroury, A. Mohamed, M. Bakr, K. Madsen, J. Søndergaard, Space mapping: The state of the art, IEEE Trans. Microwave Theory Tech. 52 (2004) 337–360.
[4] D. Echeverría, P. Hemker, Space mapping and defect correction, Comp. Methods Appl. Math. 5 (2) (2005) 107–136.
[5] K. Böhmer, P.W. Hemker, H.J. Stetter, The defect correction approach, in: K. Böhmer, H.J. Stetter (Eds.), Defect Correction Methods: Theory and Applications, Computing, Suppl. 5, Springer-Verlag, Heidelberg, Berlin, New York, Tokyo, 1984, pp. 1–32.
[6] D. Echeverría, P. Hemker, On the manifold-mapping optimization technique, Tech. Rep. MAS-E0612, CWI, submitted for publication.
[7] G.H. Golub, C.F.V. Loan, Matrix Computations, North Oxford Academic, 1983.
[8] D. Echeverría, D. Lahaye, L. Encica, E.A. Lomonova, P.W. Hemker, A.J.A. Vandenput, Manifold mapping optimization applied to linear actuator design, IEEE Trans. Magnet. 42 (4) (2006) 1183–1186.
[9] N. Takahashi, K. Ebihara, K. Yoshida, T. Nakata, K. Ohashi, K. Miyata, Investigation of simulated annealing method and its application to optimal design of die mold for orientation of magnetic powder, IEEE Trans. Magnet. 32 (3) (1996) 1210–1213.
[10] R. Storn, K. Price, Differential Evolution – A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces, Tech. Rep. TR-95-012, International Computer Science Institute (ICSI), Berkeley, California, US (March 1995).