Two-level Optimization

"Manifold Mapping"

P.W. Hemker

Bruchsal 2013

D. Echeverría and P.W. Hemker
Manifold-Mapping: a Two-Level Optimization Technique.
Dedicated to Professor Wolfgang Hackbusch on the occasion of his 60th birthday
Comput. Visual. Sci. 11:193–206, 2008.

five years old, but sufficiently interesting/unknown

References

- D. Echeverría and P.W. Hemker.
  Space mapping and defect correction.
  Comp. Methods in Appl. Math., 5(2):107–136, 2005.
- D. Echeverría and P.W. Hemker.
  A trust-region strategy for manifold-mapping optimization.
  J. Comp. Phys. 224:464–475, 2007.

download these slides from: `piet.hemker.nl`

Defect correction principle

equation to solve: $\mathcal{F}\,\mathbf{x} = \mathbf{y},$

simplified version: $\widetilde{\mathcal{F}}\,\mathbf{x} = \mathbf{y},$

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

or with $\widetilde{\mathcal{G}} = \widetilde{\mathcal{F}}^{-1}$

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

$\widetilde{\mathcal{G}}$ injective $\rightarrow$ $\mathcal{F}\,\mathbf{x} = \mathbf{y}$

Defect correction principle

equation to solve: $\mathcal{F}\,\mathbf{x} = \mathbf{y}$,

simplified version: $\widetilde{\mathcal{F}}\,\mathbf{x} = \mathbf{y}$,

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

or with $\widetilde{\mathcal{G}} = \widetilde{\mathcal{F}}^{-1}$

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

$\widetilde{\mathcal{G}}$ injective $\rightarrow$ $\mathcal{F}\,\mathbf{x} = \mathbf{y}$

Defect correction principle

equation to solve: $\mathcal{F}\,\mathbf{x} = \mathbf{y}$,
simplified version: $\widetilde{\mathcal{F}}\,\mathbf{x} = \mathbf{y}$,

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

or with $\widetilde{\mathcal{G}} = \widetilde{\mathcal{F}}^{-1}$

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

$\widetilde{\mathcal{G}}$ injective $\rightarrow$ $\mathcal{F}\,\mathbf{x} = \mathbf{y}$

Defect correction principle

equation to solve: $\mathcal{F}\,\mathbf{x} = \mathbf{y}$,
simplified version: $\widetilde{\mathcal{F}}\,\mathbf{x} = \mathbf{y}$,

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

or with $\widetilde{\mathcal{G}} = \widetilde{\mathcal{F}}^{-1}$

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

$\widetilde{\mathcal{G}}$ injective $\rightarrow$ $\mathcal{F}\,\mathbf{x} = \mathbf{y}$

Defect correction principle

equation to solve: $\mathcal{F}\,\mathbf{x} = \mathbf{y},$
simplified version: $\widetilde{\mathcal{F}}\,\mathbf{x} = \mathbf{y},$

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

or with $\widetilde{\mathcal{G}} = \widetilde{\mathcal{F}}^{-1}$

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

$\widetilde{\mathcal{G}}$ injective $\rightarrow$ $\mathcal{F}\,\mathbf{x} = \mathbf{y}$

Defect correction principle

equation to solve: $\mathcal{F}\,\mathbf{x} = \mathbf{y},$

simplified version: $\widetilde{\mathcal{F}}\,\mathbf{x} = \mathbf{y},$

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

or with $\widetilde{\mathcal{G}} = \widetilde{\mathcal{F}}^{-1}$

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

$\widetilde{\mathcal{G}}$ injective $\rightarrow$ $\mathcal{F}\,\mathbf{x} = \mathbf{y}$

Defect correction principle

equation to solve: $\mathcal{F}\,\mathbf{x} = \mathbf{y}$,
simplified version: $\widetilde{\mathcal{F}}\,\mathbf{x} = \mathbf{y}$,

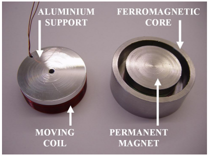
$$\begin{cases} \widetilde{\mathcal{F}}\mathbf{x}_0 & = & \mathbf{y}\,, \\ \widetilde{\mathcal{F}}\mathbf{x}_{k+1} & = & \widetilde{\mathcal{F}}\,\mathbf{x}_k - \mathcal{F}\,\mathbf{x}_k + \mathbf{y}\,. \end{cases}$$

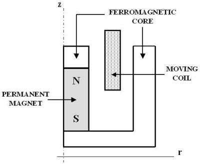
or with $\widetilde{\mathcal{G}} = \widetilde{\mathcal{F}}^{-1}$

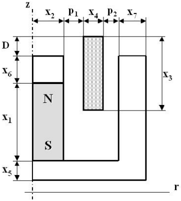
$$\begin{cases} \mathbf{x}_0 & = & \widetilde{\mathcal{G}}\,\mathbf{y}\,, \\ \mathbf{x}_{k+1} & = & \mathbf{x}_k - \widetilde{\mathcal{G}}\,\mathcal{F}\,\mathbf{x}_k + \widetilde{\mathcal{G}}\,\mathbf{y}, \end{cases}$$

$\widetilde{\mathcal{G}}$ injective $\quad\rightarrow\quad \mathcal{F}\,\mathbf{x} = \mathbf{y}$

Optimization problem







determine x_1, x_2, x_1, x_3, x_4, x_5, x_6, x_7

in optimization:

– function evaluations may be extremely costly

– inaccurate but fast approximations may be available

$\Rightarrow$ can the fast accelerate the accurate ?

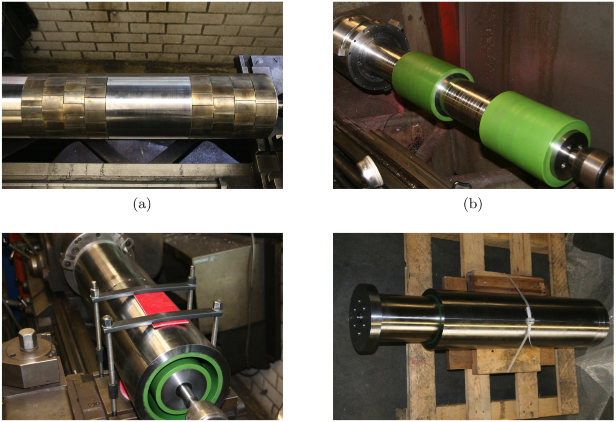
claim by Bandler's "space mapping" technique (1994)
in engineering literature

Survey paper:

John W. Bandler et al.: *Space Mapping: The State of the Art*

IEEE Transactions On Microwave Theory And Techniques, Vol. 52, No. 1, 2004

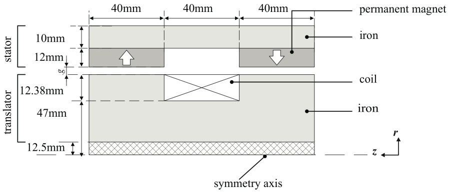
Example: ELMASP prototype



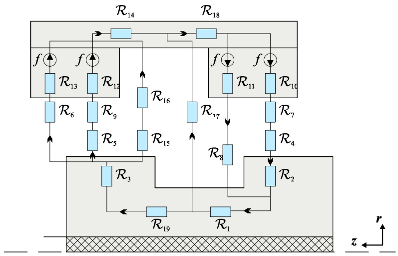
(a) (b) (c) (d)

Two models for an electric device (ELMASP shock absorber)

determine optimal dimensions



Geometric model

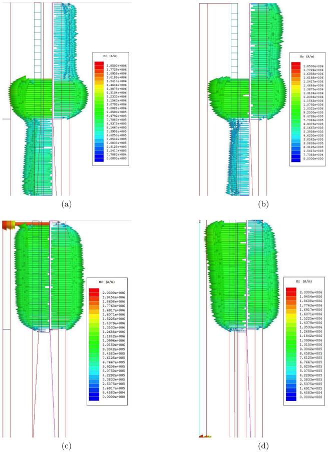


MEC model (Magnetic Equivalent Circuit)

⇕

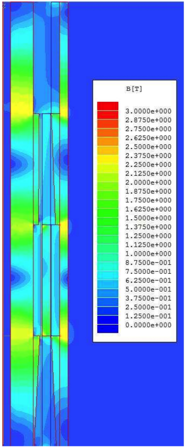
PDE model

Example: ELMASP prototype



magnetic field

FEM computation



magnetic flux

Specification of the aim: $\mathbf{y}$

Fine model: $\mathbf{f}(\mathbf{x})$ find design $\mathbf{x}^*$: $\mathbf{f}(\mathbf{x}^*) \approx \mathbf{y}$

Coarse model: $\mathbf{c}(\mathbf{z})$ find design $\mathbf{z}^*$: $\mathbf{c}(\mathbf{z}^*) \approx \mathbf{y}$

$\mathbf{f}(\mathbf{x}^*)$ gives a much more accurate result

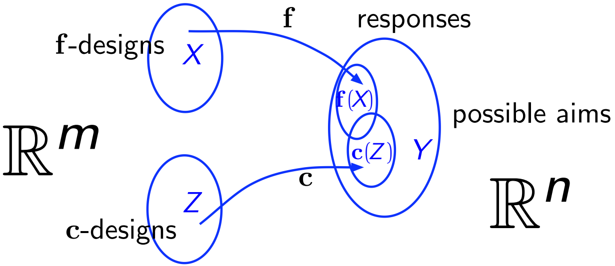
$\mathbf{c}(\mathbf{z}^*)$ is much simpler to evaluate

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

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

much simpler to solve

What aims $\mathbf{y}$ are reachable?



$\mathbf{y} \in \mathbf{f}(X)$: $\mathbf{y}$ is a fine-model reachable aim:
$\mathbf{y} \in \mathbf{c}(Z)$: $\mathbf{y}$ is a coarse-model reachable aim:

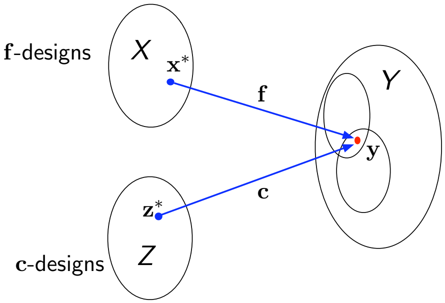
$$\dim(X) = \dim(Z) = m \leq n = \dim(Y)$$

Reachable $\mathbf{y}$?

$\mathbf{y}$ reachable $\Rightarrow$ equation

$\mathbf{y}$ non-reachable $\Rightarrow$ optimization problem

Reachable aim $\mathbf{y}$

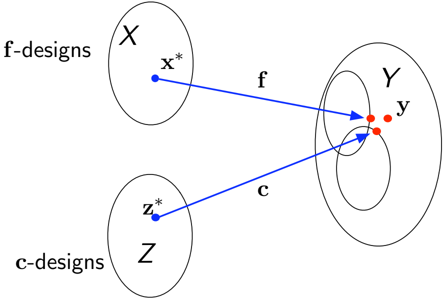


$$\mathbf{f}(\mathbf{x}^*) = \mathbf{y}$$

$$\mathbf{c}(\mathbf{z}^*) = \mathbf{y}$$

solution of an equation

Non-reachable aim $\mathbf{y}$

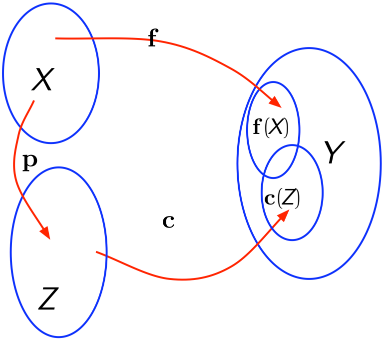


$$\mathbf{x}^* = \underset{\xi}{\operatorname{argmin}} \; \|\mathbf{f}(\xi) - \mathbf{y}\|$$

$$\mathbf{z}^* = \underset{\zeta}{\operatorname{argmin}} \; \|\mathbf{c}(\zeta) - \mathbf{y}\|$$

optimization problem

Space mapping principle



$$\mathbf{c}(\mathbf{p}(\mathbf{x}^*)) \approx \mathbf{y}$$

$$\mathbf{x}_d = \operatorname*{argmin}_{\mathbf{x}} \|\mathbf{c}(\mathbf{p}(\mathbf{x})) - \mathbf{y}\|$$

$\mathbf{c} \circ \mathbf{p}$ 'surrogate model'

space mapping method tries to find **the best possible** p

Defect correction for equations

equation to solve: $\mathcal{F}\,\mathbf{x} = \mathbf{y}$,
simplified version: $\widetilde{\mathcal{F}}\,\mathbf{x} = \mathbf{y}$,

we use

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

we use not with $\widetilde{\mathcal{G}} = \widetilde{\mathcal{F}}^{-1}$

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

Defect correction for equations

equation to solve: $\mathcal{F}\,\mathbf{x} = \mathbf{y},$

simplified version: $\widetilde{\mathcal{F}}\,\mathbf{x} = \mathbf{y},$

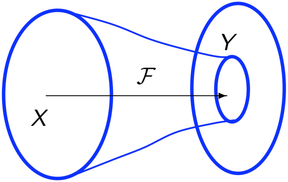
we use

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

that is with $\widetilde{\mathcal{G}} = \widetilde{\mathcal{F}}^{-1}$

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

equation solving $\leftrightarrow$ optimization



$\mathcal{F}$ injective

$\mathcal{F}$ not surjective $\rightarrow$ left-inverse $\mathcal{G}$ $\qquad \mathcal{G}\mathcal{F} = I_X$

$$\mathcal{F}\mathbf{x} = \mathbf{y} \quad \Leftrightarrow \quad \mathbf{f}(\mathbf{x}) = \mathbf{y}$$
$$\mathbf{x} = \mathcal{G}\mathbf{y} \quad \Leftrightarrow \quad \mathbf{x} = \operatorname*{argmin}_{\xi} \|\mathbf{f}(\xi) - \mathbf{y}\|$$

$$\widetilde{\mathcal{F}}\mathbf{x} = \mathbf{y} \quad \Leftrightarrow \quad \mathbf{c}(\mathbf{p}(\mathbf{x})) = \mathbf{y}$$
$$\mathbf{x} = \widetilde{\mathcal{G}}\mathbf{y} \quad \Leftrightarrow \quad \mathbf{x} = \operatorname*{argmin}_{\xi} \|\mathbf{c}(\mathbf{p}(\xi)) - \mathbf{y}\|$$

Iteration (space mapping)

$$\mathbf{z}^* = \operatorname{argmin}_\zeta \|\mathbf{c}(\zeta) - \mathbf{y}\|$$
$$\mathbf{x}_0 = \mathbf{p}^{-1}(\mathbf{z}^*)$$

$$\mathbf{x}_0 = \operatorname*{argmin}_\xi \|\mathbf{c}(\mathbf{p}(\xi)) - \mathbf{y}\|$$
$$\mathbf{x}_{k+1} = \operatorname*{argmin}_\xi \|\mathbf{c}(\mathbf{p}(\xi)) - \underbrace{\underbrace{\mathbf{c}(\mathbf{p}(\mathbf{x}_k)) + \mathbf{f}(\mathbf{x}_k)}\, - \mathbf{y}}\|$$

Iteration (space mapping)

$$\mathbf{z}^* = \operatorname{argmin}_\zeta \|\mathbf{c}(\zeta) - \mathbf{y}\|$$
$$\mathbf{x}_0 = \mathbf{p}^{-1}(\mathbf{z}^*)$$

$$\mathbf{x}_0 = \operatorname*{argmin}_\xi \|\mathbf{c}(\mathbf{p}(\xi)) - \mathbf{y}\|$$
$$\mathbf{x}_{k+1} = \operatorname*{argmin}_\xi \|\mathbf{c}(\mathbf{p}(\xi)) - \underbrace{\underbrace{\mathbf{c}(\mathbf{p}(\mathbf{x}_k)) + \mathbf{f}(\mathbf{x}_k)}_{} - \mathbf{y}}_{} \|$$

Iteration (space mapping)

$$\mathbf{z}^* = \operatorname{argmin}_\zeta \|\mathbf{c}(\zeta) - \mathbf{y}\|$$
$$\mathbf{x}_0 = \mathbf{p}^{-1}(\mathbf{z}^*)$$

$$\mathbf{x}_0 = \operatorname*{argmin}_\xi \|\mathbf{c}(\mathbf{p}(\xi)) - \mathbf{y}\|$$
$$\mathbf{x}_{k+1} = \operatorname*{argmin}_\xi \|\mathbf{c}(\mathbf{p}(\xi)) - \underbrace{\underbrace{\mathbf{c}(\mathbf{p}(\mathbf{x}_k)) + \mathbf{f}(\mathbf{x}_k)}\, - \mathbf{y}}\|$$

Iteration (space mapping)

$$\mathbf{z}^* = \text{argmin}_\zeta \|\mathbf{c}(\zeta) - \mathbf{y}\|$$
$$\mathbf{x}_0 = \mathbf{p}^{-1}(\mathbf{z}^*)$$

$$\mathbf{x}_0 = \underset{\xi}{\text{argmin}} \|\mathbf{c}(\mathbf{p}(\xi)) - \mathbf{y}\|$$
$$\mathbf{x}_{k+1} = \underset{\xi}{\text{argmin}} \|\mathbf{c}(\mathbf{p}(\xi)) - \underbrace{\underbrace{\mathbf{c}(\mathbf{p}(\mathbf{x}_k)) + \mathbf{f}(\mathbf{x}_k)}\, - \mathbf{y}}\,\|$$

Iteration (space mapping)

$$\mathbf{z}^* = \operatorname{argmin}_\zeta \|\mathbf{c}(\zeta) - \mathbf{y}\|$$
$$\mathbf{x}_0 = \mathbf{p}^{-1}(\mathbf{z}^*)$$

$$\mathbf{x}_0 = \operatorname*{argmin}_\xi \|\mathbf{c}(\mathbf{p}(\xi)) - \mathbf{y}\|$$
$$\mathbf{x}_{k+1} = \operatorname*{argmin}_\xi \|\mathbf{c}(\mathbf{p}(\xi)) - \underbrace{\underbrace{\mathbf{c}(\mathbf{p}(\mathbf{x}_k)) + \mathbf{f}(\mathbf{x}_k)} - \mathbf{y}}\|$$

Orthogonality relations

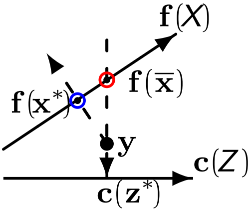
Convergence $\Rightarrow \quad \overline{\mathbf{x}} = \lim_{k\to\infty} \mathbf{x}_k$

$\overline{\mathbf{x}}$ is **not** the true solution

$\Rightarrow \quad \mathbf{f}(\overline{\mathbf{x}}) - \mathbf{y} \in \quad \mathbf{c}(Z)^{\perp}(\mathbf{z}^*)$

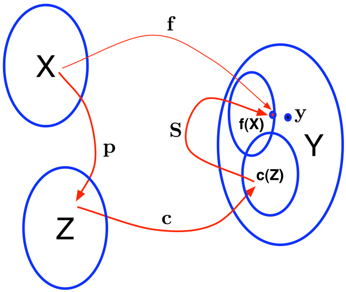
instead of

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



'space mapping' doesn't give the right answer !

We need an additional mapping, $\mathbf{S}$:



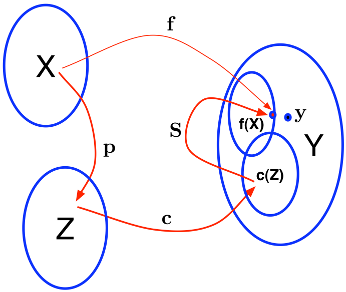
$$\mathbf{S} : \mathbf{c}(Z) \to \mathbf{f}(X)$$

$\mathbf{S}$ affine mapping
such that near solution:

$$\mathbf{f}(X) \parallel \mathbf{S}(\mathbf{c}(Z))$$

$\mathbf{S}$: ‘manifold mapping’
$\mathbf{p}$: not important

We need an additional mapping, $\mathbf{S}$:



$$\mathbf{S} : \mathbf{c}(Z) \to \mathbf{f}(X)$$

$\mathbf{S}$ affine mapping
such that near solution:

$$\mathbf{f}(X) \parallel \mathbf{S}(\mathbf{c}(Z))$$

THIS IS THE KEY !

$\mathbf{S}$: 'manifold mapping'
$\mathbf{p}$: not important

The iteration

$$\mathbf{x}_0 = \operatorname*{argmin}_{\xi} \|\mathbf{c}(\mathbf{p}(\xi)) - \mathbf{y}\|$$

$$\mathbf{S}_0 = \mathbf{I}$$

$$\mathbf{x}_{k+1} = \operatorname*{argmin}_{\xi} \|\mathbf{S}_k(\mathbf{c}(\mathbf{p}(\xi))) - \mathbf{S}_k(\mathbf{c}(\mathbf{p}(\mathbf{x}_k))) + \mathbf{f}(\mathbf{x}_k) - \mathbf{y}\|$$

$\mathbf{S}_k :$ $\mathbf{c}(Z) \to \mathbf{f}(X)$

should approximate

$$\mathbf{f}(X) \parallel \mathbf{S}_k(\mathbf{c}(Z))$$

Conclusion

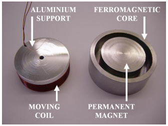
$\mathbf{p}$ can be an arbitrary bijection !

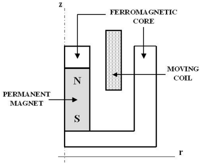
$\mathbf{S} = \lim_{k\to\infty} \mathbf{S}_k$ can be constructed during iteration

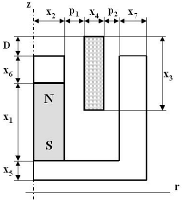
Theorem: $\lim_{k\to\infty} \mathbf{x}_k = \mathbf{x}^*$

**In contrast with space-mapping,
manifold mapping converges to the true solution !**

Typical application







determine x_1, x_2, x_1, x_3, x_4, x_5, x_6, x_7

7 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]
MM	6	81.11 g	[8.500, 9.784, 11.452, 1.883, 3.860, 3.202, 2.515]

computational work expressed in $\mathbf{f}$ evaluations

The initial guess for SQP is the coarse model optimum $\mathbf{z}^*$

Manifold Mapping (summary)

special case of defect correction

$$\mathcal{F}(\mathbf{x}) = \mathbf{f}(\mathbf{x})$$ injection (no surjection)

$$\mathcal{G}(\mathbf{y}) = \mathrm{argmin}_\xi \ \|\mathbf{f}(\xi) - \mathbf{y}\|$$ leftinverse

$$\widetilde{\mathcal{F}}_k(\mathbf{x}) = \mathbf{S}_k(\mathbf{c}(\mathbf{p}(\mathbf{x})))$$ approximates $\mathcal{F}(\mathbf{x})$

$$\widetilde{\mathcal{G}}_k(\mathbf{y}) = \mathrm{argmin}_\xi \ \|\mathbf{S}_k(\mathbf{c}(\mathbf{p}(\xi))) - \mathbf{y}\|$$

$\mathbf{S}_k$ computed from $\{\mathbf{x}_j\}_{j=0,1,\ldots,k}$

Theorem:

$\lim_{k\to\infty} \mathbf{S}_k = \mathbf{S}$

$\lim_{k\to\infty} \mathbf{x}_k = \mathbf{x}^*$

$\mathbf{S} : \mathbf{c}(Z) \to \mathbf{f}(X)$

should satisfy

$\mathbf{f}(X) \parallel \mathbf{S}(\mathbf{c}(Z))$

The iteration

$$\mathbf{x}_0 = \underset{\xi}{\operatorname{argmin}} \, \|\mathbf{c}(\mathbf{p}(\xi)) - \mathbf{y}\|$$

$$\mathbf{S}_0 = \mathbf{I}$$

$$\mathbf{x}_{k+1} = \underset{\xi}{\operatorname{argmin}} \, \|\mathbf{S}_k(\mathbf{c}(\mathbf{p}(\xi))) - \mathbf{S}_k(\mathbf{c}(\mathbf{p}(\mathbf{x}_k))) + \mathbf{f}(\mathbf{x}_k) - \mathbf{y}\|$$

$$\mathbf{S}_k$$

such that $\mathbf{S}_k \Delta C_k = \Delta F_k$

with $\Delta C_k = (\mathbf{c}(\mathbf{p}(\mathbf{x}_k)) - \mathbf{c}(\mathbf{p}(\mathbf{x}_j)), \cdots)$

with $\Delta F_k = (\mathbf{f}(\mathbf{x}_k) - \mathbf{f}(\mathbf{x}_j), \cdots)$ $\quad j = k-1, k-2, \ldots$

$$\mathbf{S}_k^\dagger = \Delta C_k \Delta F_k^\dagger$$

regularization may be needed

The iteration

$$\mathbf{x}_0 = \underset{\xi}{\operatorname{argmin}} \|\mathbf{c}(\mathbf{p}(\xi)) - \mathbf{y}\|$$

$$\mathbf{S}_0 = \mathbf{I}$$

$$\mathbf{x}_{k+1} = \underset{\xi}{\operatorname{argmin}} \|\mathbf{c}(\mathbf{p}(\xi)) - \mathbf{c}(\mathbf{p}(\mathbf{x}_k)) + \mathbf{S}_k^{\dagger}(\mathbf{f}(\mathbf{x}_k) - \mathbf{y})\|$$

$$\mathbf{S}_k$$

such that $\mathbf{S}_k \Delta C_k = \Delta F_k$

with $\Delta C_k = (\mathbf{c}(\mathbf{p}(\mathbf{x}_k)) - \mathbf{c}(\mathbf{p}(\mathbf{x}_j)), \cdots)$

with $\Delta F_k = (\mathbf{f}(\mathbf{x}_k) - \mathbf{f}(\mathbf{x}_j), \cdots)$ $\quad j = k-1, k-2, \ldots$

$$\mathbf{S}_k^{\dagger} = \Delta C_k \Delta F_k^{\dagger}$$

regularization may be needed

The iteration

$$\mathbf{x}_0 = \underset{\xi}{\operatorname{argmin}} \|\mathbf{c}(\mathbf{p}(\xi))) - \mathbf{y}\|$$

$$\mathbf{S}_0 = \mathbf{I}$$

$$\mathbf{x}_{k+1} = \underset{\xi}{\operatorname{argmin}} \|\mathbf{c}(\mathbf{p}(\xi)) - \mathbf{c}(\mathbf{p}(\mathbf{x}_k)) + \mathbf{S}_k^{\dagger}(\mathbf{f}(\mathbf{x}_k) - \mathbf{y})\|$$

$\mathbf{S}_k$

such that $\mathbf{S}_k \Delta C_k = \Delta F_k$

with $\Delta C_k = (\mathbf{c}(\mathbf{p}(\mathbf{x}_k)) - \mathbf{c}(\mathbf{p}(\mathbf{x}_j)), \cdots)$

with $\Delta F_k = (\mathbf{f}(\mathbf{x}_k) - \mathbf{f}(\mathbf{x}_j), \cdots)$ $\quad j = k-1, k-2, \ldots$

$$\mathbf{S}_k^{\dagger} = \Delta C_k \Delta F_k^{\dagger}$$

regularization may be needed

The iteration

$$\mathbf{x}_0 = \underset{\xi}{\operatorname{argmin}} \|\mathbf{c}(\mathbf{p}(\xi)) - \mathbf{y}\|$$

$$\mathbf{S}_0 = \mathbf{I}$$

$$\mathbf{x}_{k+1} = \underset{\xi}{\operatorname{argmin}} \|\mathbf{c}(\mathbf{p}(\xi)) - \mathbf{c}(\mathbf{p}(\mathbf{x}_k)) + \mathbf{S}_k^\dagger(\mathbf{f}(\mathbf{x}_k) - \mathbf{y})\|$$

$$\mathbf{S}_k$$

such that $\mathbf{S}_k \Delta C_k = \Delta F_k$

with $\Delta C_k = (\mathbf{c}(\mathbf{p}(\mathbf{x}_k)) - \mathbf{c}(\mathbf{p}(\mathbf{x}_j)), \cdots)$

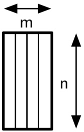
with $\Delta F_k = (\mathbf{f}(\mathbf{x}_k) - \mathbf{f}(\mathbf{x}_j), \cdots)$ $j = k-1, k-2, \ldots$

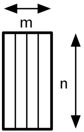
$$\mathbf{S}_k^\dagger = \Delta C_k \Delta F_k^\dagger$$

regularization may be needed

Regularization via GSVD

(Generalized Singular Value Decomposition)

$\Delta C =$ 

$\Delta F =$ 

$$\mathrm{GSVD}(\Delta C, \Delta F) \Rightarrow \{U_c, U_f, \Sigma_c, \Sigma_f, V\}$$

such that

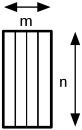
$$\Delta C = U_c \ \Sigma_c \ V$$
$$\Delta F = U_f \ \Sigma_f \ V$$

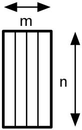
$$S_k^\dagger = \Delta C_k \Delta F_k^\dagger \quad = U_c \ \Sigma_c \ \Sigma_f^\dagger \ U_f^T \quad = U_c \ \mathrm{diag}\left(\frac{\sigma_{c,i} + \lambda \ \sigma_{c,1}}{\sigma_{f,i} + \lambda \ \sigma_{f,1}}\right) \ U_f^T$$

regularization !

Regularization via GSVD

(Generalized Singular Value Decomposition)

$\Delta C =$ 

$\Delta F =$ 

$$\mathrm{GSVD}(\Delta C, \Delta F) \Rightarrow \{U_c, U_f, \Sigma_c, \Sigma_f, V\}$$

such that

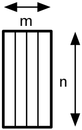
$$\Delta C = U_c \; \Sigma_c \; V$$
$$\Delta F = U_f \; \Sigma_f \; V$$

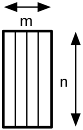
$$S_k^\dagger = \Delta C_k \Delta F_k^\dagger \quad = U_c \; \Sigma_c \; \Sigma_f^\dagger \; U_f^T \quad = U_c \; \mathrm{diag}\left(\frac{\sigma_{c,i} + \lambda \; \sigma_{c,1}}{\sigma_{f,i} + \lambda \; \sigma_{f,1}}\right) \; U_f^T$$

regularization !

Regularization via GSVD

(Generalized Singular Value Decomposition)

$\Delta C =$ 

$\Delta F =$ 

$$\mathrm{GSVD}(\Delta C, \Delta F) \Rightarrow \{U_c, U_f, \Sigma_c, \Sigma_f, V\}$$

such that

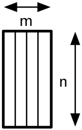
$$\Delta C = U_c \; \Sigma_c \; V$$
$$\Delta F = U_f \; \Sigma_f \; V$$

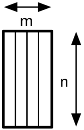
$$S_k^\dagger = \Delta C_k \Delta F_k^\dagger \quad = U_c \; \Sigma_c \; \Sigma_f^\dagger \; U_f^T \quad = U_c \; \mathrm{diag}\left(\frac{\sigma_{c,i}+\lambda \; \sigma_{c,1}}{\sigma_{f,i}+\lambda \; \sigma_{f,1}}\right) \; U_f^T$$

regularization !

Regularization via GSVD

(Generalized Singular Value Decomposition)

$\Delta C =$ 

$\Delta F =$ 

$$\mathrm{GSVD}(\Delta C, \Delta F) \Rightarrow \{U_c, U_f, \Sigma_c, \Sigma_f, V\}$$

such that

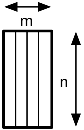
$$\Delta C = U_c\ \Sigma_c\ V$$
$$\Delta F = U_f\ \Sigma_f\ V$$

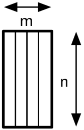
$$S_k^{\dagger} = \Delta C_k \Delta F_k^{\dagger} \quad = U_c\ \Sigma_c\ \Sigma_f^{\dagger}\ U_f^T \quad = U_c\ \mathrm{diag}\left(\frac{\sigma_{c,i} + \lambda\ \sigma_{c,1}}{\sigma_{f,i} + \lambda\ \sigma_{f,1}}\right)\ U_f^T$$

regularization !

Regularization via GSVD

(Generalized Singular Value Decomposition)

$\Delta C =$ 

$\Delta F =$ 

$$\mathrm{GSVD}(\Delta C, \Delta F) \Rightarrow \{U_c, U_f, \Sigma_c, \Sigma_f, V\}$$

such that

$$\Delta C = U_c \; \Sigma_c \; V$$
$$\Delta F = U_f \; \Sigma_f \; V$$

$$S_k^\dagger = \Delta C_k \Delta F_k^\dagger \quad = U_c \; \Sigma_c \; \Sigma_f^\dagger \; U_f^T \quad = U_c \; \mathrm{diag}\left(\frac{\sigma_{c,i} + \lambda \;\; \sigma_{c,1}}{\sigma_{f,i} + \lambda \;\; \sigma_{f,1}} \right) \; U_f^T$$

regularization !

$$\mathbf{x}_0 = \operatorname{argmin}_\xi \|\mathbf{c}(\mathbf{p}(\xi)) - \mathbf{y}\|$$
$$\mathbf{x}_1 = \operatorname{argmin}_\xi \|\mathbf{c}(\mathbf{p}(\xi)) - \mathbf{c}(\mathbf{p}(\mathbf{x}_0)) + \mathbf{f}(\mathbf{x}_0)) - \mathbf{y}\|$$

for $k = 1, 2, \cdots$
do compute $\mathbf{f}(\mathbf{x}_k)$ and $\mathbf{c}(\mathbf{p}(\mathbf{x}_k)))$

$$\Delta C = (\mathbf{c}(\mathbf{p}(\mathbf{x}_k)) - \mathbf{c}(\mathbf{p}(\mathbf{x}_j)), \cdots)$$
$$\Delta F = (\mathbf{f}(\mathbf{x}_k) - \mathbf{f}(\mathbf{x}_j), \cdots)$$
$$\{U_c, U_f, \Sigma_c, \Sigma_f, V\} = \mathrm{GSVD}(\Delta C, \Delta F)$$

$$\mathbf{x}_{k+1} = \operatorname*{argmin}_\xi \|\mathbf{c}(\mathbf{p}(\xi)) - \mathbf{c}(\mathbf{p}(\mathbf{x}_k)) + U_c \ \mathrm{diag}\left(\frac{\sigma_{c,i}+\lambda_k\ \sigma_{c,1}}{\sigma_{f,i}+\lambda_k\ \sigma_{f,1}}\right) \ U_f^T \ (\mathbf{f}(\mathbf{x}_k) - \mathbf{y})\|$$

enddo

with $\lambda_k \longrightarrow 0$ for $k \longrightarrow \infty$

The full algorithm

$$\mathbf{x}_0 = \text{argmin}_\xi \, \|\mathbf{c}(\mathbf{p}(\xi))) - \mathbf{y}\|$$

for $k = 0, 1, \cdots$
do compute $\mathbf{f}(\mathbf{x}_k)$ and $\mathbf{c}(\mathbf{p}(\mathbf{x}_k)))$

$$\Delta C = (\mathbf{c}(\mathbf{p}(\mathbf{x}_k)) - \mathbf{c}(\mathbf{p}(\mathbf{x}_j)), \cdots)$$
$$\Delta F = (\mathbf{f}(\mathbf{x}_k) - \mathbf{f}(\mathbf{x}_j), \cdots)$$
$$\{U_c, U_f, \Sigma_c, \Sigma_f, V\} = \text{GSVD}(\Delta C, \Delta F)$$

$$\mathbf{y}_k = U_c \text{ diag}\left(\frac{\sigma_{c,i}+\lambda_k \, \sigma_{c,1}}{\sigma_{f,i}+\lambda_k \, \sigma_{f,1}}\right) U_f^T(\mathbf{f}(\mathbf{x}_k) - \mathbf{y})$$

$$\mathbf{x}_{k+1} = \underset{\xi}{\text{argmin}} \, \|\mathbf{c}(\mathbf{p}(\xi)) - \mathbf{c}(\mathbf{p}(\mathbf{x}_k)) + \mathbf{y}_k\|$$

and $\lambda_k \to 0$ for $k \to \infty$

enddo

The full algorithm

$$\mathbf{x}_0 = \mathrm{argmin}_\xi \ \|\mathbf{c}(\mathbf{p}(\xi))) - \mathbf{y}\|$$

for $k = 0, 1, \cdots$
do compute $\mathbf{f}(\mathbf{x}_k)$ and $\mathbf{c}(\mathbf{p}(\mathbf{x}_k)))$

$$\Delta C = (\mathbf{c}(\mathbf{p}(\mathbf{x}_k)) - \mathbf{c}(\mathbf{p}(\mathbf{x}_j)), \cdots)$$
$$\Delta F = (\mathbf{f}(\mathbf{x}_k) - \mathbf{f}(\mathbf{x}_j), \cdots)$$
$$\{U_c, U_f, \Sigma_c, \Sigma_f, V\} = \mathrm{GSVD}(\Delta C, \Delta F)$$

$$\mathbf{y}_k = U_c \ \mathrm{diag}\left(\frac{\sigma_{c,i}+\lambda_k \ \sigma_{c,1}}{\sigma_{f,i}+\lambda_k \ \sigma_{f,1}}\right) \ U_f^T(\mathbf{f}(\mathbf{x}_k) - \mathbf{y})$$

$$\mathbf{x}_{k+1} = \underset{\xi}{\mathrm{argmin}} \ \|\mathbf{c}(\mathbf{p}(\xi)) - \mathbf{c}(\mathbf{p}(\mathbf{x}_k)) + \frac{\mathbf{y}_k}{1 + \delta \lambda_k}\|$$

with δ trust region parameter and $\lambda_k \to 0$ for $k \to \infty$

enddo