Nonlinear Mathematical Programming - From Small to Very Large Scale Applications -

Frank Vogel, inuTech GmbH
Klaus Schittkowski, Universität Bayreuth

inuTech – One Slide Summary

- We offer mathematical software and R&D services in areas such as
  - Mathematical optimization
  - Differential equations
  - Further areas that require a thorough knowledge of mathematical foundations
- with a staff of currently 12 people
- to close the gap between Engineering and Mathematics
- for customers such as:

Contents

- SQP and SCP methods for nonlinear programming
- Comparative numerical tests
- Efficient sensitivity calculation
- Small to very large scale industrial appilications
- Conclusions

Problem Formulation

Nonlinear programming problem (NLP):

$$\min f(x)$$
$$x \in \Re^n: \quad g_j(x) \leq 0, \quad j = 1, \ldots, m$$

x - Design Vector
f - Objective Function
g_j – Constraint Functions

Assumptions:

- All model functions are smooth (differentiable)
- The problem may be highly nonlinear
- The problem may become very large scale

General Procedure

Goal: Given an current iterate $x_k \in \Re^n$, determine a search direction $d_k \in \Re^n$ and a step length $\alpha \in \Re$ to calculate the next iterate

$$x_{k+1} = x_k + \alpha\, d_k$$

Search direction: Formulate and solve a "simpler" subproblem called **NLP*k*** :

$$\min \; f^k(x)$$
$$x \in \Re^n : \quad g_j^k(x) \le 0, \quad j = 1,\ldots,m$$

General Procedure (cont.)

Requirements

- **NLP$_k$** is strictly convex and smooth, i.e. **NLP$_k$** has a unique solution $d_k \in \Re^n$

- **NLP$_k$** is a first order approximation, i.e.

$$f(x_k) = f^k(x_k) \qquad \nabla f(x_k) = \nabla f^k(x_k)$$
$$g_j(x_k) = g_j^k(x_k) \qquad \nabla g_j(x_k) = \nabla g_j^k(x_k)$$

- The search direction $d_k \in \Re^n$ is a descent direction for the augmented Lagrangian merit function (Schittkowski 1982)

Line Search

New Iterate: Search for $0 < \alpha_k \leq 1$ such that $\Psi_k(\alpha)$ becomes acceptable (”principle of sufficient descent”)

$$\Psi_k(\alpha) = \Phi_{r_k}\left(x_k + \alpha\, d_k\right)$$

***i.e., L_2*-Merit Function:**

$$\Phi_r(x) = f(x) + \sum_{j \in J} \frac{1}{2} r_j g_j(x)^2$$

SQP Methods (Sequential Quadratic Programming)

Goal: Fast local convergence speed based on local quadratic approximation of the Lagrangian function and linearization of constraints (Wilson 1963, Han 1976, Powell 1978)

$$
\begin{aligned}
f^k(x) &= \nabla f(x_k)^T (x - x_k) + \frac{1}{2}(x - x_k)^T B_k (x - x_k) \\
g_j^k(x) &= g_j(x_k) + \nabla g_j(x_k)^T (x - x_k),\ j = 1,\ldots,m
\end{aligned}
$$

Motivation: In case of equality constraints only, the optimal solution d_k represents one Newton step for solving the Karush-Kuhn-Tucker optimality conditions, if B_k is the Hessian matrix of the Lagrangian function

Properties of SQP Methods

- In case of *n* active constraints, the SQP method behaves like Newton's method for solving the corresponding system of equations.

- The algorithm is globally convergent.

- The local convergence speed is quadratic if B_k is exact and superlinear if B_k is an approximation of the Hessian (i.e. BFGS):

$$\left\| x_{k+1} - x^* \right\| < \gamma_k \left\| x_k - x^* \right\| \quad \text{with } \gamma_k \to 0$$

SCP Methods (Sequential Convex Programming)

Goal: Convex, 1st order approximation of model functions.

Idea: Insert inverse variables (with or without asymptotes) depending on sign of partial derivative and linearize

$$
\begin{aligned}
f^k(x) &= \alpha_0^k + \sum\nolimits_i^+ \frac{\beta_{i,0}^k}{U_i - x_i} - \sum\nolimits_i^- \frac{\beta_{i,0}^k}{x_i - L_i} \\
g_j^k(x) &= \alpha_j^k + \sum\nolimits_i^+ \frac{\beta_{i,j}^k}{U_i - x_i} - \sum\nolimits_i^- \frac{\beta_{i,j}^k}{x_i - L_i},\ j = 1,\ldots,m
\end{aligned}
$$

$\sum_i^+$: summation over all indices with positiv partial derivative

($\sum_i^-$ vice versa)

Motivation: Mechanical engineering, i.e. statically determinated structures are linear in reciprocal variables (Fleury 1989, Svanberg 1987, Zillober 1994)

SCP Methods (cont.)

Subproblem solution

- Regularization of objective function leads to strictly convex, separable, often very large NLP's (diagonal and positive definite Hessian)

- Numerical solution by interior point method possible with $n \times n$ or $m \times m$ systems of equations

- Exploiting sparsity patterns in systems of linear equations

Numerical Tests: SQP versus SCP

306 standard (medium sized) test problems:

Percentage of successful runs (*SUCC),* avg. number of function evaluations *(NF*), and avg. number of iterations (NIT) for SQP code NLPQLP (Schittkowski) and SCP code SCPIP001 (Zillober)

code	SUCC in %	NF	NIT
NLPQLP	100	39	25
SCPIP	93	74	42

Sensitivity Analysis

Both **SQP** and **SCP** require the **sensitivity information** of the objective and constraint functions wrt. the optimization variables:

$$g = g(x, p, u, \dot{u}, \ddot{u}, t)$$

Let *g* be a function which depends on *x* (space), *p* (optimization variable), *t* (time) and u, $\dot{u}$, $\ddot{u}$ the solution of:

$$M\ddot{u} + C\dot{u} + K u = R \quad \text{(PDE)}$$

Sensitivity Analysis – numerical

$$\frac{\partial}{\partial p_i} g = \frac{g(x, p + e_i \Delta p_i, u, \dot{u}, \ddot{u}, t) - g(x, p, u, \dot{u}, \ddot{u}, t)}{\Delta p_i}, \ i = 1, \ldots, n$$

Problems:

- **(n+1) solutions of (PDE), i.e. computationally very expensive** if
  - Solution of (PDE) is time consuming
  - Number of optimization parameters (n) is large

Sensitivity Analysis – analytical

Formal differentiation of (PDE) wrt. *p* yields:

$$M\frac{\partial \ddot{u}}{\partial p}+C\frac{\partial \dot{u}}{\partial p}+K\frac{\partial u}{\partial p}=\frac{\partial R}{\partial p}-\frac{\partial M}{\partial p}\ddot{u}-\frac{\partial C}{\partial p}\dot{u}-\frac{\partial K}{\partial p}u \quad (1)$$

i.e. structural mechanics (linear):

$$K\frac{\partial u}{\partial p}=\frac{\partial R}{\partial p}-\frac{\partial K}{\partial p}u \quad (2)$$

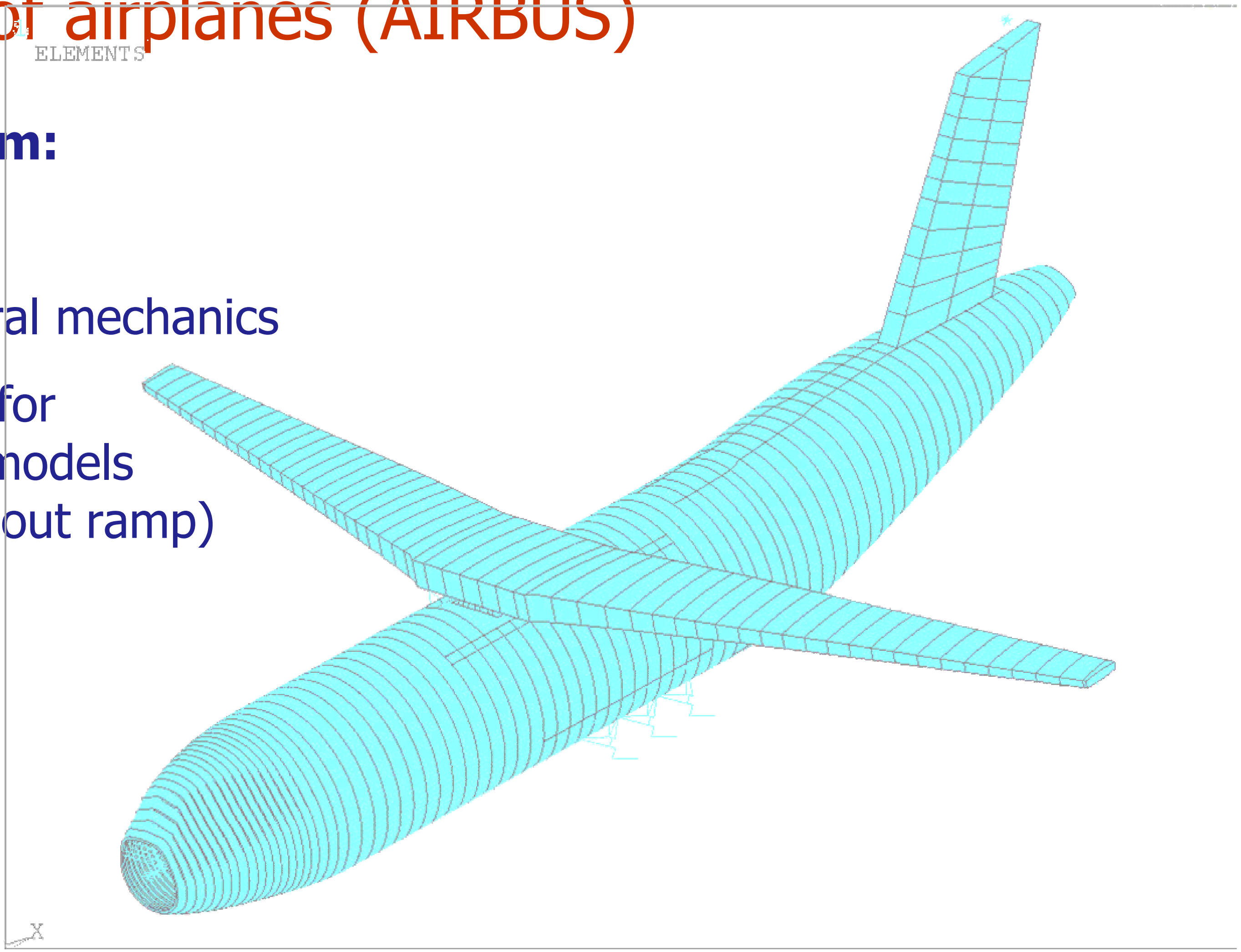
Solution of (1), (2) resp., yields sensitivities of *u* and thus of *g* in one shot!

Small to very large scale industrial Applications

Optimization of airplanes (AIRBUS)

Physical problem:

- Linear structural mechanics
- 15 load cases for
  two different models
  (with and without ramp)



Optimization of airplanes (cont.)

Optimization problem:

- Optimization variables: Beam cross sections and shell thicknesses
- Objective: Minimize weight
- Constraints: stresses, skin buckling of shells, euler buckling of beams, displacements, for all load cases simultaneously.
- Very large scale problem with > 2000 optimization variables and > 2.000.000 constraints
- Gradients are calculated semi-analytically
- Numerical solution using SCPIP in combination with an efficient active grouping / set logic

Topology Optimization

$$\min_x \quad u(x)^T p, \qquad \text{(compliance)}$$

$$\text{s.t.} \quad V(x) \leq V_{\max}, \qquad \text{(volume)}$$

$$K(x)u(x) = p, \qquad \text{(equilibrium condition)}$$

$$0 < x_{\min} \leq x_{i,j} \leq 1, \quad i = 1, \ldots, n_x, \; j = 1, \ldots, n_y.$$

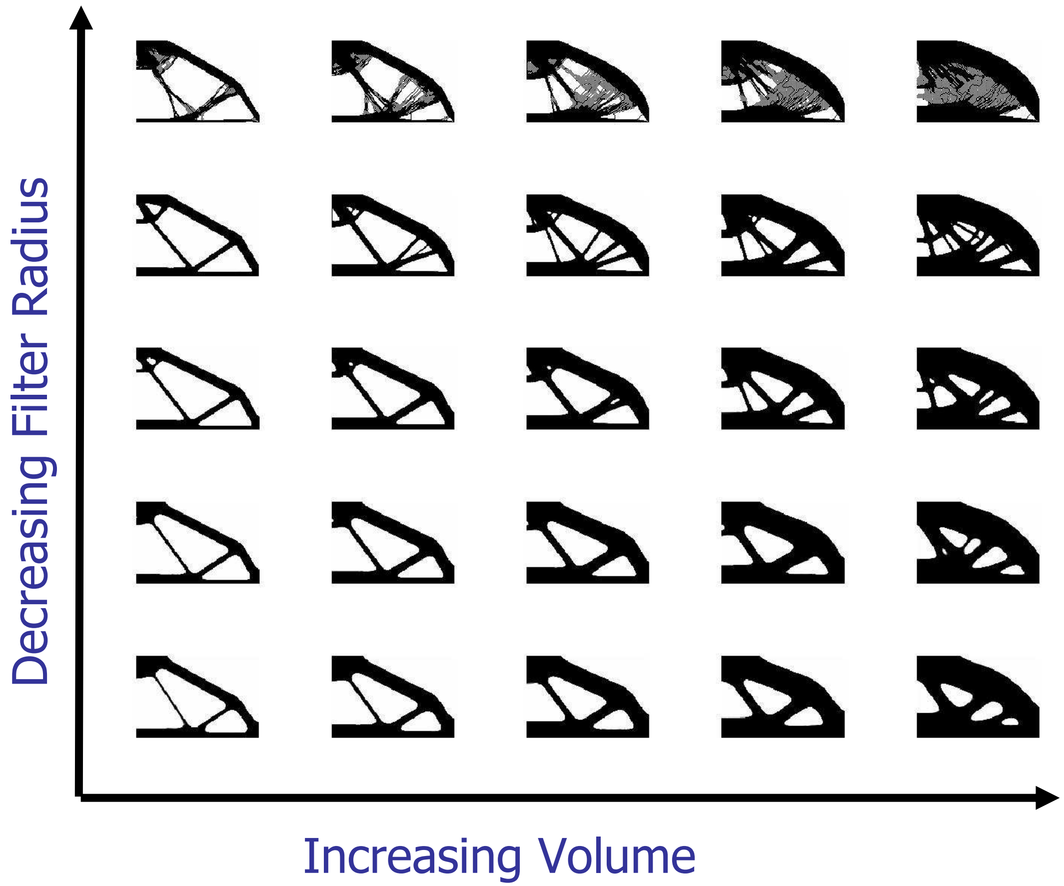
K: penalized global stiffness matrix
u: nodal displacements
V: volume
p: external load
$x_{i,j}$: relative density in element i,j
n_x: number of finite elements (x-direction)
n_y: number of finite elements (y-direction)

Topology Optimization (cont.)

Optimization problem:

- Optimization variables: Pseudo Element Densities $K(x) = \sum_{i=1}^{n} x_i^3 \cdot k_i^0$
- Objective: Minimize Compliance
- Constraints: Volume Constraints
- Very large scale problem with n (number of elements) optimization variables
- Gradients are calculated analytically
- Numerical solution using SCPIP

Topology Optimization – Case Studie (Half Beam)



n_x	n_y	n	n_{it}	$f(x)$	$\|\nabla_x L(x,u)\|$	r_f
600	400	240,000	22	52.63	1.3E-3	8
600	400	240,000	26	54.25	6.5E-4	0
1,050	700	735,000	38	54.39	4.6E-4	10
1,260	840	1,058,400	43	56.55	1.3E-5	0

n_x, n_y - number of elements in x- and y-direction
n - number of optimization variables
n_{it} - number of iterations
$f(x)$ - final objective function value
$\|\nabla_x L(x,u)\|$ - norm of final gradient of Lagrangian function
r_f - filter radius

Optimal control of Elliptic PDEs

Semi-linear elliptic control problem on $\Omega = (0,1)\times(0,1)$
(Maurer, Mittelmann 2001)

$$
\begin{array}{llll}
 & \min & \frac{1}{2}\int_\Omega \left(y(x) - \sin(2\pi x_1)\sin(2\pi x_2)\right)^2 + u(x)^2 \, dx & \\
u \in L^\infty & -\Delta y - e^y & = u & \text{on } \Omega \\
 & \partial_\nu y + y & = 0 & \text{on } \Gamma(\Omega) \\
 & y \le 0.371 & & \text{on } \Omega \\
 & -8 \le \text{u} \le 9 & &
\end{array}
$$

Optimal control of Elliptic PDEs (cont.)

Discretization: uniform square grid of size N

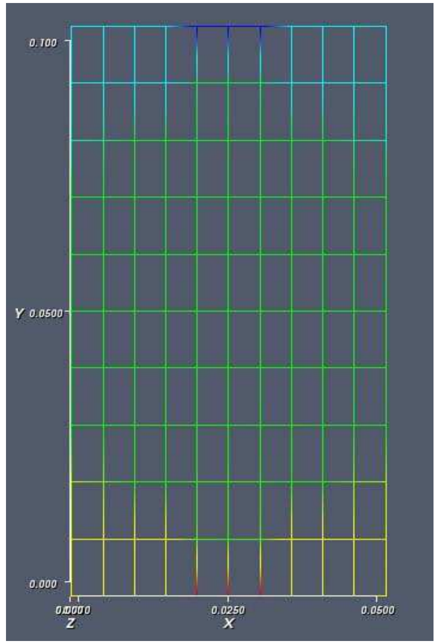
- Full discretization subject to state and control variables
- 5-star-formula for second derivatives
- Analytical calculation of gradients
- Very large scale problem solved by SCPIP

Optimal control of Elliptic PDEs (cont.)

Numerical results: SCPIP (Acc=1.0E-7)

$N+1$	n	m	n_{it}	$f(x)$
100	19,998	10,197	24	0.0527
200	79,998	40,397	21	0.0530
300	179,998	90,597	16	0.0535
400	319,998	160,797	18	0.0534
500	499,998	250,997	16	0.0536
600	719,998	312,197	12	0.0548

Optimal Design of Capacitors



Physical problem:

- Capacitor with 2 contacts
- Voltage: 0V at top, 220V at bottom, symm. conditions left and right
- Stationary model:

$$\frac{\partial}{\partial x}\left(p_x \frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(p_y \frac{\partial u}{\partial y}\right) = 0 \qquad \begin{array}{ll} u(x) = 220V & x \in \Gamma_1 \\ u(x) = 0V & x \in \Gamma_2 \\ \dfrac{\partial u}{\partial n} = 0 & x \in \Gamma_3 \end{array}$$

u – electrical potential, p_x ,p_y – material conductivity

- Model solved employing a Diffpack – Simulator (based on FEM)

Optimal Design of Capacitors (cont.)

Optimization problem:

$$\underset{p_i}{Min} \sum_{i=1}^{n} p_i \left\|\nabla u\right\|_i^2 A_i \text{ , s.t.} \qquad p_i^2 \left\|\nabla u\right\|_i^2 \le j_{\max}^2, i = 1,\ldots,n$$

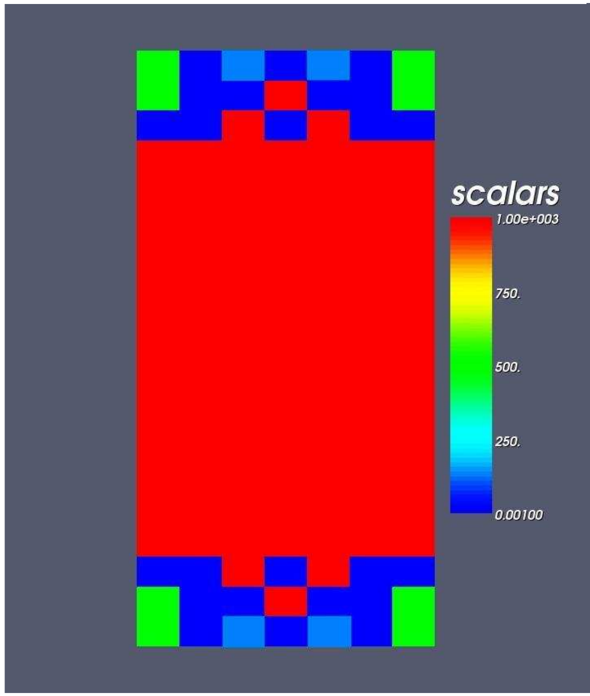
$$0.9 I_{Ziel} \le \int p \nabla u \, \vec{n} \, dx \le 1.1 I_{Ziel}$$

- Optimization variables: p_i – material conductivity for element i.
- Objective: Minimize the flow resistance on potential field
- Constraints on: Current densities for all elements, Current on bottom boundary
- Large scale problem: n (number of elements) variables, n+2 constraints
- A direct method has been implemented using Diffpack to calculate the gradients analytically: $K \frac{du}{dp_i} = \frac{df}{dp_i} - \frac{dK}{dp_i} u$

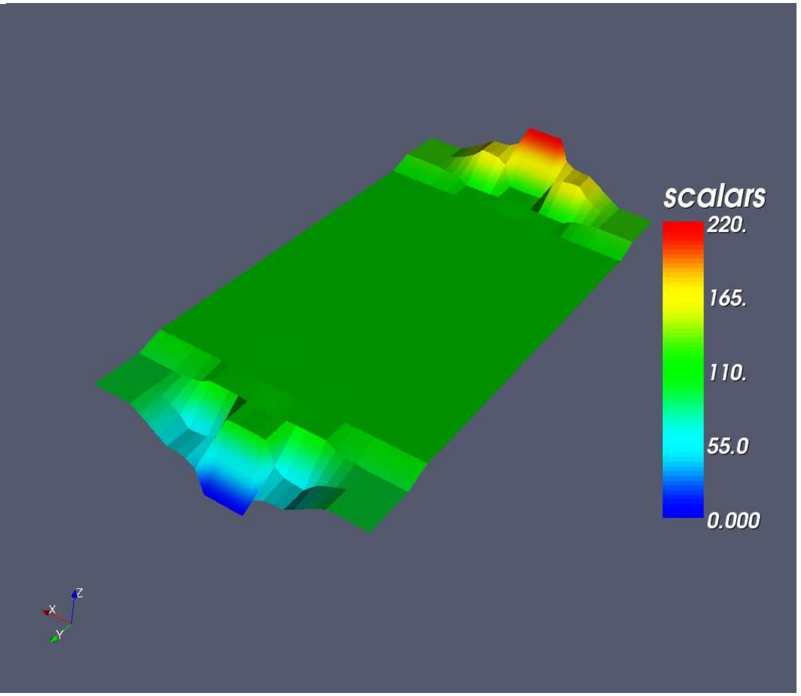
Optimal Design of Capacitors (cont.)

Optimization problem:

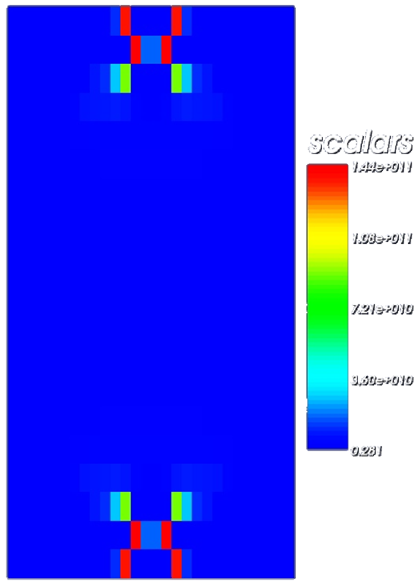
- Numerical solution using SCPIP



Optimal conductivity distribution

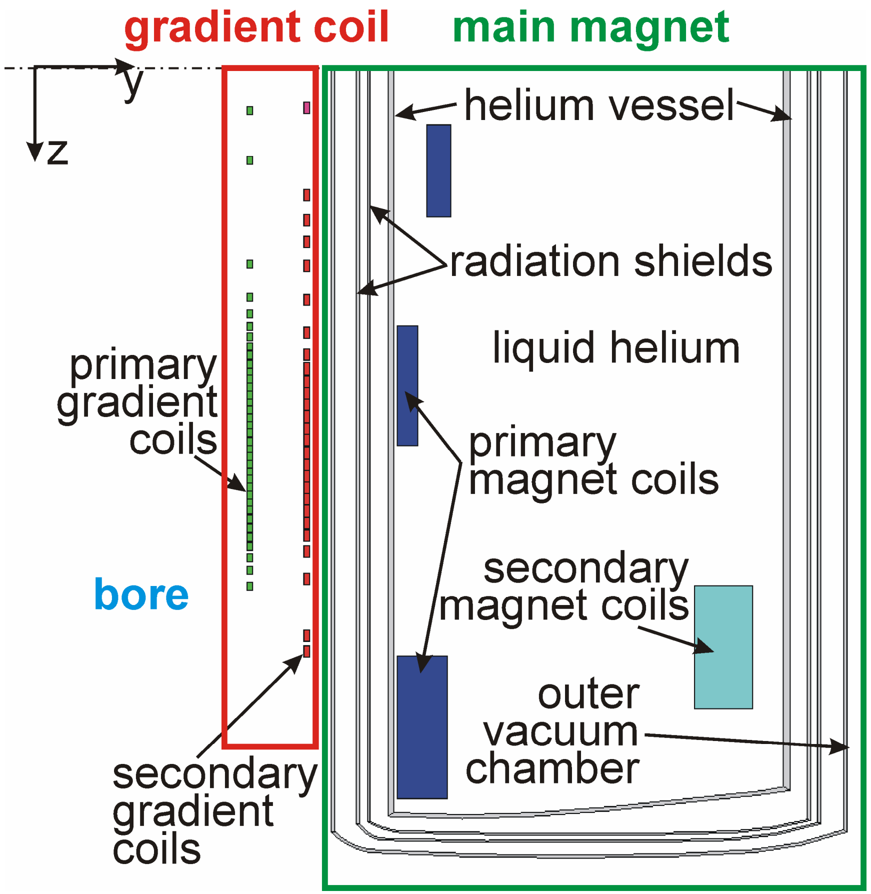


Potential u



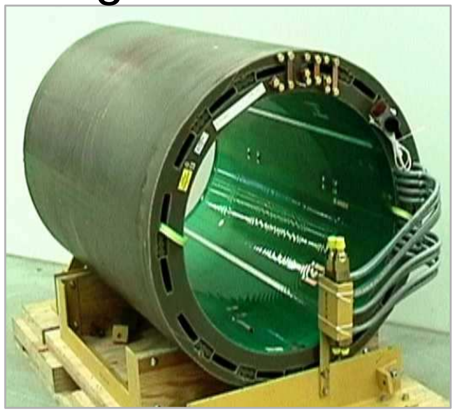
Current Densities

Optimization of MRI Scanner (SIEMENS)



main magnet

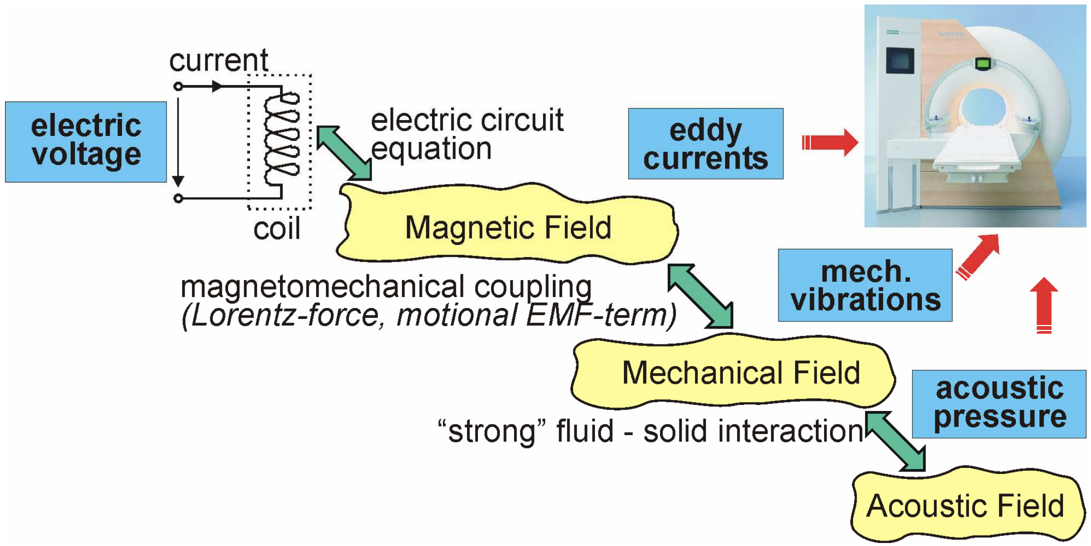
gradient coil



Optimization of MRI Scanner (cont.)

Physical problem:

- Coupled physical effects (solved using CAPA and Siemens in-house solver)



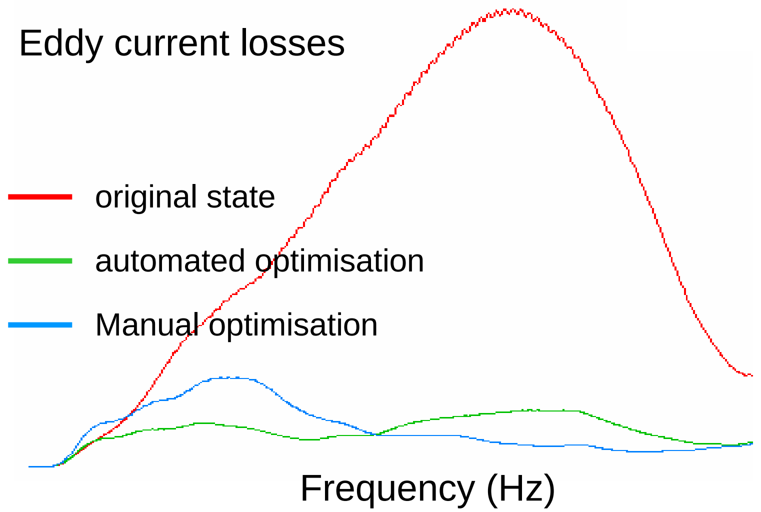
Optimization of MRI Scanner (cont.)

Optimization problem:

- Optimization variables: currents of prim. and second. gradient coils
- Objective function: Eddy current losses in frequency range (calculated by CAPA)

$$\Phi(z) = \int_{f_1}^{f_2} \omega(f) \left(\int_{\Omega} \rho \vec{A}^2 \mathrm{d}\Omega - Q_{target} \right)^p \mathrm{d}f$$

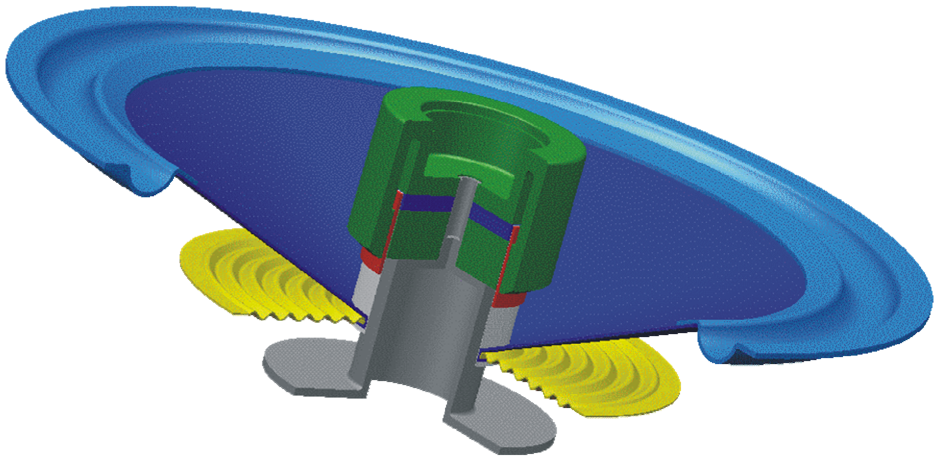
- Constraints: inductance, linearity in a given field-of-view, shielding, power dissipation, etc. (calculated by SIEMENS in-house tools)
- Gradients are calculated semi-analytically with CAPA
- Numerical solution using SQP



Electrodynamic Loudspeaker (HARMAN/KARDON)

Physical problem:

- Coupled simulation of magnetic, mechanic and acoustic domain (solved with CAPA)
- Nonlinear magnetics and coupling effects
- Time domain simulation with 2.000 time steps and more
- Single simulation run takes 0.5 – 2 h cpu time



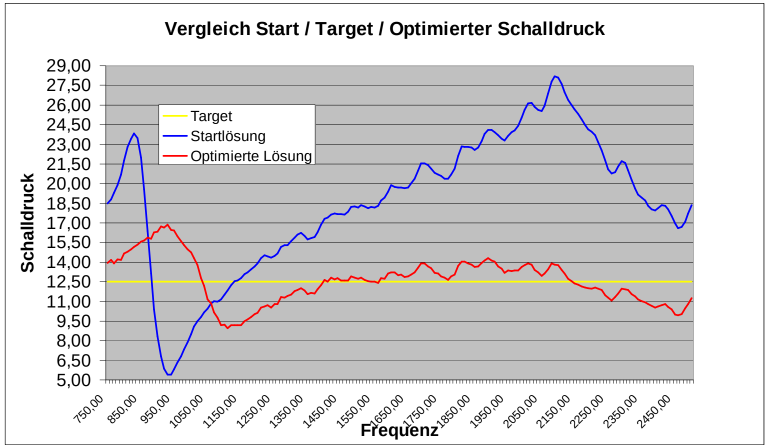
Electrodynamic Loudspeaker (cont.)

Optimization problem:

- Optimization variables: Material parameters of Loudspeaker
- Objective function: „Constant" SPL (sound pressure level) in given frequency domain:

$$F(x) = \sum_{f_i \in \{750,760,\ldots,2500\}} \left(\Phi(f_i, x) - \Phi_{Target}(f_i)\right)^2$$

- Semi-analytical calculation of sensitivities within CAPA
- Numerical solution using SQP



Conclusions

- **SQP** methods represent the *state-of-the-art* in nonlinear programming for smooth, small to medium sized and highly nonlinear optimization problems
- **SQP** methods are widely used and accepted in many engineering sciences, academia and commercial
- **SCP** methods have been invented by engineers, tuned to structural design optimization
- **SCP** methods are sometimes even more efficient than SQP methods in case of lower accuracy requirements
- **SCP** methods are applicable for solving very large scale optimization problems
- **Efficient Sensitivity Analysis** is always beneficial (especially for large scale problems)

Advantages and disadvantages

Advantages of SQP Methods

- Robust
- Highly accurate solutions with fast final convergence speed
- Highly nonlinear problems
- General purpose tools

Disadvantages of SQP Methods

- Fast convergence only in case of accurate gradients
- Large storage requirements

Advantages and disadvantages (cont.)

Advantages of SCP Methods

- Tuned for structural optimization
- No inheritance of round-off errors in function and gradient calculations
- Excellent convergence speed in special situations
- Able to solve very large scale problems

Disadvantages of SCP Methods

- Slow convergence possible

Thank you for your attention !