Department of Aerospace and Geodesy
TUM School of Engineering and Design
Technical University of Munich

Chapter 5:

Approximation Techniques

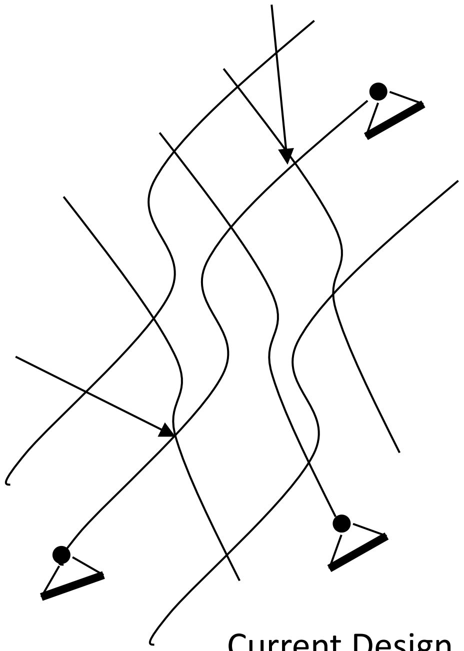
Layout

- Why using an approximation
- Local & global approximation methods
- Local approximation methods:
  - First order and reciprocal approximation
  - Higher order approximation
- Method of Moving Asymptotes MMA
- MMA & Dual algorithms

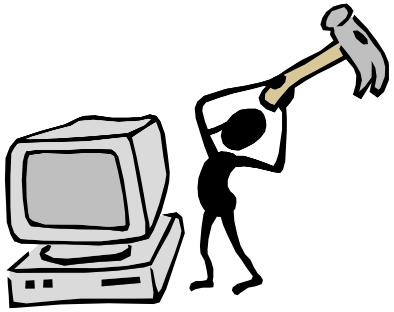
Department of Aerospace and Geodesy
TUM School of Engineering and Design
Technical University of Munich

Why using an approximation

- Usually objective and constraints are NOT explicitly known and highly nonlinear functions
- Numerical Analysis (e.g. finite element analysis) provides only a point-wise evaluation



Current Design



FE-Analyse
Sensitivity Analysis

Physical level

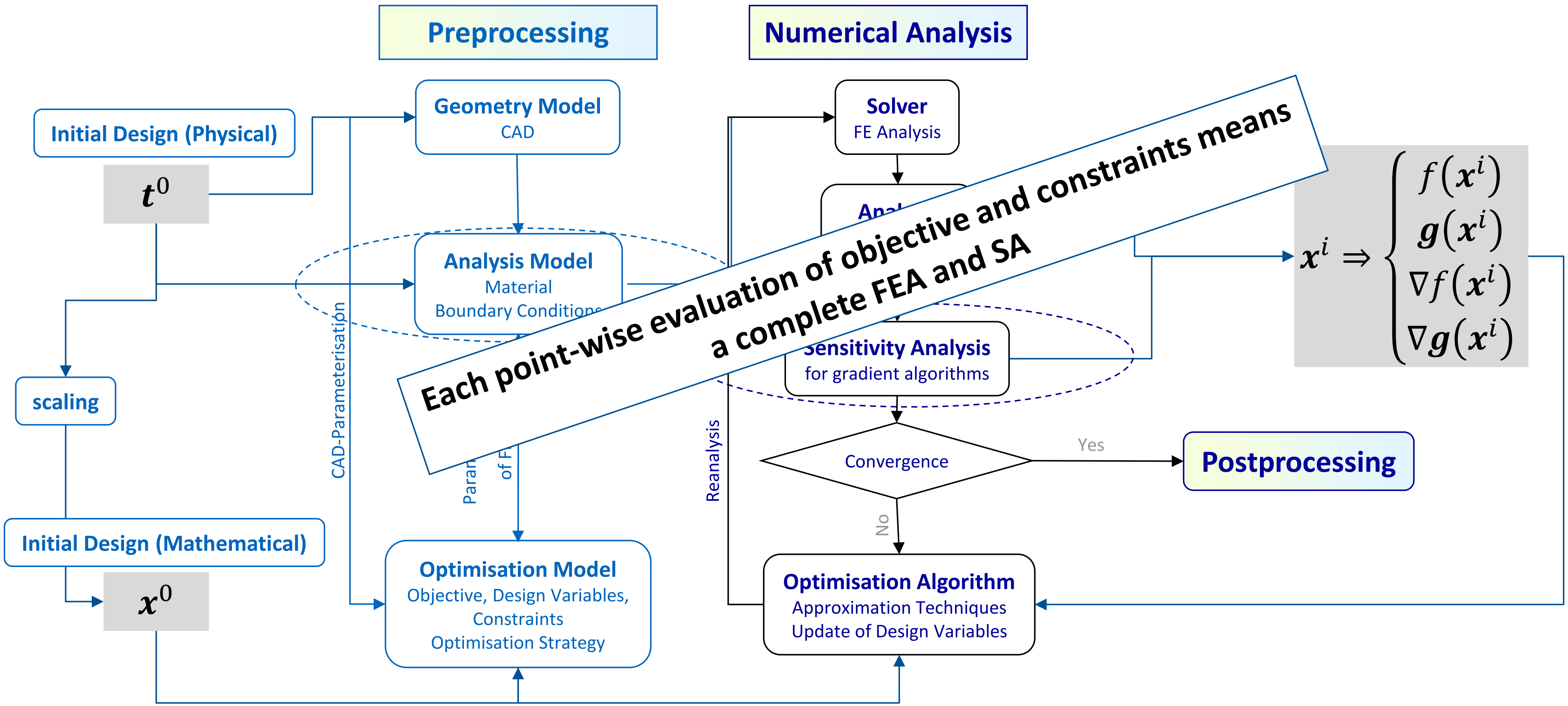
$$
\begin{cases}
t_1 = 1.59 \\
t_2 = 1.59 \\
\vdots \\
t_{239} = 34.2 \\
Weight = 32.4 \\
\sigma_{11} = 213.4 \\
\dfrac{\partial \sigma_{11}}{\partial t_{21}} = -12.9
\end{cases}
$$

Evaluation of Objective and constraints (and their sensitivities)

Mathematical level

$$
\begin{array}{l}
x_1 \\
\vdots \\
x_{60} \\
f(\boldsymbol{x}) \\
g_1(\boldsymbol{x}) \\
\dfrac{\partial g_1}{\partial x_{13}}
\end{array}
$$

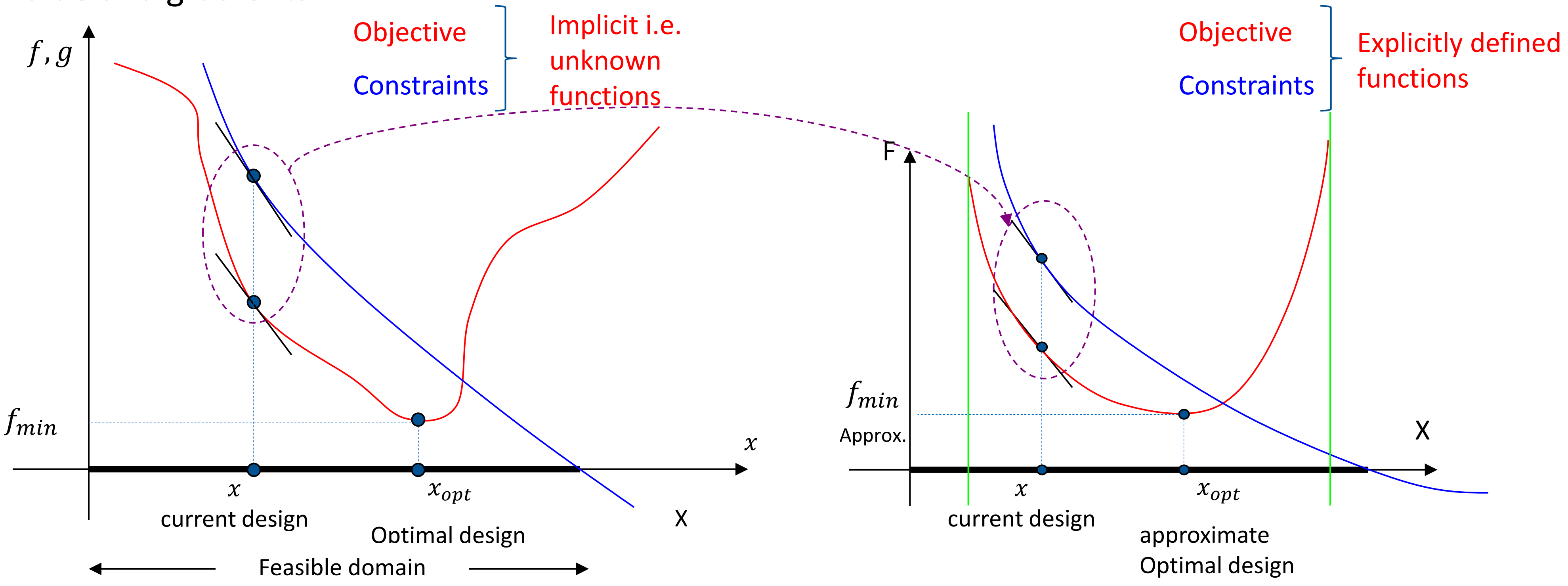
Why using an approximation



Each point-wise evaluation of objective and constraints means a complete FEA and SA

Approximation Techniques – Basic idea

- Based on point-wise evaluation of objective and constraints (function evaluation and gradients) a so-called subproblem is established
- The subproblem is established so that original problem and approximation are identical at expansion point: function value and gradients



Original problem

Subproblem

Local Approximation Techniques

- First Order and Reciprocal Approximation:
  - Easiest method of first order approximation is a Taylor series with one term

$$F(\mathbf{x}) \approx \overline{F}_L(\mathbf{x}) = F(\mathbf{x_0}) + \sum_{i=1}^{n}(x_i - x_{0i})\left(\frac{\partial F}{\partial x_i}\right)_{\mathbf{x}_0}$$

  - For highly nonlinear functions a linear approximation could be – even in the vicinity of expansion point – inaccurate
  - Most important alternative for better approximation of nonlinear functions without using higher order information is the reciprocal approximation
  - Reciprocal approximation means linearisation coupled with following variable transformation:

$$z_i = \frac{1}{x_i}$$

$$F(\mathbf{z}) \approx \overline{F}_R(\mathbf{z}) = F(\mathbf{z_0}) + \sum_{i=1}^{n}(z_i - z_{0i})\left(\frac{\partial F}{\partial z_i}\right)_{\mathbf{z}_0}$$

Local Approximation Techniques

- First Order and Reciprocal Approximation:
  - Reformulation of the approximation in original variables:

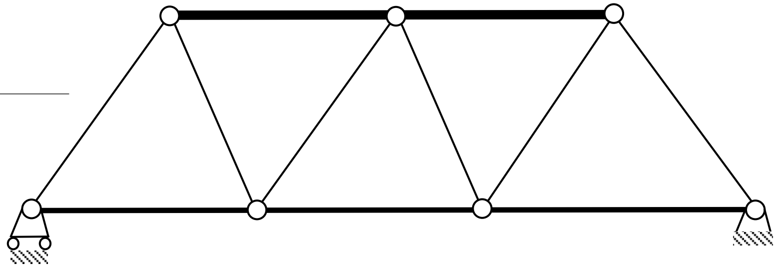
$$
\begin{aligned}
F(\mathbf{z}) \approx \overline{F}_R(\mathbf{z}) &= F(\mathbf{z_0}) + \sum_{i=1}^{n} (z_i - z_{0i}) \left(\frac{\partial F}{\partial z_i}\right)_{\mathbf{z}_0} \overset{z_i = \frac{1}{x_i}}{\Longrightarrow} \\
&= F(\mathbf{x_0}) + \sum_{i=1}^{n} \left(\frac{1}{x_i} - \frac{1}{x_{0i}}\right) \left(\frac{\partial F}{\partial \frac{1}{x_i}}\right)_{\mathbf{x_0}} \\
&= F(\mathbf{x_0}) + \sum_{i=1}^{n} \left(\frac{x_{0i} - x_i}{x_{0i} \cdot x_i}\right)(-x_{0i}^2) \left(\frac{\partial F}{\partial x_i}\right)_{\mathbf{x_0}} \\
&= F(\mathbf{x_0}) + \sum_{i=1}^{n} (x_i - x_{0i}) \frac{x_{0i}}{x_i} \left(\frac{\partial F}{\partial x_i}\right)_{\mathbf{x_0}}
\end{aligned}
$$

Local Approximation Techniques

- First Order and Reciprocal Approximation:
  - Reciprocal approximations were first successfully used in the begin of the 80s in sizing optimisation of trusses
  - The secret behind the success of reciprocal approximation in structure sizing optimisation can be understood by examining the relationship of stresses and sizing design variables in statically determinate trusses

$$\sigma = \frac{N}{\boxed{A}} = N \cdot \frac{1}{\boxed{x}}$$

design variable



  - Most structural responses in sizing or shape optimisation are reciprocal functions of design variables (cross section areas, thickness, coordinates of finite element nodes, etc.)
  - Many versions of reciprocal and mixed approximation can be found in textbooks

Local Approximation Techniques

- Second Order Approximation:
  – Taylor series expansion with two members:

$$\overline{F}_Q(\mathbf{x}) = F(\mathbf{x_0}) + \sum_{i=1}^{n} (x_i - x_{0i}) \left(\frac{\partial F}{\partial x_i}\right)_{\mathbf{x_0}} + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (x_{0i} - x_i)(x_{0j} - x_j) \left(\frac{\partial^2 F}{\partial x_i \partial x_j}\right)_{\mathbf{x_0}}$$

  – Many other more sophisticated second order approximations can be found in textbooks
  – In structural optimisation second order approximation is not relevant due difficulties in calculating second order information

Global Approximation Techniques – Basic idea

- Global approximation schemes intend – usually based on zero order information – to construct an alternative function for the whole domain of design variables i.e. not only in the vicinity of the current design
- Global approximations are at first used in experimental physics in order to reduce the number of required (usually very expensive) tests to capture the system behaviour
- These methods can be used in numerical simulation by replacing the experiments by numerical analyses
- A complete field of theoretical research is involved, the so-called Design of Experiments DOE
- Most famous methods of global approximation are the response surface method and Kriging
- This topic is out of scope of this course

Extended Method of Moving Asymptotes MMA

- History & Basic Idea:
  - One of the most famous reciprocal approximation schemes is MMA
  - First presented by Svanberg in 1987
  - The method was enhanced by Bletzinger in 1992
  - In 2007 Svanberg presented the most recent development of his MMA algorithm:
    "On a globally convergent version of MMA". In Proceedings of the Seventh World Congress of Structural and Multidisciplinary Optimization, Seoul, Korea, May 2007
  - MMA and some of its derivatives like SCP (Sequential Convex Programming) are implemented in many optimisation packages (Optistruct, LAGRANGE, COMBOX [Airbus])

Extended Method of Moving Asymptotes MMA

- History & Basic Idea:
  - Basic idea of MMA is to decompose the n-dimensional space of design variables in a sum of n one-dimensional spaces
  - Each summand in the approximation contains only the reciprocal of one design variable
  - This means that MMA approximation is
    - Separable
    - Convex
    - Reciprocal

Extended Method of Moving Asymptotes MMA

- Definition of terms and terminology
  - We need to distinguish between the data from the original problem, that are only point-wise available e.g. through numerical analysis, and the approximations defined by analytical functions
  - The zero and first order data of the original functions are denoted $\tilde{f}(\boldsymbol{x}), \tilde{\boldsymbol{g}}(\boldsymbol{x}), \tilde{\boldsymbol{h}}(\boldsymbol{x})$ and $\nabla\tilde{\boldsymbol{f}}(\boldsymbol{x}), \nabla\tilde{\boldsymbol{g}}(\boldsymbol{x}), \nabla\tilde{\boldsymbol{h}}(\boldsymbol{x})$
  - Accordingly, the Lagrange function data of the original problem
    $\tilde{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = \tilde{f}(\boldsymbol{x}) + \sum_{j=1}^{m} \lambda_j \cdot \tilde{g}_j\,(\boldsymbol{x}) + \sum_{j=m+1}^{m+me} \mu_j \cdot \tilde{h}_j\,(\boldsymbol{x})$ and
    $\nabla_{\boldsymbol{x}}\tilde{\boldsymbol{L}}(\boldsymbol{x}, \boldsymbol{\lambda}) = \nabla_{\boldsymbol{x}}\tilde{\boldsymbol{f}}(\boldsymbol{x}) + \sum_{j=1}^{m} \lambda_j \cdot \nabla_{\boldsymbol{x}}\, \tilde{\boldsymbol{g}}_j(\boldsymbol{x}) + \sum_{j=m+1}^{m+me} \mu_j \cdot \nabla_{\boldsymbol{x}}\, \tilde{\boldsymbol{h}}_j(\boldsymbol{x})$

Extended Method of Moving Asymptotes MMA

- Mathematical formulation:
  - Given the current design $\boldsymbol{x}^k$ at iteration k, lower and upper approximation asymptotic bounds L_i^k and U_i^k are defined so that:

$$L_i^k < x_i^k < U_i^k$$

  - The functions involved in the (original) optimisation problem $\tilde{P}^k$:

    $\text{Minimise} \tilde{f}(\boldsymbol{x})$

    s.t.

    $\tilde{\boldsymbol{g}}_j(\boldsymbol{x}) \leq 0 \quad \ldots \quad j \in [1, m]$

    $\tilde{\boldsymbol{h}}_j(\boldsymbol{x}) = 0 \quad \ldots \quad j \in [m+1, m+me]$

    $\underline{x}_i \leq x_i \leq \overline{x}_i \quad \ldots \quad \boldsymbol{x} \in \mathbb{R}^n$

    are approximated as follows:

Approximated optimisation problem P^k:

$$f^k(\boldsymbol{x}) = r_0^k + \sum_{i=1}^{n} \left(\frac{p_{0i}^k}{U_i^k - \boxed{x_i}} + \frac{q_{0i}^k}{\boxed{x_i} - L_i^k} \right)$$ Objective

$$g_j^k(\boldsymbol{x}) = r_j^k + \sum_{i=1}^{n} \left(\frac{p_{ji}^k}{U_i^k - \boxed{x_i}} \; OR \; \frac{q_{ji}^k}{\boxed{x_i} - L_i^k} \right)$$ Inequality constraints

$$h_j^k(\boldsymbol{x}) = r_j^k + \sum_{i=1}^{n} \left(\frac{p_{ji}^k}{U_i^k - \boxed{x_i}} - \frac{p_{ji}^k}{\boxed{x_i} - L_i^k} \right)$$ Equality constraints

  - Don’t confuse the upper and lower approximation bounds L_i^k, U_i^k with the upper and lower bounds of the design variable – the side constraints

Extended Method of Moving Asymptotes MMA

- Mathematical formulation:
  - The rules for defining those bounds, called asymptotes, L_i^k and U_i^k will be discussed later, however, it's important to know that current design variable x_i^k lies always in the **middle** between the two asymptotes i.e. $L_i^k = x_i^k - \Delta_i^k$ and $U_i^k = x_i^k + \Delta_i^k$, with Δ_i^k called "half asymptotes distance"
  - The reciprocal dependency and the separation of the design variables are both obvious
  - The constants p_{ji}^k and q_{ji}^k for inequality constraints ($j \in [1, m]$), then for equality constraints ($j \in [m+1, me]$) and finally for the objective ($j = 0$) are elaborated next
  - The upper and lower bounds of the design variables (side constraints) are not approximated

$$f^k(\boldsymbol{x}) = r_0^k + \sum_{i=1}^{n} \left(\frac{p_{0i}^k}{U_i^k - x_i} + \frac{q_{0i}^k}{x_i - L_i^k} \right)$$ Objective

$$g_j^k(\boldsymbol{x}) = r_j^k + \sum_{i=1}^{n} \left(\frac{p_{ji}^k}{U_i^k - x_i} \; OR \; \frac{q_{ji}^k}{x_i - L_i^k} \right)$$ Inequality constraints

$$h_j^k(\boldsymbol{x}) = r_j^k + \sum_{i=1}^{n} \left(\frac{p_{ji}^k}{U_i^k - x_i} - \frac{p_{ji}^k}{x_i - L_i^k} \right)$$ Equality constraints

$$g_j^k(\boldsymbol{x}) = r_j^k + \sum_{i=1}^{n}\left(\frac{p_{ji}^k}{U_i^k - x_i} OR \frac{q_{ji}^k}{x_i - L_i^k}\right)$$

Inequality constraints

MMA – approximation of inequality constraints

- The coefficients p_{ji}^k and q_{ji}^k for $j \in [1, m]$ i.e., for the inequality constraints depend on the sign of the gradient at the current design variable value x_i^k and the upper and lower approximation bounds L_i^k and U_i^k:

$$p_{ji}^k = \begin{cases} (U_i^k - x_i^k)^2 \cdot \dfrac{\partial \tilde{g}_j\left(x_i^k\right)}{\partial x_i}, \text{if } \dfrac{\partial \tilde{g}_j\left(x_i^k\right)}{\partial x_i} > 0 \\ 0, \text{if } \dfrac{\partial \tilde{g}_j\left(x_i^k\right)}{\partial x_i} \leq 0 \end{cases}$$

$$q_{ji}^k = \begin{cases} 0, \text{if } \dfrac{\partial \tilde{g}_j\left(x_i^k\right)}{\partial x_i} > 0 \\ -(x_i^k - L_i^k)^2 \cdot \dfrac{\partial \tilde{g}_j\left(x_i^k\right)}{\partial x_i}, \text{if } \dfrac{\partial \tilde{g}_j\left(x_i^k\right)}{\partial x_i} \leq 0 \end{cases}$$

The coefficients p_{ji}^k and q_{ji}^k are defined to get the same gradient of original function and constraint at expansion point

$$\frac{\partial g_j\left(x_i^k\right)}{\partial x_i} = \frac{p_{0i}^k}{\left(U_i^k - x_i\right)^2} OR \frac{-q_{0i}^k}{\left(x_i - L_i^k\right)^2}$$

- The constants r_j^k are defined that the function values at the current design variables $\boldsymbol{x}^k$ of the original function and the approximation are identical:

$$r_j^k = \tilde{g}_j\left(\boldsymbol{x}^k\right) - \sum_{i=1}^{n}\left(\frac{p_{ji}^k}{U_i^k - x_i^k} OR \frac{q_{ji}^k}{x_i^k - L_i^k}\right)$$

- The constraint approximation consists of set n hyperbolas (**one** hyperbola for each the design variable component) that is aligned to one of the two asymptotes, either L_i^k or U_i^k

MMA – approximation of inequality constraints

- Here‘s an example for an inequality constraint:

$$\boldsymbol{x}^k = \begin{bmatrix} -1.0 \\ 3.5 \end{bmatrix}$$

$$\tilde{g}(\boldsymbol{x}^k) = 10$$

$$\frac{\partial \tilde{g}(\boldsymbol{x}^k)}{\partial x_1} = 4.2$$

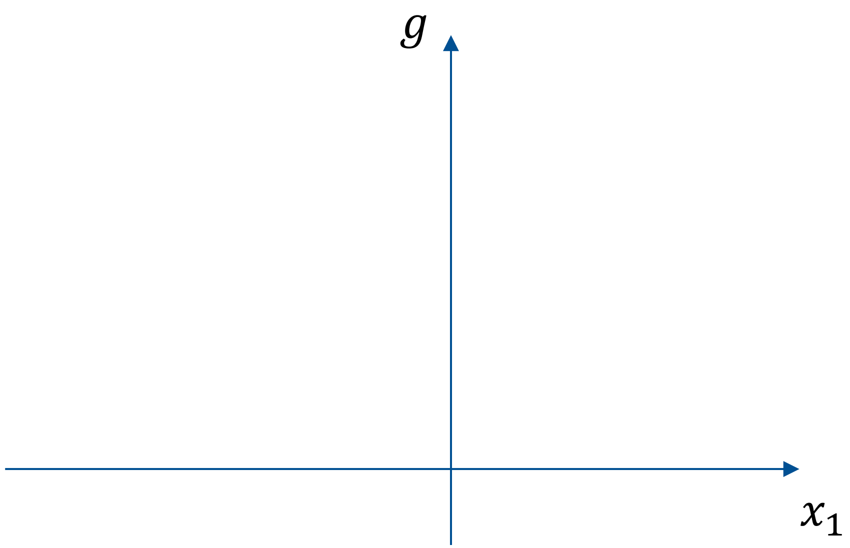
$$\frac{\partial \tilde{g}(\boldsymbol{x}^k)}{\partial x_1} = -1.4$$

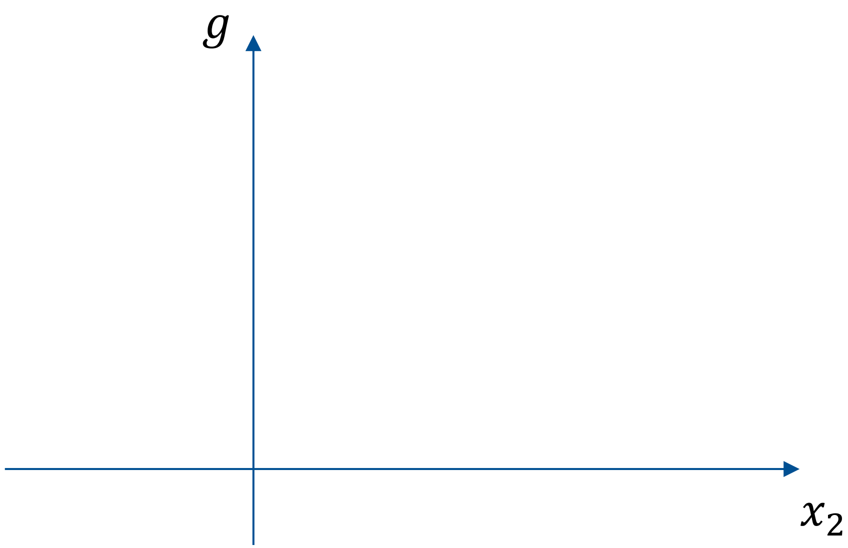
$$L_1^k = -6.0$$

$$U_1^k = 4.0$$

$$L_2^k = -3.5$$

$$U_2^k = 10.5$$





$$h_j^k(\boldsymbol{x}) = r_j^k + \sum_{i=1}^{n}\left(\frac{p_{ji}^k}{U_i^k - x_i} - \frac{p_{ji}^k}{x_i - L_i^k}\right)$$

Equality constraints

MMA – approximation of equality constraints

- The question might arise, why having a different approximation for the equality constraints i.e., why not using the approximation procedure of the inequality constraints?
- The reason lies in the necessity to guarantee the activity of the equality constraint! In other words: the approximation procedure of the inequality constraints doesn't guarantee that the root (intersection point with the x-axis) will lie between the asymptotes – potentially leading to an unsolvable optimisation problem!
- The coefficients p_{ji}^k for $j \in [m+1, m+me]$ i.e., for the equality constraints depend on the sign of the gradient at the current design variable value x_i^k and the upper and lower approximation bounds L_i^k and U_i^k:

$$p_{ji}^k = \frac{1}{2}(U_i^k - x_i^k)^2 \cdot \frac{\partial \tilde{h}_j\left(x_i^k\right)}{\partial x_i}$$

$$r_j^k = \tilde{h}_j\left(\boldsymbol{x}^k\right)$$

The coefficients p_{ji}^k are defined to get the same gradient of original function and constraint

$$\frac{\partial h_j\left(x_i^k\right)}{\partial x_i} = \frac{2p_{0i}^k}{\left(U_i^k - x_i\right)^2}$$

- Note that, $x_i^k - L_i^k = U_i^k - x_i^k = \Delta_i^k$
- The constants r_j^k are defined that the function values at the current design variables $\boldsymbol{x}^k$ of the original function and the approximation are identical
- The constraint approximation consists of set $2n$ hyperbolas (**two** hyperbola for each the design variable component) that are aligned to the two asymptotes L_i^k and U_i^k

$$h_j^k(\boldsymbol{x}) = r_j^k + \sum_{i=1}^{n} \left(\frac{p_{ji}^k}{U_i^k - x_i} - \frac{p_{ji}^k}{x_i - L_i^k} \right)$$

Equality constraints

MMA – approximation of equality constraints

- Here‘s an example for an equality constraint:

$$\boldsymbol{x}^k = \begin{bmatrix} -1.0 \\ 3.5 \end{bmatrix}$$

$$\tilde{g}(\boldsymbol{x}^k) = 10$$

$$\frac{\partial \tilde{g}(\boldsymbol{x}^k)}{\partial x_1} = 4.2$$

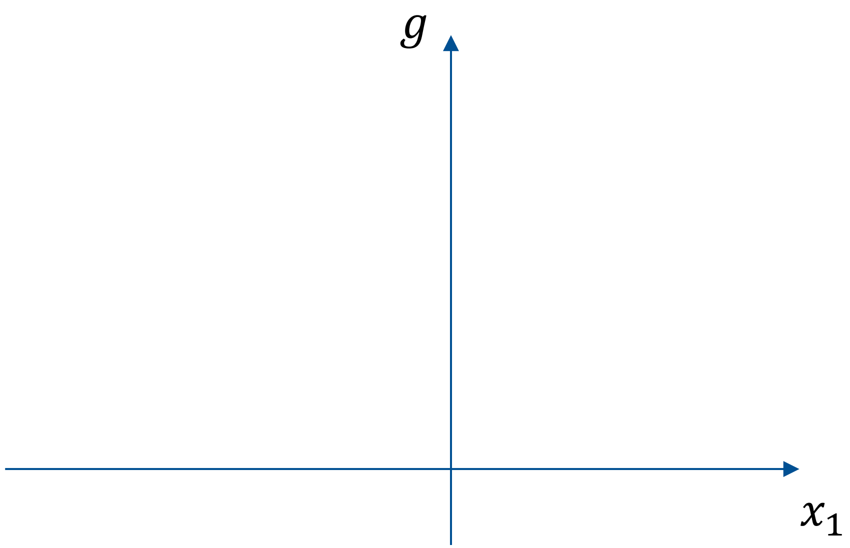
$$\frac{\partial \tilde{g}(\boldsymbol{x}^k)}{\partial x_1} = -1.4$$

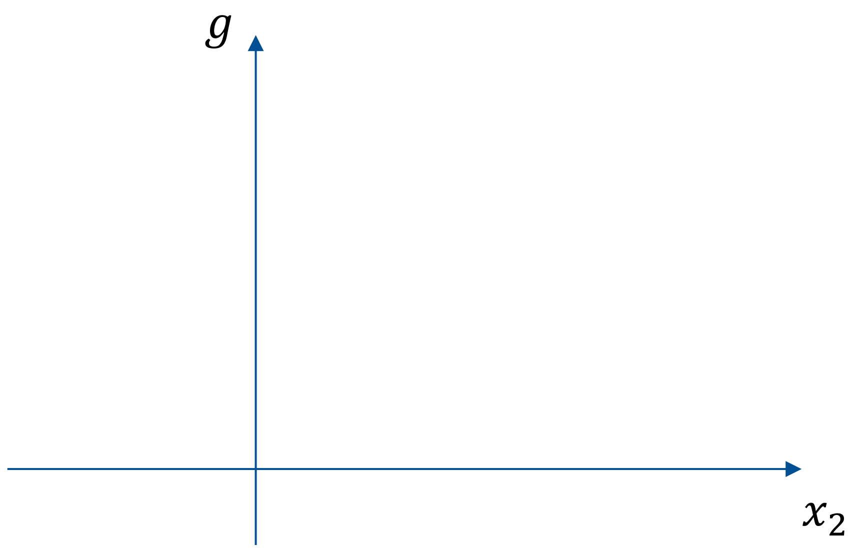
$$L_1^k = -6.0$$

$$U_1^k = 4.0$$

$$L_2^k = -3.5$$

$$U_2^k = 10.5$$





$$f^k(\boldsymbol{x}) = r_0^k + \sum_{i=1}^{n}\left(\frac{p_{0i}^k}{U_i^k - x_i} + \frac{q_{0i}^k}{x_i - L_i^k}\right)$$ Objective

MMA – the approximation of the objective function

- The objective function is also approximated by two hyperbolas, however, with the target of constructing a convex function with a qualified minimum between the asymptotes
- Enhancing the approximation of the objective function by having both coefficients p_{0i}^k and q_{0i}^k leads to $2n+1$ unknowns (r_0^k is the +1), however, with $n+1$ equations only:
  – Same objective function value at $\boldsymbol{x}^k$ (original and approximation): 1 equation
  – Same derivatives of objective function at $\boldsymbol{x}^k$ (original and approximation): n equations
- At this point, it seems appropriate to include second-order information into the approximation concept, because the asymptote distance $U_i^k - L_i^k = 2\Delta_i^k$, the curvature of the objective function and the curvature of the Lagrange function are directly related
- Since, second order information are usually computationally too expensive, they will be approximated through differences of first order data – and will be initialised by the identity matrix
- Thus, in order to gain **additional $\boldsymbol{n}$ equations**, we'll impose the equality of the second order derivative of the original Lagrange function and the approximated one!

$$f^k(\boldsymbol{x}) = r_0^k + \sum_{i=1}^{n}\left(\frac{p_{0i}^k}{U_i^k - x_i} + \frac{q_{0i}^k}{x_i - L_i^k}\right)$$ Objective

$$x_i^k - L_i^k = U_i^k - x_i^k = \Delta_i^k$$

MMA – the approximation of the objective function

- Now, the $2n + 1$ equations are ready for computing all the coefficients p_{0i}^k, q_{0i}^k and Δ_i^k:

$$L(\boldsymbol{x}^k, \boldsymbol{\lambda}^k) = f(\boldsymbol{x}^k) + \sum_{j=1}^{m} \lambda_j^k \cdot g_j(\boldsymbol{x}^k) + \sum_{J=m+1}^{m+me} \mu_j^k \cdot h_j(\boldsymbol{x}^k) = \tilde{L}(\boldsymbol{x}^k, \boldsymbol{\lambda}^k)$$

$$\boldsymbol{\nabla_x L}(\boldsymbol{x}^k, \boldsymbol{\lambda}^k) = \boldsymbol{\nabla_x f}(\boldsymbol{x}^k) + \sum_{j=1}^{m} \lambda_j^k \cdot \boldsymbol{\nabla_x g_j}(\boldsymbol{x}^k) + \sum_{J=m+1}^{m+me} \mu_j^k \cdot \boldsymbol{\nabla_x h_j}(\boldsymbol{x}^k) = \boldsymbol{\nabla_x \tilde{L}}(\boldsymbol{x}^k, \boldsymbol{\lambda}^k)$$

$$\boldsymbol{\nabla_x^2 L}(\boldsymbol{x}^k, \boldsymbol{\lambda}^k) = \boldsymbol{\nabla_x^2 f}(\boldsymbol{x}^k) + \sum_{j=1}^{m} \lambda_j^k \cdot \boldsymbol{\nabla_x^2 g_j}(\boldsymbol{x}^k) + \sum_{j=m+1}^{m+me} \mu_j^k \cdot \boldsymbol{\nabla_x^2 h_j}(\boldsymbol{x}^k) = \boldsymbol{\nabla_x^2 \tilde{L}}(\boldsymbol{x}^k, \boldsymbol{\lambda}^k)$$

- Inserting the formula for the approximation into the equations leads to:
  - The first equation provides the constant

$$r_0^k = \tilde{f}(\boldsymbol{x}^k) - \sum_{i=1}^{n}\left(\frac{p_{0i}^k}{U_i^k - x_i^k} + \frac{q_{0i}^k}{x_i^k - L_i^k}\right)$$

  - The second and third:

$$\nabla_{\boldsymbol{x}}\tilde{L}(\boldsymbol{x}^k, \boldsymbol{\lambda}^k) = \frac{1}{{\Delta_i^k}^2}\left(p_{0i}^k - q_{0i}^k + \sum_{j=1}^{m} \lambda_j^k \cdot (p_{ji}^k - q_{ji}^k) + \sum_{j=m+1}^{m+me} \mu_j^k \cdot 2p_{ji}^k\right)$$

$$\nabla_{\boldsymbol{x}}^2\tilde{L}(\boldsymbol{x}^k, \boldsymbol{\lambda}^k) = \frac{2}{{\Delta_i^k}^3}\left(p_{0i}^k + q_{0i}^k + \sum_{j=1}^{m} \lambda_j^k \cdot (p_{ji}^k + q_{ji}^k) + 0\right)$$

Curvature of equality constraints at initialisation point

$\nabla_{\boldsymbol{x}}^2\tilde{L}(\boldsymbol{x}^k, \boldsymbol{\lambda}^k)$ is approximated with finite differencing: $\nabla_{\boldsymbol{x}}^2\tilde{L}(\boldsymbol{x}^k, \boldsymbol{\lambda}^k) \approx \nabla_{\boldsymbol{x}}\tilde{L}(\boldsymbol{x}^k, \boldsymbol{\lambda}^k) - \nabla_{\boldsymbol{x}}\tilde{L}(\boldsymbol{x}^{k-1}, \boldsymbol{\lambda}^k)$

$$f^k(\boldsymbol{x}) = r_0^k + \sum_{i=1}^{n} \left(\frac{p_{0i}^k}{U_i^k - x_i} + \frac{q_{0i}^k}{x_i - L_i^k} \right)$$ Objective

MMA – the approximation of the objective function

- To simplify the notation, we denote

$$L' = \nabla_{\boldsymbol{x}} \tilde{L}(\boldsymbol{x}^k, \boldsymbol{\lambda}^k)$$
$$L'' = \nabla_{\boldsymbol{x}}^2 \tilde{L}(\boldsymbol{x}^k, \boldsymbol{\lambda}^k)$$

- And reshape the two corresponding equations (2 and 3):

$$\left.\begin{array}{ll} p_{0i}^k = \frac{1}{4}{\Delta_i^k}^3 L'' + \frac{1}{2}{\Delta_i^k}^2 L' - \sum_{j=1}^{m} \lambda_j^k \cdot p_{ji}^k - \sum_{j=m+1}^{m+me} \mu_j^k \cdot p_{ji}^k & \rightarrow {\Delta_i^k}^1 \\ q_{0i}^k = \frac{1}{4}{\Delta_i^k}^3 L'' - \frac{1}{2}{\Delta_i^k}^2 L' - \sum_{j=1}^{m} \lambda_j^k \cdot q_{ji}^k + \sum_{j=m+1}^{m+me} \mu_j^k \cdot p_{ji}^k & \rightarrow {\Delta_i^k}^2 \end{array}\right\} \Delta_i^k = \max({\Delta_i^k}^1, {\Delta_i^k}^2)$$

- For the final solution, we need to impose the necessary conditions for preserving the positive definiteness of the objective function (and accordingly of the Lagrange function):

$p_{0i}^k \geq 0$ and $q_{0i}^k \geq 0$

- Through imposing the positive definiteness conditions and solving the two cubic equations, we get the lower bound for Δ_i^k
- However, choosing Δ_i^k as the bigger value of the two solutions, will lead to one of the coefficient p_{0i}^k or q_{0i}^k being zero – with drawback of having no explicit minimum of the approximated objective between the asymptotes

$$f^k(\boldsymbol{x}) = r_0^k + \sum_{i=1}^{n}\left(\frac{p_{0i}^k}{U_i^k - x_i} + \frac{q_{0i}^k}{x_i - L_i^k}\right)$$ Objective

MMA – the approximation of the objective function

- To guarantee both $p_{0i}^k > 0$ and $q_{0i}^k > 0$ (strictly positive), we choose

  $\Delta_i^k = S_F \cdot \max\left(\Delta_i^{k^1}, \Delta_i^{k^2}\right)$

  $S_F \cdots$ scaling factor (usually, $S_F = 10$ is a good choice)

- After determining half the asymptotes distance Δ_i^k, it is possible to calculate the asymptotes:

  $L_i^k = x_i^k - \Delta_i^k$

  $U_i^k = x_i^k + \Delta_i^k$

  and the coefficients of the objective function

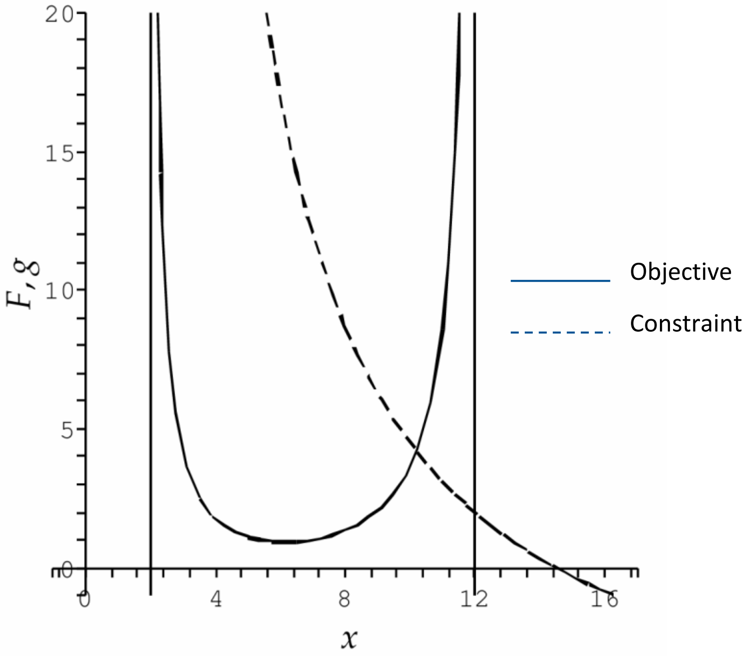
  $p_{0i}^k = \frac{1}{4}\Delta_i^{k^3} L'' + \frac{1}{2}\Delta_i^{k^2} L' - \sum_{j=1}^{m} \lambda_j^k \cdot p_{ji}^k - \sum_{j=m+1}^{m+me} \mu_j^k \cdot p_{ji}^k$

  $q_{0i}^k = \frac{1}{4}\Delta_i^{k^3} L'' - \frac{1}{2}\Delta_i^{k^2} L' - \sum_{j=1}^{m} \lambda_j^k \cdot q_{ji}^k + \sum_{j=m+1}^{m+me} \mu_j^k \cdot p_{ji}^k$

$$f^k(\boldsymbol{x}) = r_0^k + \sum_{i=1}^{n}\left(\frac{p_{0i}^k}{U_i^k - x_i} + \frac{q_{0i}^k}{x_i - L_i^k}\right)$$ Objective

MMA – the approximation of the objective function

- There is another very important set of conditions that affect the selection of the parameter Δ_i^k namely the size of the feasible domain
- By choosing Δ_i^k the position and the distance between asymptotes L_i^k and U_i^k is fixed, however – and this happens occasionally during the iterations – this might lead to an approximation of the inequality constraints with the roots (and accordingly the feasible domain) outside the asymptotes – leading to an infeasible optimisation problem
- To remedy this problem, all the roots of – at least – of the positive (violated) inequality constraints need to be computed and checked to be inside the asymptotes
- In case the root is outside the asymptotes, Δ_i^k is extended to contain the corresponding root



MMA – the approximation of the objective function

- Here‘s an example for an objective function:

$$\boldsymbol{x}^k = \begin{bmatrix} -1.0 \\ 3.5 \end{bmatrix}$$

$$\tilde{f}(\boldsymbol{x}^k) = 10$$

$$\frac{\partial \tilde{f}(\boldsymbol{x}^k)}{\partial x_1} = 4.2$$

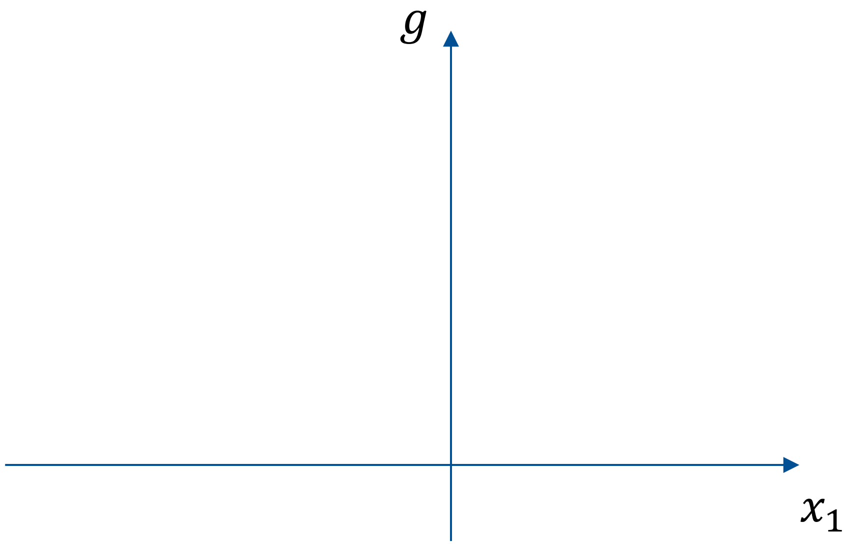
$$\frac{\partial \tilde{f}(\boldsymbol{x}^k)}{\partial x_1} = -1.4$$

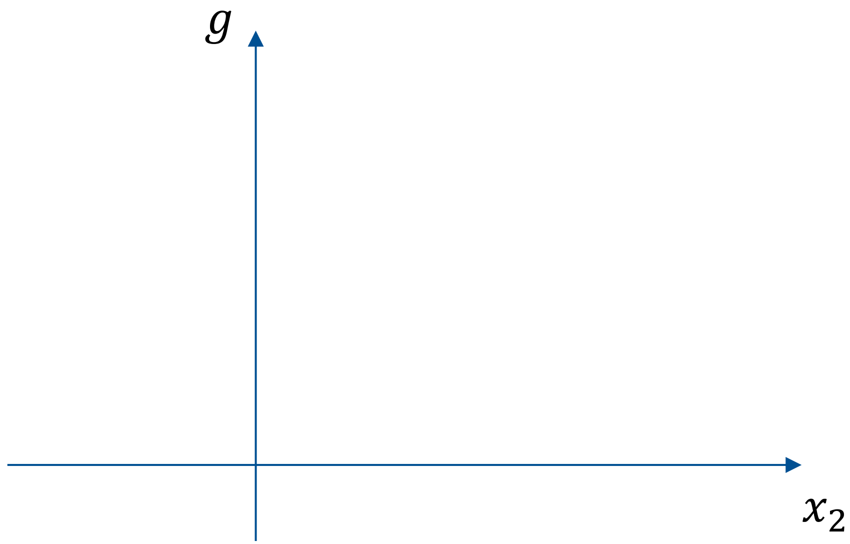
$$L_1^k = -6.0$$

$$U_1^k = 4.0$$

$$L_2^k = -3.5$$

$$U_2^k = 10.5$$





DEMMA – Dual Extended Method of Moving Asymptotes

- The Dual algorithms is based on the Lagrange function:

$$L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\boldsymbol{x}) + \sum_{j=1}^{m} \lambda_j \cdot g_j(\boldsymbol{x}) + \sum_{j=m+1}^{m+m^e} \mu_j \cdot h_j(\boldsymbol{x})$$

- Inserting the MMA approximation in the Lagrange function yields:

$$L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = r_0^k + \sum_{i=1}^{n} \left(\frac{p_{0i}^k}{U_i^k - x_i} + \frac{q_{0i}^k}{x_i - L_i^k} \right) +$$

$$\sum_{j=1}^{m} \lambda_j \cdot \left(r_j^k + \sum_{i=1}^{n} \left(\frac{p_{ji}^k}{U_i^k - x_i}\, OR\, \frac{q_{ji}^k}{x_i - L_i^k} \right) \right) + \sum_{j=m+1}^{m+m^e} \mu_j \cdot \left(r_j^k + \sum_{i=1}^{n} \left(\frac{p_{ji}^k}{U_i^k - x_i} - \frac{p_{ji}^k}{x_i - L_i^k} \right) \right)$$

- According to the Dual algorithm:
  – Minimise in primal space
  – Maximise in dual space

DEMMA – KKT in primal space

- Solution in Primal Space:

  Applying the Karush Kuhn Tucker condition in primal space leads to a recursion function $x_i(\boldsymbol{\lambda}, \boldsymbol{\mu})$:

  $$\nabla_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = 0$$

  $$\Rightarrow \frac{p_{0i}}{(U_i - x_i)^2} - \frac{q_{0i}}{(x_i - L_i)^2} + \sum_{j=1}^{m} \lambda_j \cdot \left(\frac{p_{ji}}{(U_i - x_i)^2} - \frac{q_{ji}}{(x_i - L_i)^2} \right) + \sum_{j=m+1}^{m+m^e} \mu_j \cdot \left(\frac{p_{ji}}{(U_i - x_i)^2} + \frac{p_{ji}}{(x_i - L_i)^2} \right) = 0$$

- The equation can be resolved for the primal variable x_i as a function of the dual variables λ_j and μ_j:

  $$x_i(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \frac{U_i \cdot \sqrt{Q_i} + L_i \cdot \sqrt{P_i}}{\sqrt{Q_i} + \sqrt{P_i}}$$

  with

  $P_i = p_{0i} + \sum_{j=1}^{m} \lambda_j \cdot p_{ji} + \sum_{j=m+1}^{m+m^e} \mu_j \cdot p_{ji}$

  $Q_i = q_{0i} + \sum_{j=1}^{m} \lambda_j \cdot q_{ji} - \sum_{j=m+1}^{m+m^e} \mu_j \cdot p_{ji}$

DEMMA – KKT in dual space

- Solution in Dual Space:
  - After minimisation in primal space, we need to maximise the dual function with respect to $\boldsymbol{\lambda}$ ($\boldsymbol{\lambda}, \boldsymbol{\mu}$ are not distinguished)
  - The Dual function (only function in $\boldsymbol{\lambda}$) is iteratively maximised with respect to $\boldsymbol{\lambda}$ by the general update formula: $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \alpha \cdot \mathbf{S}^k$
  - The search direction $\boldsymbol{S}^k$ can be determined by any of the previously elaborated methods – e.g. a full Newton
    - Why it is possible to use a full Newton instead of a quasi-Newton?
  - The optimal step size is determined with a Line Search
    - Either an inexact Line Search according to Wolfe, or
    - An exact Line Search based on a quadratic approximation – see next page!

DEMMA – Convergence

- Convergence of the MMA solution (subproblem)
  - KKT with MMA functions
- Convergence of original problem
  - KKT in original problem space

DEMMA – KKT in dual space – step size

- Solution in Dual Space – step size optimisation:
  - Since all first and second order information are easily accessible, it is possible to calculate the optimal step size based on an approximation second order of the Dual function:

$$L(\boldsymbol{\lambda}^k + \alpha \cdot \boldsymbol{S}^k) = L(\boldsymbol{\lambda}^k) + \alpha \cdot \nabla_{\boldsymbol{\lambda}} \boldsymbol{L} \cdot \boldsymbol{S}^k + \frac{1}{2}\alpha^2 \cdot \boldsymbol{S}^k \cdot \nabla^2_{\boldsymbol{\lambda}} \boldsymbol{L} \cdot \boldsymbol{S}^k$$

  - To obtain the optimal step size the first order derivative with respect to α is set to zero:

    $\nabla_\alpha L = \nabla_{\boldsymbol{\lambda}} \boldsymbol{L} \cdot \boldsymbol{S}^k + \alpha \cdot \boldsymbol{S}^k \cdot \nabla^2_{\boldsymbol{\lambda}} \boldsymbol{L} \cdot \boldsymbol{S}^k = 0 \Rightarrow$

$$\alpha_{opt} = -\frac{\nabla_{\boldsymbol{\lambda}} \boldsymbol{L} \cdot \boldsymbol{S}^k}{\boldsymbol{S}^k \cdot \nabla^2_{\boldsymbol{\lambda}} \boldsymbol{L} \cdot \boldsymbol{S}^k}$$

    - What is the sign of α_{opt}?
  - To calculate the optimal step size, the first and second order derivative of the Dual function with respect to $\boldsymbol{\lambda}$ must be determined.

Department of Aerospace and Geodesy
TUM School of Engineering and Design
Technical University of Munich

DEMMA – KKT in dual space – step size

- The first order derivative of the Dual function with respect to $\boldsymbol{\lambda}$ has been already elaborated:

$\nabla_{\lambda} \boldsymbol{L} = \boldsymbol{g}(\boldsymbol{x})$ or in component notation $\frac{\partial L}{\partial \lambda_j} = g_j(\boldsymbol{x})$

$\nabla_{\mu} \boldsymbol{L} = \boldsymbol{h}(\boldsymbol{x})$ or in component notation $\frac{\partial L}{\partial \mu_j} = h_j(\boldsymbol{x})$

- The second order derivative requires some more math:

$j \in [1, m]$: $$\boldsymbol{\nabla}_{\lambda}^2 \boldsymbol{L} = \frac{\partial^2 L}{\partial \lambda_j \partial \lambda_k} = \boldsymbol{\nabla}_{\lambda} \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{\nabla}_{x} \boldsymbol{g}(\boldsymbol{x}) \cdot \boldsymbol{\nabla}_{\lambda} \boldsymbol{x}$$

$$\frac{\partial^2 L}{\partial \lambda_j \partial \lambda_k} = \frac{\partial g_j}{\partial \lambda_k} = \frac{\partial g_j}{\partial \boldsymbol{x}} \cdot \frac{\partial \boldsymbol{x}}{\partial \lambda_k}$$

$j \in [m+1, m+me]$: $$\boldsymbol{\nabla}_{\mu}^2 \boldsymbol{L} = \boldsymbol{\nabla}_{\mu} \boldsymbol{h}(\boldsymbol{x}^k) = \boldsymbol{\nabla}_{x} \boldsymbol{h}(\boldsymbol{x}^k) \cdot \boldsymbol{\nabla}_{\mu} \boldsymbol{x}$$

$$\frac{\partial^2 L}{\partial \mu_j \partial \mu_k} = \frac{\partial h_j}{\partial \mu_k} = \frac{\partial h_j}{\partial \boldsymbol{x}} \cdot \frac{\partial \boldsymbol{x}}{\partial \mu_k}$$

  - The derivative components $\frac{\partial g_j}{\partial \boldsymbol{x}}$ and $\frac{\partial h_j}{\partial \boldsymbol{x}}$ are determined by the sensitivity analysis in the primal space
  - The derivative components $\frac{\partial \boldsymbol{x}}{\partial \lambda_k}$ and $\frac{\partial \boldsymbol{x}}{\partial \mu_k}$ can be calculated based on the MMA approximation – see next page

$g_j(\boldsymbol{x}),\ h_j(\boldsymbol{x}),\ \frac{\partial g_j}{\partial \boldsymbol{x}}, \frac{\partial \boldsymbol{x}}{\partial \lambda_k}$
all MMA functions NOT original ones

$P_i = p_{0i} + \sum_{j=1}^{m} \lambda_j \cdot p_{ji} + \sum_{j=m+1}^{m+m^e} \mu_j \cdot p_{ji}$
$Q_i = q_{0i} + \sum_{j=1}^{m} \lambda_j \cdot q_{ji} - \sum_{j=m+1}^{m+m^e} \mu_j \cdot p_{ji}$

DEMMA – KKT in dual space – step size

- Based on the (recursion) function $x_i(\boldsymbol{\lambda}, \boldsymbol{\mu})$:

$$x_i(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \frac{U_i \cdot \sqrt{Q_i} + L_i \cdot \sqrt{P_i}}{\sqrt{Q_i} + \sqrt{P_i}}$$

  we can derive with respect to $\boldsymbol{\lambda}$ (and $\boldsymbol{\mu}$):

$$\frac{\partial x_i}{\partial \lambda_k} = \frac{\partial x_i}{\partial Q_i} \cdot \frac{\partial Q_i}{\partial \lambda_k} + \frac{\partial x_i}{\partial P_i} \cdot \frac{\partial P_i}{\partial \lambda_k}$$

- Deriving the (recursion) function $x_i(\boldsymbol{\lambda}, \boldsymbol{\mu})$ with respect to Q_i and P_i provides:

$$\frac{\partial x_i}{\partial P_i} = -\frac{1}{2} \frac{\sqrt{Q_i} \cdot (U_i - L_i)}{\sqrt{P_i} \cdot \left(\sqrt{Q_i} + \sqrt{P_i}\right)^2}$$

$$\frac{\partial x_i}{\partial Q_i} = -\frac{1}{2} \frac{\sqrt{P_i} \cdot (U_i - L_i)}{\sqrt{Q_i} \cdot \left(\sqrt{Q_i} + \sqrt{P_i}\right)^2}$$

- And finally deriving Q_i and P_i with respect to $\boldsymbol{\lambda}$ (and $\boldsymbol{\mu}$) yields:

$\frac{\partial P_i}{\partial \lambda_k} = p_{ki}$ same for $\boldsymbol{\mu}$: $\frac{\partial P_i}{\partial \mu_k} = p_{ki}$

$$\frac{\partial Q_i}{\partial \lambda_k} = q_{ki}$$

$$\frac{\partial Q_i}{\partial \mu_k} = -p_{ki}$$

DEMMA – KKT in dual space – search direction

- What is still missing is the search direction in the dual space $\boldsymbol{S}^k(\boldsymbol{\lambda}, \boldsymbol{\mu})$
- Any of the previously presented methods can be used. Within this lecture a modified version of the Conjugate Gradients will be utilised – the reason will be shown during the derivation procedure
- The Search direction can be calculated by applying the typical formula of the conjugate gradient method

$$\boldsymbol{S}^{k+1} = \nabla_{\boldsymbol{\lambda}} \boldsymbol{L} + \beta^k \cdot \boldsymbol{S}^k$$

Where is the ⊖ (negative sign)

- Based on the theory of the method of conjugate gradients, the conjugation parameter is determined by:

$$\boldsymbol{S}^k \cdot \boldsymbol{\nabla}_{\boldsymbol{\lambda}}^2 \boldsymbol{L} \cdot \boldsymbol{S}^{k+1} = 0$$

I.e., the subsequent search directions are Hessian conjugate

- Inserting the update formula for $\boldsymbol{S}^{k+1}$ into the conjugation equation provides the conjugation parameter:

$$\boldsymbol{S}^k \cdot \boldsymbol{\nabla}_{\boldsymbol{\lambda}}^2 \boldsymbol{L} \cdot \left(\nabla_{\boldsymbol{\lambda}} \boldsymbol{L} + \beta^k \cdot \boldsymbol{S}^k\right) = 0 \Rightarrow$$

$$\boldsymbol{S}^k \cdot \boldsymbol{\nabla}_{\boldsymbol{\lambda}}^2 \boldsymbol{L} \cdot \nabla_{\boldsymbol{\lambda}} \boldsymbol{L} + \beta^k \cdot \boldsymbol{S}^k \cdot \boldsymbol{\nabla}_{\boldsymbol{\lambda}}^2 \boldsymbol{L} \cdot \boldsymbol{S}^k = 0 \Rightarrow$$

$$\boxed{\beta^k = -\frac{\boldsymbol{S}^k \cdot \boldsymbol{\nabla}_{\boldsymbol{\lambda}}^2 \boldsymbol{L} \cdot \nabla_{\boldsymbol{\lambda}} \boldsymbol{L}}{\boldsymbol{S}^k \cdot \boldsymbol{\nabla}_{\boldsymbol{\lambda}}^2 \boldsymbol{L} \cdot \boldsymbol{S}^k}}$$

DEMMA – Summary

- Cooking Recipe:
  - Establish the MMA approximation

$$f^k(\boldsymbol{x}) = r_0^k + \sum_{i=1}^{n}\left(\frac{p_{0i}^k}{U_i^k - x_i} + \frac{q_{0i}^k}{x_i - L_i^k}\right) \quad \text{Objective}$$

$$g_j^k(\boldsymbol{x}) = r_j^k + \sum_{i=1}^{n}\left(\frac{p_{ji}^k}{U_i^k - x_i}\, OR\, \frac{q_{ji}^k}{x_i - L_i^k}\right) \quad \text{Inequality constraints}$$

$$h_j^k(\boldsymbol{x}) = r_j^k + \sum_{i=1}^{n}\left(\frac{p_{ji}^k}{U_i^k - x_i} - \frac{p_{ji}^k}{x_i - L_i^k}\right) \quad \text{Equality constraints}$$

  - Minimise in $\boldsymbol{x}$:

$$x_i(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \frac{U_i \cdot \sqrt{Q_i} + L_i \cdot \sqrt{P_i}}{\sqrt{Q_i} + \sqrt{P_i}}$$

  - Maximise in $\boldsymbol{\lambda}$:

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \alpha \cdot \boldsymbol{S}^k \qquad \alpha_{opt} = -\frac{\nabla_\lambda \boldsymbol{L} \cdot \boldsymbol{S}^k}{\boldsymbol{S}^k \cdot \nabla_\lambda^2 \boldsymbol{L} \cdot \boldsymbol{S}^k} \qquad \boldsymbol{S}^{k+1} = \nabla_\lambda \boldsymbol{L} + \beta^k \cdot \boldsymbol{S}^k \qquad \beta^k = -\frac{\boldsymbol{S}^k \cdot \nabla_\lambda^2 \boldsymbol{L} \cdot \nabla_\lambda \boldsymbol{L}}{\boldsymbol{S}^k \cdot \nabla_\lambda^2 \boldsymbol{L} \cdot \boldsymbol{S}^k}$$

$$\left.\begin{array}{ll} \frac{1}{4}{\Delta_i^k}^3 L'' + \frac{1}{2}{\Delta_i^k}^2 L' - \sum_{j=1}^{m}\lambda_j^k \cdot p_{ji}^k - \sum_{j=m+1}^{m+me}\mu_j^k \cdot p_{ji}^k = 0 & \rightarrow {\Delta_i^k}^1 \\ \frac{1}{4}{\Delta_i^k}^3 L'' - \frac{1}{2}{\Delta_i^k}^2 L' - \sum_{j=1}^{m}\lambda_j^k \cdot q_{ji}^k + \sum_{j=m+1}^{m+me}\mu_j^k \cdot p_{ji}^k = 0 & \rightarrow {\Delta_i^k}^2 \end{array}\right\} \Delta_i^k = \max({\Delta_i^k}^1, {\Delta_i^k}^2)$$

$$L_i^k = x_i^k - \Delta_i^k$$
$$U_i^k = x_i^k + \Delta_i^k$$

$$p_{0i}^k = \frac{1}{4}{\Delta_i^k}^3 L'' + \frac{1}{2}{\Delta_i^k}^2 L' - \sum_{j=1}^{m}\lambda_j^k \cdot p_{ji}^k - \sum_{j=m+1}^{m+me}\mu_j^k \cdot p_{ji}^k$$
$$q_{0i}^k = \frac{1}{4}{\Delta_i^k}^3 L'' - \frac{1}{2}{\Delta_i^k}^2 L' - \sum_{j=1}^{m}\lambda_j^k \cdot q_{ji}^k + \sum_{j=m+1}^{m+me}\mu_j^k \cdot p_{ji}^k$$

$$p_{ji}^k = \begin{cases} (U_i^k - x_i^k)^2 \frac{\partial g_j}{\partial x_i}, & \text{if } \frac{\partial g_j}{\partial x_i} > 0 \\ 0, & \text{if } \frac{\partial g_j}{\partial x_i} \le 0 \end{cases} \qquad q_{ji}^k = \begin{cases} 0, & \text{if } \frac{\partial g_j}{\partial x_i} > 0 \\ -(x_i^k - L_i^k)^2 \frac{\partial g_j}{\partial x_i}, & \text{if } \frac{\partial g_j}{\partial x_i} \le 0 \end{cases}$$

$$r_j^k = \tilde{g}_j(\boldsymbol{x}^k) - \sum_{i=1}^{n}\left(\frac{p_{ji}^k}{U_i^k - x_i^k}\, OR\, \frac{q_{ji}^k}{x_i^k - L_i^k}\right)$$

$$p_{ji}^k = \frac{1}{2}(U_i^k - x_i^k)^2 \cdot \frac{\partial \tilde{h}_j(x_i^k)}{\partial x_i}$$
$$r_j^k = \tilde{h}_j(\boldsymbol{x}^k)$$

$$P_i = p_{0i} + \sum_{j=1}^{m}\lambda_j \cdot p_{ji} + \sum_{j=m+1}^{m+m^e}\mu_j \cdot p_{ji}$$
$$Q_i = q_{0i} + \sum_{j=1}^{m}\lambda_j \cdot q_{ji} - \sum_{j=m+1}^{m+m^e}\mu_j \cdot q_{ji}$$

DEMMA – Summary

- The first order derivative of the dual function:

$$\frac{\partial L}{\partial \lambda_j} = g_j(\boldsymbol{x}) \qquad \frac{\partial L}{\partial \mu_j} = h_j(\boldsymbol{x})$$

- The second order derivative of the dual function:

$$\frac{\partial^2 L}{\partial \lambda_j \partial \lambda_k} = \frac{\partial g_j}{\partial \lambda_k} = \frac{\partial g_j}{\partial \boldsymbol{x}} \cdot \frac{\partial \boldsymbol{x}}{\partial \lambda_k}$$

(identical formulas for the inequality constraints)

$$\frac{\partial x_i}{\partial \lambda_k} = \frac{\partial x_i}{\partial Q_i} \cdot \frac{\partial Q_i}{\partial \lambda_k} + \frac{\partial x_i}{\partial P_i} \cdot \frac{\partial P_i}{\partial \lambda_k}$$

$$\frac{\partial x_i}{\partial P_i} = -\frac{1}{2} \frac{\sqrt{Q_i} \cdot (U_i - L_i)}{\sqrt{P_i} \cdot \left(\sqrt{Q_i} + \sqrt{P_i}\right)^2}$$

$$\frac{\partial x_i}{\partial Q_i} = -\frac{1}{2} \frac{\sqrt{P_i} \cdot (U_i - L_i)}{\sqrt{Q_i} \cdot \left(\sqrt{Q_i} + \sqrt{P_i}\right)^2}$$

$\frac{\partial P_i}{\partial \lambda_k} = p_{ki}$ same for $\boldsymbol{\mu}$: $\frac{\partial P_i}{\partial \mu_k} = p_{ki}$

$$\frac{\partial Q_i}{\partial \lambda_k} = q_{ki}$$

$$\frac{\partial Q_i}{\partial \mu_k} = -p_{ki}$$