Nonlinear Optimization

Approaches and Challenges

Dr. Helga Ingimundardóttir ⟨helgaingim@hi.is⟩

VÉL113F: Design and Optimization

INDUSTRIAL ENGINEERING

UNIVERSITY OF ICELAND

Nonlinear optimization

Unlike linear optimization, nonlinear optimization deals with problems where the objective function **or** the constraints are nonlinear functions, making the solutions more complex.

Basic Concepts

UNIVERSITY OF ICELAND

Definition:

Nonlinear optimization focuses on finding the best solution to a problem where either the objective function, the constraints, or both are nonlinear.

- **Objective Function:** Function that needs to be optimized, which can be either maximized or minimized.
- **Constraints:** Restrictions or limitations imposed on the solution.
- **Feasible Region:** Set of all solutions that satisfy the constraints.
- **Optimal Solution:** Best possible solution from the set of feasible solutions.
- **Local vs. Global Optima:** Understanding the difference between local optima (best solution in a neighborhood) and global optima (best solution overall).

Basic Concepts II

- **Convexity:** A function is convex if the line segment between any two points on the function lies above or on the function. Important for the convergence of certain algorithms.
- **Gradient and Hessian:** First and second derivatives, respectively, essential in determining the nature of the optimum (minimum, maximum, or saddle point).

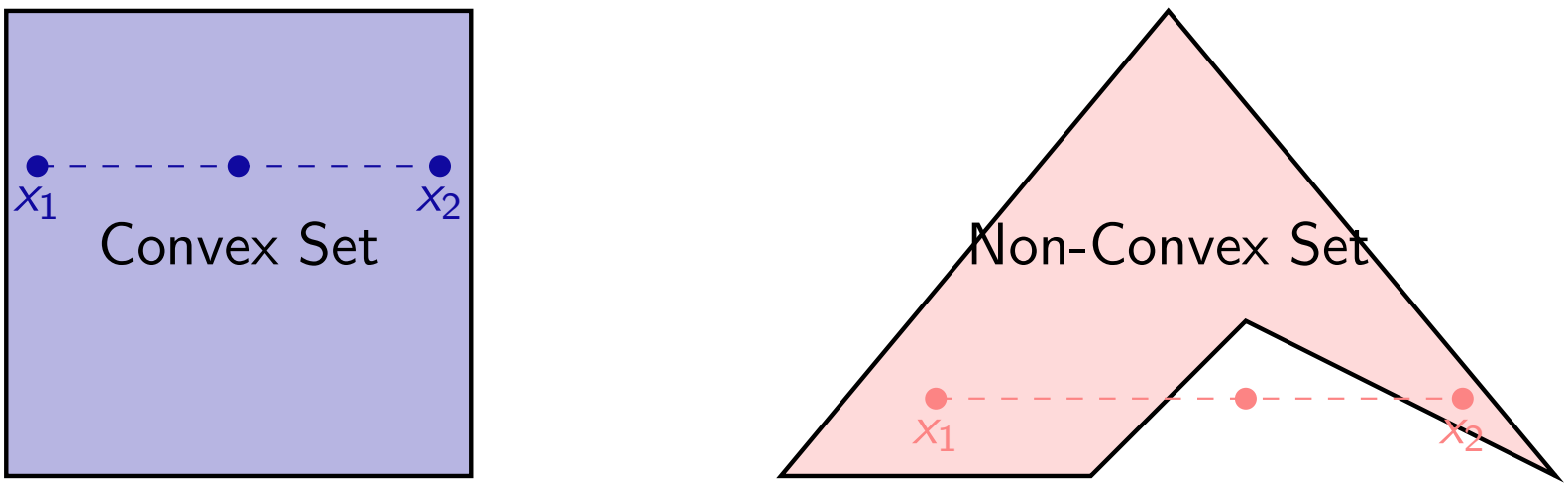
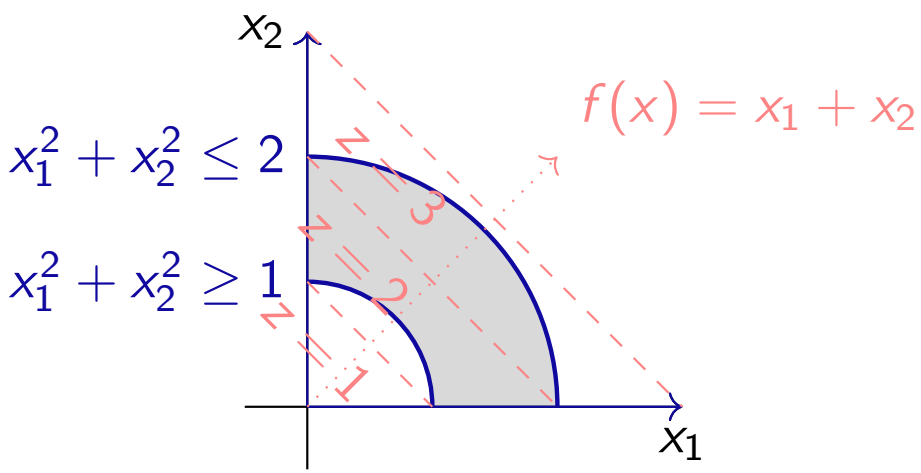


Figure: Convex and non-convex sets.

Example of Non-linear Optimization

$$
\begin{aligned}
\text{maximize} \quad & x_1 + x_2 \\
\text{subject to} \quad & x_1 \geq 0, x_2 \geq 0 \\
& x_1^2 + x_2^2 \geq 1 \\
& x_1^2 + x_2^2 \leq 2
\end{aligned}
$$



Constraints

Objective Function

Feasible region

Method of Lagrange Multipliers

Technique to find the extrema of a function $f(\mathbf{x})$ where $\mathbf{x} \in \mathbb{R}^n$, subject to constraints:

$$\text{optimize } f(\mathbf{x}) \text{ subject to } g(\mathbf{x}) = 0$$

The necessary condition for an extremum at $\mathbf{x}^*$ is that the gradient vectors of f and g are parallel at $\mathbf{x}^*$:

$$\nabla f(\mathbf{x}^*) = \lambda \nabla g(\mathbf{x}^*)$$

This leads to a system of equations: $\frac{\partial f}{\partial x_i} - \lambda \frac{\partial g}{\partial x_i} = 0, \quad \text{for } i = 1, \ldots, n.$
With the additional constraint $g(\mathbf{x}) = 0$.

Specific $n = 2$ case: For the function $f(x_1, x_2)$ with constraint $g(x_1, x_2) = 0$:

$$\left(\frac{\partial f}{\partial x_1} - \lambda \frac{\partial g}{\partial x_1} \right) \Bigg|_{(x_1^*, x_2^*)} = 0$$

$$\left(\frac{\partial f}{\partial x_2} - \lambda \frac{\partial g}{\partial x_2} \right) \Bigg|_{(x_1^*, x_2^*)} = 0$$

Along with:

$$g(x_1^*, x_2^*) = 0$$

For a solution to exist, the partial derivatives must be non-zero to define λ.

Lagrangian Function

Continuing from the Method of Lagrange Multipliers, a more general approach involves the use of the **Lagrangian function**, denoted by L:

$$L(x_1, x_2, \lambda) = f(x_1, x_2) + \lambda g(x_1, x_2)$$

To find the conditions for an extremum, consider L as a function of the variables x_1, x_2, and λ. The necessary conditions are derived by taking the partial derivatives of L with respect to each variable and setting them to zero:

$$\begin{aligned}
\frac{\partial L}{\partial x_1}(x_1, x_2, \lambda) &= \frac{\partial f}{\partial x_1}(x_1, x_2) + \lambda \frac{\partial g}{\partial x_1}(x_1, x_2) &&= 0 \\
\frac{\partial L}{\partial x_2}(x_1, x_2, \lambda) &= \frac{\partial f}{\partial x_2}(x_1, x_2) + \lambda \frac{\partial g}{\partial x_2}(x_1, x_2) &&= 0 \\
\frac{\partial L}{\partial \lambda}(x_1, x_2, \lambda) &= g(x_1, x_2) &&= 0
\end{aligned}$$

Lagrangian Function II

For the general problem with n variables and m constraints:

$$\min f(\boldsymbol{x}) \quad \text{subject to} \quad g_j(\boldsymbol{x}) = 0, \quad j = 1, \ldots, m$$

In this scenario, the Lagrangian function, L, incorporates a Lagrange multiplier, λ_j, for each constraint $g_j(\boldsymbol{x})$:

$$L(x_1, x_2, \ldots, x_n, \lambda_1, \lambda_2, \ldots, \lambda_m) = f(\boldsymbol{x}) + \sum_{j=1}^{m} \lambda_j g_j(\boldsymbol{x})$$

Lagrangian Function III

By considering L as a function of the combined $n + m$ unknowns:

$$x_1, x_2, \ldots, x_n, \lambda_1, \lambda_2, \ldots, \lambda_m$$

the necessary conditions for L to attain an extremum, which mirror the conditions for the original problem, are:

$$\frac{\partial L}{\partial x_i} = \frac{\partial f}{\partial x_i} + \sum_{j=1}^{m} \lambda_j \frac{\partial g_j}{\partial x_i} = 0, \quad i = 1, \ldots, n$$

$$\frac{\partial L}{\partial \lambda_j} = g_j(\boldsymbol{x}) = 0, \quad j = 1, \ldots, m$$

These equations represent a system of $n + m$ equations in the $n + m$ unknowns x_i and λ_j.

The solution yields:

$$\boldsymbol{x}^* = \begin{bmatrix} x_1^* \\ x_2^* \\ \vdots \\ x_n^* \end{bmatrix} \quad \text{and} \quad \boldsymbol{\lambda}^* = \begin{bmatrix} \lambda_1^* \\ \lambda_2^* \\ \vdots \\ \lambda_m^* \end{bmatrix}$$

The vector $\boldsymbol{x}^*$ corresponds to the relative constrained minimum of $f(\boldsymbol{x})$ (sufficient conditions need verification), while the vector $\boldsymbol{\lambda}^*$ provides the sensitivity information.

Lagrangian Function V

UNIVERSITY OF ICELAND

Sufficient condition: For $f(\boldsymbol{x})$ to exhibit a relative minimum at $\boldsymbol{x}^*$, the quadratic Q, defined as:

$$Q = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial^2 L}{\partial x_i \partial x_j} dx_i dx_j$$

must be positive definite at $\boldsymbol{x} = \boldsymbol{x}^*$ for all relevant $d\boldsymbol{x}$ values respecting the constraints.

In the Lagrange multiplier context, Q derived from L's Hessian reflects the local curvature of L. A positive definite Q under the constraints suggests $f(\boldsymbol{x})$ has a relative minimum at the considered point.

Lagrangian Function VI

To determine if the quadratic form Q is positive or negative definite across all admissible variations $d\boldsymbol{x}$, we turn our attention to its roots. Specifically, a necessary condition for Q to be positive (or negative) definite is that every root of the polynomial z_i be positive (or negative). These roots arise from determinantal equations based on the second partial derivatives of the Lagrangian, L, and the constraints, g_i:

$$L_{ij} = \frac{\partial^2 L}{\partial x_i \partial x_j}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$$

$$g_{ij} = \frac{\partial g_i}{\partial x_j}(\boldsymbol{x}^*)$$

Lagrangian Function VII

UNIVERSITY OF ICELAND

To further illustrate, consider the following determinant:

$$\begin{vmatrix} L_{11}-z & L_{12} & L_{13} & \cdots & L_{1n} & g_{11} & g_{12} & \cdots & g_{1m} \\ L_{21} & L_{22}-z & L_{23} & \cdots & L_{2n} & g_{21} & g_{22} & \cdots & g_{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ L_{n1} & L_{n2} & L_{n3} & \cdots & L_{nn}-z & g_{n1} & g_{n2} & \cdots & g_{nm} \\ g_{11} & g_{12} & g_{13} & \cdots & g_{1n} & 0 & 0 & \cdots & 0 \\ g_{21} & g_{22} & g_{23} & \cdots & g_{2n} & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{m1} & g_{m2} & g_{m3} & \cdots & g_{mn} & 0 & 0 & \cdots & 0 \end{vmatrix} = 0$$

This determinant, set to zero, provides the polynomial equations in z whose roots give insight into the definiteness of Q.

UNIVERSITY OF ICELAND

Example

Determine the dimensions of a cylindrical tin (with top and bottom) constructed from sheet metal that maximizes its volume, given that its total surface area is $A_0 = 24\pi$.

Setup:

Variables Let x_1 represent the radius of the base and x_2 denote the height (or length) of the cylinder.

Objective Function Maximize the volume V of a cylinder is given by:

$$f(x_1, x_2) = \pi x_1^2 x_2$$

Constraint The total surface area A_0 of the cylinder is:

$$A(x_1, x_2) = 2\pi x_1^2 + 2\pi x_1 x_2 = A_0 = 24\pi$$

Lagrangian Function IX

Hence, the optimization problem is to maximize $f(x_1, x_2)$ subject to $g(x_1, x_2) = A(x_1, x_2) - 24\pi = 0$.
The **Lagrange function** for this optimization problem is given by:

$$L(x_1, x_2, \lambda) = \pi x_1^2 x_2 + \lambda \left(2\pi x_1^2 + 2\pi x_1 x_2 - A_0\right).$$

Applying the necessary conditions for the extrema, we get:

$$\frac{\partial L}{\partial x_1} = 2\pi x_1 x_2 + 4\pi\lambda x_1 + 2\pi\lambda x_2 = 0$$
$$\implies \lambda = -\frac{x_1 x_2}{2x_1 + x_2}.$$

$$\frac{\partial L}{\partial x_2} = \pi x_1^2 + 2\pi\lambda x_1 = 0$$
$$\implies \lambda = -\frac{x_1}{2} \quad \text{or} \quad x_1 = \frac{x_2}{2}.$$

$$\frac{\partial L}{\partial \lambda} = 2\pi x_1^2 + 2\pi x_1 x_2 - A_0 = 0$$
$$\implies x_1^* = \sqrt{\frac{A_0}{6\pi}}, \quad x_2^* = \sqrt{\frac{2A_0}{3\pi}}, \quad \lambda^* = -\sqrt{\frac{A_0}{24\pi}}, \quad \text{and} \quad f^* = \sqrt{\frac{A_0^3}{54\pi}}$$

Applying the sufficiency conditions for the extrema, we get:

$$
\begin{aligned}
L_{11} &= \frac{\partial^2 L}{\partial x_1^2}\Big|_{(x^*,\lambda^*)} && = 2\pi x_2^* + 4\pi\lambda^* = 4\pi \\
L_{12} &= \frac{\partial^2 L}{\partial x_1 \partial x_2}\Big|_{(x^*,\lambda^*)} = L_{21} && = 2\pi x_1^* + 2\pi\lambda^* = 2\pi \\
L_{22} &= \frac{\partial^2 L}{\partial x_2^2}\Big|_{(x^*,\lambda^*)} && = 0 \\
g_{11} &= \frac{\partial g_1}{\partial x_1}\Big|_{(x^*,\lambda^*)} && = 4\pi x_1^* + 2\pi x_2^* = 16\pi \\
g_{12} &= \frac{\partial g_1}{\partial x_2}\Big|_{(x^*,\lambda^*)} = g_{21} && = 2\pi x_1^* = 4\pi
\end{aligned}
$$

$$
Q = \begin{vmatrix} 4\pi - z & 2\pi & 16\pi \\ 2\pi & 0 - z & 4\pi \\ 16\pi & 4\pi & 0 \end{vmatrix} = 0 \quad \Longrightarrow \quad 272\pi^2 z + 192\pi^3 = 0 \quad \Longrightarrow \quad z = -\frac{12}{17}\pi
$$

Since $z < 0$, then Q is negative definite and the point (x_1^*, x_2^*) is a relative maximum of $f(x_1, x_2)$.

Multivariable optimization with **inequality** constraints is formulated as:

$$\min f(\boldsymbol{x})$$

subject to

$$g_j(\boldsymbol{x}) \leq 0$$

for $j = 1, \ldots, m$.
To address these inequalities, we introduce non-negative slack variables y_j^2 to transform the constraints into equivalent equality constraints:

$$g_j(\boldsymbol{x}) + y_j^2 = 0$$

for $j = 1, \ldots, m$. Here, the exact values of the slack variables remain undetermined.

Reformulation: Minimize $f(\boldsymbol{x})$ subject to

$$G_j(\boldsymbol{x}, \boldsymbol{y}) = g_j(\boldsymbol{x}) + y_j^2 = 0,$$

for $j = 1, 2, \ldots, m$. Here, $\boldsymbol{y} = \begin{bmatrix} y_1 & y_2 & \cdots & y_m \end{bmatrix}^T$ is the vector of slack variables.

The above optimization problem can be effectively addressed using the method of Lagrange multipliers. Define the Lagrangian function as:

$$L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \sum_{j=1}^{m} \lambda_j G_j(\boldsymbol{x}, \boldsymbol{y})$$

Where $\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 & \lambda_2 & \vdots & \lambda_m \end{bmatrix}^T$ is the vector of Lagrange multipliers.

KKT Conditions III

The stationary points of the Lagrange function can be deduced by solving the following equations:

$$\frac{\partial L}{\partial x_i}(\boldsymbol{x}, \boldsymbol{y}, \lambda) = \frac{\partial f}{\partial x_i}(\boldsymbol{x}) + \sum_{j=1}^{m} \lambda_j \frac{\partial g_j}{\partial x_i}(\boldsymbol{x}) = 0, \qquad i = 1, 2, \ldots, n$$

$$\frac{\partial L}{\partial \lambda_j}(\boldsymbol{x}, \boldsymbol{y}, \lambda) = G_j(\boldsymbol{x}, \boldsymbol{y}) = g_j(\boldsymbol{x}) + y_j^2 = 0, \qquad j = 1, 2, \ldots, m$$

$$\frac{\partial L}{\partial y_j}(\boldsymbol{x}, \boldsymbol{y}, \lambda) = 2\lambda_j y_j = 0, \qquad j = 1, 2, \ldots, m$$

This system represents $(n + 2m)$ equations in the $(n + 2m)$ unknowns: $\boldsymbol{x}$, $\boldsymbol{\lambda}$, and $\boldsymbol{y}$. Solving this system provides us with the optimal solution $\boldsymbol{x}^*$, the Lagrange multiplier $\boldsymbol{\lambda}^*$, and the slacks $\boldsymbol{y}^*$.

KKT Conditions IV

Given a constrained minimum point $\boldsymbol{x}^*$, the conditions to be satisfied can be articulated as:

$$\frac{\partial f}{\partial x_i} + \sum_{j \in J_1} \lambda_j \frac{\partial g_j}{\partial x_i} = 0, \qquad i = 1, 2, \ldots, n$$

$$\lambda_j > 0, \qquad j \in J_1$$

where J_1 is the set of indices of the active constraints at $\boldsymbol{x}^*$.
These constraints are referred to as the **Kuhn-Tucker conditions**. They are necessary conditions to be met for $\boldsymbol{x}$ to be a relative minimum of $f(\boldsymbol{x})$.

It's crucial to note that these conditions, while necessary, may not always be sufficient to guarantee a relative minimum.

KKT Conditions for Convex Programming

UNIVERSITY OF ICELAND

For convex programming problems, the Kuhn-Tucker conditions are **both necessary and sufficient** for a global minimum.
If the set of active constraints, J_1, is not known, the Kuhn-Tucker conditions can be stated as:

$$\frac{\partial f}{\partial x_i} + \sum_{j \in J_1} \lambda_j \frac{\partial g_j}{\partial x_i} = 0, \qquad \forall i = 1, 2, \ldots, n$$

$$\lambda_j g_j = 0, \qquad \forall j = 1, 2, \ldots, m$$

$$g_j \leq 0, \qquad \forall j = 1, 2, \ldots, m$$

$$\lambda_j \geq 0, \qquad \forall j = 1, 2, \ldots, m$$

Important Observations: If the problem is maximization or if the constraints are of the type $g_j \geq 0$, the λ_j have to be nonpositive. Conversely, if the problem is one of maximization with constraints of the type $g_j \geq 0$, the λ_j have to be nonnegative.

KKT Conditions for Convex Programming II

UNIVERSITY OF ICELAND

When the optimization problem is stated as:

$$\begin{aligned} &\text{minimize} && f(\boldsymbol{x}) \\ &\text{subject to} && g_j(\boldsymbol{x}) \leq 0, \quad j = 1, 2, \ldots, m \\ &&& h_k(\boldsymbol{x}) = 0, \quad k = 1, \ldots, p \end{aligned}$$

The Kuhn-Tucker conditions become:

$$\nabla f + \sum_{j=1}^{m} \lambda_j \nabla g_j - \sum_{k=1}^{p} \beta_k \nabla h_k = \mathbf{0}$$

$$\begin{aligned} \lambda_j g_j &= 0, \quad j = 1, 2, \ldots, m \\ g_j &\leq 0, \quad j = 1, 2, \ldots, m \\ h_k &= 0, \quad k = 1, 2, \ldots, p \\ \lambda_j &\geq 0, \quad j = 1, 2, \ldots, m \end{aligned}$$

where λ_j and β_k denote the Lagrange multipliers associated with the constraints $g_j \leq 0$ and $h_k = 0$, respectively.

Numerical Methods: Iterative Refinement

Goal: Improve approximations step by step to converge on the optimum solution.

Psuedo-code:

Initialize: Start at a trial point $\boldsymbol{x}_1$.
while optimum not achieved **do**
1. Identify a direction $\boldsymbol{s}_i$ pointing towards the optimum.
2. Determine optimal step length λ_i^* along $\boldsymbol{s}_i$.
3. Update point: $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \lambda_i^* \boldsymbol{s}_i$.
4. Check if $\boldsymbol{x}_{i+1}$ is the optimum.

end while
Terminate: $\boldsymbol{x}_{i+1}$ is the optimum.

The **necessary condition** for $f(\lambda)$ to have a minimum of λ^* is that

$$f'(\lambda^*) = 0.$$

Direct Root Methods

Methods that aim to find the root (or solution) of the equation $f'(\lambda) = 0$.

Three prominent root-finding methods:

- Newton Method
- Quasi-Newton Method
- Secant Method

UNIVERSITY OF ICELAND

Overview

Newton's method, utilizing the Hessian of the function, boasts a quadratic rate of convergence. The method's foundation lies in the second-order Taylors expansion about the current design point.

$$f(\boldsymbol{x} + \Delta\boldsymbol{x}) = f(\boldsymbol{x}) + \boldsymbol{c}^T \Delta\boldsymbol{x} + \frac{1}{2}\Delta\boldsymbol{x}^T \mathbf{H} \Delta\boldsymbol{x}$$

Key Insight: If $\mathbf{H}$ is positive semidefinite, there exists a $\Delta\boldsymbol{x}$ that provides a global minimum for $f(\boldsymbol{x} + \Delta\boldsymbol{x})$.

The optimality conditions $\left(\frac{\delta f}{\delta \mathbf{x}}\right)$ for the Taylor expansion are given by:

$$\mathbf{c} + \mathbf{H}\Delta\mathbf{X} = 0 \tag{1}$$

Assuming $\mathbf{H}$ to be nonsingular, we derive:

$$\mathbf{d} = \Delta\mathbf{X} = -\mathbf{H}^{-1}\mathbf{c} \tag{2}$$

This can be employed to update the design variable:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k \tag{3}$$

Newton's Method III

Historical Context: Originally developed by Newton for tackling nonlinear equations, the method underwent further refinement by Raphson. Consequently, it's often referred to as the **Newton-Raphson method** in numerical analysis literature.

Newton's Method Characteristics

- Requires both first- and second-order derivatives of $f(\lambda)$.
- Given $f''(\lambda_i) \neq 0$, the Newton method possesses a robust convergence property, termed **quadratic convergence**.
- A caveat: If the starting point isn't proximate to the true solution λ^*, the method may diverge.

Newton's Method IV

UNIVERSITY OF ICELAND

A Simplified Analogy:

- Imagine hiking in dense fog, trying to reach the lowest point in a valley.
- Special glasses (Hessian) show the ground's contour, allowing for precise steps.
- Direct but requires more information.

Quasi-Newton Method

UNIVERSITY OF ICELAND

Challenges with Newton's Method:

- Requires the calculation of $\frac{n(n+1)}{2}$ second-order derivatives for the Hessian matrix.
- Faces issues when the Hessian of the function is singular during any iteration.

Quasi-Newton Method's Approach:

- Generates an approximation for the Hessian matrix or its inverse in every iteration.
- Utilizes only the first derivatives of the function to produce these approximations.

Quasi-Newton Method II

UNIVERSITY OF ICELAND

Approximating the Hessian:

- Uses change in design variables.
- Uses change in gradient vectors.

Properties of the Updated Hessian:

- Symmetry is preserved.
- Positive definiteness maintained.

Methods for Approximating the Hessian:

- **DFP method:** Constructs an approximation to the inverse of the Hessian of $f(\mathbf{x})$ using only the first derivatives.
- **BFGS method:** Approximates the Hessian of $f(\mathbf{x})$ itself, rather than its inverse, during each iteration.

A Simplified Analogy:

- Hiking again, but without the glasses.
- Rely on memory and feel of previous steps (approximation to Hessian) to guess the path.
- Intuitive and efficient, continuously updating the understanding with each step.

Marquardt's Modification (1963)

- Combines the desirable features of the steepest descent and Newtons methods.
- Far from the solution point: behaves like the steepest descent method.
- Near the solution point: behaves like Newtons method.

Direction Finding:

$$\mathbf{d}_k = -(\mathbf{H}_k + \lambda_k \mathbf{I})\mathbf{c}_k$$

Update Strategy:

- If the direction $\mathbf{d}_k$ does not reduce the cost function, λ is increased.
- The search direction is recomputed.

Marquardt Method II

UNIVERSITY OF ICELAND

A Simplified Analogy:

- Imagine you're navigating through a dense forest (the problem space). At first, you might just aim to move downhill (steepest descent) to find a clearing or river.
- As you get closer to a river (solution), you might use a map or compass (Newton's method) to direct you precisely to the best path or crossing point.
- If you're unsure, you might recheck your map (recompute direction) or even climb a tree (increase λ) to get a better view.

Core Concepts of the GRG Method

- Designed for problems with nonlinear inequality constraints.
- It determines a path, so when making small adjustments, the set rules (constraints) stay strictly followed.
- If a rule isn't followed precisely due to its complex nature, the method uses the Newton-Raphson technique to get back in line.

A Simplified Analogy:

- Think of it as walking on a tightrope. The GRG method ensures you move without falling (violating constraints). If you're about to fall, it catches you (Newton-Raphson) and places you back on the rope.
- In scenarios where the rules get complicated, it can modify them slightly (by adding slack variables) to make the task manageable.

Introduction to Penalty Functions

- Goal: Solve constrained optimization problems using unconstrained methods.
- Approach: Combine cost and constraint functions to form a **composite function**.
- Introduce the **penalty function** P using constraint functions.
- Purpose of P: Impose penalties for constraint violations.

Penalty Functions II

UNIVERSITY OF ICELAND

Motivation and Approach

- Utilize familiar unconstrained optimization techniques on constrained problems.
- **Penalty approach**: Increase cost for infeasible solutions, driving towards feasibility and optimum.
- **Barrier approach**: Prevent feasible solutions from crossing into infeasibility.

Implementation Details

- Treat the objective function as unconstrained during minimization.
- Adjust the penalty to strike a balance between objective and constraints.
- Employ sequential unconstrained minimization techniques.

Penalty Functions

- Target: Minimize $f(\boldsymbol{x})$ subject to constraints $g(\boldsymbol{x}) \leq 0$ and $h(\boldsymbol{x}) = 0$.
- Reformulated objective:

$$\min F(\mathbf{x}, r_p) = f(\mathbf{x}) + r_p P(\mathbf{x})$$

$$P(\mathbf{x}) = \sum_{j=1}^{n} (\max[0, g_j(\mathbf{x})])^2 + \sum_{k=1}^{m} [h_k(\mathbf{x})]^2$$

where

$f(\mathbf{x})$: Original objective function

$P(\mathbf{x})$: Penalty function

r_p : Penalty weight

p : Iteration in the unconstrained optimization

Penalty Functions IV

UNIVERSITY OF ICELAND

- If all constraints are satisfied, then $P(\mathbf{x}) = 0$.
- The penalty parameter r_p typically starts small and increases with optimization iterations.
- A small r_p: Easier minimization of $F(\mathbf{x}, r_p)$ but may violate constraints significantly.
- A large r_p: Better adherence to constraints but function may become numerically sensitive.

Simple Analogy

- Imagine trying to park a car in a tight spot without hitting other vehicles. The penalty function is the alert system in the car.
- At first, the system is forgiving. It lets you approach other cars closely (representing a small r_p).
- As you continue parking, the system becomes more sensitive, beeping faster and louder (indicating an increase in r_p).
- The increasing sensitivity ensures you park without colliding with other cars.