Lecture Topic: Optimisation beyond 1D
Optimisation beyond 1D

Beyond optimisation in 1D, we will study two directions.

First, the equivalent in $n$th dimension, $x^* \in \mathbb{R}^n$ such that $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}^n$.

Second, constrained optimisation, i.e. $x^* \in \mathbb{R}^n$ such that $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}^n$ where $g_i(x) \leq 0, i = 1 \ldots m$.

For arbitrary $f, g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, this is undecidable.

We hence focus on (in some sense) smooth $f, g_i$, where it is still NP-Hard to decide, whether a point is a local optimum.

Only for smooth and convex $f, g_i$ and under additional assumptions, can one reason about global optima.

The methods presented are used throughout all of modern machine learning and much of operations research.
Optimisation beyond 1D

Beyond optimisation in 1D, we will study two directions.

First, the equivalent in $n$th dimension, $x^* \in \mathbb{R}^n$ such that $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}^n$.

Second, constrained optimisation, i.e. $x^* \in \mathbb{R}^n$ such that $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}^n$ where $g_i(x) \leq 0$, $i = 1 \ldots m$.

For arbitrary $f, g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, this is undecidable.

We hence focus on (in some sense) smooth $f, g_i$, where it is still NP-Hard to decide, whether a point is a local optimum.

Only for smooth and convex $f, g_i$ and under additional assumptions, can one reason about global optima.

The methods presented are used throughout all of modern machine learning and much of operations research.
Optimisation beyond 1D

Beyond optimisation in 1D, we will study two directions.

First, the equivalent in $n$th dimension, $x^* \in \mathbb{R}^n$ such that $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}^n$.

Second, constrained optimisation, i.e. $x^* \in \mathbb{R}^n$ such that $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}^n$ where $g_i(x) \leq 0, i = 1 \ldots m$.

For arbitrary $f, g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, this is undecidable.

We hence focus on (in some sense) smooth $f, g_i$, where it is still NP-Hard to decide, whether a point is a local optimum.

Only for smooth and convex $f, g_i$ and under additional assumptions, can one reason about global optima.

The methods presented are used throughout all of modern machine learning and much of operations research.
Optimisation beyond 1D

Beyond optimisation in 1D, we will study two directions.

First, the equivalent in $n$th dimension, $x^* \in \mathbb{R}^n$ such that $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}^n$.

Second, constrained optimisation, i.e. $x^* \in \mathbb{R}^n$ such that $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}^n$ where $g_i(x) \leq 0$, $i = 1 \ldots m$.

For arbitrary $f, g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, this is undecidable.

We hence focus on (in some sense) smooth $f, g_i$, where it is still NP-Hard to decide, whether a point is a local optimum.

Only for smooth and convex $f, g_i$ and under additional assumptions, can one reason about global optima.

The methods presented are used throughout all of modern machine learning and much of operations research.
Optimisation beyond 1D

Beyond optimisation in 1D, we will study two directions.

First, the equivalent in \( n \)th dimension, \( x^* \in \mathbb{R}^n \) such that \( f(x^*) \leq f(x) \) for all \( x \in \mathbb{R}^n \).

Second, constrained optimisation, i.e. \( x^* \in \mathbb{R}^n \) such that \( f(x^*) \leq f(x) \) for all \( x \in \mathbb{R}^n \) where \( g_i(x) \leq 0, i = 1 \ldots m \).

For arbitrary \( f, g_i : \mathbb{R}^n \to \mathbb{R} \), this is undecidable.

We hence focus on (in some sense) smooth \( f, g_i \), where it is still NP-Hard to decide, whether a point is a local optimum.

Only for smooth and convex \( f, g_i \) and under additional assumptions, can one reason about global optima.

The methods presented are used throughout all of modern machine learning and much of operations research.
Optimisation beyond 1D

Beyond optimisation in 1D, we will study two directions.

First, the equivalent in $n$th dimension, $x^* \in \mathbb{R}^n$ such that $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}^n$.

Second, constrained optimisation, i.e. $x^* \in \mathbb{R}^n$ such that $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}^n$ where $g_i(x) \leq 0$, $i = 1 \ldots m$.

For arbitrary $f, g_i : \mathbb{R}^n \to \mathbb{R}$, this is undecidable.

We hence focus on (in some sense) smooth $f, g_i$, where it is still NP-Hard to decide, whether a point is a local optimum.

Only for smooth and convex $f, g_i$ and under additional assumptions, can one reason about global optima.

The methods presented are used throughout all of modern machine learning and much of operations research.
Beyond optimisation in 1D, we will study two directions.

First, the equivalent in $n$th dimension, $x^* \in \mathbb{R}^n$ such that $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}^n$.

Second, constrained optimisation, i.e. $x^* \in \mathbb{R}^n$ such that $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}^n$ where $g_i(x) \leq 0, i = 1 \ldots m$.

For arbitrary $f, g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, this is undecidable.

We hence focus on (in some sense) smooth $f, g_i$, where it is still NP-Hard to decide, whether a point is a local optimum.

Only for smooth and convex $f, g_i$ and under additional assumptions, can one reason about global optima.

The methods presented are used throughout all of modern machine learning and much of operations research.
Optimisation: Key Concepts

*Constrained minimisation*: $x^* \in \mathbb{R}^n$ such that $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}^n$ where $g_i(x) \leq 0$, $i = 1 \ldots m$.

*Jacobian* $\nabla f$: the $m \times n$ matrix of all first-order partial derivatives of a vector-valued function $g : \mathbb{R}^n \to \mathbb{R}^m$.


*Gradient methods*: consider $f(x + \Delta x) \approx f(x) + \nabla f(x)\Delta x$ and go in the “antigradient direction”

*Newton-type methods*: consider the quadratic approximation $f(x + \Delta x) \approx f(x) + \nabla f(x)\Delta x + \frac{1}{2} \Delta x^T H(x) \Delta x$ and multiply the “antigradient direction” with the inverse Hessian

A witness: Checking whether a point $x^* \in \mathbb{R}$ satisfies $f'(x^*) = 0$ is beyond 1D much easier than checking $x^*$ is a local (!) minimum.
Optimisation: Key Concepts

**Constrained minimisation:** \( x^* \in \mathbb{R}^n \) such that \( f(x^*) \leq f(x) \) for all \( x \in \mathbb{R}^n \) where \( g_i(x) \leq 0, \ i = 1 \ldots m \).

**Jacobian** \( \nabla f \): the \( m \times n \) matrix of all first-order partial derivatives of a vector-valued function \( g : \mathbb{R}^n \rightarrow \mathbb{R}^m \).

**Hessian** \( H \): a square matrix of second-order partial derivatives of a scalar-valued function \( f \), \( H(f)(x) = J(\nabla f)(x) \).

**Gradient methods:** consider \( f(x + \Delta x) \approx f(x) + \nabla f(x) \Delta x \) and go in the “antigradient direction”

**Newton-type methods:** consider the quadratic approximation \( f(x + \Delta x) \approx f(x) + \nabla f(x) \Delta x + \frac{1}{2} \Delta x^T H(x) \Delta x \) and multiply the “antigradient direction” with the inverse Hessian

**A witness:** Checking whether a point \( x^* \in \mathbb{R} \) satisfies \( f'(x^*) = 0 \) is beyond 1D much easier than checking \( x^* \) is a local (!) minimum.
Optimisation: Key Concepts

Constrained minimisation: $x^* \in \mathbb{R}^n$ such that $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}^n$ where $g_i(x) \leq 0, i = 1 \ldots m$.

Jacobian $\nabla f$: the $m \times n$ matrix of all first-order partial derivatives of a vector-valued function $g : \mathbb{R}^n \to \mathbb{R}^m$.


Gradient methods: consider $f(x + \Delta x) \approx f(x) + \nabla f(x)\Delta x$ and go in the “antigradient direction”

Newton-type methods: consider the quadratic approximation $f(x + \Delta x) \approx f(x) + \nabla f(x)\Delta x + \frac{1}{2}\Delta x^T H(x)\Delta x$ and multiply the “antigradient direction” with the inverse Hessian

A witness: Checking whether a point $x^* \in \mathbb{R}$ satisfies $f'(x^*) = 0$ is beyond 1D much easier than checking $x^*$ is a local (!) minimum.
Optimisation: Key Concepts

*Constrained minimisation:* \( x^* \in \mathbb{R}^n \) such that \( f(x^*) \leq f(x) \) for all \( x \in \mathbb{R}^n \) where \( g_i(x) \leq 0, \ i = 1 \ldots m \).

*Jacobian* \( \nabla f \): the \( m \times n \) matrix of all first-order partial derivatives of a vector-valued function \( g : \mathbb{R}^n \rightarrow \mathbb{R}^m \).

*Hessian* \( H \): a square matrix of second-order partial derivatives of a scalar-valued function \( f \), \( H(f)(x) = J(\nabla f)(x) \).

*Gradient methods:* consider \( f(x + \Delta x) \approx f(x) + \nabla f(x)\Delta x \) and go in the “antigradient direction”

*Newton-type methods:* consider the quadratic approximation \( f(x + \Delta x) \approx f(x) + \nabla f(x)\Delta x + \frac{1}{2}\Delta x^TH(x)\Delta x \) and multiply the “antigradient direction” with the inverse Hessian

A witness: Checking whether a point \( x^* \in \mathbb{R} \) satisfies \( f'(x^*) = 0 \) is beyond 1D much easier than checking \( x^* \) is a local (!) minimum.
Optimisation: Key Concepts

Constrained minimisation: $x^* \in \mathbb{R}^n$ such that $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}^n$ where $g_i(x) \leq 0$, $i = 1 \ldots m$.

*Jacobian* $\nabla f$: the $m \times n$ matrix of all first-order partial derivatives of a vector-valued function $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$.


Gradient methods: consider $f(x + \Delta x) \approx f(x) + \nabla f(x) \Delta x$ and go in the “antigradient direction”

Newton-type methods: consider the quadratic approximation $f(x + \Delta x) \approx f(x) + \nabla f(x) \Delta x + \frac{1}{2} \Delta x^T H(x) \Delta x$ and multiply the “antigradient direction” with the inverse Hessian

A witness: Checking whether a point $x^* \in \mathbb{R}$ satisfies $f'(x^*) = 0$ is beyond 1D much easier than checking $x^*$ is a local (!) minimum.
Optimisation: Key Concepts

**Constrained minimisation:** $x^* \in \mathbb{R}^n$ such that $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}^n$ where $g_i(x) \leq 0$, $i = 1 \ldots m$.

**Jacobian** $\nabla f$: the $m \times n$ matrix of all first-order partial derivatives of a vector-valued function $g : \mathbb{R}^n \to \mathbb{R}^m$.

**Hessian** $H$: a square matrix of second-order partial derivatives of a scalar-valued function $f$, $H(f)(x) = J(\nabla f)(x)$.

**Gradient methods:** consider $f(x + \Delta x) \approx f(x) + \nabla f(x) \Delta x$ and go in the “antigradient direction”

**Newton-type methods:** consider the quadratic approximation $f(x + \Delta x) \approx f(x) + \nabla f(x) \Delta x + \frac{1}{2} \Delta x^T H(x) \Delta x$ and multiply the “antigradient direction” with the inverse Hessian

**A witness:** Checking whether a point $x^* \in \mathbb{R}$ satisfies $f'(x^*) = 0$ is beyond 1D much easier than checking $x^*$ is a local (!) minimum.
Function Classes

- Function $f : \mathbb{R}^n \to \mathbb{R}$ is Lipschitz-continuous with constant $L$, $L$ finite, if and only if: $\|f(x) - f(y)\| \leq L\|x - y\|$ for any $x, y \in \mathbb{R}^n$.

- Any Lipschitz-continuous function can be approximated by an infinitely differentiable function within arbitrarily small accuracy.

- We denote by $C_L^{k,p}(Q)$ the class of functions defined on $Q \subseteq \mathbb{R}^n$, which are $k$ times continuously differentiable on $Q$ and whose $p$th derivative is Lipschitz-continuous on $Q$ with constant $L$.

- Function $f$ belongs to $C_L^{2,1}(\mathbb{R}^n)$ if and only if $\|f''(x)\| \leq L$ for all $x \in \mathbb{R}^n$. 

Gradient

Definition

If a scalar-valued function $f : \mathbb{R}^n \to \mathbb{R}$ has first-order partial derivatives with respect to each $x_i$, then the $n$-dimensional equivalent of the first derivative $f'(x)$ is the gradient vector

$$\nabla f(x) = \nabla f = \left( \frac{\partial f(x)}{\partial x_i} \right) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$
Directional Derivative

When the partial derivatives are not well-defined, we may consider:

Definition

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$. The directional derivative of $f$ at $x \in \mathbb{R}^n$ in the direction $v$ is

$$d_v f(x) = \frac{\partial f}{\partial v} := v \cdot \nabla f(x) \quad \text{(dot product)}$$

$$= \sum_{i=1}^{n} v_i \frac{\partial f}{\partial x_i}.$$

where $v = (v_1, \ldots, v_n)^t \in \mathbb{R}^n$. 
Definition

If the function $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ has first-order partial derivatives with respect to each $x_i$, then the $m \times n$ matrix:

$$
J = \frac{\partial f}{\partial x} = \begin{bmatrix}
\frac{\partial f}{\partial x_1} & \cdots & \frac{\partial f}{\partial x_n}
\end{bmatrix} = 
\begin{bmatrix}
\frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\
\vdots & \ddots & \vdots \\
\frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n}
\end{bmatrix}
$$

is the Jacobian.
**Hessian**

**Definition**

The $n$-dimensional equivalent of the second derivative $f''(x)$ is the Hessian matrix:

$$H_f(x) = \left( \frac{\partial^2 f(x)}{\partial x_i \partial x_j} \right) = \begin{pmatrix}
\frac{\partial^2 f(x)}{\partial x_1 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\
\frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_n \partial x_n}
\end{pmatrix}.$$

Note that $H_f(x^*)$ is a symmetric matrix since $\frac{\partial^2 f(x)}{\partial x_i \partial x_j} = \frac{\partial^2 f(x)}{\partial x_j \partial x_i}$ for all $i, j$. We omit the subscript where not needed.
Taylor Series

Definition

The Taylor series expansion of $f(x)$ about some $x_k \in \mathbb{R}^n$ is:

$$f(x) \approx f(x_k) + (\nabla f(x_k))^t (x - x_k) + \frac{1}{2}(x - x_k)^t H_f(x_k) (x - x_k) + \cdots,$$

where $f(x) \in \mathbb{R}$, $x$ and $x_k \in \mathbb{R}^n$, and $H_f(x_k) \in M_n\mathbb{R}$. 
Also we define

**Definition**

\[
\frac{\partial^2 f}{\partial v^2} := \sum_{i=1}^{n} v_i \frac{\partial f}{\partial v} \left( \frac{\partial f}{\partial x_i} \right)
\]

\[
= \sum_{i=1}^{n} v_i \left( \sum_{j=1}^{n} v_j \frac{\partial^2 f}{\partial x_i \partial x_j} \right)
\]

\[
= \sum_{i,j=1}^{n} v_i v_j \frac{\partial^2 f}{\partial x_i \partial x_j}
\]

\[
= \mathbf{v}^T H_f(x) \mathbf{v}.
\]
Minima

**Theorem**

Let $U$ be an open subset of $\mathbb{R}^n$, $f : U \to \mathbb{R}$ be a twice continuously differentiable function on $U$, and let $x^*$ be a critical point of $f$, i.e., $\nabla f(x^*) = 0$. Then

(a) $x^*$ is a local maximum of $f$ if $\frac{\partial^2 f}{\partial v^2} < 0$ for all nonzero $v \in \mathbb{R}^n$;

(b) $x^*$ is a local minimum of $f$ if $\frac{\partial^2 f}{\partial v^2} > 0$ for all nonzero $v \in \mathbb{R}^n$;

(c) $x^*$ is a saddle point of $f$ if there exist $v, w \in \mathbb{R}^n$ such that

$$\frac{\partial^2 f}{\partial v^2} < 0 < \frac{\partial^2 f}{\partial w^2}.$$

It is clear that this involves examining the sign of $v^t H_f(x)v$ for various $v$. It can be shown that this theorem leads to a practical test as follows.
The Approaches

- Derivative-free methods
- Gradient methods
- Quasi-Newton methods
- Newton-type methods
- Interior-point methods
Derivative-Free Methods

For functions for $x^* \in \mathbb{R}^n$, the convergence of derivative-free methods is provably slow.

**Theorem (Nesterov)**

For $L$-Lipschitz function $f$, $\epsilon \leq \frac{1}{2}L$, and $f$ provided as an oracle that allows $f(x)$ to be evaluated for $x$, derivative-free methods require

$$\left(\left\lfloor \frac{L}{2\epsilon} \right\rfloor \right)^n$$

calls to the oracle to reach $\epsilon$ accuracy.
Derivative-Free Methods

To put the lower bound into perspective, consider a single computer, which can sustain the performance of $10^{11}$ operations per second ("100 gigaFLOPS") and a function, which can be evaluated in $n$ operations:

For $L = 2, n = 10$, 10% accuracy, you need $10^{11}$ operations, or 1 second.

For $L = 2, n = 10$, 1% accuracy, you need $10^{21}$ operations, or 325 years.

For $L = 2, n = 10$, 0.1% accuracy, you need $10^{31}$ operations, or $10^{12}$ years.

For $L = 2, n = 100$, 1% accuracy, you need $10^{201}$ operations, or $10^{182}$ years.
To put the lower bound into perspective, consider a single computer, which can sustain the performance of $10^{11}$ operations per second (“100 gigaFLOPS”) and a function, which can be evaluated in $n$ operations:

For $L = 2, n = 10$, 10% accuracy, you need $10^{11}$ operations, or 1 second.

For $L = 2, n = 10$, 1% accuracy, you need $10^{21}$ operations, or 325 years.

For $L = 2, n = 10$, 0.1% accuracy, you need $10^{31}$ operations, or $10^{12}$ years.

For $L = 2, n = 100$, 1% accuracy, you need $10^{201}$ operations, or $10^{182}$ years.
Derivative-Free Methods

To put the lower bound into perspective, consider a single computer, which can sustain the performance of $10^{11}$ operations per second ("100 gigaFLOPS") and a function, which can be evaluated in $n$ operations:

For $L = 2$, $n = 10$, 10% accuracy, you need $10^{11}$ operations, or 1 second.
For $L = 2$, $n = 10$, 1% accuracy, you need $10^{21}$ operations, or 325 years.
For $L = 2$, $n = 10$, 0.1% accuracy, you need $10^{31}$ operations, or $10^{12}$ years.
For $L = 2$, $n = 100$, 1% accuracy, you need $10^{201}$ operations, or $10^{182}$ years.
Derivative-Free Methods

To put the lower bound into perspective, consider a single computer, which can sustain the performance of $10^{11}$ operations per second ("100 gigaFLOPS") and a function, which can be evaluated in $n$ operations:

For $L = 2$, $n = 10$, 10% accuracy, you need $10^{11}$ operations, or 1 second.
For $L = 2$, $n = 10$, 1% accuracy, you need $10^{21}$ operations, or 325 years.
For $L = 2$, $n = 10$, 0.1% accuracy, you need $10^{31}$ operations, or $10^{12}$ years.
For $L = 2$, $n = 100$, 1% accuracy, you need $10^{201}$ operations, or $10^{182}$ years.
Derivative-Free Methods

To put the lower bound into perspective, consider a single computer, which can sustain the performance of $10^{11}$ operations per second (“100 gigaFLOPS”) and a function, which can be evaluated in $n$ operations:

For $L = 2$, $n = 10$, 10% accuracy, you need $10^{11}$ operations, or 1 second.
For $L = 2$, $n = 10$, 1% accuracy, you need $10^{21}$ operations, or 325 years.
For $L = 2$, $n = 10$, 0.1% accuracy, you need $10^{31}$ operations, or $10^{12}$ years.
For $L = 2$, $n = 100$, 1% accuracy, you need $10^{201}$ operations, or $10^{182}$ years.
Gradient Methods

Let us consider a local, unconstrained minimum $x^*$ of a multi-variate function, i.e., $f(x^*) \leq f(x) \forall x$ with $||x - x^*|| \leq \epsilon$.

From the definition, at a local minimum $x^*$, we expect the variation in $f$ due to a small variation $\Delta x$ in $x$ to be non-negative: $\nabla f(x^*)\Delta x = \sum_{i=1}^{n} \frac{\partial f(x^*)}{\partial x_i} \Delta x \geq 0$.

By considering $\Delta x$ coordinate wise, we get: $\nabla f(x^*) = 0$. 
Gradient Methods

Let us consider a local, unconstrained minimum $x^*$ of a multi-variate function, i.e., $f(x^*) \leq f(x) \forall x$ with $||x - x^*|| \leq \epsilon$.

From the definition, at a local minimum $x^*$, we expect the variation in $f$ due to a small variation $\Delta x$ in $x$ to be non-negative: $\nabla f(x^*) \Delta x = \sum_{i=1}^{n} \frac{\partial f(x^*)}{\partial x_i} \Delta x \geq 0$.

By considering $\Delta x$ coordinate wise, we get: $\nabla f(x^*) = 0$. 
Gradient Methods

Let us consider a local, unconstrained minimum $x^*$ of a multi-variate function, i.e., $f(x^*) \leq f(x) \forall x$ with $||x - x^*|| \leq \epsilon$.

From the definition, at a local minimum $x^*$, we expect the variation in $f$ due to a small variation $\Delta x$ in $x$ to be non-negative: $\nabla f(x^*) \Delta x = \sum_{i=1}^{n} \frac{\partial f(x^*)}{\partial x_i} \Delta x \geq 0$.

By considering $\Delta x$ coordinate wise, we get: $\nabla f(x^*) = 0$. 
Gradient Methods

In gradient methods, you consider $x^{k+1} = x^k - h^k \nabla f(x^k)$, where $h^k$ is one of:

- **Constant step** $h^k = h$ or $h^k = h/\sqrt{k+1}$
- **Full relaxation** $h^k = \arg \min_{h \geq 0} f(x^k - h\nabla f(x^k))$
- **Armijo line search**: find $x^{k+1}$ such that the ratio $\frac{\nabla f(x^k)(x^k - x^{k+1})}{f(x^k) - f(x^{k+1})}$ is within some interval
Gradient Methods

For all of the choices above with $f \in C_L^{1,1}(R^n)$, one has

$$f(x^k) - f(x^{k+1}) \geq \frac{\omega}{L} \|\nabla f(x_k)\|^2.$$  

We hence want to bound the norm of the gradient. It turns out:

$$\min_{i=0}^k \|\nabla f(x_i)\| \leq \frac{1}{\sqrt{k+1}} \left[ \frac{1}{\omega L} (f(x_0) - f^*) \right]^{1/2}$$

This means that the norm of the gradient is less than $\epsilon$, if the number of iterations is greater than $\frac{L}{\omega \epsilon^2} (f(x_0) - f^*) - 1$. 

Jakub Mareček and Seán McGarraghy (UCD) Numerical Analysis and Software October 23, 2015 17 / 1
Gradient Methods

Theorem (Nesterov)

For \( f \in C^2_M(\mathbb{R}^n) \), \( ll_n \succeq H(f^*) \succeq Ll_n \), a certain gradient method starting from \( x^0, r^0 = \|x^0 - x^*\| \leq \frac{2l}{M} := \bar{r} \) converges as follows:

\[
\|x^k - x^*\| \leq \frac{\bar{r} r^0}{\bar{r} - \rho^0} \left( 1 - \frac{2l}{L + 3l} \right)^k
\]

This is called the (local) linear (rate of) convergence.
Newton-Type Methods

In finding a solution to a system of non-linear equations $F(x) = 0$, $x \in \mathbb{R}^n$, $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$, displacement $\Delta x$ as a solution to $F(x) + \nabla F(x)\Delta x = 0$, which is known as the Newton system.

Assuming $[\nabla F]^{-1}$ exists, we can use:

$$x^{k+1} = x^k - [\nabla F(x^k)]^{-1} F(x^k).$$

When we move from finding zeros of $F(x)$ to minimising $f(x)$, $x \in \mathbb{R}^n$, $f : \mathbb{R}^n \rightarrow \mathbb{R}$ by finding zeros of $\nabla f(x) = 0$, we obtain:

$$x^{k+1} = x^k - [\nabla^2 f(x^k)]^{-1} \nabla f(x^k).$$
Newton-Type Methods

In finding a solution to a system of non-linear equations $F(x) = 0$, $x \in \mathbb{R}^n$, $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$, displacement $\Delta x$ as a solution to $F(x) + \nabla F(x) \Delta x = 0$, which is known as the Newton system.

Assuming $[\nabla F]^{-1}$ exists, we can use:

$$x^{k+1} = x^k - [\nabla F(x^k)]^{-1} F(x^k).$$

When we move from finding zeros of $F(x)$ to minimising $f(x)$, $x \in \mathbb{R}^n$, $f : \mathbb{R}^n \rightarrow \mathbb{R}$ by finding zeros of $\nabla f(x) = 0$, we obtain:

$$x^{k+1} = x^k - [\nabla^2 f(x^k)]^{-1} \nabla f(x^k).$$
Newton-Type Methods

In finding a solution to a system of non-linear equations $F(x) = 0$, $x \in \mathbb{R}^n$, $F : \mathbb{R}^n \to \mathbb{R}^n$, displacement $\Delta x$ as a solution to $F(x) + \nabla F(x) \Delta x = 0$, which is known as the Newton system.

Assuming $[\nabla F]^{-1}$ exists, we can use:

$$x^{k+1} = x^k - [\nabla F(x^k)]^{-1} F(x^k).$$

When we move from finding zeros of $F(x)$ to minimising $f(x), x \in \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}$ by finding zeros of $\nabla f(x) = 0$, we obtain:

$$x^{k+1} = x^k - [\nabla^2 f(x^k)]^{-1} \nabla f(x^k).$$
Newton-Type Methods

In finding a solution to a system of non-linear equations \( F(x) = 0 \), \( x \in \mathbb{R}^n, F : \mathbb{R}^n \to \mathbb{R}^n \), displacement \( \Delta x \) as a solution to \( F(x) + \nabla F(x) \Delta x = 0 \), which is known as the Newton system.

Assuming \( [\nabla F]^{-1} \) exists, we can use:

\[
x^{k+1} = x^k - [\nabla F(x^k)]^{-1} F(x^k).
\]

When we move from finding zeros of \( F(x) \) to minimising \( f(x), x \in \mathbb{R}^n, f : \mathbb{R}^n \to \mathbb{R} \) by finding zeros of \( \nabla f(x) = 0 \), we obtain:

\[
x^{k+1} = x^k - [\nabla^2 f(x^k)]^{-1} \nabla f(x^k).
\]
Newton-Type Methods

In finding a solution to a system of non-linear equations $F(x) = 0$, $x \in \mathbb{R}^n$, $F : \mathbb{R}^n \to \mathbb{R}^n$, displacement $\Delta x$ as a solution to $F(x) + \nabla F(x) \Delta x = 0$, which is known as the Newton system.

Assuming $[\nabla F]^{-1}$ exists, we can use:

$$x^{k+1} = x^k - [\nabla F(x^k)]^{-1} F(x^k).$$

When we move from finding zeros of $F(x)$ to minimising $f(x)$, $x \in \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}$ by finding zeros of $\nabla f(x) = 0$, we obtain:

$$x^{k+1} = x^k - [\nabla^2 f(x^k)]^{-1} \nabla f(x^k).$$
Newton-Type Methods

In finding a solution to a system of non-linear equations $F(x) = 0$, $x \in \mathbb{R}^n$, $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$,
displacement $\Delta x$ as a solution to $F(x) + \nabla F(x) \Delta x = 0$, which is known as the Newton system.

Assuming $[\nabla F]^{-1}$ exists, we can use:

$$x^{k+1} = x^k - [\nabla F(x^k)]^{-1} F(x^k).$$

When we move from finding zeros of $F(x)$ to minimising $f(x), x \in \mathbb{R}^n$, $f : \mathbb{R}^n \rightarrow \mathbb{R}$ by finding zeros of $\nabla f(x) = 0$, we obtain:

$$x^{k+1} = x^k - [\nabla^2 f(x^k)]^{-1} \nabla f(x^k).$$
Newton-Type Methods

Alternatively, let us consider a quadratic approximation of $f$ at point $x^k$, i.e.,

$$f(x^k) + \langle \nabla f(x^k), x - x^k \rangle + \frac{1}{2} \langle H(x^k)(x - x^k), x - x^k \rangle.$$

Assuming that $H(x^k) \succeq 0$, one should like to choose $x^{k+1}$ by minimising the approximation, i.e. solving

$$\nabla f(x^k) + H(x^k)(x^{k+1} - x^k) = 0 \text{ for } x^{k+1}:$$

$$x^{k+1} = x^k - [\nabla^2 f(x^k)]^{-1} \nabla f(x^k).$$
Newton-Type Methods

Alternatively, let us consider a quadratic approximation of $f$ at point $x^k$, i.e.,

$$f(x^k) + \langle \nabla f(x^k), x - x^k \rangle + \frac{1}{2} \langle H(x^k)(x - x^k), x - x^k \rangle.$$

Assuming that $H(x^k) \succeq 0$, one should like to choose $x^{k+1}$ by minimising the approximation, i.e. solving

$$\nabla f(x^k) + H(x^k)(x^{k+1} - x^k) = 0 \text{ for } x^{k+1}:
$$

$$x^{k+1} = x^k - [\nabla^2 f(x^k)]^{-1} \nabla f(x^k).$$
Newton-Type Methods

Alternatively, let us consider a quadratic approximation of $f$ at point $x^k$, i.e.,

$$f(x^k) + \langle \nabla f(x^k), x - x^k \rangle + \frac{1}{2} \langle H(x^k)(x - x^k), x - x^k \rangle.$$ 

Assuming that $H(x^k) \succeq 0$, one should like to choose $x^{k+1}$ by minimising the approximation, i.e. solving

$$\nabla f(x^k) + H(x^k)(x^{k+1} - x^k) = 0 \text{ for } x^{k+1}:$$

$$x^{k+1} = x^k - [\nabla^2 f(x^k)]^{-1} \nabla f(x^k).$$
Newton-Type Methods

Alternatively, let us consider a quadratic approximation of \( f \) at point \( x^k \), i.e.,

\[
f(x^k) + \langle \nabla f(x^k), x - x^k \rangle + \frac{1}{2} \langle H(x^k)(x - x^k), x - x^k \rangle.
\]

Assuming that \( H(x^k) \succeq 0 \), one should like to choose \( x^{k+1} \) by minimising the approximation, i.e. solving

\[
\nabla f(x^k) + H(x^k)(x^{k+1} - x^k) = 0 \quad \text{for} \quad x^{k+1}:
\]

\[
x^{k+1} = x^k - [\nabla^2 f(x^k)]^{-1} \nabla f(x^k).
\]
Newton-Type Methods

Alternatively, let us consider a quadratic approximation of $f$ at point $x^k$, i.e.,

$$f(x^k) + \langle \nabla f(x^k), x - x^k \rangle + \frac{1}{2} \langle H(x^k)(x - x^k), x - x^k \rangle.$$

Assuming that $H(x^k) \succeq 0$, one should like to choose $x^{k+1}$ by minimising the approximation, i.e. solving

$$\nabla f(x^k) + H(x^k)(x^{k+1} - x^k) = 0 \text{ for } x^{k+1}:$$

$$x^{k+1} = x^k - [\nabla^2 f(x^k)]^{-1} \nabla f(x^k).$$
Theorem (Nesterov)

For $f \in C^2_m(\mathbb{R}^n)$, where there exists a local minimum with positive definite Hessian $H(f^*) \preceq ll_n$, $x_0$ close enough to $x^*$, i.e. $\|x_0 - x^*\| < \frac{2l}{3M}$, Newton method starting from $x^0$ converges as follows:

$$\|x^{k+1} - x^*\| \leq \frac{M\|x^k - x^*\|^2}{2l - 2M\|x^k - x^*\|}$$

This is called the (local) quadratic (rate of) convergence.
Newton-Type Methods

Newton method is only locally convergent, but the region of convergence is similar for gradient and Newton methods.

One can try to address the possible divergence by considering damping:
\[ x^{k+1} = x^k - h^k \left[ \nabla^2 f(x^k) \right]^{-1} \nabla f(x^k), \]
where \( h^k \geq 0 \) is a step-size, which usually goes to 1 as \( k \) goes to infinity, or other “regularisations”.

One can try to make a single iteration cheaper by either exploiting sparsity of the Hessian or by considering some approximation of its inverse.
Newton-Type Methods

Newton method is only locally convergent, but the region of convergence is similar for gradient and Newton methods.

One can try to address the possible divergence by considering damping:
\[ x^{k+1} = x^k - h^k \left[ \nabla^2 f(x^k) \right]^{-1} \nabla f(x^k), \]
where \( h^k \geq 0 \) is a step-size, which usually goes to 1 as \( k \) goes to infinity, or other “regularisations”.

One can try to make a single iteration cheaper by either exploiting sparsity of the Hessian or by considering some approximation of its inverse.
Newton-Type Methods

Newton method is only locally convergent, but the region of convergence is similar for gradient and Newton methods.

One can try to address the possible divergence by considering damping:
\[ x^{k+1} = x^k - h^k \left[ \nabla^2 f(x^k) \right]^{-1} \nabla f(x^k), \]
where \( h^k \geq 0 \) is a step-size, which usually goes to 1 as \( k \) goes to infinity, or other “regularisations”.

One can try to make a single iteration cheaper by either exploiting sparsity of the Hessian or by considering some approximation of its inverse.
Quasi-Newton Methods

Quasi-Newton methods build up a sequence of approximations $H^k$ of the inverse of the Hessian and use it in computing the step. Starting with $H^0 = I_n$ and some $x^0$, in each iteration: $x^{k+1} = x^k + h^k H^k \nabla f(x^k)$ for some step-length $h^k$ and $H^{k+1} = H^k + \Delta H^k$ where:

- In rank-one methods, $\Delta H^k = \frac{(\delta^k - H^k \gamma^k)(\delta^k - H^k \gamma^k)^T}{\langle \delta^k - H^k \gamma^k, \gamma^k \rangle}$
- In Broyden-Fletcher-Goldfarb-Shanno (BFGS):
  $\Delta H^k = \frac{H^k \gamma^k (\delta^k)^T + \delta^k (\gamma^k)^T H^k}{\langle H^k \gamma^k, \gamma^k \rangle} - \beta^k \frac{H^k \gamma^k (\gamma^k)^T H^k}{\langle H^k \gamma^k, \gamma^k \rangle}$

where $\delta^k = x^{k+1} - x^k$, $\gamma^k = \nabla f(x^{k+1}) - \nabla f(x^k)$, and $\beta^k = 1 + \frac{\langle \gamma^k, \delta^k \rangle}{\langle H^k \gamma^k, \gamma^k \rangle}$. These methods are very successful in practice, although their rates of convergence are very hard to bound.
Consider a constrained minimisation problem
\[
\min f(x) \text{ subject to } g(x) \leq 0, \ x \in \mathbb{R}^n \text{ where } f(x) : \mathbb{R}^n \rightarrow \mathbb{R}, \ g(x) : \mathbb{R}^n \rightarrow \mathbb{R}^m \text{ and the global optimum at } x^* \text{ is } f^*.
\]

Let us consider \( g \) as \( m \) inequalities \( g_i(x) \leq 0, \ i = 1 \ldots m \) and let us introduce Lagrange multipliers (also known as dual variables) \( y = (y_1, y_2, \ldots, y_m)^T, \ y_i \geq 0, \) one scalar for each inequality \( g_i \).

The Lagrangian of the constrained minimisation problem is:
\[
L(x, y) = f(x) + y^T g(x).
\]

One can extend this to an additional constraint \( x \in X \subseteq \mathbb{R}^n \).
Consider a constrained minimisation problem
\[
\min f(x) \text{ subject to } g(x) \leq 0, \ x \in \mathbb{R}^n \text{ where } f(x) : \mathbb{R}^n \to \mathbb{R}, g(x) : \mathbb{R}^n \to \mathbb{R}^m \text{ and the global optimum at } x^* \text{ is } f^*.
\]

Let us consider \( g \) as \( m \) inequalities \( g_i(x) \leq 0, i = 1 \ldots m \) and let us introduce Lagrange multipliers (also known as dual variables) \( y = (y_1, y_2, \ldots, y_m)^T \), \( y_i \geq 0 \), one scalar for each inequality \( g_i \).

The Lagrangian of the constrained minimisation problem is:
\[
L(x, y) = f(x) + y^T g(x).
\]
One can extend this to an additional constraint \( x \in X \subseteq \mathbb{R}^n \).
Consider a constrained minimisation problem
\[
\min f(x) \text{ subject to } g(x) \leq 0, \ x \in \mathbb{R}^n \text{ where } f(x) : \mathbb{R}^n \to \mathbb{R}, \ g(x) : \mathbb{R}^n \to \mathbb{R}^m \text{ and the global optimum at } x^* \text{ is } f^*.
\]

Let us consider \( g \) as \( m \) inequalities \( g_i(x) \leq 0, \ i = 1 \ldots m \) and let us introduce Lagrange multipliers (also known as dual variables) \( y = (y_1, y_2, \ldots, y_m)^T, \ y_i \geq 0, \) one scalar for each inequality \( g_i \).

The Lagrangian of the constrained minimisation problem is:
\[
L(x, y) = f(x) + y^T g(x).
\]

One can extend this to an additional constraint \( x \in X \subseteq \mathbb{R}^n \).
Lagrangian

Consider a constrained minimisation problem
\[
\min f(x) \text{ subject to } g(x) \leq 0, \ x \in \mathbb{R}^n \text{ where } f(x) : \mathbb{R}^n \to \mathbb{R}, \ g(x) : \mathbb{R}^n \to \mathbb{R}^m \text{ and the global optimum at } x^* \text{ is } f^*.
\]

Let us consider \( g \) as \( m \) inequalities \( g_i(x) \leq 0, \ i = 1 \ldots m \) and let us introduce Lagrange multipliers (also known as dual variables) \( y = (y_1, y_2, \ldots, y_m)^T, \ y_i \geq 0 \), one scalar for each inequality \( g_i \).

The Lagrangian of the constrained minimisation problem is:
\[
L(x, y) = f(x) + y^T g(x).
\]

One can extend this to an additional constraint \( x \in X \subseteq \mathbb{R}^n \).
The “Lagrangian primal” is $L_P(x) = \max_{y \geq 0} L(x, y)$, with $L_P(x) := \infty$ if any inequality is violated.

The “Lagrangian dual” is $L_D(y) = \min_{x \in X} L(x, y)$.

Its value clearly depends on the choice of $y$. For any $y \geq 0$, however,

$$f^* \geq L_D(y), \text{ i.e.}$$

$$f^* \geq \max_{y \geq 0} \min_{x \in X} L(x, y), \text{ and}$$

$$f^* = \min_{x \in X} \max_{y \geq 0} L(x, y).$$

$$\min_{x \in X} \max_{y \geq 0} L(x, y) \geq \max_{y \geq 0} \min_{x \in X} L(x, y) \quad \text{ (“weak duality”)}.$$ 

Any primal feasible solution provides an upper bound for the dual problem, and any dual feasible solution provides a lower bound for the primal problem.
Lagrangian

The “Lagrangian primal” is \( L_P(x) = \max_{y \geq 0} L(x, y) \), with \( L_P(x) := \infty \) if any inequality is violated.

The “Lagrangian dual” is \( L_D(y) = \min_{x \in X} L(x, y) \).

Its value clearly depends on the choice of \( y \). For any \( y \geq 0 \), however,

\[ f^* \geq L_D(y) \], i.e.

\[ f^* \geq \max_{y \geq 0} \min_{x \in X} L(x, y) \], and

\[ f^* = \min_{x \in X} \max_{y \geq 0} L(x, y) \].

\[ \min_{x \in X} \max_{y \geq 0} L(x, y) \geq \max_{y \geq 0} \min_{x \in X} L(x, y) \] ("weak duality").

Any primal feasible solution provides an upper bound for the dual problem, and any dual feasible solution provides a lower bound for the primal problem.
Lagrangian

The “Lagrangian primal” is $L_P(x) = \max_{y \geq 0} L(x, y)$, with $L_P(x) := \infty$ if any inequality is violated.

The “Lagrangian dual” is $L_D(y) = \min_{x \in X} L(x, y)$.

Its value clearly depends on the choice of $y$. For any $y \geq 0$, however,

$$f^* \geq L_D(y), \text{ i.e. }$$

$$f^* \geq \max_{y \geq 0} \min_{x \in X} L(x, y), \text{ and }$$

$$f^* = \min_{x \in X} \max_{y \geq 0} L(x, y).$$

$$\min_{x \in X} \max_{y \geq 0} L(x, y) \geq \max_{y \geq 0} \min_{x \in X} L(x, y) \text{ ("weak duality")}.$$  

Any primal feasible solution provides an upper bound for the dual problem, and any dual feasible solution provides a lower bound for the primal problem.
The “Lagrangian primal” is $L_P(x) = \max_{y \geq 0} L(x, y)$, with $L_P(x) := \infty$ if any inequality is violated.

The “Lagrangian dual” is $L_D(y) = \min_{x \in X} L(x, y)$.

Its value clearly depends on the choice of $y$. For any $y \geq 0$, however,

$$f^* \geq L_D(y), \text{ i.e.}$$

$$f^* \geq \max_{y \geq 0} \min_{x \in X} L(x, y), \text{ and}$$

$$f^* = \min_{x \in X} \max_{y \geq 0} L(x, y).$$

$$\min_{x \in X} \max_{y \geq 0} L(x, y) \geq \max_{y \geq 0} \min_{x \in X} L(x, y) \text{ ("weak duality")}.$$

Any primal feasible solution provides an upper bound for the dual problem, and any dual feasible solution provides a lower bound for the primal problem.
Lagrangian

The “Lagrangian primal” is $L_P(x) = \max_{y \geq 0} L(x, y)$, with $L_P(x) := \infty$ if any inequality is violated.

The “Lagrangian dual” is $L_D(y) = \min_{x \in X} L(x, y)$.

Its value clearly depends on the choice of $y$. For any $y \geq 0$, however,

$$f^* \geq L_D(y), \text{ i.e.}$$

$$f^* \geq \max_{y \geq 0} \min_{x \in X} L(x, y), \text{ and}$$

$$f^* = \min_{x \in X} \max_{y \geq 0} L(x, y).$$

$$\min_{x \in X} \max_{y \geq 0} L(x, y) \geq \max_{y \geq 0} \min_{x \in X} L(x, y) \text{ ("weak duality").}$$

Any primal feasible solution provides an upper bound for the dual problem, and any dual feasible solution provides a lower bound for the primal problem.
Lagrangian

The “Lagrangian primal” is \( L_P(x) = \max_{y \geq 0} L(x, y) \), with \( L_P(x) := \infty \) if any inequality is violated.

The “Lagrangian dual” is \( L_D(y) = \min_{x \in X} L(x, y) \).

Its value clearly depends on the choice of \( y \). For any \( y \geq 0 \), however,

\[
\begin{align*}
 f^* & \geq L_D(y), \text{ i.e. } \\
 f^* & \geq \max_{y \geq 0} \min_{x \in X} L(x, y), \text{ and } \\
 f^* & = \min_{x \in X} \max_{y \geq 0} L(x, y).
\end{align*}
\]

\[
\min_{x \in X} \max_{y \geq 0} L(x, y) \geq \max_{y \geq 0} \min_{x \in X} L(x, y) \quad \text{ (“weak duality”).}
\]

Any primal feasible solution provides an upper bound for the dual problem, and any dual feasible solution provides a lower bound for the primal problem.
Lagrangian

The “Lagrangian primal” is \( L_P(x) = \max_{y \geq 0} L(x, y) \), with \( L_P(x) := \infty \) if any inequality is violated.

The “Lagrangian dual” is \( L_D(y) = \min_{x \in X} L(x, y) \).

Its value clearly depends on the choice of \( y \). For any \( y \geq 0 \), however,
\[
f^* \geq L_D(y), \text{ i.e.} \]
\[
f^* \geq \max_{y \geq 0} \min_{x \in X} L(x, y), \text{ and} \]
\[
f^* = \min_{x \in X} \max_{y \geq 0} L(x, y). \]
\[
\min_{x \in X} \max_{y \geq 0} L(x, y) \geq \max_{y \geq 0} \min_{x \in X} L(x, y) \text{ (”weak duality”)}.
\]

Any primal feasible solution provides an upper bound for the dual problem, and any dual feasible solution provides a lower bound for the primal problem.
Lagrangian

The “Lagrangian primal” is \( L_P(x) = \max_{y \geq 0} L(x, y) \), with \( L_P(x) := \infty \) if any inequality is violated.

The “Lagrangian dual” is \( L_D(y) = \min_{x \in X} L(x, y) \).

Its value clearly depends on the choice of \( y \). For any \( y \geq 0 \), however,

\[
  f^* \geq L_D(y), \text{ i.e.} \\
  f^* \geq \max_{y \geq 0} \min_{x \in X} L(x, y), \text{ and} \\
  f^* = \min_{x \in X} \max_{y \geq 0} L(x, y).
\]

\[
  \min_{x \in X} \max_{y \geq 0} L(x, y) \geq \max_{y \geq 0} \min_{x \in X} L(x, y) \text{ ("weak duality")}.
\]

Any primal feasible solution provides an upper bound for the dual problem, and any dual feasible solution provides a lower bound for the primal problem.
Strong Duality and KKT Conditions

Assuming differentiability of $L$, Karush-Kuhn-Tucker (KKT) conditions are composed of stationarity ($\nabla_x L(x, y) = 0$), primal feasibility ($g(x) \leq 0, x \in X$), dual feasibility ($y \geq 0$), and “complementarity slackness” ($y_ig_i(x) = 0$).

Under some “regularity” assumptions (also known as constraint qualifications), we are guaranteed that a point $x$ satisfying the KKT condition exists.

If $X \subseteq \mathbb{R}^n$ is convex, $f$ and $g$ are convex, optimum $f^*$ is finite, and the regularity assumptions hold, then we have

$$\min_{x \in X} \max_{y \geq 0} L(x, y) = \max_{y \geq 0} \min_{x \in X} L(x, y) \quad \text{ (“strong duality”) and KKT conditions guarantee global optimality.}$$

For example, Slater’s constraint qualifications is: $\exists x \in \text{int}(X)$ such that $g(x) < 0$.

If $f$ and $g$ are linear, no further constraint qualifications is needed and KKT conditions suffice.
Strong Duality and KKT Conditions

Assuming differentiability of $L$, Karush-Kuhn-Tucker (KKT) conditions are composed of stationarity ($\nabla_x L(x, y) = 0$), primal feasibility ($g(x) \leq 0, x \in X$), dual feasibility ($y \geq 0$), and “complementarity slackness” ($y_i g_i(x) = 0$).

Under some “regularity” assumptions (also known as constraint qualifications), we are guaranteed that a point $x$ satisfying the KKT condition exists.

If $X \subseteq \mathbb{R}^n$ is convex, $f$ and $g$ are convex, optimum $f^*$ is finite, and the regularity assumptions hold, then we have

$$\min_{x \in X} \max_{y \geq 0} L(x, y) = \max_{y \geq 0} \min_{x \in X} L(x, y)$$

(“strong duality”) and KKT conditions guarantee global optimality.

For example, Slater’s constraint qualifications is: $\exists x \in \text{int}(X)$ such that $g(x) < 0$.

If $f$ and $g$ are linear, no further constraint qualifications is needed and KKT conditions suffice.
Strong Duality and KKT Conditions

Assuming differentiability of $L$, Karush-Kuhn-Tucker (KKT) conditions are composed of stationarity ($\nabla_x L(x, y) = 0$), primal feasibility ($g(x) \leq 0, x \in X$), dual feasibility ($y \geq 0$), and “complementarity slackness” ($y_i g_i(x) = 0$).

Under some “regularity” assumptions (also known as constraint qualifications), we are guaranteed that a point $x$ satisfying the KKT condition exists.

If $X \subseteq \mathbb{R}^n$ is convex, $f$ and $g$ are convex, optimum $f^*$ is finite, and the regularity assumptions hold, then we have

$$\min_{x \in X} \max_{y \geq 0} L(x, y) = \max_{y \geq 0} \min_{x \in X} L(x, y)$$

(“strong duality”) and KKT conditions guarantee global optimality.

For example, Slater’s constraint qualifications is: $\exists x \in \text{int}(X)$ such that $g(x) < 0$.

If $f$ and $g$ are linear, no further constraint qualifications is needed and KKT conditions suffice.
Strong Duality and KKT Conditions

Assuming differentiability of \( L \), Karush-Kuhn-Tucker (KKT) conditions are composed of stationarity \( (\nabla_x L(x, y) = 0) \), primal feasibility \( (g(x) \leq 0, x \in X) \), dual feasibility \( (y \geq 0) \), and “complementarity slackness” \( (y_i g_i(x) = 0) \).

Under some “regularity” assumptions (also known as constraint qualifications), we are guaranteed that a point \( x \) satisfying the KKT condition exists.

If \( X \subseteq \mathbb{R}^n \) is convex, \( f \) and \( g \) are convex, optimum \( f^* \) is finite, and the regularity assumptions hold, then we have

\[
\min_{x \in X} \max_{y \geq 0} L(x, y) = \max_{y \geq 0} \min_{x \in X} L(x, y) \quad \text{ (“strong duality”)}
\]

and KKT conditions guarantee global optimality.

For example, Slater’s constraint qualifications is: \( \exists x \in \text{int}(X) \) such that \( g(x) < 0 \).

If \( f \) and \( g \) are linear, no further constraint qualifications is needed and KKT conditions suffice.
Strong Duality and KKT Conditions

Assuming differentiability of $L$, Karush-Kuhn-Tucker (KKT) conditions are composed of stationarity ($\nabla_x L(x, y) = 0$), primal feasibility ($g(x) \leq 0, x \in X$), dual feasibility ($y \geq 0$), and “complementarity slackness” ($y_i g_i(x) = 0$).

Under some “regularity” assumptions (also known as constraint qualifications), we are guaranteed that a point $x$ satisfying the KKT condition exists.

If $X \subseteq \mathbb{R}^n$ is convex, $f$ and $g$ are convex, optimum $f^*$ is finite, and the regularity assumptions hold, then we have

$$\min_{x \in X} \max_{y \geq 0} L(x, y) = \max_{y \geq 0} \min_{x \in X} L(x, y)$$

(“strong duality”) and KKT conditions guarantee global optimality.

For example, Slater’s constraint qualifications is: $\exists x \in \text{int}(X)$ such that $g(x) < 0$.

If $f$ and $g$ are linear, no further constraint qualifications is needed and KKT conditions suffice.
Strong Duality and KKT Conditions

Assuming differentiability of \( L \), Karush-Kuhn-Tucker (KKT) conditions are composed of stationarity \( (\nabla_x L(x, y) = 0) \), primal feasibility \( (g(x) \leq 0, x \in X) \), dual feasibility \( (y \geq 0) \), and “complementarity slackness” \( (y_i g_i(x) = 0) \).

Under some “regularity” assumptions (also known as constraint qualifications), we are guaranteed that a point \( x \) satisfying the KKT condition exists.

If \( X \subseteq \mathbb{R}^n \) is convex, \( f \) and \( g \) are convex, optimum \( f^* \) is finite, and the regularity assumptions hold, then we have

\[
\min_{x \in X} \max_{y \geq 0} L(x, y) = \max_{y \geq 0} \min_{x \in X} L(x, y) \quad (\text{“strong duality”}) \text{ and KKT conditions guarantee global optimality.}
\]

For example, Slater’s constraint qualifications is: \( \exists x \in \text{int}(X) \) such that \( g(x) < 0 \).

If \( f \) and \( g \) are linear, no further constraint qualifications is needed and KKT conditions suffice.
Penalties and Barriers

Notice that the Lagrangian, as defined above, is not Lipschitz-continuous and is not differentiable. Let us consider a closed set $G$ defined by $g$, and let us assume it has non-empty interior.

A penalty $\phi$ for $G$ is a continuous function, such that $\phi(x) = 0$ for any $x \in G$ and $\phi(x) \geq 0$ for any $x \notin G$. E.g. $\sum_{i=1}^{m} \max\{g_i(x), 0\}$ (non-smooth), $\sum_{i=1}^{m} (\max\{g_i(x), 0\})^2$ (smooth).

A barrier $\phi$ for $G$ is a continuous function, such that $\phi(x) \to \infty$ as $x$ approaches the boundary of $G$ and is bounded from below elsewhere. E.g. $\sum_{i=1}^{m} \frac{1}{(-g_i(x))^p}$, $p \geq 1$ (power), $-\sum_{i=1}^{m} \ln(-g(x))$ (logarithmic).

One can consider (variants of) the Lagrangian of a constrained problem, which involve a barrier for the inequalities.

Using such Lagrangians, one can develop interior-point methods.
Penalties and Barriers

Notice that the Lagrangian, as defined above, is not Lipschitz-continuous and is not differentiable. Let us consider a closed set $G$ defined by $g$, and let us assume it has non-empty interior.

A penalty $\phi$ for $G$ is a continuous function, such that $\phi(x) = 0$ for any $x \in G$ and $\phi(x) \geq 0$ for any $x \notin G$. E.g. $\sum_{i=1}^{m} \max\{g_i(x), 0\}$ (non-smooth), $\sum_{i=1}^{m} (\max\{g_i(x), 0\})^2$ (smooth).

A barrier $\phi$ for $G$ is a continuous function, such that $\phi(x) \to \infty$ as $x$ approaches the boundary of $G$ and is bounded from below elsewhere. E.g. $\sum_{i=1}^{m} \frac{1}{(-g_i(x))^p}$, $p \geq 1$ (power), $-\sum_{i=1}^{m} \ln(-g(x))$ (logarithmic).

One can consider (variants of) the Lagrangian of a constrained problem, which involve a barrier for the inequalities.

Using such Lagrangians, one can develop interior-point methods.
Penalties and Barriers

Notice that the Lagrangian, as defined above, is not Lipschitz-continuous and is not differentiable. Let us consider a closed set $G$ defined by $g$, and let us assume it has non-empty interior.

A penalty $\phi$ for $G$ is a continuous function, such that $\phi(x) = 0$ for any $x \in G$ and $\phi(x) \geq 0$ for any $x \notin G$. E.g. $\sum_{i=1}^{m} \max\{g_i(x), 0\}$ (non-smooth), $\sum_{i=1}^{m} (\max\{g_i(x), 0\})^2$ (smooth).

A barrier $\phi$ for $G$ is a continuous function, such that $\phi(x) \to \infty$ as $x$ approaches the boundary of $G$ and is bounded from below elsewhere. E.g. $\sum_{i=1}^{m} \frac{1}{(-g_i(x))^p}$, $p \geq 1$ (power), $-\sum_{i=1}^{m} \ln(-g(x))$ (logarithmic).

One can consider (variants of) the Lagrangian of a constrained problem, which involve a barrier for the inequalities.

Using such Lagrangians, one can develop interior-point methods.
Penalties and Barriers

Notice that the Lagrangian, as defined above, is not Lipschitz-continuous and is not differentiable. Let us consider a closed set $G$ defined by $g$, and let us assume it has non-empty interior.

A penalty $\phi$ for $G$ is a continuous function, such that $\phi(x) = 0$ for any $x \in G$ and $\phi(x) \geq 0$ for any $x \notin G$. E.g. $\sum_{i=1}^{m} \max\{g_i(x), 0\}$ (non-smooth), $\sum_{i=1}^{m} (\max\{g_i(x), 0\})^2$ (smooth).

A barrier $\phi$ for $G$ is a continuous function, such that $\phi(x) \to \infty$ as $x$ approaches the boundary of $G$ and is bounded from below elsewhere. E.g. $\sum_{i=1}^{m} \frac{1}{(-g_i(x))^p}$, $p \geq 1$ (power), $-\sum_{i=1}^{m} \ln(-g(x))$ (logarithmic).

One can consider (variants of) the Lagrangian of a constrained problem, which involve a barrier for the inequalities.

Using such Lagrangians, one can develop interior-point methods.
Penalties and Barriers

Notice that the Lagrangian, as defined above, is not Lipschitz-continuous and is not differentiable. Let us consider a closed set $G$ defined by $g$, and let us assume it has non-empty interior.

A penalty $\phi$ for $G$ is a continuous function, such that $\phi(x) = 0$ for any $x \in G$ and $\phi(x) \geq 0$ for any $x \notin G$. E.g. $\sum_{i=1}^{m} \max\{g_i(x), 0\}$ (non-smooth), $\sum_{i=1}^{m} (\max\{g_i(x), 0\})^2$ (smooth).

A barrier $\phi$ for $G$ is a continuous function, such that $\phi(x) \to \infty$ as $x$ approaches the boundary of $G$ and is bounded from below elsewhere. E.g. $\sum_{i=1}^{m} \frac{1}{(-g_i(x))^p}$, $p \geq 1$ (power), $-\sum_{i=1}^{m} \ln(-g(x))$ (logarithmic).

One can consider (variants of) the Lagrangian of a constrained problem, which involve a barrier for the inequalities.

Using such Lagrangians, one can develop interior-point methods.
Interior-Point Methods

Interior-point methods solve progressively less relaxed first-order optimality conditions of a problem, which is equivalent to a constrained optimisation problem and uses barriers.

Consider a constrained minimisation \( \min f(x) \) subject to \( g(x) \leq 0 \) where \( f(x) : \mathbb{R}^n \to \mathbb{R}, g(x) : \mathbb{R}^n \to \mathbb{R}^m \) are convex and twice differentiable.

A nonnegative slack variable \( z \in \mathbb{R}^m \) can be used to replace the inequality by equality \( g(x) + z = 0 \).

Negative \( z \) can be avoided by using a barrier \( \mu \sum_{i=1}^{m} \ln z_i \).

The (variant of) Lagrangian is: \( L(x, y, z; \mu) = f(x) + y^T (g(x) + z) \mu \sum_{i=1}^{m} \ln z_i \).
Interior-Point Methods

Interior-point methods solve progressively less relaxed first-order optimality conditions of a problem, which is equivalent to a constrained optimisation problem and uses barriers.

Consider a constrained minimisation \( \min f(x) \) subject to \( g(x) \leq 0 \) where \( f(x): \mathbb{R}^n \to \mathbb{R}, g(x): \mathbb{R}^n \to \mathbb{R}^m \) are convex and twice differentiable.

A nonnegative slack variable \( z \in \mathbb{R}^m \) can be used to replace the inequality by equality \( g(x) + z = 0 \).

Negative \( z \) can be avoided by using a barrier \( \mu \sum_{i=1}^{m} \ln z_i \).

The (variant of) Lagrangian is: \( L(x, y, z; \mu) = f(x) + y^T (g(x) + z) \mu \sum_{i=1}^{m} \ln z_i \).
Interior-Point Methods

Interior-point methods solve progressively less relaxed first-order optimality conditions of a problem, which is equivalent to a constrained optimisation problem and uses barriers.

Consider a constrained minimisation $\min f(x)$ subject to $g(x) \leq 0$ where $f(x) : \mathbb{R}^n \to \mathbb{R}$, $g(x) : \mathbb{R}^n \to \mathbb{R}^m$ are convex and twice differentiable.

A nonnegative slack variable $z \in \mathbb{R}^m$ can be used to replace the inequality by equality $g(x) + z = 0$.

Negative $z$ can be avoided by using a barrier $\mu \sum_{i=1}^m \ln z_i$.

The (variant of) Lagrangian is: $L(x, y, z; \mu) = f(x) + y^T (g(x) + z) \mu \sum_{i=1}^m \ln z_i$. 
Interior-Point Methods

Interior-point methods solve progressively less relaxed first-order optimality conditions of a problem, which is equivalent to a constrained optimisation problem and uses barriers.

Consider a constrained minimisation $\min f(x)$ subject to $g(x) \leq 0$ where $f(x) : \mathbb{R}^n \to \mathbb{R}, g(x) : \mathbb{R}^n \to \mathbb{R}^m$ are convex and twice differentiable.

A nonnegative slack variable $z \in \mathbb{R}^m$ can be used to replace the inequality by equality $g(x) + z = 0$.

Negative $z$ can be avoided by using a barrier $\mu \sum_{i=1}^m \ln z_i$.

The (variant of) Lagrangian is: $L(x, y, z; \mu) = f(x) + y^T (g(x) + z) \mu \sum_{i=1}^m \ln z_i$. 

Jakub Mareček and Seán McGarraghy (UCD) Numerical Analysis and Software October 23, 2015 28 / 1
Interior-Point Methods

Interior-point methods solve progressively less relaxed first-order optimality conditions of a problem, which is equivalent to a constrained optimisation problem and uses barriers.

Consider a constrained minimisation \( \min f(x) \) subject to \( g(x) \leq 0 \) where \( f(x) : \mathbb{R}^n \to \mathbb{R} \), \( g(x) : \mathbb{R}^n \to \mathbb{R}^m \) are convex and twice differentiable.

A nonnegative slack variable \( z \in \mathbb{R}^m \) can be used to replace the inequality by equality \( g(x) + z = 0 \).

Negative \( z \) can be avoided by using a barrier \( \mu \sum_{i=1}^m \ln z_i \).

The (variant of) Lagrangian is: \( L(x, y, z; \mu) = f(x) + y^T (g(x) + z) \mu \sum_{i=1}^m \ln z_i \).
Now we can differentiate:

\[ \nabla_x L(x, y, z; \mu) = \nabla f(x) + \nabla g(x)^T y \]  
\[ \nabla_y L(x, y, z; \mu) = g(x) + z \]  
\[ \nabla_z L(x, y, z; \mu) = y \mu Z^{-1} e, \]

where \( Z = \text{diag}(z_1, z_2, \ldots, z_m) \) and \( e = [1, 1, \ldots, 1]^T \).
Interior-Point Methods

The first-order optimality conditions obtained by setting the partial derivatives to zero are:

\[ \nabla f(x) + \nabla g(x)^T y = 0 \]  \hspace{1cm} (8.4)

\[ g(x) + z = 0 \]  \hspace{1cm} (8.5)

\[ YZe = \mu e \]  \hspace{1cm} (8.6)

\[ y, z \geq 0 \]  \hspace{1cm} (8.7)

where \( Y = \text{diag}(y_1, y_2, \ldots, y_m) \) and the parameter \( \mu \) is reduced to 0 in the large number of iterations.
Interior-Point Methods

This can be solved using the Newton method, where at each step, one solves a linear system:

\[
\begin{bmatrix}
-H(x, y) & B(x)^T \\
B(x) & XY^{-1}
\end{bmatrix}
\begin{bmatrix}
\Delta x \\
-\Delta y
\end{bmatrix}
= 
\begin{bmatrix}
\nabla f(x) + B(x)^T y \\
-g(x) - \mu Y^{-1} e
\end{bmatrix}
\]

where \( H(x, y) = \nabla^2 f(x) + \sum_{i=1}^{m} y_i \nabla^2 g_i(x) \in \mathbb{R}^{n \times n} \) and \( B(x) = \nabla g(x) \in \mathbb{R}^{m \times n} \). This is a saddle point system, which has often positive semidefinite \( A \). For convex \( f, g, H(x, y) \) is positive semidefinite and diagonal matrix \( ZY^{-1} \) is also positive definite. A variety of methods works very well.
Condition Numbers

Assume the instance \( d := (A; b; c) \) is given. One can formalise the following notion, due to Renegar:

\[
C(d) := \frac{||d||}{\inf\{||\Delta d|| : \text{instance } d + \Delta d \text{ is infeasible or unbounded } \}}.
\]

The system (8.4–8.7) will have the condition number \( C(d)/\mu \).
Condition Numbers

Assume the instance \( d := (A; b; c) \) is given. One can formalise the following notion, due to Renegar:

\[
C(d) := \frac{||d||}{\inf\{||\Delta d|| : \text{instance } d + \Delta d \text{ is infeasible or unbounded } \}}.
\]

The system (8.4–8.7) will have the condition number \( C(d)/\mu \).
Condition Numbers in Analyses

For a variety of methods, including:

- Interior-point methods
- Ellipsoid method
- Perceptron method
- Von Neumann method

assuming $A$ is invertible, one can show a bound on the number of iterations is logarithmic in $C(d)$. This highlights the need for preconditioners.
Condition Numbers in Analyses

For a variety of methods, including:

- Interior-point methods
- Ellipsoid method
- Perceptron method
- Von Neumann method

assuming $A$ is invertible, one can show a bound on the number of iterations is logarithmic in $C(d)$.

This highlights the need for preconditioners.
Conclusions

Constrained optimisation is the work-horse of operations research.

Interior-point methods have been used on problems in dimensions $10^9$. Still, there are many open problems, including Smale’s 9th problem: Is feasibility of a linear system of inequalities $Ax \geq b$ in $P$ over reals, i.e., solvable in polynomial time on the BSS machine?
Conclusions

Constrained optimisation is the work-horse of operations research. Interior-point methods have been used on problems in dimensions $10^9$. Still, there are many open problems, including Smale’s 9th problem: Is feasibility of a linear system of inequalities $Ax \geq b$ in $P$ over reals, i.e., solvable in polynomial time on the BSS machine?
Conclusions

Constrained optimisation is the work-horse of operations research. Interior-point methods have been used on problems in dimensions $10^9$. Still, there are many open problems, including Smale’s 9th problem:

Is feasibility of a linear system of inequalities $Ax \geq b$ in $P$ over reals, i.e., solvable in polynomial time on the BSS machine?
Conclusions

Constrained optimisation is the work-horse of operations research. Interior-point methods have been used on problems in dimensions $10^9$. Still, there are many open problems, including Smale’s 9th problem: Is feasibility of a linear system of inequalities $Ax \geq b$ in $P$ over reals, i.e., solvable in polynomial time on the BSS machine?
Conclusions

Constrained optimisation is the work-horse of operations research. Interior-point methods have been used on problems in dimensions $10^9$. Still, there are many open problems, including Smale’s 9th problem: Is feasibility of a linear system of inequalities $Ax \geq b$ in $P$ over reals, i.e., solvable in polynomial time on the BSS machine?