CHEMICAL ENGINEERING 4G03

Module 02

Mathematical Concepts of Optimization

Dr. Jake Nease

Chemical Engineering

McMaster University

Updated January 26, 2023

Outline of Module

This module consists of the following topics. It is important to note that **we are going to add to this module as we progress through the course**. We will introduce new concepts as they are required in each future module.

This section can be a little math-y, but please do not let it get to you. We are learning the mathematical concepts now so that we can exploit them later. Besides… Math can be fun sometimes‼

- **Part (I) - NOW**
  - Local and Global Optima
  - Direction Change Paradigm
  - Numerical Method: Improving Search
  - Convexity
- **Part (II) – In the NLP Section**
  - Gradient and Hessian Review
  - Conditions of Optimality
  - Application of Convexity to NLPs

Suggested Readings

Rardin (1st edition): Chapter 3

Rardin (2nd edition): Chapter 3

Types of Optima

There are two types of optima that you may encounter while solving an optimization problem: **local** and **global**. The characterizations of each of these optima are important for the understanding of an optimization solution, and we will discuss them here. We will use some nonlinear examples in this section because they make for good visual examples, but rest assured all the conclusions we come to in this section are applicable to linear systems as well.

Before we can define either type of optimum, we need to define the concept of a **neighbourhood**. Both types of optima are defined *around a neighbourhood*.

Neighbourhood

The **neighbourhood** $N_\delta(\boldsymbol{x}^o)$ of a point $\boldsymbol{x}^o$ consists of *all nearby points*. That is, all points within some small distance $\delta > 0$ of the point $\boldsymbol{x}^o$:

$$N_\delta(\boldsymbol{x}^o) \triangleq \{\boldsymbol{x} : \|\boldsymbol{x} - \boldsymbol{x}^o\| < \delta\}$$

The neighbourhood around $\boldsymbol{x}^o$ is defined as the set of all possible values of $\boldsymbol{x}$ such that the distance (magnitude in > 1D) between the location $\boldsymbol{x}$ and $\boldsymbol{x}^o$ is less than some small number δ.

Local and Global Optima

A local optimum is a point such that all points in the neighbourhood surrounding it result in a ***worse*** value for the objective function.

Local Optimum (Minimum)

A point $\boldsymbol{x}^\star$ is a **local minimum** for the function $\phi : \mathbb{R}^n \to \mathbb{R}$ (maps n dimensions to a single numerical objective) on the set $\mathcal{S}$ (recall that this set defines our feasible region) if it is **feasible** ($\boldsymbol{x}^\star \in \mathcal{S}$) and if a sufficiently small neighbourhood δ surrounding it contains **no points that are both feasible and lower in objective value**:

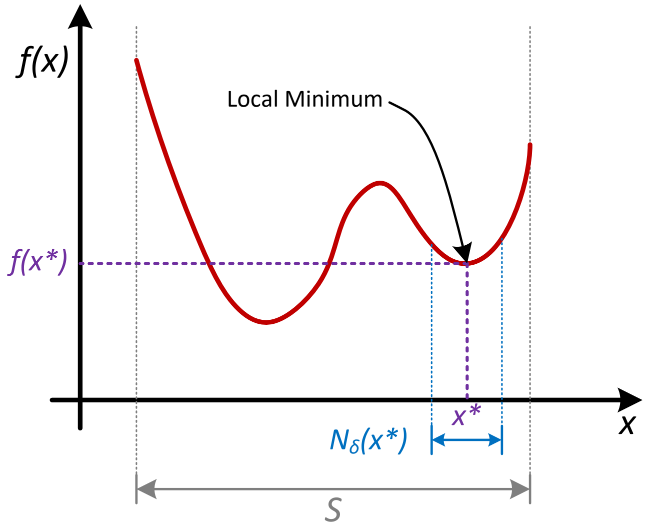
$$\exists\, \delta > 0\colon\ \phi(\boldsymbol{x}^\star) \le \phi(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in \mathcal{S} \cap N_\delta(\boldsymbol{x}^\star)$$

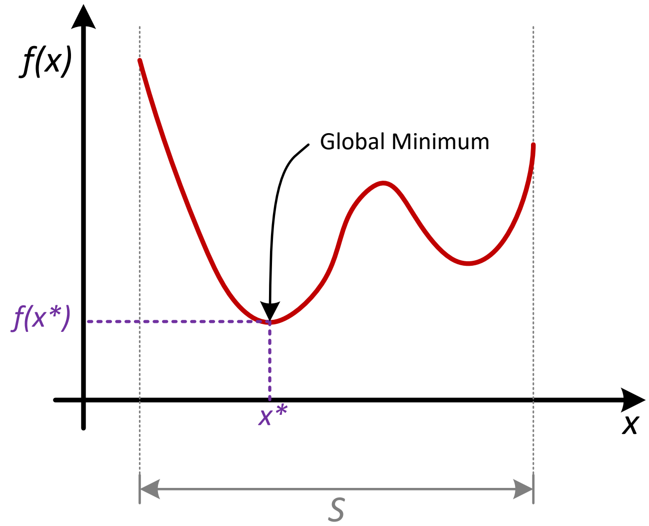
$$strict\colon \exists\, \delta > 0\colon\ \phi(\boldsymbol{x}^\star) < \phi(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in \mathcal{S} \cap N_\delta(\boldsymbol{x}^\star) \setminus \{\boldsymbol{x}^\star\}$$

Some **REMARKS on the notation** in the math above (I promise we won't be introducing too many more symbols and the like):

- $\exists$ — "There exists"
- $\mathbb{R}^n \to \mathbb{R}$ — "Converts n variables to a scalar value"
- $\mathcal{S} \cap N_\delta$ — "Intersection of regions defined by $\mathcal{S}$ and N_δ"
- *Strict* — "No points of identical value"
- $\setminus \{\boldsymbol{x}^\star\}$ — "Except $\boldsymbol{x}^\star$"

For example, consider the (nonlinear) univariate function $f(x)$ in the following figure. We can identify the feasible region $\mathcal{S}$, the neighbourhood around the point in question $N_\delta(x^\star)$, and the location of the local minimum.





Global Optimum (Minimum)

A point $\boldsymbol{x}^\star$ is a **global minimum** for the function $\phi : \mathbb{R}^n \to \mathbb{R}$ (maps n dimensions to a single numerical objective) on the set $\mathcal{S}$ (recall that this set defines our feasible region) if it is **feasible** ($\boldsymbol{x}^\star \in \mathcal{S}$) and if **no other feasible solution has a lower objective value**:

$$\phi(\boldsymbol{x}^\star) \leq \phi(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in \mathcal{S}$$

$$strict: \phi(\boldsymbol{x}^\star) < \phi(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in \mathcal{S} \setminus \{\boldsymbol{x}^\star\}$$

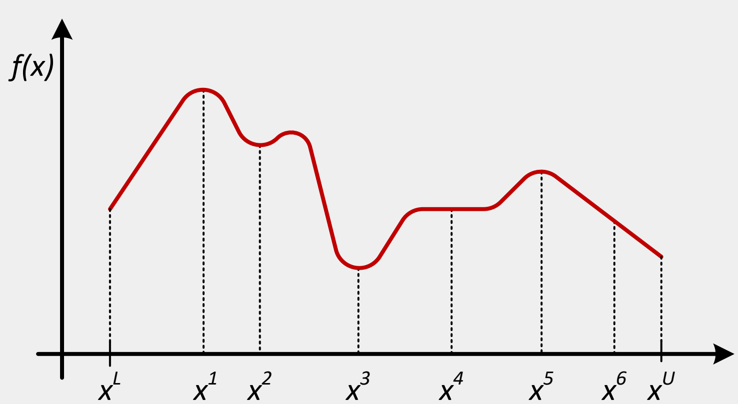
When considering types of optima consider the following **remarks**:

- Any global optimum is **also** a local optimum.
- Any local optimum **is not guaranteed** to be a global optimum (but might be).
- Definitions for local/global **maximums** are analogous to those above with the inequalities flipped.

The figure above also illustrates the location of the global minimum for the same function $f(x)$.

Class Workshop – Identifying Maxima/Minima

Consider the function in the figure below. Identify the various types of minima and maxima for the feasible region $\mathcal{S} \triangleq [x^L, x^U]$.



Workshop Solution – Identifying Maxima/Minima

Locating Optima Numerically

Recall that there are **three methods** for optimization:

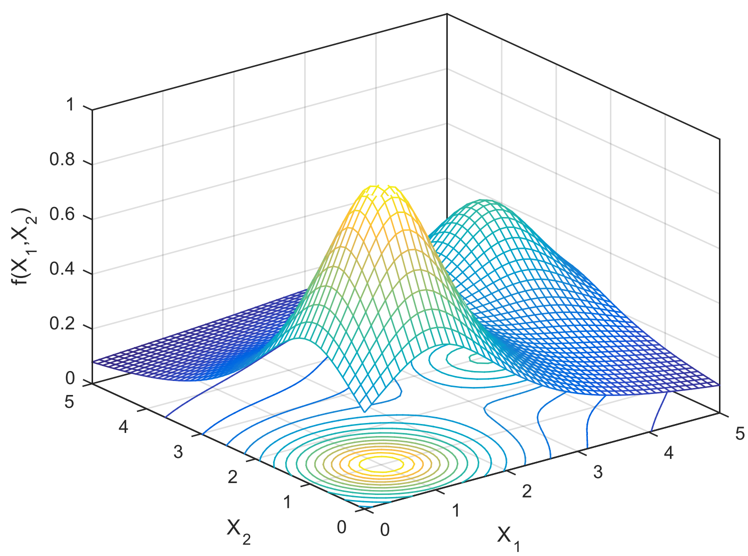
1. Graphical
   Works great if you can visualize the function; Gives insight into function behaviour

2. Analytical
   Gives exact solution; Allows for simple analysis to model changes; Not practical for most problems

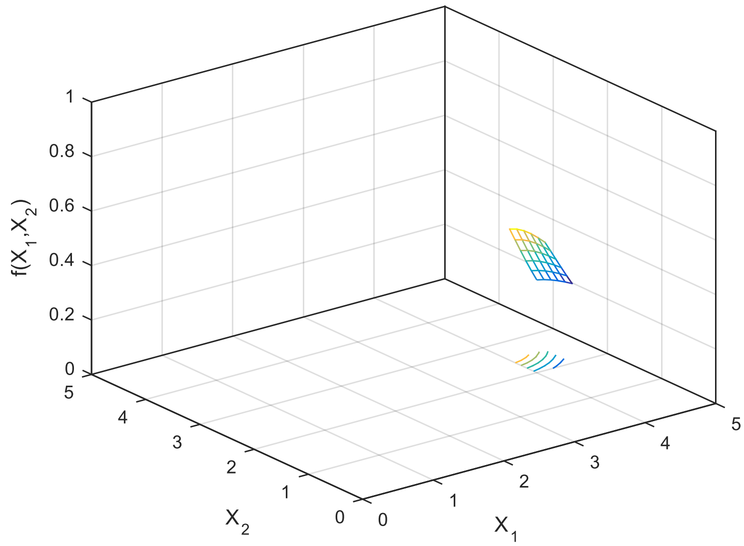
3. **NUMERICAL**
   Only practical method for large problems; Only guarantees LOCAL optima; Challenges for sensitivity analyses; Parameters can be uncertain

The primary issue with numerical optimization searches is that we only know **local information** about the function. This may lead us to finding only **local optima** instead of global because, well, we just don't know any better. For example, consider the maximization of the following function with two local maxima shown in the figure below:

$$\max_{0\leq \{x_1,x_2\} \leq 5} \phi(x_1,x_2) = \frac{1}{1+(x_1-1)^2+(x_2-1)^2} + \frac{0.5}{1+(x_1-4)^2+(x_2-3)^2}$$



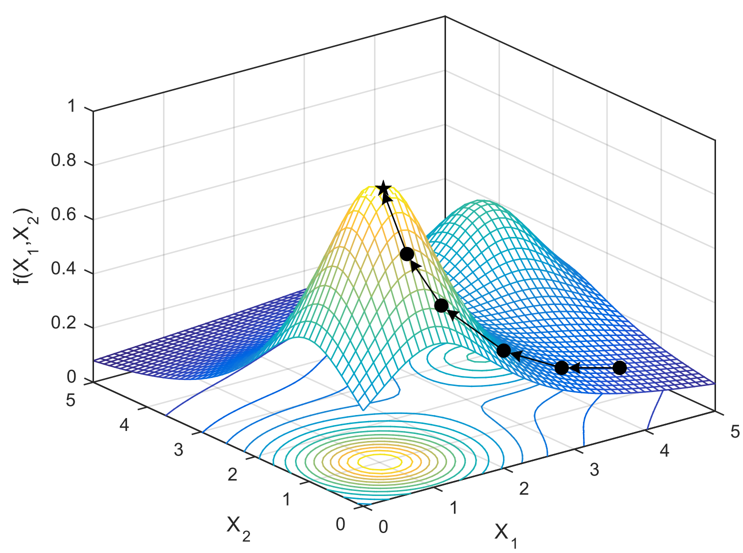
The **global** maximum is clearly at the point $(x_1, x_2) = (1,1)$, but there is also a **local** maximum at the point $(x_1, x_2) = (4,3)$. You can verify this by inspecting the equation and noticing that the denominators are minimized for these values. It is easy to see graphically where the global optimum is, but recall that a numerical optimization routine only knows **local** function information. So, for the same function at the point $(x_1, x_2) = (4.25, 2.25)$ it might only see the following:



From this point, it *seems* reasonable to want to proceed upward toward the maximum, but little does our numerical optimization routine know that going *down* will eventually lead to a higher *up*. There are ways around this, but sadly none of them can **guarantee** that we will not find a local optimum. Speaking of finding optimums, how do we accomplish this? Well, many numerical optimization methods follow the same general procedure known as the **improving search**.

Improving Search

Improving search methods are numerical algorithms that begin at a **feasible** initial condition and advance along a search path consisting of **feasible points** with **continuously improving function values**.



It therefore falls to us to determine how, at a current point $\boldsymbol{x}^{(k)}$, we identify a **direction** of change, the **magnitude** of change, and **whether I can continue to improve** once the change is made. We can write this out in the following equation defining the improving search:

General Improving Search Equation

The improving search advances from a current point $x^{(k)}$ to a new point $x^{(k+1)}$ as:

$$\boldsymbol{x}^{(k+1)} = \begin{bmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ \vdots \\ x_n^{(k+1)} \end{bmatrix} = \boldsymbol{x}^{(k)} + \alpha \Delta \boldsymbol{x} = \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \vdots \\ x_n^{(k)} \end{bmatrix} + \alpha \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_n \end{bmatrix}$$

- $\Delta \boldsymbol{x}$ is the **move direction** of a solution change at $x^{(k)}$
- $\Delta \boldsymbol{x}$ is normalized to have a length of 1 $\Rightarrow \|\Delta \boldsymbol{x}\| = 1$
- $\alpha > 0$ is the **movement magnitude**, or how far we want to pursue this direction

Obviously, we want to make sure that $\Delta \boldsymbol{x}$ makes our objective function better. Thus, we want to assure that $\Delta \boldsymbol{x}$ is an **improving direction**:

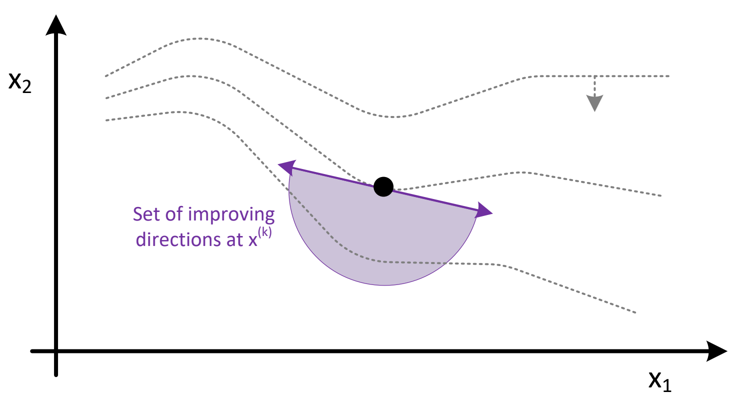
Improving Direction (Minimization)

The search direction $\Delta \boldsymbol{x}$ is an **improving direction** at the current point $\boldsymbol{x}^{(k)}$ if the objective function value at $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha \Delta \boldsymbol{x}$ is superior to that of $\boldsymbol{x}^{(k)}$ for any $\alpha > 0$ that is sufficiently small:

$$\exists\ \bar{\alpha} > 0\ :\ \phi\left(\boldsymbol{x}^{(k+1)}\right) < \phi\left(\boldsymbol{x}^{(k)}\right)\ \forall\ \alpha\ \epsilon\ (0, \bar{\alpha}]$$

For any value of α between 0 and $\bar{\alpha}$, my objective function at the new point determined via improving search must be BETTER than it was at my old point.

For example, consider the figure below, which shows the set of improving directions for a point $\boldsymbol{x}^{(k)}$ on a contour map.



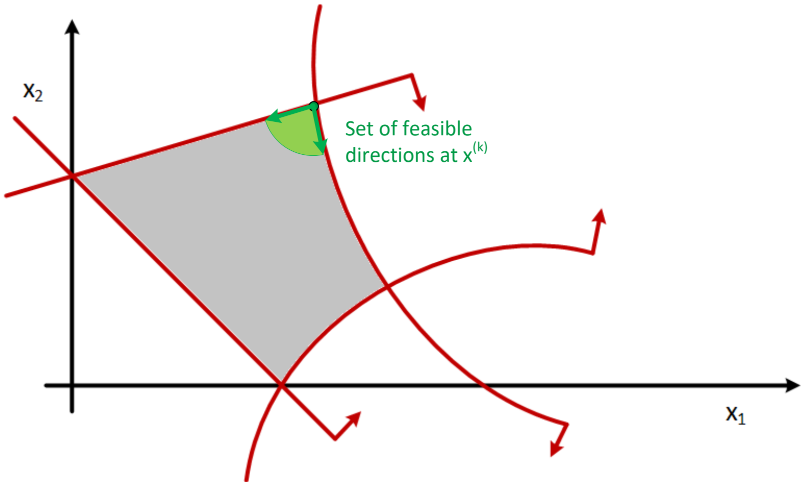
However, it is insufficient to have a direction **improve** the objective function only. We also require that it is a **feasible** direction, or that it remains inside the feasible region $\mathcal{S}$:

Feasible Direction

The search direction $\Delta \boldsymbol{x}$ is a **feasible direction** at the current point $\boldsymbol{x}^{(k)}$ if the point $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha \Delta \boldsymbol{x}$ remains in the feasible region $\mathcal{S}$ for all $\alpha > 0$ that is sufficiently small:

$$\exists\ \bar{\alpha} > 0\ :\ \boldsymbol{x}^{(k)} + \alpha \Delta \boldsymbol{x}\ \epsilon\ \mathcal{S}, \qquad \forall\ \alpha\ \epsilon\ (0, \bar{\alpha}]$$

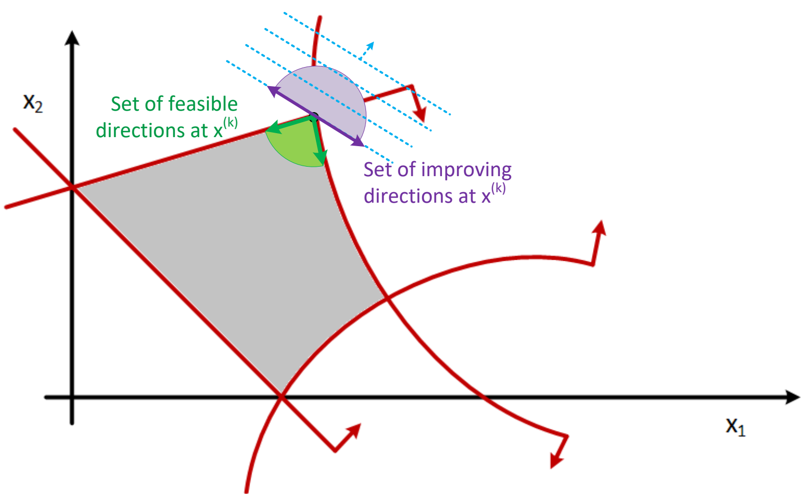
As an example, consider the figure below showing point at the corner of a feasible region. The feasible directions are only those that prevent us from exiting that region.



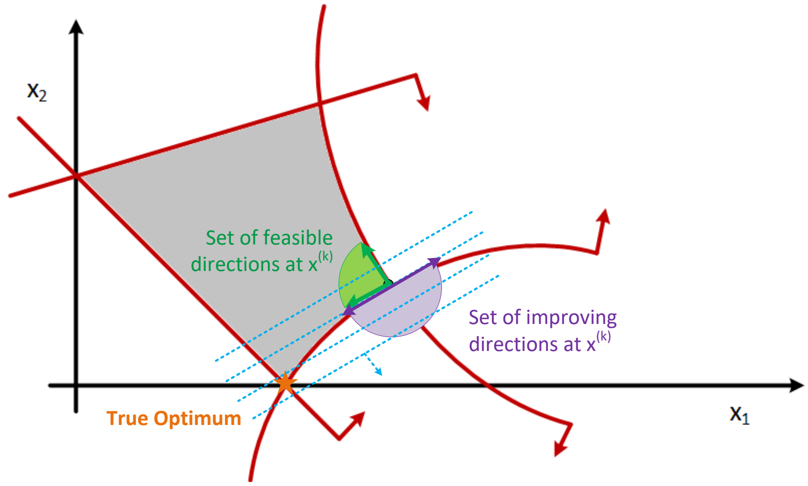
In order for any optimization method to approve a search direction, the direction must be **both feasible and improving**. If we can improve our objective function value while remaining within the feasible region, we are able to achieve a better value of the objective function. This leads us to one of the **necessary conditions of optimality**:

Necessary Condition of Optimality (NCO)
No candidate optimization model solution $\boldsymbol{x}^{(k)}$ at which an *improving feasible direction* is available can be a local (or global) optimum.

A great way to visualize the above NCO is by recognizing that we cannot be at an optimum *unless* any improving directions are **not feasible**.

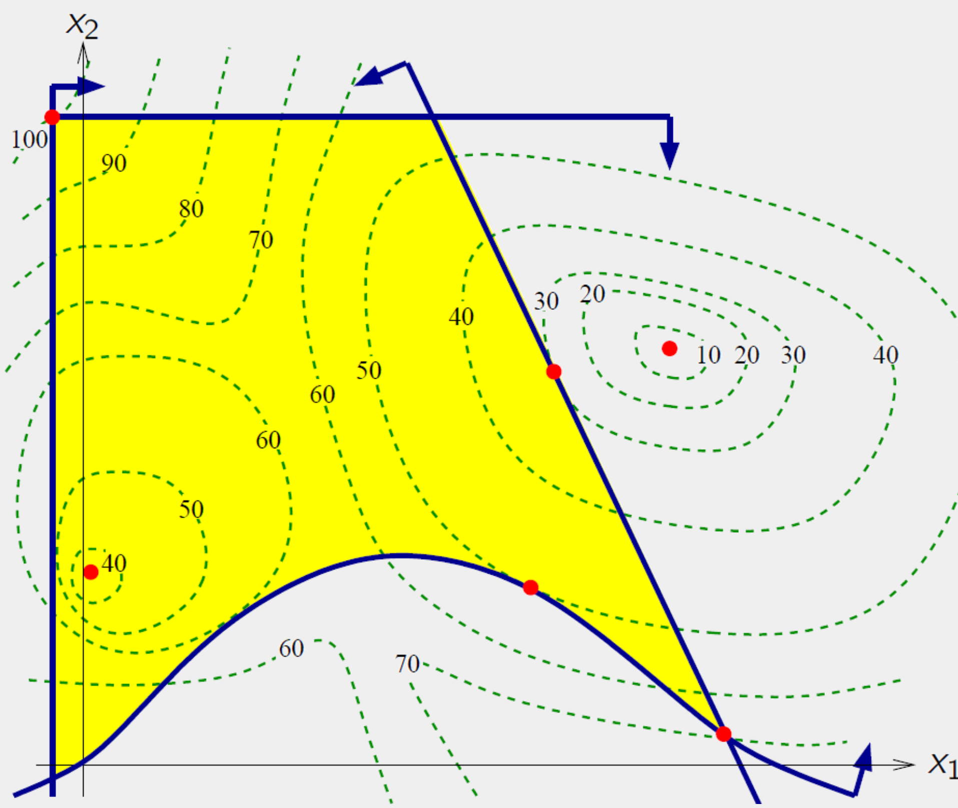


One word of CAUTION when it comes to the NCO described above. While the NCO claims that **no feasible improving search direction may exist to be at an optimum**, it **DOES NOT** guarantee that *if* no feasible improving direction exists, *then* we are at an optimum. This is very important. One example of this scenario is called a **saddle point**. We will cover this in more detail later (it only exists in nonlinear programs that are non-convex, which will be discussed below), but for now we can visualize another scenario, a **local optimum**, on the same figure as above (with different objective contours):



Class Workshop – Identifying Constrained Maxima/Minima

Determine if each of the points in the figure below is a local/global minimum, a local/global maximum, or neither (figure courtesy of Dr. Chachuat). Label the points as necessary!



Workshop Solution – Identifying Constrained Maxima/Minima

Now that we know the definitions of optimality and feasible/improving search directions, we may define the **[continuous] improving search algorithm**. As for numerically computing a feasible improving direction $\Delta\boldsymbol{x}$, we are not concerned about that (yet). Each type of optimization model (LP, NLP, *etc*.) has its own methods of computing the actual directions. We are just going to focus on the *idea* for now.

Algorithm: Continuous Improving Search

1. **INITIALIZATION**
   Select any feasible starting point $\boldsymbol{x}^{(0)}$ with counter $k = 0$.

2. **MOVE DIRECTION**
   If no improving feasible direction $\Delta\boldsymbol{x}$ can be found, STOP.
   OTHERWISE, determine an improving feasible direction $\Delta\boldsymbol{x}$.

3. **STEP SIZE**
   If there is *no limit* to the improvement when going in the direction $\Delta\boldsymbol{x}$ while also maintaining feasibility, STOP: Model is *unbounded*.
   OTHERWISE, choose the largest allowable step size α that improves the objective while remaining feasible.

4. **UPDATE**
   $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha\Delta\boldsymbol{x}$
   $k = k + 1$
   Return to step (2)

Finally, when it comes to the improving search algorithm, consider the following **remarks**:

- The basic improving search may terminate at a suboptimal (saddle) point.
- The basic improving search *cannot* distinguish between local and global optima.

Class Workshop – Improving Search

1. What are some (conceptual) ways to identify a search direction?
2. What are some (conceptual) ways to determine the search length?
3. Are there any practical implementation restrictions to your ideas?

Workshop Solution – Improving Search

Convexity

One thing that can make our lives *MUCH* easier in optimization is to deal with optimization models that are **convex**. This may sound familiar from your high school days, and in fact it means the same thing. However, we need to be a little stricter in how we define convexity.

Convex Set

A set $\mathcal{S} \subset \mathbb{R}^n$ is said to be **convex** if **every point** on a line connecting **any two points** $\{\boldsymbol{x}, \boldsymbol{y}\}$ **in** $\mathcal{S}$ is ALSO in the region $\mathcal{S}$:

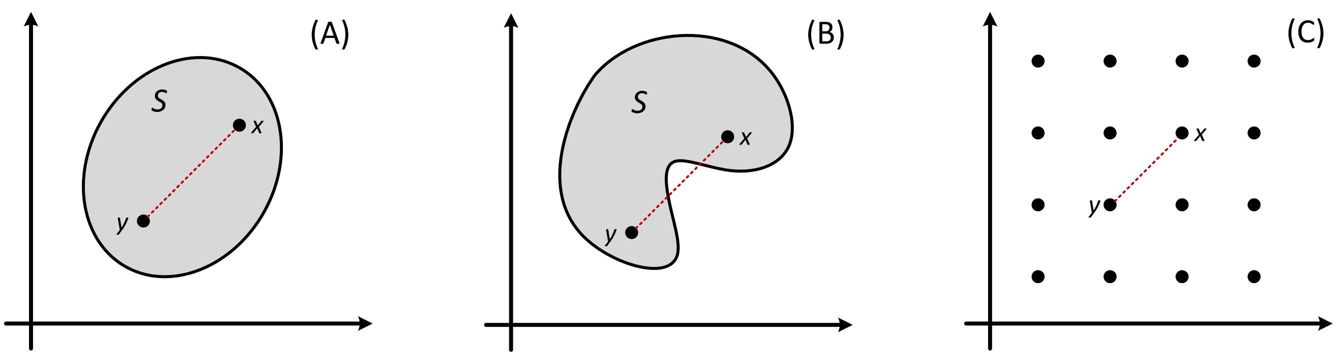
$$\gamma \boldsymbol{x} + (1-\gamma)\boldsymbol{y} \in \mathcal{S}, \quad \forall\, \gamma \in [0,1]$$

NB – Any non-connected sets (discrete sets) are nonconvex.

Some **REMARKS** on the notation in the math above

- $\subset$ "Subset"
- $\mathcal{S} \subset \mathbb{R}^n$ "$\mathcal{S}$ is a SUBSET of $\mathbb{R}^n$"

This can be visualized in the figure below. The region $\mathcal{S}$ shown in panel (A) is convex because any line drawn from one point in the region to another is entirely enclosed by the region. In panel (B), the line leaves the region $\mathcal{S}$ for a period and is therefore nonconvex. The set $\mathcal{S}$ described in panel (C) corresponds to a discrete set and is thus also nonconvex.



Now that we know what a convex set is, we can define a convex function. It is important to note that a convex function cannot exist unless it is defined on a convex set. Also, note that ***any*** continuous range of numbers is a convex set.

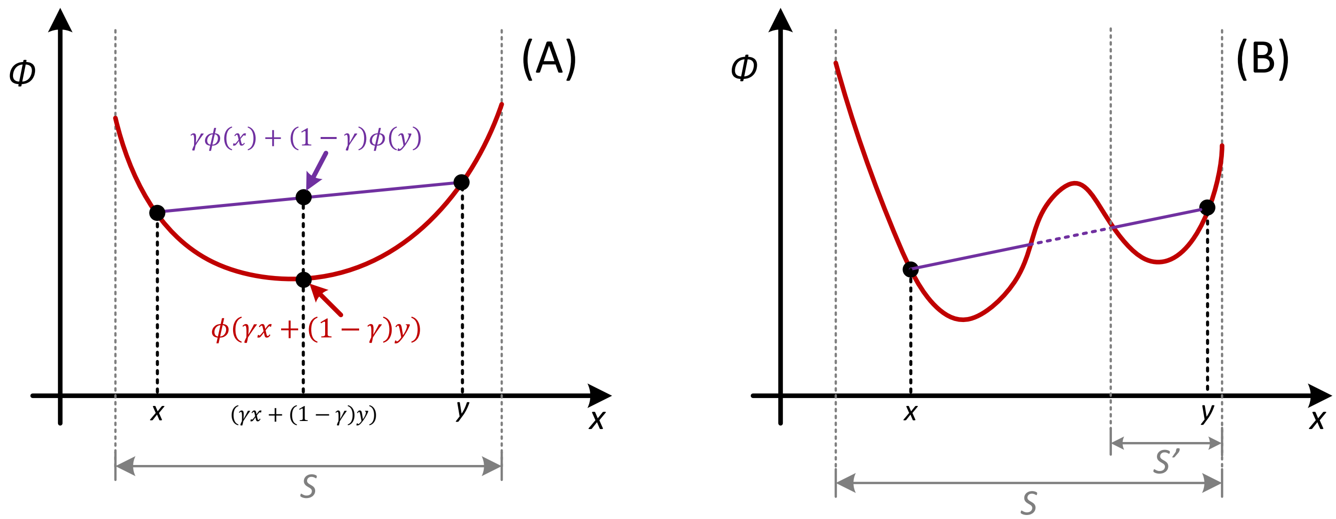
Convex Function

A function $\phi: \mathcal{S} \to \mathbb{R}$, defined over a convex set $\mathcal{S}$, is said to be **convex on** $\mathcal{S}$ if the line segment connecting $\phi(\boldsymbol{x})$ and $\phi(\boldsymbol{y})$ for ANY two points $\{\boldsymbol{x}, \boldsymbol{y}\} \in \mathcal{S}$ exists exclusively **above** the function evaluations between points $\boldsymbol{x}$ and $\boldsymbol{y}$:

$$\phi(\gamma \boldsymbol{x} + (1-\gamma)\boldsymbol{y}) \leq \gamma\phi(\boldsymbol{x}) + (1-\gamma)\phi(\boldsymbol{y}), \ \ \forall\, \gamma \in [0,1]$$

- **STRICT** convexity is achieved by swapping the $\leq$ with a $<$ in the above expression.
- A function is said to be [strictly] **CONCAVE** on $\mathcal{S}$ if $(-\phi)$ is [strictly] convex on $\mathcal{S}$

An example of a convex function defined over a convex set $\mathcal{S}$ is shown in the figure below (panel A). A nonconvex function on the same set $\mathcal{S}$ is shown in panel (B). Note however that the function in panel (B) is convex on the set $\mathcal{S}' \subset \mathcal{S}$.



Defining Sets with Constraints

When it comes to optimization problems, the **feasible region** defines our **set of possible solutions** (search space, if you will). You will recall that the feasible region is the intersection of the domains defined by our set of constraints. This leads us to a very important result: If we define our search space as the feasible region consisting of a set of convex constraints, our feasible region represents a convex set. We are not going to prove this, but it is worth noting the formal result.

Convex Sets via Inequality Constraints

A set defined as the intersection of a series of inequalities $\boldsymbol{g}(\boldsymbol{x}) \triangleq \{g_1(\boldsymbol{x}), g_2(\boldsymbol{x}), \cdots\}$:

$$\mathcal{S} \triangleq \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{g}(\boldsymbol{x}) \leq 0\}$$

IS CONVEX if $\boldsymbol{g}(\boldsymbol{x})$ is a set of functions convex in $\mathbb{R}^n$.
Moreover, the set $\mathcal{S} \triangleq \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{g}(\boldsymbol{x}) \geq 0\}$ is convex if $-g(\boldsymbol{x})$ is convex (or $g(\boldsymbol{x})$ is concave)

Convex Sets via Equality Constraints

A set defined as the intersection of a series of equalities $\boldsymbol{h}(\boldsymbol{x}) \triangleq \{h_1(\boldsymbol{x}), h_2(\boldsymbol{x}), \cdots\}$:

$$\mathcal{S} \triangleq \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{h}(\boldsymbol{x}) = 0\}$$

IS CONVEX *if and only if* $\boldsymbol{h}(\boldsymbol{x})$ is affine (allows for preservation of parallel relationships).

It might be worth pointing out here that "affine" is a fancy term for a function (or set of functions) that have constant slope. This is different from a linear function (or set of functions) in that linear functions are fixed to the origin, whereas affine functions are linear functions translated off the origin.

Mathematically, you would say a function $f(\boldsymbol{x}) = A\boldsymbol{x}$ is linear, and $f(\boldsymbol{x}) = A\boldsymbol{x} + \boldsymbol{b}$ with $\boldsymbol{b}$ a vector of constants is affine. You'll probably read "affine" as "linear" from now on, but hey I've done what I can to describe the difference.

Class Workshop – Convex Sets

Consider the definitions of convex sets defined by constraints above. Why do you think that the equality constraint definition says "if and only if", whereas the inequality constraint only uses "if"? It is a subtle difference, but there is a good explanation!

Workshop Solution – Convex Sets

These are the conditions that define convex constraints. Since the intersection of convex sets is ALSO convex, we can combine the two definitions above to claim that any set $\mathcal{S}$ that is defined as the region defined by any convex inequalities *and* equalities is convex, or:

$$\mathcal{S} \triangleq \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{g}(\boldsymbol{x}) \leq 0 \,, \boldsymbol{h}(\boldsymbol{x}) = 0\}$$

Is convex if $\boldsymbol{g}(\boldsymbol{x})$ and $\boldsymbol{h}(\boldsymbol{x})$ are convex and affine, respectively.

Now that we know what makes a convex set, we can finally describe a convex program and come to a critical conclusion about convexity and global optima. The notion of convex programs helps us immensely when making conclusions about the optimum for a given problem.

Convex Optimization program

The constrained minimization program of the form:

$$\begin{aligned} \min_{\boldsymbol{x}} \phi &= f(\boldsymbol{x}) \\ \textit{s.t.} & \\ \boldsymbol{h}(\boldsymbol{x}) &= 0 \\ \boldsymbol{g}(\boldsymbol{x}) &\leq 0 \end{aligned}$$

Is called a **Convex Optimization Program** if $f(\boldsymbol{x})$ and $\boldsymbol{g}(\boldsymbol{x})$ are convex and $\boldsymbol{h}(\boldsymbol{x})$ is affine.

Something that is **very** important about convex programs are that they have a well-defined sufficient condition for **GLOBAL** optimality:

Sufficient Condition for Global Optimality

A (strict) ***local*** optimum of a convex program is also a (strict) ***global*** optimum. If the problem is non-convex, any local optimum *might* also be global, but this is not guaranteed.

Conclusions

OK! That was a lot of math. Boo, right? Well, as it turns out, learning the math now should make things much easier to explain in the future (I hope). Being able to recognize a convex program will do us wonders when it comes to solving optimization problems since we can be sure that the solution we have found is the true global optimum. As we will find out shortly, **linear programs** are convex, and allow us to use some nifty tricks to find the best possible solution in very little time.

How might this material be assessed? Well, it is certainly fair for me to ask you to solve an optimization problem early in this course by hand (analytically). I would then ask you to *prove* whether you can guarantee your solution is a local or global optimum through the power of convexity. If your program is not convex, you might have to plot things out to show why and where this is not so. Moreover, you may be asked to show that a certain feasible region is convex mathematically, graphically, and so on.

The recurring mathematical ideas (feasible directions and improving directions and the general improving search algorithm) will be explored in significant detail throughout the course since each type of problem we may encounter will have its own applications of these topics.

~~ END OF MODULE ~~

\|T|/