CHEMICAL ENGINEERING 4G03

Module 08

Nonlinear Programming (I)

Mathematical Concepts

Dr. Jake Nease
Chemical Engineering
McMaster University

Updated March 21, 2023

Outline of Module

Suggested Readings

Rardin (1st edition): Chapter 13.3-13.4

Rardin (2nd edition): Chapter 16.3-16.4

Introduction and Perspective

OK, this is where things begin to get a little weird. We are going to revisit the mathematical concepts for optimization in this section so we may exploit them to know whether or not we are at candidate optima. Unfortunately, unlike the linear programming case, nonlinear programming contains a huge number of **nuances and local behaviours** that we have to watch out for. There is no easy way to do this, BUT the good news is that we can **combine mathematical theory (this section) with numerical methods (next section)** to locate and characterize optima.

In this section we will explore some of the mathematical concepts of **unconstrained nonlinear optimization**. I realize that most practical optimization problems are constrained (but not all! Think of regression), but it is a fact that *many nonlinear optimization problems include the solutions of unconstrained sub-problems*. Our discussion here will focus mainly on the objective function since we do not consider constraints, and will include:

- Optimality conditions for single-variable objectives.
- Optimality conditions for multi-variable objectives.
- A revisit to convex optimization objective functions.

Single-Variable Objective Functions

Many will argue that single-variable objective functions are far and few between, and you may be right. However, certain sub-problems for *multi-variable optimization* such as Line Search problems are actually single-variable.

Algebraic Characterizations of Local Optima

Let's return back to Module 02. Where we defined the existence of a local optimum as a point in which a sufficiently small neighbourhood surrounding it had an objective function value no lower (minimization) or higher (maximization) than the candidate optimum.

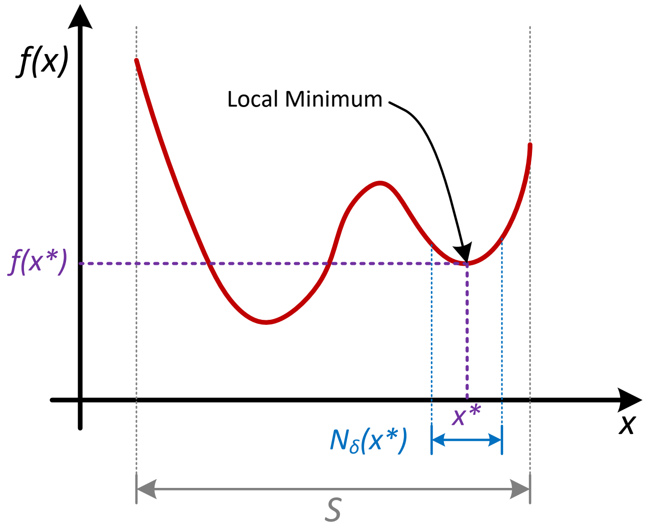
Local Optimum (Minimum)

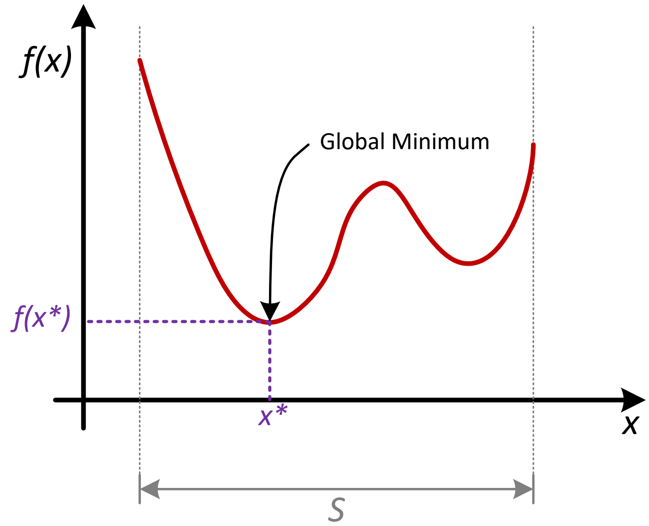
A point $x^\star$ is a **local minimum** for the function $\phi : \mathbb{R}^n \to \mathbb{R}$ (maps n dimensions to a single numerical objective) on the set $\mathcal{S}$ (recall that this set defines our feasible region) if it is **feasible** ($x^\star \,\epsilon\, \mathcal{S}$) and if a sufficiently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

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

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

We illustrated this fact using the figure below, in which no points in the neighbourhood N_δ were better than the candidate optimum $x^\star$. That is fine, but as some of you might recall from assignment 1, showing that something is convex (more on this later) and/or a local optimum using the definition above is quite tedious. There is a better way, and it involves the use of a **fundamental theorem of calculus**!





The definition of the **derivative** of a function, for those of you that may not recall, is:

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Should this derivative exist and be continuous, we have:

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \geq 0 \quad \forall\, h \in (0, \delta) \ \Rightarrow f'(x) \geq 0$$

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \leq 0 \quad \forall\, h \in (-\delta, 0) \ \Rightarrow f'(x) \leq 0$$

As we shrink the neighbourhood size δ to zero, we obtain the so-called **1st-Order Necessary Condition of Optimality**

1st-Order Necessary Condition of Optimality

Any local optimum must exist at a point $x^\star$ at which the objective function $\phi(x^\star)$ has a first derivative identically equal to zero. The point $x^\star$ is known as a **stationary point**.

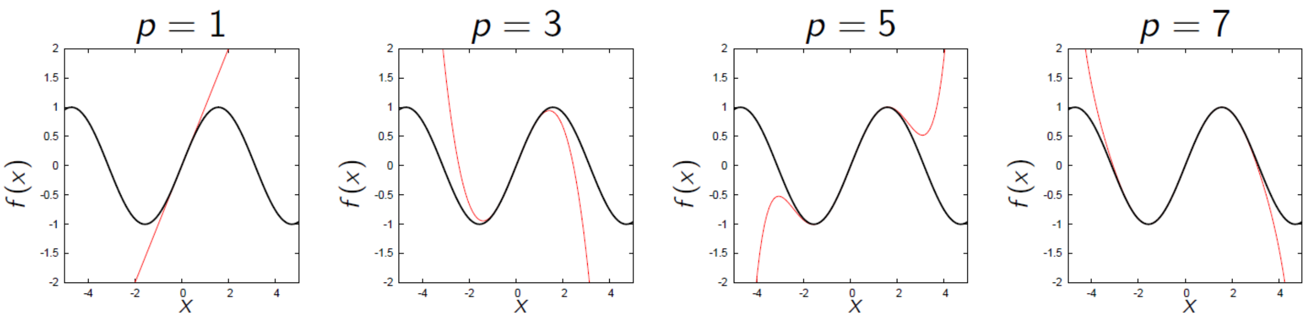
$$\boldsymbol{x}^\star \ local\ optimum \Rightarrow \phi'(\boldsymbol{x}^\star) = 0$$

NB – this only works ONE WAY. IF $x^\star$ is a local optimum, THEN it is a stationary point. If $x^\star$ is a stationary point, it MIGHT be a local optimum, but this is not guaranteed (hence "necessary").

OK, that is useful! We know that the derivative of the objective function at any potential optimum MUST be equal to zero. **Does this also correspond to linear programs?** As it turns out, it does not, because linear programs are *constrained* (we are dealing with unconstrained NLPs at the moment). Our next task is to identify whether or not we are at a **local minimum or maximum**, not just a stationary point. For this purpose I would like to remind you of the **Taylor Series Approximation of a function $\boldsymbol{f(x)}$**:

$$f(x+h) \approx f(x) + hf'(x) + \frac{h^2}{2} f''(x) + \cdots + \frac{h^p}{p} f^{(p)}(x)$$

We may recall from math or 3E04 that the Taylor Series approximation of a function some distance h from a centering point x improves the more terms we add. Equivalently, the approximation becomes valid for a greater range of h values as the number of terms are added as well, such as in the figure below which illustrates the approximation of $f(x) = \sin(x)$ at the candidate point $x = 0$[1].



If we truncate with the first two terms of the Taylor series we get:

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2} f''(\xi)$$

Where $f''(\xi)$ represents the **truncation error** corresponding to the remaining terms left out of the expression. Furthermore, $f'(x) = 0$ at a stationary point $x^\star$. We therefore have:

$$f(x^\star + h) = f(x^\star) + \underbrace{\boldsymbol{hf'(x^\star)}}_{=\mathbf{0}} + \frac{h^2}{2} f''(\xi)$$

We may re-arrange this equation to explicitly solve for the **second-derivative error term** as follows:

$$f''(\xi) = \frac{2}{h^2}[f(x^\star + h) - f(x^\star)]$$

We know that the **inner term** of the above expression is ≥ 0 when $x^\star$ is a *local minimum*, and ≤ 0 when $x^\star$ is a *local maximum* due to the first-order sufficient condition outlined above. We can therefore arrive at the **2nd-Order Necessary Condition of Optimality**:

2nd-Order Necessary Condition of Optimality

Any local minimum at a stationary point $x^\star$ with the twice-differentiable objective function $\phi(x^\star)$ has a second derivative greater than or equal to zero. The opposite is true for a local maximum.

$$\boldsymbol{x}^\star \ local\ minimum \Rightarrow \phi''(\boldsymbol{x}^\star) \geq 0$$

$$\boldsymbol{x}^\star \ local\ maximum \Rightarrow \phi''(\boldsymbol{x}^\star) \leq 0$$

NB – this only works ONE WAY (hence "necessary" again).

I am sure you guys remember high school. I mean, I remember high school, and I am older than the average 4th/5th year undergrad. THIS is the reason that your math teachers used to say "you have a minimum if the second derivative is positive" and so-on. It turns out that this claim is grounded in actual mathematical theory, and that theory relies heavily on the Taylor Series Expansion of a function. This is not the last time we will see it, either. If you have ever taken 3E04 with me, you will know that the Taylor Series is huge in numerical methods. OK. Enough talk. Let's practice!

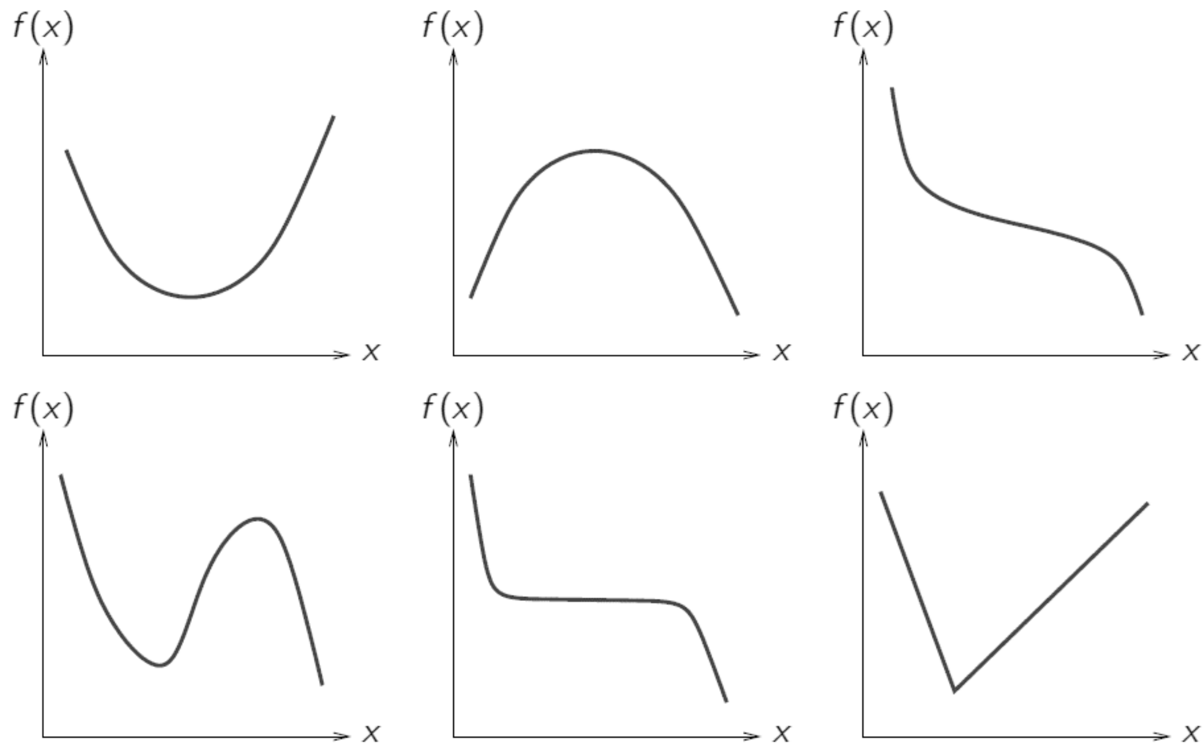
[1] Figure courtesy of B. Chachuat.

Class Workshop – Stationary Points / 1st and 2nd Order Necessary Conditions

Consider the functions in the workshop solution bubble below.

1. Identify the stationary points.
2. Determine (graphically) if the stationary points are local maxima, minima, or saddle points.
3. Conclude whether or not each stationary point satisfies the 1st and 2nd necessary conditions for optimality. **PERSPECTIVE** – computer algorithms can't see the function, so they rely on these conditions alone.

Workshop Solution – Stationary Points / 1st and 2nd Order Necessary Conditions



Graphical interpretations are nice, but it is much more useful for us to understand how the computer does things – which means we need to try things algebraically as well.

Class Workshop – Applying 2nd Order Necessary Conditions

For each unconstrained NLP below, identify any stationary points and identify them as potential optima by applying the 2nd order necessary conditions.

1. $\min_{x} \phi = 3 + 2x + 8x^2$

2. $\min_{x} \phi = 1 + 2x^3$

Workshop Solution – Applying 2nd Order Necessary Conditions

OK, that is all well and good, but we know that necessary conditions are a little bit weak. We know they **must be true at an optimum**, but they don't really help us conclude **whether or not we are actually at an optimum** (kind of weird, right?). What we want is to therefore derive some **second order sufficient conditions for optimality**. This is done in a similar fashion using the Tayler Series expansion, but we will forego the derivation for the sake of time and frustration. I can put up the derivation at your request, or you can check out the textbook. Either way:

2nd-Order Sufficient Conditions of Optimality

Consider a function $\phi(x^\star)$ with $x^\star$ a stationary point. If the function may have $2n$ derivatives taken (in other words, it is $2n$ continuous or $\mathcal{C}^{(2n)}$ and the first non-zero derivative evaluated at $x^\star$ corresponds to the $2n^{\text{th}}$ derivative, the stationary point $x^\star$ is a local minimum (maximum) if $\phi^{(2n)} > 0$ (< 0):

$$\begin{matrix} \phi'(x^\star) = \cdots = \phi^{(2n-1)}(x^\star) = 0 \\ \phi^{(2n)} > 0 \end{matrix} \quad \Rightarrow \quad \boldsymbol{x^\star} \textbf{ strict local minimum}$$

$$\begin{matrix} \phi'(x^\star) = \cdots = \phi^{(2n-1)}(x^\star) = 0 \\ \phi^{(2n)} < 0 \end{matrix} \quad \Rightarrow \quad \boldsymbol{x^\star} \textbf{ strict local maximum}$$

$$\begin{matrix} \phi'(x^\star) = \cdots = \phi^{(2n)}(x^\star) = 0 \\ \phi^{(2n+1)} \neq 0 \end{matrix} \quad \Rightarrow \quad \boldsymbol{x^\star} \textbf{ saddle point}$$

NB – Note the direction of the ⇒ (implies) has changed in this scenario.

This is **VERY USEFUL** – we now have a way to know algebraically whether or not we have found a saddle point or a local max/min. Of course, we run into difficulty when the function is infinitely differentiable, but there are worse things. Furthermore, we can't really use this logic in the event that we have **constrained optimization functions**, which is why many constrained optimization methods use penalty or barrier functions (such as Lagrange Multipliers) along with the sufficient and necessary conditions above.

Class Workshop – Applying 2nd Order Sufficient Conditions

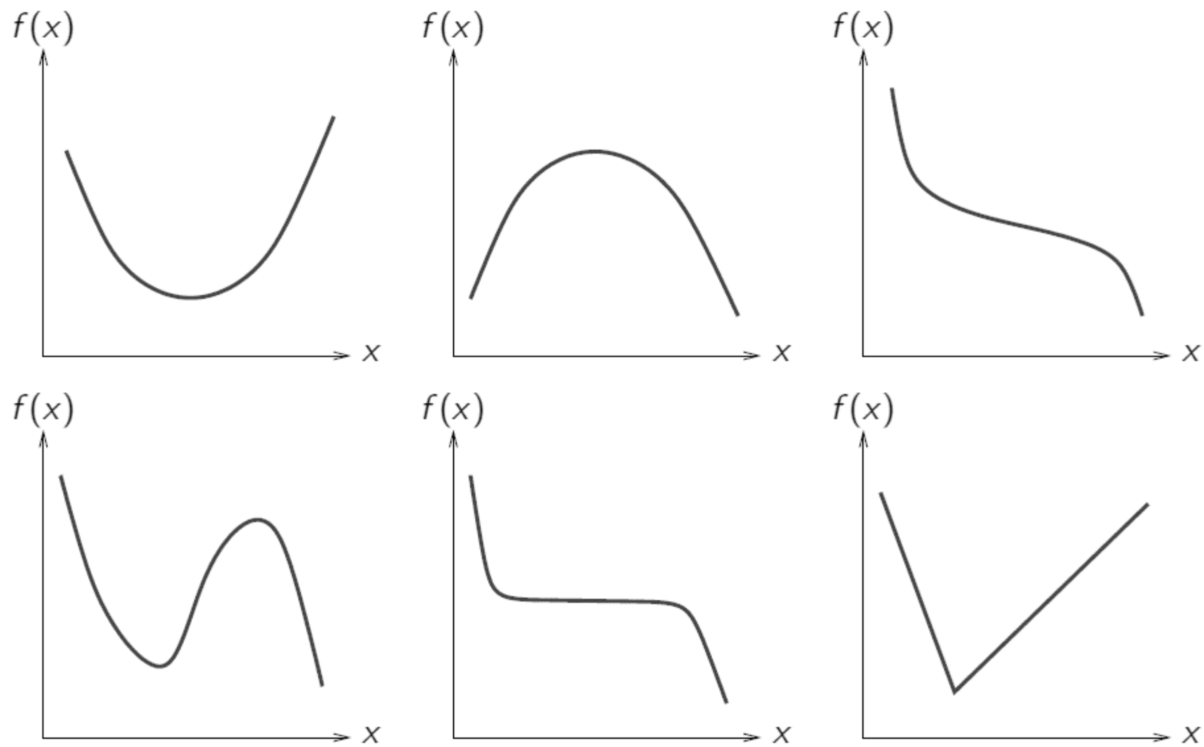
GRAPHICALLY identify on the plots below which points satisfy the 2nd order sufficient conditions of optimality for local optima and saddle points. Then, algebraically check whether the candidate optima for the NLPs below correspond to strict local minima or maxima.

1. $\min_x \phi = 3 + 2x + 8x^2$

2. $\min_x \phi = 1 + 2x^3$

3. $\min_x \phi = 1 + x^6$

Workshop Solution – Stationary Points / 1st and 2nd Order Necessary Conditions



Multi-Variable Objective Functions

Math Refresher – Gradients and Hessians

As a reminder in case you have forgotten calculus (that would never happen though, right?), we can extend our findings for single-variable functions into the multi-dimensional space with a little extra effort and a few definitions. Consider any function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ at some point $\boldsymbol{x} \in \mathbb{R}^n$.

- The **GRADIENT** $\nabla f(\boldsymbol{x})$ is the vector of first partial derivatives **evaluated at** $\boldsymbol{x}$.
- The **HESSIAN** $H(\boldsymbol{x})$ is the matrix of second partial derivatives **evaluated at** $\boldsymbol{x}$**.**

Gradient and Hessian of $\boldsymbol{f(x)}$

$$\nabla f(\boldsymbol{x}) \triangleq \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \frac{\partial f}{\partial x_2}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{bmatrix} \qquad H(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2}(x) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(x) & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(x) & \frac{\partial^2 f}{\partial x_2^2}(x) & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(x) & \frac{\partial^2 f}{\partial x_n \partial x_2}(x) & \cdots & \frac{\partial^2 f}{\partial x_n^2}(x) \end{bmatrix}$$

Features

- The gradient describes the *rate of change* of $f(\boldsymbol{x})$ with small increments to x_1, x_2, … , x_n around the point $\boldsymbol{x}$.
- The Hessian describes the *curvature* of $f(\boldsymbol{x})$ with respect to all directions around the point $\boldsymbol{x}$.
- The Hessian is *symmetric* if $f(\boldsymbol{x})$ is twice differentiable (C^2).

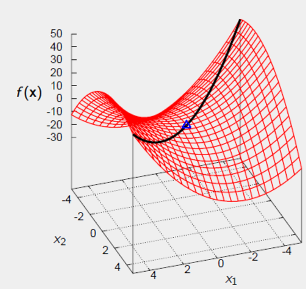
With these tools in the bag, we can now extend our Taylor Series approximation to multiple dimensions! Infinite dimensions, even! The only key difference here is that instead of taking the Taylor Series "around" point x, I need to do it **in some direction around point** $\boldsymbol{x}$. This is key because in multiple dimensions I have to give myself some sort of reference direction. If I assign this direction as δ, and treat $f'(x)$ and $f''(x)$ as $\nabla f(x)$ and $H(x)$ respectively, we have:

$$f(\boldsymbol{x} + h\boldsymbol{\delta}) = f(\boldsymbol{x}) + h\nabla f(\boldsymbol{x})^T \boldsymbol{\delta} + \frac{h^2}{2} \boldsymbol{\delta}^T H(x + \xi\boldsymbol{\delta})\boldsymbol{\delta}$$

Noting above that $0 \leq \xi \leq h$ (the error term is no greater than h).

Class Workshop – Taylor Approximation of Multivariate Functions

Find the second-order Taylor Series approximation of the function $f(x) = \frac{3}{2}x_1^2 - x_2^2 + x_1 x_2$ at the point $x = (0,0)$ and in the direction $\delta = (1,1)$, corresponding to the figure at the right.



Workshop Solution – Taylor Approximation of Multivariate Functions

Conditions of Optimality for Multivariate Objectives

Now, using the same logic as before, for ANY direction $\boldsymbol{\delta}$ for a local minimum $x^\star$ we have:

$$f'(x) = \lim_{h\to 0}\frac{f(x^\star + h\delta) - f(x^\star)}{h} \geq 0 \quad \forall\, h \in (0,\delta) \quad \Rightarrow \frac{d}{dh}[f(x^\star + h\delta)] \geq 0$$

$$f'(x) = \lim_{h\to 0}\frac{f(x^\star + h\delta) - f(x^\star)}{h} \leq 0 \quad \forall\, h \in (-\delta,0) \quad \Rightarrow \frac{d}{dh}[f(x^\star + h\delta)] \leq 0$$

Which arrives us at the (similar) **1st Order Necessary Condition for Optimality for a Multivariate Function**:

1st-Order Necessary Condition of Multivariate Optimality

Any local optimum must exist at a point $\boldsymbol{x}^\star$ at which the objective function $\phi(\boldsymbol{x}^\star)$ has a gradient identically equal to zero. The point $x^\star$ is known as a **stationary point**.

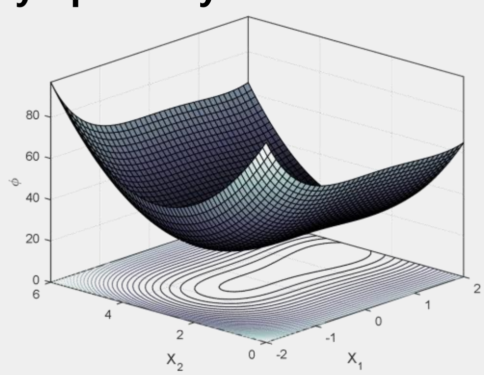
$$\boldsymbol{x^\star \; local\ optimum \Rightarrow \nabla\phi(x^\star) = 0}$$

NB – this only works ONE WAY. IF $x^\star$ is a local optimum, THEN it is a stationary point. If $x^\star$ is a stationary point, it MIGHT be a local optimum, but this is not guaranteed (hence "necessary").

Class Workshop – Multivariate 1st Order Necessary Optimality

Find the candidate optima (stationary points) of the following unconstrained optimization program:

$$\min_{x} \phi = 20 + x_1^3(x_1 - 2) + 5(x_2 - 3)^2$$



Workshop Solution – Multivariate 1st Order Necessary Optimality

NOW, if I go ahead and apply the same logic with regards to the second derivative as with the univariate Taylor's theorem, I get:

$$\boldsymbol{\delta}^T H(\boldsymbol{x}^\star + \xi\boldsymbol{\delta})\boldsymbol{\delta} = \frac{\mathbf{2}}{\boldsymbol{h}^2}[f(\boldsymbol{x}^\star + h\boldsymbol{\delta}) - f(\boldsymbol{x}^\star)] \geq \mathbf{0}$$

Taking the limit as $h \to 0$ we achieve:

$$\boldsymbol{\delta}^T H(\boldsymbol{x}^\star)\boldsymbol{\delta} \geq 0 \;\; \forall \boldsymbol{\delta} \in \mathbb{R}^n$$

This looks very similar to the second-order necessary condition of optimality in the univariate case, but we have that pesky vector δ pre- and post-multiplying. What can we do about this? As it turns out, there is a much more useful result from this than you might realize. First, notice that this has to happen or ALL δ, including those with negative components. Furthermore, recall that $\boldsymbol{H}(\boldsymbol{x})$ **is symmetric for any twice-differentiable function**. We can thus claim that the expression $\boldsymbol{\delta}^T H(\boldsymbol{x}^\star)\boldsymbol{\delta} \geq 0$ **if and only if** the matrix $H(x)$ is **positive (semi)definite.**

Matrix Definiteness

Any $n \times n$ matrix A is said to be:

- **POSITIVE DEFINITE** if $\delta^T A \delta > 0 \;\forall\; \delta \neq 0$
- **POSITIVE SEMIDEFINITE** if $\delta^T A \delta \geq 0 \;\forall\; \delta$
- **INDEFINITE** if $\delta^T A \delta \{>, <, =\} 0 \;\forall\; \delta \neq 0$

OK, that is fine, but which types of matrices are **easy to check** for definiteness? We can't just use any matrix… It would be way too much work to check it against all possible δ vectors. We need something that will be guaranteed to work for **any** δ, after all. Consider the **diagonal matrix** Λ:

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_n \end{bmatrix}$$

In this case, the expression $\delta^T \Lambda \delta$ corresponds to the **equivalent statements**:

- $\delta^T \Lambda \delta > 0 \;\forall \delta \neq 0 \;\;\Leftrightarrow\;\; \lambda_1 \dots \lambda_n > 0$
- $\delta^T \Lambda \delta \geq 0 \;\forall \delta \;\;\Leftrightarrow\;\; \lambda_1 \dots \lambda_n \geq 0$

Not only that, but the diagonal matrix Λ can be formed through the **Eigenvalue Decomposition** of matrix A. In other words, all $n \times n$ invertible matrices that are invertible have Eigenvalues λ corresponding to the solutions of the equation:

$$\det(A - \lambda I) = 0$$

Not only this, but **REAL Eigenvalues are guaranteed to exist for ANY real symmetric matrix A**. What's more is that we know for a fact that any twice-differentiable function has a Hessian that is real and symmetric!

Theorem: Characterization of Positive Definiteness

Any $n \times n$ matrix A is said to be:

- **POSITIVE DEFINITE ($\succ$)** *if and only if* all Eigenvalues $\lambda > 0$
- **POSITIVE SEMIDEFINITE ($\succcurlyeq$)** *if and only if* all Eigenvalues $\lambda \geq 0$

Great! This is much more useful than $\boldsymbol{\delta}^T H(\boldsymbol{x}^\star)\boldsymbol{\delta} \geq 0$. We may now finally declare the **2nd Order Necessary Conditions for Multivariate Optimality**:

2nd-Order Necessary Condition of Multivariate Optimality

Any local minimum at a stationary point $\boldsymbol{x}^\star$ with the twice-differentiable objective function $\phi(\boldsymbol{x}^\star)$ has a Hessian with Eigenvalues greater than or equal to zero. The opposite is true for a local maximum.

$$\boldsymbol{x}^\star \textit{ local minimum} \Rightarrow H(\boldsymbol{x}^\star) \succcurlyeq 0$$

$$\boldsymbol{x}^\star \textit{ local maximum} \Rightarrow H(\boldsymbol{x}^\star) \preccurlyeq 0$$

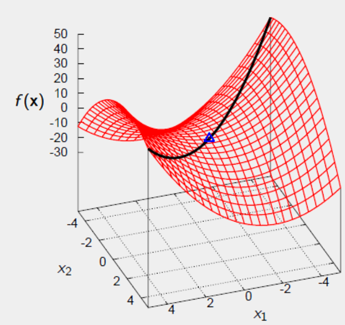
NB – If $x^\star$ is a saddle point, $H(\boldsymbol{x}^\star)$ is **indefinite**.

Class Workshop – 2nd Order Conditions of Multivariate Optimality

For the function

$$f(x) = \frac{3}{2}x_1^2 - x_2^2 + x_1 x_2$$

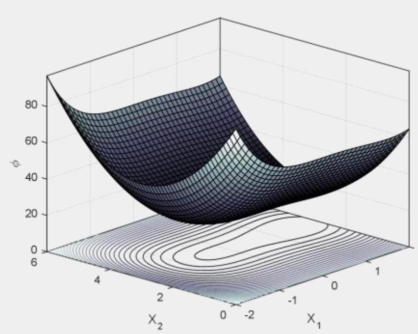
Characterize $x = (0,0)$ as a potential local minimum or maximum (or saddle point!).



For the candidate optima (stationary points) of the following unconstrained optimization program:

$$\min_{\boldsymbol{x}} \phi = 20 + x_1^3(x_1 - 2) + 5(x_2 - 3)^2$$

Determine if they have potential to be a local minimum, maximum, or saddle point.



Workshop Solution – 2nd Order Conditions of Multivariate Optimality

Last but **certainly** not least, we can also derive the **2nd order sufficient conditions of optimality** for multivariate objective functions as below. It follows a similar derivation to the univariate case, but we are (again) going to forego the detailed derivation for the sake of simplicity and focus instead on the result.

2nd-Order Sufficient Conditions of Multivariate Optimality

Consider a function $\phi(x^\star)$ with $x^\star$ a stationary point. If the function is twice differentiable ($\mathcal{C}^2$):

$$\nabla\phi(x^\star) = 0, \qquad H(x^\star) \succ 0 \;\; \Rightarrow \;\; \boldsymbol{x}^\star \text{ strict local minimum}$$

$$\nabla\phi(x^\star) = 0, \qquad H(x^\star) \prec 0 \;\; \Rightarrow \;\; \boldsymbol{x}^\star \text{ strict local maximum}$$

$$\nabla\phi(x^\star) = 0, \qquad H(x^\star)\{\npreceq \nsucceq\}\, 0 \;\; \Rightarrow \;\; \boldsymbol{x}^\star \text{ saddle point}$$

We are now able to identify whether or not a given stationary point is in fact a strict local maximum, a strict local minimum, or even a saddle point! Note that semi-definiteness still does not tell us anything *sufficient* (that is, no conclusions about a stationary point may be drawn from semidefinite Hessians).

Class Workshop – Applying 2nd Order Multivariate Sufficient Conditions

Determine if the stationary points in the previous two examples correspond to (strict) local minima, maxima, or saddle points using the 2nd order sufficient conditions of optimality.

Workshop Solution – Applying 2nd Order Multivariate Sufficient Conditions

Convexity

As it turns out, we can actually use some of the same properties about first- and second-order optimality conditions in our original discussion of convexity. As we now know, **nonlinear programs may in fact be convex**, and if they are it gives us a whole BUNCH of useful properties when solving. Let's revisit our discussion of convexity and see if we can't do a better job characterizing it.

Review: Convex Functions

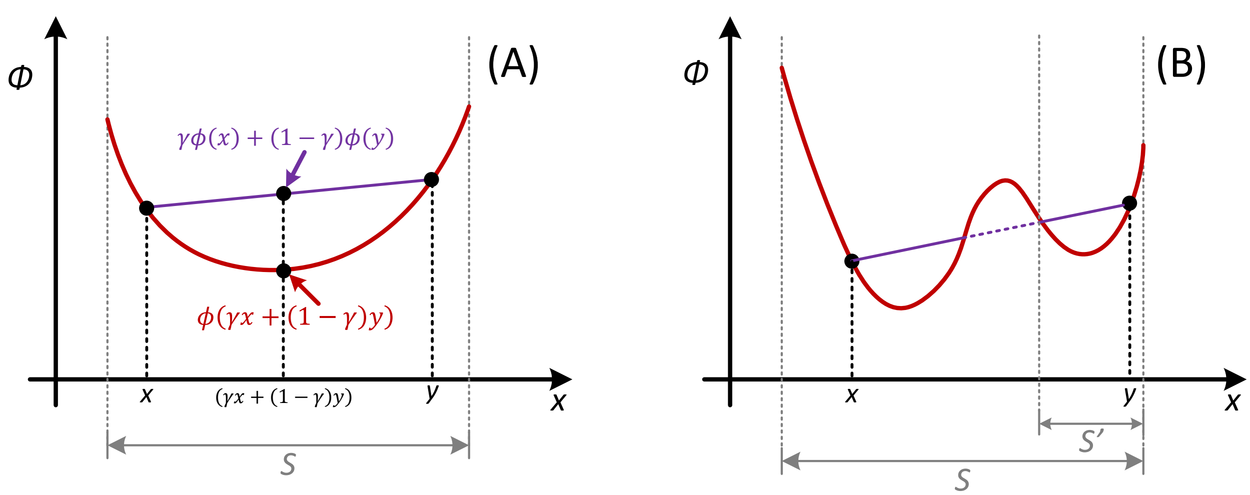
Consider our definition of a convex function from Module 02 along with the figure accompanying it. Is there some way to **connect the optimality conditions with our definition of convexity**?

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 $f(x)$ and $f(y)$ for ANY two points $(x, y) \in \mathcal{S}$ lies **above** the function between the points x and y:

$$\phi(\gamma x + (1-\gamma)y) \le \gamma\phi(x) + (1-\gamma)\phi(y), \ \ \forall\, \gamma \in (0,1)$$

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



Connecting Convexity to Global Optimality

We know from our linear programming section that *any local optimum* to a convex program is in fact a *global optimum*. Extending this to nonlinear programs only, we have:

Sufficient Condition for Global Optimality

A ***local*** (or strict local) optimum $x^\star$ of a convex **unconstrained** nonlinear program:

$$\min_{x \in \mathbb{R}^n} \phi = f(x)$$

is also a ***global*** (or strict global) optimum.

It is worth noting here that this concept falls apart rather rapidly when constraints are introduced into the scenario. Of course, a convex constrained set combined with a convex objective function is still OK, but things get out of control when we have non-convex constraints, or *even convex constraint sets with a non-convex objective function*. Yikes!

Relating the Second-Order Optimality Conditions to Convexity

OK, let's think for a second. **We KNOW** that in order to find a local minimum for some stationary point $x^\star$, we require the Hessian $H(x^\star) \succcurlyeq 0$. However, what if we were to extend that idea to **all points $\boldsymbol{x}^\circ \in \mathcal{S}$ for which the optimization program is defined?** Well, in effect we would be able to algebraically conclude the *exact same thing* as:

$$\phi(\gamma x + (1-\gamma)y) \leq \gamma\phi(x) + (1-\gamma)\phi(y), \quad \forall\, \gamma \in (0,1)$$

In other words, I can extend the definition of a local minimum to **also correspond to a convex function**. All that needs to be changed is that I have to **claim that the Hessian is not only positive (semi)definite for a stationary point, but for ALL points in my set $\mathcal{S}$!**

Gradient and Hessian Tests for Convexity

Consider a function $\phi: \mathcal{S} \subset \mathbb{R}^n \to \mathbb{R}$ in $\mathcal{C}^2$. The function ϕ is convex on $\mathcal{S}$ if at each $x^o \in \mathcal{S}$:

- **Gradient Test**: $\phi(\boldsymbol{x}) \geq \phi(\boldsymbol{x}^o) + \nabla\phi(\boldsymbol{x}^o)^T(\boldsymbol{x} - \boldsymbol{x}^o) \ \forall\, \boldsymbol{x} \in \mathcal{S}$
- **Hessian Test**: $H(\boldsymbol{x}^o) \succcurlyeq 0$ (positive semi-definite)

NB – STRICT convexity is determined by making the inequalities strict.

Why are these related? Well it turns out that they are grounded in the same mathematical theory and are closely related in the definitions of the Karsh-Karun-Tucker conditions for optimality (we will not go there). Either way we now know how to prove whether or not any (unconstrained) nonlinear program is convex!

Class Workshop – Determining Convexity

Determine if the following unconstrained objective functions are convex on $\mathcal{S}$.

- $\phi = x_1^2 + x_1x_2 + 3x_2 + 4$ with $\mathcal{S} = \mathbb{R}^2$
- $\phi = (x_1+4)^4 + x_1x_2 + (x_2+1)^4$ with $\mathcal{S} \subset \mathbb{R}^2: (x_1, x_2) \geq 0$

Workshop Solution – Determining Convexity

In retrospect I should have shown this to you before assignment 1…

Conclusions

OK! That was a lot of math. Well, maybe not a lot. It was some math. BUT the real question is, **why do you care**? In our last couple of classes we will be looking at how to solve unconstrained and constrained nonlinear programs, the algorithms for which depend heavily on using Taylor Series expansions and quadratic approximations of search directions (known as a line search) in order to locate the optimum. We will also focus on derivative-free methods, which use only function evaluations but still rely heavily on the theory presented in this section.

Moreover, we can now assess (not necessarily by hand, but in a generalized computer algorithm such as the solvers used by GAMS) whether or not we have found a potential optimum, and avoid saddle points. We can also determine of a function is convex over the set on which it is defined, helping us conclude **global optimality** rather than just local. A few of you are already running into local optimality problems in your projects, so I am sure you can agree with just how important convexity is for nonlinear programs.

~~ END OF MODULE ~~

\|T|/