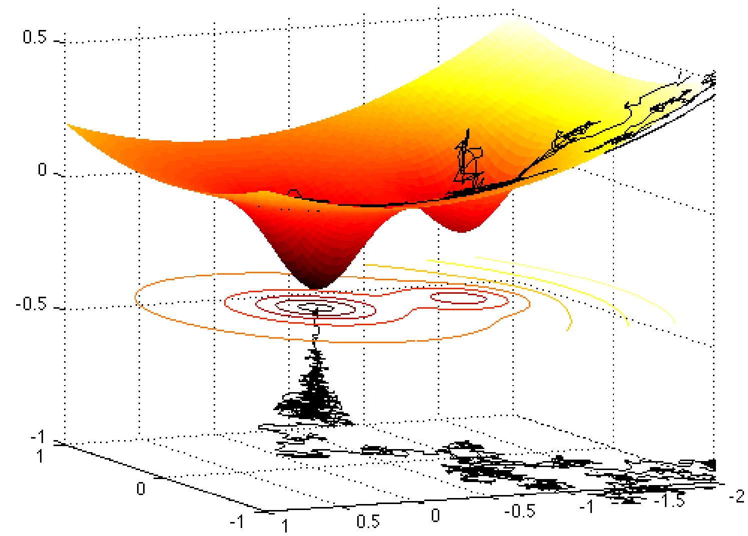
Florida International University
Department of Civil and Environmental Engineering
Optimization in Water Resources Engineering, Spring 2020

# LECTURE: CLASSICAL OPTIMIZATION OVERVIEW




**Arturo S. Leon, Ph.D., P.E., D.WRE**

# Optimization problem

- Design variables: variables with which the design problem is parameterized:

$$x = (x_1, x_2, \dots x_n)$$

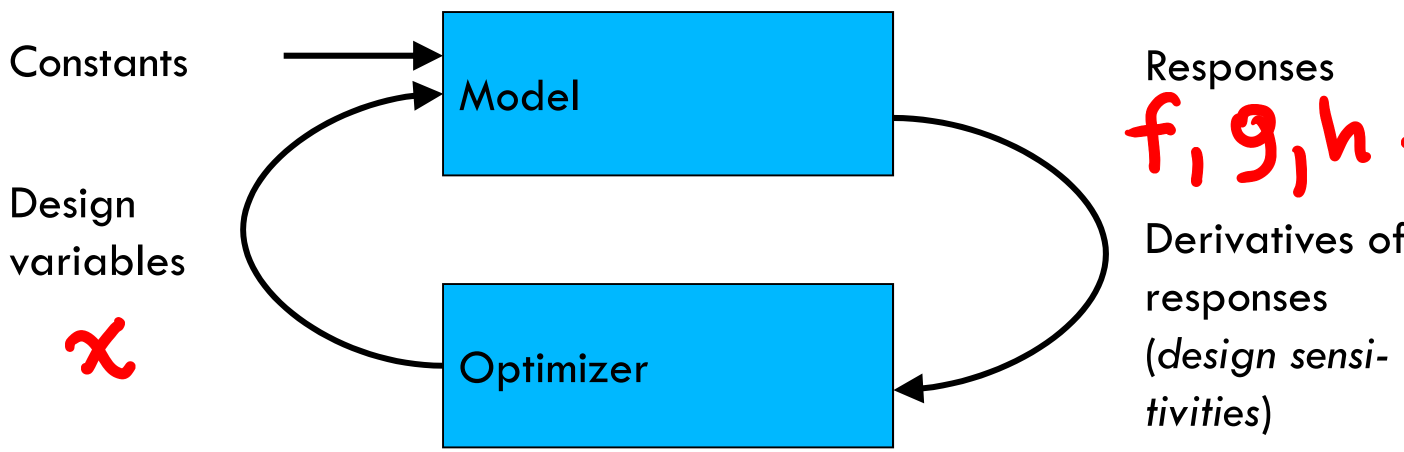
- Objective: quantity that is to be minimized (maximized)
  Usually denoted by:
  ( "cost function")

$$f(x)$$

- Constraint: condition that has to be satisfied
  - Inequality constraint: $g(x) \leq 0$
  - Equality constraint: $h(x) = 0$

# Solving optimization problems

- Optimization problems are typically solved using an iterative algorithm:




$x$

$f, g, h \ldots$

$\frac{\partial f}{\partial x}, \frac{\partial g}{\partial x}, \frac{\partial h}{\partial x} \ldots$

# Defining an optimization problem

1. Choose **design variables** and their bounds
2. Formulate **objective**
3. Formulate **constraints** (restrictions)
4. Choose suitable **optimization algorithm**

# *Example – Design of a SODA Can*

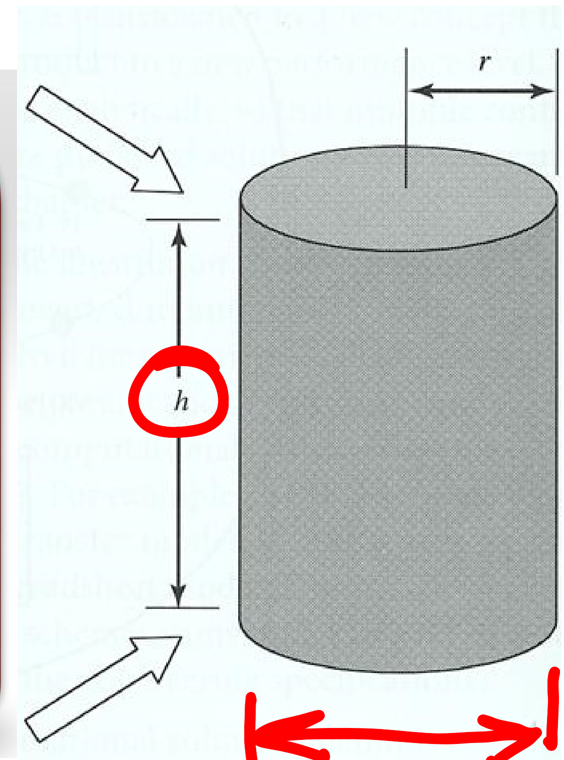
- Design a SODA can to hold an specified amount of SODA and other requirements.
- The cans will be produced in billions, so it is desirable to minimize the cost of manufacturing.
- Since the cost is related directly to the surface area of the sheet metal used, it is reasonable to minimize the sheet metal required to fabricate the can.

FANTA & COCA COLA CAN ICONS

## *Example – Design of a SODA Can (Cont.)*

**Requirements:**

1. The diameter of the can should be no more than 8 cm and no less than 3.5 cm.
2. The height of the can should be no more than18 cm and no less than 8 cm.
3. The can is required to hold at least 400 ml of fluid.


D

$3.5\text{ cm} \leq D \leq 8\text{ cm}$

$8\text{ cm} \leq h \leq 18\text{ cm}$

$\text{Vol} \geq 400\text{ ml}$

# *Example – Design of a SODA Can (Cont.)*

**Design variables**

$D$ = diameter of the can (cm)

$H$ = height of the can (cm)

**Objective function** $f(D, H)$

The design objective is to minimize the surface area

$$f = \pi D H + \frac{\pi D^2}{2}$$

## *Example – Design of a SODA Can (Cont.)*

The constraints must be formulated in terms of design variables.

The first constraint is that the can must hold at least 400 ml of fluid.

$$\frac{\pi D^2}{4} H \geq 400 \text{ ml}$$

The other constraints on the size of the can are:

$$3.5 \leq D \leq 8 \quad , \quad 8 \leq H \leq 18$$

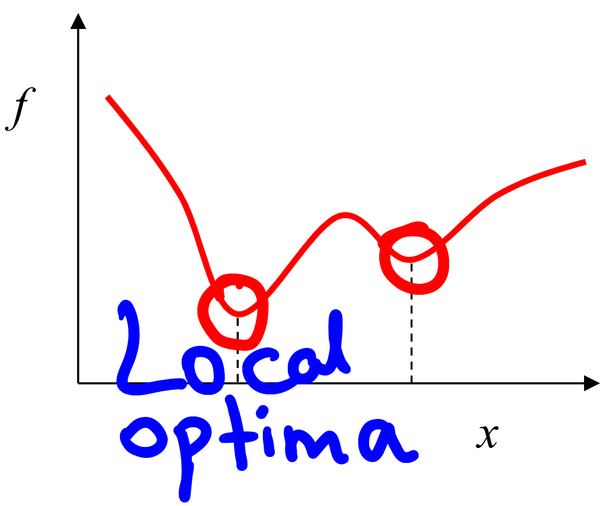
The problem has two independent design variables and five explicit constraints.

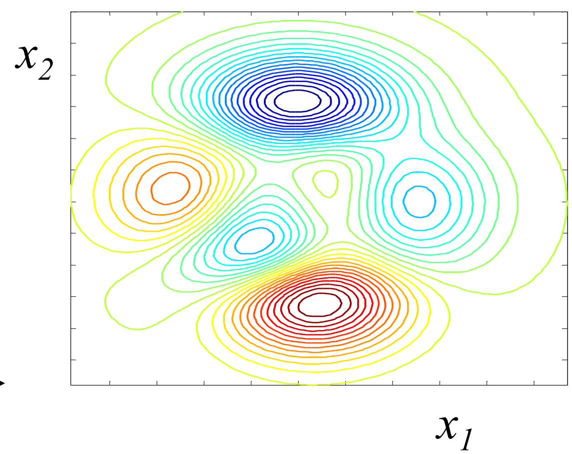
$$H = 8 \text{ cm}$$

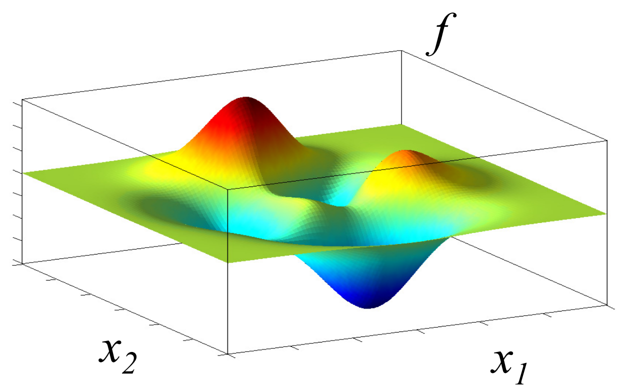
$$D = 8 \text{ cm}$$

# Optimization Problem Characteristics
## Linearity

- Nonlinear objective functions can have multiple *local optima*:



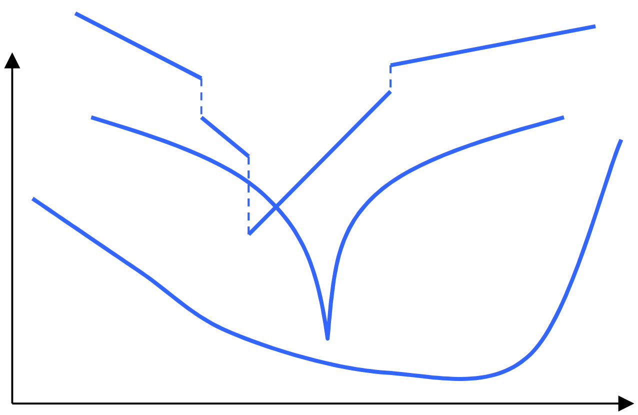



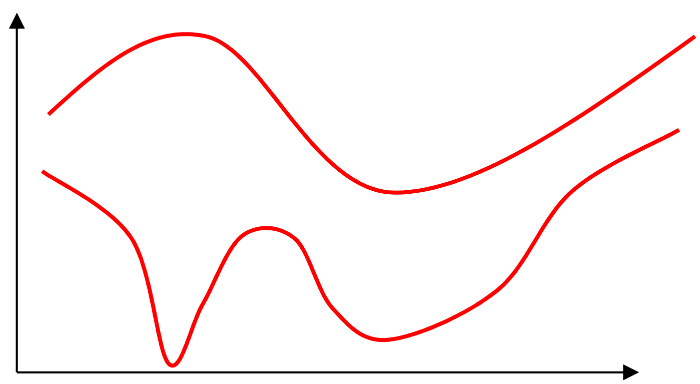

- Challenge: finding the *global optimum*.

# Unimodality

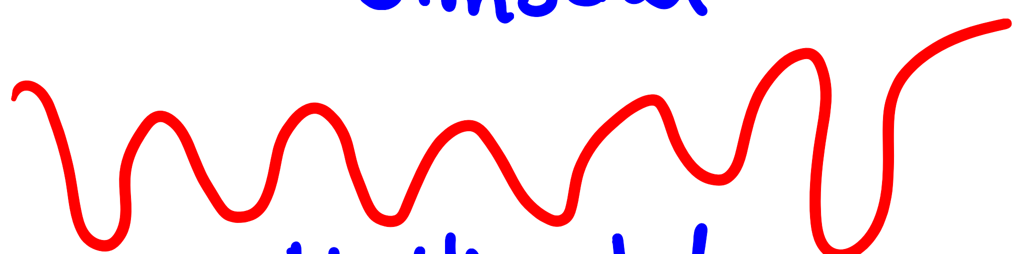
- Bracketing and sectioning methods work best for *unimodal* functions:
  *"An unimodal function consists of exactly one monotonically increasing and decreasing part"*



unimodal



bimodal

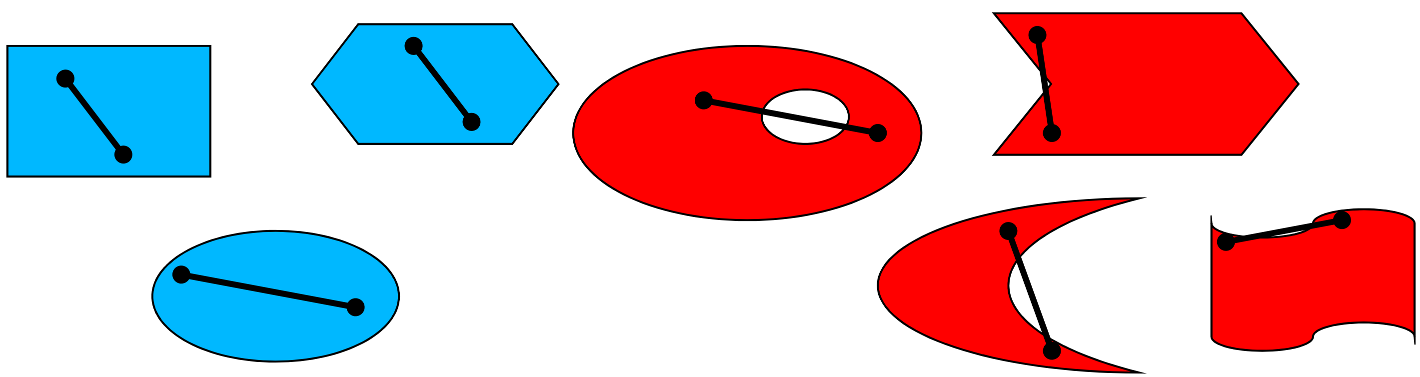


Multimodal

# Convexity

- Convex set:

  *"A set $S$ is convex if for every two points $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ in $S$, the connecting line also lies completely inside $S$"*



Convex

Non-convex

# Lagrange Multipliers

- The method of Lagrange multipliers gives a set of necessary conditions to identify optimal points of equality constrained optimization problems.
- This is done by converting a constrained problem to an equivalent unconstrained problem with the help of certain unspecified parameters known as Lagrange multipliers.

# Finding an Optimum using Lagrange Multipliers

- The classical problem formulation

  minimize $f(x_1, x_2, \ldots, x_n)$

  Subject to $h_1(x_1, x_2, \ldots, x_n) = 0$

  can be converted to

  minimize $\mathbf{L(x, \lambda) = f(x) - \lambda\, h_1(x)}$

where

$L(x, \lambda)$ is the Lagrangian function

$\lambda$ is an unspecified positive or negative constant called the *Lagrangian Multiplier*

# Lagrange Multipliers Method

1. Original problem is rewritten as:

   **minimize $L(x, \lambda) = f(x) - \lambda\, h_1(x)$**

2. Take derivatives of $L(x, \lambda)$ with respect to $x_i$ and set them equal to zero.
   - If there are n variables (i.e., $x_1, \ldots, x_n$) then you will get n equations with n + 1 unknowns (i.e., n variables $x_i$ and one Lagrangian multiplier $\lambda$)
3. Express all $x_i$ in terms of *Langrangian multiplier $\lambda$*
4. Plug x in terms of $\lambda$ in constraint $h_1(x) = 0$ and solve $\lambda$.
5. Calculate x by using the just found value for $\lambda$.

- Note that the n derivatives and one constraint equation result in n+1 equations for n+1 variables!

# Multiple constraints

- The *Lagrangian multiplier* method can be used for any number of equality constraints.
- Suppose we have a classical problem formulation with *k* equality constraints

minimize $f(x_1, x_2, ..., x_n)$

Subject to $h_1(x_1, x_2, ..., x_n) = 0$

……

$h_k(x_1, x_2, ..., x_n) = 0$

This can be converted in

minimize $L(x, \lambda) = f(x) - \lambda^T h(x)$

Where $\lambda^T$ is the transpose vector of Lagrangian multpliers and has length k

# EXAMPLE

A factory manufactures HONDA CITY and HONDA CIVIC cars. Determine the optimal number of HONDA CITY and HONDA CIVIC cars produced if the factory capacity is 90 cars per day, and the cost of manufacturing is $C(x, y) = 6x^2 + 12y^2$, where $x$ is the number of HONDA CITY cars and $y$ is the number of HONDA CIVIC cars produced.


# EXAMPLE (Cont.)

- VARIABLES

  $x$ = No. of HONDA CITY cars produced

  $y$ = No. of HONDA CIVIC cars produced

- COST of Manufacturing;

  $C(x, y) = 6x^2 + 12y^2$

- OBJECTIVE:

  **MINIMIZE COST** = Minimize $f = 6x^2 + 12y^2$

- CONSTRAINT: 90 cars per day

  $x + y = 90$ $\quad h = x + y - 90 = 0$

- Original problem is rewritten as:

  **minimize** $\mathbf{L(x, \lambda) = f(x) - \lambda\, h_1(x)}$

Minimize $L(x, \lambda) = 6x^2 + 12y^2 - \lambda(x + y - 90)$

# EXAMPLE (Cont.)

$$\frac{\partial L}{\partial x} = 12x - \lambda = 0 \longrightarrow x = \lambda/12$$

$$\frac{\partial L}{\partial y} = 24y - \lambda = 0 \longrightarrow y = \lambda/24$$

Also: $x + y - 90 = 0$

$$\frac{\lambda}{12} + \frac{\lambda}{24} - 90 = 0$$

$$\frac{3\lambda}{24} = 90 \longrightarrow \lambda = 720$$

$$\therefore \quad x = \frac{720}{12} = 60$$

$$y = \frac{720}{24} = 30$$

# Unconstrained optimization algorithms

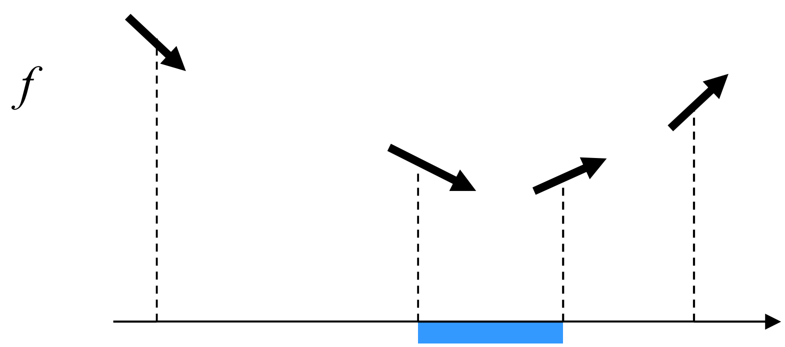
- Single-variable methods
  - $0^{th}$ order (involving only $f$) $f$
  - $1^{st}$ order (involving $f$ and $f'$) $f, f'$ (first derivative)
  - $2^{nd}$ order (involving $f$, $f'$ and $f''$) $f, f', f''$
- Multiple variable methods
  - Gradient Descent Methods
  - Simplex Method
  - Sequential Linear Programming
  - Sequential Quadratic Programming
  - Etc.

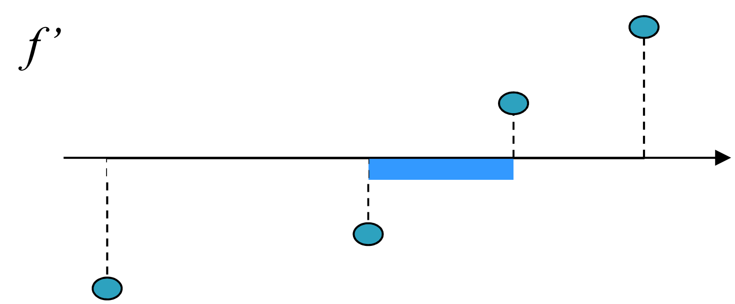
# Single-variable methods
# Bisection method

- Optimality conditions: minimum at stationary point

  $\Rightarrow$ Root finding of $f'$

- Similar to sectioning methods, but uses derivative:




Interval is halved in each iteration. Note, this is better than any of the direct methods

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!}f''(x) + \frac{h^3}{3!}f'''(x) + \dots \frac{h^n}{n!}f^{n}(x)$$

# Newton's method

- Again, root finding of $f'$
- Basis: Taylor approximation of $f'$:

$$f'(x+h) = f'(x) + f''(x)h + \cancel{o(h^2)}$$

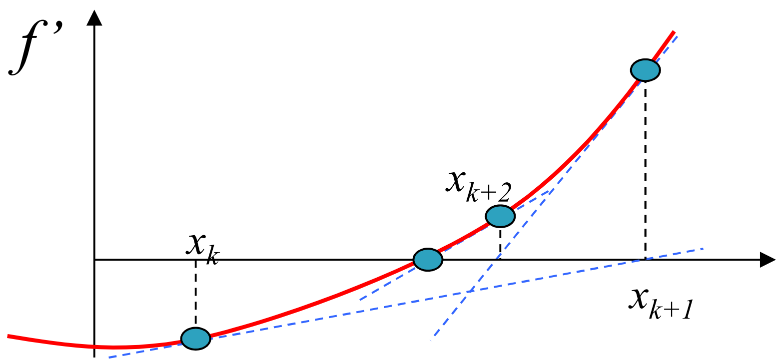
Linear approximation

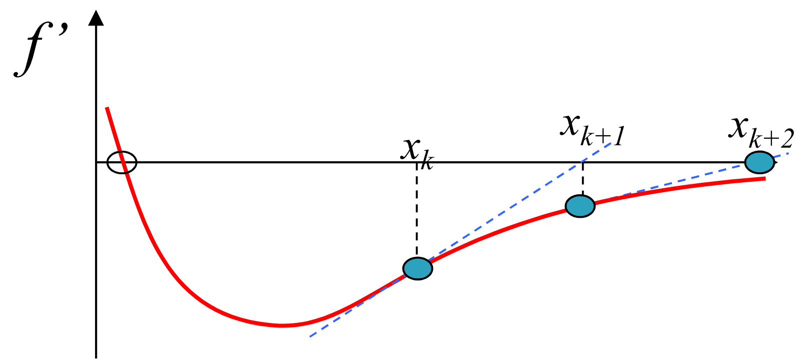
$$\Rightarrow h = -\frac{f'(x)}{f''(x)}$$

- New guess: $x_{k+1} = x_k + h_k = x_k - \frac{f'(x_k)}{f''(x_k)}$

# Newton's method (cont.)

- Best convergence of all methods:




- Unless it diverges

# Summary single variable methods

- Bracketing +

| | |
| --- | --- |
| Dichotomous sectioning | $0^{th}$ order |
| Fibonacci sectioning | |
| Golden ratio sectioning | |
| Quadratic interpolation | |
| Cubic interpolation | $1^{st}$ order |
| Bisection method | |
| Secant method | |
| Newton method | $2^{nd}$ order |

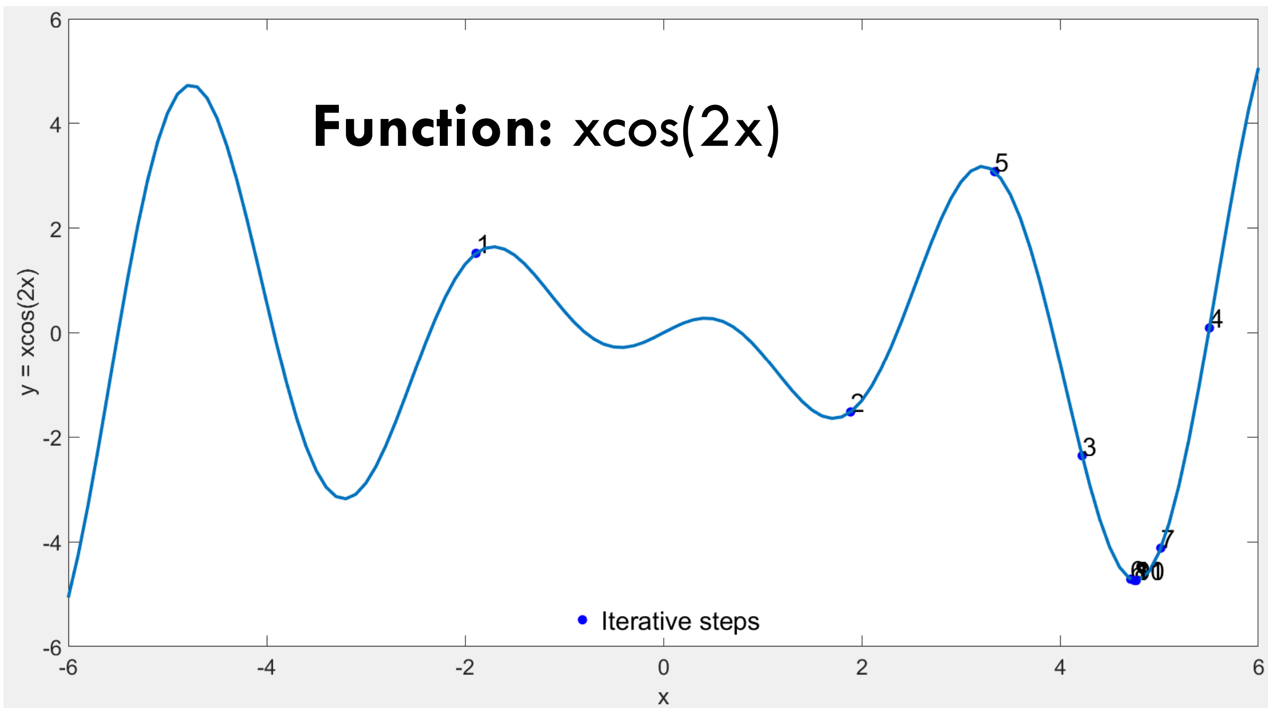
- And many, many more!

# MATLAB DEMO: Single variable Minimization

This demo will show a number of ways to minimize f(x) starting at multiple initial points.

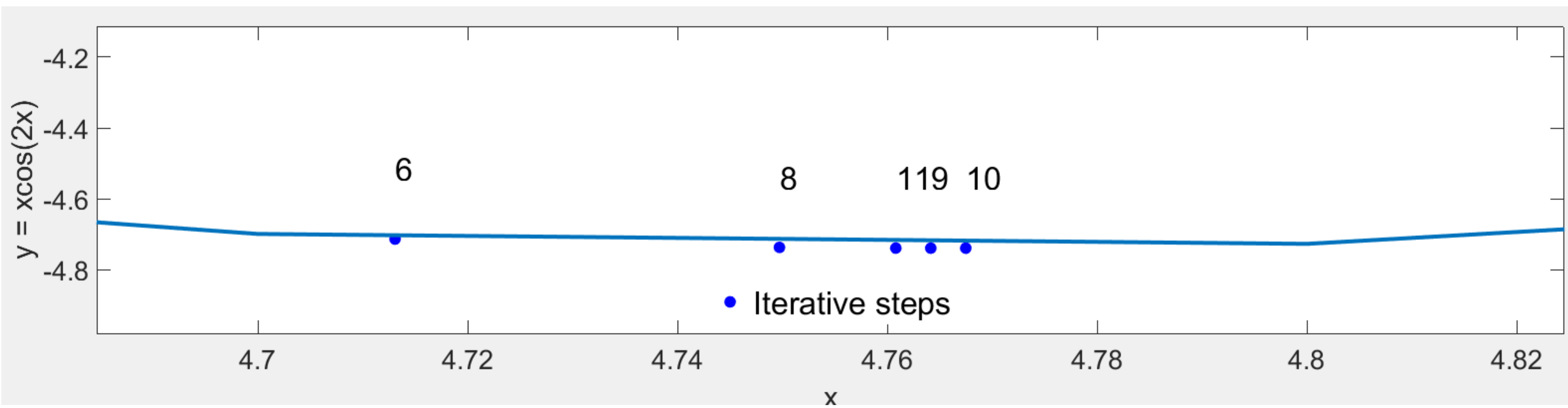
**Demo Folder:** Single_variable_Classical_Optimization (Download file from **Canvas**)

**Demo File:** Main_File_Single_Variable.m



# Single variable Minimization (cont.)

(1) Change starting points
(2) Discuss and show sensitivity of solutions



# Multiple variable methods
# GRADIENT DESCENT METHODS

- Consider a function $J(x),\; x=\left[x_1, x_2, \ldots . x_n\right]$

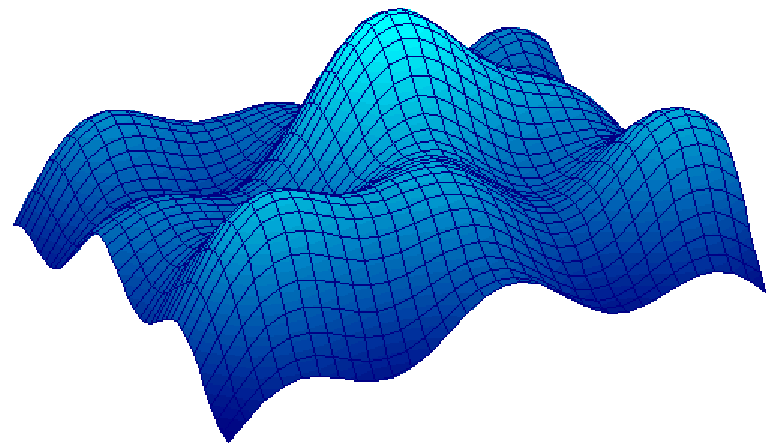
- The gradient of *J(x)* at a point $x^0$ is a vector of length *n*.

$$\nabla J(x^0)=\begin{bmatrix} \frac{\partial J}{\partial x_1}(x^0) \\ \frac{\partial J}{\partial x_2}(x^0) \\ . \\ . \\ . \\ \frac{\partial J}{\partial x_n}(x^0) \end{bmatrix}$$

- Each element in the vector is evaluated at the point $x^0$.

# GRADIENT DESCENT METHODS (cont.)

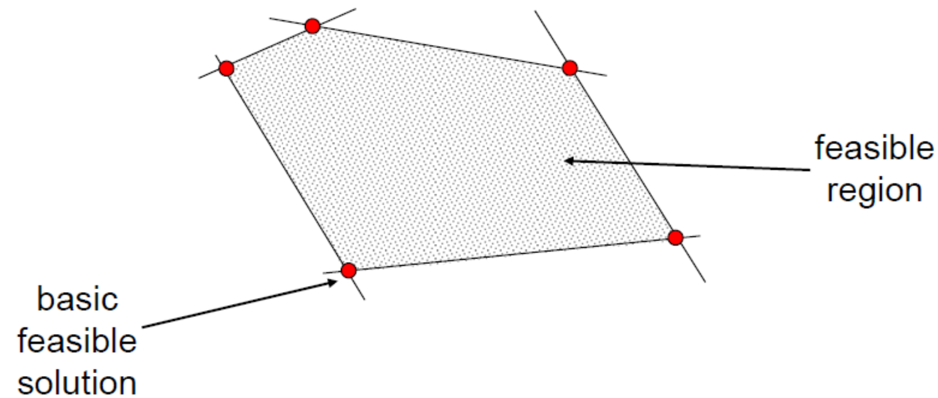
- Linear Programming
- Simplex Method
- Newton-Raphson Method
- Secant Method
- Bisection Method
- Line Search Methods
- Sequential Linear Programming
- Sequential Quadratic Programming
- Karush-Kuhn-Tucker Conditions (KKT)



# SIMPLEX METHOD

- Solutions at the "vertices" of the design space are called **basic feasible solutions.**

- The Simplex algorithm moves from BFS to BFS so that the objective always improves.



# SEQUENTIAL LINEAR PROGRAMMING

- Consider a general nonlinear problem linearized via first order Taylor series:

$$\begin{aligned} \min \quad & J(\mathbf{x}) \approx J(\mathbf{x}^0) + \nabla J(\mathbf{x}^0)^T \delta\mathbf{x} \\ \text{s.t.} \quad & g_j(\mathbf{x}) \approx g_j(\mathbf{x}^0) + \nabla g_j(\mathbf{x}^0)^T \delta\mathbf{x} \le 0 \\ & h_k(\mathbf{x}) \approx h_k(\mathbf{x}^0) + \nabla h_k(\mathbf{x}^0)^T \delta\mathbf{x} = 0 \\ & x_i^\ell \le x_i + \delta x_i \le x_i^u \end{aligned}$$

$$\text{where} \qquad \delta\mathbf{x} = \mathbf{x} - \mathbf{x}^0$$

- This is an LP problem with the design variables contained in $\delta x$. The functions and gradients evaluated at $x^0$ are constant coefficients.

# SEQUENTIAL LINEAR PROGRAMMING (Cont.)

1. Initial guess $x^0$
2. Linearize about $x^0$ using first order Taylor series
3. Solve resulting LP to find $\delta x$
4. Update $x^1 = x^0 + \delta x$
5. Linearize about $x^1$ and repeat:

$$x^q = x^{q-1} + \delta x$$

Where $\delta x$ is the solution of LP (model linearized about $x^{q-1}$.

# SEQUENTIAL QUADRATIC PROGRAMMING

- Create a quadratic approximation to the Lagrangian
- Create linear approximations to the constraints
- Solve the quadratic problem to find the search direction, S
- Perform the 1-D search
- Update the approximation to the Lagrangian

# Newton method

Expand $f(\mathbf{x})$ by its Taylor series about the point $\mathbf{x}_k$

$$f(\mathbf{x}_k + \delta\mathbf{x}) \approx f(\mathbf{x}_k) + \mathbf{g}_k^T \delta\mathbf{x} + \frac{1}{2}\delta\mathbf{x}^T \mathbf{H}_k \delta\mathbf{x}$$

where the gradient is the vector

$$\mathbf{g}_k = \nabla f(\mathbf{x}_k) = \left[\frac{\partial f}{x_1} \dots \frac{\partial f}{x_N}\right]^T$$

Gradient

and the Hessian is the symmetric matrix

$$\mathbf{H}_k = \mathbf{H}(\mathbf{x}_k) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_N \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_N^2} \end{bmatrix}$$

Hessian

# Newton method (Cont.)

For a minimum we require that $\nabla f(\mathbf{x}) = \mathbf{0}$, and so

$$\nabla f(\mathbf{x}) = \mathbf{g}_k + \mathbf{H}_k \delta\mathbf{x} = \mathbf{0}$$

with solution $\delta\mathbf{x} = -\mathbf{H}_k^{-1}\mathbf{g}_k$. This gives the iterative update

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}_k^{-1}\mathbf{g}_k$$

- If $f(\mathbf{x})$ is quadratic, then the solution is found in one step.
- The method has quadratic convergence (as in the 1D case).
- The solution $\delta\mathbf{x} = -\mathbf{H}_k^{-1}\mathbf{g}_k$ is guaranteed to be a downhill direction.
- Rather than jump straight to the minimum, it is better to perform a line minimization which ensures global convergence

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{H}_k^{-1}\mathbf{g}_k$$

# Summary of MATLAB Multiple Variable Methods

- **Fminsearch:** Find minimum of unconstrained multivariable function using derivative-free method
- **Fminunc:** Nonlinear programming solver. Finds minimum of unconstrained multivariable function. Gradient and Hessian may be supplied.
- **Lsqnonlin**: Solves nonlinear least-squares curve fitting problems of the form

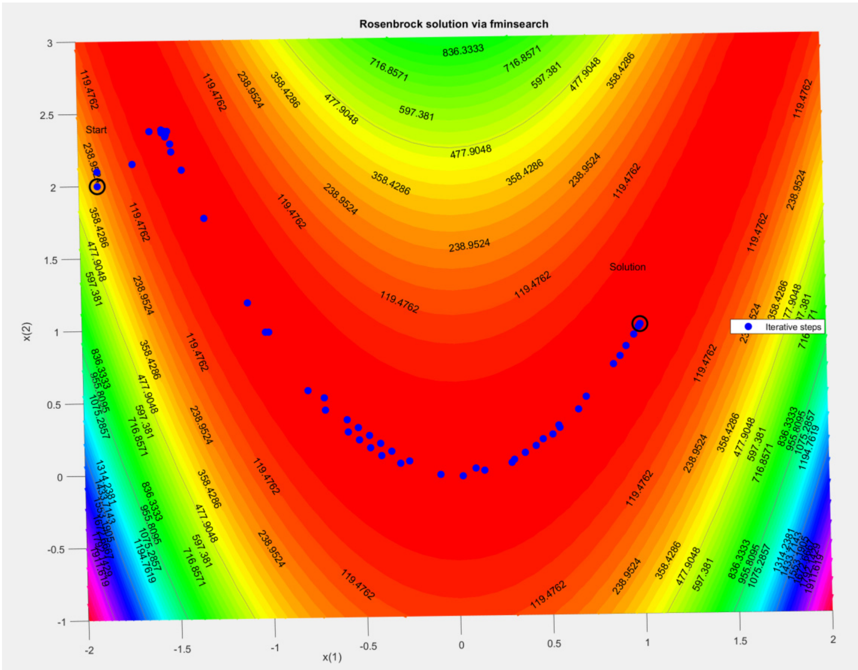
$$\min_x \| f(x) \|_2^2 = \min_x \left( f_1(x)^2 + f_2(x)^2 + \ldots + f_n(x)^2 \right)$$

# MATLAB DEMO: Banana Function Minimization

Minimize Rosenbrock's "banana function"

$$f(x) = 100(x(2) - x(1)^2)^2 + (1 - x(1))^2$$

- f(x) is called the banana function because of its curvature around the origin.
- It is notorious in optimization examples because of the slow convergence most methods exhibit when trying to solve this problem
- f(x) has a unique minimum at the point x = [1, 1] where f(x) = 0



# Banana Function Minimization (cont.)

This demo will show a number of ways to minimize *f*(*x*) starting at multiple initial points.

**Demo Folder:** BananaFunction_Classical_Optimization (Download file from **Canvas**)