# Constrained gradient-based optimization



ME 564/SYS 564
Wed Oct 10, 2018
Steven Hoffenson

$\nabla f$

<u>Goal of Week 7</u>: To learn the optimality conditions for constrained problems, be able to solve problems with them, and understand how some common algorithms work

# Recap: How to optimize

1. **Formulate** the problem

   (Weeks 1-3, 9-12)

   a) Define system boundaries
   b) Develop analytical models
   c) Explore/reduce the problem space
   d) Formalize optimization problem

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f(\mathbf{x}, \mathbf{p}) \\ \text{subject to} \quad & \mathbf{g}(\mathbf{x}, \mathbf{p}) \leq 0 \\ & \mathbf{h}(\mathbf{x}, \mathbf{p}) = 0 \end{aligned}$$



TODAY

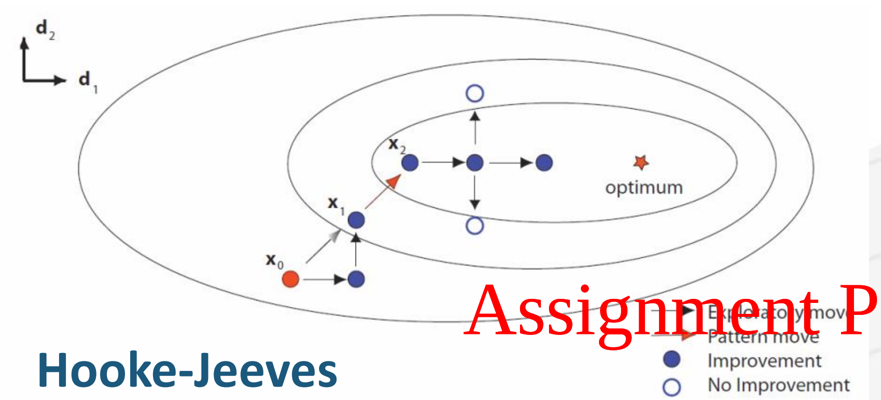
2. **Solve** the problem

   (Weeks 4-8, 13)

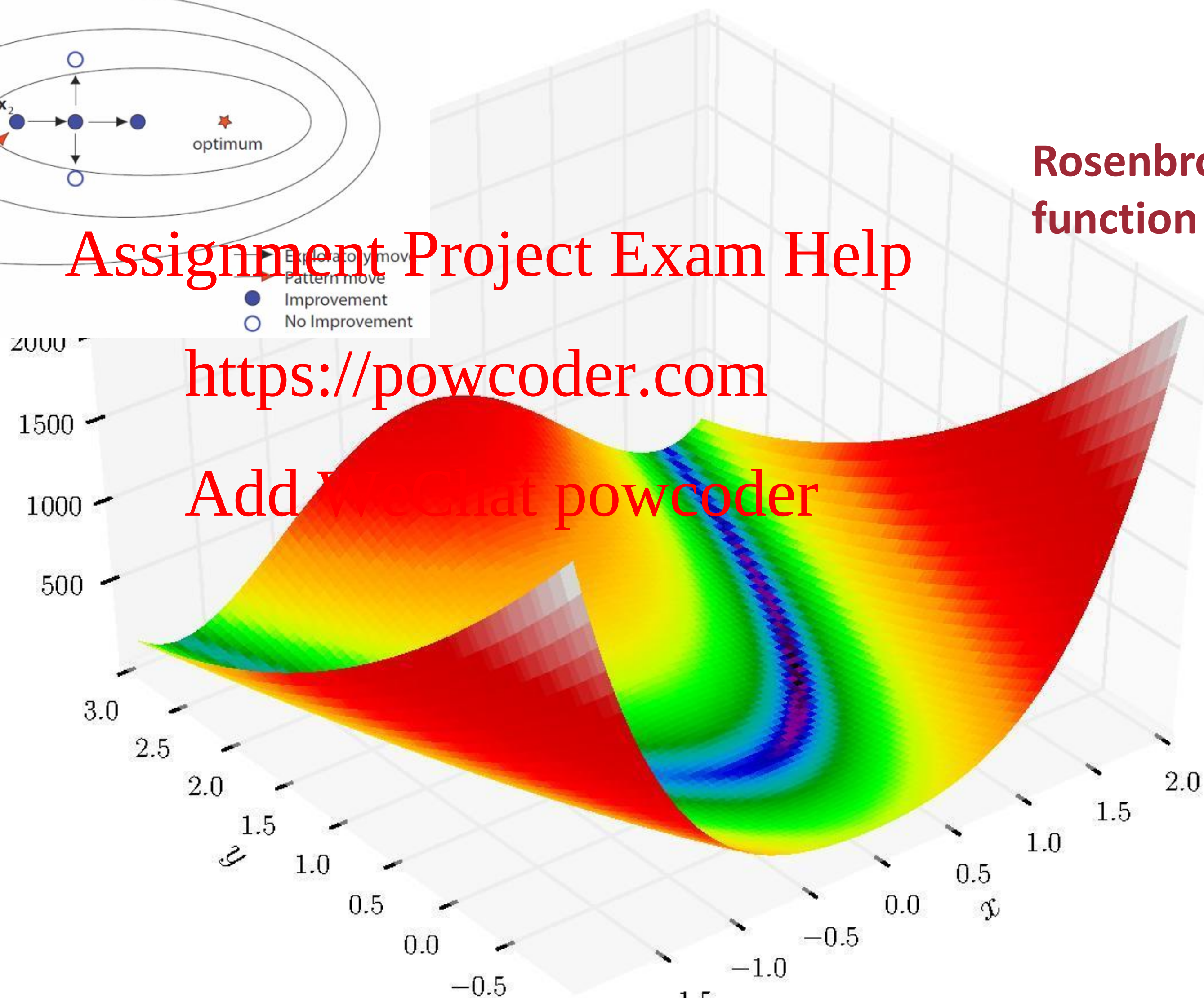
   a) Choose the right approach/algorithm
   b) Solve (by hand, code, or software)
   c) Interpret the results
   d) Iterate if needed

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\mathbf{H}(\mathbf{x}_k)]^{-1} \nabla f(\mathbf{x}_0)$$

# Discuss: HW2 P3




**Hooke-Jeeves algorithm**




**Rosenbrock function**

# Recap: Weeks 5 & 6 (and HW3)

- The **optimality conditions** can be used to solve for or prove an interior optimum
  - The **First-Order Necessary Condition** identifies stationary points
  - The **Second-Order Sufficiency Condition** identifies the nature (minima, maxima, saddle) of stationary points
- **Taylor series** approximation is used to generate derivative-based local optimization directions
  - The **gradient descent** algorithm uses 1st-order info
  - **Newton's method** (algorithm) uses 2nd-order info
- **Convexity** can be used to prove a local optimum is global

# Recap: Gradient descent algorithm

*Local optimization algorithm for interior optima*

1. Begin with a feasible point $\mathbf{x}_0$
2. Find the gradient at that point $\nabla f(\mathbf{x}_0)$
3. Move in the direction of the negative gradient to find an improved $\mathbf{x}_1$

$$\mathbf{x}_1 = \mathbf{x}_0 - \nabla f(\mathbf{x}_0)$$

4. Continue to iterate until you stop improving

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \nabla f(\mathbf{x}_{k-1})$$

*We can also add a scale factor α.*

# Recap: Newton's method

*Local optimization algorithm for interior optima*

1. Begin with a feasible point $\mathbf{x}_0$
2. Find the gradient and Hessian that point
3. Move in the following way:



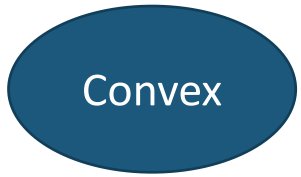
$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\mathbf{H}(\mathbf{x}_k)]^{-1} \nabla f(\mathbf{x}_k)$$

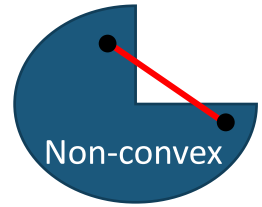


*Similar to gradient descent, but multiply the Hessian inverse by gradient. We can also add a scale factor α.*

**Note: This is very effective for quadratic objectives.**

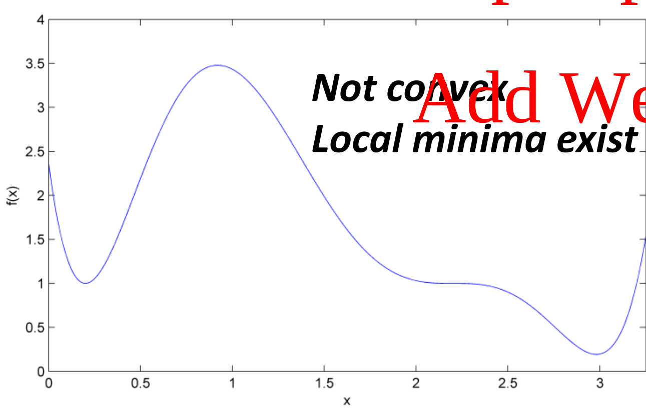
# Recap: Convexity

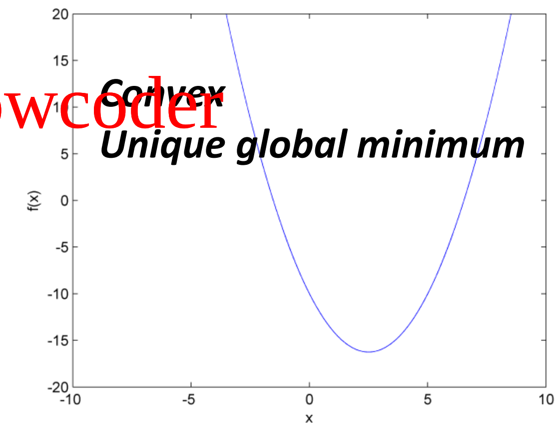



*If you can prove convexity, then any local optimum is a global optimum!*






If the Hessian of the objective function is positive definite **everywhere**, then the problem is convex! This can help you conclude that you have found a **global** solution.

# HW3 tips

- Some versions of MATLAB do not allow functions in the same file as the code. If my "week6_ex419gd_2018.m" does not run on your computer, you will need to separate this into 3 files (main script + 2 functions)! This is similar to the fmincon beam example. Ask me if you need help with this.



- Watch out for matrix dimensions. These algorithms are written with the x's and gradients as column vectors, but with the x's as a row vector in our code examples, this can cause MATLAB problems. You cannot add a row vector and a column vector, so you need to be consistent or use transposes.



- **Start early**, and make an appointment with me if you have questions. I want everyone to get an A on this HW.

# Constrained gradient-based optimization



subject to $\mathbf{g}(\mathbf{x}, \mathbf{p}) \leq 0$

$\mathbf{h}(\mathbf{x}, \mathbf{p}) = 0$

$$\nabla f$$

# Today's topics

- Reduced gradient approach
  - Extensions to FONC and SOSC
  - Generalized Reduced Gradient (GRG) algorithm
  - Active set strategies
- Lagrangian approach
  - Lagrange multipliers
  - Karush-Kuhn-Tucker (KKT) conditions
- Two common algorithms:
  - Quasi-Newton methods
  - Sequential Quadratic Programming (SQP)

# Reduced gradient



An approach for using updated FONC and SOSC to solve problems with equality (or active inequality) constraints that cannot be substituted out numerically

Alone, Reduced Gradient is not an algorithm! However, many common algorithms are built on its premise.

# Reduced gradient: Rationale

- In the past, with equality constraints or *active* inequality constraints, we substituted variables out

$$\min f = x_1 + x_2$$
$$\text{s.t. } h = x_1 - 1 = 0$$
$$g = -x_2 + 1 \le 0$$

$$h:\ {x_1}^* = 1 \quad \text{equality}$$
$$g:\ {x_2}^* = 1 \quad \text{MP1}$$
$$f^* = 2$$



- What if you cannot substitute?

$$\min f = x_1 + x_2$$
$$\text{s.t. } h = x_1 - x_2 + \boxed{\cos x_1 x_2} = 0$$

Because of this, we cannot isolate either variable!

# Reduced gradient: Overview

1. Partition variables $\mathbf{x}$ into sets of $\mathbf{s}$ "state variables" and $\mathbf{d}$ "decision variables"
2. Take all partial derivatives:

$$\partial f/\partial \mathbf{s},\ \partial f/\partial \mathbf{d},\ \partial \mathbf{h}/\partial \mathbf{s},\ \partial \mathbf{h}/\partial \mathbf{d}$$

3. Calculate reduced gradient $\partial z/\partial \mathbf{d}$ using Step 2, and solve for $\mathbf{d}^*$ by setting $\partial z/\partial \mathbf{d} = \mathbf{0}$

4. Calculate $\mathbf{s}^*$ from optimal $\mathbf{d}^*$ using $\mathbf{h}$, then calculate $f^*$

# Reduced gradient: Partition

- Since we have $n$ variables/unknowns and $m$ equations, we have $n - m$ degrees of freedom
- We can partition the $n$-dimensional vector $\mathbf{x}$ into $m$ "state variables" and $n - m$ "decision variables"



I.e., $\mathbf{x} = [x_1, x_2, \dots x_n]^T$ with $m < n$ active constraints can be split into:





$$\mathbf{s} = [x_1, x_2, \dots x_m]^T$$ ← state variables

$$\mathbf{d} = [x_{m+1}, x_{m+2}, \dots x_n]^T$$ ← decision variables

- The choice of which variables are state and which are decision does not matter – you should choose which will be easiest computationally

# Reduced gradient: Re-frame $\partial \mathbf{h}$

- Separate $\mathbf{s}$ and $\mathbf{d}$ in framing of $\partial \mathbf{h}$:

$$\partial h_j = \sum_{i=1}^{m} \frac{\partial h_j}{\partial s_i} \partial s_i + \sum_{i=1}^{n-m} \frac{\partial h_j}{\partial d_i} \partial d_i = 0 \ , j = \{1, \ldots, m\}$$



- In vector form: $\partial \mathbf{h} = (\partial \mathbf{h}/\partial \mathbf{s})\partial \mathbf{s} + (\partial \mathbf{h}/\partial \mathbf{d})\partial \mathbf{d} = \mathbf{0}$



  where,

$$\partial \mathbf{h}/\partial \mathbf{s} = \begin{bmatrix} \partial h_1/\partial s_1 & \partial h_1/\partial s_2 & \ldots & \partial h_1/\partial s_m \\ \partial h_2/\partial s_1 & \partial h_2/\partial s_2 & \ldots & \partial h_2/\partial s_m \\ \ldots & \ldots & \ldots & \ldots \\ \partial h_m/\partial s_1 & \partial h_m/\partial s_2 & \ldots & \partial h_m/\partial s_m \end{bmatrix}$$

# Reduced gradient: Solve for $\partial \mathbf{s}$

Rearrange the separated equation

$$\partial \mathbf{h} = (\partial \mathbf{h}/\partial \mathbf{s})\partial \mathbf{s} + (\partial \mathbf{h}/\partial \mathbf{d})\partial \mathbf{d} = \mathbf{0}$$



to get:



$$\partial \mathbf{s} = -(\partial \mathbf{h}/\partial \mathbf{s})^{-1}(\partial \mathbf{h}/\partial \mathbf{d})\partial \mathbf{d}$$



This derivation gives us a very useful expression for $\partial \mathbf{s}/\partial \mathbf{d}$

# Reduced gradient: Re-frame objective

$$\min_{\mathbf{d}} f(\mathbf{x}) = z(\mathbf{d}, \mathbf{s}(\mathbf{d}))$$

Take the gradient of this "reduced" objective:



$$\partial z / \partial \mathbf{d} = (\partial f / \partial \mathbf{d}) + (\partial f / \partial \mathbf{s}) \boxed{(\partial \mathbf{s} / \partial \mathbf{d})}$$



Plugging in for our previously-derived $\partial \mathbf{s} / \partial \mathbf{d}$:

$$\boxed{\partial z / \partial \mathbf{d} = (\partial f / \partial \mathbf{d}) - (\partial f / \partial \mathbf{s})(\partial \mathbf{h} / \partial \mathbf{s})^{-1}(\partial \mathbf{h} / \partial \mathbf{d})}$$

**This is the reduced gradient!**

# Reduced gradient: Solve with FONC

$$\min_{\mathbf{d}} f(\mathbf{x}) = z(\mathbf{d}, \mathbf{s}(\mathbf{d}))$$

1. From FONC, find stationary points: $\partial z/\partial \mathbf{d} = \mathbf{0}^T$



2. Next, plug into formulation from before:



$$\partial z/\partial \mathbf{d} = (\partial f/\partial \mathbf{d}) - (\partial f/\partial \mathbf{s})(\partial \mathbf{h}/\partial \mathbf{s})^{-1}(\partial \mathbf{h}/\partial \mathbf{d}) = \mathbf{0}^T$$

3. Then, from optimal $\mathbf{d}^*$, find $\mathbf{s}^*$ with $\mathbf{h}(\mathbf{s}^*, \mathbf{d}^*) = \mathbf{0}$

4. Finally, from optimizers $\mathbf{d}^*$ and $\mathbf{s}^*$, calculate $f^*$

# Example 1: Reduced Gradient

$$\min_{x_1, x_2} \quad f = (x_1 - 2)^2 + (x_2 - 2)^2$$

$$\text{subject to} \qquad h = x_1^2 + x_2^2 - 4 = 0$$



**Step 1:** Partition $\mathbf{x}$ into $n - m$ decision variables $\mathbf{d}$ and $m$ state variables $\mathbf{s}$

**Step 2:** Formulate $\partial f / \partial \mathbf{s}$, $\partial f / \partial \mathbf{d}$, $\partial \mathbf{h} / \partial \mathbf{s}$, $\partial \mathbf{h} / \partial \mathbf{d}$

**Step 3:** Calculate reduced gradient $\partial z / \partial \mathbf{d}$ using Step 2, and solve for $\partial z / \partial \mathbf{d}$= $\mathbf{0}$

**Step 4:** Solve for $\mathbf{s}^*$ from $\mathbf{d}^*$ using $\mathbf{h}$, then find $f^*$

# Example 1: Reduced gradient

$$\min_{x_1, x_2} \quad f = (x_1 - 2)^2 + (x_2 - 2)^2$$

$$\text{subject to} \qquad h = x_1^2 + x_2^2 - 4 = 0$$



1. Since we have 1 active constraint and 2 variables, we should only have 2-1 = 1 degree of freedom.

   **Partition:** $s = x_1,\ d = x_2$





2. **Formulate:** $f = (s-2)^2 + (d-2)^2$

   $h = s^2 + d^2 - 4 = 0$

3. **Calculate:** $\dfrac{\partial z}{\partial d} = \dfrac{\partial f}{\partial d} - \dfrac{\partial f}{\partial s}\left(\dfrac{\partial h}{\partial s}\right)^{-1}\left(\dfrac{\partial h}{\partial d}\right)$

$$\frac{\partial z}{\partial d} = (2d - 4) - (2s - 4)(2s)^{-1}(2d)$$

# Example 1: Reduced gradient

$$\min_{x_1, x_2} \quad f = (x_1 - 2)^2 + (x_2 - 2)^2$$

$$\text{subject to} \qquad h = x_1^2 + x_2^2 - 4 = 0$$



$$\frac{\partial z}{\partial d} = (2d - 4) - (2s - 4)(2s)^{-1}(2d)$$



Reduce: $$\frac{\partial z}{\partial d} = (2d - 4) - \left(2d - 4\frac{d}{s}\right)$$



$$\frac{\partial z}{\partial d} = 4\frac{d}{s} - 4$$

Solve using FONC, $\frac{\partial z}{\partial d} = 0$:

$$4\frac{d}{s} - 4 = 0 \rightarrow \frac{d}{s} = 1 \rightarrow d = s$$

# Example 1: Reduced gradient

$$\min_{x_1, x_2} \quad f = (x_1 - 2)^2 + (x_2 - 2)^2$$

$$\text{subject to} \qquad h = x_1^2 + x_2^2 - 4 = 0$$



4. Now that we know $d = s$, plug back into $h$:



$$h = s^2 + s^2 - 4 = 0$$



$$2s^2 = 4 \;\rightarrow\; s^* = \sqrt{2} = d^*$$

Plug back into $f$:

$$f^* = (\sqrt{2} - 2)^2 + (\sqrt{2} - 2)^2 = 2(2 - 4\sqrt{2} + 4)$$

$$f^* = 12 - 8\sqrt{2} \approx 0.6863$$

Note: This is just an updated version of the FONC and SOSC solution approach!

# Generalized Reduced Gradient (GRG)

**GRG is an iterative <u>algorithm</u> similar to gradient descent, but using the Reduced Gradient concept**

In each iteration, update $\mathbf{x}_k = [\mathbf{d}_k, \mathbf{s}_k]$:



First, update decision variables in each iteration



$$\mathbf{d}_{k+1} = \mathbf{d}_k - \alpha_k \left( {}^{\partial z}/_{\partial \mathbf{d}} \right)_k^T$$



and then adjust the state variables by starting with

$$\mathbf{s}'_{k+1} = \mathbf{s}_k + \alpha_k \left( {}^{\partial \mathbf{h}}/_{\partial \mathbf{s}} \right)_k^{-1} \left( {}^{\partial \mathbf{h}}/_{\partial \mathbf{d}} \right)_k \left( {}^{\partial z}/_{\partial \mathbf{d}} \right)_k^T$$

and updating it with the following until convergence:

$$[\mathbf{s}_{k+1}]_{j+1} = \left[ \mathbf{s}_{k+1} - \left( {}^{\partial \mathbf{h}}/_{\partial \mathbf{s}} \right)_{k+1}^{-1} \mathbf{h}(\mathbf{d}_{k+1}, \mathbf{s}_{k+1}) \right]_j$$

# Active set strategy

- Recall that the Reduced Gradient approach only works with **active constraints**, or when we know which constraints are active
- We can get around this with an active set strategy
- An **active set strategy** is an algorithmic approach that assumes a set of active constraints and updates them. In each iteration, it will:
  - Remove inactive constraints
  - Add violated constraints



This is similar to the LP algorithm!

# Lagrange multipliers



Another approach to dealing with constrained problems, which can be useful in algorithms

# Lagrange multipliers

Using FONC on Reduced Gradient formulation

$$\frac{\partial z}{\partial \mathbf{d}} = \frac{\partial f}{\partial \mathbf{d}} - \frac{\partial f}{\partial \mathbf{s}}\left(\frac{\partial \mathbf{h}}{\partial \mathbf{s}}\right)^{-1}\left(\frac{\partial \mathbf{h}}{\partial \mathbf{d}}\right) = \mathbf{0}$$





we redefine term in red as a new set of "variables"



$$\frac{\partial z}{\partial \mathbf{d}} = \frac{\partial f}{\partial \mathbf{d}} + \boldsymbol{\lambda}^T\left(\frac{\partial \mathbf{h}}{\partial \mathbf{d}}\right) = \mathbf{0}$$

$$\boldsymbol{\lambda}^T = -\frac{\partial f}{\partial \mathbf{s}}\left(\frac{\partial \mathbf{h}}{\partial \mathbf{s}}\right)^{-1}$$

The elements of this vector are called **Lagrange multipliers**

# Lagrange multipliers

- Rearrange:

$$\boldsymbol{\lambda}^T = -\frac{\partial f}{\partial \mathbf{s}}\left(\frac{\partial \mathbf{h}}{\partial \mathbf{s}}\right)^{-1}$$

$$\frac{\partial f}{\partial \mathbf{s}} + \boldsymbol{\lambda}^T\left(\frac{\partial \mathbf{h}}{\partial \mathbf{s}}\right) = \mathbf{0}^T$$



- Recall that we still have Reduced Gradient FONC:

$$\frac{\partial f}{\partial \mathbf{d}} + \boldsymbol{\lambda}^T\left(\frac{\partial \mathbf{h}}{\partial \mathbf{d}}\right) = \mathbf{0}^T$$

- We can combine these, since $\mathbf{x} = [\mathbf{s}, \mathbf{d}]$:

$$\frac{\partial f}{\partial \mathbf{x}} + \boldsymbol{\lambda}^T\left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right) = \mathbf{0}^T$$

To solve, we need more equations, since $\boldsymbol{\lambda}^T$ introduces more "variables":

$$\mathbf{h}(\mathbf{x}) = \mathbf{0}$$

$n$ equations

$m$ equations

# Lagrangian defined

- We define the **Lagrangian** as:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{h}(\mathbf{x})$$

- Which yields the equations we just discussed:



$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \mathbf{0}: \quad \frac{\partial f}{\partial \mathbf{x}} + \boldsymbol{\lambda}^T \left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right) = \mathbf{0}^T$$



$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}} = \mathbf{0}: \quad \mathbf{h}(\mathbf{x}) = \mathbf{0}$$



These are the FONCs for a strictly-constrained problem!

- With the Hessian of the Lagrangian, $\mathcal{L}_{xx}$, and a FONC point $\mathbf{x}^*$:

$$(\partial \mathbf{x}^*)^T \mathcal{L}_{xx} (\partial \mathbf{x}^*) > 0$$

$$\frac{\partial \mathbf{h}}{\partial \mathbf{x}} \partial \mathbf{x}^* = 0$$

SOSCs for a strictly-constrained problem!

# Solving with Lagrange multipliers

1. Define $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{h}(\mathbf{x})$
2. Find $\mathbf{x}^*$ using FONC: $\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \mathbf{0}$ and $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}} = \mathbf{0}$
3. Check SOSCs
   a) With Hessian $\mathcal{L}_{xx}$ at $\mathbf{x}^*$, $(\partial \mathbf{x}^*)^T \mathcal{L}_{xx} (\partial \mathbf{x}^*) > 0$
   b) $\frac{\partial \mathbf{h}}{\partial \mathbf{x}} \partial \mathbf{x}^* = 0$

# Example: Lagrange Multipliers

$$\max_{x_1, x_2} \quad f = x_1 x_2 + x_2 x_3 + x_1 x_3$$

$$\text{subject to} \quad h = x_1 + x_2 + x_3 - 3 = 0$$

Remember to convert to a minimization problem!

1. Define $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{h}(\mathbf{x})$:

$$\mathcal{L}(\mathbf{x}, \lambda) = -x_1 x_2 - x_2 x_3 - x_1 x_3 + \lambda x_1 + \lambda x_2 + \lambda x_3 - 3\lambda$$

2. Find $\mathbf{x}^*$ using FONC: $\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \mathbf{0}$ and $\mathbf{h}(\mathbf{x}) = \mathbf{0}$

$$\frac{\partial \mathcal{L}}{\partial x_1} = -x_2 - x_3 + \lambda = 0$$

$$\frac{\partial \mathcal{L}}{\partial x_2} = -x_1 - x_3 + \lambda = 0$$

$$\frac{\partial \mathcal{L}}{\partial x_3} = -x_1 - x_2 + \lambda = 0$$

$$h(\mathbf{x}) = x_1 + x_2 + x_3 - 3 = 0$$

$${x_1}^* = 1$$

$${x_2}^* = 1$$

$${x_3}^* = 1$$

$$\lambda^* = 2$$

$$f^* = 3$$

# Example: Lagrange Multipliers

$$\max_{x_1, x_2} \quad f = x_1x_2 + x_2x_3 + x_1x_3$$
$$\text{subject to} \quad h = x_1 + x_2 + x_3 - 3 = 0$$

3. With Hessian $\mathcal{L}_{xx}$ at $\mathbf{x}^*$, $(\partial\mathbf{x}^*)^T\mathcal{L}_{xx}(\partial\mathbf{x}^*) > 0$



4. $\dfrac{\partial\mathbf{h}}{\partial\mathbf{x}}\partial\mathbf{x}^* = 0$





$$\mathcal{L}_{xx} = \begin{bmatrix} 0 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & 0 \end{bmatrix}$$

$$[\partial x_1 \quad \partial x_2 \quad \partial x_3]\begin{bmatrix} 0 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & 0 \end{bmatrix}\begin{bmatrix} \partial x_1 \\ \partial x_2 \\ \partial x_3 \end{bmatrix} = -2(\partial x_1\partial x_2 + \partial x_2\partial x_3 + \partial x_1\partial x_3) \overset{?}{>} 0$$

Since $\frac{\partial\mathbf{h}}{\partial\mathbf{x}}\partial\mathbf{x}^* = \partial x_1 + \partial x_2 + \partial x_3 = 0 \quad \rightarrow \quad 2\left(\left(\partial x_1 + \frac{\partial x_2}{2}\right)^2 + \frac{3}{4}\partial {x_2}^2\right) > 0$

Therefore, $\mathbf{x}^* = [1,1,1]^T$ is a global minimum

# Karush-Kuhn-Tucker (KKT) conditions



FONC for problems with **both** equality and inequality constraints, regardless of activity

# Lagrangian with inequalities

$$\min_{\mathbf{x}} f(\mathbf{x})$$

$$\text{s.t.} \quad \mathbf{h}(\mathbf{x}) = \mathbf{0}$$

$$\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$$

$$\min_{\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu}} f(\mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{h}(\mathbf{x}) + \boldsymbol{\mu}^T \mathbf{g}(\mathbf{x})$$



Original Constrained Problem

New Lagrangian

$$\mathcal{L}(\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu}) = f(\mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{h}(\mathbf{x}) + \boldsymbol{\mu}^T \mathbf{g}(\mathbf{x})$$

Now both $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are vectors of Lagrange multipliers!

# Karush-Kuhn-Tucker (KKT) conditions

*First order necessary conditions for constrained problems*

1. $\nabla_{\mathbf{x}}\mathcal{L} = \nabla f(\mathbf{x}^*) + \boldsymbol{\lambda}^T \nabla \mathbf{h}(\mathbf{x}^*) + \boldsymbol{\mu}^T \nabla \mathbf{g}(\mathbf{x}^*) = \mathbf{0}^T$



2. $\mathbf{h}(\mathbf{x}^*) = \mathbf{0}$, $\mathbf{g}(\mathbf{x}^*) \leq \mathbf{0}$



3. $\boldsymbol{\lambda} \neq \mathbf{0}$, $\boldsymbol{\mu} \geq \mathbf{0}$



4. $\boldsymbol{\mu}^{\mathrm{T}} \mathbf{g}(\mathbf{x}^*) = \mathbf{0}$

**Note:** A design point $\mathbf{x}^*$ satisfying these 4 conditions is called a **KKT point**, which may or may not be a minimizer

# Second Order Sufficiency Conditions

Given a KKT point $\mathbf{x}^*$, if the Hessian of the Lagrangian in *feasible directions* is positive definite, then the point is a local minimizer:



$$\partial \mathbf{x}_*{}^T \mathcal{L}_{xx} \partial \mathbf{x}_* > 0$$



A *feasible direction* must satisfy:



$$\frac{\partial \mathbf{h}}{\partial \mathbf{x}} \partial \mathbf{x}^* = 0$$

$$\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \partial \mathbf{x}^* = 0$$

# Example: KKT Conditions

**Example 5.10** Consider the problem with $x_1, x_2 > 0$:

$$\min f = 8x_1^2 - 8x_1x_2 + 3x_2^2$$
$$\text{subject to } g_1 = x_1 - 4x_2 + 3 \le 0,$$
$$g_2 = -x_1 + 2x_2 \le 0.$$



**Step 1:** $\nabla_{\mathbf{x}}\mathcal{L} = \nabla f(\mathbf{x}^*) + \boldsymbol{\lambda}^T \nabla \mathbf{h}(\mathbf{x}^*) + \boldsymbol{\mu}^T \nabla \mathbf{g}(\mathbf{x}^*) = \mathbf{0}^T$



$$\nabla f(\mathbf{x}) = \begin{bmatrix} 16x_1 - 8x_2 \\ -8x_1 + 6x_2 \end{bmatrix}, \nabla \mathbf{g}(\mathbf{x}) = \begin{bmatrix} 1 & -4 \\ -1 & 2 \end{bmatrix},$$

$$\nabla \mathbf{h}(\mathbf{x}) = [\,]$$

$$\nabla_{\mathbf{x}}\mathcal{L} = \begin{bmatrix} 16x_1 - 8x_2 + \mu_1 - \mu_2 \\ -8x_1 + 6x_2 - 4\mu_1 + 2\mu_2 \end{bmatrix} = 0$$

# Example: KKT Conditions

**Example 5.10** Consider the problem with $x_1, x_2 > 0$:

$$\min f = 8x_1^2 - 8x_1x_2 + 3x_2^2$$
$$\text{subject to } g_1 = x_1 - 4x_2 + 3 \leq 0,$$
$$g_2 = -x_1 + 2x_2 \leq 0.$$



$$\nabla_{\mathbf{x}}\mathcal{L} = \begin{bmatrix} 16x_1 - 8x_2 + \mu_1 - \mu_2 \\ -8x_1 + 6x_2 - 4\mu_1 + 2\mu_2 \end{bmatrix} = 0$$



**Step 2: FONC**

$$16x_1 - 8x_2 + \mu_1 - \mu_2 = 0$$



$$-8x_1 + 6x_2 - 4\mu_1 + 2\mu_2 = 0$$
$$\mu_1(x_1 - 4x_2 + 3) = 0$$
$$\mu_2(-x_1 + 2x_2) = 0$$

To solve this, we can look at different scenarios for the bottom two equations, depending on whether each $\mu$ is zero:

Scenario 1: $\mu_1 = 0, \mu_2 = 0$

Scenario 2: $\mu_1 = 0, \mu_2 \neq 0$

Scenario 3: $\mu_1 \neq 0, \mu_2 = 0$

Scenario 4: $\mu_1 \neq 0, \mu_2 \neq 0$

# Example: KKT Conditions

**Step 2: FONC**

$$16x_1 - 8x_2 + \mu_1 - \mu_2 = 0$$
$$-8x_1 + 6x_2 - 4\mu_1 + 2\mu_2 = 0$$
$$\mu_1(x_1 - 4x_2 + 3) = 0$$
$$\mu_2(-x_1 + 2x_2) = 0$$



**Scenario 1:** $\mu_1 = 0, \mu_2 = 0$

$16x_1 - 8x_2 = 0$
$-8x_1 + 6x_2 = 0$
$g_1: x_1 - 4x_2 + 3 \leq 0$
$g_2: -x_1 + 2x_2 \leq 0$

$x_1 = x_2 = 0$
→ violates constraint $g_1$
**No Solution**

**Scenario 2:** $\mu_1 = 0, \mu_2 \neq 0$

$16x_1 - 8x_2 - \mu_2 = 0$
$-8x_1 + 6x_2 + 2\mu_2 = 0$
$-x_1 + 2x_2 = 0$
$g_1: x_1 - 4x_2 + 3 \leq 0$

$x_1 = x_2 = 0$
→ requires $\mu_2 = 0$; not allowed
**No solution**

# Example: KKT Conditions

**Step 2: FONC**

$$16x_1 - 8x_2 + \mu_1 - \mu_2 = 0$$
$$-8x_1 + 6x_2 - 4\mu_1 + 2\mu_2 = 0$$
$$\mu_1(x_1 - 4x_2 + 3) = 0$$
$$\mu_2(-x_1 + 2x_2) = 0$$



**Scenario 3:** $\mu_1 \neq 0, \mu_2 = 0$

$$16x_1 - 8x_2 + \mu_1 = 0$$
$$-8x_1 + 6x_2 - 4\mu_1 = 0$$
$$x_1 - 4x_2 + 3 = 0$$
$$g_2: -x_1 + 2x_2 \leq 0$$

$$x_1 = \frac{13}{33}, x_2 = \frac{28}{33}, \mu_1 = \frac{16}{33}$$

→ violates constraint $g_2$

→ No Solution

**Scenario 4:** $\mu_1 \neq 0, \mu_2 \neq 0$

$$16x_1 - 8x_2 + \mu_1 - \mu_2 = 0$$
$$-8x_1 + 6x_2 - 4\mu_1 + 2\mu_2 = 0$$
$$x_1 - 4x_2 + 3 = 0$$
$$-x_1 + 2x_2 = 0$$

$$x_1 = 3, x_2 = \frac{3}{2}, \mu_1 = \frac{57}{2}, \mu_2 = \frac{129}{2}$$

**KKT point!**

# Example: KKT Conditions

**KKT point:** $x_1 = 3, x_2 = \frac{3}{2}, \mu_1 = \frac{57}{2}, \mu_2 = \frac{129}{2}$

**Step 3: Check SOSC**



$$\nabla_{\mathbf{x}}\mathcal{L} = \begin{bmatrix} 16x_1 - 8x_2 + \mu_1 - \mu_2 \\ -8x_1 + 6x_2 - 4\mu_1 + 2\mu_2 \end{bmatrix} = 0$$



$$\mathcal{L}_{xx} = \begin{bmatrix} 16 & -8 \\ -8 & 6 \end{bmatrix}$$

Since $\mathcal{L}_{xx}$ is pos. def. everywhere, the unique KKT point $\mathbf{x}^* = \begin{bmatrix} 3 \\ 1.5 \end{bmatrix}$ is a global minimizer at $f^* = 42.75$.

# Example 2: KKT Conditions

$$\min_{x} \quad f = -x_1$$

$$\text{subject to} \quad g_1 = x_2 - (1 - x_1)^3 \leq 0$$

$$g_2 = -x_1 \leq 0$$



$$g_3 = -x_2 \leq 0$$



**Step 1:** $\nabla_{\mathbf{x}}\mathcal{L} = \nabla f(\mathbf{x}^*) + \boldsymbol{\lambda}^T \nabla \mathbf{h}(\mathbf{x}^*) + \boldsymbol{\mu}^T \nabla \mathbf{g}(\mathbf{x}^*) = \mathbf{0}^T$



$$\nabla f(\mathbf{x}) = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \nabla \mathbf{g}(\mathbf{x}) = \begin{bmatrix} 3(1-x_1)^2 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, \nabla \mathbf{h}(\mathbf{x}) = [\,]$$

$$\nabla_{\mathbf{x}}\mathcal{L} = \begin{bmatrix} -1 + 3(1-x_1)^2\mu_1 - \mu_2 \\ \mu_1 - \mu_3 \end{bmatrix} = 0$$

$$\mu_2 = 3(1-x_1)^2\mu_1 - 1$$
$$\mu_1 = \mu_3$$

2 equations, 5 unknowns...

# Example: KKT Conditions

$$\min_{x} \quad f = -x_1$$

$$\text{subject to} \quad g_1 = x_2 - (1-x_1)^3 \leq 0$$

$$g_2 = -x_1 \leq 0$$



$$g_3 = -x_2 \leq 0$$

2 eqns from previous slide:

$$\mu_2 = 3(1-x_1)^2\mu_1 - 1$$

$$\mu_1 = \mu_3$$

$$\boldsymbol{\mu}^T \mathbf{g}(\mathbf{x}^*) = 0$$





3 more equations

$$\mu_1[x_2 - (1-x_1)^3] = 0$$

$$-\mu_2 x_1 = 0$$

$$-\mu_3 x_2 = 0$$

Solution to 5 eqns:

$$x_1{}^* = 0$$

$$x_2{}^* = [0,1]$$

(anywhere in range)

$$\mu_1 = 0$$

$$\mu_2 = -1$$

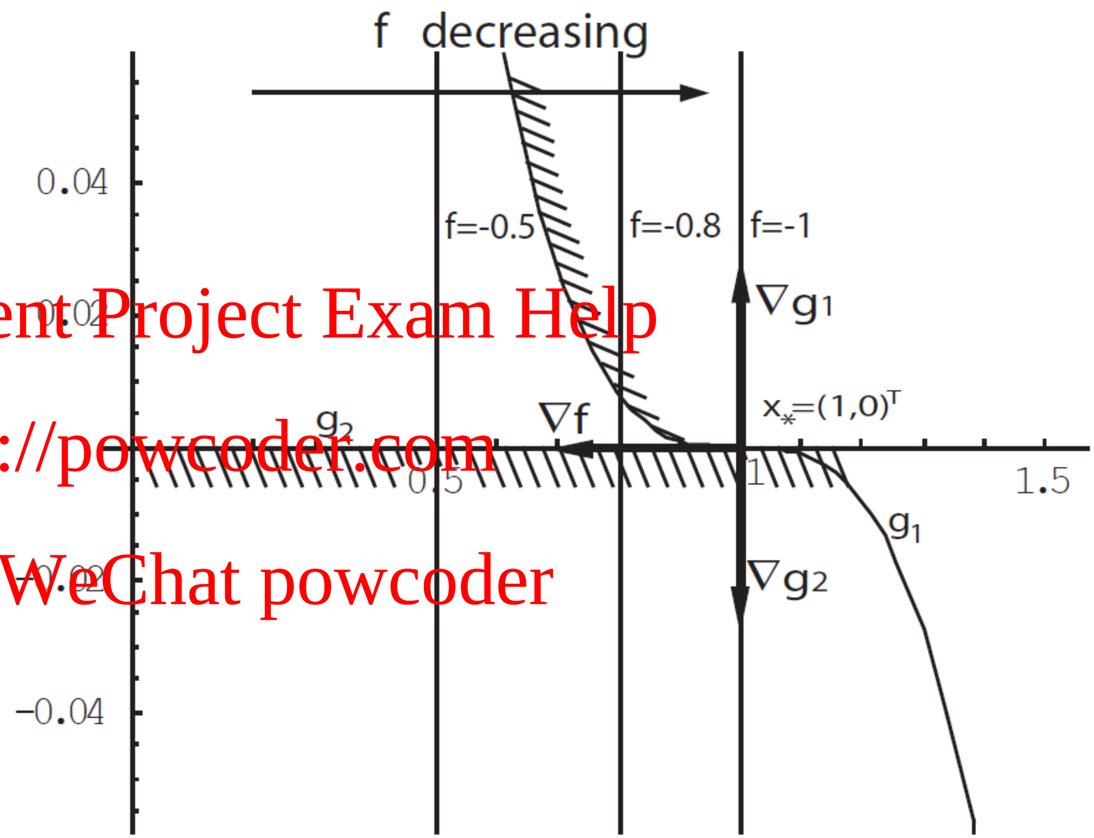
$$\mu_3 = 0$$

This violates $\boldsymbol{\mu} \geq \mathbf{0}$ !!

Now we have 5 equations, 5 unknowns

# Example 2: What went wrong?

The constraints are not linearly independent, causing an inability to find a valid KKT point.






<u>Important caveat:</u> The approaches in Chapter 5 assume that any KKT point found is a "regular point", where the active constraints are all linearly independent

# Quasi-Newton and SQP algorithms





Two common algorithms that use derivatives

# Quasi-Newton methods

***H** and **H**$^{-1}$ are difficult to compute, so we can avoid them by approximating **H** or **H**$^{-1}$ using only first derivatives:*

1. Begin with $\mathbf{x}_0$ and some assumed $\mathbf{H}_0^{-1}$.
2. For iteration $k$, set $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{H}_k^{-1} \nabla f(\mathbf{x}_k)$.
3. Compute an update matrix $\hat{\mathbf{H}}_k^{-1}$ as some function of $[\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)]$, $[\mathbf{x}_{k+1} - \mathbf{x}_k]$, and $\mathbf{H}_k^{-1}$.
4. Update inverse Hessian approximation: $\mathbf{H}_{k+1}^{-1} = \mathbf{H}_k^{-1} + \hat{\mathbf{H}}_k^{-1}$.



*Example update function: Davidon-Fletcher-Powell (DFP)*

$$\hat{\mathbf{H}}_{k+1}^{-1} = \hat{\mathbf{H}}_k^{-1} + \left[\frac{(\mathbf{x}_{k+1} - \mathbf{x}_k)(\mathbf{x}_{k+1} - \mathbf{x}_k)^T}{(\mathbf{x}_{k+1} - \mathbf{x}_k)^T (\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k))}\right] - \left[\frac{\left(\hat{\mathbf{H}}_k^{-1} (\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k))\right)\left(\hat{\mathbf{H}}_k^{-1} (\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k))\right)^T}{(\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k))^T \hat{\mathbf{H}}_k^{-1} (\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k))}\right]$$

For the derivation of this, see Ch. 6

# Sequential quadratic programming (SQP)

1. Initialize problem
2. Solve the QP sub-problem to determine search direction $\boldsymbol{s}_k$



$$\underset{\mathbf{s}_k}{\text{minimize}} \quad q(\mathbf{s}_k) = f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T \mathbf{s}_k + \frac{1}{2}\mathbf{s}_k^T \nabla_{xx}^2 \mathscr{L}(\mathbf{x}_k, \lambda_k, \mu_k)\mathbf{s}_k$$



$$\text{where} \quad \mathscr{L}(\mathbf{x}_k, \lambda_k, \mu_k) = f(\mathbf{x}_k) - \lambda^T \mathbf{h}(\mathbf{x}_k) - \mu^T \mathbf{g}(\mathbf{x}_k)$$



$$\text{subject to} \quad \mathbf{g}(\mathbf{x}_k) + \nabla \mathbf{g}(\mathbf{x}_k)^T \mathbf{s}_k \leq \mathbf{0}$$

$$\mathbf{h}(\mathbf{x}_k) + \nabla \mathbf{h}(\mathbf{x}_k)^T \mathbf{s}_k = \mathbf{0}$$

3. Set $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k$
4. Continue (2) and (3) until satisfied

This is one of the most common, robust algorithms

# Constrained problems

*We can handle constraints in a number of ways*

1. **Lagrange multipliers:**

$$\min \; L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{h}(\mathbf{x}) + \boldsymbol{\mu}^T \mathbf{g}(\mathbf{x})$$



2. **Penalty function:** add to the objective such that it is zero on the interior and positive when violating a constraint





$$\min T(\mathbf{x}) = f(\mathbf{x}) + \sum_{j=1}^{m} \left[\max\{0, g_j(\mathbf{x})\}\right]^2$$

3. **Barrier function:** add to the objective such that it gets really high at the constraint boundary

$$\min \; T(\mathbf{x}) = f(\mathbf{x}) - \sum_{j=1}^{m} \ln(-g(\mathbf{x}))$$

# Gradient-based approaches summary

*Using derivatives to find a local minimum of a differentiable function (or surrogate model)*

- **Unconstrained**
  - **FONC and SOSC** (when math is simple enough to solve)
  - **Gradient descent** (algorithm with linear convergence)
  - **Newton method** (algorithm with hyper-linear convergence)
- **Constrained**
  - **Reduced gradient** (analytical with known active constraints)
  - **Generalized Reduced Gradient** (algorithm w active constr)
  - **Active set strategy** (algorithm w updating set of active constr)
  - **Lagrangian** (equality or active inequality constr)
  - **KKT conditions** (with any inequality and equality constr)
  - **Quasi-Newton methods** ($2^{nd}$-derivative-free)
  - **SQP** (efficiently handles constraints)

# Acknowledgements

- Much of this material came from Chapters 5 & 6 of the textbook, *Principles of Optimal Design*
- Some of these slides and examples came from Drs. Emrah Bayrak and Namwoo Kang at the University of Michigan

# Announcements

- HW3 is due on Tuesday at noon
- My office hours are Monday 2-4pm
- I will be available for appointments on Thursday and Friday of this week, so start early! I am happy to answer ***specific*** questions about your work or code over email. I probably will not respond over the weekend.
- Project progress reports are due in 2.5 weeks! There is a template online that you should use.

# Exercises

# Exercise (optional): Lagrange Multipliers

$$\min_{x_1,x_2,x_3} \quad f = x_1^2 + x_2^2 + x_3^2$$

$$\text{subject to} \quad 5x_1^2 + 4x_2^2 + x_3^2 - 20 = 0$$

$$x_1 + x_2 - x_3 = 0$$

# Exercise (optional): Gradient descent

*Apply gradient descent in MATLAB to the Rosenbrock function:*

$$f(x, y) = (x-1)^2 + 100\left(y - x^2\right)^2$$



1. Begin with a feasible point $\mathbf{x}_0$



2. Find the gradient at that point $\boldsymbol{\nabla} f(\mathbf{x}_0)$



3. Move in the direction of the negative gradient to find an improved $\mathbf{x}_1$

$$\mathbf{x}_1 = \mathbf{x}_0 - \alpha \boldsymbol{\nabla} f(\mathbf{x}_0)$$

4. Continue to iterate until you stop improving

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \left[\frac{\boldsymbol{\nabla} f^T(\mathbf{x}_{k-1}) \boldsymbol{\nabla} f(\mathbf{x}_{k-1})}{\boldsymbol{\nabla} f^T(\mathbf{x}_{k-1}) \mathbf{H}(\mathbf{x}_{k-1}) \boldsymbol{\nabla} f(\mathbf{x}_{k-1})}\right] \boldsymbol{\nabla} f(\mathbf{x}_{k-1})$$