Intro to gradient-based optimization

ME 564/SYS 564
Wed Sep 26, 2018
Steven Hoffenson

∇f

<u>Goal of Week 5</u>: To learn the optimality conditions for unconstrained problems, be able to solve problems with them, and know two derivative-based algorithms

Recap: How to optimize

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

(Weeks 1-2, 4, 9-12)

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

TODAY

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

(Weeks 3, 5-8, 12)

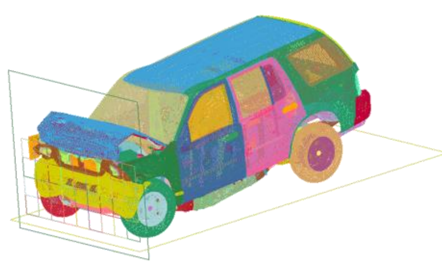
$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\mathbf{H}(\mathbf{x}_k)]^{-1}\nabla f(\mathbf{x}_0)$$

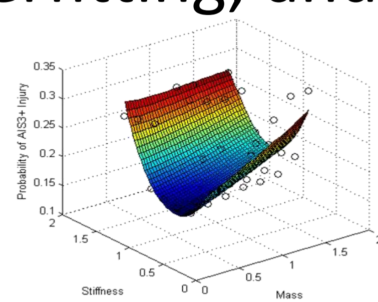
Recap: Week 3

- **Linear programs** are special cases
  - All functions **monotonic**
  - Solutions must lie on **boundary** of design space
  - **Simplex algorithm** is efficient
- **Derivative-free algorithms** for nonlinear problems are **straightforward** and **robust**, but may **take longer** and often converge on **local optima**
  - Coordinate search
  - Nelder-Meade
  - Space-filling DIRECT
  - Simulated Annealing

Recap: DOEs and surrogate modeling

- **Surrogate modeling** is fitting a mathematical function to your data to speed up evaluations and optimization
- This involves **four general steps**:
  - Gather data (e.g., using a DOE)
  - Choose a function structure (e.g., linear, polynomial, kriging, ANN)
  - Fit a function to the data
  - Assess fitness
- **Watch out** for outliers, underfitting, and overfitting



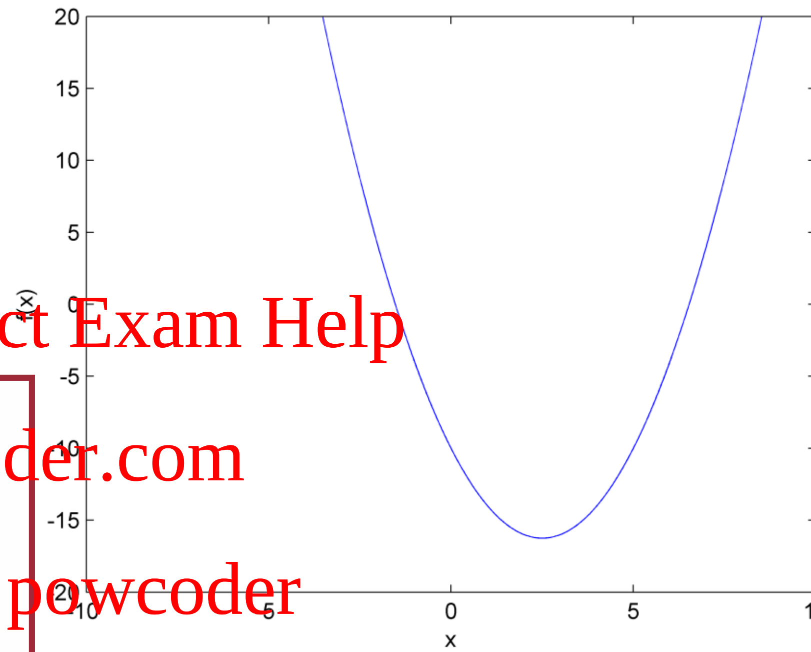


Unconstrained gradient-based methods

"Unconstrained" means we are talking about problems that have interior optima, not optima that lie on a constraint (like what monotonicity analysis can help with)

Basic nonlinear problem

$$\min_{x \in R} \quad f(x) = x^2 - 5x - 10$$



How do we solve this?

take derivative:	$\frac{\partial f}{\partial x} = 2x - 5$
set equal to 0:	$2x^* - 5 = 0$
solve for x:	$x^* = 5/2$
plug into function:	$f(x^*) = (5/2)^2 - 5(5/2) - 10$
	$f(x^*) = -16.25$

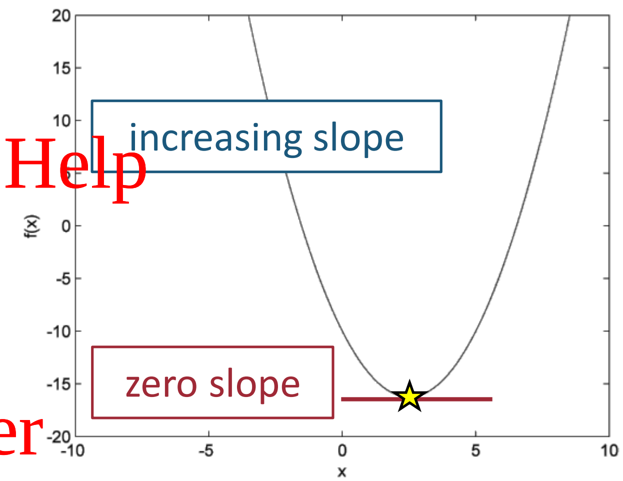
How do we know it's a minimizer?

take 2nd derivative:	$\frac{\partial^2 f}{\partial x^2} = 2$
plug in x^*:	$\frac{\partial^2 f}{\partial x^2}(5/2) = 2$
if positive:	x^* is a (local) minimum
if negative:	x^* is a (local) maximum

1-variable optimality conditions

1. *First-order necessary condition*

If $f(x)$ is differentiable and x^* is a local minimum, then $\frac{\partial f}{\partial x}(x^*) = 0.$



Note: **this is not *sufficient*** - x^* could be a local maximum or saddle point (rather than a minimum)

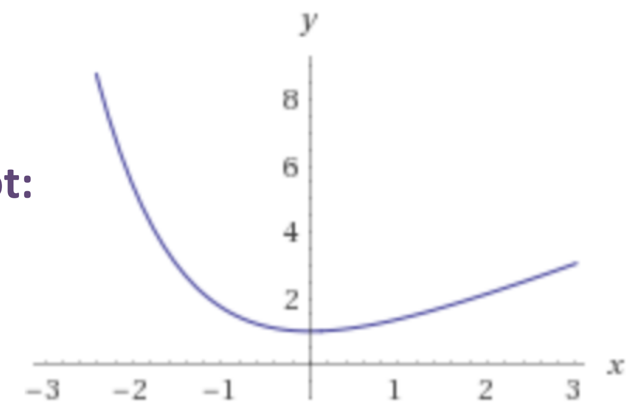
2. *Second-order sufficient condition*

If $\frac{\partial f}{\partial x}(x^*) = 0$ and $\frac{\partial^2 f}{\partial x^2}(x^*) > 0$, then x^* is a local minimum.

Example

$$\min f(x) = x + e^{-x}$$

Confirm with plot:



First-order necessary condition

$$\frac{df}{dx} = 1 - e^{-x}$$

$$\frac{df}{dx}(x^*) = 1 - e^{-x^*} = 0$$

$$1 = e^{-x^*}$$

$$\ln(1) = -x^*$$

$$x^* = 0$$

This is a stationary point

Second-order sufficient condition

$$\frac{d^2 f}{dx^2} = e^{-x}$$

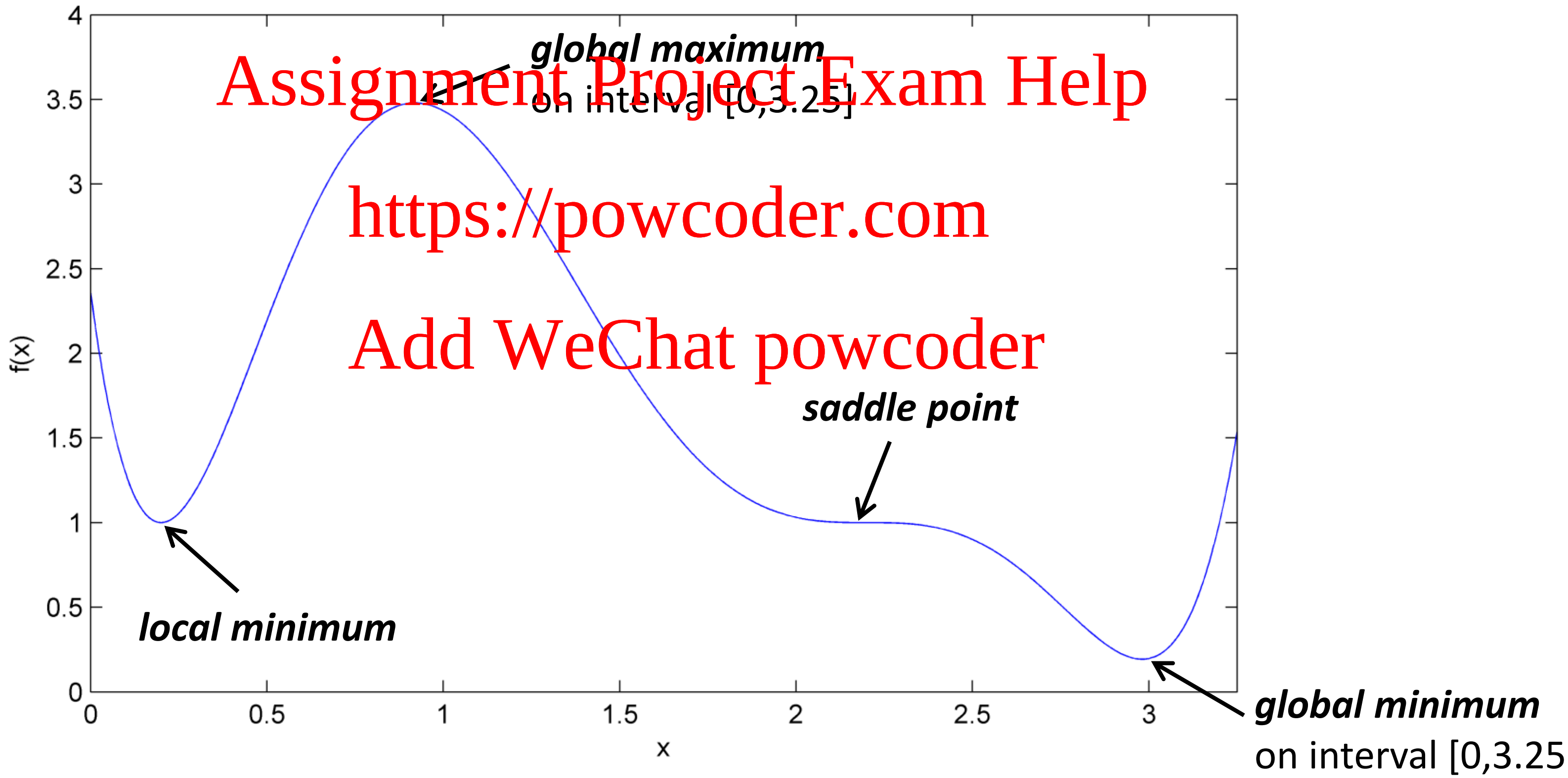
$$\frac{d^2 f}{dx^2}(x^*) = e^{-x^*}$$

$$\frac{d^2 f}{dx^2}(0) = e^0 = 1$$

Since this is >0, it is a minimum

Global vs. local optima

Stationary points $\left[\frac{\partial f}{\partial x}(x^*) = 0\right]$ *can be minima, maxima, or saddle points*



Multiple variables

Gradient: $$\nabla f(\mathbf{x}) \triangleq \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n}\right)$$

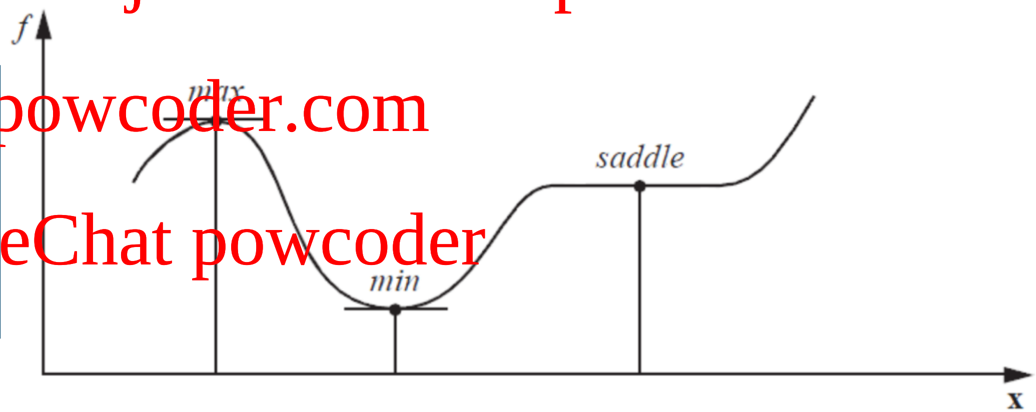
Hessian: $$\mathbf{H}(\mathbf{x}) \triangleq \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_n x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

Multi-variable optimality conditions

1. *First-order necessary condition*

If $f(\mathbf{x})$ is differentiable and $\mathbf{x}^*$ is a local minimum, then $\nabla f(\mathbf{x}^*) = \mathbf{0}$.

Recall: Points that satisfy the necessary condition are called "stationary points," and they are not all minima!



2. *Second-order sufficient condition*

If $\nabla f(\mathbf{x}^*) = \mathbf{0}$ and $\mathbf{H}(\mathbf{x}^*)$ is positive-definite, then $\mathbf{x}^*$ is a local minimum

Hessian properties

Note: $\partial\mathbf{x} = \mathbf{x} - \mathbf{x}^*$

A Hessian is ***positive-definite*** if $\partial\mathbf{x}^T\mathbf{H}(\mathbf{x}^*)\partial\mathbf{x} > 0$ for all $\partial\mathbf{x} \neq \mathbf{0}$.

A matrix is positive-definite if and only if any of these hold:

1. *All of its eigenvalues are positive.*
2. *All determinants of its leading principal minors are positive.*
3. *All the pivots are positive when the matrix is reduced to row-echelon form.*

Other matrix classification definitions

Replace positive-definite with:	Replace > with:	Nature of x*
negative-definite	<	local maximum
positive-semidefinite	≥	probable valley
negative-semidefinite	≤	probable ridge
indefinite	have both + & -	saddle point

Note: The 3 "tricks" above only apply for positive-definite, and cannot all be used to prove negative- or semi-definiteness

Sometimes we can solve problems using the optimality conditions

1. Apply First Order Necessary Condition (FONC)

   Find stationary points $\mathbf{x}^*$ where

$$\nabla f(\mathbf{x}^*) \triangleq \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}\right) = (0,0,\ldots,0)$$

2. Apply Second Order Sufficient Condition (SOSC)

   Test each stationary point $\mathbf{x}^*$ for a positive-definite Hessian

$$\partial\mathbf{x}^T \mathbf{H}(\mathbf{x}^*)\partial\mathbf{x} > 0 \text{ for all } \partial\mathbf{x} \neq \mathbf{0}$$

If a point $\mathbf{x}^*$ meets both conditions, it is a local minimum!

Example (4.6 in book)

Find the minimum of the following function:

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

$$\nabla f(\mathbf{x}) = [2 - 2x_1^{-3} \quad 2 - 2x_2^{-3}] \xrightarrow{\text{FONC}} \mathbf{x}_* = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Stationary Point

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} 6x_1^{-4} & 0 \\ 0 & 6x_2^{-4} \end{bmatrix} \longrightarrow \mathbf{H}(\mathbf{x}_*) = \begin{bmatrix} 6 & 0 \\ 0 & 6 \end{bmatrix}$$

$$\partial \mathbf{x} = \begin{bmatrix} \partial x_1 \\ \partial x_2 \end{bmatrix} = \mathbf{x} - \mathbf{x}_* = \begin{bmatrix} x_1 - 1 \\ x_2 - 1 \end{bmatrix}$$

Therefore, [1;1] is a minimum!

SOSC

$$\partial \mathbf{x}^{\mathrm{T}} \mathbf{H}(\mathbf{x}_*) \partial \mathbf{x} = 6(\partial x_1)^2 + 6(\partial x_2)^2 > 0$$ Positive Definite

Eigenvalues to test SOSC

Rather than checking the quadratic term $\partial \mathbf{x}^T \mathbf{H}(\mathbf{x}^*) \partial \mathbf{x} > 0$, we can check eigenvalues of $\mathbf{H}(\mathbf{x}^*)$

Eigenvalues of $\mathbf{H}(\mathbf{x}_*)$	Hessian Matrix	Nature of $\mathbf{x}_*$
All Positive (>0)	Positive definite	Local min
All Negative (<0)	Negative definite	Local max
All Nonnegative (≥ 0)	Positive semidefinite	Probable valley
All Nonpositive (≤ 0)	Negative semidefinite	Probable ridge
Any sign	Indefinite	Saddle point

Example

Recall our example

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} 6x_1^{-4} & 0 \\ 0 & 6x_2^{-4} \end{bmatrix} \qquad \mathbf{H}(\mathbf{x}_*) = \begin{bmatrix} 6 & 0 \\ 0 & 6 \end{bmatrix}$$

Eigenvalues of the general Hessian:

$\lambda_1 = 6x_1^{-4}$ and $\lambda_2 = 6x_2^{-4}$

Positive definite everywhere

Eigenvalues at $\mathbf{x}_*$:

$\lambda_1 = 6$ and $\lambda_2 = 6$

Positive definite at $\mathbf{x}_*$

Local Minimum

Determinants to test SOSC

If your matrix is 2x2, you can simply check the determinant of $\mathbf{H}(\mathbf{x}_*)$

Determinant of $\mathbf{H}(\mathbf{x}_*)$	Hessian Matrix	Nature of $\mathbf{x}_*$
Positive and $h_{11} > 0$	Positive definite	Local min
Positive and $h_{11} < 0$	Negative definite	Local max
Zero and $h_{11} > 0$	Positive semidefinite	Probable valley
Zero and $h_{11} < 0$	Negative semidefinite	Probable ridge
Negative	Indefinite	Saddle point

Note: h_{11} is the first element (upper-left value) of **H**

Another note: h_{11} is the first leading principal minor of **H**, and the full matrix is the second leading principal minor. This follows option 2 from Slide 12.

Example

Recall our example

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} 6x_1^{-4} & 0 \\ 0 & 6x_2^{-4} \end{bmatrix} \Rightarrow \mathbf{H}(\mathbf{x}_*) = \begin{bmatrix} 6 & 0 \\ 0 & 6 \end{bmatrix}$$

$$\det(\mathbf{H}\,(\mathbf{x}_*)) = 36 > 0$$

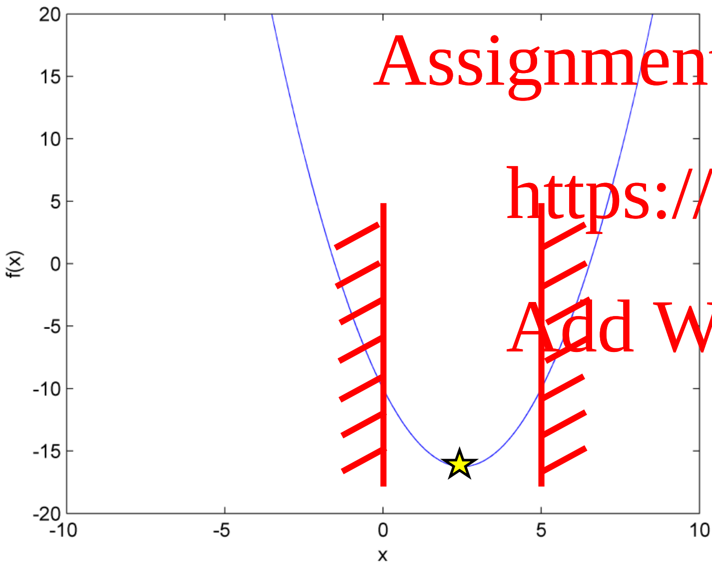
$$h_{11} = 6 > 0$$

Positive definite at $\mathbf{x}_*$

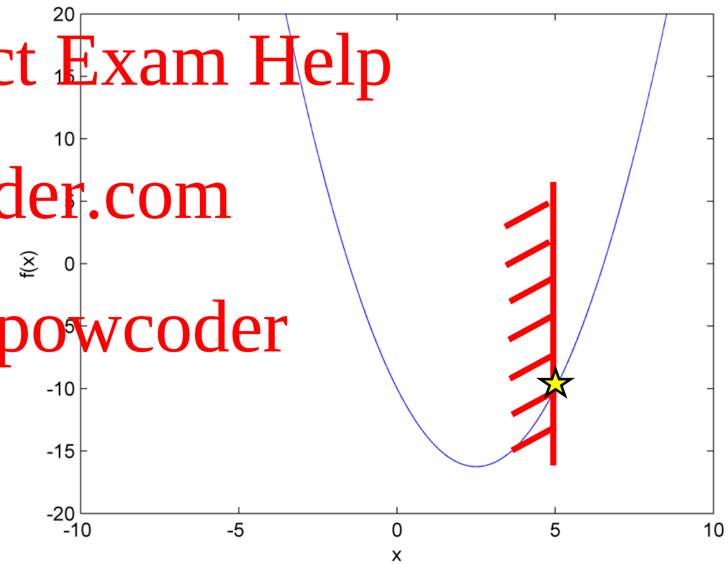
Local Minimum

Interior vs. boundary optima

Interior optimum



Boundary optimum



Note that the necessary and sufficient conditions usually do not apply to boundary optima

Local approximation

A review of Taylor series expansion for approximating function behavior (in a local neighborhood)

Taylor Series Approximation

Taylor series approximation for a single variable function

$$f(x) = f(x_0) + \frac{f'(x_0)}{1!}(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \frac{f^{(3)}(x_0)}{3!}(x - x_0)^3 + \cdots$$

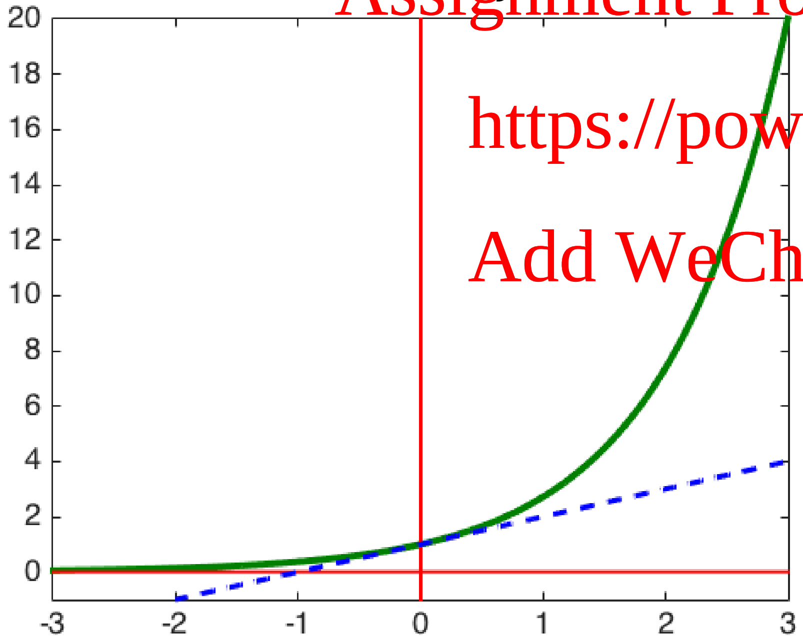
Higher-order terms

Approximation is valid only in the neighborhood of x_0

Taylor Series Approximation

Linear Approximation

$$f(x) = e^x$$



$$f(x) \approx f(x_0) + f'(x_0)(x - x_0)$$

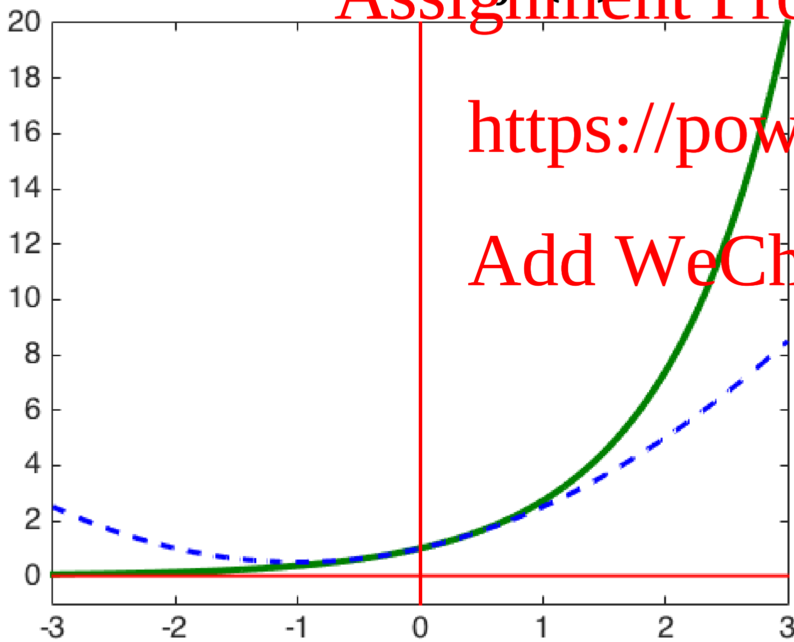
for $x_0 = 0$

$$f(x) \approx 1 + x$$

Taylor Series Approximation

Quadratic Approximation

$f(x) = e^x$



$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2} f''(x_0)(x - x_0)^2$$

for $x_0 = 0$

$$f(x) \approx 1 + x + \frac{1}{2} x^2$$

Taylor Series Approximation

In higher dimensions

$$f(\mathbf{x}) \approx f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0)[\mathbf{x} - \mathbf{x}_0] + \frac{1}{2}[\mathbf{x} - \mathbf{x}_0]^{\mathrm{T}}\mathbf{H}(\mathbf{x}_0)[\mathbf{x} - \mathbf{x}_0] + \cdots$$

Recall:

Gradient vector

$$\nabla f(\mathbf{x}) = \left[{}^{\partial f}/_{\partial x_1}, \quad {}^{\partial f}/_{\partial x_2}, \quad \dots, \quad {}^{\partial f}/_{\partial x_n} \right]$$

Hessian matrix

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} {}^{\partial^2 f}/_{\partial x_1^2} & \cdots & {}^{\partial^2 f}/_{\partial x_1 x_n} \\ \vdots & \ddots & \vdots \\ {}^{\partial^2 f}/_{\partial x_n x_1} & \cdots & {}^{\partial^2 f}/_{\partial x_n^2} \end{bmatrix}$$

Taylor Series Approximation

$$f(\mathbf{x}) \approx f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0)[\mathbf{x}-\mathbf{x}_0] + \frac{1}{2}[\mathbf{x}-\mathbf{x}_0]^{\mathrm{T}}\mathbf{H}(\mathbf{x}_0)[\mathbf{x}-\mathbf{x}_0]$$

$$\underbrace{f(\mathbf{x}) - f(\mathbf{x}_0)}_{\partial f} \approx \nabla f(\mathbf{x}_0)\underbrace{[\mathbf{x}-\mathbf{x}_0]}_{\partial \mathbf{x}} + \frac{1}{2}[\mathbf{x}-\mathbf{x}_0]^{\mathrm{T}}\mathbf{H}(\mathbf{x}_0)[\mathbf{x}-\mathbf{x}_0]$$

(Perturbation in f) (Perturbation in $\mathbf{x}$)

$$\partial f \approx \nabla f(\mathbf{x}_0)\partial \mathbf{x} + \frac{1}{2}\partial \mathbf{x}^{\mathrm{T}}\mathbf{H}(\mathbf{x}_0)\partial \mathbf{x}$$

We will use this to develop our first two algorithms!

Basic gradient-based algorithms

1st-order: Gradient descent

2nd-order: Newton’s method

1^{st}-order algorithm: Gradient descent

Starting at a point $\mathbf{x}_0$, we want to find a direction that will lower the objective value

1. Using 1^{st}-order terms only from Taylor approx.,

$$\partial f \approx \nabla f(\mathbf{x}_0)\partial\mathbf{x}$$

We want $\partial f < 0$. If we choose $\partial\mathbf{x} = -\nabla f(\mathbf{x}_0)$, then we know $\partial f \approx -[\nabla f(\mathbf{x}_0)]^2 < 0$

This is our direction of descent!

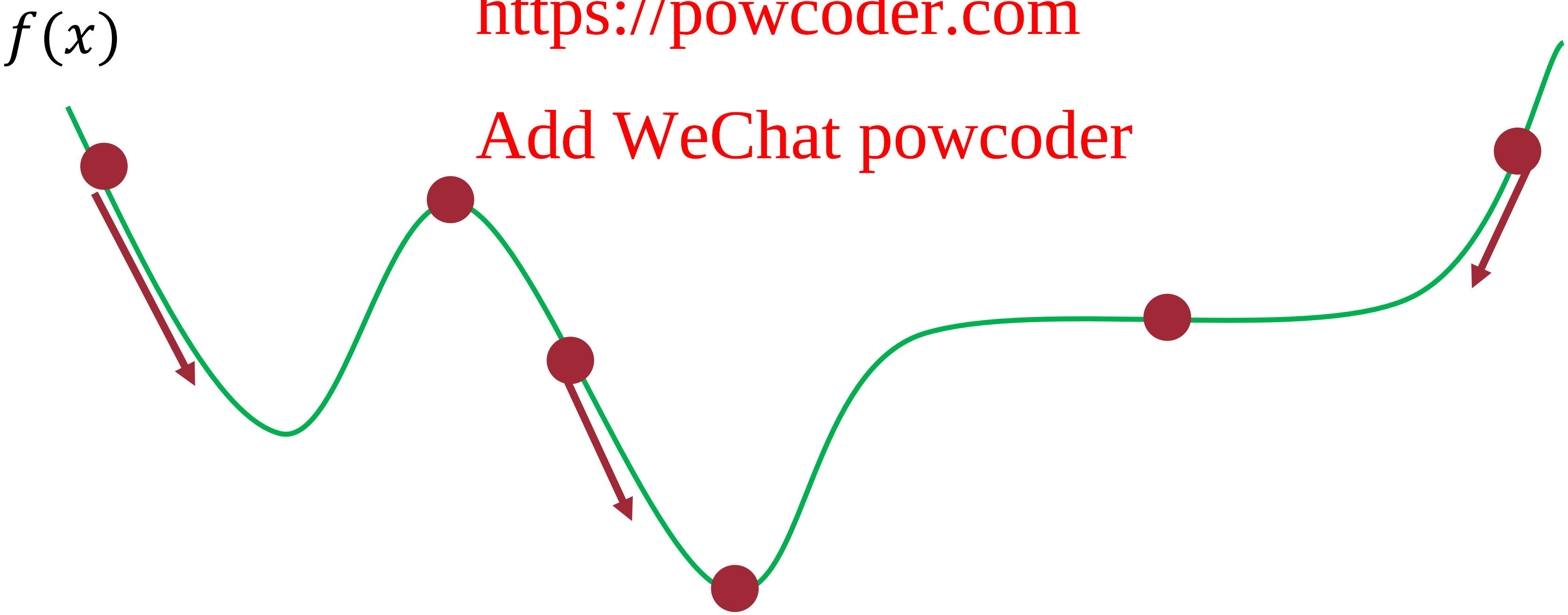
Gradient Method

gradient (not constraint)

$$\partial \mathbf{x}_k = -[\nabla f(\mathbf{x}_k)]^T = -\mathbf{g}_k$$

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

$$\partial \mathbf{x}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$$



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

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

Gradient descent

Slight modification: add a scale factor to avoid jumping past the minimum:

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

Optimizing the step size α gives us:

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

provided the Hessian is positive-definite

p. 155 of 2nd edition or p. 189 of 3rd edition explains step size optimization

Gradient descent example

Problem: $\min_{\mathbf{x}} \; f = x_1^2 + 2x_1x_2 + 3x_1x_3 + 4x_2^2 + 5x_2x_3 + 6x_3^2$

Gradient & Hessian: $\nabla f = \begin{bmatrix} 2x_1 + 2x_2 + 3x_3 \\ 2x_1 + 8x_2 + 5x_3 \\ 3x_1 + 5x_2 + 12x_3 \end{bmatrix} \qquad \mathbf{H} = \begin{bmatrix} 2 & 2 & 3 \\ 2 & 8 & 5 \\ 3 & 5 & 12 \end{bmatrix}$

Algorithm: $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k), \quad \alpha = \left[\dfrac{\nabla f^T(\mathbf{x}_k) \nabla f(\mathbf{x}_k)}{\nabla f^T(\mathbf{x}_k) \mathbf{H}(\mathbf{x}_k) \nabla f(\mathbf{x}_k)} \right]$

Initial point: $\mathbf{x}_0 = [1 \;\; 1 \;\; 1]^T \qquad f(\mathbf{x}_0) = 21$

k	$\nabla f^T(\mathbf{x}_k)$	α	$\mathbf{x}_{k+1}^T$	$f(\mathbf{x}_{k+1})$
0	$[7, 15, 20]$	0.0615	$[0.569, 0.077, -0.230]$	0.2719
1	$[0.603, 0.607, -0.667]$	0.2932	$[0.393, -0.101, -0.035]$	0.0994
2	$[0.480, -0.194, 0.258]$	0.2338	$[0.280, -0.055, -0.095]$	0.0604
3	$[0.165, -0.357, -0.576]$	0.0772	$[0.268, -0.028, -0.051]$	0.0416
4	$[0.328, 0.060, 0.057]$	0.2254	$[0.194, -0.041, -0.063]$	0.0287

Summary

- The **optimality conditions** can be used to 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, which we didn't get to today…

Acknowledgements

- Much of this material came from Chapter 4 of the textbook, *Principles of Optimal Design*
- Some of the slides and examples came from Drs. Emrah Bayrak, Alex Burnap, and Namwoo Kang at the University of Michigan

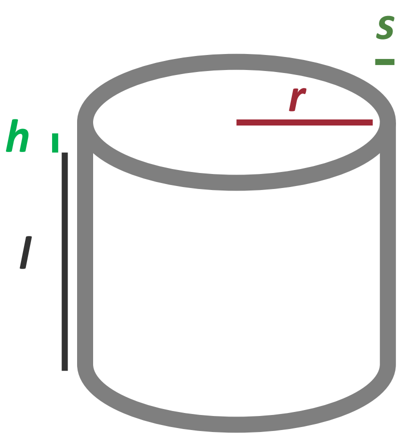
Question

Who would be interesting in a HW1 review session with Amineh sometime on Friday (9/28) or Monday (10/1)?

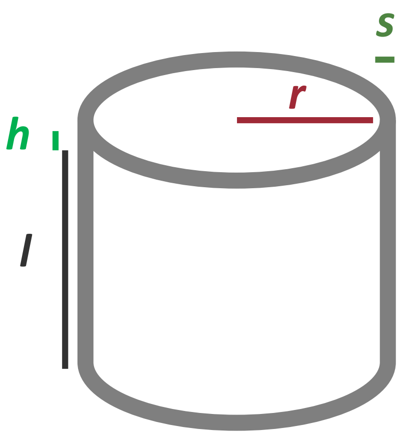
ANSWER: There is interest! Amineh will host a session to review the HW1 solutions on Monday at 11am. Location is TBD, and an announcement will go out on Canvas soon.

Another problem set-up and monotonicity example

Designing an air tank



Air tank problem set-up



minimize the volume of metal:

$$\min_{h,l,r,s} \quad V(h,l,r,s) = \pi\left[(r+s)^2 - r^2\right] l + 2\pi(r+s)^2 h$$

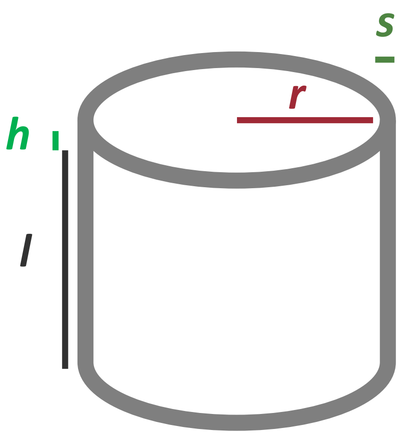
$$\min_{h,l,r,s} \quad V(h,l,r,s) = \pi\left[(2rs+s^2)l + 2(r+s)^2 h\right]$$

Solution

$$V^* = 0$$

$$h^* = 0, l^* = 0, r^* = 0, s^* = 0$$

Example: Air tank



add constraints:

$$\min_{\{h,l,r,s\}\in P^4} V(h,l,r,s) = \pi\left[(2rs+s^2)l + 2(r+s)^2 h\right]$$

subject to

$g_1 : -\pi l r^2 + 2.12(10^7) \leq 0$ minimum inner volume

$g_2 : -h/r + 130(10^{-3}) \leq 0$ ASME code ratio limit

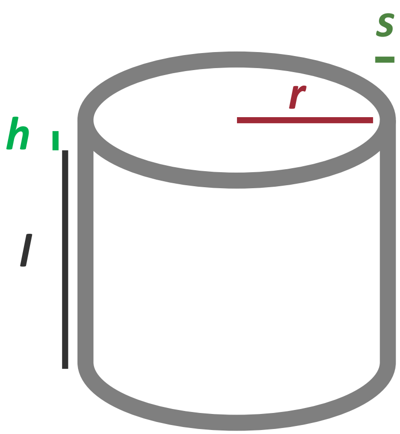
$g_3 : -s/r + 9.59(10^{-3}) \leq 0$ ASME code ratio limit

$g_4 : -l + 10 \leq 0$ allow space to attach nozzles

$g_5 : r + s - 150 \leq 0$ space limitation

Okay, what's next?

Example: Air tank



monotonicity analysis:

$$\min_{\{h,l,r,s\}\in P^4} \quad V(h^+,l^+,r^+,s^+) = \pi\left[(2rs+s^2)l + 2(r+s)^2 h\right]$$

subject to

$$g_1(l^-,r^-) = -\pi l r^2 + 2.12(10^7) \leq 0$$

$$g_2(h^-,r^+) = -h/r + 130(10^{-3}) \leq 0$$

$$g_3(s^-,r^+) = -s/r + 9.59(10^{-3}) \leq 0$$

$$g_4(l^-) = -l + 10 \leq 0$$

$$g_5(r^+,s^+) = r + s - 150 \leq 0$$

MP1: Every increasing variable (in the objective) is bounded below by at least one non-increasing active constraint

Okay, what's next?

Example: Air tank

Remove critical constraints:

$$\min_{\{h,l,r,s\}\in P^4} \quad V(h^+,l^+,r^+,s^+) = \pi\left[(2rs+s^2)l + 2(r+s)^2 h\right]$$

subject to

$$g_1(l^-,\underline{r}^-) = -\pi r^2 l + 2.12(10^7) \leq 0$$

$$g_2(\underline{h}^-,r^+) = -h/r + 130(10^{-3}) \leq 0$$

$$g_3(\underline{s}^-,r^+) = -s/r + 9.59(10^{-3}) \leq 0$$

$$g_4(l^-) = -l + 10 \leq 0$$

$$g_5(r^+,s^+) = r + s - 150 \leq 0$$

Solve each for variable

$$r = \sqrt{\frac{2.12(10^7)}{\pi l}}$$

$$h = 130(10^{-3})r$$

$$s = 9.59(10^{-3})r$$

Plug into objective and constraints

$$\min_{l\in P} \quad V(l^-) = \pi\left[130.1(10^3) + 4.647(10^9)l^{-3/2}\right]$$

subject to

$$g_4(l^-) = -l + 10 \leq 0$$

$$g_5(l^-) = -l + 306 \leq 0$$

Infimum at $V^* = 130.1(10^3)\pi$, where $l^* = \infty$: **<u>not well bounded</u>**