- Changes:
  - no AK office hours today
  - very short AK office hours Thu
  - no AK office hours Thu / Luke teaching
  - no class Nov 8
- 4-credit assignment I posted
- Exam 1b 4 ongoing

\[ A = \begin{bmatrix} \end{bmatrix} \]

\[ A x = u (v^T x) \]

\[ = U V^T \]
$f'(x^*) = 0$

unimodal:

$f(x^* + h) = f(x^*) + f''(x^*) h^2 / 2$
Golden Section Search

Suppose we have an interval with $f$ unimodal:

Would like to maintain unimodality.

Pick $x_1, x_2$

1. If $f(x_1) > f(x_2)$, reduce to $(x_1, b)$
2. If $f(x_1) \leq f(x_2)$, reduce to $(a, x_1)$
Golden Section Search: Efficiency

Where to put \( x_1, x_2 \)?

\[
x_1 = a + (1 - \phi^2) (b - a) \\
x_2 = a + \phi^2 (b - a)
\]

Demo: Golden Section Proportions [cleared]

\[
\phi^2 = 2 - 1 \quad \phi = (\sqrt{5} - 1)/2 \\
\text{golden section search}
\]

Convergence rate?

Liwen
Newton’s Method

Reuse the Taylor approximation idea, but for optimization.

\[ f(x + h) \approx f(x) + f'(x) \cdot h + f''(x) \cdot \frac{h^2}{2} = \hat{f}(h) \]

\[ f'(h) = f'(x) + f''(x) \cdot h = 0 \]

\[ h = -\frac{f'(x)}{f''(x)} \]

\[ x_0 = \text{(starting guess)} \]

\[ x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)} \]

**Demo:** Newton’s Method in 1D [cleared]
Steepest Descent/Gradient Descent

Given a scalar function $f : \mathbb{R}^n \to \mathbb{R}$ at a point $x$, which way is down?

**Direction of steepest descent:** $- \nabla f$

- $x_0$ = (starting guess)
- $s_n = -\nabla f(x_n)$
- Use a line search to choose $\alpha$ to minimize the function $\alpha \to f(x_n + \alpha s_n)$
- $x_{n+1} = x_n + \alpha_n s_n$

**Demo:** Steepest Descent [cleared] (Part 1)
Steepest Descent: Convergence

Consider quadratic model problem:

\[ f(x) = \frac{1}{2} x^T A x + c^T x \]

where \( A \) is SPD. (A good model of \( f \) near a minimum.)

Define error \( e_u = x_{u+1} - x_u \)

\[ \| e_{u+1} \|_A = \sqrt{e_{u+1}^T A e_{u+1}} \]

\[ = \frac{\sigma_{\max}(A) - \sigma_{\min}(A)}{\sigma_{\max}(A) + \sigma_{\min}(A)} \| e_u \|_A \]

\[ = \frac{\kappa(A) - 1}{\kappa(A) + 1} \| e_u \|_A \]
Hacking Steepest Descent for Better Convergence

Extrapolation methods:

\[ x_{k+1} = x_k - \alpha_k \nabla f(x_k) + \beta_k (x_k - x_{k-1}) \]

Heavy ball method:

\[ \alpha_k = \alpha \quad \beta_k = \beta \]

Demo: Steepest Descent [cleared] (Part 2)
What is stochastic gradient descent (SGD)?

\[ f(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{x}) \]

- **Batch**: \[ \mathbf{x}_{n+1} = \mathbf{x}_n - \alpha \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\mathbf{x}_n) \]
- **Mini-batch**: a subset of the data used in each iteration of the learning algorithm.
- **ADAM**: moving averages of the gradient and the square of the gradient.
Conjugate Gradient Methods

Can we optimize in \textit{the space spanned} by the last two step directions?

\[
(x_n, \beta_n) = \arg\min_{\alpha, \beta} \left[ f(x_n - \alpha \nabla f(x_n) + \beta (x_n - x_{n-1})) \right]
\]

**Demo:** Conjugate Gradient Method [cleared]
Nelder-Mead Method

Idea: 

**Demo:** Nelder-Mead Method [cleared]